$99

TRSDOS/LS - DOS 6.x

"THE SOURCE"

```
LD      A,(BUFFER$+1)    ;P/u buffer hi-order addr
LD      D,A
LD      BC,13            ;Move name/ext into dest
LDIR
LD      D,(IY+9)         ;P/u dir cyl of dest
POP     BC               ;Rcvr DEC of source
PUSH    BC
LD      A,B              ;Calc dir sector for
AND     1FH              ;  source SYS module
ADD     A,2
LD      E,A
LD      HL,(BUFFER$)     ;P/u buffer ptr for dest
CALL    WRSYS            ;Write the dir to dest
LD      A,18             ;Init "Dir write error
JP      NZ,EXIT3         ;  and quit on bad write

The HIT entries were transferred prior

POP     BC               ;Rcvr DEC of source
PUSH    BC
LD      A,B              ;Test for SYSØ
CP      2
JP      NZ,DOFILØ        ;Bypass if not SYSØ
CALL    PMTSRC           ;Prompt source
IF      @MOD4
LD      B,16             ;Init to xfer BOOT track
LD      DE,Ø             ;Init track Ø, sector Ø
ENDIF
IF      @MOD2
LD      DE,(PROTSEC)     ;Get sysinfo sector
LD      A,D
OR      A
LD      B,5
```

LOGICAL SYSTEMS INC.

Volume 2

The LIBRARIES

"THE SOURCE"

CONTENTS

Introduction

APPEND - See COPY
ATTRIB ..... Page 1
AUTO ..... Page 19
BUILD ..... Page 27
CAT - See DIR
CLS - See LIB
COPY ..... Page 39
CREATE ..... Page 73
DATE ..... Page 83
DEBUG ..... Page 99
DEVICE ..... Page 109
DIR ..... Page 125
DO ..... Page 175
DUMP ..... Page 197
FILTER - See SET
FORMS ..... Page 205
FREE ..... Page 227
LIB ..... Page 249
LINK ..... Page 255
LIST ..... Page 261
LOAD ..... Page 279
MEMORY ..... Page 287
PURGE ..... Page 301
REMOVE ..... Page 319
RENAME ..... Page 327
RESET ..... Page 337
ROUTE ..... Page 345
RUN - See LOAD
SET ..... Page 355
SETCOM ..... Page 363
SETKI ..... Page 387
SPOOL ..... Page 397
SYSGEN ..... Page 427
SYSTEM ..... Page 443
TIME - See DATE
TOF - See LIB
VERIFY - See DEBUG

Appendix

LOGICAL SYSTEMS INC.

8970 North 55th Street
P.O. Box 23956
Milwaukee, WI 53223

Introduction to Volume Two

This is volume two of three in the set of commented source code listings for LS-DOS/TRSDOS 6.2, as assembled for the TRS-8Ø Model 4/4P computer. This volume contains the Library commands found in SYS6/SYS, SYS7/SYS, and SYS8/SYS.

The individual commands will be presented in alphabetical order without regard to their particular /SYS module. The lead-in page for each command will give the /SYS file containing the command, the ISAM number assigned to the command, and the names of the different commands (if more than one) that the code contains. A symbol table listing will follow each assembly listing.

The SVC macro file used during assembly of the utilities will be listed in Appendix A, along with the equate files generated during assembly of the resident part of the operating system.

This book should by no means be considered a tutorial on assembly language or on the workings of the LS-DOS/TRSDOS operating system. It is only the commented source code used to assemble the system Libraries. It can be used for reference purposes and to view examples of different program structures. It is not meant to replace the normal technical reference manual available from the computer manufacturer.

This product is sold on an as-is basis, and is totally unsupported by Logical Systems, Inc. No questions regarding any aspect of the source code will be answered by LSI customer or technical support. Support for LS-DOS users is provided through their OEM dealer. Support for TRSDOS 6.x is provided by Tandy Corporation. Comments or suggestions may be sent to Logical Systems, Inc. in care of the Source Code Technical Editor, but correspondence concerning these comments will not be made.

Command: ATTRIB

Library: SYS7/SYS

ISAM # : 51H

```
                    00100 ;LBATTRIB/ASM - ATTRIB Command
0000                00110         TITLE   <ATTRIB - LS-DOS 6.2>
                    00120 ;
000D                00130 CR      EQU     13
4296                00140 BLNKMPW EQU     4296H
42E0                00150 PASSWORD        EQU     42E0H
                    00160 ;
0000                00170 *GET    SVCMAC:3                ;SVC Macro equivalents
                    00010 ;SVCMAC/ASM - LS-DOS Version VI
                    00020 *LIST   OFF
                    03900 *LIST   ON
                    00180 ;
2400                00190         ORG     2400H
                    00200 ;
                    00210 ATTRIB
2400 ED733124       00220         LD      (SAVESP+1),SP   ;Save stack pointer
2404 CD3724         00230         CALL    ATTRIB1         ;Call attrib code
2407 210000         00240         LD      HL,0            ;Set no error
240A 1824           00250         JR      SAVESP          ;P/u stack & return
                    00260 ;
                    00270 ;       I/O Error Handling
                    00280 ;
240C FE3F           00290 IOERR   CP      63              ;Extended error?
240E 281A           00300         JR      Z,EXTERR
2410 6F             00310         LD      L,A             ;Error # to HL
2411 2600           00320         LD      H,0
2413 F6C0           00330         OR      0C0H            ;Abbrev & return
2415 4F             00340         LD      C,A
2416                00350         @@ERROR
2416 3E1A           00001         LD      A,26
2418 EF             00002         RST     40
2419 1815           00360         JR      SAVESP          ;P/u Stack & RETurn
                    00370 ;
                    00380 ;       Internal Error Message Handling
                    00390 ;
241B 21FA28         00400 ABORT   LD      HL,ABORT$       ;"Command aborted
241E DD             00410         DB      0DDH
241F 210A28         00420 SPCREQ  LD      HL,SPCREQ$      ;"File spec required
2422 DD             00430         DB      0DDH
2423 211D28         00440 NOTDUN  LD      HL,NOTDUN$      ;"Specifications req.
2426 DD             00450         DB      0DDH
2427 213528         00460 ATBERR  LD      HL,ATBERR$      ;"Attr. specification error
242A                00470 EXTERR  @@LOGOT
                    00003         IFEQ    00H,1
                    00004         LD      HL,
                    00005         ENDIF
242A 3E0C           00006         LD      A,12
242C EF             00007         RST     40
242D 21FFFF         00480         LD      HL,-1           ;Set abort error
2430 310000         00490 SAVESP  LD      SP,$-$          ;Reload stack pointer
2433                00500         @@CKBRKC                ;Clean up <BREAK>
2433 3E6A           00008         LD      A,106
2435 EF             00009         RST     40
2436 C9             00510         RET
                    00520 ;
                    00530 ;       ATTRIB1 - Set Attributes of a file/disk
                    00540 ;
                    00550 ATTRIB1
2437 113F29         00560         LD      DE,FCB          ;Check filespec or drive #
243A                00570         @@FSPEC
243A 3E4E           00010         LD      A,78
```

```
243C EF        00011          RST     40
243D C2DB25    00580          JP      NZ,PROT          ;Assume drive # if error
2440 1A        00590          LD      A,(DE)           ;Cannot be a device
2441 FE2A      00600          CP      '*'
2443 CA1F24    00610          JP      Z,SPCREQ
2446           00620          @@FLAGS                  ;Get flag table pointer
2446 3E65      00012          LD      A,101
2448 EF        00013          RST     40
2449 E5        00630          PUSH    HL               ;Save cmdline ptr
244A 21002A    00640          LD      HL,BUFFER        ;Use local buffer
244D 45        00650          LD      B,L              ;Open the file
244E FDCB12C6  00660          SET     0,(IY+'S'-'A')   ;Don't set file open bit
2452           00670          @@OPEN
2452 3E3B      00014          LD      A,59
2454 EF        00015          RST     40
2455 E1        00680          POP     HL               ;Rcvr cmdline ptr
2456 C20C24    00690          JP      NZ,IOERR         ;Jump if non-existant
2459 3A4029    00700          LD      A,(FCB+1)        ;P/u protection
245C E607      00710          AND     7                ;Mask other bits
245E 3E25      00720          LD      A,25H            ;Init for access denied
2460 C20C24    00730          JP      NZ,IOERR         ;Jump if no can do
2463 AF        00740          XOR     A
2464 32A125    00750          LD      (PRMCOD+1),A     ;Init prot to 0
               00760 ;
               00770 ;        Convert command line to upper case
               00780 ;
2467 E5        00790          PUSH    HL               ;Save cmdline ptr
2468 7E        00800 ATT0     LD      A,(HL)           ;  & cvrt lc to UC
2469 FE0D      00810          CP      CR
246B 2811      00820          JR      Z,ATT02          ;Done if CR
246D FE03      00830          CP      3                ;  ETX
246F 280D      00840          JR      Z,ATT02
2471 FE61      00850          CP      'a'              ;Not lc?
2473 3806      00860          JR      C,ATT01
2475 FE7B      00870          CP      'z'+1
2477 3002      00880          JR      NC,ATT01
2479 CBAE      00890          RES     5,(HL)           ;<a-z> to <A-Z>
247B 23        00900 ATT01    INC     HL               ;Bump to next char
247C 18EA      00910          JR      ATT0             ;Loop
               00920 ;
247E E1        00930 ATT02    POP     HL               ;Rcvr orig cmdline ptr
               00940 ;
               00950 ;        Scan command line for parameters
               00960 ;
247F 7E        00970 ATT1     LD      A,(HL)           ;Scan for start of parm
2480 FE28      00980          CP      '('              ;There yet?
2482 2807      00990          JR      Z,ATT2           ;Jump if so
2484 FE20      01000          CP      ' '              ;Ignore spaces
2486 2005      01010          JR      NZ,ATT3          ;Assume parm on dif char
2488 23        01020          INC     HL
2489 18F4      01030          JR      ATT1
248B 23        01040 ATT2     INC     HL               ;Bump past '('
248C 7E        01050          LD      A,(HL)
248D FE49      01060 ATT3     CP      'I'              ;Ck for INV
248F 2870      01070          JR      Z,DOINV
2491 FE56      01080          CP      'V'              ;Ck for VIS
2493 287E      01090          JR      Z,DOVIS
2495 FE55      01100          CP      'U'              ;Ck for USER
2497 CA2525    01110          JP      Z,DOUSE
249A FE4F      01120          CP      'O'              ;Ck for OWNER
249C CA3A25    01130          JP      Z,DOOWN
```

```
249F FE5Ø     Ø114Ø          CP      'P'             ;Ck for PROT
24A1 C22724   Ø115Ø          JP      NZ,ATBERR       ;Err if none of the above
              Ø116Ø ;
              Ø117Ø ;        Process PROT=parm
              Ø118Ø ;
24A4 CD6925   Ø119Ø          CALL    PRSFLD          ;Parse field
24A7 CA2724   Ø12ØØ          JP      Z,ATBERR        ;Error if end of line
24AA E5       Ø121Ø          PUSH    HL              ;Save ptr to next char
24AB Ø6Ø8     Ø122Ø          LD      B,8             ;Init for 8 prots
24AD ED5B3329 Ø123Ø          LD      DE,(PSWDBUF)    ;P/u 1st 2 chars
24B1 215328   Ø124Ø          LD      HL,PROTS$       ;Pt to various prots
24B4 7E       Ø125Ø DOPRØ1   LD      A,(HL)          ;P/u 1st prot char
24B5 23       Ø126Ø          INC     HL              ;Bump pointer
24B6 BB       Ø127Ø          CP      E               ;Does 1st match?
24B7 CCC124   Ø128Ø          CALL    Z,DOPRØ2        ;Check 2nd if 1st OK
24BA 23       Ø129Ø          INC     HL              ;Bump to next
24BB 1ØF7     Ø13ØØ          DJNZ    DOPRØ1          ;Loop for all 8
24BD E1       Ø131Ø          POP     HL              ;Stack integrity
24BE C32724   Ø132Ø          JP      ATBERR          ;Abort if no match
              Ø133Ø ;
              Ø134Ø ;        Check 2nd prot= char for match
              Ø135Ø ;
24C1 7E       Ø136Ø DOPRØ2   LD      A,(HL)          ;P/u 2nd table char
24C2 BA       Ø137Ø          CP      D               ;Match user's entry?
24C3 CØ       Ø138Ø          RET     NZ              ;Go back if not
24C4 F1       Ø139Ø          POP     AF              ;Pop the ret addr
24C5 78       Ø14ØØ          LD      A,B             ;Calculate which prot was
24C6 3D       Ø141Ø          DEC     A               ;  entered by the user
24C7 2814     Ø142Ø          JR      Z,DOPRØ3        ;Jump on PROT=FU
24C9 FEØ5     Ø143Ø          CP      5               ;REname, REad, REmove?
24CB 2Ø1Ø     Ø144Ø          JR      NZ,DOPRØ3       ;Go if none of the above
24CD 3A3529   Ø145Ø          LD      A,(PSWDBUF+2)   ;P/u user's 3rd char
24DØ FE4E     Ø146Ø          CP      'N'             ;Was it 'N'?
24D2 3EØ2     Ø147Ø          LD      A,2             ;Init for REname
24D4 28Ø7     Ø148Ø          JR      Z,DOPRØ3        ;Go if REName
24D6 3D       Ø149Ø          DEC     A               ;Else init to REMove
24D7 FE4D     Ø15ØØ          CP      'M'             ;  & test entry
24D9 28Ø2     Ø151Ø          JR      Z,DOPRØ3
24DB 3EØ5     Ø152Ø          LD      A,5             ;  else assume REAd
24DD 32A925   Ø153Ø DOPRØ3   LD      (PROTLVL+1),A   ;Stuff protection level
24EØ E1       Ø154Ø          POP     HL              ;Rcvr INBUF$ pointer
24E1 Ø6Ø1     Ø155Ø          LD      B,1             ;Init to show PROT given
24E3 7E       Ø156Ø DOPRØ4   LD      A,(HL)          ;P/u next parm
24E4 FE22     Ø157Ø          CP      '"'             ;Closing quote on last?
24E6 2ØØ1     Ø158Ø          JR      NZ,DOPRØ5       ;Go if something else
24E8 23       Ø159Ø          INC     HL              ;Ignore closing quote
24E9 3AA125   Ø16ØØ DOPRØ5   LD      A,(PRMCOD+1)    ;P/u parm test bits
24EC BØ       Ø161Ø          OR      B               ;Merge PROT entered
24ED 32A125   Ø162Ø          LD      (PRMCOD+1),A    ;Restuff parm test bits
24FØ 7E       Ø163Ø          LD      A,(HL)          ;P/u next char
24F1 FEØD     Ø164Ø          CP      CR              ;End of line?
24F3 28Ø2     Ø165Ø          JR      Z,$+4           ;Go on end-of-line
24F5 FE29     Ø166Ø          CP      ')'             ;End of parms?
24F7 CA9525   Ø167Ø          JP      Z,UPDDIR        ;Go on end of parms
24FA FE2C     Ø168Ø          CP      ','             ;More parms?
24FC 288D     Ø169Ø          JR      Z,ATT2          ;Loop on more parms
24FE C32724   Ø17ØØ          JP      ATBERR          ;Exit on wrong char
              Ø171Ø ;
              Ø172Ø ;        Process INV parm
              Ø173Ø ;
25Ø1 CD6925   Ø174Ø DOINV    CALL    PRSFLD          ;Parse parm
```

```
25Ø4 C22724   Ø175Ø          JP      NZ,ATBERR       ;Go on parm error
25Ø7 3AB125   Ø176Ø          LD      A,(IVCOD+1)     ;Set bit 3 to indicate
25ØA F6Ø8     Ø177Ø          OR      8               ;  that INV given
25ØC 32B125   Ø178Ø          LD      (IVCOD+1),A
25ØF Ø6Ø8     Ø179Ø          LD      B,8             ;Show vis/inv done
2511 18D6     Ø18ØØ          JR      DOPRØ5          ;Merge w/prev. parms
              Ø181Ø ;
              Ø182Ø ;        Process VIS parm
              Ø183Ø ;
2513 CD6925   Ø184Ø DOVIS    CALL    PRSFLD          ;Parse parm
2516 C22724   Ø185Ø          JP      NZ,ATBERR       ;Quit on parm error
2519 3AB125   Ø186Ø          LD      A,(IVCOD+1)     ;Strip bit 3
251C E6F7     Ø187Ø          AND     ØF7H
251E 32B125   Ø188Ø          LD      (IVCOD+1),A
2521 Ø6Ø8     Ø189Ø          LD      B,8             ;Show vis/inv done
2523 18C4     Ø19ØØ          JR      DOPRØ5          ;Merge w/prev. parms
              Ø191Ø ;
              Ø192Ø ;        Process USER parm
              Ø193Ø ;
2525 CD6925   Ø194Ø DOUSE    CALL    PRSFLD          ;Parse parm
2528 CA2724   Ø195Ø          JP      Z,ATBERR        ;Quit on parm error
252B E5       Ø196Ø          PUSH    HL              ;Save cmdline ptr
252C 113329   Ø197Ø          LD      DE,PSWDBUF
252F CD4727   Ø198Ø          CALL    DOHASH          ;Hash the password
2532 223D29   Ø199Ø          LD      (HASHBUF+2),HL  ;Set into position
2535 E1       Ø2ØØØ          POP     HL
2536 Ø6Ø2     Ø2Ø1Ø          LD      B,2             ;Show user done
2538 18A9     Ø2Ø2Ø          JR      DOPRØ4          ;Merge w/prev. parms
              Ø2Ø3Ø ;
              Ø2Ø4Ø ;        Process OWNER parm
              Ø2Ø5Ø ;
253A CD6925   Ø2Ø6Ø DOOWN    CALL    PRSFLD          ;Parse parm
253D CA2724   Ø2Ø7Ø          JP      Z,ATBERR        ;Quit on parm error
254Ø E5       Ø2Ø8Ø          PUSH    HL              ;Save cmdline ptr
2541 113329   Ø2Ø9Ø          LD      DE,PSWDBUF
2544 CD4727   Ø21ØØ          CALL    DOHASH          ;Hash the password
2547 223B29   Ø211Ø          LD      (HASHBUF),HL
254A E1       Ø212Ø          POP     HL
254B Ø6Ø4     Ø213Ø          LD      B,4             ;Show OWNER done
254D 1894     Ø214Ø          JR      DOPRØ4          ;Merge w/prev. parms
              Ø215Ø ;
              Ø216Ø ;        Transfer the field, 1st char alpha
              Ø217Ø ;
254F 7E       Ø218Ø XSPEC8A  LD      A,(HL)          ;P/u a filespec character
255Ø 23       Ø219Ø          INC     HL              ;  & 1st test for A-Z
2551 18Ø9     Ø22ØØ          JR      XSPEC1Ø
2553 7E       Ø221Ø XSPEC9   LD      A,(HL)          ;P/u a filespec character
2554 23       Ø222Ø          INC     HL              ;Advance to next one
2555 FE3Ø     Ø223Ø          CP      'Ø'             ;Check for Ø-9
2557 D8       Ø224Ø          RET     C               ;Quit if < Ø
2558 FE3A     Ø225Ø          CP      '9'+1
255A 38Ø6     Ø226Ø          JR      C,XSPEC11       ;Go if numeric
255C FE41     Ø227Ø XSPEC1Ø  CP      'A'             ;Check for A-Z
255E D8       Ø228Ø          RET     C
255F FE5B     Ø229Ø          CP      'Z'+1
2561 DØ       Ø23ØØ          RET     NC
2562 12       Ø231Ø XSPEC11  LD      (DE),A          ;Character is valid
2563 13       Ø232Ø          INC     DE              ;Advance to next one
2564 1ØED     Ø233Ø          DJNZ    XSPEC9          ;  & loop
2566 7E       Ø234Ø          LD      A,(HL)          ;P/u following character
2567 23       Ø235Ø          INC     HL
```

```
2568 C9          Ø236Ø          RET                        ;Go home
                 Ø237Ø ;
                 Ø238Ø ;        Parse rest of parm (ignore until separator)
                 Ø239Ø ;
2569 23          Ø24ØØ PRSFLD   INC      HL
256A 7E          Ø241Ø          LD       A,(HL)            ;Get next char
256B FEØD        Ø242Ø          CP       CR                ;Ret on end of line
256D C8          Ø243Ø          RET      Z
256E FE29        Ø244Ø          CP       ')'               ;Ret on closing paren
257Ø C8          Ø245Ø          RET      Z
2571 FE2C        Ø246Ø          CP       ','               ;Ret on separator
2573 C8          Ø247Ø          RET      Z
2574 FE3D        Ø248Ø          CP       '='               ;Assignment operator?
2576 2ØF1        Ø249Ø          JR       NZ,PRSFLD         ;Loop if not
2578 23          Ø25ØØ          INC      HL
2579 7E          Ø251Ø          LD       A,(HL)
257A FE22        Ø252Ø          CP       '"'               ;Is quote there?
257C 2ØØ1        Ø253Ø          JR       NZ,$+3
257E 23          Ø254Ø          INC      HL                ;Bypass the quote
257F 113329      Ø255Ø          LD       DE,PSWDBUF
2582 Ø6Ø8        Ø256Ø          LD       B,8
2584 D5          Ø257Ø          PUSH     DE
2585 C5          Ø258Ø          PUSH     BC
2586 3E2Ø        Ø259Ø          LD       A,' '             ;Space out the buffer
2588 12          Ø26ØØ PRSFØ1   LD       (DE),A
2589 13          Ø261Ø          INC      DE
258A 1ØFC        Ø262Ø          DJNZ     PRSFØ1
258C C1          Ø263Ø          POP      BC
258D D1          Ø264Ø          POP      DE
258E CD4F25      Ø265Ø          CALL     XSPEC8A           ;Transfer the spec
2591 2B          Ø266Ø          DEC      HL
2592 F6Ø1        Ø267Ø          OR       1                 ;Show got a parm
2594 C9          Ø268Ø          RET
                 Ø269Ø ;
                 Ø27ØØ ;        Routine updates file's directory data
                 Ø271Ø ;
2595 ED4B4529    Ø272Ø UPDDIR   LD       BC,(FCB+6)        ;P/u drive & DEC
2599             Ø273Ø          @@DIRRD                    ;Read its directory
2599 3E57        ØØØ16          LD       A,87
259B EF          ØØØ17          RST      4Ø
259C C2ØC24      Ø274Ø          JP       NZ,IOERR          ;Quit on read error
259F 7E          Ø275Ø          LD       A,(HL)            ;P/u attributes byte
25AØ 16ØØ        Ø276Ø PRMCOD   LD       D,$-$             ;P/u parm test bits
25A2 CB42        Ø277Ø          BIT      Ø,D               ;Was Prot entered?
25A4 28Ø4        Ø278Ø          JR       Z,UPDIR1          ;Jump if not
25A6 E6F8        Ø279Ø          AND      ØF8H              ;Rmv prot level
25A8 F6ØØ        Ø28ØØ PROTLVL  OR       Ø                 ;Merge new prot level
25AA CB5A        Ø281Ø UPDIR1   BIT      3,D               ;Was Inv or Vis entered?
25AC 28Ø4        Ø282Ø          JR       Z,UPDIR2          ;Bypass if not
25AE E6F7        Ø283Ø          AND      ØF7H              ;Remove any vis/inv
25BØ F6ØØ        Ø284Ø IVCOD    OR       Ø                 ;Merge new inv/vis
25B2 77          Ø285Ø UPDIR2   LD       (HL),A            ;  & update dir rec
25B3 7D          Ø286Ø          LD       A,L               ;Pt to owner pswd
25B4 C61Ø        Ø287Ø          ADD      A,16
25B6 6F          Ø288Ø          LD       L,A
25B7 CB52        Ø289Ø          BIT      2,D               ;Was OWN parm entered?
25B9 28ØA        Ø29ØØ          JR       Z,UPDIR3          ;Bypass if not
25BB 3A3B29      Ø291Ø          LD       A,(HASHBUF)       ;Xfer new hashed pswd
25BE 77          Ø292Ø          LD       (HL),A            ;  into the directory
25BF 3A3C29      Ø293Ø          LD       A,(HASHBUF+1)     ;  OWNER psw position
25C2 23          Ø294Ø          INC      HL
```

```
25C3 77        02950          LD      (HL),A
25C4 2B        02960          DEC     HL
25C5 23        02970 UPDIR3   INC     HL              ;Bypass own pswd field
25C6 23        02980          INC     HL
25C7 CB4A      02990          BIT     1,D             ;Was USER parm entered?
25C9 2809      03000          JR      Z,UPDIR4        ;Bypass if not
25CB 3A3D29    03010          LD      A,(HASHBUF+2)   ;Xfer the new hashed pswd
25CE 77        03020          LD      (HL),A          ;  into the directory
25CF 3A3E29    03030          LD      A,(HASHBUF+3)   ;  USER psw position
25D2 23        03040          INC     HL
25D3 77        03050          LD      (HL),A
25D4           03060 UPDIR4   @@DIRWR                 ;Write dir back to disk
25D4 3E58      00018          LD      A,88
25D6 EF        00019          RST     40
25D7 C2C624    03070          JP      NZ,IOERR        ;Abort on write error
25DA C9        03080          RET                     ;Done - return
               03090 ;
               03100 ;        Change attributes of entire disk
               03110 ;
25DB 0E00      03120 PROT     LD      C,0             ;Init for drive 0
25DD 2B        03130          DEC     HL              ;Backup to separator
25DE 7E        03140          LD      A,(HL)          ;Ck for drive entered
25DF FE3A      03150          CP      ':'             ;Colon indicator?
25E1 2018      03160          JR      NZ,PROT01       ;Bypass if not
25E3 23        03170          INC     HL              ;Point to drive #
               03180 ;
               03190 ;        Is the drivespec legal (0-7) ?
               03200 ;
25E4 7E        03210          LD      A,(HL)          ;P/u drive
25E5 D630      03220          SUB     '0'             ;Cvrt to binary
25E7 3804      03230          JR      C,ILLDRVN       ;Less than 0, illegal
25E9 FE08      03240          CP      7+1             ;Greater than 7 ?
25EB 3F        03250          CCF
25EC 4F        03260          LD      C,A
25ED 3E20      03270 ILLDRVN  LD      A,32            ;Init "Illegal Drive #
25EF DA0C24    03280          JP      C,IOERR
               03290 ;
               03300 ;        Drive # is legal - Check it out
               03310 ;
25F2           03320          @@CKDRV                 ;Do a check drive
25F2 3E21      00020          LD      A,33
25F4 EF        00021          RST     40
25F5 3E20      03330          LD      A,32            ;Init "Illegal drive
25F7 C20C24    03340          JP      NZ,IOERR        ;Go if bad
               03350 ;
25FA 23        03360          INC     HL              ;Bump line pointer
25FB 79        03370 PROT01   LD      A,C
25FC 321427    03380          LD      (TSTMPW+1),A    ;Stuff drive for later
25FF           03390          @@GTDCT                 ;Get DCT -> IY
25FF 3E51      00022          LD      A,81
2601 EF        00023          RST     40
2602 110A29    03400          LD      DE,PRMTBL$      ;Get parms
2605           03410          @@PARAM
2605 3E11      00024          LD      A,17
2607 EF        00025          RST     40
2608 C20C24    03420          JP      NZ,IOERR        ;Quit on parm error
260B 3A6626    03430          LD      A,(PPARM+1)     ;Make sure a parm
260E 218926    03440          LD      HL,LPARM+1      ;  was entered
2611 B6        03450          OR      (HL)
2612 219326    03460          LD      HL,UPARM+1
2615 B6        03470          OR      (HL)
```

```
2616 212426   03480          LD      HL,NPARM+1
2619 B6       03490          OR      (HL)
261A CA2324   03500          JP      Z,NOTDUN          ;Quit if none entered
261D CD1327   03510          CALL    TSTMPW            ;Test master password
2620 C20C24   03520          JP      NZ,IOERR          ;Quit on error
2623 110000   03530 NPARM    LD      DE,0              ;P/U Name parm
2626 7A       03540          LD      A,D
2627 B3       03550          OR      E
2628 283B     03560          JR      Z,PPARM           ;Jump if name not entered
262A 218828   03570          LD      HL,PACKNM$        ;Get name into buf1
262D CD4A27   03580          CALL    GMPW1
2630 C20C24   03590          JP      NZ,IOERR          ;Quit on error
2633 21002A   03600          LD      HL,BUFFER         ;HL = new name
2636 11D02B   03610          LD      DE,GATBUF+0D0H    ;Where new name goes
2639 D5       03620          PUSH    DE
263A E5       03630          PUSH    HL
263B 010800   03640          LD      BC,8              ;Name len
263E EDB0     03650          LDIR                      ;Xfer new name
2640 E1       03660          POP     HL
2641 110200   03670          LD      DE,2              ;Trk 0, sect 2
2644 3A1427   03680          LD      A,(TSTMPW+1)      ;P/u drive
2647 4F       03690          LD      C,A
2648          03700          @@RDSEC                   ;Read SIS
2648 3E31     00026          LD      A,49
264A EF       00027          RST     40
264B E1       03710          POP     HL                ;Get Name pointer
264C C20C24   03720          JP      NZ,IOERR          ;Quit if read error
264F C5       03730          PUSH    BC                ;Save drive
2650 11102A   03740          LD      DE,BUFFER+10H     ;Point to where name goes
2653 010800   03750          LD      BC,8
2656 EDB0     03760          LDIR                      ;Xfer new name to SIS
2658 C1       03770          POP     BC                ;Recover drive
2659 110200   03780          LD      DE,2
265C 21002A   03790          LD      HL,BUFFER
265F          03800          @@WRSEC                   ;Write new SIS
265F 3E35     00028          LD      A,53
2661 EF       00029          RST     40
2662 C20C24   03810          JP      NZ,IOERR          ;Quit on write error
2665 110000   03820 PPARM    LD      DE,0              ;Was PW parm used?
2668 7A       03830          LD      A,D
2669 B3       03840          OR      E
266A 280C     03850          JR      Z,PROT02          ;Jump if PW not entered
266C 21A028   03860          LD      HL,NEWMPW$        ;Buffer for new MPW
266F CD3A27   03870          CALL    GETMPW            ;Input the new one
2672 C20C24   03880          JP      NZ,IOERR          ;Quit if bad PW
2675 22CE2B   03890          LD      (GATBUF+0CEH),HL        ;Stuff PW
2678 FD5609   03900 PROT02   LD      D,(IY+9)          ;Dir cl => reg D
267B 3A1427   03910          LD      A,(TSTMPW+1)      ;P/u drive
267E 4F       03920          LD      C,A
267F CDCD27   03930          CALL    GATWR             ;Write sector 0 from buf
2682 C20C24   03940          JP      NZ,IOERR          ;Jump on write error
2685 2ACE2B   03950          LD      HL,(GATBUF+0CEH)        ;P/u pack MPW
              03960 ;
              03970 ;        Check on Lock or Unlock
              03980 ;
2688 010000   03990 LPARM    LD      BC,0              ;Lock parm used?
268B 78       04000          LD      A,B
268C B1       04010          OR      C
268D 119642   04020          LD      DE,BLNKMPW        ;P/u blank MPW for test
2690 2007     04030          JR      NZ,PROT03         ;Jump if LOCK entered
2692 010000   04040 UPARM    LD      BC,0              ;Unlock parm used?
```

```
2695 78        Ø4Ø5Ø          LD      A,B
2696 B1        Ø4Ø6Ø          OR      C
2697 C8        Ø4Ø7Ø          RET     Z               ;Neither LOCK or UNLOCK
2698 EB        Ø4Ø8Ø          EX      DE,HL           ;Switch New & Test MPW
               Ø4Ø9Ø ;
               Ø41ØØ ;        Lock to pack MPW or unlock pswds to blanks
               Ø411Ø ;
2699 22EA26    Ø412Ø PROTØ3   LD      (REVMPW+1),HL   ;Stuff New MPW
269C ED53E126  Ø413Ø          LD      (THISPW+1),DE   ;Stuff test MPW
26AØ 3A1427    Ø414Ø          LD      A,(TSTMPW+1)    ;P/u drive #
26A3 4F        Ø415Ø          LD      C,A
26A4 FD56Ø9    Ø416Ø          LD      D,(IY+9)        ;Get dir cyl => reg D
26A7 1EØ1      Ø417Ø          LD      E,1             ;Point to HIT
26A9 21ØØ2C    Ø418Ø          LD      HL,HITBUF       ;Point to HIT buffer
26AC           Ø419Ø          @@RDSSC                 ;Read it into buffer
26AC 3E55      ØØØ3Ø          LD      A,85
26AE EF        ØØØ31          RST     4Ø
26AF C2ØC24    Ø42ØØ          JP      NZ,IOERR        ;Quit on read error
26B2 7E        Ø421Ø PROTØ4   LD      A,(HL)          ;P/u a DEC
26B3 B7        Ø422Ø          OR      A
26B4 2842      Ø423Ø          JR      Z,PROTØ9        ;Loop on spare
26B6 45        Ø424Ø          LD      B,L             ;Put DEC in reg B
26B7 C5        Ø425Ø          PUSH    BC              ;  & save it in stack
26B8 7D        Ø426Ø          LD      A,L             ;Ck if this DEC points
26B9 E6EØ      Ø427Ø          AND     ØEØH            ;  to same dir sector as
26BB 6F        Ø428Ø          LD      L,A             ;  the previous DEC
26BC A8        Ø429Ø          XOR     B
26BD FEFF      Ø43ØØ PROTØ5   CP      ØFFH            ;1st time, no DEC
26BF 28ØD      Ø431Ø          JR      Z,PROTØ6        ;Jump if the same
26C1 32BE26    Ø432Ø          LD      (PROTØ5+1),A    ;Save it for testing
26C4           Ø433Ø          @@DIRRD                 ;Read this dir sector
26C4 3E57      ØØØ32          LD      A,87
26C6 EF        ØØØ33          RST     4Ø
26C7 C2ØC24    Ø434Ø          JP      NZ,IOERR        ;Quit on read error
26CA 7C        Ø435Ø          LD      A,H             ;Set hi-order SBUFF$
26CB 32CF26    Ø436Ø          LD      (PROTØ6+1),A
26CE 26ØØ      Ø437Ø PROTØ6   LD      H,$-$           ;Point to buf hi-order
26DØ 7E        Ø438Ø          LD      A,(HL)          ;P/u type code
26D1 E6F8      Ø439Ø          AND     ØF8H            ;Remove protection
26D3 FE1Ø      Ø44ØØ          CP      1ØH             ;Jump if INV, SYS, FXDE
26D5 2Ø1D      Ø441Ø          JR      NZ,PROTØ8
26D7 7D        Ø442Ø          LD      A,L             ;Point to password fields
26D8 C61Ø      Ø443Ø          ADD     A,16
26DA 6F        Ø444Ø          LD      L,A
26DB D5        Ø445Ø          PUSH    DE              ;Save reg DE
26DC E5        Ø446Ø PROTØ7   PUSH    HL              ;Save pointer to OWNER
26DD 5E        Ø447Ø          LD      E,(HL)          ;P/u owner MPW
26DE 23        Ø448Ø          INC     HL
26DF 56        Ø449Ø          LD      D,(HL)
26EØ 21ØØØØ    Ø45ØØ THISPW   LD      HL,$-$          ;P/u test MPW & see
26E3 AF        Ø451Ø          XOR     A               ;  if this one matches
26E4 ED52      Ø452Ø          SBC     HL,DE
26E6 E1        Ø453Ø          POP     HL              ;Restore ptr to OWN
26E7 2ØØA      Ø454Ø          JR      NZ,PROTØ7B      ;Don't change if diff
26E9 11ØØØØ    Ø455Ø REVMPW   LD      DE,$-$          ;  else p/u new MPW
26EC 73        Ø456Ø          LD      (HL),E          ;  & insert it
26ED 2C        Ø457Ø          INC     L
26EE 72        Ø458Ø          LD      (HL),D
26EF 2C        Ø459Ø          INC     L
26FØ 73        Ø46ØØ          LD      (HL),E          ;Change USER pw
26F1 2C        Ø461Ø          INC     L
```

```
26F2 72        04620          LD      (HL),D
26F3 D1        04630 PROT07B  POP     DE              ;Restore reg DE
26F4 C1        04640 PROT08   POP     BC              ;Recover DEC
26F5 262C      04650          LD      H,HITBUF<-8     ;Point to HIT hi-order
26F7 68        04660          LD      L,B             ;Stuff HIT lo-order
26F8 7D        04670 PROT09   LD      A,L             ;Point to next entry
26F9 C620      04680          ADD     A,32            ;  for this dir sector
26FB 6F        04690          LD      L,A
26FC 30B4      04700          JR      NC,PROT04       ;Jump if still in same
26FE 3ABE26    04710          LD      A,(PROT05+1)    ;P/u current DEC
2701 AD        04720          XOR     L
2702 2008      04730          JR      NZ,PROT10       ;Jump if different
2704 E5        04740          PUSH    HL
2705           04750          @@DIRWR                 ;Write out this sector
2705 3E58      00034          LD      A,88
2707 EF        00035          RST     40
2708 E1        04760          POP     HL
2709 C20C24    04770          JP      NZ,IOERR        ;Quit on write error
270C 7D        04780 PROT10   LD      A,L             ;Advance to the next
270D 2C        04790          INC     L               ;  directory sector
270E FE1F      04800          CP      1FH             ;At end of disk?
2710 20A0      04810          JR      NZ,PROT04       ;Loop if not
2712 C9        04820          RET                     ;  else go home
               04830 ;
               04840 ;        Routine to test master password for match
               04850 ;
2713 0E00      04860 TSTMPW   LD      C,$-$           ;Init to drive requested
2715 CDCC27    04870          CALL    GATRD           ;Read GAT into GATBUF
2718 C0        04880          RET     NZ              ;Back on error
2719 2ACE2B    04890          LD      HL,(GATBUF+0CEH)
271C 11E042    04900          LD      DE,PASSWORD     ;Password=PASSWORD?
271F AF        04910          XOR     A
2720 ED52      04920          SBC     HL,DE
2722 C8        04930          RET     Z               ;Back if PASSWORD
               04940 ;
               04950 ;        MPW is not "PASSWORD" - check entry match
               04960 ;
2723 110000    04970 MPARM    LD      DE,0            ;P/u MPW string addr
2726 21B828    04980          LD      HL,CURMPW$      ;Init prompt
2729 CD3A27    04990          CALL    GETMPW          ;Hash parm or entry
272C C0        05000          RET     NZ              ;Back on bad PW
272D EB        05010          EX      DE,HL           ;Xfer hashed MPW to DE
272E 2ACE2B    05020          LD      HL,(GATBUF+0CEH)        ;Grab pack MPW &
2731 AF        05030          XOR     A               ;  check if user entered
2732 ED52      05040          SBC     HL,DE           ;  the pack MPW
2734 21D028    05050          LD      HL,BADMPW$      ;Init error pointer
2737 3E3F      05060          LD      A,63            ;Set extended error
2739 C9        05070          RET                     ;Z or NZ
               05080 ;
               05090 ;        Enter SYS2 & hash the password
               05100 ;
273A CD4A27    05110 GETMPW   CALL    GMPW1           ;Get MPW into buffer
273D 2808      05120          JR      Z,DOHASH
273F FE3F      05130          CP      63              ;Extended error?
2741 C0        05140          RET     NZ
2742 21D028    05150          LD      HL,BADMPW$      ;Switch error message
2745 B7        05160          OR      A               ;  to password error
2746 C9        05170          RET
2747 3EE4      05180 DOHASH   LD      A,0E4H          ;Hash password (DE) to HL
2749 EF        05190          RST     28H             ;Ret to what called
               05200 ;
```

```
                    Ø521Ø ;       Routine places a password field into buffer
                    Ø522Ø ;
274A 7A             Ø523Ø GMPW1   LD      A,D             ;Test if user entered MPW
274B B3             Ø524Ø         OR      E
274C 281D           Ø525Ø         JR      Z,GMPW3         ;Prompt if not
274E 3C             Ø526Ø         INC     A               ;  or if no operand
274F 281A           Ø527Ø         JR      Z,GMPW3
                    Ø528Ø ;
                    Ø529Ø ;       Place entered password into buffer
                    Ø53ØØ ;
2751 21ØØ2A         Ø531Ø         LD      HL,BUFFER       ;Point to buffer
2754 E5             Ø532Ø         PUSH    HL
2755 Ø6Ø8           Ø533Ø         LD      B,8             ;Init for 8 chars
2757 1A             Ø534Ø GMPW2   LD      A,(DE)          ;P/u a char
2758 FEØD           Ø535Ø         CP      CR              ;End of line?
275A 282F           Ø536Ø         JR      Z,GMPW4
275C FE2C           Ø537Ø         CP      ','             ;Comma separator?
275E 282B           Ø538Ø         JR      Z,GMPW4
276Ø FE22           Ø539Ø         CP      '"'             ;Closing quote?
2762 2827           Ø54ØØ         JR      Z,GMPW4
2764 13             Ø541Ø         INC     DE              ;Bump input pointer
2765 77             Ø542Ø         LD      (HL),A          ;Transfer character
2766 23             Ø543Ø         INC     HL              ;Bump output pointer
2767 1ØEE           Ø544Ø         DJNZ    GMPW2           ;Loop until done
2769 1825           Ø545Ø         JR      CKMPW
                    Ø546Ø ;
                    Ø547Ø ;       MPW not entered - Prompt & fetch
                    Ø548Ø ;
276B CDF827         Ø549Ø GMPW3   CALL    CKINDO          ;Can't prompt in <DO>
276E CØ             Ø55ØØ         RET     NZ
276F                Ø551Ø         @@DSPLY                 ;Display prompt
                    ØØØ36         IFEQ    ØØH,1
                    ØØØ37         LD      HL,
                    ØØØ38         ENDIF
276F 3EØA           ØØØ39         LD      A,1Ø
2771 EF             ØØØ4Ø         RST     4Ø
2772 CØ             Ø552Ø         RET     NZ
2773 Ø1ØØØ8         Ø553Ø         LD      BC,8<8          ;Init for 8 chars
2776 21ØØ2A         Ø554Ø         LD      HL,BUFFER       ;Point to buffer
2779 E5             Ø555Ø         PUSH    HL
277A                Ø556Ø         @@KEYIN                 ;Get parm input
277A 3EØ9           ØØØ41         LD      A,9
277C EF             ØØØ42         RST     4Ø
277D DA1B24         Ø557Ø         JP      C,ABORT         ;Quit on Break
278Ø EB             Ø558Ø         EX      DE,HL           ;Start pointer to reg DE
2781 26ØØ           Ø559Ø         LD      H,Ø             ;Calculate trailing
2783 68             Ø56ØØ         LD      L,B             ;  spaces needed for MPW
2784 19             Ø561Ø         ADD     HL,DE
2785 3EØ8           Ø562Ø         LD      A,8
2787 9Ø             Ø563Ø         SUB     B
2788 28Ø6           Ø564Ø         JR      Z,CKMPW         ;Go if 8 chars entered
278A 47             Ø565Ø         LD      B,A             ;Set loop count
278B 362Ø           Ø566Ø GMPW4   LD      (HL),' '        ;  and fill to end
278D 23             Ø567Ø         INC     HL              ;  with spaces
278E 1ØFB           Ø568Ø         DJNZ    GMPW4
                    Ø569Ø ;
                    Ø57ØØ ;       Convert (SP) through (SP)+7 to upper case
                    Ø571Ø ;
279Ø E1             Ø572Ø CKMPW   POP     HL              ;Get buffer start
2791 E5             Ø573Ø         PUSH    HL
2792 Ø6Ø8           Ø574Ø         LD      B,8             ;Init loop 8 chars
```

```
2794 7E       Ø5750          LD      A,(HL)          ;P/u 1st char
2795 18ØE     Ø5760          JR      CKMPW2          ;  & check <A-Z>
2797 23       Ø5770 CKMPW1   INC     HL
2798 7E       Ø5780          LD      A,(HL)
2799 FE2Ø     Ø5790          CP      ' '             ;Got to a space?
279B 2823     Ø5800          JR      Z,CKMPW7
279D FE3Ø     Ø5810          CP      'Ø'             ;Less than 'Ø' is error
279F 3823     Ø5820          JR      C,INVNAM
27A1 FE3A     Ø5830          CP      '9'+1           ;<Ø-9> is okay for 2-n
27A3 3812     Ø5840          JR      C,CKMPW3
27A5 FE41     Ø5850 CKMPW2   CP      'A'             ;Less than "A" is error
27A7 381B     Ø5860          JR      C,INVNAM
27A9 FE5B     Ø5870          CP      'Z'+1           ;<A-Z> is okay
27AB 38ØA     Ø5880          JR      C,CKMPW3
27AD FE61     Ø5890          CP      'a'             ;Ck convert to upper
27AF 3813     Ø5900          JR      C,INVNAM
27B1 FE7B     Ø5910          CP      'z'+1
27B3 3ØØF     Ø5920          JR      NC,INVNAM
27B5 CBAE     Ø5930          RES     5,(HL)
27B7 1ØDE     Ø5940 CKMPW3   DJNZ    CKMPW1          ;Loop if more
27B9 D1       Ø5950 CKMPW4   POP     DE              ;Point to buffer start
27BA AF       Ø5960          XOR     A               ;Set Z = good
27BB C9       Ø5970          RET
              Ø5980 ;
27BC 23       Ø5990 CKMPW5   INC     HL
27BD BE       Ø6ØØØ          CP      (HL)            ;No imbedded spaces
27BE 2ØØ4     Ø6Ø1Ø          JR      NZ,INVNAM
27CØ 1ØFA     Ø6Ø2Ø CKMPW7   DJNZ    CKMPW5          ;A space found, now
27C2 18F5     Ø6Ø3Ø          JR      CKMPW4          ;  must be all spaces
27C4 21E828   Ø6Ø4Ø INVNAM   LD      HL,BADNAM$      ;Pt to error string
27C7 3E3F     Ø6Ø5Ø          LD      A,63            ;Init extended error
27C9 B7       Ø6Ø6Ø          OR      A               ;Set NZ
27CA D1       Ø6Ø7Ø          POP     DE
27CB C9       Ø6Ø8Ø          RET
              Ø6Ø9Ø ;
              Ø61ØØ ;        Read or write the granule allocation table
              Ø611Ø ;
27CC F6       Ø612Ø GATRD    DB      ØF6H            ;Set NZ for test
27CD AF       Ø613Ø GATWR    XOR     A               ;Set Z for test
27CE D5       Ø614Ø          PUSH    DE
27CF E5       Ø615Ø          PUSH    HL
27DØ F5       Ø616Ø          PUSH    AF
27D1 FDE5     Ø617Ø          PUSH    IY
27D3          Ø618Ø          @@GTDCT                 ;DCT to reg IY
27D3 3E51     ØØØ43          LD      A,81
27D5 EF       ØØØ44          RST     4Ø
27D6 FD56Ø9   Ø619Ø          LD      D,(IY+9)        ;Get dir track
27D9 FDE1     Ø62ØØ          POP     IY
27DB 21ØØ2B   Ø621Ø          LD      HL,GATBUF
27DE 5D       Ø622Ø          LD      E,L             ;Set to sector Ø
27DF F1       Ø623Ø          POP     AF
27EØ 28Ø7     Ø624Ø          JR      Z,GATRW1        ;Go if GAT write
27E2          Ø625Ø          @@RDSSC
27E2 3E55     ØØØ45          LD      A,85
27E4 EF       ØØØ46          RST     4Ø
27E5 3E14     Ø626Ø          LD      A,14H           ;Init "GAT read error
27E7 18ØC     Ø627Ø          JR      GATRW3
27E9          Ø628Ø GATRW1   @@WRSSC
27E9 3E36     ØØØ47          LD      A,54
27EB EF       ØØØ48          RST     4Ø
27EC 2ØØ3     Ø629Ø          JR      NZ,GATRW2       ;Skip verify if error
```

```
27EE            06300           @@VRSEC                 ;Verify the write
27EE 3E32       00049           LD      A,50
27F0 EF         00050           RST     40
27F1 FE06       06310 GATRW2    CP      6               ;Error 6 expected
27F3 3E15       06320           LD      A,15H           ;Init "GAT write error
27F5 E1         06330 GATRW3    POP     HL
27F6 D1         06340           POP     DE
27F7 C9         06350           RET
                06360 ;
27F8 FDE5       06370 CKINDO    PUSH    IY
27FA            06380           @@FLAGS
27FA 3E65       00051           LD      A,101
27FC EF         00052           RST     40
27FD FDCB126E   06390           BIT     5,(IY+'S'-'A')  ;Ck on DO in effect
2801 FDE1       06400           POP     IY
2803 C8         06410           RET     Z               ;Go back if not in DO
2804 216328     06420           LD      HL,NOINDO$      ;  else set error code
2807 3E3F       06430           LD      A,63            ;Set extended error
2809 C9         06440           RET                     ;  & back with NZ
                06450 ;
                06460 ;         Messages
                06470 ;
280A 46         06480 SPCREQ$ DB        'File spec required',CR
     69 6C 65 20 73 70 65 63
     20 72 65 71 75 69 72 65
     64 0D
281D 53         06490 NOTDUN$ DB        'Specifications Required',CR
     70 65 63 69 66 69 63 61
     74 69 6F 6E 73 20 52 65
     71 75 69 72 65 64 0D
2835 41         06500 ATBERR$ DB        'Attribute specification error',CR
     74 74 72 69 62 75 74 65
     20 73 70 65 63 69 66 69
     63 61 74 69 6F 6E 20 65
     72 72 6F 72 0D
2853 4E         06510 PROTS$  DB        'NOEXREUPWRRNRMFU'
     4F 45 58 52 45 55 50 57
     52 52 4E 52 4D 46 55
2863 49         06520 NOINDO$ DB        'Invalid command during DO processing',CR
     6E 76 61 6C 69 64 20 63
     6F 6D 6D 61 6E 64 20 64
     75 72 69 6E 67 20 44 4F
     20 70 72 6F 63 65 73 73
     69 6E 67 0D
2888 4E         06530 PACKNM$ DB        'New disk pack name ?   ',3
     65 77 20 64 69 73 6B 20
     70 61 63 6B 20 6E 61 6D
     65 20 3F 20 20 20 03
28A0 4E         06540 NEWMPW$ DB        'New master password ?  ',3
     65 77 20 6D 61 73 74 65
     72 20 70 61 73 73 77 6F
     72 64 20 3F 20 20 03
28B8 4D         06550 CURMPW$ DB        'Master password ?      ',3
     61 73 74 65 72 20 70 61
     73 73 77 6F 72 64 20 3F
     20 20 20 20 20 20 03
28D0 49         06560 BADMPW$ DB        'Invalid master password',CR
     6E 76 61 6C 69 64 20 6D
     61 73 74 65 72 20 70 61
     73 73 77 6F 72 64 0D
28E8 49         06570 BADNAM$ DB        'Invalid disk name',CR
```

```
     6E 76 61 6C 69 64 2Ø 64
     69 73 6B 2Ø 6E 61 6D 65
     ØD
28FA 43       Ø658Ø ABORT$  DB      'Command aborted',CR
     6F 6D 6D 61 6E 64 2Ø 61
     62 6F 72 74 65 64 ØD
               Ø659Ø ;
290A 8Ø       Ø66ØØ PRMTBL$ DB      8ØH
               Ø661Ø ;
ØØ8Ø          Ø662Ø VAL     EQU     8ØH
ØØ4Ø          Ø663Ø SW      EQU     4ØH
ØØ2Ø          Ø664Ø STR     EQU     2ØH
ØØ1Ø          Ø665Ø SGL     EQU     1ØH
               Ø666Ø ;
290B 62       Ø667Ø         DB      SW!STR!2,'PW',Ø
     5Ø 57 ØØ
290F 6626     Ø668Ø         DW      PPARM+1
2911 54       Ø669Ø         DB      SW!SGL!4,'LOCK',Ø
     4C 4F 43 4B ØØ
2917 8926     Ø67ØØ         DW      LPARM+1
2919 56       Ø671Ø         DB      SW!SGL!6,'UNLOCK',Ø
     55 4E 4C 4F 43 4B ØØ
2921 9326     Ø672Ø         DW      UPARM+1
2923 74       Ø673Ø         DB      SW!STR!SGL!4,'NAME',Ø
     4E 41 4D 45 ØØ
2929 2426     Ø674Ø         DW      NPARM+1
292B 73       Ø675Ø         DB      SW!STR!SGL!3,'MPW',Ø
     4D 5Ø 57 ØØ
293Ø 2427     Ø676Ø         DW      MPARM+1
2932 ØØ       Ø677Ø         NOP
               Ø678Ø ;
ØØØ8          Ø679Ø PSWDBUF DS      8               ;Password buffer
ØØØ4          Ø68ØØ HASHBUF DS      4               ;Owner & user hashes
ØØ2Ø          Ø681Ø FCB     DS      32
               Ø682Ø ;
2AØØ          Ø683Ø         ORG     $<-8+1<+8
Ø1ØØ          Ø684Ø BUFFER  DS      256
Ø1ØØ          Ø685Ø GATBUF  DS      256
Ø1ØØ          Ø686Ø HITBUF  DS      256
               Ø687Ø ;
24ØØ          Ø688Ø         END     ATTRIB
```

@@1	ØØØØ	@@2	ØØØØ	@@3	ØØØØ	@@4	ØØØØ
@MOD2	ØØØØ	@MOD4	FFFF	ABORT	241B	ABORT$	28FA
ATBERR	2427	ATBERR$	2835	ATTØ	2468	ATTØ1	247B
ATTØ2	247E	ATT1	247F	ATT2	248B	ATT3	248D
ATTRIB	24ØØ	ATTRIB1	2437	BADMPW$	28DØ	BADNAM$	28E8
BLNKMPW	4296	BUFFER	2AØØ	CKINDO	27F8	CKMPW	279Ø
CKMPW1	2797	CKMPW2	27A5	CKMPW3	27B7	CKMPW4	27B9
CKMPW5	27BC	CKMPW7	27CØ	CR	ØØØD	CURMPW$	28B8
DOHASH	2747	DOINV	25Ø1	DOOWN	253A	DOPRØ1	24B4
DOPRØ2	24C1	DOPRØ3	24DD	DOPRØ4	24E3	DOPRØ5	24E9
DOUSE	2525	DOVIS	2513	EXTERR	242A	FCB	293F
GATBUF	2BØØ	GATRD	27CC	GATRW1	27E9	GATRW2	27F1
GATRW3	27F5	GATWR	27CD	GETMPW	273A	GMPW1	274A
GMPW2	2757	GMPW3	276B	GMPW4	278B	HASHBUF	293B
HITBUF	2CØØ	ILLDRVN	25ED	INVNAM	27C4	IOERR	24ØC
IVCOD	25BØ	LPARM	2688	MPARM	2723	NEWMPW$	28AØ
NOINDO$	2863	NOTDUN	2423	NOTDUN$	281D	NPARM	2623
PACKNM$	2888	PASSWORD	42EØ	PPARM	2665	PRMCOD	25AØ
PRMTBL$	29ØA	PROT	25DB	PROTØ1	25FB	PROTØ2	2678
PROTØ3	2699	PROTØ4	26B2	PROTØ5	26BD	PROTØ6	26CE
PROTØ7	26DC	PROTØ7B	26F3	PROTØ8	26F4	PROTØ9	26F8
PROT1Ø	27ØC	PROTLVL	25A8	PROTS$	2853	PRSFØ1	2588
PRSFLD	2569	PSWDBUF	2933	REVMPW	26E9	SAVESP	243Ø
SGL	ØØ1Ø	SPCREQ	241F	SPCREQ$	28ØA	STR	ØØ2Ø
SW	ØØ4Ø	THISPW	26EØ	TSTMPW	2713	UPARM	2692
UPDDIR	2595	UPDIR1	25AA	UPDIR2	25B2	UPDIR3	25C5
UPDIR4	25D4	VAL	ØØ8Ø	XSPEC1Ø	255C	XSPEC11	2562
XSPEC8A	254F	XSPEC9	2553	@@ABORT	ACED	@@ADTSK	AD8Ø
@@BANK	B298	@@BKSP	AF78	@@BREAK	B2AE	@@CHNIO	ACD8
@@CKBRKC	B2FC	@@CKDRV	ADD4	@@CKEOF	AF8D	@@CKTSK	AD6B
@@CLOSE	AF63	@@CLS	B2E6	@@CMNDI	AD17	@@CMNDR	AD2C
@@CTL	AB3C	@@DATE	ACAE	@@DCSTAT	AE13	@@DEBUG	AD56
@@DECHEX	B218	@@DIRRD	B185	@@DIRWR	B19A	@@DIV16	B2Ø3
@@DIV8	B1EE	@@DODIR	ADE9	@@DSP	ABØØ	@@DSPLY	ABAØ
@@ERROR	AD41	@@EXIT	AD02	@@FEXT	BØF2	@@FLAGS	B282
@@FNAME	B1Ø7	@@FSPEC	BØDD	@@GATRD	B17Ø	@@GATWR	B1AF
@@GET	AB14	@@GTDCB	B131	@@GTDCT	B11C	@@GTMOD	B146
@@HDFMT	AEBB	@@HEX16	B257	@@HEX8	B242	@@HEXDEC	B22D
@@HIGH$	B26C	@@INIT	AF39	@@KBD	AB78	@@KEY	AAEC
@@KEYIN	AB8C	@@KLTSK	ADBF	@@LOAD	BØB3	@@LOC	AFA2
@@LOF	AFB7	@@LOGER	ABD7	@@LOGOT	ABEC	@@MSG	AC23
@@MUL16	B1D9	@@MUL8	B1C4	@@OPEN	AF4E	@@PARAM	AC99
@@PAUSE	AC84	@@PEOF	AFCC	@@POSN	AFE1	@@PRINT	AC38
@@PRT	AB5Ø	@@PUT	AB28	@@RAMDIR	ADFE	@@RDSEC	AE91
@@RDSSC	B15B	@@READ	AFF6	@@REMOV	AF24	@@RENAM	AFØF

```
@@REW           BØØB @@RMTSK         AD95 @@RPTSK         ADAA
@@RREAD         BØ2Ø @@RSLCT         AE7C @@RSTOR         AE3D
@@RUN           BØC8 @@RWRIT         BØ35 @@SEEK          AE67
@@SEEKSC        BØ4A @@SKIP          BØ5F @@SLCT          AE28
@@STEPI         AE52 @@TIME          ACC3 @@VDCTL         AC6F
@@VER           BØ74 @@VRSEC         AEA6 @@WEOF          BØ89
@@WHERE         AB64 @@WRITE         BØ9E @@WRSEC         AEDØ
@@WRSSC         AEE5 @@WRTRK         AEFA
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

Command: AUTO

Library: SYS7/SYS

ISAM # : 11H

```
                00100 ;LBAUTO/ASM - AUTO Command
0000            00110         TITLE   <AUTO - LS-DOS 6.2>
                00120 ;
                00130 ;
0000            00140 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
0000            00150 *GET    VALUES:3                ;Misc. equates
                03920 ;VALUES/ASM - Version 6
                03930 *LIST OFF
                04200 *LIST ON
                00160 ;
0002            00170 SECTOR  EQU     0<8+2
0020            00180 OFFSET  EQU     20H             ;Offset to Auto Buffer
0028            00190 RST28   EQU     28H             ;SVC restart
0019            00200 @CMNDR  EQU     25              ;Execute & return
0031            00210 @RDSEC  EQU     49              ;Read a sector SVC #
0035            00220 @WRSEC  EQU     53              ;Write a sector SVC #
                00230 ;
2400            00240         ORG     2400H
                00250 ;
                00260 AUTO
2400 ED730B24   00270         LD      (SAVESP+1),SP   ;Save SP address
2404 CD0E24     00280         CALL    PARSE           ;Do AUTO command
2407 210000     00290         LD      HL,0            ;Set no error
                00300 ;
                00310 ;       Reinstall SP & RETurn
                00320 ;
240A 310000     00330 SAVESP  LD      SP,$-$          ;P/u old SP
240D C9         00340         RET
                00350 ;
                00360 ;       Skip any leading spaces
                00370 ;
240E 2B         00380 PARSE   DEC     HL              ;Back one
240F 23         00390 PARSE2  INC     HL              ;Bump buff ptr
2410 7E         00400         LD      A,(HL)          ;P/u character
2411 FE20       00410         CP      ' '             ;Space ?
2413 28FA       00420         JR      Z,PARSE2        ;Go til non-space
                00430 ;
                00440 ;       Display auto Buffer on drive :d ?
                00450 ;
2415 FE3F       00460         CP      '?'             ;Display auto buffer ?
2417 2809       00470         JR      Z,DISAUTO
                00480 ;
                00490 ;       Execute Auto Buffer on drive :d ?
                00500 ;
2419 FE3D       00510         CP      '='             ;Execute auto buffer ?
241B 280F       00520         JR      Z,EXAUTO
                00530 ;
                00540 ;       Install the command in the auto buffer
                00550 ;
241D CD5124     00560         CALL    GETDRV          ;P/u drive #
2420 184D       00570         JR      WRBUFF          ;Write new auto & RETurn
                00580 ;
                00590 ;       Display command in auto buffer
                00600 ;
2422 CD4724     00610 DISAUTO CALL    GETAUTO         ;P/u drive #
2425            00620         @@LOGOT                 ;Display it
                00001         IFEQ    00H,1
                00002         LD      HL,
```

```
                00003          ENDIF
2425 3E0C       00004          LD      A,12
2427 EF         00005          RST     40
2428 C2A024     00630          JP      NZ,IOERR        ;NZ - I/O Error
242B C9         00640          RET                     ;Z - done
                00650 ;
                00660 ;        Execute command in auto buffer
                00670 ;
242C CD4724     00680 EXAUTO   CALL    GETAUTO         ;P/u drive #
242F            00690          @@FLAGS
242F 3E65       00006          LD      A,101
2431 EF         00007          RST     40
2432 7E         00700          LD      A,(HL)          ;P/u first character
2433 FE2A       00710          CP      '*'             ;Non-<BREAK>able auto ?
2435 2005       00720          JR      NZ,DOAUTO       ;No - do it
                00730 ;
                00740 ;        Non <BREAK>able auto command - skip "*"
                00750 ;
2437 23         00760          INC     HL              ;Bump to next char
2438 FDCB12E6   00770          SET     4,(IY+SFLAG$)   ;Disable <BREAK>
                00780 ;
                00790 ;        Exit via @CMNDI or @CMNDR if requested
                00800 ;
243C FDCB024E   00810 DOAUTO   BIT     1,(IY+CFLAG$)   ;If CMNDR executing, then
2440 3E19       00820          LD      A,@CMNDR        ;  exit via CMNDR
2442 C22800     00830          JP      NZ,RST28
2445 3D         00840          DEC     A               ;  else readjust to
2446 EF         00850          RST     40              ;  CMNDI & exit
                00860 ;
                00870 ;        GETAUTO - Get Auto Command from BOOT sector
                00880 ;
                00890 ;        Advance to next byte in line
                00900 ;
2447 23         00910 GETAUTO  INC     HL              ;Bump to next char
2448 7E         00920          LD      A,(HL)          ;  & p/u 1st char of drvspc
2449 CD5124     00930          CALL    GETDRV          ;Get drive #
244C CD8424     00940          CALL    RDBOOT          ;Read BOOT sector
244F EB         00950          EX      DE,HL           ;Pt HL to auto buffer
2450 C9         00960          RET                     ;RETurn
                00970 ;
                00980 ;        GETDRV - Check if (HL) contains a legal drive #
                00990 ;
2451 FE0E       01000 GETDRV   CP      CR+1            ;No drivespec ?
2453 D8         01010          RET     C               ;Use drive 0 as default
                01020 ;
                01030 ;        Drivespec indicator entered ?
                01040 ;
2454 FE3A       01050          CP      ':'             ;Drivespec indicator ?
2456 C0         01060          RET     NZ              ;No - use drive 0
                01070 ;
                01080 ;        Drivespec ":" entered - p/u drive #
                01090 ;
2457 23         01100          INC     HL              ;Position to drive #
2458 46         01110          LD      B,(HL)          ;P/u drive #
                01120 ;
                01130 ;        Position to command following drivspec
                01140 ;
2459 23         01150          INC     HL              ;Position to command
245A 7E         01160          LD      A,(HL)          ;C/R following spec ?
245B FE0E       01170          CP      CR+1
245D 3801       01180          JR      C,SKIPINC       ;Yes - don't INC
```

```
245F 23        Ø119Ø          INC     HL              ;Position to command
               Ø12ØØ ;
               Ø121Ø ;        Is the drive number in legal range ?
               Ø122Ø ;
246Ø 78        Ø123Ø SKIPINC  LD      A,B             ;P/u Drive #
2461 D63Ø      Ø124Ø          SUB     'Ø'             ;Less than "Ø" ?
2463 329624    Ø125Ø          LD      (DRIVE+1),A     ;Stuff away drive #
2466 38Ø3      Ø126Ø          JR      C,ILDRIVE       ;C - Illegal drive #
2468 FEØ8      Ø127Ø          CP      7+1             ;Greater than 7 ?
246A D8        Ø128Ø          RET     C               ;RETurn if legal
               Ø129Ø ;
               Ø13ØØ ;        Illegal Drive Number - Display & abort
               Ø131Ø ;
246B 3E2Ø      Ø132Ø ILDRIVE  LD      A,32            ;Init "Illegal drive #
246D 1831      Ø133Ø          JR      IOERR           ;Abort
               Ø134Ø ;
               Ø135Ø ;        WRBUFF - Write Auto Command to Boot
               Ø136Ø ;
246F CD8424    Ø137Ø WRBUFF   CALL    RDBOOT          ;Read in BOOT
               Ø138Ø ;
               Ø139Ø ;        Skip Leading spaces
               Ø14ØØ ;
2472 2B        Ø141Ø          DEC     HL              ;Skip leading spaces
2473 23        Ø142Ø SKPSPCS  INC     HL
2474 7E        Ø143Ø          LD      A,(HL)          ;P/u char
2475 FE2Ø      Ø144Ø          CP      ' '             ;Space ?
2477 28FA      Ø145Ø          JR      Z,SKPSPCS       ;Yes - skip it
               Ø146Ø ;
               Ø147Ø ;        Transfer command into buffer
               Ø148Ø ;
2479 7E        Ø149Ø XFLP     LD      A,(HL)          ;P/u byte
247A EDAØ      Ø15ØØ          LDI                     ;(HL) => (DE) INC HL & DE
247C FEØE      Ø151Ø          CP      CR+1            ;Eol ?
247E 3ØF9      Ø152Ø          JR      NC,XFLP
               Ø153Ø ;
               Ø154Ø ;        RDBOOT/WRBOOT - Read/Write BOOT sector
               Ø155Ø ;
               Ø156Ø ;        Set A = @WRSEC Supervisory CALL #
               Ø157Ø ;
248Ø 3E35      Ø158Ø WRBOOT   LD      A,@WRSEC        ;Write Sector SVC #
2482 18ØD      Ø159Ø          JR      BOOTIO          ;Go to I/O routine
               Ø16ØØ ;
               Ø161Ø ;        RDBOOT - Check if the Drive is there
               Ø162Ø ;
2484 3A9624    Ø163Ø RDBOOT   LD      A,(DRIVE+1)     ;P/u drive #
2487 4F        Ø164Ø          LD      C,A             ;Xfer to C for @CKDRV
2488           Ø165Ø          @@CKDRV                 ;Drive alive
2488 3E21      ØØØØ8          LD      A,33
248A EF        ØØØØ9          RST     4Ø
248B 3E2Ø      Ø166Ø          LD      A,32            ;Init "Illegal drive #
248D 2Ø11      Ø167Ø          JR      NZ,IOERR        ;Abort if bad drive
               Ø168Ø ;
               Ø169Ø ;        Read GAT from directory if Model II
               Ø17ØØ ;
               Ø171Ø          IF      @MOD2
               Ø172Ø          PUSH    IY              ;Save
248F           Ø173Ø          @@GTDCT                 ;Locate DCT
               ØØØ1Ø          LD      A,81
               ØØØ11          RST     4Ø
               Ø174Ø          LD      D,(IY+9)        ;Dir cyl
               Ø175Ø          LD      A,(IY+3)        ;Get DCT data
```

```
                    01760          LD      E,(IY+4)        ;Get dct data
                    01770          POP     IY              ;Restore
                    01780          AND     28H             ;8" floppy?
                    01790          CP      20H             ;Yes?
                    01800          JR      NZ,SETSYSI      ;Nope, sysinfo on 0
                    01810          LD      A,E             ;Get +4
                    01820          AND     50H             ;Bit 6/4
                    01830          CP      40H             ;DD not alien?
                    01840          JR      NZ,SETSYSI      ;Go if alien
                    01850 ;
                    01860          PUSH    HL              ;Save
                    01870          LD      HL,BUFF         ;Start buffer
                    01880          LD      E,0             ;GAT table
248F                01890          @@RDSSC                 ;Read directory
                    00012          LD      A,85
                    00013          RST     40
                    01900          LD      DE,(BUFF+0CDH)  ;Get GAT info byte
                    01910          POP     HL              ;Restore buffer
                    01920          JR      NZ,IOERR        ;Go on disk error
                    01930          BIT     7,E             ;System disk?
                    01940 SETSYSI  LD      DE,SECTOR       ;Sysinfo sector
                    01950          JR      NZ,$+3          ;Go if data disk
                    01960          INC     D               ;Sysinfo on cyl 1
                    01970          LD      (SYSINFO),DE    ;Save sysinfo sector
                    01980          ENDIF
248F 3E31           01990          LD      A,@RDSEC        ;A = @RDSEC SVC #
                    02000 ;
                    02010 ;        Pt HL => Buffer, DE = T/S, C = :d, A = SVC #
                    02020 ;
2491 E5             02030 BOOTIO   PUSH    HL              ;Save command ptr
2492 210025         02040          LD      HL,BUFF         ;HL => I/O buffer
2495 0E00           02050 DRIVE    LD      C,$-$           ;P/u drive # in C
                    02060 ;
2497 110200         02070          LD      DE,SECTOR       ;DE = Track 0, Sector 2
2498                02080 SYSINFO  EQU     $-2
                    02090 ;
                    02100 ;        Issue SVC & point DE => Auto comm buffer
                    02110 ;
249A EF             02120          RST     40              ;Read or Write Sector
249B E1             02130          POP     HL              ;Recover command ptr
249C 112025         02140          LD      DE,BUFF+OFFSET  ;DE => Auto command buff
249F C8             02150          RET     Z               ;RETurn if successful
                    02160 ;
                    02170 ;        I/O Error Handler - Clean up stack & Abort
                    02180 ;
24A0 6F             02190 IOERR    LD      L,A             ;Set HL = Error #
24A1 2600           02200          LD      H,0
24A3 F6C0           02210          OR      0C0H            ;Short error message
24A5 4F             02220          LD      C,A             ;Xfer to C for @ERROR
24A6                02230          @@ERROR                 ;Display error
24A6 3E1A           00014          LD      A,26
24A8 EF             00015          RST     40
24A9 C3A024         02240          JP      SAVESP          ;Exit
                    02250 ;
2500                02260 BUFF     EQU     $<-8+1<+8       ;Next page boundary
                    02270 ;
2400                02280          END     AUTO
```

```
@@1             0000 @@2             0000 @@3             0000
@@4             0000 @CMNDR          0019 @MOD2           0000
@MOD4           FFFF @RDSEC          0031 @WRSEC          0035
ABB             001Ø AP              0027 AUTO            24ØØ
BOOTIO          2491 BREAK           ØØ8Ø BS              ØØØ8
BUFF            25ØØ CFLAG$          ØØØ2 CR              ØØØD
DFLAG$          ØØØ3 DISAUTO         2422 DOAUTO          243C
DRIVE           2495 ETX             ØØØ3 EXAUTO          242C
FLAG            ØØ4Ø GETAUTO         2447 GETDRV          2451
ILDRIVE         246B IOERR           24AØ KFLAG$          ØØØA
LF              ØØØA NUM             ØØ8Ø OFFSET          ØØ2Ø
PARSE           24ØE PARSE2          24ØF PAR_ERR         ØØ2C
RDBOOT          2484 RST28           ØØ28 SAVESP          24ØA
SECTOR          ØØØ2 SFLAG$          ØØ12 SKIPINC         246Ø
SKPSPCS         2473 STR             ØØ2Ø SYSINFO         2498
TAB             ØØØ9 VFLAG$          ØØ15 WRBOOT          248Ø
WRBUFF          246F XFLP            2479 @@ABORT         7FD8
@@ADTSK         8Ø6B @@BANK          8583 @@BKSP          8263
@@BREAK         8599 @@CHNIO         7FC3 @@CKBRKC        85E7
@@CKDRV         8ØBF @@CKEOF         8278 @@CKTSK         8Ø56
@@CLOSE         824E @@CLS           85D1 @@CMNDI         8ØØ2
@@CMNDR         8Ø17 @@CTL           7E27 @@DATE          7F99
@@DCSTAT        8ØFE @@DEBUG         8Ø41 @@DECHEX        85Ø3
@@DIRRD         847Ø @@DIRWR         8485 @@DIV16         84EE
@@DIV8          84D9 @@DODIR         8ØD4 @@DSP           7DEB
@@DSPLY         7E8B @@ERROR         8Ø2C @@EXIT          7FED
@@FEXT          83DD @@FLAGS         856D @@FNAME         83F2
@@FSPEC         83C8 @@GATRD         845B @@GATWR         849A
@@GET           7DFF @@GTDCB         841C @@GTDCT         84Ø7
@@GTMOD         8431 @@HDFMT         81A6 @@HEX16         8542
@@HEX8          852D @@HEXDEC        8518 @@HIGH$         8557
@@INIT          8224 @@KBD           7E63 @@KEY           7DD7
@@KEYIN         7E77 @@KLTSK         8ØAA @@LOAD          839E
@@LOC           828D @@LOF           82A2 @@LOGER         7EC2
@@LOGOT         7ED7 @@MSG           7FØE @@MUL16         84C4
@@MUL8          84AF @@OPEN          8239 @@PARAM         7F84
@@PAUSE         7F6F @@PEOF          82B7 @@POSN          82CC
@@PRINT         7F23 @@PRT           7E3B @@PUT           7E13
@@RAMDIR        8ØE9 @@RDSEC         817C @@RDSSC         8446
@@READ          82E1 @@REMOV         82ØF @@RENAM         81FA
@@REW           82F6 @@RMTSK         8Ø8Ø @@RPTSK         8Ø95
@@RREAD         83ØB @@RSLCT         8167 @@RSTOR         8128
@@RUN           83B3 @@RWRIT         832Ø @@SEEK          8152
@@SEEKSC        8335 @@SKIP          834A @@SLCT          8113
@@STEPI         813D @@TIME          7FAE @@VDCTL         7F5A
@@VER           835F @@VRSEC         8191 @@WEOF          8374
@@WHERE         7E4F @@WRITE         8389 @@WRSEC         81BB
@@WRSSC         81DØ @@WRTRK         81E5
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

Command: BUILD

Library: SYS7/SYS

ISAM # : 33H

```
                00100 ;LBBUILD/ASM - BUILD Command
0000            00110         TITLE   <BUILD - LS-DOS 6.2>
                00120 ;
                00130 ;
0050            00140 CPL     EQU     80              ;Characters per line
                00150 ;
0000            00160 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
0000            00170 *GET    VALUES:3                ;Misc. equates
                03920 ;VALUES/ASM - Version 6
                03930 *LIST OFF
                04200 *LIST ON
                00180 ;
2400            00190         ORG     2400H
                00200 ;
                00210 ;       Was the <BREAK> key hit ?
                00220 ;
                00230 START
2400            00240         @@CKBRKC                ;Break key down?
2400 3E6A       00001         LD      A,106
2402 EF         00002         RST     40
2403 2804       00250         JR      Z,$+6           ;Go if not
2405 21FFFF     00260         LD      HL,-1           ;  else abort
2408 C9         00270         RET
                00280 ;
                00290 ;       <BREAK> not hit - execute module
                00300 ;
2409 ED731424   00310         LD      (SAVESP+1),SP   ;Save SP address
240D CD3C24     00320         CALL    BUILD           ;Build a file
2410 210000     00330         LD      HL,0            ;Set no error
2413 310000     00340 SAVESP  LD      SP,$-$          ;P/u original SP addr.
2416            00350         @@CKBRKC
2416 3E6A       00003         LD      A,106
2418 EF         00004         RST     40
2419 C9         00360         RET                     ;Exit with retcode
                00370 ;
                00380 ;       I/O Error Handler
                00390 ;
241A 2600       00400 IOERR   LD      H,0             ;Set HL = Error #
241C 6F         00410         LD      L,A             ;
241D F6C0       00420         OR      0C0H            ;Short error mess & RET
241F 4F         00430         LD      C,A             ;Stuff in C
2420            00440         @@ERROR                 ;Display error
2420 3E1A       00005         LD      A,26
2422 EF         00006         RST     40
2423 18EE       00450         JR      SAVESP          ;ABORT
                00460 ;
                00470 ;       Internal Error Message Handling
                00480 ;
2425 219825     00490 BADIGS  LD      HL,BADIGS$      ;"Bad Hex digit"
2428 DD         00500         DB      0DDH
2429 218425     00510 ODDIGS  LD      HL,ODDIGS$      ;"Odd # of hex digits"
242C DD         00520         DB      0DDH
242D 21B225     00530 SPCREQ  LD      HL,SPCREQ$      ;"Filespec Required"
2430 DD         00540         DB      0DDH
2431 21C525     00550 EXISTS  LD      HL,EXISTS$      ;"File already exists"
2434            00560         @@LOGOT                 ;Log error message
                00007         IFEQ    00H,1
                00008         LD      HL,
```

```
                00009          ENDIF
2434 3E0C       00010          LD      A,12
2436 EF         00011          RST     40
2437 21FFFF     00570 ABORT    LD      HL,-1           ;Set abort code
243A 18D7       00580          JR      SAVESP          ;Exit
                00590 ;
                00600 ;        BUILD - Build A file
                00610 ;
243C 112126     00620 BUILD    LD      DE,FCB1         ;DE => FCB
243F            00630          @@FSPEC                 ;Legal Filespec ?
243F 3E4E       00012          LD      A,78
2441 EF         00013          RST     40
2442 20E9       00640          JR      NZ,SPCREQ       ;No - "Filespec req"
                00650 ;
                00660 ;        Legal Filespec - Stuff default /ext of JCL
                00670 ;
2444 E5         00680          PUSH    HL              ;Save cmd buf ptr
2445 210626     00690          LD      HL,JCLEXT       ;HL => "JCL"
2448            00700          @@FEXT                  ;Fetch extension
2448 3E4F       00014          LD      A,79
244A EF         00015          RST     40
244B E1         00710          POP     HL              ;Rcvr command ptr
                00720 ;
                00730 ;        Pick up parameters if any
                00740 ;
244C 110E26     00750          LD      DE,PRMTBL$      ;DE => Parameter table
244F            00760          @@PARAM                 ;Get any parameters
244F 3E11       00016          LD      A,17
2451 EF         00017          RST     40
2452 20C6       00770          JR      NZ,IOERR        ;Quit on parm error
                00780 ;
                00790 ;        Position to Extension
                00800 ;
2454 212126     00810          LD      HL,FCB1         ;Point to start of FCB
2457 7E         00820 SLASH?   LD      A,(HL)          ;P/u a char
2458 23         00830          INC     HL              ;  & bump pointer
2459 FE0D       00840          CP      CR              ;End of line?
245B 281B       00850          JR      Z,NOTKSM        ;Yes - not a KSM file
245D FE2F       00860          CP      '/'             ;Start of EXT?
245F 20F6       00870          JR      NZ,SLASH?       ;Loop if not
                00880 ;
                00890 ;        Is the extension KSM ?
                00900 ;
2461 E5         00910          PUSH    HL              ;HL => Extension
2462 7E         00920          LD      A,(HL)          ;P/u character
2463 23         00930          INC     HL              ;Bump ptr
2464 FE4B       00940          CP      'K'             ;Match K?
2466 2010       00950          JR      NZ,NOTKSM
2468 7E         00960          LD      A,(HL)
2469 23         00970          INC     HL
246A FE53       00980          CP      'S'             ;Match S?
246C 200A       00990          JR      NZ,NOTKSM
246E 7E         01000          LD      A,(HL)
246F FE4D       01010          CP      'M'             ;Match M?
2471 2005       01020          JR      NZ,NOTKSM
                01030 ;
                01040 ;        Extension is /KSM - stuff 0FFH in indicator
                01050 ;
2473 3EFF       01060          LD      A,0FFH
2475 32DE24     01070          LD      (KSM?+1),A      ;Stuff KSM indicator
                01080 ;
```

```
                01090 ;       Is the extension /JCL ?
                01100 ;
2478 E1         01110 NOTKSM  POP     HL              ;HL => Extension field
2479 7E         01120         LD      A,(HL)          ;Ck if EXT is JCL
247A 23         01130         INC     HL
247B FE4A       01140         CP      'J'             ;Match J?
247D 2010       01150         JR      NZ,INIT
247F 7E         01160         LD      A,(HL)
2480 23         01170         INC     HL
2481 FE43       01180         CP      'C'             ;Match C?
2483 200A       01190         JR      NZ,INIT
2485 7E         01200         LD      A,(HL)
2486 FE4C       01210         CP      'L'             ;Match L?
2488 2005       01220         JR      NZ,INIT
248A 3E4F       01230         LD      A,CPL-1         ;Max 79 cpl on JCL
248C 32F524     01240         LD      (LINLEN+2),A
                01250 ;
                01260 ;       Init the file with LRL of 256
                01270 ;
248F 210027     01280 INIT    LD      HL,IOBUF        ;HL => I/O buffer
2492 112126     01290         LD      DE,FCB1         ;DE => FCB
2495 0600       01300         LD      B,0             ;B = LRL = 256
2497            01310         @@INIT                  ;Init the file
2497 3E3A       00018         LD      A,58
2499 EF         00019         RST     40
249A 2025       01320         JR      NZ,IOERRA       ;Jump on error
                01330 ;
                01340 ;       Stuff Filespec into Buffer
                01350 ;
249C F5         01360         PUSH    AF              ;Save Carry
249D 11D925     01370         LD      DE,FILEBUF      ;DE => Filespec
24A0 ED4B2726   01380         LD      BC,(FCB1+6)     ;B = DEC, C = Drive #
24A4 3A2126     01390         LD      A,(FCB1)        ;P/u to test device/file
24A7 CD6E25     01400         CALL    $FNAME
24AA 2015       01410         JR      NZ,IOERRA
24AC F1         01420         POP     AF              ;F = Status from @INIT
                01430 ;
24AD 21E825     01440         LD      HL,BMESS1       ;Default "Building :"
24B0 3818       01450         JR      C,SETBUF        ;Jump if New file
                01460 ;
                01470 ;       File already exists - Was APPEND specified ?
                01480 ;
24B2 010000     01490 APPEND  LD      BC,$-$          ;P/u APPEND parameter
24B5 112126     01500         LD      DE,FCB1         ;DE => FCB
24B8 0C         01510         INC     C               ;Specified ?
24B9 2809       01520         JR      Z,APP1          ;Go if so
24BB            01530         @@CLOSE                 ;Close to reset open bit
24BB 3E3C       00020         LD      A,60
24BD EF         00021         RST     40
24BE CA3124     01540         JP      Z,EXISTS        ;Quit with "file exists...
24C1 C31A24     01550 IOERRA  JP      IOERR           ;  or if error on close
                01560 ;
                01570 ;       Position to end of file for append
                01580 ;
24C4            01590 APP1    @@PEOF                  ;Position to EOF
24C4 3E41       00022         LD      A,65
24C6 EF         00023         RST     40
24C7 21F325     01600         LD      HL,BMESS2       ;"Appending *KI to "
                01610 ;
                01620 ;       Display Building/Appending Message
                01630 ;
```

```
24CA CD5425   01640 SETBUF  CALL    DSPLY           ;Display message
24CD 21D925   01650         LD      HL,FILEBUF      ;Filename buffer
24D0 CD5425   01660         CALL    DSPLY
24D3 0E0D     01670         LD      C,CR            ;End line
24D5          01680         @@DSP
24D5 3E02     00024         LD      A,2
24D7 EF       00025         RST     40
24D8 20E7     01690         JR      NZ,IOERRA
24DA 210028   01700         LD      HL,BUFFER       ;HL => Input buffer
              01710 ;
              01720 ;       Is this a KSM File ?
              01730 ;
24DD 3E00     01740 KSM?    LD      A,$-$           ;Not zero if KSM
24DF B7       01750         OR      A
24E0 2811     01760         JR      Z,LINLEN        ;Z - not a KSM
              01770 ;
              01780 ;       KSM loop - p/u current letter & increment
              01790 ;
24E2 E5       01800 KSM1    PUSH    HL              ;Save text pointer
24E3 210926   01810         LD      HL,LETBUF       ;"A=> "
24E6 34       01820         INC     (HL)            ;Increment letter
24E7 7E       01830         LD      A,(HL)          ;P/u letter
              01840 ;
              01850 ;       Finished with all the KSM keys ?
              01860 ;
24E8 FE5B     01870         CP      'Z'+1           ;Go past Z?
24EA 2003     01880         JR      NZ,DISSTR       ;No - display string
24EC E1       01890         POP     HL              ;Recover text ptr
24ED 183C     01900         JR      GOTEND          ;Finished
              01910 ;
              01920 ;       Display letter & "=> "
              01930 ;
24EF CD5425   01940 DISSTR  CALL    DSPLY           ;Display string
24F2 E1       01950         POP     HL              ;Recover text ptr
              01960 ;
              01970 ;       Input line with either 255 or 79 characters
              01980 ;
24F3 0100FF   01990 LINLEN  LD      BC,255<8        ;255 or 79 (JCL)
24F6          02000         @@KEYIN                 ;Input line
24F6 3E09     00026         LD      A,9
24F8 EF       00027         RST     40
24F9 382C     02010         JR      C,GOTBRK        ;Exit on <BREAK>
24FB 2025     02020         JR      NZ,TSTEOF       ;Ck if EOF key used
              02030 ;
              02040 ;       Got a line of input, check if HEX parameter
              02050 ;
24FD 110000   02060 HPARM   LD      DE,$-$          ;P/u HEX parameter
2500 1C       02070         INC     E               ;Specified ?
2501 2018     02080         JR      NZ,NOTHEX       ;No - ASCII input
              02090 ;
              02100 ;       HEX parm was entered - convert input to hex
              02110 ;
2503 54       02120         LD      D,H             ;Point DE => Input
2504 5D       02130         LD      E,L
              02140 ;
2505 CD5B25   02150 HP1     CALL    CVRTHEX         ;Convert char @ DE
2508 05       02160         DEC     B               ;Decrement count
2509 CA2924   02170         JP      Z,ODDIGS        ;Done ? - odd # of digits
              02180 ;
              02190 ;       Stuff first digit into high order nibble
              02200 ;
```

```
250C 07       02210          RLCA                    ;Shift to hi-order nybble
250D 07       02220          RLCA
250E 07       02230          RLCA
250F 07       02240          RLCA
2510 4F       02250          LD      C,A             ;Save in C
              02260 ;
              02270 ;        P/u low-order digit & OR with high order
              02280 ;
2511 CD5B25   02290          CALL    CVRTHEX         ;Convert char @ DE
2514 B1       02300          OR      C               ;OR with hi-order nibble
2515 77       02310          LD      (HL),A          ;Stuff in buffer
              02320 ;
              02330 ;        Increment converted input ptr & count down
              02340 ;
2516 23       02350          INC     HL              ;Bump conv input ptr
2517 10EC     02360          DJNZ    HP1             ;B hex digits
2519 18C2     02370          JR      KSM?            ;Done conv, back to loop
              02380 ;
              02390 ;        ASCII input, point HL to next free location
              02400 ;
251B 48       02410 NOTHEX   LD      C,B             ;Advance memory buffer
251C 0600     02420          LD      B,0             ;To end of this line
251E 09       02430          ADD     HL,BC           ;HL => End of line
251F 23       02440          INC     HL              ;Bump to 1st free posn
2520 18BB     02450          JR      KSM?            ;Loop for input
              02460 ;
              02470 ;        EOF or BREAK hit - Test Number of chars entered
              02480 ;        If not at line start, convert to CR and continue
              02490 ;
2522 FE1C     02500 TSTEOF   CP      1CH             ;EOF?
2524 C21A24   02510          JP      NZ,IOERR        ;Real error if not
2527 78       02520 GOTBRK   LD      A,B             ;Any characters ?
2528 B7       02530          OR      A
2529 20D2     02540          JR      NZ,HPARM        ;Yes - continue Build
              02550 ;
              02560 ;        Save input pointer address
              02570 ;
252B 223225   02580 GOTEND   LD      (ENDTXT+1),HL   ;Save buffer posn
252E 210028   02590          LD      HL,BUFFER       ;HL => Start of input
2531 110000   02600 ENDTXT   LD      DE,$-$          ;DE => Last used address
              02610 ;
              02620 ;        Is there any more text to write out ?
              02630 ;
2534 EB       02640          EX      DE,HL           ;Swap for math
2535 AF       02650          XOR     A               ;Clear carry
2536 ED52     02660          SBC     HL,DE           ;Any more text to write ?
2538 280B     02670          JR      Z,ATEND         ;No - don't write any
              02680 ;
              02690 ;        Write a byte to the file
              02700 ;
253A EB       02710          EX      DE,HL           ;HL => Byte to output
253B 112126   02720          LD      DE,FCB1         ;DE => FCB
253E 4E       02730          LD      C,(HL)          ;P/u a char
253F 23       02740          INC     HL              ;Bump ptr
2540          02750          @@PUT                   ;Output char
2540 3E04     00028          LD      A,4
2542 EF       00029          RST     40
2543 28EC     02760          JR      Z,ENDTXT        ;Loop for more if i/o OK
              02770 ;
              02780 ;        Done writing - either end of text or error
              02790 ;
```

```
2545 F5       02800 ATEND   PUSH    AF              ;Save error code if any
              02810 ;
              02820 ;       Close the file
              02830 ;
2546 112126   02840         LD      DE,FCB1         ;DE => FCB
2549          02850         @@CLOSE                 ;Close the file
2549 3E3C     00030         LD      A,60
254B EF       00031         RST     40
254C C21A24   02860         JP      NZ,IOERR        ;Can't - I/O error
              02870 ;
              02880 ;       Make sure @PUT didn't RETurn any I/O error
              02890 ;
254F F1       02900         POP     AF              ;Recover error code
2550 C8       02910         RET     Z               ;Ok - Return
2551 C31A24   02920         JP      IOERR           ;NZ - I/O Error
              02930 ;
              02940 ;       DSPLY - Display Line & HL
              02950 ;
              02960 ;
2554          02970 DSPLY   @@DSPLY                 ;Display line
              00032         IFEQ    00H,1
              00033         LD      HL,
              00034         ENDIF
2554 3E0A     00035         LD      A,10
2556 EF       00036         RST     40
2557 C8       02980         RET     Z               ;Back if good
2558 C31A24   02990         JP      IOERR           ;  else abort
              03000 ;
              03010 ;       CVRTHEX - Convert character at DE to hex
              03020 ;
255B 1A       03030 CVRTHEX LD      A,(DE)          ;Get a char
255C 13       03040         INC     DE              ;Bump ptr
255D D630     03050         SUB     '0'             ;Convert to binary
255F 380A     03060         JR      C,BADIGSA       ;Can't be < '0'
2561 FE0A     03070         CP      10              ;Numeric character ?
2563 D8       03080         RET     C               ;Yes - return
2564 CBAF     03090         RES     5,A             ;No - convert to U/C
2566 D607     03100         SUB     7               ;Adjust A-F to be 10-15
2568 FE10     03110         CP      16              ;Legal hex digit ?
256A D8       03120         RET     C               ;Yes - return
256B C32524   03130 BADIGSA JP      BADIGS          ;No - Bad Hex Digit
              03140 ;
              03150 ;       Routine to pick up device/file name
              03160 ;
256E CB7F     03170 $FNAME  BIT     7,A             ;Test device/file
2570 2804     03180         JR      Z,FNAME1        ;Go if device
2572          03190         @@FNAME
2572 3E50     00037         LD      A,80
2574 EF       00038         RST     40
2575 C9       03200         RET
2576 3E2A     03210 FNAME1  LD      A,'*'           ;Stuff device indicator
2578 12       03220         LD      (DE),A
2579 13       03230         INC     DE
257A 79       03240         LD      A,C             ;Stuff 1st character
257B 12       03250         LD      (DE),A
257C 13       03260         INC     DE
257D 78       03270         LD      A,B             ;Stuff 2nd character
257E 12       03280         LD      (DE),A
257F 13       03290         INC     DE
2580 3E03     03300         LD      A,3             ;Stuff ETX
2582 12       03310         LD      (DE),A
```

```
2583 C9          03320          RET
                 03330 ;
                 03340 ;        ERROR Messages
                 03350 ;
2584 4F          03360 ODDIGS$ DB       'Odd # of hex digits',CR
     64 64 20 23 20 6F 66 20
     68 65 78 20 64 69 67 69
     74 73 0D
2598 42          03370 BADIGS$ DB       'Bad hex digit encountered',CR
     61 64 20 68 65 78 20 64
     69 67 69 74 20 65 6E 63
     6F 75 6E 74 65 72 65 64
     0D
25B2 46          03380 SPCREQ$ DB       'File spec required',CR
     69 6C 65 20 73 70 65 63
     20 72 65 71 75 69 72 65
     64 0D
25C5 46          03390 EXISTS$ DB       'File already exists',CR
     69 6C 65 20 61 6C 72 65
     61 64 79 20 65 78 69 73
     74 73 0D
                 03400 ;
000F             03410 FILEBUF DS       15
25E8 42          03420 BMESS1  DB       'Building: ',ETX
     75 69 6C 64 69 6E 67 3A
     20 03
25F3 41          03430 BMESS2  DB       'Appending: *K1 to ',ETX
     70 70 65 6E 64 69 6E 67
     3A 20 2A 4B 49 20 74 6F
     20 03
                 03440 ;
2606 4A          03450 JCLEXT  DB       'JCL'
     43 4C
                 03460 ;
2609 40          03470 LETBUF  DB       'A'-1,'=> ',ETX
     3D 3E 20 03
                 03480 ;
                 03490 ;        PARAMETER TABLE
                 03500 ;
260E 80          03510 PRMTBL$ DB       80H             ;6.x Parameter table
                 03520 ;
                 03530 ;        HEX (H) parameter - Flag input only
                 03540 ;
260F 53          03550          DB      FLAG!ABB!3
2610 48          03560          DB      'HEX'
     45 58
2613 00          03570          DB      0
2614 FE24        03580          DW      HPARM+1
                 03590 ;
                 03600 ;        APPEND (A) parameter - Flag input only
                 03610 ;
2616 56          03620          DB      FLAG!ABB!6
2617 41          03630          DB      'APPEND'
     50 50 45 4E 44
261D 00          03640          DB      0
261E B324        03650          DW      APPEND+1
                 03660 ;
2620 00          03670          DB      0
                 03680 ;
                 03690 ;        Buffer Area
                 03700 ;
```

```
2621 00         03710 FCB1      DB      0
001F            03720           DS      31
2700            03730           ORG     $<-8+1<8
0100            03740 IOBUF     DS      256
0100            03750 BUFFER    DS      256
                03760 ;
2400            03770           END     START
```

```
$FNAME          256E @@1        0000 @@2        0000
@@3             0000 @@4        0000 @MOD2      0000
@MOD4           FFFF ABB        0010 ABORT      2437
AP              0027 APP1       24C4 APPEND     24B2
ATEND           2545 BADIGS     2425 BADIGS$    2598
BADIGSA         256B BMESS1     25E8 BMESS2     25F3
BREAK           0080 BS         0008 BUFFER     2800
BUILD           243C CFLAG$     0002 CPL        0050
CR              000D CVRTHEX    255B DFLAG$     0003
DISSTR          24EF DSPLY      2554 ENDTXT     2531
ETX             0003 EXISTS     2431 EXISTS$    25C5
FCB1            2621 FILEBUF    25D9 FLAG       0040
FNAME1          2576 GOTBRK     2527 GOTEND     252B
HP1             2505 HPARM      24FD INIT       248F
IOBUF           2700 IOERR      241A IOERRA     24C1
JCLEXT          2606 KFLAG$     000A KSM1       24E2
KSM?            24DD LETBUF     2609 LF         000A
LINLEN          24F3 NOTHEX     251B NOTKSM     2478
NUM             0080 ODDIGS     2429 ODDIGS$    2584
PAR_ERR         002C PRMTBL$    260E SAVESP     2413
SETBUF          24CA SFLAG$     0012 SLASH?     2457
SPCREQ          242D SPCREQ$    25B2 START      2400
STR             0020 TAB        0009 TSTEOF     2522
VFLAG$          0015 @@ABORT    8B5C @@ADTSK    8BEF
@@BANK          9107 @@BKSP     8DE7 @@BREAK    911D
@@CHNIO         8B47 @@CKBRKC   916B @@CKDRV    8C43
@@CKEOF         8DFC @@CKTSK    8BDA @@CLOSE    8DD2
@@CLS           9155 @@CMNDI    8B86 @@CMNDR    8B9B
@@CTL           89AB @@DATE     8B1D @@DCSTAT   8C82
@@DEBUG         8BC5 @@DECHEX   9087 @@DIRRD    8FF4
@@DIRWR         9009 @@DIV16    9072 @@DIV8     905D
@@DODIR         8C58 @@DSP      896F @@DSPLY    8A0F
@@ERROR         8BB0 @@EXIT     8B71 @@FEXT     8F61
@@FLAGS         90F1 @@FNAME    8F76 @@FSPEC    8F4C
@@GATRD         8FDF @@GATWR    901E @@GET      8983
@@GTDCB         8FA0 @@GTDCT    8F8B @@GTMOD    8FB5
@@HDFMT         8D2A @@HEX16    90C6 @@HEX8     90B1
@@HEXDEC        909C @@HIGH$    90DB @@INIT     8DA8
@@KBD           89E7 @@KEY      895B @@KEYIN    89FB
@@KLTSK         8C2E @@LOAD     8F22 @@LOC      8E11
@@LOF           8E26 @@LOGER    8A46 @@LOGOT    8A5B
@@MSG           8A92 @@MUL16    9048 @@MUL8     9033
@@OPEN          8DBD @@PARAM    8B08 @@PAUSE    8AF3
@@PEOF          8E3B @@POSN     8E50 @@PRINT    8AA7
@@PRT           89BF @@PUT      8997 @@RAMDIR   8C6D
@@RDSEC         8D00 @@RDSSC    8FCA @@READ     8E65
@@REMOV         8D93 @@RENAM    8D7E @@REW      8E7A
@@RMTSK         8C04 @@RPTSK    8C19 @@RREAD    8E8F
@@RSLCT         8CEB @@RSTOR    8CAC @@RUN      8F37
@@RWRIT         8EA4 @@SEEK     8CD6 @@SEEKSC   8EB9
@@SKIP          8ECE @@SLCT     8C97 @@STEPI    8CC1
@@TIME          8B32 @@VDCTL    8ADE @@VER      8EE3
@@VRSEC         8D15 @@WEOF     8EF8 @@WHERE    89D3
@@WRITE         8F0D @@WRSEC    8D3F @@WRSSC    8D54
@@WRTRK         8D69
2400 is the transfer address
00000 Total errors
```

NOTES:

NOTES:

Command: COPY, APPEND

Library: SYS6/SYS

ISAM # : 32H, 31H

```
                00100 ;LBCOPY/ASM - Copy/Append Commands
0000            00110         TITLE   <COPY/APPEND - LS-DOS 6.2>
                00120 ;
0000            00130 *GET    SVCMAC:3
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
0000            00140 *GET    VALUES:3
                03920 ;VALUES/ASM - Version 6
                03930 *LIST OFF
                04200 *LIST ON
                00150 ;
001E            00160 EL      EQU     30              ;Erase Line
001D            00170 BL      EQU     29              ;Beginning of Line
003A            00180 @INIT   EQU     58              ;@INIT SVC #
003B            00190 @OPEN   EQU     59              ;@OPEN SVC #
0080            00200 BREAK   EQU     80H             ;BREAK character
                00210 ;
2400            00220         ORG     2400H
                00230 ;
2400            00240 *GET    LBCOPYA:3
                04210 ;LBCOPYA/ASM - Copy/Append commands
2400            04220         SUBTTL  '<LBCOPYA - APPEND Mainline>'
```

LBCOPYA - APPEND Mainline

```
2400            04230           PAGE
                04240 ;
                04250 ;         Jump to COPY Entry Point
                04260 ;
2400 C3F524     04270 COPY      JP      COPYST          ;Go to COPY
                04280 ;
                04290 ;         APPEND Entry Point - Was the <BREAK> hit ?
                04300 ;
                04310 APPEND
2403            04320           @@CKBRKC                ;Check for break
2403 3E6A       00001           LD      A,106
2405 EF         00002           RST     40
2406 2804       04330           JR      Z,APPENDA       ;Continue if not
2408 21FFFF     04340           LD      HL,-1           ;  else abort
240B C9         04350           RET
                04360 ;
                04370 ;         <BREAK> not hit - Execute Module
                04380 ;
                04390 APPENDA
240C ED734924   04400           LD      (SAVESP+1),SP   ;Save SP address
2410 CD4F24     04410           CALL    APCODE          ;Call Append code
2413 210000     04420 EXIT      LD      HL,0            ;Good exit
2416 1830       04430           JR      SAVESP
                04440 ;
                04450 ;         I/O Error Display & Abort Routine
                04460 ;
2418 F5         04470 IOERR     PUSH    AF              ;Save error code
2419 CD1629     04480           CALL    PMTSYS          ;Prompt SYSTEM Disk
241C F1         04490           POP     AF              ;Rcvr error code
241D F6C0       04500           OR      0C0H            ;Set for abbrev error msg
241F 4F         04510           LD      C,A             ;Save Error # in C
2420            04520           @@ERROR                 ;Display & abort
2420 3E1A       00003           LD      A,26
2422 EF         00004           RST     40
2423 1823       04530           JR      SAVESP          ;Good bye
                04540 ;
                04550 ;         Load HL with error message string to display
                04560 ;
2425 21C92A     04570 SAMERR    LD      HL,SAMERR$      ;"Source & Dest same"
2428 DD         04580           DB      0DDH
2429 218F2A     04590 SPCREQ    LD      HL,SPCREQ$      ;"File spec required"
242C DD         04600           DB      0DDH
242D 21A22A     04610 NOINDO    LD      HL,NOINDO$      ;"Invalid during <DO>"
2430 DD         04620           DB      0DDH
2431 21492A     04630 DIFLRL    LD      HL,DIFLRL$      ;"Files have diff LRLs"
2434 DD         04640           DB      0DDH
2435 21632A     04650 DSTREQ    LD      HL,DSTREQ$      ;"Dest spec Required"
2438            04660           @@LOGOT                 ;Log error message
                00005           IFEQ    00H,1
                00006           LD      HL,
                00007           ENDIF
2438 3E0C       00008           LD      A,12
243A EF         00009           RST     40
                04670 ;
                04680 ;         Attempt to close any OPEN destination file
                04690 ;
243B 11B32B     04700           LD      DE,FCB2         ;Point to dest FCB
243E 1A         04710           LD      A,(DE)          ;Is the file OPEN?
243F 07         04720           RLCA
2440 3003       04730           JR      NC,ABORT        ;No - abort
```

LBCOPYA - APPEND Mainline

```
2442            04740           @@CLOSE
2442 3E3C       00010           LD      A,60
2444 EF         00011           RST     40
2445 21FFFF     04750 ABORT     LD      HL,-1           ;Abort code to HL
                04760 ;
                04770 ;         P/u Stack, clear any <BREAK> & return
                04780 ;
2448 310000     04790 SAVESP    LD      SP,$-$          ;P/u stack
244B            04800           @@CKBRKC                ;Clear any <BREAK>
244B 3E6A       00012           LD      A,106
244D EF         00013           RST     40
244E C9         04810           RET                     ;  and RETurn
                04820 ;
                04830 ;         APCODE - Append spec to spec
                04840 ;
                04850 APCODE
244F AF         04860           XOR     A               ;Turn off CLONE parm
2450 328026     04870           LD      (CPARM+1),A
2453 328126     04880           LD      (CPARM+2),A
2456 327828     04890           LD      (APPFLAG+1),A   ;We're in APPEND not COPY
                04900 ;
2459 CDC228     04910           CALL    DOINIT          ;Set High memory
                04920 ;
                04930 ;         Check if Filespec/Devspec #1 is legal
                04940 ;
245C 11072C     04950           LD      DE,FCB1         ;DE => File #1 FCB
245F            04960           @@FSPEC                 ;Check out filespec
245F 3E4E       00014           LD      A,78
2461 EF         00015           RST     40
2462 C22924     04970           JP      NZ,SPCREQ       ;NZ - Filespec Required
                04980 ;
                04990 ;         Check if Filespec/Devspec #2 is legal
                05000 ;
2465 11B32B     05010           LD      DE,FCB2         ;DE => File #2 FCB
2468            05020           @@FSPEC                 ;Check if legal
2468 3E4E       00016           LD      A,78
246A EF         00017           RST     40
246B C43027     05030           CALL    NZ,CVRTUC       ;Convert line to U/C
                05040 ;
                05050 ;         Is the second FCB a device ?
                05060 ;
246E 3AB32B     05070 APND1     LD      A,(FCB2)        ;P/u byte 0 of FCB2
2471 FE2A       05080           CP      '*'             ;Is this a devspec ?
2473 CA2924     05090           JP      Z,SPCREQ        ;Z - Filespec required
                05100 ;
                05110 ;         Parse any parameters entered
                05120 ;
2476 11812B     05130           LD      DE,APPTBL       ;DE => Parameter Table
2479            05140           @@PARAM                 ;Check out parameters
2479 3E11       00018           LD      A,17
247B EF         00019           RST     40
247C C21824     05150           JP      NZ,IOERR        ;NZ - Parameter Error
                05160 ;
247F CD5A27     05170           CALL    PRSPC           ;P/u FCB ptr in DE
                05180 ;
                05190 ;         Open Filespec #2 with LRL of 256
                05200 ;
2482 CD252A     05210           CALL    OPENSR2         ;Open Filespec #2
2485 CD6128     05220           CALL    PUTDEST         ;Xfer Dest filespec
```

LBCOPYA - APPEND Mainline

```
2488 CD9628   Ø5230          CALL    GETLRL          ;Get LRL from DIR entry
              Ø5240 ;
248B 32A424   Ø5250          LD      (LRL2+1),A      ;Set dir LRL into parm
248E 327D26   Ø5260          LD      (GEOF1+1),A     ;Also stuff for later
              Ø5270 ;
              Ø5280 ;        Open Filespec #1 with LRL of 256
              Ø5290 ;
2491 CD1B2A   Ø53ØØ          CALL    OPENSRC         ;Open Filespec #1
2494 CD5B28   Ø5310          CALL    PUTSOUR         ;Xfer source filespec
              Ø5320 ;
              Ø5330 ;        Is the Source a Device ?
              Ø5340 ;
2497 EB       Ø5350          EX      DE,HL
2498 CB7E     Ø5360          BIT     7,(HL)          ;P/u FCB+Ø of source
249A EB       Ø5370          EX      DE,HL           ;Device ?
249B CC7228   Ø5380          CALL    Z,CPYFILE       ;Display "Appending ..."
249E 2819     Ø5390          JR      Z,APND2         ;Yes - don't check LRLs
              Ø54ØØ ;
              Ø5410 ;        File Source - Check if LRLs are different
              Ø5420 ;
24AØ CD9628   Ø5430          CALL    GETLRL          ;P/u LRL of Filespec #1
24A3 Ø6ØØ     Ø5440 LRL2     LD      B,$-$           ;P/u LRL of Filespec #2
24A5 A8       Ø5450          XOR     B               ;Same ?
24A6 C23124   Ø5460          JP      NZ,DIFLRL       ;No - Different LRLs
24A9 CD7228   Ø5470          CALL    CPYFILE         ;"Appending : "
              Ø5480 ;
              Ø5490 ;        Files have same LRLs, check STRIP parameter
              Ø55ØØ ;
24AC 11ØØØØ   Ø5510 SPARM    LD      DE,$-$          ;P/u strip parameter
24AF 7A       Ø5520          LD      A,D             ;If STRIP, then must do
24BØ B3       Ø5530          OR      E               ;  byte I/O
24B1 2ØØ6     Ø5540          JR      NZ,APND2        ;Go if STRIP
              Ø5550 ;
              Ø5560 ;        Pick up End of File offset byte from FCB
              Ø5570 ;
24B3 3ABB2B   Ø5580          LD      A,(FCB2+8)      ;Get eof mark
24B6 B7       Ø5590          OR      A               ;If full sectors, use
24B7 281F     Ø56ØØ          JR      Z,APND3         ;Sector I/O
              Ø5610 ;
              Ø5620 ;        EOF not on page boundary - use byte I/O
              Ø5630 ;
24B9 11B32B   Ø5640 APND2    LD      DE,FCB2
24BC          Ø5650          @@PEOF                  ;Position to end of file
24BC 3E41     ØØØ2Ø          LD      A,65
24BE EF       ØØØ21          RST     4Ø
              Ø5660 ;
              Ø5670 ;        If STRIP, then backspace the dest by 1 byte
              Ø5680 ;
24BF 3AAD24   Ø5690          LD      A,(SPARM+1)     ;P/u SPARM
24C2 B7       Ø57ØØ          OR      A               ;Specified ?
24C3 28ØA     Ø5710          JR      Z,APND2A        ;No - don't backspace
              Ø5720 ;
              Ø5730 ;        SPARM specified - Backspace one byte
              Ø5740 ;
24C5 21BC2B   Ø5750          LD      HL,FCB2+9       ;HL => LRL of FCB #2
24C8 46       Ø5760          LD      B,(HL)          ;P/u current dest LRL
24C9 3601     Ø5770          LD      (HL),1          ;Reset LRL=1
24CB          Ø5780          @@BKSP                  ;Backspace 1 byte
24CB 3E3D     ØØØ22          LD      A,61
```

LBCOPYA - APPEND Mainline

```
24CD EF          ØØØ23          RST     4Ø
24CE 7Ø          Ø579Ø          LD      (HL),B          ;Reset LRL back
                 Ø58ØØ ;
                 Ø581Ø ;        Replace the I/O buffer in FCB #2
                 Ø582Ø ;
24CF 21ØØ2F      Ø583Ø APND2A   LD      HL,BUF2         ;HL => New buffer addr
24D2 22B62B      Ø584Ø          LD      (FCB2+3),HL     ;Stuff in FCB
24D5 C3F426      Ø585Ø          JP      BYTIOØ          ;
                 Ø586Ø ;
                 Ø587Ø ;        EOF on page boundary, use sector I/O
                 Ø588Ø ;
24D8 ED4B132C    Ø589Ø APND3    LD      BC,(FCB1+12)    ;P/u ERN of source
24DC 78          Ø59ØØ          LD      A,B             ;If source is a null
24DD B1          Ø591Ø          OR      C               ;  file, don't do any
24DE CAA326      Ø592Ø          JP      Z,GEOF3         ;  appending, just close
                 Ø593Ø ;
                 Ø594Ø ;        Write Ending Record Number
                 Ø595Ø ;
24E1 2ABF2B      Ø596Ø          LD      HL,(FCB2+12)    ;P/u ERN of dest
24E4 E5          Ø597Ø          PUSH    HL              ;Save it for later
24E5 Ø9          Ø598Ø          ADD     HL,BC           ;Add the two to find new
24E6 44          Ø599Ø          LD      B,H             ;  ERN & Xfer new ERN to BC
24E7 4D          Ø6ØØØ          LD      C,L
24E8 CDB526      Ø6Ø1Ø          CALL    WRERN           ;Write a data sector
24EB E1          Ø6Ø2Ø          POP     HL              ;Recover original ERN
24EC 22BF2B      Ø6Ø3Ø          LD      (FCB2+12),HL    ;  & reset FCB to it
24EF             Ø6Ø4Ø          @@PEOF                  ;Position to end of file
24EF 3E41        ØØØ24          LD      A,65
24F1 EF          ØØØ25          RST     4Ø
24F2 C31126      Ø6Ø5Ø          JP      XFER5
24F5             Ø6Ø6Ø          SUBTTL  '<LBCOPYA - COPY Mainline>'
```

LBCOPYA - COPY Mainline

```
24F5            Ø6Ø7Ø          PAGE
                Ø6Ø8Ø ;
                Ø6Ø9Ø ;        COPY Entry Point - was <BREAK> hit ?
                Ø61ØØ ;
                Ø611Ø COPYST
24F5            Ø612Ø          @@CKBRKC                ;Check for break
24F5 3E6A       ØØØ26          LD      A,1Ø6
24F7 EF         ØØØ27          RST     4Ø
24F8 28Ø4       Ø613Ø          JR      Z,COPYSTA       ;Continue if not
24FA 21FFFF     Ø614Ø          LD      HL,-1           ;  else abort
24FD C9         Ø615Ø          RET
                Ø616Ø ;
                Ø617Ø ;        <BREAK> not hit - execute module
                Ø618Ø ;
24FE ED734924   Ø619Ø COPYSTA  LD      (SAVESP+1),SP   ;Save SP address
25Ø2 CDØ825     Ø62ØØ          CALL    COPYCD          ;Execute Copy code
25Ø5 C31324     Ø621Ø          JP      EXIT            ;Go to common exit
                Ø622Ø ;
                Ø623Ø ;        COPYCD - Copy spec to spec
                Ø624Ø ;
                Ø625Ø COPYCD
25Ø8 CDC228     Ø626Ø          CALL    DOINIT          ;Set high mem test byte
                Ø627Ø ;
                Ø628Ø ;        Check if Source Filespec is legal
                Ø629Ø ;
25ØB 11Ø72C     Ø63ØØ          LD      DE,FCB1         ;DE => Source FCB
25ØE            Ø631Ø          @@FSPEC                 ;Check out filespec
25ØE 3E4E       ØØØ28          LD      A,78
2510 EF         ØØØ29          RST     4Ø
2511 C22924     Ø632Ø          JP      NZ,SPCREQ       ;NZ - Filespec required
                Ø633Ø ;
                Ø634Ø ;        Check if Destination Filespec is legal
                Ø635Ø ;
2514 11B32B     Ø636Ø          LD      DE,FCB2         ;DE => Destination FCB
2517            Ø637Ø          @@FSPEC                 ;Check out filespec
2517 3E4E       ØØØ3Ø          LD      A,78
2519 EF         ØØØ31          RST     4Ø
251A C43Ø27     Ø638Ø          CALL    NZ,CVRTUC       ;Convert line to U/C
                Ø639Ø ;
                Ø64ØØ ;        Process any parameters entered
                Ø641Ø ;
251D 11942B     Ø642Ø COPY1    LD      DE,COPYTBL      ;DE => Parameter Table
2520            Ø643Ø          @@PARAM                 ;Check out parameters
2520 3E11       ØØØ32          LD      A,17
2522 EF         ØØØ33          RST     4Ø
2523 C21824     Ø644Ø          JP      NZ,IOERR        ;NZ - Parameter Error
                Ø645Ø ;
                Ø646Ø ;        Test if X parameter was entered
                Ø647Ø ;
2526 11ØØØØ     Ø648Ø XPARM    LD      DE,$-$          ;P/u (X) parm - We don't
2529 7A         Ø649Ø          LD      A,D             ;  XFER devices
252A B3         Ø65ØØ          OR      E               ;
252B 2ØØB       Ø651Ø          JR      NZ,XFER         ;
                Ø652Ø ;
                Ø653Ø ;        Is the Source or Destination a device ?
                Ø654Ø ;
252D CDEA29     Ø655Ø          CALL    CKDEV           ;Device ?
253Ø CAD526     Ø656Ø          JP      Z,BYTEIO        ;Yes - use byte I/O
                Ø657Ø ;
                Ø658Ø ;        Pick up Defaults for source and destination
```

LBCOPYA - COPY Mainline

```
                    06590 ;
2533 CD5A27         06600           CALL    PRSPC           ;P/u defaults
2536 185B           06610           JR      OPNSRC
                    06620 ;
                    06630 ;         XFER initialization code
                    06640 ;
2538                06650 XFER      @@FLAGS                 ;Position IY to flags
2538 3E65           00034           LD      A,101
253A EF             00035           RST     40
253B FDCB126E       06660           BIT     5,(IY+SFLAG$)   ;DO in Effect ?
253F C22D24         06670           JP      NZ,NOINDO       ;Yes - abort
                    06680 ;
                    06690 ;         If the Source or Dest is a Device - abort
                    06700 ;
2542 CDEA29         06710           CALL    CKDEV           ;Device ?
2545 CA2924         06720           JP      Z,SPCREQ        ;Yes - Filespecs required
                    06730 ;
                    06740 ;         P/u Drivespec of Source Filespec if entered
                    06750 ;
2548 21072C         06760           LD      HL,FCB1         ;HL => FCB #1
254B 0E00           06770           LD      C,0             ;Init to drive zero
                    06780 ;
                    06790 ;         Loop to Pick up Drive # or terminator
                    06800 ;
254D 7E             06810 XFER1     LD      A,(HL)          ;Look for drive spec
254E 23             06820           INC     HL
254F FE3A           06830           CP      ':'             ;Colon indicator?
2551 2806           06840           JR      Z,XFER2         ;Jump if found
2553 FE20           06850           CP      ' '             ;Jump on end
2555 3806           06860           JR      C,XFER3
2557 18F4           06870           JR      XFER1           ;Loop
                    06880 ;
                    06890 ;         Colon indicator present - p/u drive #
                    06900 ;
2559 7E             06910 XFER2     LD      A,(HL)          ;P/u user drive
255A D630           06920           SUB     '0'             ;Cvrt to binary
255C 4F             06930           LD      C,A             ;  & stuff in C
                    06940 ;
                    06950 ;         Save Source drive number
                    06960 ;
255D 211C29         06970 XFER3     LD      HL,XFRDRV+1     ;HL => Drive #
2560 71             06980           LD      (HL),C          ;Save drive # for later
                    06990 ;
                    07000 ;         Stuff drive # into Prompt strings
                    07010 ;
2561 3E30           07020           LD      A,'0'           ;Cvt drive # to ASCII
2563 81             07030           ADD     A,C
2564 32322B         07040           LD      (SRC_DR),A      ;Source Drive #
2567 326C2B         07050           LD      (DEST_DR),A     ;Destination Drive #
                    07060 ;
                    07070 ;         Transfer source FCB to destination FCB
                    07080 ;
256A 21072C         07090           LD      HL,FCB1         ;HL => Source FCB
256D 11B32B         07100           LD      DE,FCB2         ;DE => Destination FCB
2570 012000         07110           LD      BC,32           ;32 bytes to Xfer
2573 EDB0           07120           LDIR                    ;Xfer
                    07130 ;
2575 CDE429         07140           CALL    GETSYS2         ;Load SYS2 for OPEN
                    07150 ;
```

LBCOPYA - COPY Mainline

```
                    07160 ;       Flash "Insert Source Disk" Message
                    07170 ;
2578 21122B         07180         LD      HL,PMTSRC$      ;Prompt for source
257B CD7729         07190         CALL    FLASH           ;  and wait for <ENTER>
                    07200 ;
                    07210 ;       Read in the GAT of the source disk
                    07220 ;
257E 3A1C29         07230         LD      A,(XFRDRV+1)    ;P/u source drive
2581 4F             07240         LD      C,A             ;Stuff in C
2582 CDF629         07250         CALL    RDGAT           ;Read in GAT
2585 C21824         07260         JP      NZ,IOERR        ;Abort on GAT error
                    07270 ;
                    07280 ;       Xfer Password, Name, & Date to destination
                    07290 ;
2588 21CE2D         07300         LD      HL,GAT+ØCEH     ;Disk pw, name, date
258B 11272C         07310         LD      DE,SRCSTR       ;DE => Destination
258E 011200         07320         LD      BC,18
2591 EDB0           07330         LDIR                    ;Xfer
                    07340 ;
                    07350 ;       OPEN the Source File with LRL of 256
                    07360 ;
2593 CD1B2A         07370 OPNSRC  CALL    OPENSRC         ;Open source file
2596 CD5B28         07380         CALL    PUTSOUR         ;Xfer source filespec
2599 CDA528         07390         CALL    GETCLON         ;Get clone data
                    07400 ;
                    07410 ;       Pick up Source LRL
                    07420 ;
259C 7D             07430         LD      A,L             ;Pt back to LRL of source
259D D610           07440         SUB     16
259F 6F             07450         LD      L,A
25A0 7E             07460         LD      A,(HL)          ;P/u source LRL
                    07470 ;
                    07480 ;       Save LRL from source FCB or LRL Parameter
                    07490 ;
25A1 0100FF         07500 LPARM   LD      BC,ØFF00H       ;P/u LRL
25A4 04             07510         INC     B
25A5 2001           07520         JR      NZ,USEREGC      ;If parm entered, use it
25A7 4F             07530         LD      C,A
25A8 217D26         07540 USEREGC LD      HL,GEOF1+1      ;HL => stuff LRL here
25AB 71             07550         LD      (HL),C          ;Stuff LRL for close here
                    07560 ;
                    07570 ;       Ignore this if not COPY (X)
                    07580 ;
25AC 3A2725         07590         LD      A,(XPARM+1)     ;Bypass if not (X)
25AF B7             07600         OR      A
25B0 282D           07610         JR      Z,OPNDST
                    07620 ;
                    07630 ;       Flash "Insert Destination Disk" message
                    07640 ;
25B2 21472B         07650         LD      HL,PMTDST$      ;Prompt destination
25B5 CD7729         07660         CALL    FLASH           ;Flash until loaded
                    07670 ;
                    07680 ;       Read in GAT of Destination Disk
                    07690 ;
25B8 3A1C29         07700         LD      A,(XFRDRV+1)    ;P/u drive
25BB 4F             07710         LD      C,A             ;Read GAT from dest
25BC CDF629         07720         CALL    RDGAT
25BF C21824         07730         JP      NZ,IOERR        ;Jump on GAT read error
                    07740 ;
```

LBCOPYA - COPY Mainline

```
                07750 ;       Xfer Name, Password & Date to destination
                07760 ;
25C2 21CE2D     07770         LD      HL,GAT+0CEH     ;HL => GAT + X'CE'
25C5 11392C     07780         LD      DE,DSTSTR       ;DE => Destination
25C8 011200     07790         LD      BC,18           ;To match up when
25CB D5         07800         PUSH    DE
25CC EDB0       07810         LDIR                    ;  swapping disks
25CE D1         07820         POP     DE              ;Restore Dest ptr
                07830 ;
                07840 ;       Check if Source ID = Destination ID
                07850 ;
25CF 21272C     07860         LD      HL,SRCSTR       ;Compare source & dest
25D2 0612       07870         LD      B,18            ;CANNOT be same
25D4 CD6C29     07880         CALL    CPRHLDE         ;Ck MPW, PackID, Date
25D7 2006       07890         JR      NZ,OPNDST       ;Bypass if different
                07900 ;
                07910 ;       Display "Source & Dest. Disks Identical"
                07920 ;
25D9 CD1629     07930         CALL    PMTSYS          ;Prompt for SYSTEM
25DC C32524     07940         JP      SAMERR          ;Disk packs are identical
                07950 ;
                07960 ;       OPEN the destination File
                07970 ;
25DF 11B32B     07980 OPNDST  LD      DE,FCB2         ;DE => FCB #2
25E2 21002E     07990         LD      HL,BUF1         ;HL => I/O buffer #1
25E5 CD172A     08000         CALL    INITDES         ;Init the file
25E8 CD6128     08010         CALL    PUTDEST         ;Xfer Dest filespec
                08020 ;
                08030 ;       Check if X parm entered
                08040 ;
25EB 3A2725     08050         LD      A,(XPARM+1)     ;If (X), then source &
25EE B7         08060         OR      A               ;  dest can be same file
25EF 200D       08070         JR      NZ,XF2          ;Bypass if (X)
                08080 ;
                08090 ;       Does Source & Dest. have same DEC & drive #
                08100 ;
25F1 2A0D2C     08110         LD      HL,(FCB1+6)     ;If SRC & DST have same
25F4 ED5BB92B   08120         LD      DE,(FCB2+6)     ;  DEC & drive, they are
25F8 AF         08130         XOR     A               ;  identical, abort if so
25F9 ED52       08140         SBC     HL,DE
25FB CA3524     08150         JP      Z,DSTREQ        ;Same - dest spec needed
                08160 ;
                08170 ;       Write revised ERN for space check
                08180 ;
25FE CD7228     08190 XF2     CALL    CPYFILE         ;"Copying : ..."
2601 ED4B132C   08200         LD      BC,(FCB1+12)    ;P/u ESN
2605 CDB526     08210         CALL    WRERN           ;Write a FORMAT sector
                08220 ;
                08230 ;       Reset Destination ESN to Zero
                08240 ;
2608 210000     08250         LD      HL,0            ;Rewind file
260B 22BF2B     08260         LD      (FCB2+12),HL    ;
260E            08270         @@REW                   ;Rewind the file
260E 3E44       00036         LD      A,68
2610 EF         00037         RST     40
                08280 ;
2611 CD2829     08290 XFER5   CALL    PMTSRC          ;Display "Insert source"
                08300 ;
                08310 ;       Stuff Correct Buffer Address in Source FCB
```

LBCOPYA - COPY Mainline

```
                    08320 ;
2614 21002E         08330           LD      HL,BUF1         ;Stuff in FCB
2617 220A2C         08340 RDREC1    LD      (FCB1+3),HL     ;Set buffer addr
                    08350 ;
                    08360 ;         Read in a Source Sector
                    08370 ;
261A 11072C         08380           LD      DE,FCB1         ;DE => Source FCB
261D                08390           @@READ                  ;Read a sector
261D 3E43           00038           LD      A,67
261F EF             00039           RST     40
2620 280B           08400           JR      Z,RDREC2        ;Bypass if no error
                    08410 ;
                    08420 ;         Some sort of I/O Error - Check it out
                    08430 ;
2622 FE1C           08440           CP      1CH             ;EOF?
2624 2827           08450           JR      Z,GOTEOF
2626 FE1D           08460           CP      1DH             ;NRN>ERN?
2628 2823           08470           JR      Z,GOTEOF
262A C31824         08480           JP      IOERR           ;Abort
                    08490 ;
                    08500 ;         Successful READ - is there enough memory ?
                    08510 ;
262D 24             08520 RDREC2    INC     H               ;Bump memory pointer
262E 7C             08530           LD      A,H             ;Go past top?
262F FE00           08540 RDREC3    CP      $-$
2631 20E4           08550           JR      NZ,RDREC1       ;Loop if not
                    08560 ;
                    08570 ;         Read in all we could - display "Insert Dest"
                    08580 ;
2633 CD4A29         08590           CALL    PMTDST          ;Get destination
                    08600 ;
                    08610 ;         Stuff Source FCB buffer into Destination FCB
                    08620 ;
2636 21002E         08630           LD      HL,BUF1         ;Set buffer start
2639 22B62B         08640 RDREC4    LD      (FCB2+3),HL
                    08650 ;
                    08660 ;         Loop to WRITE Destination file
                    08670 ;
263C 11B32B         08680           LD      DE,FCB2         ;DE => Destination FCB
263F                08690           @@WRITE                 ;Write a sector
263F 3E4B           00040           LD      A,75
2641 EF             00041           RST     40
2642 C21824         08700           JP      NZ,IOERR        ;Jump on write error
                    08710 ;
                    08720 ;         Bump memory ptr & check if finished
                    08730 ;
2645 24             08740           INC     H               ;Else bump memory pointer
2646 7C             08750           LD      A,H             ;At top?
2647 FE00           08760 RDREC5    CP      $-$
2649 20EE           08770           JR      NZ,RDREC4       ;Loop if not
264B 18C4           08780           JR      XFER5           ;Else go back to source
                    08790 ;
                    08800 ;         Got EOF error from source - Write out EOF
                    08810 ;
264D CDF228         08820 GOTEOF    CALL    GEOF5           ;Write any memory left
2650 2A0F2C         08830           LD      HL,(FCB1+8)     ;P/u EOF & LRL
2653 22BB2B         08840           LD      (FCB2+8),HL     ;Xfer to FCB2
                    08850 ;
                    08860 ;         Get @CLOSE module if needed
```

LBCOPYA - COPY Mainline

```
                    Ø887Ø ;
2656 CD1629         Ø888Ø           CALL    PMTSYS          ;Prompt SYSTEM if needed
2659 CDE729         Ø889Ø           CALL    GETSYS3         ;Load SYS3 for CLOSE
265C 3A1C29         Ø89ØØ           LD      A,(XFRDRV+1)    ;P/u drive #
265F B7             Ø891Ø           OR      A               ;Is it zero ?
266Ø CC4A29         Ø892Ø           CALL    Z,PMTDST        ;Get dest if drive Ø
                    Ø893Ø ;
                    Ø894Ø ;         Close the destination file
                    Ø895Ø ;
2663 ED4BB92B       Ø896Ø           LD      BC,(FCB2+6)     ;P/u drive # & DEC
2667 11B32B         Ø897Ø           LD      DE,FCB2         ;DE => Destin file FCB
266A                Ø898Ø           @@CLOSE                 ;Close the dest file
266A 3E3C           ØØØ42           LD      A,6Ø
266C EF             ØØØ43           RST     4Ø
266D C21824         Ø899Ø           JP      NZ,IOERR        ;Jump on error
                    Ø9ØØØ ;
                    Ø9Ø1Ø ;         Get the destination file directory record
                    Ø9Ø2Ø ;
267Ø                Ø9Ø3Ø           @@DIRRD                 ;Get destin dir entry
267Ø 3E57           ØØØ44           LD      A,87
2672 EF             ØØØ45           RST     4Ø
2673 C21824         Ø9Ø4Ø           JP      NZ,IOERR        ;I/O error - abort
                    Ø9Ø5Ø ;
                    Ø9Ø6Ø ;         Stuff New LRL into directory entry
                    Ø9Ø7Ø ;
2676 C5             Ø9Ø8Ø           PUSH    BC              ;Save drive & DEC
                    Ø9Ø9Ø ;
2677 E5             Ø91ØØ           PUSH    HL              ;HL => DIR+Ø of dest
2678 3EØ4           Ø911Ø           LD      A,4             ;Posn to LRL byte
267A 85             Ø912Ø           ADD     A,L             ;
267B 6F             Ø913Ø           LD      L,A             ;HL => DIR+4 (LRL)
267C 36ØØ           Ø914Ø GEOF1     LD      (HL),$-$        ;GEOF1+1 contains LRL
267E E1             Ø915Ø           POP     HL              ;Restore HL
                    Ø916Ø ;
                    Ø917Ø ;         Pick up the Clone Parameter
                    Ø918Ø ;
267F 11FFFF         Ø919Ø CPARM     LD      DE,-1           ;Default = ON
2682 7A             Ø92ØØ           LD      A,D             ;Was it changed ?
2683 B3             Ø921Ø           OR      E
2684 2811           Ø922Ø           JR      Z,GEOF2         ;CLONE = N
                    Ø923Ø ;
                    Ø924Ø ;         CLONE = Yes, Transfer Attributes & Date
                    Ø925Ø ;
2686 EB             Ø926Ø           EX      DE,HL
2687 214B2C         Ø927Ø           LD      HL,CLONSAV      ;HL => Attr, DE => DIR+Ø
268A Ø1Ø3ØØ         Ø928Ø           LD      BC,3            ;Move in prot/date, etc
268D EDBØ           Ø929Ø           LDIR
                    Ø93ØØ ;
                    Ø931Ø ;         Transfer Password fields to entry
                    Ø932Ø ;
268F 3EØD           Ø933Ø           LD      A,13            ;Pt to dir pswd fields
2691 83             Ø934Ø           ADD     A,E
2692 5F             Ø935Ø           LD      E,A             ;DE => DIR+16
2693 ØE04           Ø936Ø           LD      C,4             ;BC = 4 bytes to xfer
2695 EDBØ           Ø937Ø           LDIR
                    Ø938Ø ;
                    Ø939Ø ;         Write out Directory entry
                    Ø94ØØ ;
2697 C1             Ø941Ø GEOF2     POP     BC              ;Rcvr drive & DEC
```

LBCOPYA - COPY Mainline

```
2698            Ø942Ø          @@DIRWR                  ;Write Sector with entry
2698 3E58       ØØØ46          LD      A,88
269A EF         ØØØ47          RST     4Ø
269B 18ØC       Ø943Ø          JR      GEOF4            ;Go to Error check
                Ø944Ø ;
                Ø945Ø ;        CLOSE the destination file
                Ø946Ø ;
269D 21FFFF     Ø947Ø GEOF3A   LD      HL,-1            ;Abort JCL
26AØ 22BØ26     Ø948Ø          LD      (RETCOD+1),HL    ;  if <BREAK> hit
26A3 11B32B     Ø949Ø GEOF3    LD      DE,FCB2          ;DE => Destination FCB
26A6            Ø95ØØ          @@CLOSE                  ;Close the file
26A6 3E3C       ØØØ48          LD      A,6Ø
26A8 EF         ØØØ49          RST     4Ø
26A9 C21824     Ø951Ø GEOF4    JP      NZ,IOERR         ;I/O Error - Abort
                Ø952Ø ;
                Ø953Ø ;        Flash "Insert SYSTEM disk" & exit
                Ø954Ø ;
26AC CD1629     Ø955Ø GOHOME   CALL    PMTSYS           ;Prompt SYSTEM if needed
26AF 21ØØØØ     Ø956Ø RETCOD   LD      HL,$-$           ;P/u return code Ø=good
26B2 C34824     Ø957Ø          JP      SAVESP           ;Finished
                Ø958Ø ;
                Ø959Ø ;        WRERN - Write a format sector on FILE #2
                Ø96ØØ ;
26B5 11B32B     Ø961Ø WRERN    LD      DE,FCB2          ;DE => File #2 FCB
26B8 78         Ø962Ø          LD      A,B              ;Don't bother to write
26B9 B1         Ø963Ø          OR      C                ;  a sector if source
26BA C8         Ø964Ø          RET     Z                ;  is empty
                Ø965Ø ;
                Ø966Ø ;        Position to ERN of File #2
                Ø967Ø ;
26BB ØB         Ø968Ø          DEC     BC               ;Adj for ERN
26BC            Ø969Ø          @@POSN                   ;Position to ERN
26BC 3E42       ØØØ5Ø          LD      A,66
26BE EF         ØØØ51          RST     4Ø
26BF D5         Ø97ØØ          PUSH    DE               ;Save FCB ptr
                Ø971Ø ;
                Ø972Ø ;        Fill a buffer of X'E5's
                Ø973Ø ;
26CØ 21ØØ2E     Ø974Ø          LD      HL,BUF1          ;HL => I/O buffer
26C3 11Ø12E     Ø975Ø          LD      DE,BUF1+1        ;DE => I/O buffer+1
26C6 Ø1FFØØ     Ø976Ø          LD      BC,255           ;255+1 bytes to fill
26C9 36E5       Ø977Ø          LD      (HL),ØE5H        ;Format byte = X'E5'
26CB EDBØ       Ø978Ø          LDIR                     ;Fill buffer
                Ø979Ø ;
                Ø98ØØ ;        Write ERN of File #2
                Ø981Ø ;
26CD D1         Ø982Ø          POP     DE               ;DE => FCB #2
26CE            Ø983Ø          @@WRITE                  ;Write sector
26CE 3E4B       ØØØ52          LD      A,75
26DØ EF         ØØØ53          RST     4Ø
26D1 C8         Ø984Ø          RET     Z                ;RETurn if no error
26D2 C31824     Ø985Ø          JP      IOERR            ;Error - abort
                Ø986Ø ;
                Ø987Ø ;        BYTEIO - OPEN Source or dest using byte I/O
                Ø988Ø ;
26D5 CD1B2A     Ø989Ø BYTEIO   CALL    OPENSRC          ;OPEN source file
26D8 CD5B28     Ø99ØØ          CALL    PUTSOUR          ;Get source filespec
                Ø991Ø ;
                Ø992Ø ;        INIT the dest device with LRL from parm
```

LBCOPYA - COPY Mainline

```
                    Ø993Ø ;
26DB 3AA225         Ø994Ø           LD      A,(LPARM+1)     ;P/u LRL from Parm
26DE 47             Ø995Ø           LD      B,A             ;Open destination
26DF 11B32B         Ø996Ø           LD      DE,FCB2         ;DE => FCB #2
26E2 21ØØ2F         Ø997Ø           LD      HL,BUF2         ;Different buffer
26E5 3E3A           Ø998Ø           LD      A,@INIT         ;@INIT SVC #
26E7 CD2C2A         Ø999Ø           CALL    GETFILE         ;Issue it
26EA CD6128         1ØØØØ           CALL    PUTDEST         ;Get dest devspec
26ED CD7228         1ØØ1Ø           CALL    CPYFILE         ;"Copying/Appending : .."
26FØ AF             1ØØ2Ø           XOR     A               ;Reset LRL = Ø
26F1 32BC2B         1ØØ3Ø           LD      (FCB2+9),A      ;For sector I/O
                    1ØØ4Ø ;
                    1ØØ5Ø ;         Turn on cursor
                    1ØØ6Ø ;
26F4 ØEØE           1ØØ7Ø BYTIOØ    LD      C,14            ;Turn cursor on
26F6 CDCA29         1ØØ8Ø           CALL    DISPB           ;Display byte
                    1ØØ9Ø ;
                    1Ø1ØØ ;         BYTIO1 Loop - File - Dev, Dev - File, Dev - Dev
                    1Ø11Ø ;         Was the <BREAK> key hit ?
                    1Ø12Ø ;
                    1Ø13Ø BYTIO1
26F9 CD1229         1Ø14Ø           CALL    CKBRK           ;Was the <BREAK> key
26FC C29D26         1Ø15Ø E_O_F     JP      NZ,GEOF3A       ;  hit ????
                    1Ø16Ø ;
                    1Ø17Ø ;         The <BREAK> was not hit - get a character
                    1Ø18Ø ;
26FF 11Ø72C         1Ø19Ø           LD      DE,FCB1         ;DE => Source FCB
2702                1Ø2ØØ           @@GET                   ;Get a byte
2702 3E03           ØØØ54           LD      A,3
2704 EF             ØØØ55           RST     4Ø
2705 28ØB           1Ø21Ø           JR      Z,BYTIO4        ;Good - stuff it
                    1Ø22Ø ;
                    1Ø23Ø ;         If Error # = Ø, then try @GET again
                    1Ø24Ø ;
2707 B7             1Ø25Ø           OR      A               ;Error # = Ø ?
2708 28EF           1Ø26Ø           JR      Z,BYTIO1        ;Yes - @GET again
                    1Ø27Ø ;
                    1Ø28Ø ;         Is the Error an "End of File" error ?
                    1Ø29Ø ;
270A FE1C           1Ø3ØØ           CP      1CH             ;EOF?
270C CAA326         1Ø31Ø           JP      Z,GEOF3         ;Yes - finished
270F C31824         1Ø32Ø           JP      IOERR           ;I/O error - abort
                    1Ø33Ø ;
                    1Ø34Ø ;         Was the source character a <BREAK> ?
                    1Ø35Ø ;
2712 FE8Ø           1Ø36Ø BYTIO4    CP      BREAK           ;<BREAK> character ?
2714 2ØØ7           1Ø37Ø           JR      NZ,BYTIO4A      ;No - @PUT it
                    1Ø38Ø ;
                    1Ø39Ø ;         Source = <BREAK> --- is the BREAK bit set ?
                    1Ø4ØØ ;
2716 CD1229         1Ø41Ø           CALL    CKBRK           ;<BREAK> bit set ?
2719 2ØE1           1Ø42Ø           JR      NZ,E_O_F        ;Yes - stop
271B 3E8Ø           1Ø43Ø           LD      A,BREAK         ;Restore A
                    1Ø44Ø ;
                    1Ø45Ø ;         Output byte to destination
                    1Ø46Ø ;
271D 11B32B         1Ø47Ø BYTIO4A   LD      DE,FCB2         ;DE => Dest. Device/File
272Ø 4F             1Ø48Ø           LD      C,A             ;Stuff byte in C for @PUT
2721                1Ø49Ø           @@PUT                   ;Output byte
```

LBCOPYA - COPY Mainline

```
2721 3EØ4      ØØØ56          LD      A,4
2723 EF        ØØØ57          RST     4Ø
2724 C21824    1Ø5ØØ          JP      NZ,IOERR        ;NZ - I/O Error
               1Ø51Ø ;
               1Ø52Ø ;        Echo byte if parameter set
               1Ø53Ø ;
2727 11ØØØØ    1Ø54Ø EPARM    LD      DE,$-$          ;P/u ECHO parm
272A 14        1Ø55Ø          INC     D               ;Specified ?
272B CCCA29    1Ø56Ø          CALL    Z,DISPB         ;Echo byte
272E 18C9      1Ø57Ø          JR      BYTIO1          ;Go til EOF or BREAK
               1Ø58Ø ;
273Ø           ØØ25Ø *GET     LBCOPYB:3
               1Ø59Ø ;LBCOPYB/ASM - COPY/APPEND Common Routines
273Ø           1Ø6ØØ          SUBTTL  '<LBCOPYB - COPY/APPEND Common Routines>'
```

LBCOPYB - COPY/APPEND Common Routines

```
2730            10610           PAGE
                10620 ;
                10630 ;         CVRTUC - Transfer partspec & convert to U/C
                10640 ;         HL => Buffer containing characters to parse
                10650 ;         DE => Destination of converted line
                10660 ;
2730 CD3827     10670 CVRTUC    CALL    CVRT            ;Convert line
2733 3E0D       10680           LD      A,CR            ;Ensure end-of-line
2735 12         10690           LD      (DE),A          ;  with a carriage return
2736 C9         10700           RET
                10710 ;
                10720 ;         Position to First non-space character
                10730 ;
2737 23         10740 CVRT0     INC     HL
2738 7E         10750 CVRT      LD      A,(HL)          ;P/u possible dest char
2739 FE0D       10760           CP      CR              ;Exit on CR
273B C8         10770           RET     Z
273C FE20       10780           CP      ' '             ;Loop on space
273E 28F7       10790           JR      Z,CVRT0
2740 2B         10800           DEC     HL              ;Backup to 1st separator
                10810 ;
                10820 ;         HL => Filespec or Devspec, convert to U/C
                10830 ;
2741 0620       10840           LD      B,32            ;Max 32 chars
2743 7E         10850 COP1      LD      A,(HL)          ;Transfer the partial
2744 FE61       10860           CP      'a'             ;Cvrt lc to uc
2746 3806       10870           JR      C,COP2
2748 FE7B       10880           CP      'z'+1
274A 3002       10890           JR      NC,COP2
274C CBAF       10900           RES     5,A
274E 12         10910 COP2      LD      (DE),A          ;Filespec until paren
274F FE0D       10920           CP      CR              ;RETurn if terminator
2751 C8         10930           RET     Z
2752 FE28       10940           CP      '('             ;Parameter ?
2754 C8         10950           RET     Z
2755 23         10960           INC     HL              ;Bump ptr
2756 13         10970           INC     DE
2757 10EA       10980           DJNZ    COP1            ;32 characters max
2759 C9         10990           RET
                11000 ;
                11010 ;         PRSPC - Match Source & Dest. specs for defaults
                11020 ;
275A 21072C     11030 PRSPC     LD      HL,FCB1         ;HL => Source FCB
275D 11B32B     11040           LD      DE,FCB2         ;DE => Destin FCB
                11050 ;
                11060 ;         Abort if first character is illegal
                11070 ;
2760 1A         11080           LD      A,(DE)          ;P/u 1st char of dest
2761 FE0D       11090           CP      CR              ;Is it a C/R ?
2763 CA3524     11100           JP      Z,DSTREQ        ;"Destination spec req"
2766 FE30       11110           CP      '0'             ;Numeric ?
2768 3805       11120           JR      C,CHKDEST
276A FE3A       11130           CP      '9'+1
276C DA3524     11140           JP      C,DSTREQ        ;Yes - "Dest spec req"
                11150 ;
276F D5         11160 CHKDEST   PUSH    DE              ;Save Destination FCB ptr
                11170 ;
                11180 ;         If no dest. filename, xfer source filename
                11190 ;
2770 1A         11200           LD      A,(DE)          ;P/u a dest char
```

LBCOPYB - COPY/APPEND Common Routines

```
2771 FE41     11210         CP      'A'             ;Filename present ?
2773 DCAA27   11220         CALL    C,PRSPC7        ;No - xfer src filename
2776 Ø62F     11230         LD      B,'/'           ;With an alpha, init to
2778 CD8327   11240         CALL    PRSPC2          ;  test for extension
277B Ø62E     11250         LD      B,'.'           ;Init to test for pswd
277D CD8327   11260         CALL    PRSPC2
2780 D1       11270         POP     DE
2781 C9       11280         RET
              11290 ;
              11300 ;       PRSPC2 - Transfer Field to destination
              11310 ;       DE => Start of Destination Field
              11320 ;        B  = Delimiter to Position one byte after
              11330 ;
2782 13       11340 PRSPC1  INC     DE
              11350 ;
              11360 ;       Finished with Field (Delimiter hit) ?
              11370 ;
2783 1A       11380 PRSPC2  LD      A,(DE)          ;Is the next char the
2784 B8       11390         CP      B               ;Separator to look for
2785 28ØE     11400         JR      Z,PRSPC3
              11410 ;
              11420 ;       Skip over an existing destination field
              11430 ;
2787 FE41     11440         CP      'A'             ;Alphabetic ?
2789 3ØF7     11450         JR      NC,PRSPC1       ;Yes - skip it
278B FE3Ø     11460         CP      'Ø'             ;Numeric ?
278D 38Ø8     11470         JR      C,PRSPC4        ;No - use source field
278F FE3A     11480         CP      '9'+1           ;Numeric ?
2791 38EF     11490         JR      C,PRSPC1        ;Yes - skip it
              11500 ;
2793 18Ø2     11510         JR      PRSPC4          ;Use source field
              11520 ;
              11530 ;       Hit the delimiter - Position to next char
              11540 ;
2795 13       11550 PRSPC3  INC     DE              ;Position to next field
2796 C9       11560         RET                     ;  and RETurn
              11570 ;
              11580 ;       Scan Source spec & see if it contains field
              11590 ;
2797 E5       11600 PRSPC4  PUSH    HL              ;Save ptr to source
2798 7E       11610 PRSPC5  LD      A,(HL)          ;Grab a source char
2799 23       11620         INC     HL              ;Position to next char
              11630 ;
              11640 ;       Is the character a Terminator ?
              11650 ;
279A FEØ3     11660         CP      ETX             ;End of line ?
279C 28ØA     11670         JR      Z,PRSPC6        ;If so, not in source
279E FEØD     11680         CP      CR              ;End of line?
27AØ 28Ø6     11690         JR      Z,PRSPC6        ;If so, not in source
              11700 ;
              11710 ;       is the character the field delimiter ?
              11720 ;
27A2 B8       11730         CP      B               ;Delimiter ?
27A3 2ØF3     11740         JR      NZ,PRSPC5       ;Nope, continue
              11750 ;
              11760 ;       Transfer Source Field to Destination
              11770 ;
27A5 CDB627   11780         CALL    MVFLD1          ;Yes, xfer the SRC field
              11790 ;
```

LBCOPYB - COPY/APPEND Common Routines

```
27A8 E1       11800 PRSPC6  POP     HL              ;Rcvr source ptr
27A9 C9       11810         RET
              11820 ;
              11830 ;       PRSPC7 - Shoehorn source into destination field
              11840 ;
              11850 ;       Check if 1st char in (HL) is alphanumeric
              11860 ;
27AA 7E       11870 PRSPC7  LD      A,(HL)          ;P/u source char
27AB FE30     11880         CP      '0'             ;RET NC if alphanumeric
27AD D8       11890         RET     C               ;C - not alphanumeric
27AE FE3A     11900         CP      '9'+1           ;Numeric (0-9) ?
27B0 3803     11910         JR      C,MVFLD         ;Xfer if it is
27B2 FE41     11920         CP      'A'             ;Not numeric -
27B4 D8       11930         RET     C               ;Must be alphabetic
              11940 ;
              11950 ;       Shoehorn a source field byte into dest FCB
              11960 ;
27B5 23       11970 MVFLD   INC     HL              ;Bump source pointer
27B6 E5       11980 MVFLD1  PUSH    HL
27B7 62       11990         LD      H,D             ;Xfer dest ptr to HL
27B8 6B       12000         LD      L,E
              12010 ;
              12020 ;       P/u current char, & stuff in last char
              12030 ;
27B9 4E       12040 MVFLD2  LD      C,(HL)          ;P/u dest char
27BA 77       12050         LD      (HL),A          ;Stuff in last character
27BB 23       12060         INC     HL              ;Posn to next char
              12070 ;
              12080 ;       Finished with field ?
              12090 ;
27BC 79       12100         LD      A,C             ;Test dest for
27BD FE03     12110         CP      ETX             ;End of line ?
27BF 2804     12120         JR      Z,MVFLD3
              12130 ;
27C1 FE0D     12140         CP      CR              ;End of line ?
27C3 20F4     12150         JR      NZ,MVFLD2       ;Ripple the destination
              12160 ;
              12170 ;       Done stuffing 1 source byte into dest field
              12180 ;
27C5 77       12190 MVFLD3  LD      (HL),A          ;Stuff the terminator
27C6 E1       12200         POP     HL              ;Restore Source FCB
27C7 13       12210         INC     DE              ;Advance to next pos
27C8 18E0     12220         JR      PRSPC7          ;Go get next source byte
              12230 ;
              12240 ;       DOSVC - Get Filespec/Devspec & Issue @OPEN/@INIT
              12250 ;
27CA F5       12260 DOSVC   PUSH    AF              ;Save SVC #
27CB E5       12270         PUSH    HL              ;Buffer ptr
27CC D5       12280         PUSH    DE              ;FCB
              12290 ;
              12300 ;       Transfer Filespec/Devspec into TEMBUF
              12310 ;
27CD 21D32B   12320         LD      HL,TEMBUF
27D0 1A       12330 TLP     LD      A,(DE)          ;P/u byte from FCB
27D1 FE2F     12340         CP      '/'             ;Extension ?
27D3 2008     12350         JR      NZ,STUFCHR      ;No - save the char
27D5 13       12360         INC     DE              ;Is the next character
27D6 1A       12370         LD      A,(DE)          ;  valid ?
27D7 FE41     12380         CP      'A'
```

LBCOPYB - COPY/APPEND Common Routines

```
27D9 38Ø2      12390          JR      C,STUFCHR       ;No - don't output it
27DB 1B        12400          DEC     DE              ;Back one
27DC 1A        12410          LD      A,(DE)          ;P/u slash
27DD 77        12420 STUFCHR  LD      (HL),A          ;Xfer to TEMBUF
27DE 23        12430          INC     HL
27DF 13        12440          INC     DE
27EØ FEØE      12450          CP      CR+1            ;Done ?
27E2 38Ø8      12460          JR      C,DUN
27E4 FE3A      12470          CP      ':'
27E6 28Ø4      12480          JR      Z,DUN
27E8 FE2E      12490          CP      '.'
27EA 2ØE4      12500          JR      NZ,TLP
               12510 ;
               12520 ;        Found valid terminator - Is this a device ?
               12530 ;
27EC 2B        12540 DUN      DEC     HL              ;Back up to term
27ED D1        12550          POP     DE              ;DE => FCB+Ø
27EE 1A        12560          LD      A,(DE)          ;Device ?
27EF FE2A      12570          CP      '*'
27F1 28Ø7      12580          JR      Z,DUN2          ;Yes - done
27F3 363A      12590          LD      (HL),':'        ;No - overwrite with ":"
27F5 23        12600          INC     HL              ;Bump
27F6 225828    12610          LD      (DSPEC+1),HL    ;Save drivespec location
27F9 23        12620          INC     HL              ;Bump
27FA 36Ø3      12630 DUN2     LD      (HL),ETX        ;End with X'Ø3'
27FC E1        12640          POP     HL              ;HL => Disk I/O buffer
27FD F1        12650          POP     AF              ;A = SVC #
27FE 323228    12660          LD      (SVCNUM+1),A    ;Save SVC #
               12670 ;
               12680 ;        Issue Supervisory Call
               12690 ;
2801 EF        12700          RST     28H             ;Do it
2802 F5        12710          PUSH    AF              ;Save SVC condition
2803 CD4A28    12720          CALL    GETDRIV         ;P/u drv (in C) if file
2806 2817      12730          JR      Z,DEVICE        ;Z - this is a device
               12740 ;
               12750 ;        This is a file - did we @INIT it ?
               12760 ;
2808 3A3228    12770          LD      A,(SVCNUM+1)    ;P/u SVC #
280B FE3A      12780          CP      @INIT           ;Is it @INIT ?
280D 2Ø1Ø      12790          JR      NZ,DEVICE       ;No - don't worry
               12800 ;
               12810 ;        Was the @INIT successful ?
               12820 ;
280F F1        12830          POP     AF              ;Don't perform
2810 F5        12840          PUSH    AF              ;  a @CKDRV if you didn't
2811 2804      12850          JR      Z,DO_CKDR       ;  successfully @INIT it.
2813 FE2A      12860          CP      42              ;LRL Open Fault ?
2815 2ØØ8      12870          JR      NZ,DEVICE       ;No - Don't @CKDRV
               12880 ;
               12890 ;        @INIT was ok - is this a Write Prot Disk?
               12900 ;
2817           12910 DO_CKDR  @@CKDRV                 ;Write Protected Disk ?
2817 3E21      ØØØ58          LD      A,33
2819 EF        ØØØ59          RST     4Ø
281A 3EØF      12920          LD      A,15            ;Init WP disk error code
281C DA1824    12930          JP      C,IOERR         ;Yes - abort
               12940 ;
               12950 ;        Check status on the @INIT we did
```

LBCOPYB - COPY/APPEND Common Routines

```
                    12960 ;
281F F1             12970 DEVICE  POP     AF              ;Recover status
2820 2803           12980         JR      Z,CHKPROT       ;Check PROTection status
                    12990 ;
                    13000 ;       Ignore Error #42 - "LRL Open Fault"
                    13010 ;
2822 FE2A           13020         CP      42              ;Ignore this error
2824 C0             13030         RET     NZ              ;NZ - Abort
                    13040 ;
                    13050 ;       Check if File has proper Access
                    13060 ;
2825 DDCB007E       13070 CHKPROT BIT     7,(IX)          ;Is this a filespec ?
2829 281D           13080         JR      Z,OKYDOKY       ;No - don't check
282B DD7E01         13090         LD      A,(IX+1)        ;P/u protection byte
282E E607           13100         AND     7
2830 47             13110         LD      B,A             ;Xfer to B
2831 3E00           13120 SVCNUM  LD      A,$-$           ;P/u SVC #
2833 FE3A           13130         CP      @INIT           ;@INIT ?
2835 78             13140         LD      A,B             ;P/u protection level
2836 280C           13150         JR      Z,INIT1         ;Z - Must be < 5
2838 FE06           13160         CP      6               ;Read Access ?
283A 380C           13170         JR      C,OKYDOKY       ;Yes - set Z & RETurn
                    13180 ;
                    13190 ;       Illegal Access to protected file
                    13200 ;
283C                13210 ILLACC  @@CLOSE                 ;Close File
283C 3E3C           00060         LD      A,60
283E EF             00061         RST     40
283F 3E19           13220         LD      A,25            ;File Access Denied
2841 C31824         13230         JP      IOERR           ;Error - Regardless
                    13240 ;
2844 FE05           13250 INIT1   CP      5               ;Update Access ?
2846 30F4           13260         JR      NC,ILLACC       ;No - Illegal Access
                    13270 ;
2848 AF             13280 OKYDOKY XOR     A               ;Set Z flag
2849 C9             13290         RET                     ;And RETurn
                    13300 ;
                    13310 ;       GETDRIV - P/u Drive # from FCB & convert to ASCII
                    13320 ;
284A D5             13330 GETDRIV PUSH    DE              ;Get fcb to
284B DDE1           13340         POP     IX              ;  here.
284D 1A             13350         LD      A,(DE)          ;Device ?
284E CB7F           13360         BIT     7,A
2850 C8             13370         RET     Z               ;Device - RETurn
                    13380 ;
                    13390 ;       Stuff Drive # into Buffer
                    13400 ;
2851 DD7E06         13410         LD      A,(IX+6)
2854 4F             13420         LD      C,A             ;Xfer to C
2855 C630           13430         ADD     A,'0'           ;Convert to ASCII
2857 320000         13440 DSPEC   LD      ($-$),A
285A C9             13450         RET                     ;RETurn w/ condition
                    13460 ;
                    13470 ;       PUTSOUR/PUTDEST - Create Src/Dest File/Dev specs
                    13480 ;
285B E5             13490 PUTSOUR PUSH    HL              ;Save HL
285C 21E32B         13500         LD      HL,FROM         ;Xfer there
285F 1804           13510         JR      XBUFF
2861 E5             13520 PUTDEST PUSH    HL              ;Save HL
```

LBCOPYB - COPY/APPEND Common Routines

```
2862 21F72B   13530          LD      HL,TO2
              13540 ;
              13550 ;        XBUFF - Transfer TEMBUF contents to User Buffer
              13560 ;        HL => Destination of Filespec/Devspec
              13570 ;
2865 D5       13580 XBUFF    PUSH    DE              ;Save FCB ptr
2866 11D32B   13590          LD      DE,TEMBUF       ;DE => Source
2869 EB       13600          EX      DE,HL           ;Swap for LDIR
286A Ø1ØFØØ   13610          LD      BC,15           ;15 bytes Max
286D EDBØ     13620          LDIR                    ;Xfer
286F D1       13630          POP     DE              ;DE => FCB+Ø
287Ø E1       13640          POP     HL              ;Recover Buffer ptr
2871 C9       13650          RET                     ;Done
              13660 ;
              13670 ;        CPYFILE - Display Copying/Appending Message
              13680 ;
2872 E5       13690 CPYFILE  PUSH    HL              ;Save registers
2873 D5       13700          PUSH    DE
2874 21332A   13710          LD      HL,COPYMS       ;"Copying : "
2877 3EØ1     13720 APPFLAG  LD      A,$-$+1         ;Append or Copy ?
2879 B7       13730          OR      A
287A 2ØØ3     13740          JR      NZ,ITSCOPY
287C 213D2A   13750          LD      HL,APPDMS       ;"Appending : "
287F CDD129   13760 ITSCOPY  CALL    DSPLY
2882 21E32B   13770          LD      HL,FROM         ;Source
2885 CDD129   13780          CALL    DSPLY
2888 21F32B   13790          LD      HL,TO           ;" TO "
288B CDD129   13800          CALL    DSPLY
288E ØEØD     13810          LD      C,CR            ;End line
289Ø CDCA29   13820          CALL    DISPB           ;Display byte
2893 D1       13830          POP     DE              ;Restore Regs
2894 E1       13840          POP     HL
2895 C9       13850          RET
              13860 ;
              13870 ;        GETLRL - Get LRL from a Directory Entry
              13880 ;        DE => FCB of filespec
              13890 ;        IX <= FCB of filespec
              13900 ;        A  <= LRL of file
              13910 ;
2896 D5       13920 GETLRL   PUSH    DE              ;Xfer fcb to IX
2897 DDE1     13930          POP     IX
2899 DD7EØ7   13940          LD      A,(IX+7)        ;P/u DEC
289C E6EØ     13950          AND     ØEØH            ;Calculate Record Number
289E C6Ø4     13960          ADD     A,4             ;Pt to LRL
28AØ 6F       13970          LD      L,A
28A1 26ØØ     13980 SYSBUF   LD      H,$-$           ;P/u hi-byte of SBUFF$
28A3 7E       13990          LD      A,(HL)          ;P/u LRL
28A4 C9       14000          RET
              14010 ;
              14020 ;        GETCLON - Recover Cloning info from Source
              14030 ;
28A5 D5       14040 GETCLON  PUSH    DE
28A6 DDE1     14050          POP     IX              ;Xfer FCB to IX
28A8 DD7EØ7   14060          LD      A,(IX+7)        ;P/u source DEC
28AB E6EØ     14070          AND     ØEØH            ;Pt to start of record
28AD 6F       14080          LD      L,A             ;Pt to core record
28AE 26ØØ     14090 SYSBUF2  LD      H,$-$           ;P/u hi-byte of SBUFF$
28BØ 114B2C   14100          LD      DE,CLONSAV      ;Save clone info
28B3 Ø1Ø3ØØ   14110          LD      BC,3
```

LBCOPYB - COPY/APPEND Common Routines

```
28B6 EDBØ      14120          LDIR
28B8 3EØD      14130          LD      A,13            ;Pt to password fields
28BA 85        14140          ADD     A,L
28BB 6F        14150          LD      L,A
28BC Ø1Ø4ØØ    14160          LD      BC,4            ;Save them also
28BF EDBØ      14170          LDIR
28C1 C9        14180          RET
               14190 ;
               14200 ;        DOINIT - Initialization for APPEND or COPY
               14210 ;
28C2           14220 DOINIT   EQU     $
               14230 ;
               14240 ;        Calculate high byte of available memory
               14250 ;
28C2 E5        14260          PUSH    HL              ;Save HL
28C3           14270          @@FLAGS                 ;IY => System Flags
28C3 3E65      ØØØ62          LD      A,1Ø1
28C5 EF        ØØØ63          RST     4Ø
28C6 21ØØØØ    14280          LD      HL,Ø            ;P/u HIGH$
28C9 45        14290          LD      B,L
28CA FDCBØ24E  14300          BIT     1,(IY+CFLAG$)   ;@CMNDR ?
28CE 28Ø1      14310          JR      Z,USEHI         ;No  - use HIGH$
28DØ Ø4        14320          INC     B               ;Yes - use LOW$
28D1           14330 USEHI    @@HIGH$
28D1 3E64      ØØØ64          LD      A,1ØØ
28D3 EF        ØØØ65          RST     4Ø
               14340 ;
               14350 ;        Stuff high byte into memory check locations
               14360 ;
28D4 23        14370          INC     HL              ;  256-byte block
28D5 25        14380          DEC     H
28D6 7C        14390          LD      A,H             ;Set up test bytes
28D7 323Ø26    14400          LD      (RDREC3+1),A
28DA 324826    14410          LD      (RDREC5+1),A
28DD 32F728    14420          LD      (GEOF6+1),A
               14430 ;
               14440 ;        Pick up high byte of System Buffer (SBUFF$)
               14450 ;
28EØ Ø1ØØØØ    14460          LD      BC,Ø            ;DEC = Ø, Drive = Ø
28E3           14470          @@DIRRD                 ;Read in BOOT/SYS
28E3 3E57      ØØØ66          LD      A,87
28E5 EF        ØØØ67          RST     4Ø
28E6 7C        14480          LD      A,H             ;P/u high byte of SBUFF$
28E7 32A228    14490          LD      (SYSBUF+1),A    ;Stuff away for LRL
28EA 32AF28    14500          LD      (SYSBUF2+1),A   ;Stuff away for CLONE
               14510 ;
               14520 ;        Reset <PAUSE> & <ENTER> bits
               14530 ;
28ED CDBE29    14540          CALL    RESKFLG         ;Reset bits 1-2
28FØ E1        14550          POP     HL
28F1 C9        14560          RET
               14570 ;
               14580 ;        GEOF5 - Write out the last sector of destination
               14590 ;
28F2 7C        14600 GEOF5    LD      A,H             ;P/u hi-order pointer
28F3 FE2E      14610          CP      BUF1<-8         ;Same as start?
28F5 C8        14620          RET     Z               ;Ret if finished
               14630 ;
               14640 ;        At the end of buffer area ?
```

LBCOPYB - COPY/APPEND Common Routines

```
                14650 ;
28F6 FEØØ       14660 GEOF6   CP      $-$             ;GEOF6+1 = msb of top
28F8 C8         14670         RET     Z               ;Return if finished
                14680 ;
                14690 ;       Flash "Insert Dest. Disk" & write remainder
                14700 ;
28F9 CD4A29     14710         CALL    PMTDST          ;Prompt destination
28FC 44         14720         LD      B,H             ;Save highest used
28FD 21ØØ2E     14730         LD      HL,BUF1         ;Pt to start
                14740 ;
                14750 ;       Loop to write out remainder of dest. file
                14760 ;
2900 22B62B     14770 GEOF7   LD      (FCB2+3),HL     ;Set dest buffer address
2903 11B32B     14780         LD      DE,FCB2         ;DE => Dest. FCB
2906            14790         @@WRITE                 ;Write a record
2906 3E4B       00068         LD      A,75
2908 EF         00069         RST     4Ø
2909 C21824     14800         JP      NZ,IOERR        ;Jump on write error
                14810 ;
                14820 ;       Finished writing ?
                14830 ;
290C 24         14840         INC     H               ;Bump buff ptr
290D 7C         14850         LD      A,H
290E B8         14860         CP      B               ;Finished?
290F 20EF       14870         JR      NZ,GEOF7        ;Loop if not
2911 C9         14880         RET                     ;Return
                14890 ;
                14900 ;       CKBRK - Check for BREAK
                14910 ;       NZ - <BREAK> key was depressed
                14920 ;
                14930 CKBRK
2912            14940         @@CKBRKC                ;<BREAK> hit ?
2912 3E6A       00070         LD      A,106
2914 EF         00071         RST     4Ø
2915 C9         14950         RET                     ;Z - No, NZ - Yes
                14960 ;
                14970 ;       PMTSYS - Prompt for a system disk
                14980 ;
2916 3A2725     14990 PMTSYS  LD      A,(XPARM+1)     ;X parameter entered ?
2919 B7         15000         OR      A
291A C8         15010         RET     Z               ;No - need to prompt
                15020 ;
                15030 ;       If Xfer drive number isn't Ø - don't prompt
                15040 ;
291B 3EØØ       15050 XFRDRV  LD      A,$-$           ;XFRDRV+1 contains drv #
291D B7         15060         OR      A               ;Is it zero ?
291E CØ         15070         RET     NZ              ;Ret if not SYSTEM drive
                15080 ;
                15090 ;       Flash "Insert SYSTEM disk" message
                15100 ;
291F E5         15110         PUSH    HL              ;Save HL
2920 21F32A     15120         LD      HL,PMTSYS$      ;"Insert SYSTEM disk"
2923 CD7729     15130         CALL    FLASH           ;Flash, & RET if <ENTER>
2926 E1         15140         POP     HL              ;Restore HL
2927 C9         15150         RET
                15160 ;
                15170 ;       PMTSRC - Prompt for Source Disk
                15180 ;
2928 3A2725     15190 PMTSRC  LD      A,(XPARM+1)     ;X parameter entered ?
```

LBCOPYB - COPY/APPEND Common Routines

```
292B B7       15200          OR      A
292C C8       15210          RET     Z               ;No - don't display it
              15220 ;
              15230 ;        Flash "Insert Source disk" message
              15240 ;
292D E5       15250          PUSH    HL
292E 21122B   15260          LD      HL,PMTSRC$      ;Init for source
2931 CD7729   15270          CALL    FLASH           ;Dsp msg & await reply
              15280 ;
              15290 ;        Read in the GAT from the Source Disk
              15300 ;
2934 3A1C29   15310          LD      A,(XFRDRV+1)    ;P/u drive
2937 4F       15320          LD      C,A             ;Stuff in C
2938 CDF629   15330          CALL    RDGAT           ;Get GAT
              15340 ;
              15350 ;        Is this the correct source disk ?
              15360 ;
293B 21CE2D   15370          LD      HL,GAT+ØCEH     ;HL => Name & Date field
293E 11272C   15380          LD      DE,SRCSTR       ;DE => Original Source
2941 Ø612     15390          LD      B,18            ;Same source disk ?
2943 CD6C29   15400          CALL    CPRHLDE         ;Z - same, NZ - different
2946 E1       15410          POP     HL              ;Restore HL
2947 2ØDF     15420          JR      NZ,PMTSRC       ;Re-request if not match
2949 C9       15430          RET                     ;Okay
              15440 ;
              15450 ;        PMTDST - Prompt for Destination disk
              15460 ;
294A 3A2725   15470 PMTDST   LD      A,(XPARM+1)     ;X parameter entered ?
294D B7       15480          OR      A
294E C8       15490          RET     Z               ;No - RETurn
              15500 ;
              15510 ;        Flash "Insert Destination disk" message
              15520 ;
294F E5       15530          PUSH    HL
2950 21472B   15540          LD      HL,PMTDST$      ;Pt to msg
2953 CD7729   15550          CALL    FLASH           ;Prompt & await reply
              15560 ;
              15570 ;        Read in GAT from original destination disk
              15580 ;
2956 3A1C29   15590          LD      A,(XFRDRV+1)    ;P/u drive #
2959 4F       15600          LD      C,A             ;Stuff in C
295A CDF629   15610          CALL    RDGAT           ;Get GAT
              15620 ;
              15630 ;        Is this the same destination disk ?
              15640 ;
295D 21CE2D   15650          LD      HL,GAT+ØCEH     ;HL => Name & Date field
2960 11392C   15660          LD      DE,DSTSTR       ;DE => Original dest
2963 Ø612     15670          LD      B,18            ;Same destination disk ?
2965 CD6C29   15680          CALL    CPRHLDE         ;Z - Same, NZ - different
2968 E1       15690          POP     HL              ;Restore HL
2969 2ØDF     15700          JR      NZ,PMTDST       ;Re-request if wrong disk
296B C9       15710          RET
              15720 ;
              15730 ;        CPRHLDE - Compare string @ HL to string @ DE
              15740 ;        B => Number of characters to compare
              15750 ;        Z -  Set if strings match
              15760 ;
296C 1A       15770 CPRHLDE  LD      A,(DE)          ;P/u character @ DE
296D BE       15780          CP      (HL)            ;Same ?
```

LBCOPYB - COPY/APPEND Common Routines

```
296E CØ          15790          RET     NZ              ;Ret (NZ) if no match
296F 23          15800          INC     HL              ;Bump each
2970 13          15810          INC     DE
2971 10F9        15820          DJNZ    CPRHLDE         ;Check B characters
2973 C9          15830          RET                     ;Matched - RETurn Z
                 15840 ;
                 15850 ;        FLASH & FLASHØ - Flash a message string
                 15860 ;        HL => Message string to flash
                 15870 ;
2974 CDBE29      15880 FLASHØ   CALL    RESKFLG         ;Reset Pause, Enter
                 15890 ;
                 15900 ;        Pause briefly
                 15910 ;
2977 Ø1FD41      15920 FLASH    LD      BC,16893        ;Delay count
297A             15930          @@PAUSE
297A 3E10        ØØØ72          LD      A,16
297C EF          ØØØ73          RST     4Ø
                 15940 ;
                 15950 ;        Wait for no Enter or Break
                 15960 ;
297D FD7EØA      15970          LD      A,(IY+KFLAG$)   ;P/u KFLAG$
2980 E605        15980          AND     4!1             ;Wait until no ENTER!BRK
2982 20F0        15990          JR      NZ,FLASHØ       ;Still down, go flash
2984 CDBE29      16000          CALL    RESKFLG         ;Reset
                 16010 ;
                 16020 ;        Display the message & wait for 1/4 second
                 16030 ;
2987 CDD129      16040 FLS1     CALL    DSPLY
298A Ø1ØØ4Ø      16050          LD      BC,4ØØØH        ;Delay 4ØØØ iterations
298D CD9D29      16060          CALL    FLS2            ;  & scan for Break,Enter
                 16070 ;
                 16080 ;        Erase the message & wait
                 16090 ;
2990 ØE1E        16100          LD      C,EL            ;Erase line
2992 CDCA29      16110          CALL    DISPB
2995 Ø13333      16120          LD      BC,3333H        ;Delay 3333 iterations
2998 CD9D29      16130          CALL    FLS2            ; & scan for Break,Enter
299B 18EA        16140          JR      FLS1            ;Loop if neither
                 16150 ;
                 16160 ;        FLS2 - Delay BC loops & scan for <ENTER>
                 16170 ;
299D CD1229      16180 FLS2     CALL    CKBRK           ;Check for <BREAK>
29AØ C24524      16190          JP      NZ,ABORT
                 16200 ;
                 16210 ;        Was the <ENTER> pressed ?
                 16220 ;
29A3 FDCBØA56    16230          BIT     2,(IY+KFLAG$)   ;<ENTER> hit ?
29A7 2006        16240          JR      NZ,FLS4         ;Go on ENTER down
                 16250 ;
                 16260 ;        Nothing hit - Count down
                 16270 ;
29A9 ØB          16280          DEC     BC              ;Decrement count
29AA 78          16290          LD      A,B             ;Done ?
29AB B1          16300          OR      C
29AC 20EF        16310          JR      NZ,FLS2         ;No - check again
29AE C9          16320          RET                     ;Yes - RETurn
                 16330 ;
                 16340 ;        <ENTER> hit - POP ret addr & clr type ahead
                 16350 ;
```

LBCOPYB - COPY/APPEND Common Routines

```
29AF F1       16360 FLS4     POP     AF              ;Pop return address
29BØ          16370 CLRTYPE  @@KBD                   ;Clear type ahead
29BØ 3EØ8     00074          LD      A,8
29B2 EF       00075          RST     4Ø
29B3 28FB     16380          JR      Z,CLRTYPE       ;Good - get another
29B5 B7       16390          OR      A               ;O.K. ?
29B6 C21824   16400          JP      NZ,IOERR        ;No - I/O Error
              16410 ;
              16420 ;        Erase message line
              16430 ;
29B9 ØE1E     16440          LD      C,EL            ;Erase line
29BB CDCA29   16450          CALL    DISPB           ;Fall into RESKFLG
              16460 ;
              16470 ;        RESKFLG - Reset <PAUSE> & <ENTER> bits in KFLAG
              16480 ;
29BE          16490 RESKFLG  @@FLAGS                 ;IY => Flag Table
29BE 3E65     00076          LD      A,1Ø1
29CØ EF       00077          RST     4Ø
29C1 FD7EØA   16500          LD      A,(IY+KFLAG$)   ;P/u KFLAG$
29C4 E6F9     16510          AND     ØF9H            ;Reset bits 2 & 1
29C6 FD77ØA   16520          LD      (IY+KFLAG$),A   ;Stuff in KFLAG$
29C9 C9       16530          RET                     ;RETurn
              16540 ;
              16550 ;        DISPB - Output a byte to the Video
              16560 ;
29CA          16570 DISPB    @@DSP                   ;Output byte
29CA 3EØ2     00078          LD      A,2
29CC EF       00079          RST     4Ø
29CD C8       16580          RET     Z               ;Good - RETurn
29CE C31824   16590          JP      IOERR           ;Bad - I/O error
              16600 ;
              16610 ;        DSPLY - Display line to video
              16620 ;
29D1          16630 DSPLY    @@DSPLY                 ;Display
              00080          IFEQ    ØØH,1
              00081          LD      HL,
              00082          ENDIF
29D1 3EØA     00083          LD      A,1Ø
29D3 EF       00084          RST     4Ø
29D4 C8       16640          RET     Z
29D5 C31824   16650          JP      IOERR           ;Bad - I/O Error
              16660 ;
              16670 ;        STOP - Display Transfer aborted & Abort
              16680 ;
29D8 217D2A   16690 STOP     LD      HL,STOP$        ;"Transfer Aborted"
29DB          16700          @@LOGOT                 ;Log message
              00085          IFEQ    ØØH,1
              00086          LD      HL,
              00087          ENDIF
29DB 3EØC     00088          LD      A,12
29DD EF       00089          RST     4Ø
29DE CD1629   16710          CALL    PMTSYS          ;"Insert SYSTEM disk"
29E1 C34524   16720          JP      ABORT           ;Abort
              16730 ;
              16740 ;        GETSYS2 - Bring in SYS2
              16750 ;
29E4 3E84     16760 GETSYS2  LD      A,84H
29E6 EF       16770          RST     28H
              16780 ;
```

LBCOPYB - COPY/APPEND Common Routines

```
                16790 ;       GETSYS3 - Bring in SYS3
                16800 ;
29E7 3E85       16810 GETSYS3 LD      A,85H
29E9 EF         16820         RST     28H
                16830 ;
                16840 ;       CKDEV - Is the source or Destination a device ?
                16850 ;       Z - Either Source or Destination is a device
                16860 ;
29EA 3A072C     16870 CKDEV   LD      A,(FCB1)        ;Is source a device?
29ED FE2A       16880         CP      '*'
29EF C8         16890         RET     Z
29F0 3AB32B     16900         LD      A,(FCB2)        ;Is destination a device?
29F3 FE2A       16910         CP      '*'
29F5 C9         16920         RET
                16930 ;
                16940 ;       RDGAT - Read GAT of Drive C
                16950 ;
29F6 D5         16960 RDGAT   PUSH    DE              ;Save DE & HL
29F7 E5         16970         PUSH    HL
29F8 FDE5       16980         PUSH    IY              ;Save IY
                16990 ;
                17000 ;       Set D = Cyl, E = Sector, HL => Buffer
                17010 ;
29FA            17020         @@GTDCT                 ;Point IY to DCT
29FA 3E51       00090         LD      A,81
29FC EF         00091         RST     40
29FD FD5609     17030         LD      D,(IY+9)        ;P/u dir cyl in D
2A00 1E00       17040         LD      E,0             ;GAT is sector 0
2A02 21002D     17050         LD      HL,GAT          ;HL => GAT I/O Buffer
2A05            17060         @@RDSSC                 ;Read Track D, Sector E
2A05 3E55       00092         LD      A,85
2A07 EF         00093         RST     40
                17070 ;
                17080 ;       Restore Registers
                17090 ;
2A08 FDE1       17100         POP     IY              ;Restore IY
2A0A E1         17110         POP     HL              ;Restore HL & DE
2A0B D1         17120         POP     DE
2A0C 3E14       17130         LD      A,14H           ;Else reset to GAT error
2A0E C9         17140         RET                     ;RETurn with condition
                17150 ;
                17160 ;       OPENDES - OPEN Destination File
                17170 ;
2A0F 21002F     17180 OPENDES LD      HL,BUF2         ;HL => Dest I/O Buffer
2A12 11B32B     17190         LD      DE,FCB2         ;DE => Dest FCB
2A15 1811       17200         JR      OPENFIL         ;Open the file
                17210 ;
                17220 ;       INITDES - INIT the destination file
                17230 ;
2A17 3E3A       17240 INITDES LD      A,@INIT         ;A = @INIT SVC Number
2A19 180F       17250         JR      INITFIL         ;Go into OPEN routine
                17260 ;
                17270 ;       OPENSRC - Open Source File
                17280 ;
2A1B 11072C     17290 OPENSRC LD      DE,FCB1         ;DE => Source FCB
                17300 ;
                17310 ;       Set the File OPEN inhibit flag
                17320 ;
2A1E            17330         @@FLAGS                 ;IY => System Flag Table
```

LBCOPYB - COPY/APPEND Common Routines

```
2A1E 3E65      ØØØ94          LD      A,1Ø1
2AØ EF         ØØØ95          RST     4Ø
2A21 FDCB12C6  17340          SET     Ø,(IY+SFLAG$)   ;Set file open inhibit
               17350 ;
2A25 21ØØ2E    17360 OPENSR2  LD      HL,BUF1         ;HL => Source I/O buffer
2A28 3E3B      17370 OPENFIL  LD      A,@OPEN         ;A = @OPEN SVC number
2A2A ØEØØ      17380 INITFIL  LD      B,Ø             ;B = LRL = 256
               17390 ;
               17400 ;        OPEN or INIT the File
               17410 ;
2A2C CDCA27    17420 GETFILE  CALL    DOSVC           ;OPEN or INIT the file
2A2F C8        17430          RET     Z               ;RETurn with Z set
2A3Ø C31824    17440          JP      IOERR           ;NZ - I/O Error
               17450 ;
               17460 ;        Error & Informative message strings
               17470 ;
2A33 43        17480 COPYMS   DB      'Copying: ',ETX
     6F 7Ø 79 69 6E 67 3A 2Ø
     Ø3
2A3D 41        17490 APPDMS   DB      'Appending: ',ETX
     7Ø 7Ø 65 6E 64 69 6E 67
     3A 2Ø Ø3
               17500 ;
2A49 46        17510 DIFLRL$  DB      'Files have different LRLs',CR
     69 6C 65 73 2Ø 68 61 76
     65 2Ø 64 69 66 66 65 72
     65 6E 74 2Ø 4C 52 4C 73
     ØD
2A63 44        17520 DSTREQ$  DB      'Destination spec required',CR
     65 73 74 69 6E 61 74 69
     6F 6E 2Ø 73 7Ø 65 63 2Ø
     72 65 71 75 69 72 65 64
     ØD
2A7D 1D        17530 STOP$    DB      BL,'Transfer aborted',CR
     54 72 61 6E 73 66 65 72
     2Ø 61 62 6F 72 74 65 64
     ØD
2A8F 46        17540 SPCREQ$  DB      'File spec required',CR
     69 6C 65 2Ø 73 7Ø 65 63
     2Ø 72 65 71 75 69 72 65
     64 ØD
2AA2 49        17550 NOINDO$  DB      'Invalid command during <DO> '
     6E 76 61 6C 69 64 2Ø 63
     6F 6D 6D 61 6E 64 2Ø 64
     75 72 69 6E 67 2Ø 3C 44
     4F 3E 2Ø
2ABE 7Ø        17560          DB      'processing',CR
     72 6F 63 65 73 73 69 6E
     67 ØD
2AC9 53        17570 SAMERR$  DB      'Source and destination disks'
     6F 75 72 63 65 2Ø 61 6E
     64 2Ø 64 65 73 74 69 6E
     61 74 69 6F 6E 2Ø 64 69
     73 6B 73
2AE5 2Ø        17580          DB      ' are the same',CR
     61 72 65 2Ø 74 68 65 2Ø
     73 61 6D 65 ØD
               17590 ;
2AF3 1D        17600 PMTSYS$  DB      BL,EL,' Insert SYSTEM disk <ENTER>'
```

LBCOPYB - COPY/APPEND Common Routines

```
     1E 2Ø 49 6E 73 65 72 74
     2Ø 53 59 53 54 45 4D 2Ø
     64 69 73 6B 2Ø 3C 45 4E
     54 45 52 3E
2B1Ø 1D       17161Ø          DB      BL,ETX
     Ø3
              17162Ø ;
2B12 1D       17163Ø PMTSRC$ DB       BL,EL,' Insert SOURCE disk '
     1E 2Ø 49 6E 73 65 72 74
     2Ø 53 4F 55 52 43 45 2Ø
     64 69 73 6B 2Ø
2B28 69       17164Ø          DB      'in drive :'
     6E 2Ø 64 72 69 76 65 2Ø
     3A
2B32 3Ø       17165Ø SRC_DR   DB      'Ø and press <ENTER>'
     2Ø 61 6E 64 2Ø 7Ø 72 65
     73 73 2Ø 3C 45 4E 54 45
     52 3E
2B45 1D       17166Ø          DB      BL,ETX
     Ø3
              17167Ø ;
2B47 1D       17168Ø PMTDST$ DB       BL,EL,' Insert DESTINATION disk '
     1E 2Ø 49 6E 73 65 72 74
     2Ø 44 45 53 54 49 4E 41
     54 49 4F 4E 2Ø 64 69 73
     6B 2Ø
2B62 69       17169Ø          DB      'in drive :'
     6E 2Ø 64 72 69 76 65 2Ø
     3A
2B6C 3Ø       17170Ø DEST_DR  DB      'Ø and press <ENTER>'
     2Ø 61 6E 64 2Ø 7Ø 72 65
     73 73 2Ø 3C 45 4E 54 45
     52 3E
2B7F 1D       17171Ø          DB      BL,ETX
     Ø3
              17172Ø ;
              17173Ø ;
              17174Ø ;        APPEND PARAMETER TABLE
              17175Ø ;
2B81 8Ø       17176Ø APPTBL   DB      8ØH             ;Use new @PARAM
              17177Ø ;
2B82 55       17178Ø          DB      FLAG!ABB!5
2B83 53       17179Ø          DB      'STRIP'
     54 52 49 5Ø
2B88 ØØ       17180Ø          DB      Ø
2B89 AD24     17181Ø          DW      SPARM+1
              17182Ø ;
2B8B 54       17183Ø          DB      FLAG!ABB!4
2B8C 45       17184Ø          DB      'ECHO'
     43 48 4F
2B9Ø ØØ       17185Ø          DB      Ø
2B91 2827     17186Ø          DW      EPARM+1
2B93 ØØ       17187Ø          DB      Ø
              17188Ø ;
              17189Ø ;        COPY PARAMETER TABLE
              17190Ø ;
2B94 8Ø       17191Ø COPYTBL  DB      8ØH             ;New @PARAM
              17192Ø ;
2B95 54       17193Ø          DB      FLAG!ABB!4
```

LBCOPYB - COPY/APPEND Common Routines

```
2B96 45          17940          DB      'ECHO'
     43 48 4F
2B9A ØØ          17950          DB      Ø
2B9B 2827        17960          DW      EPARM+1
                 17970 ;
2B9D 93          17980          DB      NUM!ABB!3
2B9E 4C          17990          DB      'LRL'
     52 4C
2BA1 ØØ          18ØØØ          DB      Ø
2BA2 A225        18Ø1Ø          DW      LPARM+1
                 18Ø2Ø ;
2BA4 55          18Ø3Ø          DB      FLAG!ABB!5
2BA5 43          18Ø4Ø          DB      'CLONE'
     4C 4F 4E 45
2BAA ØØ          18Ø5Ø          DB      Ø
2BAB 8Ø26        18Ø6Ø          DW      CPARM+1
                 18Ø7Ø ;
2BAD 41          18Ø8Ø          DB      FLAG!1
2BAE 58          18Ø9Ø          DB      'X'
2BAF ØØ          181ØØ          DB      Ø
2BBØ 2725        1811Ø          DW      XPARM+1
2BB2 ØØ          1812Ø          DB      Ø               ;End of COPY parm table
                 1813Ø ;
                 1814Ø ;        I/O & Storage Buffers
                 1815Ø ;
2BB3 ØØ          1816Ø FCB2     DB      Ø               ;Show closed on LOAD
ØØ1F             1817Ø          DS      31
ØØ1Ø             1818Ø TEMBUF   DS      16
ØØ1Ø             1819Ø FROM     DS      16
2BF3 2Ø          182ØØ TO       DB      ' to '
     74 6F 2Ø
ØØ1Ø             1821Ø TO2      DS      16
2CØ7 ØØ          1822Ø FCB1     DB      Ø
ØØ1F             1823Ø          DS      31
ØØ12             1824Ø SRCSTR   DS      18
ØØ12             1825Ø DSTSTR   DS      18
ØØØ7             1826Ø CLONSAV  DS      7
                 1827Ø ;
2DØØ             1828Ø          ORG     $<-8+1<+8
                 1829Ø ;
Ø1ØØ             183ØØ GAT      DS      256
Ø1ØØ             1831Ø BUF1     DS      256
Ø1ØØ             1832Ø BUF2     DS      256
                 1833Ø ;
                 ØØ26Ø ;
3ØØØ             ØØ27Ø          SUBTTL  <>
24ØØ             ØØ28Ø          END     COPY
```

@@1	ØØØØ	@@2	ØØØØ	@@3	ØØØØ		
@@4	ØØØØ	@INIT	ØØ3A	@MOD2	ØØØØ		
@MOD4	FFFF	@OPEN	ØØ3B	ABB	ØØ1Ø		
ABORT	2445	AP	ØØ27	APCODE	244F		
APND1	246E	APND2	24B9	APND2A	24CF		
APND3	24D8	APPDMS	2A3D	APPEND	24Ø3		
APPENDA	24ØC	APPFLAG	2877	APPTBL	2B81		
BL	ØØ1D	BREAK	ØØ8Ø	BS	ØØØ8		
BUF1	2EØØ	BUF2	2FØØ	BYTEIO	26D5		
BYTIOØ	26F4	BYTIO1	26F9	BYTIO4	2712		
BYTIO4A	271D	CFLAG$	ØØØ2	CHKDEST	276F		
CHKPROT	2825	CKBRK	2912	CKDEV	29EA		
CLONSAV	2C4B	CLRTYPE	29BØ	COP1	2743		
COP2	274E	COPY	24ØØ	COPY1	251D		
COPYCD	25Ø8	COPYMS	2A33	COPYST	24F5		
COPYSTA	24FE	COPYTBL	2B94	CPARM	267F		
CPRHLDE	296C	CPYFILE	2872	CR	ØØØD		
CVRT	2738	CVRTØ	2737	CVRTUC	273Ø		
DEST_DR	2B6C	DEVICE	281F	DFLAG$	ØØØ3		
DIFLRL	2431	DIFLRL$	2A49	DISPB	29CA		
DOINIT	28C2	DOSVC	27CA	DO_CKDR	2817		
DSPEC	2857	DSPLY	29D1	DSTREQ	2435		
DSTREQ$	2A63	DSTSTR	2C39	DUN	27EC		
DUN2	27FA	EL	ØØ1E	EPARM	2727		
ETX	ØØØ3	EXIT	2413	E_O_F	26FC		
FCB1	2CØ7	FCB2	2BB3	FLAG	ØØ4Ø		
FLASH	2977	FLASHØ	2974	FLS1	2987		
FLS2	299D	FLS4	29AF	FROM	2BE3		
GAT	2DØØ	GEOF1	267C	GEOF2	2697		
GEOF3	26A3	GEOF3A	269D	GEOF4	26A9		
GEOF5	28F2	GEOF6	28F6	GEOF7	29ØØ		
GETCLON	28A5	GETDRIV	284A	GETFILE	2A2C		
GETLRL	2896	GETSYS2	29E4	GETSYS3	29E7		
GOHOME	26AC	GOTEOF	264D	ILLACC	283C		
INIT1	2844	INITDES	2A17	INITFIL	2A2A		
IOERR	2418	ITSCOPY	287F	KFLAG$	ØØØA		
LF	ØØØA	LPARM	25A1	LRL2	24A3		
MVFLD	27B5	MVFLD1	27B6	MVFLD2	27B9		
MVFLD3	27C5	NOINDO	242D	NOINDO$	2AA2		
NUM	ØØ8Ø	OKYDOKY	2848	OPENDES	2AØF		
OPENFIL	2A28	OPENSR2	2A25	OPENSRC	2A1B		
OPNDST	25DF	OPNSRC	2593	PAR_ERR	ØØ2C		
PMTDST	294A	PMTDST$	2B47	PMTSRC	2928		
PMTSRC$	2B12	PMTSYS	2916	PMTSYS$	2AF3		
PRSPC	275A	PRSPC1	2782	PRSPC2	2783		
PRSPC3	2795	PRSPC4	2797	PRSPC5	2798		
PRSPC6	27A8	PRSPC7	27AA	PUTDEST	2861		
PUTSOUR	285B	RDGAT	29F6	RDREC1	2617		
RDREC2	262D	RDREC3	262F	RDREC4	2639		
RDREC5	2647	RESKFLG	29BE	RETCOD	26AF		
SAMERR	2425	SAMERR$	2AC9	SAVESP	2448		
SFLAG$	ØØ12	SPARM	24AC	SPCREQ	2429		
SPCREQ$	2A8F	SRCSTR	2C27	SRC_DR	2B32		
STOP	29D8	STOP$	2A7D	STR	ØØ2Ø		
STUFCHR	27DD	SVCNUM	2831	SYSBUF	28A1		
SYSBUF2	28AE	TAB	ØØØ9	TEMBUF	2BD3		
TLP	27DØ	TO	2BF3	TO2	2BF7		
USEHI	28D1	USEREGC	25A8	VFLAG$	ØØ15		

```
WRERN           26B5 XBUFF          2865 XF2            25FE
XFER            2538 XFER1          254D XFER2          2559
XFER3           255D XFER5          2611 XFRDRV         291B
XPARM           2526 @@ABORT        6D39 @@ADTSK        6DCC
@@BANK          72E4 @@BKSP         6FC4 @@BREAK        72FA
@@CHNIO         6D24 @@CKBRKC       7348 @@CKDRV        6E2Ø
@@CKEOF         6FD9 @@CKTSK        6DB7 @@CLOSE        6FAF
@@CLS           7332 @@CMNDI        6D63 @@CMNDR        6D78
@@CTL           6B88 @@DATE         6CFA @@DCSTAT       6E5F
@@DEBUG         6DA2 @@DECHEX       7264 @@DIRRD        71D1
@@DIRWR         71E6 @@DIV16        724F @@DIV8         723A
@@DODIR         6E35 @@DSP          6B4C @@DSPLY        6BEC
@@ERROR         6D8D @@EXIT         6D4E @@FEXT         713E
@@FLAGS         72CE @@FNAME        7153 @@FSPEC        7129
@@GATRD         71BC @@GATWR        71FB @@GET          6B6Ø
@@GTDCB         717D @@GTDCT        7168 @@GTMOD        7192
@@HDFMT         6FØ7 @@HEX16        72A3 @@HEX8         728E
@@HEXDEC        7279 @@HIGH$        72B8 @@INIT         6F85
@@KBD           6BC4 @@KEY          6B38 @@KEYIN        6BD8
@@KLTSK         6EØB @@LOAD         7ØFF @@LOC          6FEE
@@LOF           7ØØ3 @@LOGER        6C23 @@LOGOT        6C38
@@MSG           6C6F @@MUL16        7225 @@MUL8         721Ø
@@OPEN          6F9A @@PARAM        6CE5 @@PAUSE        6CDØ
@@PEOF          7Ø18 @@POSN         7Ø2D @@PRINT        6C84
@@PRT           6B9C @@PUT          6B74 @@RAMDIR       6E4A
@@RDSEC         6EDD @@RDSSC        71A7 @@READ         7Ø42
@@REMOV         6F7Ø @@RENAM        6F5B @@REW          7Ø57
@@RMTSK         6DE1 @@RPTSK        6DF6 @@RREAD        7Ø6C
@@RSLCT         6EC8 @@RSTOR        6E89 @@RUN          7114
@@RWRIT         7Ø81 @@SEEK         6EB3 @@SEEKSC       7Ø96
@@SKIP          7ØAB @@SLCT         6E74 @@STEPI        6E9E
@@TIME          6DØF @@VDCTL        6CBB @@VER          7ØCØ
@@VRSEC         6EF2 @@WEOF         7ØD5 @@WHERE        6BBØ
@@WRITE         7ØEA @@WRSEC        6F1C @@WRSSC        6F31
@@WRTRK         6F46
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

Command: CREATE

Library: SYS7/SYS

ISAM # : 13H

```
                00100 ;LBCREATE/ASM - CREATE Command
0000            00110         TITLE   <CREATE - LS-DOS 6.2>
                00120 ;
0000            00130 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
0000            00140 *GET    VALUES:3                ;Misc. equates
                03920 ;VALUES/ASM - Version 6
                03930 *LIST OFF
                04200 *LIST ON
                00150 ;
2400            00160         ORG     2400H
                00170 ;
2400            00180 CREATE  EQU     $
2400            00190         @@CKBRKC                ;Break key down?
2400 3E6A       00001         LD      A,106
2402 EF         00002         RST     40
2403 2804       00200         JR      Z,BEGINA        ;Ok if not
2405 21FFFF     00210         LD      HL,-1           ;  else abort
2408 C9         00220         RET
                00230 ;
2409 ED731E25   00240 BEGINA  LD      (SAVESP+1),SP   ;Save SP address
240D CD2824     00250         CALL    CREATCD         ;Do the CREATE code
                00260 ;
                00270 ;       Set ERN & offset in FCB = value at @INIT
                00280 ;
2410 210000     00290 ERN     LD      HL,$-$          ;P/u ERN (probably 0)
2413 22B125     00300         LD      (FCB+12),HL     ;Stuff into FCB
2416 3E00       00310 OFFSET  LD      A,$-$           ;P/u offset byte
2418 32AD25     00320         LD      (FCB+8),A       ;Stuff into FCB
                00330 ;
                00340 ;       CLOSE the file if possible
                00350 ;
241B 11A525     00360         LD      DE,FCB          ;DE => FCB
241E            00370         @@CLOSE                 ;Close file
241E 3E3C       00003         LD      A,60
2420 EF         00004         RST     40
2421 C21425     00380         JP      NZ,IOERR        ;NZ - I/O Error
                00390 ;
                00400 ;       Exit Routine
                00410 ;
2424 210000     00420         LD      HL,0            ;Successful - HL = 0
2427 C9         00430         RET                     ;Return
                00440 ;
                00450 ;       Parse the Filespec given
                00460 ;
2428 11A525     00470 CREATCD LD      DE,FCB          ;Fetch filespec
242B            00480         @@FSPEC
242B 3E4E       00005         LD      A,78
242D EF         00006         RST     40
242E C23725     00490         JP      NZ,SPCREQ       ;Quit on bad name
                00500 ;
                00510 ;       Check out parameter input
                00520 ;
2431 118525     00530         LD      DE,PRMTBL$      ;Get parms
2434            00540         @@PARAM
2434 3E11       00007         LD      A,17
2436 EF         00008         RST     40
2437 C21425     00550         JP      NZ,IOERR        ;Jump on parm error
                00560 ;
```

```
                    00570 ;       Check If Rec or LRL were specified
                    00580 ;
243A 3A9925         00590         LD      A,(LRESP)       ;P/u LRL response
243D 47             00600         LD      B,A             ;Save in B
243E 3A9225         00610         LD      A,(RRESP)       ;P/u REC response
2441 B0             00620         OR      B               ;Either specified ?
2442 2807           00630         JR      Z,RPARM         ;No - check # records
                    00640 ;
                    00650 ;       If Size parm also was specified - Parameter Error
                    00660 ;
2444 3A8B25         00670         LD      A,(SRESP)       ;Size can't be used
2447 B7             00680         OR      A               ;  with REC or LRL
2448 C21225         00690         JP      NZ,PRMERR       ;Specified ? - Error
                    00700 ;
                    00710 ;       Check Record count
                    00720 ;
244B 010000         00730 RPARM   LD      BC,$-$          ;P/u # of records
244E 78             00740         LD      A,B
244F B1             00750         OR      C
2450 200D           00760         JR      NZ,LPARM
                    00770 ;
                    00780 ;       Zero Records - Use Size instead
                    00790 ;
2452 210000         00800 SPARM   LD      HL,$-$          ;P/u size parm
2455 7C             00810         LD      A,H
2456 B5             00820         OR      L
2457 CA1225         00830         JP      Z,PRMERR        ;Err if size not entered
                    00840 ;
                    00850 ;       Multiply HL x 4 to get # of sectors
                    00860 ;
245A 29             00870         ADD     HL,HL           ;X 2
245B 29             00880         ADD     HL,HL           ;X 4
245C 224C24         00890         LD      (RPARM+1),HL    ;Pretend it's rec input
                    00900 ;
                    00910 ;       Make sure LRL input is valid
                    00920 ;
245F 010000         00930 LPARM   LD      BC,$-$          ;P/u LRL
2462 78             00940         LD      A,B             ;Test for > 256
2463 B7             00950         OR      A               ;If hi-order = 0,
2464 2808           00960         JR      Z,LP1           ;Just use lo-order
2466 3D             00970         DEC     A               ;Test hi-order = 1
2467 C21225         00980         JP      NZ,PRMERR       ;Quit if any other
246A B1             00990         OR      C               ;P/u lo-order
246B C21225         01000         JP      NZ,PRMERR       ;Lo-order must be 0
246E B1             01010 LP1     OR      C               ;Merge lo-order
                    01020 ;
                    01030 ;       Open the File with the LRL specified
                    01040 ;
246F 11A525         01050         LD      DE,FCB          ;Open the file
2472 210026         01060         LD      HL,BUFFER
2475 47             01070         LD      B,A             ;LRL = 256, or user entry
2476                01080         @@INIT
2476 3E3A           00009         LD      A,58
2478 EF             00010         RST     40
2479 C21425         01090         JP      NZ,IOERR        ;Jump on init error
                    01100 ;
                    01110 ;       Display "Creating : Filespec" String
                    01120 ;
247C 117625         01130         LD      DE,FILESP       ;DE => Filespec
247F ED4BAB25       01140         LD      BC,(FCB+6)      ;P/u drive #, DEC
2483 3AA525         01150         LD      A,(FCB)         ;P/u to test device/file
```

```
2486 CD2125   Ø1160          CALL    $FNAME
2489 C21425   Ø1170          JP      NZ,IOERR        ;Leave on error
248C 216C25   Ø1180          LD      HL,CREATE$      ;"Creating : "
248F          Ø1190          @@DSPLY                 ;Display it
              ØØØ11         IFEQ    ØØH,1
              ØØØ12          LD      HL,
              ØØØ13         ENDIF
248F 3EØA     ØØØ14          LD      A,1Ø
2491 EF       ØØØ15          RST     4Ø
2492 C21425   Ø12ØØ          JP      NZ,IOERR        ;Quit on dsply error
2495 ØEØD     Ø121Ø          LD      C,CR            ;End line
2497          Ø122Ø          @@DSP
2497 3EØ2     ØØØ16          LD      A,2
2499 EF       ØØØ17          RST     4Ø
249A C21425   Ø123Ø          JP      NZ,IOERR
              Ø124Ø ;
              Ø125Ø ;        Save ERN & offset from FCB for later
              Ø126Ø ;
249D 2AB125   Ø127Ø          LD      HL,(FCB+12)     ;P/u ERN
24AØ 221124   Ø128Ø          LD      (ERN+1),HL
24A3 3AAD25   Ø129Ø          LD      A,(FCB+8)       ;P/u offset byte
24A6 321724   Ø13ØØ          LD      (OFFSET+1),A
              Ø131Ø ;
              Ø132Ø ;        Check if the New Size > Old Size ?
              Ø133Ø ;
24A9 11A525   Ø134Ø BIGGER   LD      DE,FCB          ;DE => FCB+Ø
24AC          Ø135Ø          @@LOF                   ;Get length of file
24AC 3E4Ø     ØØØ18          LD      A,64
24AE EF       ØØØ19          RST     4Ø
24AF 6Ø       Ø136Ø          LD      H,B             ;Move len to HL
24BØ 69       Ø137Ø          LD      L,C
24B1 ED4B4C24 Ø138Ø          LD      BC,(RPARM+1)    ;P/u # of records
24B5 AF       Ø139Ø          XOR     A               ;Clear carry
24B6 E5       Ø14ØØ          PUSH    HL              ;Save ERN
24B7 ED42     Ø141Ø          SBC     HL,BC           ;Is new ERN > old ERN?
24B9 E1       Ø142Ø          POP     HL              ;HL = ERN
24BA D23B25   Ø143Ø          JP      NC,BADSIZ       ;Go if not
              Ø144Ø ;
              Ø145Ø ;        Position FCB to Ending Record Number
              Ø146Ø ;
24BD ØB       Ø147Ø          DEC     BC              ;Reduce to offset from Ø
24BE          Ø148Ø          @@POSN                  ;Position to new ERN
24BE 3E42     ØØ2Ø           LD      A,66
24CØ EF       ØØØ21          RST     4Ø
              Ø149Ø ;
              Ø15ØØ ;        Fill a 256 byte buffer with X'E5' bytes
              Ø151Ø ;
24C1 21ØØ26   Ø152Ø          LD      HL,BUFFER       ;Buffer area
24C4 11Ø126   Ø153Ø          LD      DE,BUFFER+1     ;"format" a sector
24C7 Ø1FFØØ   Ø154Ø          LD      BC,255
24CA 36E5     Ø155Ø          LD      (HL),ØE5H
24CC EDBØ     Ø156Ø          LDIR
              Ø157Ø ;
              Ø158Ø ;        Write the last Record of the file
              Ø159Ø ;
24CE 11A525   Ø16ØØ          LD      DE,FCB          ;Write the new ERN sector
24D1 25       Ø161Ø          DEC     H               ;Set HL = buffer start
24D2 23       Ø162Ø          INC     HL
24D3 CDØD25   Ø163Ø          CALL    WRITE           ;Write the last record
24D6          Ø164Ø          @@REW                   ;Rewind File
24D6 3E44     ØØ22           LD      A,68
```

```
24D8 EF          ØØØ23          RST     4Ø
                 Ø165Ø ;
                 Ø166Ø ;        Read in the directory entry
                 Ø167Ø ;
24D9 ED4BAB25    Ø168Ø          LD      BC,(FCB+6)      ;Get drive # & DEC
24DD             Ø169Ø          @@DIRRD                 ;Read in record
24DD 3E57        ØØØ24          LD      A,87
24DF EF          ØØØ25          RST     4Ø
24EØ 2Ø32        Ø17ØØ          JR      NZ,IOERR        ;Jump on read error
                 Ø171Ø ;
                 Ø172Ø ;        Set the CREATE bit, and write it back out
                 Ø173Ø ;
24E2 23          Ø174Ø          INC     HL              ;Point to FCB+1 &
24E3 CBFE        Ø175Ø          SET     7,(HL)          ;  set the CREATE bit
24E5             Ø176Ø          @@DIRWR                 ;Write entry back
24E5 3E58        ØØØ26          LD      A,88
24E7 EF          ØØØ27          RST     4Ø
24E8 2Ø2A        Ø177Ø          JR      NZ,IOERR        ;Jump on write error
                 Ø178Ø ;
                 Ø179Ø ;        Do we have to Fill the file ?
                 Ø18ØØ ;
24EA Ø1ØØØ1      Ø181Ø FILL     LD      BC,Ø1ØØH        ;P/u FILL parm
24ED Ø5          Ø182Ø          DEC     B
24EE C8          Ø183Ø          RET     Z               ;RETurn if no Fill
                 Ø184Ø ;
                 Ø185Ø ;        Create a Buffer with the FILL bytes
                 Ø186Ø ;
24EF D5          Ø187Ø          PUSH    DE              ;Save FCB pointer
24FØ 21ØØ26      Ø188Ø          LD      HL,BUFFER       ;I/O buffer
24F3 71          Ø189Ø          LD      (HL),C          ;Byte to xfer
24F4 11Ø126      Ø19ØØ          LD      DE,BUFFER+1
24F7 Ø1FFØ1      Ø191Ø          LD      BC,255+256      ;Hit both buffers
24FA EDBØ        Ø192Ø          LDIR                    ;Xfer into buffer
                 Ø193Ø ;
                 Ø194Ø ;        Pt HL => User Buff, DE => FCB, BC = last Rec
                 Ø195Ø ;
24FC D1          Ø196Ø          POP     DE              ;Restore FCB pointer
24FD ED4B4C24    Ø197Ø          LD      BC,(RPARM+1)    ;P/u last record
25Ø1 21ØØ27      Ø198Ø          LD      HL,UBUFF        ;User Buffer
                 Ø199Ø ;
                 Ø2ØØØ ;        Loop to write logical records
                 Ø2Ø1Ø ;
25Ø4 78          Ø2Ø2Ø WRLOOP   LD      A,B             ;Is rec cnt = Ø ?
25Ø5 B1          Ø2Ø3Ø          OR      C
25Ø6 C8          Ø2Ø4Ø          RET     Z               ;Yes - done
25Ø7 CDØD25      Ø2Ø5Ø          CALL    WRITE           ;Write Record
25ØA ØB          Ø2Ø6Ø          DEC     BC              ;Dec one
25ØB 18F7        Ø2Ø7Ø          JR      WRLOOP          ;Do til BC = Ø
                 Ø2Ø8Ø ;
                 Ø2Ø9Ø ;        Write the buffer contents
                 Ø21ØØ ;
25ØD             Ø211Ø WRITE    @@WRITE                 ;Write buffer
25ØD 3E4B        ØØØ28          LD      A,75
25ØF EF          ØØØ29          RST     4Ø
2510 C8          Ø212Ø          RET     Z               ;Good - RETurn
2511 21          Ø213Ø          DB      21H             ;Skip LD A,## instruction
2512 3E2C        Ø214Ø PRMERR   LD      A,PAR_ERR       ;Parameter Error
                 Ø215Ø ;
                 Ø216Ø ;        I/O error display & abort routine
                 Ø217Ø ;
2514 6F          Ø218Ø IOERR    LD      L,A             ;Save error # in HL
```

```
2515 2600      02190          LD      H,0
2517 F6C0      02200          OR      0C0H            ;Short error message
2519 4F        02210          LD      C,A             ;Stuff in C for @ERROR
251A           02220          @@ERROR                 ;Display error message
251A 3E1A      00030          LD      A,26
251C EF        00031          RST     40
251D 310000    02230 SAVESP   LD      SP,$-$          ;P/u original SP
2520 C9        02240          RET                     ;Done, RETurn
               02250 ;
               02260 ;        Routine to pick up device/file name
               02270 ;
2521 CB7F      02280 $FNAME   BIT     7,A             ;Test device/file
2523 2804      02290          JR      Z,FNAME1        ;Go if device
2525           02300          @@FNAME
2525 3E50      00032          LD      A,80
2527 EF        00033          RST     40
2528 C9        02310          RET
2529 3E2A      02320 FNAME1   LD      A,'*'           ;Stuff device indicator
252B 12        02330          LD      (DE),A
252C 13        02340          INC     DE
252D 79        02350          LD      A,C             ;Stuff 1st character
252E 12        02360          LD      (DE),A
252F 13        02370          INC     DE
2530 78        02380          LD      A,B             ;Stuff 2nd character
2531 12        02390          LD      (DE),A
2532 13        02400          INC     DE
2533 3E03      02410          LD      A,3             ;Stuff ETX
2535 12        02420          LD      (DE),A
2536 C9        02430          RET
               02440 ;
               02450 ;        Error Message Display routine
               02460 ;
2537 214625    02470 SPCREQ   LD      HL,SPCREQ$
253A DD        02480          DB      0DDH
253B 215925    02490 BADSIZ   LD      HL,BADSIZ$
               02500 ;
               02510 ;        Log Error Message & Abort
               02520 ;
253E           02530          @@LOGOT                 ;Log error message
               00034          IFEQ    00H,1
               00035          LD      HL,
               00036          ENDIF
253E 3E0C      00037          LD      A,12
2540 EF        00038          RST     40
2541 21FFFF    02540          LD      HL,-1           ;Set abort code
2544 18D7      02550          JR      SAVESP          ;Exit
               02560 ;
               02570 ;        Messages
               02580 ;
2546 46        02590 SPCREQ$ DB       'File spec required',CR
     69 6C 65 20 73 70 65 63
     20 72 65 71 75 69 72 65
     64 0D
2559 46        02600 BADSIZ$ DB       'File exists larger',CR
     69 6C 65 20 65 78 69 73
     74 73 20 6C 61 72 67 65
     72 0D
256C 43        02610 CREATE$ DB       'Creating: '
     72 65 61 74 69 6E 67 3A
     20
000F           02620 FILESP  DS       15
```

```
                    02630 ;
                    02640 ;PARAMETER TABLE
                    02650 ;
2585 80             02660 PRMTBL$ DB      80H             ;6.x Parameter Table
                    02670 ;
                    02680 ;       SIZE (S) - Accept Numeric Input only
                    02690 ;
2586 94             02700         DB      NUM!ABB!4
2587 53             02710         DB      'SIZE'
     49 5A 45
258B 00             02720 SRESP   DB      0
258C 5324           02730         DW      SPARM+1
                    02740 ;
                    02750 ;       REC (R) - Accept Numeric input only
                    02760 ;
258E 93             02770         DB      NUM!ABB!3
258F 52             02780         DB      'REC'
     45 43
2592 00             02790 RRESP   DB      0
2593 4C24           02800         DW      RPARM+1
                    02810 ;
                    02820 ;       LRL (L) - Accept Numeric input only
                    02830 ;
2595 93             02840         DB      NUM!ABB!3
2596 4C             02850         DB      'LRL'
     52 4C
2599 00             02860 LRESP   DB      0
259A 6024           02870         DW      LPARM+1
                    02880 ;
                    02890 ;       FILL (F) - Accept Numeric or Flag input
                    02900 ;
259C D4             02910         DB      FLAG!NUM!ABB!4
259D 46             02920         DB      'FILL'
     49 4C 4C
25A1 00             02930 FRESP   DB      0
25A2 EB24           02940         DW      FILL+1
25A4 00             02950         DB      0
                    02960 ;
                    02970 ;       I/O buffer
                    02980 ;
25A5 00             02990 FCB     DB      0
001F                03000         DS      31
2600                03010         ORG     $<-8+1<8
0100                03020 BUFFER  DS      256
0100                03030 UBUFF   DS      256
                    03040 ;
2400                03050         END     CREATE
```

```
$FNAME          2521 @@1        0000 @@2        0000
@@3             0000 @@4        0000 @MOD2      0000
@MOD4           FFFF ABB        0010 AP         0027
BADSIZ          253B BADSIZ$    2559 BEGINA     2409
BIGGER          24A9 BREAK      0080 BS         0008
BUFFER          2600 CFLAG$     0002 CR         000D
CREATCD         2428 CREATE     2400 CREATE$    256C
DFLAG$          0003 ERN        2410 ETX        0003
FCB             25A5 FILESP     2576 FILL       24EA
FLAG            0040 FNAME1     2529 FRESP      25A1
IOERR           2514 KFLAG$     000A LF         000A
LP1             246E LPARM      245F LRESP      2599
NUM             0080 OFFSET     2416 PAR_ERR    002C
PRMERR          2512 PRMTBL$    2585 RPARM      244B
RRESP           2592 SAVESP     251D SFLAG$     0012
SPARM           2452 SPCREQ     2537 SPCREQ$    2546
SRESP           258B STR        0020 TAB        0009
UBUFF           2700 VFLAG$     0015 WRITE      250D
WRLOOP          2504 @@ABORT    8400 @@ADTSK    8493
@@BANK          89AB @@BKSP     868B @@BREAK    89C1
@@CHNIO         83EB @@CKBRKC   8A0F @@CKDRV     84E7
@@CKEOF         86A0 @@CKTSK    847E @@CLOSE    8676
@@CLS           89F9 @@CMNDI    842A @@CMNDR    843F
@@CTL           824F @@DATE     83C1 @@DCSTAT   8526
@@DEBUG         8469 @@DECHEX   892B @@DIRRD    8898
@@DIRWR         88AD @@DIV16    8916 @@DIV8     8901
@@DODIR         84FC @@DSP      8213 @@DSPLY    82B3
@@ERROR         8454 @@EXIT     8415 @@FEXT     8805
@@FLAGS         8995 @@FNAME    881A @@FSPEC    87F0
@@GATRD         8883 @@GATWR    88C2 @@GET      8227
@@GTDCB         8844 @@GTDCT    882F @@GTMOD    8859
@@HDFMT         85CE @@HEX16    896A @@HEX8     8955
@@HEXDEC        8940 @@HIGH$    897F @@INIT     864C
@@KBD           828B @@KEY      81FF @@KEYIN    829F
@@KLTSK         84D2 @@LOAD     87C6 @@LOC      86B5
@@LOF           86CA @@LOGER    82EA @@LOGOT    82FF
@@MSG           8336 @@MUL16    88EC @@MUL8     88D7
@@OPEN          8661 @@PARAM    83AC @@PAUSE    8397
@@PEOF          86DF @@POSN     86F4 @@PRINT    834B
@@PRT           8263 @@PUT      823B @@RAMDIR   8511
@@RDSEC         85A4 @@RDSSC    886E @@READ     8709
@@REMOV         8637 @@RENAM    8622 @@REW      871E
@@RMTSK         84A8 @@RPTSK    84BD @@RREAD    8733
@@RSLCT         858F @@RSTOR    8550 @@RUN      87DB
@@RWRIT         8748 @@SEEK     857A @@SEEKSC   875D
@@SKIP          8772 @@SLCT     853B @@STEPI    8565
@@TIME          83D6 @@VDCTL    8382 @@VER      8787
@@VRSEC         85B9 @@WEOF     879C @@WHERE    8277
@@WRITE         87B1 @@WRSEC    85E3 @@WRSSC    85F8
@@WRTRK         860D
2400 is the transfer address
00000 Total errors
```

NOTES:

NOTES:

Command: DATE, TIME

Library: SYS7/SYS

ISAM # : 15H, 16H

```
                00100 ;LBDATE/ASM - Date/Time Commands
0000            00110         TITLE   <DATE/TIME - LS-DOS 6.2>
                00120 ;
0000            00130 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00140 *LIST OFF                       ;Get LDOS60/EQU
                00160 *LIST ON
                00170 ;
000D            00180 CR      EQU     13
001D            00190 @ADTSK  EQU     29              ;Add Task SVC #
001E            00200 @RMTSK  EQU     30              ;Remove Task SVC #
0005            00210 CLK_SLT EQU     5               ;Clock Task Slot #
                00220 ;
                00230 ;
                00240 ;       DATE$ Storage
                00250 ;
                00260 ; DATE$+0   Year (80-87)
                00270 ; DATE$+1   Day of the month (1-31)
                00280 ; DATE$+2   Month (1-12)
                00290 ; DATE$+3   Bits 0-7 of Day of Year
                00300 ; Date$+4   Bit 0 = bit 8 of Day of Year
                00310 ;           Bits 1-3 contain Day of Week
                00320 ;           Bit 7 set if a leap year
                00330 ;
                00340 ;       TIME$ Storage
                00350 ;
                00360 ; TIME$+0   Seconds (0-59)
                00370 ; TIME$+1   Minutes (0-59)
                00380 ; TIME$+2   Hours (0-23)
                00390 ;
                00400 ;
                00410 ;
2400            00420         ORG     2400H
                00430 ;
                00440 ;       Branch to TIME entry point
                00450 ;
2400 C30A25     00460         JP      TIME            ;Time entry point
                00470 ;
2403            00480         SUBTTL  '<LBDATE - DATE Code>'
                00490 ;
2403            00500 DATE    @@CKBRKC                ;Break key down?
2403 3E6A       00001         LD      A,106
2405 EF         00002         RST     40
2406 2804       00510         JR      Z,BEGINA        ;Ok if not
2408 21FFFF     00520         LD      HL,-1           ;  else abort
240B C9         00530         RET
                00540 ;
                00550 ;       DATE - Pick up DATE$+0 pointer & stuff in IY
                00560 ;
240C E5         00570 BEGINA  PUSH    HL              ;Save command ptr
240D 218E26     00580         LD      HL,DUMBUF       ;HL => Dummy Buffer
2410            00590         @@DATE                  ;DE <= DATE$+0
2410 3E12       00003         LD      A,18
2412 EF         00004         RST     40
2413 E1         00600         POP     HL              ;Recover command ptr
2414 D5         00610         PUSH    DE              ;Xfer ptr to IY
2415 FDE1       00620         POP     IY
                00630 ;
                00640 ;       Was a Date entered on the command line ?
```

LBDATE - DATE Code

```
                  00650 ;
2417 7E           00660           LD      A,(HL)          ;P/u character
2418 FE0E         00670           CP      CR+1            ;Date entry ?
241A DA5B24       00680           JP      C,DSPDATE       ;No - display date
                  00690 ;
                  00700 ;       Date entered - check if legal format
                  00710 ;
241D 0E30         00720           LD      C,'0'           ;Init separator
241F CDA025       00730           CALL    PARSDAT         ;Parse entry
2422 C20325       00740           JP      NZ,BADFMT       ;Bad entry - abort
                  00750 ;
                  00760 ;       Legal Date - If Intl date - swap DTBUF+1 & 2
                  00770 ;
                  00780           IF      @INTL
                  00790           INC     DE              ;DE => DTBUF+1
                  00800           LD      H,D             ;HL => DTBUF+1
                  00810           LD      L,E
                  00820           LD      C,(HL)          ;C = Intl MONTH
                  00830           INC     HL              ;HL => DTBUF+2
                  00840           LD      A,(HL)          ;A = Intl DAY
                  00850           LD      (DE),A          ;Set DTBUF+1 = DAY
                  00860           LD      (HL),C          ;Set DTBUF+2 = MONTH
                  00870           DEC     DE              ;DE => DTBUF+0
                  00880           ELSE
2425 00           00890           DC      9,0             ;Pad US ver to match Intl
     00 00 00 00 00 00 00 00
                  00900           ENDIF
                  00910 ;
                  00920 ;       Is the year legal ?
                  00930 ;
242E 1A           00940           LD      A,(DE)          ;P/u year entry
242F D650         00950           SUB     80              ;Accept only 80-87
2431 DA0325       00960           JP      C,BADFMT        ;Less than 80 - bad
2434 FE08         00970           CP      8               ;Greater than 87 ?
2436 3018         00980           JR      NC,BADFMT2      ;Yes - bad
                  00990 ;
                  01000 ;       If Year is 1980 or 84 then set FEB = 29 days
                  01010 ;
2438 E603         01020           AND     3               ;0 or 4 ?
243A 214D26       01030           LD      HL,MAXDAYS+2    ;Set Feb to have 29 days
243D 2001         01040           JR      NZ,NOTLEAP      ;No - don't inc
243F 34           01050           INC     (HL)            ;Leap year - inc max days
                  01060 ;
                  01070 ;       Check Range of month - must be 1 - 12
                  01080 ;
2440 3A8D26       01090 NOTLEAP   LD      A,(DTBUF+2)     ;P/u month
2443 3D           01100           DEC     A               ;Set month = 0-11
2444 FE0C         01110           CP      12              ;Valid Month ?
2446 3008         01120           JR      NC,BADFMT2      ;Abort if 0 or >12
                  01130 ;
                  01140 ;       Valid month - point HL to max days/month
                  01150 ;
2448 2B           01160           DEC     HL              ;HL => Max day table
2449 85           01170           ADD     A,L             ;Add month # to start
244A 6F           01180           LD      L,A             ;HL => Max days for month
                  01190 ;
                  01200 ;       Check for Day entry is valid
                  01210 ;
244B 3A8C26       01220           LD      A,(DTBUF+1)     ;P/u day entry
```

LBDATE - DATE Code

```
244E 3D       01230          DEC     A               ;Reduce for test (0->FF)
244F BE       01240          CP      (HL)            ;More than max days ?
2450 D20325   01250 BADFMT2  JP      NC,BADFMT       ;Go if too large (or 0)
              01260 ;
              01270 ;        Transfer Date into buffer
              01280 ;
2453 EB       01290          EX      DE,HL           ;Point HL to DTBUF
2454 FDE5     01300          PUSH    IY              ;Point DE = DATE$
2456 D1       01310          POP     DE
2457 0E03     01320          LD      C,3             ;BC = 3 chars to xfer
2459 EDB0     01330          LDIR                    ;Xfer 3 chars
              01340 ;
              01350 ;        Display "No Date in System" if illegal Date
              01360 ;
245B FD7E02   01370 DSPDATE  LD      A,(IY+2)        ;P/u month
245E 215826   01380          LD      HL,NODATE$      ;"No Date in system"
2461 B7       01390          OR      A               ;Better not be zero
2462 CA0625   01400          JP      Z,LOGABRT       ;Log & abort
2465 47       01410 GOTDATE  LD      B,A             ;Xfer month to B
2466 214D26   01420          LD      HL,MAXDAYS+2    ;Adjust February if
2469 7E       01430          LD      A,(HL)          ;  year is leap year
246A D61D     01440          SUB     29              ;  & not already adjusted
246C 2808     01450          JR      Z,PUDAY         ;  in parsing date entry
              01460 ;
              01470 ;        Pick up Year & increment max days if leap yr
              01480 ;
246E FD7E00   01490          LD      A,(IY)          ;P/u year
2471 E603     01500          AND     3               ;1980 and 1984 are lp yrs
2473 2001     01510          JR      NZ,PUDAY        ;Not leap year - fine
2475 34       01520          INC     (HL)            ;Bump to 29
              01530 ;
              01540 ;        Set HL = day # this month, DE => Max table
              01550 ;
2476 FD6E01   01560 PUDAY    LD      L,(IY+1)        ;P/u day # this month
2479 2600     01570          LD      H,0             ;  in HL
247B 114B26   01580          LD      DE,MAXDAYS      ;DE => Max day table
              01590 ;
              01600 ;        Loop to Count up total # of days up to now
              01610 ;
247E 1A       01620 DAYLP    LD      A,(DE)          ;P/u max day
247F 85       01630          ADD     A,L             ;Add to HL
2480 6F       01640          LD      L,A
2481 8C       01650          ADC     A,H
2482 95       01660          SUB     L
2483 67       01670          LD      H,A
2484 13       01680          INC     DE              ;Bump days ptr
2485 10F7     01690          DJNZ    DAYLP           ;B months of max days
              01700 ;
              01710 ;        Stuff days (9 bits) into DATE$
              01720 ;
2487 FD7503   01730          LD      (IY+3),L        ;Stuff in lsb
248A 7C       01740          LD      A,H             ;Get bit "8"
248B FDB604   01750          OR      (IY+4)          ;  and OR it in
248E FD7704   01760          LD      (IY+4),A        ;Then put it back
              01770 ;
              01780 ;        Pick up year in E (0-7)
              01790 ;
2491 FD7E00   01800          LD      A,(IY)          ;P/u year
2494 D650     01810          SUB     80              ;Offset from 80
```

LBDATE - DATE Code

```
2496 5F       Ø1820          LD      E,A             ;Put Ø-7 in E
2497 C6Ø3     Ø1830          ADD     A,3             ;Ck for year >= 84
2499 ØF       Ø1840          RRCA
249A ØF       Ø1850          RRCA
249B E6Ø3     Ø1860          AND     3               ;Keep bits Ø,1
249D 83       Ø1870          ADD     A,E             ;Add back to year
249E 5F       Ø1880          LD      E,A             ;  & save in DE
249F 16ØØ     Ø1890          LD      D,Ø
24A1 19       Ø19ØØ          ADD     HL,DE           ;Add to days in year
24A2 23       Ø1910          INC     HL              ;To start in right place
              Ø1920 ;
              Ø1930 ;        HL = desired number to divide by seven
              Ø1940 ;
24A3 Ø1Ø7ØØ   Ø1950          LD      BC,7            ;Now divide by 7
24A6 AF       Ø1960          XOR     A
24A7 ED42     Ø1970 DIV7     SBC     HL,BC           ;Subtract weeks (7-days)
24A9 3ØFC     Ø1980          JR      NC,DIV7         ;  until under flow
              Ø1990 ;
              Ø2ØØØ ;        Correct # for division, & put in bits 1-3
              Ø2Ø1Ø ;
24AB 7D       Ø2Ø2Ø          LD      A,L
24AC C6Ø8     Ø2Ø3Ø          ADD     A,8             ;Add back to get 1-7
24AE 47       Ø2Ø4Ø          LD      B,A             ;Save day of week
24AF Ø7       Ø2Ø5Ø          RLCA                    ;Shift to bits 1-3
24BØ 4F       Ø2Ø6Ø          LD      C,A             ;  to store in DATE$
              Ø2Ø7Ø ;
              Ø2Ø8Ø ;        Merge day of week with bit 9 of day of year
              Ø2Ø9Ø ;
24B1 FD7EØ4   Ø21ØØ          LD      A,(IY+4)        ;P/u DATE$ + 4
24B4 E6F1     Ø211Ø          AND     ØF1H            ;Keep lp yr bit & bit Ø
24B6 B1       Ø212Ø          OR      C               ;Merge day of week
24B7 FD77Ø4   Ø213Ø          LD      (IY+4),A        ;Stuff back in
              Ø214Ø ;
              Ø215Ø ;        Transfer Day string into display buffer
              Ø216Ø ;
24BA 21ØØ26   Ø217Ø          LD      HL,DAYTBL       ;HL => Day string table
24BD 113926   Ø218Ø          LD      DE,DATEBUF      ;Date display buffer
24CØ D5       Ø219Ø          PUSH    DE              ;Save start
24C1 CD9125   Ø22ØØ          CALL    DSPMDY          ;Write out the day
              Ø221Ø ;
              Ø222Ø ;        Position DE to month destination in buffer
              Ø223Ø ;
24C4 13       Ø224Ø          INC     DE              ;Bump
24C5 13       Ø225Ø          INC     DE
              Ø226Ø ;
              Ø227Ø ;        Pick up month, & stuff string into buffer
              Ø228Ø ;
24C6 FD7EØ2   Ø229Ø          LD      A,(IY+2)        ;P/u month number
24C9 47       Ø23ØØ          LD      B,A             ;Stuff in B
24CA 211526   Ø231Ø          LD      HL,MONTBL       ;HL => Month string table
24CD CD9125   Ø232Ø          CALL    DSPMDY          ;Write out the month name
              Ø233Ø ;
              Ø234Ø ;        P/u day of the month & convert to ASCII
              Ø235Ø ;
24DØ 13       Ø236Ø          INC     DE              ;DE => Day destination
24D1 FD7EØ1   Ø237Ø          LD      A,(IY+1)        ;P/u day
24D4 Ø6FF     Ø238Ø          LD      B,-1            ;Init # of tens to -1
              Ø239Ø ;
              Ø24ØØ ;        Divide day of the month by 1Ø
```

LBDATE - DATE Code

```
                02410 ;
24D6 04         02420 DIV10     INC     B               ;Divide by 10
24D7 D60A       02430           SUB     10              ;  with quotient in B
24D9 30FB       02440           JR      NC,DIV10        ;Subtract until carry
                02450 ;
                02460 ;         Convert to tens digit to ASCII
                02470 ;
24DB F5         02480           PUSH    AF              ;Save (10-remainder)
24DC 78         02490           LD      A,B             ;P/u quotient
24DD C630       02500           ADD     A,'0'           ;Change to ASCII
                02510 ;
                02520 ;         Change to a space if it's a leading zero
                02530 ;
24DF FE30       02540           CP      '0'             ;Zero?
24E1 2002       02550           JR      NZ,NOTLD0       ;No - use it
24E3 3E20       02560           LD      A,' '           ;Change leading 0 to ' '
24E5 12         02570 NOTLD0    LD      (DE),A          ;Stuff in buffer
                02580 ;
                02590 ;         Convert remainder to ASCII & stuff in buffer
                02600 ;
24E6 13         02610           INC     DE              ;DE => ones destination
24E7 F1         02620           POP     AF              ;Get back remainder
24E8 C63A       02630           ADD     A,3AH           ;Change to ASCII
24EA 12         02640           LD      (DE),A          ;Stuff in buffer
                02650 ;
                02660 ;         P/u year & stuff lower digit + "0" in buffer
                02670 ;
24EB FD7E00     02680           LD      A,(IY)          ;Form last year digit
24EE D650       02690           SUB     80              ;A = 0-7
24F0 C630       02700           ADD     A,'0'           ;Convert to ASCII
24F2 324926     02710           LD      (DATEBUF+16),A  ;Stuff year
                02720 ;
                02730 ;         Set B = 0 (Normal Exit)
                02740 ;
24F5 0600       02750 LOGDT     LD      B,0             ;B = 0 (normal exit)
24F7 E1         02760           POP     HL              ;HL => Date/Time string
                02770 ;
                02780 ;         Display Date or Time String
                02790 ;
24F8 C5         02800 LOGMSG    PUSH    BC              ;Save Error #, B (exit)
24F9            02810           @@LOGOT                 ;Log message
                00005           IFEQ    00H,1
                00006           LD      HL,
                00007           ENDIF
24F9 3E0C       00008           LD      A,12
24FB EF         00009           RST     40
24FC C1         02820           POP     BC              ;B = exit condition
                02830 ;
                02840 ;         If B = 0 then exit HL = 0, otherwise HL = -1
                02850 ;
24FD 60         02860           LD      H,B             ;Set HL = -1 or 0
24FE 68         02870           LD      L,B             ;
24FF            02880           @@CKBRKC                ;Clear any break
24FF 3E6A       00010           LD      A,106
2501 EF         00011           RST     40
2502 C9         02890           RET                     ;RETurn with condition
                02900 ;
                02910 ;         Bad Format - display error & abort
                02920 ;
```

LBDATE - DATE Code

```
25Ø3 216B26   Ø293Ø BADFMT  LD      HL,BADDAT$      ;Illegal Date/Time
25Ø6 Ø6FF     Ø294Ø LOGABRT LD      B,-1            ;Abort Condition
25Ø8 18EE     Ø295Ø         JR      LOGMSG          ;Log Message
              Ø296Ø ;
25ØA          Ø297Ø         SUBTTL  '<LBDATE - TIME code>'
```

LBDATE - TIME code

```
250A            02980           PAGE
                02990 ;
250A            03000 TIME      @@CKBRKC                ;Break key down?
250A 3E6A       00012           LD      A,106
250C EF         00013           RST     40
250D 2804       03010           JR      Z,BEGINB        ;Ok if not
250F 21FFFF     03020           LD      HL,-1           ;  else abort
2512 C9         03030           RET
                03040 ;
                03050 ;         TIME entry point - Any Parms entered ?
                03060 ;
2513 E5         03070 BEGINB    PUSH    HL              ;Save pointer
2514 7E         03080 TLOOP     LD      A,(HL)          ;P/u character
2515 FE28       03090           CP      '('             ;Any parameters ?
2517 2807       03100           JR      Z,GETPRMS       ;Yes - get 'em
2519 FE0D       03110           CP      CR              ;End of line ?
251B 2809       03120           JR      Z,CLRSTK        ;Yes - go check time
251D 23         03130           INC     HL              ;Bump ptr
251E 18F4       03140           JR      TLOOP           ;Do til terminator
                03150 ;
                03160 ;         Process any Parameters
                03170 ;
2520 11D925     03180 GETPRMS   LD      DE,PRMTBL$      ;DE => Parameter Table
2523            03190           @@PARAM                 ;Get parameters
2523 3E11       00014           LD      A,17
2525 EF         00015           RST     40
                03200 ;
                03210 ;         Stuff "Illegal Time" Mess in error routine
                03220 ;
2526 217B26     03230 CLRSTK    LD      HL,BADTIM$      ;Chg "Bad date format"
2529 220425     03240           LD      (BADFMT+1),HL   ;  to "Bad time...
252C E1         03250           POP     HL              ;Recover command ptr
252D 280A       03260           JR      Z,GDPARMS       ;Z - ok to continue
                03270 ;
                03280 ;         Parameter Error - Display & abort
                03290 ;
252F 6F         03300 IOERR     LD      L,A             ;Xfer errcod to HL
2530 2600       03310           LD      H,0
2532 F6C0       03320           OR      0C0H            ;Short error
2534 4F         03330           LD      C,A             ;Xfer to C
2535            03340           @@ERROR                 ;Log error
2535 3E1A       00016           LD      A,26
2537 EF         00017           RST     40
2538 C9         03350           RET
                03360 ;
                03370 ;         Was there a TIME string entered ?
                03380 ;
2539 7E         03390 GDPARMS   LD      A,(HL)          ;P/u char
253A FE28       03400           CP      '('             ;Parms only?
253C CA8425     03410           JP      Z,DSPTIME       ;Display old time
253F FE0D       03420           CP      CR              ;End of line?
2541 CA8425     03430           JP      Z,DSPTIME       ;Display old time
                03440 ;
                03450 ;         Requested time set - Check if legal format
                03460 ;
2544 0E30       03470           LD      C,'0'           ;Init separator
2546 CDA025     03480           CALL    PARSDAT         ;Parse entry
2549 20B8       03490           JR      NZ,BADFMT       ;Bad - abort
                03500 ;
                03510 ;         Legal Format - Check if Hours are legal
```

LBDATE - TIME code

```
                    03520 ;
254B 218D26         03530           LD      HL,DTBUF+2      ;HL => Hours byte
254E 3E17           03540           LD      A,23            ;Greater than 23 ?
2550 BE             03550           CP      (HL)
2551 38B0           03560           JR      C,BADFMT        ;Yes - bad format
                    03570 ;
                    03580 ;         Hours legal - Check if minutes legal
                    03590 ;
2553 2B             03600           DEC     HL              ;HL => Minutes
2554 3E3B           03610           LD      A,59            ;Greater than 59 ?
2556 BE             03620           CP      (HL)
2557 38AA           03630           JR      C,BADFMT        ;Yes - bad format
                    03640 ;
                    03650 ;         Minutes legal - Check if seconds legal
                    03660 ;
2559 2B             03670           DEC     HL              ;HL => Seconds
255A BE             03680           CP      (HL)            ;Greater than 59 ?
255B 38A6           03690           JR      C,BADFMT        ;Yes - bad format
                    03700 ;
                    03710 ;         Legal input - transfer to TIME$ storage area
                    03720 ;
255D E5             03730           PUSH    HL              ;Save TIME buffer ptr
255E 218E26         03740           LD      HL,DUMBUF       ;HL => dummy buffer
2561                03750           @@TIME                  ;DE <= TIME$+0
2561 3E13           00018           LD      A,19
2563 EF             00019           RST     40
2564 E1             03760           POP     HL              ;Recover TIME buffer ptr
2565 010300         03770           LD      BC,3            ;3 bytes to xfer
2568 EDB0           03780           LDIR                    ;Xfer
                    03790 ;
                    03800 ;         Was the CLOCK (C) parameter entered ?
                    03810 ;
256A 210000         03820 DOCLOCK   LD      HL,0            ;HL = 0 (Normal Exit)
256D 3AE025         03830           LD      A,(CRESP)       ;P/u response
2570 B7             03840           OR      A
2571 C8             03850           RET     Z               ;RETurn if no response
                    03860 ;
                    03870 ;         CLOCK (C) parameter entered - ON or OFF ?
                    03880 ;
2572 110000         03890 CLOCK     LD      DE,$-$          ;P/u parm = FFFF or 0000
2575                03900           @@FLAGS                 ;IY => System Flags
2575 3E65           00020           LD      A,101
2577 EF             00021           RST     40
                    03910 ;
                    03920 ;         Just Set/Reset CLOCK bit if Model IV version
                    03930 ;
                    03940           IF      @MOD4
2578 FDCB15E6       03950           SET     4,(IY+'V'-'A')  ;Set Clock bit
257C 1C             03960           INC     E               ;Return if CLOCK = YES
257D C8             03970           RET     Z               ;
257E FDCB15A6       03980           RES     4,(IY+'V'-'A')  ;Otherwise Reset bit
2582 1D             03990           DEC     E               ;Set Z flag
2583 C9             04000           RET                     ;Done - RETurn
                    04010           ENDIF
                    04020 ;
                    04030 ;         Also Add or Remove Task if Model II Version
                    04040 ;
                    04050           IF      @MOD2
                    04060           RES     4,(IY+'V'-'A')  ;Reset clock bit
```

LBDATE - TIME code

```
                    04070         LD      C,CLK_SLT       ;Set C = Clock Slot #
                    04080         LD      A,@RMTSK        ;A = Remove Task SVC #
                    04090         INC     E               ;Clock Off ?
                    04100         JR      NZ,CLOFF        ;Yes - remove task
                    04110 CLON    LD      DE,DO_CLOCK     ;Clock on - DE => Address
                    04120         SET     4,(IY+'V'-'A')  ;Set clock bit
                    04130         LD      A,@ADTSK        ;A = Add Task SVC #
                    04140 CLOFF   RST     40              ;Issue SVC
                    04150         XOR     A               ;Set Z for no error
                    04160         LD      H,A             ;Pass to HL
                    04170         LD      L,A             ;For normal exit
                    04180         RET
                    04190         ENDIF
                    04200 ;
                    04210 ;       Display the Time
                    04220 ;
2584 214226         04230 DSPTIME LD      HL,DATEBUF+9    ;Pt to space for time str
2587 E5             04240         PUSH    HL              ;Save pointer
2588                04250         @@TIME                  ;Xfer time into buffer
2588 3E13           00022         LD      A,19
258A EF             00023         RST     40
258B CD6A25         04260         CALL    DOCLOCK         ;Set/Reset Clock bit
258E C3F524         04270         JP      LOGDT           ;Log it & exit
2591                04280         SUBTTL  '<LBDATE - DATE/TIME Common Routines>'
```

LBDATE - DATE/TIME Common Routines

```
2591            04290           PAGE
                04300 ;
                04310 ;         DSPMDY - Xfer 3 char string from table to buffer
                04320 ;
                04330 ;         B  => Entry # in table to display
                04340 ;         HL => Table to fetch data from
                04350 ;         DE => Buffer to receive string
                04360 ;
2591 05         04370 DSPMDY    DEC     B               ;B = entry, Ø-6
2592 7D         04380           LD      A,L             ;P/u lsb of table start
2593 80         04390           ADD     A,B
2594 80         04400           ADD     A,B             ;Entries 3 bytes long
2595 80         04410           ADD     A,B
2596 6F         04420           LD      L,A             ;HL => Table entry
                04430 ;
                04440 ;         Transfer string into buffer
                04450 ;
2597 0603       04460           LD      B,3             ;Three chars to xfer
2599 7E         04470 DSPM1     LD      A,(HL)          ;P/u char from table
259A 12         04480           LD      (DE),A          ;Stuff into buffer
259B 23         04490           INC     HL              ;Bump
259C 13         04500           INC     DE
259D 10FA       04510           DJNZ    DSPM1           ;Three chars to xfer
259F C9         04520           RET                     ;Done - RETurn
                04530 ;
                04540 ;         PARSDAT - Parse TIME/DATE string entry
                04550 ;
                04560 ;         HL => Buffer containing string to parse
                04570 ;         C  => Delimiter (<"Ø" = DATE, <"Ø"or=":" = TIME)
                04580 ;
                04590 ;         DTBUF-DTBUF+2 <= Data in compressed format
                04600 ;         Z  - Set if successful
                04610 ;
25A0 118D26     04620 PARSDAT   LD      DE,DTBUF+2      ;Point to buf end
25A3 0603       04630           LD      B,3             ;Process 3 fields
                04640 ;
                04650 ;         Parse a field - Return NZ if bad
                04660 ;
25A5 D5         04670 PRS1      PUSH    DE              ;Save pointer
25A6 CDBC25     04680           CALL    PRS2            ;Get a digit pair
25A9 D1         04690           POP     DE              ;Recover pointer
25AA C0         04700           RET     NZ              ;Ret if bad digit pair
                04710 ;
                04720 ;         Good field - Stuff in buff, dec ptr, & count
                04730 ;
25AB 12         04740           LD      (DE),A          ;  else stuff the value
25AC 05         04750           DEC     B               ;Loop countdown
25AD C8         04760           RET     Z               ;Do for 3 fields
25AE 1B         04770           DEC     DE              ;Backup the pointer
                04780 ;
                04790 ;         Parsed a field - is the separator valid ?
                04800 ;
25AF 7E         04810           LD      A,(HL)          ;P/u separator
25B0 23         04820           INC     HL              ;Bump pointer
25B1 FE3A       04830           CP      ':'             ;Check for ':'
25B3 28F0       04840           JR      Z,PRS1          ;  loop if so
25B5 B9         04850           CP      C               ;Correct ?
25B6 3002       04860           JR      NC,PRSRET       ;NC = bad
25B8 18EB       04870           JR      PRS1            ;Loop if OK
                04880 ;
```

LBDATE - DATE/TIME Common Routines

```
25BA B7       Ø489Ø PRSRET  OR      A               ;Set NZ
25BB C9       Ø49ØØ         RET                     ;Return bad entry
              Ø491Ø ;
              Ø492Ø ;       PRS2 - Parse a digit pair at HL
              Ø493Ø ;
25BC CDD225   Ø494Ø PRS2    CALL    PRS4            ;Get a digit
25BF 3ØØF     Ø495Ø         JR      NC,PRS3         ;Illegal - return
              Ø496Ø ;
              Ø497Ø ;       Legal Digit - Multiply by 1Ø
              Ø498Ø ;
25C1 5F       Ø499Ø         LD      E,A             ;Multiply by ten
25C2 Ø7       Ø5ØØØ         RLCA                    ;X 2
25C3 Ø7       Ø5Ø1Ø         RLCA                    ;X 4
25C4 83       Ø5Ø2Ø         ADD     A,E             ;X 5
25C5 Ø7       Ø5Ø3Ø         RLCA                    ;X 1Ø
25C6 5F       Ø5Ø4Ø         LD      E,A             ;Stuff in E
              Ø5Ø5Ø ;
              Ø5Ø6Ø ;       Get another digit
              Ø5Ø7Ø ;
25C7 CDD225   Ø5Ø8Ø         CALL    PRS4            ;Get ones digit
25CA 3ØØ4     Ø5Ø9Ø         JR      NC,PRS3         ;Bad - return NZ
              Ø51ØØ ;
              Ø511Ø ;       Legal digit - Add to tens digit & set Z flag
              Ø512Ø ;
25CC 83       Ø513Ø         ADD     A,E             ;Accumulate new digit
25CD 5F       Ø514Ø         LD      E,A             ;Save 2-digit value
25CE BF       Ø515Ø         CP      A               ;Clear flags
25CF C9       Ø516Ø         RET                     ;Return Z
              Ø517Ø ;
              Ø518Ø ;       Force NZ & Return
              Ø519Ø ;
25DØ B7       Ø52ØØ PRS3    OR      A               ;Set NZ
25D1 C9       Ø521Ø         RET                     ;RETurn
              Ø522Ø ;
              Ø523Ø ;       Pick up a digit and convert to binary
              Ø524Ø ;
25D2 7E       Ø525Ø PRS4    LD      A,(HL)          ;P/u a digit &
25D3 23       Ø526Ø         INC     HL              ;  bump ptr
25D4 D63Ø     Ø527Ø         SUB     'Ø'             ;Convert to binary
25D6 FEØA     Ø528Ø         CP      1Ø              ;Legal ?
25D8 C9       Ø529Ø         RET                     ;C - legal, NC - illegal
              Ø53ØØ ;
              Ø531Ø ;       Parameter table
              Ø532Ø ;
ØØ8Ø          Ø533Ø NUM     EQU     8ØH
ØØ4Ø          Ø534Ø FLAG    EQU     4ØH
ØØ2Ø          Ø535Ø STR     EQU     2ØH
ØØ1Ø          Ø536Ø ABB     EQU     1ØH
              Ø537Ø ;
25D9 8Ø       Ø538Ø PRMTBL$ DB      8ØH             ;6.x Parameter Table
              Ø539Ø ;
25DA 55       Ø54ØØ         DB      FLAG!ABB!5
25DB 43       Ø541Ø         DB      'CLOCK'
     4C 4F 43 4B
25EØ ØØ       Ø542Ø CRESP   DB      Ø
25E1 7325     Ø543Ø         DW      CLOCK+1
25E3 ØØ       Ø544Ø         DB      Ø
              Ø545Ø ;
26ØØ          Ø546Ø         ORG     $<-8+1<+8
```

LBDATE - DATE/TIME Common Routines

```
                    05470 ;
2600 53             05480 DAYTBL  DB      'SunMonTueWedThuFriSat'
     75 6E 4D 6F 6E 54 75 65
     57 65 64 54 68 75 46 72
     69 53 61 74
2615 4A             05490 MONTBL  DB      'JanFebMarAprMayJunJulAugSepOctNovDec'
     61 6E 46 65 62 4D 61 72
     41 70 72 4D 61 79 4A 75
     6E 4A 75 6C 41 75 67 53
     65 70 4F 63 74 4E 6F 76
     44 65 63
2639 44             05500 DATEBUF DB      'Day, Mon xx, 198x',CR
     61 79 2C 20 4D 6F 6E 20
     78 78 2C 20 31 39 38 78
     0D
264B 00             05510 MAXDAYS DB      0,31,28,31,30,31,30,31,31,30,31,30,31
     1F 1C 1F 1E 1F 1E 1F 1F
     1E 1F 1E 1F
2658 44             05520 NODATE$ DB      'Date not in system',CR
     61 74 65 20 6E 6F 74 20
     69 6E 20 73 79 73 74 65
     6D 0D
266B 42             05530 BADDAT$ DB      'Bad Date format',CR
     61 64 20 44 61 74 65 20
     66 6F 72 6D 61 74 0D
267B 42             05540 BADTIM$ DB      'Bad Time format',CR
     61 64 20 54 69 6D 65 20
     66 6F 72 6D 61 74 0D
                    05550 ;
                    05560 ;
268B                05570 DTBUF   EQU     $
268E                05580 DUMBUF  EQU     $+3
                    05590 ;
2403                05600         END     DATE
```

LBDATE - DATE/TIME Common Routines

@$SYS	Ø8FØ	@@1	ØØØØ	@@2	ØØØØ
@@3	ØØØØ	@@4	ØØØØ	@ADTSK	ØØ1D
@BANK	Ø877	@BYTEIO	13ØØ	@CHNIO	Ø689
@CKBRKC	Ø553	@CLS	Ø545	@CTL	Ø623
@DATE	Ø7A8	@DIV16	Ø6E3	@DSP	Ø642
@DSPLY	Ø52D	@FRENCH	ØØØØ	@GERMAN	ØØØØ
@GET	Ø638	@HEX16	Ø7BD	@HEX8	Ø7C2
@HEXDEC	Ø6F6	@HZ5Ø	ØØØØ	@INTL	ØØØØ
@JCL	Ø63Ø	@KBD	Ø635	@KEY	Ø628
@KEYIN	Ø585	@KITSK	ØØ89	@LOGER	Ø5Ø3
@LOGOT	Ø5ØØ	@MOD2	ØØØØ	@MOD4	FFFF
@MSG	Ø53Ø	@MUL16	Ø6C9	@OPREG	ØØ84
@PRINT	Ø528	@PRT	Ø63D	@PUT	Ø645
@RMTSK	ØØ1E	@RSTNMI	ØFE9	@RSTREG	Ø68Ø
@TIME	Ø78D	@USA	FFFF	@VDCTL	ØB99
@VDCTL3	ØD38	@_VDCTL	ØD42	ABB	ØØ1Ø
ADDR_2_ROWCOL	ØDF1	BADDAT$	266B	BADFMT	25Ø3
BADFMT2	245Ø	BADTIM$	267B	BAR$	Ø2Ø1
BEGINA	240C	BEGINB	2513	BOOTST$	439D
BUR$	Ø2ØØ	CASHK$	ØA7B	CFLAG$	ØØ6C
CLK_SLT	ØØØ5	CLOCK	2572	CLRSTK	2526
CORE$	Ø3ØØ	CR	ØØØD	CRESP	25EØ
CRTBGN$	F8ØØ	DATE	24Ø3	DATE$	ØØ33
DATEBUF	2639	DAYLP	247E	DAYTBL	26ØØ
DAYTBL$	Ø4C7	DCBKL$	ØØ31	DCT$	Ø47Ø
DFLAG$	ØØ6D	DIS_DO_RAM	Ø846	DIV1Ø	24D6
DIV7	24A7	DOCLOCK	256A	DODATA$	ØB94
DODCB$	Ø21Ø	DO_CONTROL	ØC44	DO_DSPCHAR	ØCB8
DO_INVERT_DIS	ØC8C	DO_INVERT_ENA	ØC89	DO_INVERT_OFF	ØC9B
DO_MASK	ØØØØ	DO_RET	ØBCB	DO_RET1	ØBCC
DO_SCROLL	ØCCE	DO_TABS	ØBEA	DSKTYP$	Ø4CØ
DSPDATE	245B	DSPM1	2599	DSPMDY	2591
DSPTIME	2584	DTBUF	268B	DTPMT$	Ø4C2
DUMBUF	268E	DVREND$	ØFF4	DVRHI$	Ø2Ø6
ENADIS_DO_RAM	Ø817	FDDINT$	ØØØE	FLAG	ØØ4Ø
FLGTAB$	ØØ6A	GDPARMS	2539	GETPRMS	252Ø
GET_@_ROWCOL	ØDAE	GOTDATE	2465	HERTZ$	Ø75Ø
HIGH$	Ø4ØE	IFLAG$	ØØ72	INBUF$	Ø42Ø
INTVC$	ØØ3E	IOERR	252F	JCLCB$	Ø2Ø3
JLDCB$	Ø23Ø	KCK@	Ø7D6	KFLAG$	ØØ74
KIDATA$	Ø8FC	KIDCB$	Ø2Ø8	LBANK$	Ø2Ø2
LOGABRT	25Ø6	LOGDT	24F5	LOGMSG	24F8
MAXDAY$	Ø4Ø1	MAXDAYS	264B	MODOUT$	ØØ76
MONTBL	2615	MONTBL$	Ø4DC	NFLAG$	ØØ77
NODATE$	2658	NOTLDØ	24E5	NOTLEAP	244Ø
NUM	ØØ8Ø	OPREG$	ØØ78	OPREG_SV_AREA	Ø86E
OPREG_SV_PTR	Ø835	PAKNAM$	Ø41Ø	PARSDAT	25AØ
PAUSE@	Ø382	PCSAVE$	Ø7AF	PDRV$	ØØ1B
PRDCB$	Ø218	PRMTBL$	25D9	PRS1	25A5
PRS2	25BC	PRS3	25DØ	PRS4	25D2
PRSRET	25BA	PUDAY	2476	PUTA@DE	ØDCD
PUT_@	ØDCA	PUT_@_ROWCOL	ØDC6	RFLAG$	ØØ7B
ROWCOL_2_ADDR	ØDDØ	RSTOR$	Ø4C4	S1DCB$	Ø238
SET_SCROLL	ØCF3	SFLAG$	ØØ7C	SIDCB$	Ø22Ø
SODCB$	Ø228	STACK$	Ø38Ø	START$	ØØØØ
STR	ØØ2Ø	TIME	25ØA	TIME$	ØØ2D
TIMER$	ØØ2C	TIMSL$	ØØ2B	TIMTSK$	Ø713
TLOOP	2514	TMPMT$	Ø4C3	TRACE_INT	Ø7B1

LBDATE - DATE/TIME Common Routines

TYPHK$	ØA8F	TYPTSK$	ØB26	VFLAG$	ØØ7F
ZERO$	Ø4Ø1	@@ABORT	9EØ9	@@ADTSK	9E9C
@@BANK	A3B4	@@BKSP	AØ94	@@BREAK	A3CA
@@CHNIO	9DF4	@@CKBRKC	A418	@@CKDRV	9EFØ
@@CKEOF	AØA9	@@CKTSK	9E87	@@CLOSE	AØ7F
@@CLS	A4Ø2	@@CMNDI	9E33	@@CMNDR	9E48
@@CTL	9C58	@@DATE	9DCA	@@DCSTAT	9F2F
@@DEBUG	9E72	@@DECHEX	A334	@@DIRRD	A2A1
@@DIRWR	A2B6	@@DIV16	A31F	@@DIV8	A3ØA
@@DODIR	9FØ5	@@DSP	9C1C	@@DSPLY	9CBC
@@ERROR	9E5D	@@EXIT	9E1E	@@FEXT	A2ØE
@@FLAGS	A39E	@@FNAME	A223	@@FSPEC	A1F9
@@GATRD	A28C	@@GATWR	A2CB	@@GET	9C3Ø
@@GTDCB	A24D	@@GTDCT	A238	@@GTMOD	A262
@@HDFMT	9FD7	@@HEX16	A373	@@HEX8	A35E
@@HEXDEC	A349	@@HIGH$	A388	@@INIT	AØ55
@@KBD	9C94	@@KEY	9CØ8	@@KEYIN	9CA8
@@KLTSK	9EDB	@@LOAD	A1CF	@@LOC	AØBE
@@LOF	AØD3	@@LOGER	9CF3	@@LOGOT	9DØ8
@@MSG	9D3F	@@MUL16	A2F5	@@MUL8	A2EØ
@@OPEN	AØ6A	@@PARAM	9DB5	@@PAUSE	9DAØ
@@PEOF	AØE8	@@POSN	AØFD	@@PRINT	9D54
@@PRT	9C6C	@@PUT	9C44	@@RAMDIR	9F1A
@@RDSEC	9FAD	@@RDSSC	A277	@@READ	A112
@@REMOV	AØ4Ø	@@RENAM	AØ2B	@@REW	A127
@@RMTSK	9EB1	@@RPTSK	9EC6	@@RREAD	A13C
@@RSLCT	9F98	@@RSTOR	9F59	@@RUN	A1E4
@@RWRIT	A151	@@SEEK	9F83	@@SEEKSC	A166
@@SKIP	A17B	@@SLCT	9F44	@@STEPI	9F6E
@@TIME	9DDF	@@VDCTL	9D8B	@@VER	A19Ø
@@VRSEC	9FC2	@@WEOF	A1A5	@@WHERE	9C8Ø
@@WRITE	A1BA	@@WRSEC	9FEC	@@WRSSC	AØØ1
@@WRTRK	AØ16				

24Ø3 is the transfer address
ØØØØØ Total errors

NOTES:

Command: DEBUG, VERIFY

Library: SYS7/SYS

ISAM # : 14H, 1BH

```
                00100 ;LBDEBUG/ASM - DEBUG VERIFY Commands
0000            00110         TITLE   <DEBUG/VERIFY - LS-DOS 6.2>
                00120 ;
000D            00130 CR      EQU     13
0000            00140 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00150 *LIST   OFF                     ;Get SYS0/EQU
                00170 *LIST   ON
                00180 ;
2400            00190         ORG     2400H
                00200 ;
2400 C37824     00210         JP      VERIFY          ;Verify entry point
                00220 ;
                00230 ;       Debug entry
                00240 ;
2403 11F124     00250 DEBUG   LD      DE,PRM1TBL      ;Pt to y/n on/off parms
2406            00260         @@PARAM                 ;Get parameters
2406 3E11       00001         LD      A,17
2408 EF         00002         RST     40
2409 C2C324     00270         JP      NZ,IOERR        ;Jump on parm error
240C CDB624     00280         CALL    NPARM           ;Ck on yes or no entry
240F 2821       00290         JR      Z,DBG2          ;Go if no (OFF)
2411 3A7324     00300         LD      A,(EPARM)       ;Ck on extended DEBUG
2414 B7         00310         OR      A               ;  requested
2415 280A       00320         JR      Z,DBG0          ;Go if not
2417 FDCB0246   00330         BIT     0,(IY+'C'-'A')  ;Can't put extended DEBUG
241B C2CD24     00340         JP      NZ,CANT         ;  if himem not available
241E CD7524     00350         CALL    DBG3            ;Call if EXT entered
2421 FDCB12FE   00360 DBG0    SET     7,(IY+'S'-'A')  ;Turn on DEBUG flag
2425 1608       00370         LD      D,8             ;Set to mask bit 3
2427 CD9824     00380         CALL    GET$DO
242A AF         00390 DBG0A   XOR     A               ;  & enable DEBUG
242B 329F19     00400 DBG1    LD      (@DBGHK),A
242E 210000     00410 EXIT    LD      HL,0            ;Set no error
2431 C9         00420         RET
                00430 ;
2432 FDCB12BE   00440 DBG2    RES     7,(IY+'S'-'A')  ;Turn off DEBUG flag
2436 1680       00450         LD      D,80H           ;Set to mask bit 7
2438 CD9824     00460         CALL    GET$DO
243B 3A7324     00470 DBG2A   LD      A,(EPARM)       ;Ck on extended DEBUG
243E B7         00480         OR      A               ;  requested
243F 282E       00490         JR      Z,DBG4          ;Go if not
                00500 ;
                00510 ;       User requested we remove extended DEBUG
                00520 ;
2441 FDCB0246   00530         BIT     0,(IY+'C'-'A')  ;Can we alter HIGH$?
2445 2028       00540         JR      NZ,DBG4         ;Go if not
2447 ED5BA419   00550         LD      DE,(EXTDBG$)    ;Is extended resident?
244B 2124EB     00560         LD      HL,-ORARET@     ;Ck if vector same as
244E ED5A       00570         ADC     HL,DE           ;  the default
2450 281D       00580         JR      Z,DBG4          ;Go if not resident
2452 210000     00590         LD      HL,0            ;Set to get HIGH$
2455 45         00600         LD      B,L
2456            00610         @@HIGH$                 ;P/u current pointer
2456 3E64       00003         LD      A,100
2458 EF         00004         RST     40
2459 23         00620         INC     HL              ;Point to assumed module
245A AF         00630         XOR     A               ;  & check if extended
245B ED52       00640         SBC     HL,DE           ;  debugger is lowest
```

```
245D 201Ø      ØØ65Ø          JR      NZ,DBG4         ;Can't release if not
245F 13        ØØ66Ø          INC     DE              ;Point to last byte used
246Ø 13        ØØ67Ø          INC     DE
2461 EB        ØØ68Ø          EX      DE,HL           ;This will be new
2462 5E        ØØ69Ø          LD      E,(HL)
2463 23        ØØ7ØØ          INC     HL              ;  HIGH$
2464 56        ØØ71Ø          LD      D,(HL)
2465 EB        ØØ72Ø          EX      DE,HL           ;New high to HL
2466           ØØ73Ø          @@HIGH$                 ;Reset high$
2466 3E64      ØØØØ5          LD      A,1ØØ
2468 EF        ØØØØ6          RST     4Ø
2469 21DC14    ØØ74Ø          LD      HL,ORARET@      ;Reset extended debugger
246C 22A419    ØØ75Ø          LD      (EXTDBG$),HL    ;  vector to default
246F 3EC9      ØØ76Ø DBG4     LD      A,ØC9H          ;Turn off DEBUG vector
2471 18B8      ØØ77Ø          JR      DBG1
               ØØ78Ø ;
2473 ØØØØ      ØØ79Ø EPARM    DW      Ø
2475 3E9B      ØØ8ØØ DBG3     LD      A,9BH           ;Get SYS9 loaded & up
2477 EF        ØØ81Ø          RST     28H
               ØØ82Ø ;
               ØØ83Ø ;        Verify entry
               ØØ84Ø ;
2478 11Ø125    ØØ85Ø VERIFY   LD      DE,PRM2TBL      ;Pt to ext + above
247B           ØØ86Ø          @@PARAM
247B 3E11      ØØØØ7          LD      A,17
247D EF        ØØØØ8          RST     4Ø
247E 2Ø43      ØØ87Ø          JR      NZ,IOERR        ;Quit on parm error
248Ø CDB624    ØØ88Ø          CALL    NPARM           ;Ck yes/no
2483 213115    ØØ89Ø          LD      HL,@WRITE       ;Init to vector WRITE
2486 FDCBØ396  ØØ9ØØ          RES     2,(IY+'D'-'A')  ;Indicate VERIFY=OFF
248A 28Ø7      ØØ91Ø          JR      Z,VER1          ;Use WRITE if VER=(OFF)
248C 216Ø15    ØØ92Ø          LD      HL,@VER         ;Init to vector VERIFY
248F FDCBØ3D6  ØØ93Ø          SET     2,(IY+'D'-'A')  ;Indicate VERIFY=ON
2493 2296Ø1    ØØ94Ø VER1     LD      (75*2+SVCTAB$),HL       ;Change SVCTAB vector
2496 1896      ØØ95Ø          JR      EXIT
               ØØ96Ø ;
               ØØ97Ø ;        Routine to save or restore SPACE COMPRESSION state
               ØØ98Ø ;
2498 D5        ØØ99Ø GET$DO   PUSH    DE              ;Save mask value
2499 11B224    Ø1ØØØ          LD      DE,$DO$
249C           Ø1Ø1Ø          @@GTMOD
249C 3E53      ØØØØ9          LD      A,83
249E EF        ØØØ1Ø          RST     4Ø
249F 21Ø4ØØ    Ø1Ø2Ø          LD      HL,4            ;Point to 1st data byte
24A2 19        Ø1Ø3Ø          ADD     HL,DE
24A3 D1        Ø1Ø4Ø          POP     DE              ;Rcvr mask value
24A4 CØ        Ø1Ø5Ø          RET     NZ              ;Return if not found
24A5 7E        Ø1Ø6Ø          LD      A,(HL)          ;P/u existing state
24A6 A2        Ø1Ø7Ø          AND     D               ;Strip state of space comp
24A7 Ø7        Ø1Ø8Ø          RLCA                    ;  & move to bit 7 or 3
24A8 Ø7        Ø1Ø9Ø          RLCA
24A9 Ø7        Ø11ØØ          RLCA
24AA Ø7        Ø111Ø          RLCA
24AB 5F        Ø112Ø          LD      E,A             ;Tempy save
24AC 7E        Ø113Ø          LD      A,(HL)
24AD E677      Ø114Ø          AND     77H             ;Strip bits 3 and 7
24AF B3        Ø115Ø          OR      E               ;Merge comp state
24BØ 77        Ø116Ø          LD      (HL),A
24B1 C9        Ø117Ø          RET
               Ø118Ø ;
24B2 24        Ø119Ø $DO$     DB      '$DO',3
```

```
     44 4F 03
                00120 ;
                00121 ;       Parameter parsing of yes/no
                00122 ;
24B6 010000     00123 NPARM   LD      BC,0            ;P/u "no" parm entry
24B9            00124         @@FLAGS                 ;Point to flag table
24B9 3E65       00011         LD      A,101
24BB EF         00012         RST     40
24BC 78         00125         LD      A,B
24BD B1         00126         OR      C               ;If NO, make Z
24BE EEFF       00127         XOR     0FFH
24C0 C9         00128         RET
24C1 0000       00129 YPARM   DW      0
                00130 ;
                00131 ;       I/O Error Processing
                00132 ;
24C3 6F         00133 IOERR   LD      L,A             ;Set HL = error code
24C4 2600       00134         LD      H,0             ;
24C6 F6C0       00135         OR      0C0H            ;Short error
24C8 4F         00136         LD      C,A             ;Xfer error # to C
24C9            00137         @@ERROR                 ;Display error
24C9 3E1A       00013         LD      A,26
24CB EF         00014         RST     40
24CC C9         00138         RET                     ;Return to DOS
                00139 ;
                00140 ;       Error exits
                00141 ;
24CD 21D724     00142 CANT    LD      HL,CANT$
24D0            00143         @@LOGOT
                00015         IFEQ    00H,1
                00016         LD      HL,
                00017         ENDIF
24D0 3E0C       00018         LD      A,12
24D2 EF         00019         RST     40
24D3 21FFFF     00144         LD      HL,-1
24D6 C9         00145         RET
                00146 ;
24D7 4E         00147 CANT$   DB      'No memory space available',CR
     6F 20 6D 65 6D 6F 72 79
     20 73 70 61 63 65 20 61
     76 61 69 6C 61 62 6C 65
     0D
                00148 ;
24F1 45         00149 PRM1TBL DB      'EXT    '
     58 54 20 20 20
24F7 7324       00150         DW      EPARM
24F9 45         00151         DB      'E      '
     20 20 20 20 20
24FF 7324       00152         DW      EPARM
2501 4F         00153 PRM2TBL DB      'ON     '
     4E 20 20 20 20
2507 C124       00154         DW      YPARM
2509 4F         00155         DB      'OFF    '
     46 46 20 20 20
250F B724       00156         DW      NPARM+1
2511 59         00157         DB      'Y      '
     20 20 20 20 20
2517 C124       00158         DW      YPARM
2519 4E         00159         DB      'N      '
     20 20 20 20 20
251F B724       00160         DW      NPARM+1
```

```
2521 00         01610           NOP
                01620 ;
2403            01630           END     DEBUG
```

$A1	Ø3B7	$A2	Ø3B8	$A3	Ø3B9		
$CKEOF	147Ø	DO	24B2	@$SYS	Ø8FØ		
@@1	ØØØØ	@@2	ØØØØ	@@3	ØØØØ		
@@4	ØØØØ	@ABORT	1BØ8	@ADTSK	1CDA		
@BANK	Ø877	@BKSP	1486	@BREAK	196F		
@BYTEIO	13ØØ	@CHNIO	Ø689	@CKBRKC	Ø553		
@CKDRV	1993	@CKEOF	158F	@CKTSK	1CF5		
@CLOSE	1999	@CLS	Ø545	@CMNDI	197E		
@CMNDR	197B	@CTL	Ø623	@DATE	Ø7A8		
@DBGHK	199F	@DCINIT	19CØ	@DCRES	19C4		
@DCSTAT	19B5	@DCTBYT	1A2B	@DEBUG	19AØ		
@DECHEX	Ø3E1	@DIRCYL	18F7	@DIRRD	18BB		
@DIRWR	18Ø3	@DIV16	Ø6E3	@DIV8	1927		
@DODIR	19AF	@DOKEY	19A9	@DSP	Ø642		
@DSPLY	Ø52D	@ERROR	1BØF	@EXIT	1BØB		
@FEXT	1984	@FLAGS	196A	@FNAME	199C		
@FRENCH	ØØØØ	@FSPEC	1981	@GATRD	1874		
@GATWR	1875	@GERMAN	ØØØØ	@GET	Ø638		
@GTDCB	199Ø	@GTDCT	1A1E	@GTMOD	19B2		
@HDFMT	19E4	@HEX16	Ø7BD	@HEX8	Ø7C2		
@HEXDEC	Ø6F6	@HIGH$	1948	@HITRD	1897		
@HITWR	1898	@HZ5Ø	ØØØØ	@ICNFG	ØØ86		
@INIT	198D	@INTL	ØØØØ	@IPL	1BF2		
@JCL	Ø63Ø	@KBD	Ø635	@KEY	Ø628		
@KEYIN	Ø585	@KITSK	ØØ89	@KLTSK	1CDØ		
@LOAD	1B38	@LOC	14B3	@LOF	14DE		
@LOGER	Ø5Ø3	@LOGOT	Ø5ØØ	@MOD2	ØØØØ		
@MOD4	FFFF	@MSG	Ø53Ø	@MUL16	Ø6C9		
@MUL8	19ØA	@NMI	ØØ66	@OPEN	198A		
@OPREG	ØØ84	@PARAM	1987	@PAUSE	Ø382		
@PEOF	14A2	@POSN	1434	@PRINT	Ø528		
@PRT	Ø63D	@PUT	Ø645	@RAMDIR	19AC		
@RDHDR	19D8	@RDSEC	19F4	@RDSSC	18D8		
@RDTRK	19EØ	@READ	1513	@REMOVE	19A6		
@RENAME	1996	@REW	149B	@RMTSK	1CD7		
@RPTSK	1CEB	@RREAD	1473	@RSLCT	19D4		
@RSTØØ	ØØØØ	@RST08	ØØØ8	@RST1Ø	ØØ1Ø		
@RST18	ØØ18	@RST2Ø	ØØ2Ø	@RST28	ØØ28		
@RST3Ø	ØØ3Ø	@RST38	ØØ38	@RSTNMI	ØFE9		
@RSTOR	19C8	@RSTREG	Ø68Ø	@RUN	1B1D		
@RWRIT	13AD	@SEEK	19DØ	@SEEKSC	1421		
@SKIP	143Ø	@SLCT	19BC	@SOUND	Ø392		
@STEPI	19CC	@TIME	Ø78D	@USA	FFFF		
@VDCTL	ØB99	@VDCTL3	ØD38	@VER	156Ø		
@VRSEC	19DC	@WEOF	14EC	@WHERE	1979		
@WRITE	1531	@WRSEC	19E8	@WRSSC	19EC		
@WRTRK	19FØ	@_VDCTL	ØD42	ADDR_2_ROWCOL	ØDF1		
AFLAG$	ØØ6A	AUTO?	1FF1	BAR$	Ø2Ø1		
BOOTST$	439D	BREAK?	1C6Ø	BRKVEC$	1C88		
BUR$	Ø2ØØ	CANT	24CD	CANT$	24D7		
CASHK$	ØA7B	CFCB$	ØØEØ	CFGFCB$	ØØEØ		
CFLAG$	ØØ6C	CKMOD@	1A7F	CKOPEN@	1568		
CONFIG$	2Ø3F	CORE$	Ø3ØØ	CR	ØØØD		
CRTBGN$	F8ØØ	CYL_GRN	16AE	D@FBYT8	1A26		
DATE$	ØØ33	DAYTBL$	Ø4C7	DBGØ	2421		
DBGØA	242A	DBG1	242B	DBG2	2432		
DBG2A	243B	DBG3	2475	DBG4	246F		
DBGSV$	ØØAØ	DCBKL$	ØØ31	DCT$	Ø47Ø		
DCTBYT8@	1A29	DCTFLD@	1A34	DEBUG	24Ø3		
DFLAG$	ØØ6D	DIRBUF$	23ØØ	DIS_DO_RAM	Ø846		

DODATA$	0B94	DODCB$	0210	DO_CONTROL	0C44
DO_DSPCHAR	0CB8	DO_INVERT_DIS	0C8C	DO_INVERT_ENA	0C89
DO_INVERT_OFF	0C9B	DO_MASK	0000	DO_RET	0BCB
DO_RET1	0BCC	DO_SCROLL	0CCE	DO_TABS	0BEA
DSKTYP$	04C0	DTPMT$	04C2	DVREND$	0FF4
DVRHI$	0206	EFLAG$	006E	ENADIS_DO_RAM	0817
EPARM	2473	EXIT	242E	EXTDBG$	19A4
FDDINT$	000E	FEMSK$	006F	FLGTAB$	006A
GET$DO	2498	GET_@_ROWCOL	0DAE	HERTZ$	0750
HIGH$	040E	HKRES$	1A6C	IFLAG$	0072
INBUF$	0420	INTIM$	003C	INTMSK$	003D
INTVC$	003E	IOERR	24C3	JCLCB$	0203
JDCB$	0024	JFCB$	00C0	JLDCB$	0230
JRET$	0026	KCK@	07D6	KFLAG$	0074
KIDATA$	08FC	KIDCB$	0208	LBANK$	0202
LDRV$	0023	LFLAG$	0075	LNKFCB@	1566
LOW$	001E	LSVC$	000D	MAXCOR$	2400
MAXDAY$	0401	MINCOR$	3000	MODOUT$	0076
MONTBL$	04DC	NFLAG$	0077	NPARM	24B6
OPREG$	0078	OPREG_SV_AREA	086E	OPREG_SV_PTR	0835
ORARET@	14DC	OSRLS$	003B	OSVER$	0085
OVRLY$	0069	PAKNAM$	0410	PAUSE@	0382
PCSAVE$	07AF	PDRV$	001B	PHIGH$	001C
PRDCB$	0218	PRM1TBL	24F1	PRM2TBL	2501
PUTA@DE	0DCD	PUT_@	0DCA	PUT_@_ROWCOL	0DC6
RFLAG$	007B	ROWCOL_2_ADDR	0DD0	RST38@	1BFF
RSTOR$	04C4	RWRIT@	13A2	S1DCB$	0238
SBUFF$	1D00	SET@EXEC	1A79	SET_SCROLL	0CF3
SFCB$	008C	SFLAG$	007C	SIDCB$	0220
SODCB$	0228	SPACE4$	2142	STACK$	0380
START$	0000	SVCRET$	000B	SVCTAB$	0100
SYSERR$	1B13	TCB$	004E	TFLAG$	007D
TIME$	002D	TIMER$	002C	TIMSL$	002B
TIMTSK$	0713	TMPMT$	04C3	TRACE_INT	07B1
TYPHK$	0A8F	TYPTSK$	0B26	USTOR$	0013
VER1	2493	VERIFY	2478	VFLAG$	007F
WRINT$	0080	YPARM	24C1	ZERO$	0401
ZEROA@	13A0	@@ABORT	798E	@@ADTSK	7A21
@@BANK	7F39	@@BKSP	7C19	@@BREAK	7F4F
@@CHNIO	7979	@@CKBRKC	7F9D	@@CKDRV	7A75
@@CKEOF	7C2E	@@CKTSK	7A0C	@@CLOSE	7C04
@@CLS	7F87	@@CMNDI	79B8	@@CMNDR	79CD
@@CTL	77DD	@@DATE	794F	@@DCSTAT	7AB4
@@DEBUG	79F7	@@DECHEX	7EB9	@@DIRRD	7E26
@@DIRWR	7E3B	@@DIV16	7EA4	@@DIV8	7E8F
@@DODIR	7A8A	@@DSP	77A1	@@DSPLY	7841
@@ERROR	79E2	@@EXIT	79A3	@@FEXT	7D93
@@FLAGS	7F23	@@FNAME	7DA8	@@FSPEC	7D7E
@@GATRD	7E11	@@GATWR	7E50	@@GET	77B5
@@GTDCB	7DD2	@@GTDCT	7DBD	@@GTMOD	7DE7
@@HDFMT	7B5C	@@HEX16	7EF8	@@HEX8	7EE3
@@HEXDEC	7ECE	@@HIGH$	7F0D	@@INIT	7BDA
@@KBD	7819	@@KEY	778D	@@KEYIN	782D
@@KLTSK	7A60	@@LOAD	7D54	@@LOC	7C43
@@LOF	7C58	@@LOGER	7878	@@LOGOT	788D
@@MSG	78C4	@@MUL16	7E7A	@@MUL8	7E65
@@OPEN	7BEF	@@PARAM	793A	@@PAUSE	7925
@@PEOF	7C6D	@@POSN	7C82	@@PRINT	78D9
@@PRT	77F1	@@PUT	77C9	@@RAMDIR	7A9F
@@RDSEC	7B32	@@RDSSC	7DFC	@@READ	7C97
@@REMOV	7BC5	@@RENAM	7BB0	@@REW	7CAC

```
@@RMTSK         7A36 @@RPTSK         7A4B @@RREAD         7CC1
@@RSLCT         7B1D @@RSTOR         7ADE @@RUN           7D69
@@RWRIT         7CD6 @@SEEK          7BØ8 @@SEEKSC        7CEB
@@SKIP          7DØØ @@SLCT          7AC9 @@STEPI         7AF3
@@TIME          7964 @@VDCTL         791Ø @@VER           7D15
@@VRSEC         7B47 @@WEOF          7D2A @@WHERE         78Ø5
@@WRITE         7D3F @@WRSEC         7B71 @@WRSSC         7B86
@@WRTRK         7B9B
24Ø3 is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

Command: DEVICE

Library: SYS6/SYS

ISAM # : 61H

```
                00100 ;LBDEVICE/ASM - DEVICE Command
0000            00110         TITLE   <DEVICE - LS-DOS 6.2>
                00120 ;
000A            00130 LF      EQU     10
000D            00140 CR      EQU     13
0000            00150 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00160 ;
2400            00170         ORG     2400H
                00180 ;
                00190 DEVICE
2400            00200         @@CKBRKC                ;Check for break
2400 3E6A       00001         LD      A,106
2402 EF         00002         RST     40
2403 2804       00210         JR      Z,DEVICEA       ;Continue if not
2405 21FFFF     00220         LD      HL,-1           ;  else abort
2408 C9         00230         RET
                00240 ;
2409 ED73CD27   00250 DEVICEA LD      (SAVESP+1),SP   ;Save stack pointer
240D 11B928     00260         LD      DE,PRMTBL$      ;First check for user parms
2410            00270         @@PARAM
2410 3E11       00003         LD      A,17
2412 EF         00004         RST     40
2413 C29527     00280         JP      NZ,IOERR        ;Go if parm error
2416            00290         @@FLAGS                 ;Get flag table pointer
2416 3E65       00005         LD      A,101
2418 EF         00006         RST     40
2419 CDBD27     00300         CALL    RESKFL          ;Reset Pause and Enter
241C FB         00310         EI                      ;Make sure they're on
241D 21FFFF     00320 DPARM   LD      HL,-1           ;Check Drive parameter
2420 7C         00330         LD      A,H
2421 B5         00340         OR      L
2422 CA7F25     00350         JP      Z,DEND          ;Go if D=NO
2425 0E00       00360         LD      C,0             ;Init to drive 0
2427 C5         00370 DEV1    PUSH    BC              ;Save drive #
2428 AF         00380         XOR     A               ;Reset flag stuff
2429 327C24     00390         LD      (WPTEST+1),A    ;  location
242C            00400         @@GTDCT                 ;Get DCT address
242C 3E51       00007         LD      A,81
242E EF         00008         RST     40
242F FD7E00     00410         LD      A,(IY+0)        ;Is this drive disabled?
2432 FEC3       00420         CP      0C3H
2434 C27725     00430         JP      NZ,POPDRV       ;Ignore if it is
2437            00440         @@CKDRV                 ;This drive available?
2437 3E21       00009         LD      A,33
2439 EF         00010         RST     40
243A 2022       00450         JR      NZ,DEV2         ;Go if no diskette
243C 1F         00460         RRA                     ;Shift C-flag to bit-7
243D 327C24     00470         LD      (WPTEST+1),A    ;  & save for WP test
2440 210029     00480         LD      HL,BUFFER       ;Pick up the GAT for the
2443 FD5609     00490         LD      D,(IY+9)        ;  pack name
2446 5D         00500         LD      E,L
2447            00510         @@RDSSC
2447 3E55       00011         LD      A,85
2449 EF         00012         RST     40
244A 3E14       00520         LD      A,20            ;"GAT read error
244C C29527     00530         JP      NZ,IOERR
244F 21D829     00540         LD      HL,BUFFER+0D8H  ;Shove bracket ETX
2452 365D       00550         LD      (HL),']'
```

```
2454 2C       ØØ56Ø          INC     L
2455 362Ø     ØØ57Ø          LD      (HL),' '
2457 2C       ØØ58Ø          INC     L
2458 36Ø3     ØØ59Ø          LD      (HL),3
245A 2EDØ     ØØ6ØØ          LD      L,ØDØH          ;Point to start of name
245C 18ØB     ØØ61Ø          JR      DEV2A
              ØØ62Ø ;
              ØØ63Ø ;        Drive info for this active drive
              ØØ64Ø ;
245E 21D827   ØØ65Ø DEV2     LD      HL,NOPACK$      ;Display pack name
2461 11DØ29   ØØ66Ø          LD      DE,BUFFER+ØDØH
2464 Ø1ØC ØØ  ØØ67Ø          LD      BC,12
2467 EDBØ     ØØ68Ø          LDIR
2469 3E3A     ØØ69Ø DEV2A    LD      A,':'           ;Output the colon
246B CD8227   ØØ7ØØ          CALL    BYTOUT
246E C1       ØØ71Ø          POP     BC              ;Get drive # back
246F C5       ØØ72Ø          PUSH    BC
247Ø 79       ØØ73Ø          LD      A,C             ;Get drive # converted
2471 C63Ø     ØØ74Ø          ADD     A,'Ø'           ;  to ASCII & display it
2473 CD8227   ØØ75Ø          CALL    BYTOUT
2476 3E2Ø     ØØ76Ø          LD      A,' '           ;Space out one
2478 CD8227   ØØ77Ø          CALL    BYTOUT
247B 3EØØ     ØØ78Ø WPTEST   LD      A,Ø             ;P/u CKDRV FDC status
247D Ø7       ØØ79Ø          RLCA                    ;Hardware write protect?
247E 38ØA     ØØ8ØØ          JR      C,DEV2B         ;Force "WP" if it is
248Ø FDCBØ37E ØØ81Ø          BIT     7,(IY+3)        ;Test software WP
2484 3E2Ø     ØØ82Ø          LD      A,' '           ;Output '  ' for read &
2486 Ø62Ø     ØØ83Ø          LD      B,' '           ;  write access or
2488 28Ø4     ØØ84Ø          JR      Z,$+6
248A 3E57     ØØ85Ø DEV2B    LD      A,'W'           ;  WP for read only
248C Ø65Ø     ØØ86Ø          LD      B,'P'
248E CD8227   ØØ87Ø          CALL    BYTOUT
2491 78       ØØ88Ø          LD      A,B             ;Xfer the 2nd char
2492 CD8227   ØØ89Ø          CALL    BYTOUT          ;  & display it
2495 3E2Ø     ØØ9ØØ          LD      A,' '
2497 CD8227   ØØ91Ø          CALL    BYTOUT
249A 3E5B     ØØ92Ø          LD      A,'['           ;Left bracket
249C CD8227   ØØ93Ø          CALL    BYTOUT
249F 21DØ29   ØØ94Ø          LD      HL,BUFFER+ØDØH  ;Write the pack name
24A2 CD7227   ØØ95Ø          CALL    LINOUT
              ØØ96Ø ;
              ØØ97Ø ;        Determine if 5" or 8"
              ØØ98Ø ;
24A5 FDCBØ36E ØØ99Ø          BIT     5,(IY+3)        ;Test 5"/8" drive
24A9 3E35     Ø1ØØØ          LD      A,'5'           ;Init to 5
24AB 28Ø2     Ø1Ø1Ø          JR      Z,$+4           ;Bypass if not 8
24AD 3E38     Ø1Ø2Ø          LD      A,'8'           ;  else init to 8
24AF CD8227   Ø1Ø3Ø          CALL    BYTOUT
24B2 FDCBØ35E Ø1Ø4Ø          BIT     3,(IY+3)        ;Test rigid/floppy
24B6 216Ø28   Ø1Ø5Ø          LD      HL,FLOPY$       ;Init to floppy
24B9 28Ø3     Ø1Ø6Ø          JR      Z,$+5           ;Bypass if that kind
24BB 216B28   Ø1Ø7Ø          LD      HL,RIGID$       ;  else is hard
24BE CD7227   Ø1Ø8Ø          CALL    LINOUT
24C1 FD7EØ4   Ø1Ø9Ø          LD      A,(IY+4)        ;Output drive select addr
24C4 E6ØF     Ø11ØØ          AND     ØFH             ;  in ASCII
24C6 C69Ø     Ø111Ø          ADD     A,9ØH
24C8 27       Ø112Ø          DAA
24C9 CE4Ø     Ø113Ø          ADC     A,4ØH
24CB 27       Ø114Ø          DAA
24CC CD8227   Ø115Ø          CALL    BYTOUT
24CF FD6EØ6   Ø116Ø DEV3     LD      L,(IY+6)        ;P/u highest cylinder
```

```
24D2 26ØØ      Ø117Ø          LD      H,Ø
24D4 23        Ø118Ø          INC     HL              ;Adjust for zero offset
24D5 FDCBØ35E  Ø119Ø          BIT     3,(IY+3)        ;Hard drive?
24D9 28Ø7      Ø12ØØ          JR      Z,DEV4          ;Bypass if soft
24DB FDCBØ46E  Ø121Ø          BIT     5,(IY+4)        ;2-sided hard drives
24DF 28Ø1      Ø122Ø          JR      Z,DEV4          ;  are 2*cyl
24E1 29        Ø123Ø          ADD     HL,HL           ;  & multiply by 2
24E2 117D28    Ø124Ø DEV4     LD      DE,COMMA$       ;Convert # of cyls to
24E5 CDA726    Ø125Ø          CALL    CVRTDEC         ;  decimal & stuff in msg
24E8 217628    Ø126Ø          LD      HL,CYLS$        ;Display cyls=xxx
24EB CD7227    Ø127Ø          CALL    LINOUT
24EE FDCBØ35E  Ø128Ø          BIT     3,(IY+3)        ;Bypass if soft drive
24F2 2811      Ø129Ø          JR      Z,FLOPPY
24F4 FDCBØ356  Ø13ØØ          BIT     2,(IY+3)        ;Test fixed/removable
24F8 218328    Ø131Ø          LD      HL,REMOV$       ;Init to removable
24FB 28Ø3      Ø132Ø          JR      Z,$+5           ;Bypass if that way
24FD 218D28    Ø133Ø          LD      HL,FIXED$       ;  else init fixed
25ØØ CD7227    Ø134Ø          CALL    LINOUT
25Ø3 186F      Ø135Ø          JR      ENDLINE         ;Bypass DEN, STEP, DLY
               Ø136Ø ;
               Ø137Ø ;        Next section deals only with floppies
               Ø138Ø ;
25Ø5 FDCBØ376  Ø139Ø FLOPPY   BIT     6,(IY+3)        ;Test SDEN/DDEN
25Ø9 3E53      Ø14ØØ          LD      A,'S'           ;Init to sden
25ØB 28Ø2      Ø141Ø          JR      Z,$+4           ;Bypass if sden
25ØD 3E44      Ø142Ø          LD      A,'D'           ;  else init to dden
25ØF CD8227    Ø143Ø          CALL    BYTOUT
2512 219328    Ø144Ø          LD      HL,DEN$         ;Now display "den"
2515 CD7227    Ø145Ø          CALL    LINOUT
2518 FDCBØ46E  Ø146Ø          BIT     5,(IY+4)        ;Test # of sides
251C 3E31      Ø147Ø          LD      A,'1'           ;Init to 1
251E 28Ø1      Ø148Ø          JR      Z,$+3           ;Bypass if single sided
252Ø 3C        Ø149Ø          INC     A               ;  else bump to 2
2521 CD8227    Ø15ØØ          CALL    BYTOUT
2524 219F28    Ø151Ø          LD      HL,STEP$        ;Display "step="
2527 CD7227    Ø152Ø          CALL    LINOUT
252A FD7EØ3    Ø153Ø          LD      A,(IY+3)        ;P/u step rate & 8/5
252D E623      Ø154Ø          AND     23H             ;Convert step rate to an
252F 47        Ø155Ø          LD      B,A             ;  index into the table
2530 ØF        Ø156Ø          RRCA
2531 ØF        Ø157Ø          RRCA                    ;5/8 bit to bit 2
2532 ØF        Ø158Ø          RRCA
2533 BØ        Ø159Ø          OR      B               ;Merge step rate
2534 Ø7        Ø16ØØ          RLCA
2535 E6ØE      Ø161Ø          AND     ØEH             ;Mask off garbage
2537 215Ø28    Ø162Ø          LD      HL,STPRAT$      ;Get table base
253A 85        Ø163Ø          ADD     A,L             ;Add table lo order
253B 6F        Ø164Ø          LD      L,A             ;Set lo-order
253C 8C        Ø165Ø          ADC     A,H
253D 95        Ø166Ø          SUB     L
253E 67        Ø167Ø          LD      H,A
253F 7E        Ø168Ø          LD      A,(HL)          ;P/u 1st step char
2540 23        Ø169Ø          INC     HL              ;Bump to second
2541 CD8227    Ø17ØØ          CALL    BYTOUT          ;Display the first
2544 7E        Ø171Ø          LD      A,(HL)          ;P/u the second
2545 CD8227    Ø172Ø          CALL    BYTOUT          ;Display the second
2548 21A728    Ø173Ø          LD      HL,MS$          ;Display "ms,"
254B CD7227    Ø174Ø          CALL    LINOUT
254E FDCBØ36E  Ø175Ø          BIT     5,(IY+3)        ;Bypass DELAY if 8"
2552 2Ø2Ø      Ø176Ø          JR      NZ,ENDLINE      ;8" drives always running
2554 21AA28    Ø177Ø          LD      HL,DLY$         ;Display "dly="
```

```
2557 CD7227   01780          CALL    LINOUT
255A FDCB0356 01790          BIT     2,(IY+3)        ;Test off/on
255E 3E20     01800          LD      A,' '           ;1 sec if DELAY=ON
2560 0631     01810          LD      B,'1'
2562 2804     01820          JR      Z,$+6
2564 3E2E     01830          LD      A,'.'           ;0.5 sec if DELAY=OFF
2566 0635     01840          LD      B,'5'
2568 CD8227   01850          CALL    BYTOUT
256B 78       01860          LD      A,B
256C CD8227   01870          CALL    BYTOUT
256F 3E73     01880          LD      A,'s'           ;Indicate seconds
2571 CD8227   01890          CALL    BYTOUT
2574 CDA027   01900 ENDLINE  CALL    CKPAWS          ;Check pause of display
2577 C1       01910 POPDRV   POP     BC              ;Recover drive #
2578 0C       01920          INC     C               ;Bump to next drive
2579 79       01930          LD      A,C
257A FE08     01940          CP      8               ;Loop thru all 8
257C C22724   01950          JP      NZ,DEV1
257F          01960 DEND     EQU     $
              01970 ;
              01980 ;        Byte I/O devices
              01990 ;
257F 210000   02000 BPARM    LD      HL,$-$          ;Check B parameter
2582 7C       02010          LD      A,H
2583 B5       02020          OR      L
2584 CAC826   02030          JP      Z,BEND          ;Go if B=NO (default)
              02040 ;
              02050 ;        Display the device vectoring
              02060 ;
2587 114B49   02070          LD      DE,'IK'         ;Start of device tables
258A          02080          @@GTDCB
258A 3E52     00013          LD      A,82
258C EF       00014          RST     40
258D C29527   02090          JP      NZ,IOERR
2590 7E       02100 LOGDCB   LD      A,(HL)          ;Bypass this device if
2591 B7       02110          OR      A               ;  table shows spare
2592 CA4A26   02120          JP      Z,DVRB2
2595 11002A   02130          LD      DE,STRBUF       ;Pt to string buffer
2598 E5       02140          PUSH    HL              ;Save origin ptr
2599 CDB826   02150          CALL    MOVNAM          ;Move dev name -> strbuf
259C E1       02160          POP     HL              ;Rcvr org of table
259D E5       02170          PUSH    HL
259E CB5E     02180 LOGDCB1  BIT     3,(HL)          ;If NIL, don't show
25A0 201C     02190          JR      NZ,DVRADDR      ;  any routes
25A2 CB66     02200          BIT     4,(HL)          ;Is device routed?
25A4 2818     02210          JR      Z,DVRADDR       ;Bypass if not
              02220 ;
              02230 ;        This device is routed
              02240 ;
25A6 2C       02250 LOGRTE   INC     L               ;Pt to vector & get it
25A7 7E       02260          LD      A,(HL)
25A8 2C       02270          INC     L
25A9 66       02280          LD      H,(HL)
25AA 6F       02290          LD      L,A
25AB CB7E     02300          BIT     7,(HL)          ;Is the route to a file?
25AD C25426   02310          JP      NZ,RTEFCB       ;Jump if a file
25B0 E5       02320          PUSH    HL              ;Hang onto this vector
25B1 CD8226   02330          CALL    DCBDIR          ;Get device direction
25B4 CDB826   02340          CALL    MOVNAM          ;Move dev name -> strbuf
25B7 E1       02350          POP     HL              ;Rcvr org of routee
25B8 CB66     02360          BIT     4,(HL)          ;Is routee also routed?
```

```
25BA 20EA     02370          JR      NZ,LOGRTE       ;Loop de loop if yes
25BC 187F     02380          JR      DVRB1           ;  else go display the line
              02390 ;
              02400 ;        Device has no routes - show its driver address
              02410 ;
25BE CD8226   02420 DVRADDR  CALL    DCBDIR          ;Get device direction
25C1 CB5E     02430          BIT     3,(HL)          ;Is this a NIL device
25C3 C27826   02440          JP      NZ,MOVNIL       ;No address if NIL
              02450 ;
              02460 ;        If linked, show device name of link
              02470 ;
25C6 CB6E     02480          BIT     5,(HL)          ;Any link DCB?
25C8 2824     02490          JR      Z,DVRA0         ;Go if none
25CA 2C       02500          INC     L               ;Get address of link DCB
25CB 7E       02510          LD      A,(HL)
25CC 2C       02520          INC     L
25CD 66       02530          LD      H,(HL)
25CE 6F       02540          LD      L,A
              02550 ;
              02560 ;        Now move in the name of the linked DCB
              02570 ;
25CF E5       02580          PUSH    HL
25D0 E5       02590          PUSH    HL
25D1 CDB826   02600          CALL    MOVNAM          ;Move name of LINK DCB
25D4 3E7C     02610          LD      A,'|'           ;Get separator for display and
25D6 12       02620          LD      (DE),A          ;  put in the buffer
25D7 13       02630          INC     DE
25D8 FDE1     02640          POP     IY              ;Pop address to IY
25DA FD6E04   02650          LD      L,(IY+4)        ;P/u linked DCB address
25DD FD6605   02660          LD      H,(IY+5)
25E0 CDB826   02670          CALL    MOVNAM          ;Move name of linked DCB
25E3 E1       02680          POP     HL              ;Recover address
25E4 EB       02690          EX      DE,HL           ;Switch tempy, HL to
25E5 3620     02700          LD      (HL),' '        ;  display buffer
25E7 23       02710          INC     HL
25E8 3626     02720          LD      (HL),'&'        ;Show the link
25EA 23       02730          INC     HL
25EB EB       02740          EX      DE,HL           ;Back to normal
25EC 18B0     02750          JR      LOGDCB1         ;Go ck this one
              02760 ;
              02770 ;        If filtered, find the filter DCB
              02780 ;
25EE CB76     02790 DVRA0    BIT     6,(HL)          ;If filtered, recover the
25F0 2835     02800          JR      Z,DVRB0         ;  original data by
25F2 E5       02810          PUSH    HL              ;  swapping back the
25F3 3E5B     02820          LD      A,'['
25F5 12       02830          LD      (DE),A
25F6 13       02840          INC     DE
25F7 D5       02850          PUSH    DE
25F8 54       02860          LD      D,H
25F9 5D       02870          LD      E,L
25FA 2C       02880          INC     L               ;  1st three bytes with
25FB 7E       02890          LD      A,(HL)          ;  the FILTER DCB
25FC 2C       02900          INC     L
25FD 66       02910          LD      H,(HL)
25FE 6F       02920          LD      L,A
25FF 010400   02930          LD      BC,4            ;HL now points to the
2602 09       02940          ADD     HL,BC           ;  entry point. Get its
2603 4E       02950          LD      C,(HL)          ;  DCB address by peeking
2604 0C       02960          INC     C               ;  past the name field
2605 09       02970          ADD     HL,BC
```

```
2606 7E       02980          LD      A,(HL)          ;Get low-order
2607 23       02990          INC     HL
2608 66       03000          LD      H,(HL)          ;Get hi-order
2609 6F       03010          LD      L,A
260A E5       03020          PUSH    HL              ;If DCB is itself, then
260B ED52     03030          SBC     HL,DE           ;  bring in the "inactive
260D E1       03040          POP     HL
260E D1       03050          POP     DE              ;Recover string buf ptr
260F 200A     03060          JR      NZ,DVRA1
2611 21E327   03070          LD      HL,INACT$
2614 010800   03080          LD      BC,8
2617 EDB0     03090          LDIR
2619 1803     03100          JR      DVRA2
              03110 ;
261B CDB826   03120 DVRA1    CALL    MOVNAM          ;Move name of filter DCB
261E 3E5D     03130 DVRA2    LD      A,']'           ;Put dsp chars into buffer
2620 12       03140          LD      (DE),A
2621 13       03150          INC     DE
2622 3E20     03160          LD      A,' '
2624 12       03170          LD      (DE),A
2625 13       03180          INC     DE
2626 E1       03190          POP     HL              ;Recover orig DCB ptr
              03200 ;
              03210 ;        Routine to construct address "X'xxxx'"
              03220 ;
2627 3E58     03230 DVRB0    LD      A,'X'           ;Show address as
2629 12       03240          LD      (DE),A          ;  X'dddd'
262A 13       03250          INC     DE
262B 3E27     03260          LD      A,27H           ;Single quote
262D 12       03270          LD      (DE),A
262E 13       03280          INC     DE
262F 2C       03290          INC     L
2630 7E       03300          LD      A,(HL)          ;P/u lo-order vector
2631 2C       03310          INC     L
2632 66       03320          LD      H,(HL)          ;P/u hi-order vector
2633 6F       03330          LD      L,A             ;Put lo in place
2634 EB       03340          EX      DE,HL           ;Vector value to DE
2635          03350          @@HEX16                 ;Convert to hex digits
2635 3E63     00015          LD      A,99
2637 EF       00016          RST     40
2638 EB       03360          EX      DE,HL           ;Restore strbuf ptr to DE
2639 3E27     03370          LD      A,27H           ;Closing '
263B 12       03380          LD      (DE),A
263C 13       03390          INC     DE
263D 3E0D     03400 DVRB1    LD      A,CR
263F 12       03410          LD      (DE),A          ;Stuff end-of-line
2640 21002A   03420          LD      HL,STRBUF       ;Display the info
2643 CD7227   03430          CALL    LINOUT
2646 CDA527   03440          CALL    CKPAWS0         ;Ck with no CR
2649 E1       03450          POP     HL              ;Rcvr table org
264A 7D       03460 DVRB2    LD      A,L             ;Advance to next table
264B C608     03470 TABLEN   ADD     A,8
264D 6F       03480          LD      L,A
264E DAC826   03490          JP      C,SPARM         ;Exit if finished
2651 C39025   03500          JP      LOGDCB          ;  else loop
              03510 ;
              03520 ;        Device routed to a file - grab its filespec
              03530 ;
2654 E5       03540 RTEFCB   PUSH    HL              ;Save control block org
2655 21B428   03550          LD      HL,IO$          ;Show 2-way device
2658 010500   03560          LD      BC,5
```

```
265B EDB0      03570          LDIR
265D E1        03580          POP     HL
265E 7D        03590          LD      A,L             ;Pt to file route data
265F C606      03600          ADD     A,6             ;  by indexing into FCB
2661 6F        03610          LD      L,A
2662 8C        03620          ADC     A,H
2663 95        03630          SUB     L
2664 67        03640          LD      H,A             ;HL = FCB+6
2665 4E        03650          LD      C,(HL)          ;P/u drive #
2666 23        03660          INC     HL
2667 46        03670          LD      B,(HL)          ;P/u DEC
2668 D5        03680          PUSH    DE
2669           03690          @@FNAME                 ;Fetch filename
2669 3E50      00017          LD      A,80
266B EF        00018          RST     40
266C D1        03700          POP     DE
266D C29527    03710          JP      NZ,IOERR
2670 1A        03720 RTEF1    LD      A,(DE)          ;Find end of filename
2671 FE03      03730          CP      3
2673 28C8      03740          JR      Z,DVRB1         ;Exit on ETX to put CR
2675 13        03750          INC     DE
2676 18F8      03760          JR      RTEF1
               03770 ;
               03780 ;        Move in 'NIL' as driver address
               03790 ;
2678 21B128    03800 MOVNIL   LD      HL,NIL$         ;Move in NIL
267B 010300    03810          LD      BC,3
267E EDB0      03820          LDIR
2680 18BB      03830          JR      DVRB1
               03840 ;
               03850 ;        Routine to denote i/o direction
               03860 ;
2682 3E20      03870 DCBDIR   LD      A,' '           ;1st need a space
2684 12        03880          LD      (DE),A
2685 13        03890          INC     DE
2686 CB46      03900          BIT     0,(HL)          ;Test if input device
2688 2802      03910          JR      Z,DCBD1         ;Put another space if not
268A 3E3C      03920          LD      A,'<'           ;Else show input capable
268C 12        03930 DCBD1    LD      (DE),A
268D 13        03940          INC     DE
268E 3E3D      03950          LD      A,'='           ;Always need this
2690 CB76      03960          BIT     6,(HL)          ;If a filter, then
2692 2802      03970          JR      Z,$+4           ;  reset to '#'
2694 3E23      03980          LD      A,'#'
2696 12        03990          LD      (DE),A
2697 13        04000          INC     DE
2698 3E20      04010          LD      A,' '           ;Init a space
269A CB4E      04020          BIT     1,(HL)          ;Output device?
269C 2802      04030          JR      Z,DCBD2         ;Use space if not
269E 3E3E      04040          LD      A,'>'           ;Else show output capable
26A0 12        04050 DCBD2    LD      (DE),A
26A1 13        04060          INC     DE
26A2 3E20      04070          LD      A,' '           ;Close with a space
26A4 12        04080          LD      (DE),A
26A5 13        04090          INC     DE
26A6 C9        04100          RET
               04110 ;
               04120 ;        Convert HL to 3-place decimal & stuff into (DE)
               04130 ;
26A7 D5        04140 CVRTDEC  PUSH    DE              ;Save place
26A8 110029    04150          LD      DE,BUFFER
```

```
26AB            04160         @@HEXDEC                ;Convert to decimal ASCII
26AB 3E61       00019         LD      A,97
26AD EF         00020         RST     40
26AE 210229     04170         LD      HL,BUFFER+2     ;Skip leading spaces
26B1 D1         04180         POP     DE
26B2 010300     04190         LD      BC,3
26B5 EDB0       04200         LDIR
26B7 C9         04210         RET
                04220 ;
                04230 ;       Move device name into string buffer
                04240 ;
26B8 7D         04250 MOVNAM  LD      A,L             ;Pt to name field
26B9 C606       04260         ADD     A,6
26BB 6F         04270         LD      L,A
26BC 3E2A       04280         LD      A,'*'           ;Stuff * in string buf
26BE 12         04290         LD      (DE),A
26BF 13         04300         INC     DE              ;Bump ptr to next pos
26C0 7E         04310         LD      A,(HL)          ;P/u 1st char of name
26C1 12         04320         LD      (DE),A          ;  & stuff it
26C2 13         04330         INC     DE              ;Do the same for char 2
26C3 2C         04340         INC     L
26C4 7E         04350         LD      A,(HL)
26C5 12         04360         LD      (DE),A
26C6 13         04370         INC     DE
26C7 C9         04380         RET
26C8            04390 BEND    EQU     $
                04400 ;
                04410 ;       Show high memory device drivers
                04420 ;
26C8 21FFFF     04430 SPARM   LD      HL,-1           ;Check S parameter
26CB 7C         04440         LD      A,H
26CC B5         04450         OR      L
26CD CAD327     04460         JP      Z,EXIT          ;Exit if through
26D0 21EB27     04470         LD      HL,DVCHDR$      ;Display header
26D3 CD7227     04480         CALL    LINOUT
26D6            04490         @@FLAGS                 ;Get flag table pointer
26D6 3E65       00021         LD      A,101
26D8 EF         00022         RST     40
26D9 FD7E03     04500         LD      A,(IY+'D'-'A')  ;P/u device flag
26DC B7         04510         OR      A               ;Exit if none in use
26DD F5         04520         PUSH    AF              ;Save flag
26DE 282D       04530         JR      Z,SHOWFS        ;Go if nothing on
26E0 21F527     04540         LD      HL,DVCS$        ;Pt to word string
26E3 01FF08     04550         LD      BC,8<8!0FFH     ;Init for 8 flag bits
26E6 F1         04560 DOD1    POP     AF              ;Rcvr link
26E7 0F         04570         RRCA                    ;Test if active
26E8 F5         04580         PUSH    AF
26E9 301B       04590         JR      NC,DOD3         ;Bypass if inactive
26EB 0C         04600         INC     C               ;Do we do the comma?
26EC 3E2C       04610         LD      A,','           ;End of word, do comma
26EE C48227     04620         CALL    NZ,BYTOUT
26F1 3E20       04630         LD      A,' '           ;Start with a space
26F3 CD8227     04640         CALL    BYTOUT
26F6 7E         04650 DOD2    LD      A,(HL)          ;Display word until carry
26F7 23         04660         INC     HL
26F8 F5         04670         PUSH    AF
26F9 E67F       04680         AND     7FH             ;Strip possible carry
26FB CD8227     04690         CALL    BYTOUT          ;Display the char
26FE F1         04700         POP     AF
26FF 07         04710         RLCA                    ;Was carry set
2700 30F4       04720         JR      NC,DOD2         ;Loop if not
```

```
2702 10E2      04730          DJNZ    DOD1            ;Loop for 8 bits
2704 1807      04740          JR      SHOWFS          ;Exit the loop
2706 7E        04750 DOD3     LD      A,(HL)          ;Loop & ignore word
2707 23        04760          INC     HL
2708 07        04770          RLCA                    ;Carry set on last char
2709 30FB      04780          JR      NC,DOD3
270B 10D9      04790          DJNZ    DOD1            ;Loop for 8 bits
270D FDCB125E  04800 SHOWFS   BIT     3,(IY+'S'-'A')  ;Show FAST or SLOW
2711 2005      04810          JR      NZ,FAST
2713 212928    04820          LD      HL,SLOW$        ;Point to slow$
2716 1803      04830          JR      SHOWIT
2718 212228    04840 FAST     LD      HL,FAST$        ;Point to fast$
271B FD7E03    04850 SHOWIT   LD      A,(IY+'D'-'A')  ;Check if others shown
271E B7        04860          OR      A
271F 2001      04870          JR      NZ,COMAOK
2721 23        04880          INC     HL              ;Bypass comma
2722 CD7227    04890 COMAOK   CALL    LINOUT
               04900 ;
               04910 ;        Display system modules resident
               04920 ;
2725 F1        04930 DORES    POP     AF              ;Stack integrity
2726 CDA027    04940 NOTON    CALL    CKPAWS
2729 113028    04950          LD      DE,RES$         ;Check if driver resident
272C           04960          @@GTMOD                 ;  in memory
272C 3E53      00023          LD      A,83
272E EF        00024          RST     40
272F C2D327    04970          JP      NZ,EXIT         ;Done if nothing res'd
2732 210500    04980          LD      HL,5
2735 19        04990          ADD     HL,DE           ;Point to hi-order table
2736 E5        05000          PUSH    HL
2737 213728    05010          LD      HL,SYSRES$      ;Display header
273A CD7227    05020          CALL    LINOUT
273D E1        05030          POP     HL
273E 01FF10    05040          LD      BC,16<8!0FFH    ;Init for 16 modules
2741 7E        05050 DORES1   LD      A,(HL)          ;P/u a high-order vector
2742 23        05060          INC     HL              ;Bump pointer to next
2743 23        05070          INC     HL
2744 B7        05080          OR      A               ;Is this module resident?
2745 2824      05090          JR      Z,DORES3        ;Go if not
2747 0C        05100          INC     C
2748 3E2C      05110          LD      A,','           ;Need comma if 2nd
274A C48227    05120          CALL    NZ,BYTOUT
274D 3E20      05130          LD      A,' '           ;Start with a space
274F CD8227    05140          CALL    BYTOUT
2752 3E10      05150          LD      A,16
2754 90        05160          SUB     B               ;Calculate module #
2755 16FF      05170          LD      D,-1
2757 14        05180 DORES2   INC     D
2758 D60A      05190          SUB     10
275A 30FB      05200          JR      NC,DORES2
275C F5        05210          PUSH    AF              ;Save units place
275D 7A        05220          LD      A,D             ;Test tens place
275E C630      05230          ADD     A,'0'           ;  for non-zero
2760 FE30      05240          CP      '0'
2762 C48227    05250          CALL    NZ,BYTOUT       ;Output if non-zero
2765 F1        05260          POP     AF              ;Get units
2766 C63A      05270          ADD     A,'0'+10        ;Adjust to ASCII
2768 CD8227    05280          CALL    BYTOUT
276B 10D4      05290 DORES3   DJNZ    DORES1
276D CDA027    05300          CALL    CKPAWS          ;One last ck for CR
2770 1861      05310          JR      EXIT
```

```
                    05320 ;
                    05330 ;       Output display routines
                    05340 ;
2772                05350 LINOUT  @@DSPLY
                    00025         IFEQ    00H,1
                    00026         LD      HL,
                    00027         ENDIF
2772 3E0A           00028         LD      A,10
2774 EF             00029         RST     40
2775 201E           05360         JR      NZ,IOERR
2777 3A8A27         05370         LD      A,(PPARM+1)     ;Ck P-parm
277A B7             05380         OR      A
277B C8             05390         RET     Z
277C                05400         @@PRINT                 ;Also print if needed
                    00030         IFEQ    00H,1
                    00031         LD      HL,
                    00032         ENDIF
277C 3E0E           00033         LD      A,14
277E EF             00034         RST     40
277F 2014           05410         JR      NZ,IOERR
2781 C9             05420         RET
                    05430 ;
2782 C5             05440 BYTOUT  PUSH    BC
2783 4F             05450         LD      C,A
2784                05460         @@DSP                   ;Display it
2784 3E02           00035         LD      A,2
2786 EF             00036         RST     40
2787 200A           05470         JR      NZ,POPBC
2789 110000         05480 PPARM   LD      DE,0            ;P/u P-parm
278C 7B             05490         LD      A,E
278D B2             05500         OR      D
278E 2803           05510         JR      Z,POPBC
2790                05520         @@PRT                   ;Print chr if needed
2790 3E06           00037         LD      A,6
2792 EF             00038         RST     40
2793 C1             05530 POPBC   POP     BC
2794 C8             05540         RET     Z
2795 6F             05550 IOERR   LD      L,A             ;Save error code
2796 2600           05560         LD      H,0
2798 F6C0           05570         OR      0C0H            ;Abbrev & return
279A 4F             05580         LD      C,A
279B                05590         @@ERROR
279B 3E1A           00039         LD      A,26
279D EF             00040         RST     40
279E 182C           05600         JR      SAVESP
                    05610 ;
                    05620 ;       Routine to ck on pause or break
                    05630 ;
27A0 3E0D           05640 CKPAWS  LD      A,CR            ;End line first
27A2 CD8227         05650         CALL    BYTOUT
27A5                05660 CKPAWS0 @@FLAGS                 ;Get flag table pointer
27A5 3E65           00041         LD      A,101
27A7 EF             00042         RST     40
27A8 FD7E0A         05670         LD      A,(IY+'K'-'A')  ;P/u KFLAG
27AB CB47           05680         BIT     0,A             ;Check for break
27AD 2017           05690         JR      NZ,BREAK        ;  if so exit
27AF CB4F           05700         BIT     1,A             ;Check for pause
27B1 C8             05710         RET     Z               ;Ret if not
27B2                05720 CKPAW1  @@KEY                   ;Wait for key input
27B2 3E01           00043         LD      A,1
27B4 EF             00044         RST     40
```

```
27B5 FE60      05730          CP      60H
27B7 28F9      05740          JR      Z,CKPAW1        ;Loop on pause
27B9 FE80      05750          CP      80H             ;Abort on BREAK
27BB 2809      05760          JR      Z,BREAK
27BD FD7E0A    05770 RESKFL   LD      A,(IY+'K'-'A')  ;Reset Pause & Enter bits
27C0 E6F9      05780          AND     0F9H            ;
27C2 FD770A    05790          LD      (IY+'K'-'A'),A
27C5 C9        05800          RET
               05810 ;
               05820 ;        BREAK handler routine
               05830 ;
27C6 CDBD27    05840 BREAK    CALL    RESKFL
27C9 21FFFF    05850          LD      HL,-1
27CC 310000    05860 SAVESP   LD      SP,$-$          ;Restore the stack
27CF           05870          @@CKBRKC                ;Clear any <BREAK>
27CF 3E6A      00045          LD      A,106
27D1 EF        00046          RST     40
27D2 C9        05880          RET                     ;  and RETurn
27D3 210000    05890 EXIT     LD      HL,0            ;Init to no error
27D6 18F4      05900          JR      SAVESP          ;P/u stack & return
               05910 ;
               05920 ;        String area
               05930 ;
27D8 4E        05940 NOPACK$ DB       'No  Disk] ',3
     6F 20 20 44 69 73 6B 5D
     20 03
27E3 49        05950 INACT$   DB       'Inactive'
     6E 61 63 74 69 76 65
27EB 0A        05960 DVCHDR$ DB       LF,'Options:',3
     4F 70 74 69 6F 6E 73 3A
     03
27F5 53        05970 DVCS$    DB       'Spoole','r'!80H,'Typ','e'!80H
     70 6F 6F 6C 65 F2 54 79
     70 E5
2800 56        05980          DB       'Verif','y'!80H,'Smoot','h'!80H
     65 72 69 66 F9 53 6D 6F
     6F 74 E8
280C 4D        05990          DB       'Memdis','k'!80H,'Form','s'!80H
     65 6D 64 69 73 EB 46 6F
     72 6D F3
2818 4B        06000          DB       'KS','M'!80H,'Graphi','c'!80H
     53 CD 47 72 61 70 68 69
     E3
2822 2C        06010 FAST$    DB       ', Fast',3
     20 46 61 73 74 03
2829 2C        06020 SLOW$    DB       ', Slow',3
     20 53 6C 6F 77 03
2830 53        06030 RES$     DB       'SYSRES',3
     59 53 52 45 53 03
2837 53        06040 SYSRES$ DB       'System modules resident:',3
     79 73 74 65 6D 20 6D 6F
     64 75 6C 65 73 20 72 65
     73 69 64 65 6E 74 3A 03
2850 20        06050 STPRAT$ DB       ' 6122030 3 61015'
     36 31 32 32 30 33 30 20
     33 20 36 31 30 31 35
2860 22        06060 FLOPY$   DB       '" Floppy #',3
     20 46 6C 6F 70 70 79 20
     23 03
286B 22        06070 RIGID$   DB       '" Rigid  #',3
     20 52 69 67 69 64 20 20
```

```
     23 Ø3
2876 2C          Ø6Ø8Ø CYLS$   DB      ', Cyls='
     2Ø 43 79 6C 73 3D
287D 2Ø          Ø6Ø9Ø COMMA$  DB      '   , ',3
     2Ø 2Ø 2C 2Ø Ø3
2883 52          Ø61ØØ REMOV$  DB      'Removable',3
     65 6D 6F 76 61 62 6C 65
     Ø3
288D 46          Ø611Ø FIXED$  DB      'Fixed',3
     69 78 65 64 Ø3
2893 64          Ø612Ø DEN$    DB      'den, Sides=',3
     65 6E 2C 2Ø 53 69 64 65
     73 3D Ø3
289F 2C          Ø613Ø STEP$   DB      ', Step=',3
     2Ø 53 74 65 7Ø 3D Ø3
28A7 6D          Ø614Ø MS$     DB      'ms',3
     73 Ø3
28AA 2C          Ø615Ø DLY$    DB      ', Dly=',3
     2Ø 44 6C 79 3D Ø3
28B1 4E          Ø616Ø NIL$    DB      'Nil'
     69 6C
28B4 2Ø          Ø617Ø IO$     DB      ' <=> '
     3C 3D 3E 2Ø
28B9             Ø618Ø PRMTBL$ EQU     $
ØØ8Ø             Ø619Ø VAL     EQU     8ØH
ØØ4Ø             Ø62ØØ SW      EQU     4ØH
ØØ2Ø             Ø621Ø STR     EQU     2ØH
ØØ1Ø             Ø622Ø SGL     EQU     1ØH
28B9 8Ø          Ø623Ø         DB      8ØH
28BA 56          Ø624Ø         DB      SW!SGL!6,'BYTEIO',Ø
     42 59 54 45 49 4F ØØ
28C2 8Ø25        Ø625Ø         DW      BPARM+1
28C4 56          Ø626Ø         DB      SW!SGL!6,'DRIVES',Ø
     44 52 49 56 45 53 ØØ
28CC 1E24        Ø627Ø         DW      DPARM+1
28CE 55          Ø628Ø         DB      SW!SGL!5,'PRINT',Ø
     5Ø 52 49 4E 54 ØØ
28D5 8A27        Ø629Ø         DW      PPARM+1
28D7 56          Ø63ØØ         DB      SW!SGL!6,'STATUS',Ø
     53 54 41 54 55 53 ØØ
28DF C926        Ø631Ø         DW      SPARM+1
28E1 56          Ø632Ø         DB      SW!SGL!6,'OPTION',Ø
     4F 5Ø 54 49 4F 4E ØØ
28E9 C926        Ø633Ø         DW      SPARM+1
28EB ØØ          Ø634Ø         NOP
                 Ø635Ø ;
2ØØØ             Ø636Ø         ORG     $<-8+1<8
Ø1ØØ             Ø637Ø BUFFER  DS      256
2AØØ             Ø638Ø STRBUF  EQU     $
                 Ø639Ø ;
24ØØ             Ø64ØØ         END     DEVICE
```

@@1	ØØØØ	@@2	ØØØØ	@@3	ØØØØ	@@4	ØØØØ
@MOD2	ØØØØ	@MOD4	FFFF	BEND	26C8	BPARM	257F
BREAK	27C6	BUFFER	29ØØ	BYTOUT	2782	CKPAW1	27B2
CKPAWS	27AØ	CKPAWSØ	27A5	COMAOK	2722	COMMA$	287D
CR	ØØØD	CVRTDEC	26A7	CYLS$	2876	DCBD1	268C
DCBD2	26AØ	DCBDIR	2682	DEN$	2893	DEND	257F
DEV1	2427	DEV2	245E	DEV2A	2469	DEV2B	248A
DEV3	24CF	DEV4	24E2	DEVICE	24ØØ	DEVICEA	24Ø9
DLY$	28AA	DOD1	26E6	DOD2	26F6	DOD3	27Ø6
DORES	2725	DORES1	2741	DORES2	2757	DORES3	276B
DPARM	241D	DVCHDR$	27EB	DVCS$	27F5	DVRAØ	25EE
DVRA1	261B	DVRA2	261E	DVRADDR	25BE	DVRBØ	2627
DVRB1	263D	DVRB2	264A	ENDLINE	2574	EXIT	27D3
FAST	2718	FAST$	2822	FIXED$	288D	FLOPPY	25Ø5
FLOPY$	286Ø	INACT$	27E3	IO$	28B4	IOERR	2795
LF	ØØØA	LINOUT	2772	LOGDCB	259Ø	LOGDCB1	259E
LOGRTE	25A6	MOVNAM	26B8	MOVNIL	2678	MS$	28A7
NIL$	28B1	NOPACK$	27D8	NOTON	2726	POPBC	2793
POPDRV	2577	PPARM	2789	PRMTBL$	28B9	REMOV$	2883
RES$	283Ø	RESKFL	27BD	RIGID$	286B	RTEF1	267Ø
RTEFCB	2654	SAVESP	27CC	SGL	ØØ1Ø	SHOWFS	27ØD
SHOWIT	271B	SLOW$	2829	SPARM	26C8	STEP$	289F
STPRAT$	285Ø	STR	ØØ2Ø	STRBUF	2AØØ	SW	ØØ4Ø
SYSRES$	2837	TABLEN	264B	VAL	ØØ8Ø	WPTEST	247B
@@ABORT	A46E	@@ADTSK	A5Ø1	@@BANK	AA19	@@BKSP	A6F9
@@BREAK	AA2F	@@CHNIO	A459	@@CKBRKC	AA7D	@@CKDRV	A555
@@CKEOF	A7ØE	@@CKTSK	A4EC	@@CLOSE	A6E4	@@CLS	AA67
@@CMNDI	A498	@@CMNDR	A4AD	@@CTL	A2BD	@@DATE	A42F
@@DCSTAT	A594	@@DEBUG	A4D7	@@DECHEX	A999	@@DIRRD	A9Ø6
@@DIRWR	A91B	@@DIV16	A984	@@DIV8	A96F	@@DODIR	A56A
@@DSP	A281	@@DSPLY	A321	@@ERROR	A4C2	@@EXIT	A483
@@FEXT	A873	@@FLAGS	AAØ3	@@FNAME	A888	@@FSPEC	A85E
@@GATRD	A8F1	@@GATWR	A93Ø	@@GET	A295	@@GTDCB	A8B2
@@GTDCT	A89D	@@GTMOD	A8C7	@@HDFMT	A63C	@@HEX16	A9D8
@@HEX8	A9C3	@@HEXDEC	A9AE	@@HIGH$	A9ED	@@INIT	A6BA
@@KBD	A2F9	@@KEY	A26D	@@KEYIN	A3ØD	@@KLTSK	A54Ø
@@LOAD	A834	@@LOC	A723	@@LOF	A738	@@LOGER	A358
@@LOGOT	A36D	@@MSG	A3A4	@@MUL16	A95A	@@MUL8	A945
@@OPEN	A6CF	@@PARAM	A41A	@@PAUSE	A4Ø5	@@PEOF	A74D
@@POSN	A762	@@PRINT	A3B9	@@PRT	A2D1	@@PUT	A2A9
@@RAMDIR	A57F	@@RDSEC	A612	@@RDSSC	A8DC	@@READ	A777
@@REMOV	A6A5	@@RENAM	A69Ø	@@REW	A78C	@@RMTSK	A516
@@RPTSK	A52B	@@RREAD	A7A1	@@RSLCT	A5FD	@@RSTOR	A5BE
@@RUN	A849	@@RWRIT	A7B6	@@SEEK	A5E8	@@SEEKSC	A7CB
@@SKIP	A7EØ	@@SLCT	A5A9	@@STEPI	A5D3	@@TIME	A444

```
@@VDCTL     A3FØ @@VER      A7F5 @@VRSEC    A627
@@WEOF      A8ØA @@WHERE    A2E5 @@WRITE    A81F
@@WRSEC     A651 @@WRSSC    A666 @@WRTRK    A67B

24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: DIR, CAT

Library: SYS6/SYS

ISAM # : 21H, 2ØH

```
                00100 ;LBDIR/ASM - DIR / CAT Command
0000            00110         TITLE   <DIR - LS-DOS 6.2>
                00120 ;
2400            00130         ORG     2400H
                00140 ;
2400 C3D92D     00150 ENTRY   JP      DIR             ;Go if DIR
                00160 ;
2403 E5         00170 CATBGN  PUSH    HL              ;Here if CAT
2404 210000     00180         LD      HL,0            ;Set the DIR (A
2407 220628     00190         LD      (APARM+1),HL    ;  parameter to OFF
240A E1         00200         POP     HL              ;  and do a DIR
240B 18F3       00210         JR      ENTRY           ;  command
                00220 ;
4296            00230 BLKHASH EQU     4296H           ;Hash code of blank password
                00240 ;
240D            00250 *GET SVCMAC:3                   ;Get SVC Macro equivalents and
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
240D            00260 *GET  VALUES:3                  ; other misc. equates
                03920 ;VALUES/ASM - Version 6
                03930 *LIST OFF
                04200 *LIST ON
                00270 ;
240D            00280 *GET LBDIRA:3
                04210 ;LBDIRA/ASM - DIR main processing loop
240D            04220         SUBTTL  '<LBDIRA - Mainline Program>'
```

LBDIRA - Mainline Program

```
240D            04230           PAGE
                04240 ;
                04250 ;         error processing
                04260 ;
240D 217A2D     04270 NOMEM     LD      HL,NOMEM$
2410 DD         04280           DB      0DDH
2411 219E2D     04290 BADFMT    LD      HL,BADFMT$
2414            04300           @@LOGOT
                00001           IFEQ    00H,1
                00002           LD      HL,
                00003           ENDIF
2414 3E0C       00004           LD      A,12
2416 EF         00005           RST     40
                04310 ;
2417 21FFFF     04320 ABORT     LD      HL,-1           ;Set HL = -1
241A 1810       04330           JR      SAVESP          ;Abort
                04340 ;
                04350 ;         I/O Error Routine
                04360 ;
241C 3E20       04370 ERR32     LD      A,32            ;"Illegal Drive Number"
241E 6F         04380 IOERR     LD      L,A             ;Set HL = Error #
241F 2600       04390           LD      H,0
2421 F6C0       04400           OR      0C0H            ;Set short error
2423 4F         04410           LD      C,A             ;Stuff in C
2424            04420           @@ERROR                 ;Display error
2424 3E1A       00006           LD      A,26
2426 EF         00007           RST     40
2427 1803       04430           JR      SAVESP          ;Abort
                04440 ;
                04450 ;         Clear stack & Exit
                04460 ;
2429 210000     04470 EXIT      LD      HL,0            ;Good exit
242C 310000     04480 SAVESP    LD      SP,$-$          ;P/u old SP address
242F            04490 ABORT3    @@CKBRKC                ;Clear break
242F 3E6A       00008           LD      A,106
2431 EF         00009           RST     40
2432 C9         04500           RET                     ;Go home now
                04510 ;
                04520 ;         Init to 4 files/line & Drive # in string
                04530 ;
2433 C5         04540 DIR4      PUSH    BC              ;Save drive #
2434 3E04       04550           LD      A,4             ;4 filespecs/line
2436 320E28     04560           LD      (DONAM9+1),A    ;Save
2439 79         04570           LD      A,C             ;P/u drive #
243A C630       04580           ADD     A,'0'           ;Convert to ASCII
243C 320E2C     04590           LD      (DRIVE),A       ;  & stuff in message
243F 32222D     04600           LD      (NDRIVE),A      ;Also stuff in No Disk
                04610 ;
                04620 ;         Is the starting Drive available ?
                04630 ;
2442            04640           @@GTDCT                 ;IY => DCT+0
2442 3E51       00010           LD      A,81
2444 EF         00011           RST     40
2445            04650           @@CKDRV                 ;Drive alive ?
2445 3E21       00012           LD      A,33
2447 EF         00013           RST     40
2448 F5         04660           PUSH    AF              ;Save RETurn condition
2449 CD932A     04670           CALL    CKPAWS          ;<BREAK> hit ?
244C F1         04680           POP     AF              ;NZ - couldn't log drive
244D 2826       04690           JR      Z,GDCKDRV       ;Z - Logged drive succ
```

LBDIRA - Mainline Program

```
                04700 ;
                04710 ;       Is this Drive enabled ?
                04720 ;
244F FD7EØØ     04730         LD      A,(IY)          ;P/u Enable/Disable byte
2452 FEC3       04740         CP      ØC3H            ;Enabled ?
2454 28Ø9       04750         JR      Z,NO_DISK       ;Yes - display No Disk
                04760 ;
                04770 ;       If this is not global - Illegal Drive #
                04780 ;
2456 3A3626     04790         LD      A,(SPECIF+1)    ;Specific drive # ?
2459 B7         04800         OR      A               ;
245A CA1C24     04810         JP      Z,ERR32         ;Yes - illegal drive #
245D 1813       04820         JR      NEXTDRV         ;No - get next drive
                04830 ;
                04840 ;       Enabled Drive - Display "No Disk" string
                04850 ;
245F 32882A     04860 NO_DISK LD      (NOTITLE+1),A   ;Turn off title
2462 CD652A     04870         CALL    CKPAGE          ;Check for scroll
2465 211B2D     04880         LD      HL,NODISK       ;HL => "No Disk" string
2468 CDB929     04890         CALL    LINOUT          ;Display line
246B CD652A     04900         CALL    CKPAGE          ;Check for scroll
246E AF         04910         XOR     A               ;Turn on Title
246F 32882A     04920         LD      (NOTITLE+1),A   ;
                04930 ;
2472 C33426     04940 NEXTDRV JP      CKHIT4          ;Get next drive
                04950 ;
                04960 ;       Calculate quantity of Sectors/Gran
                04970 ;
2475 C5         04980 GDCKDRV PUSH    BC              ;Save Drive #
2476 FD7EØ8     04990         LD      A,(IY+8)        ;P/u # Sectors/Gran
2479 E61F       05000         AND     1FH             ;Mask off junk
247B 3C         05010         INC     A               ;Bump for zero offset
247C 32ED29     05020         LD      (CALCK1+1),A    ;Stuff it
                05030 ;
                05040 ;       P/u # Cylinders from DCT & stuff in string
                05050 ;
247F FD6EØ6     05060         LD      L,(IY+6)        ;P/u cyl count
2482 2C         05070         INC     L               ;Offset from Ø
2483 26ØØ       05080         LD      H,Ø             ;Stuff in HL
2485 111B2C     05090         LD      DE,CYLCNT       ;DE => Destination
2488 3E2Ø       05100         LD      A,' '           ;Leading space
248A CD382A     05110         CALL    CVD1ØØ          ;Stuff in string
                05120 ;
                05130 ;       Create "DDEN" String or "HARD" string
                05140 ;
248D 11242C     05150         LD      DE,DENSITY      ;Destination
2490 21FF2B     05160         LD      HL,DEN
2493 3E44       05170         LD      A,'D'
2495 FDCBØ376   05180         BIT     6,(IY+3)        ;Ck density
2499 2ØØ2       05190         JR      NZ,DUBDEN
249B 3E53       05200         LD      A,'S'
249D 77         05210 DUBDEN  LD      (HL),A
249E Ø1Ø4ØØ     05220         LD      BC,4            ;4 chars to Xfer
24A1 FDCBØ35E   05230         BIT     3,(IY+3)        ;Hard Drive ?
24A5 28Ø3       05240         JR      Z,DOLDIR
24A7 21Ø32C     05250         LD      HL,HARD         ;HL => "HARD"
24AA EDBØ       05260 DOLDIR  LDIR                    ;Xfer string
                05270 ;
                05280 ;       Drive logged in - Read in GAT
```

LBDIRA - Mainline Program

```
                05290 ;
24AC C1         05300           POP     BC              ;Recover Drive #
24AD 21002E     05310           LD      HL,GAT          ;HL => GAT buffer
24B0 FD5609     05320           LD      D,(IY+9)        ;D = Directory Cyl
24B3 1E00       05330           LD      E,0             ;E = Gat Sector
24B5            05340           @@RDSSC                 ;Read Sector
24B5 3E55       00014           LD      A,85
24B7 EF         00015           RST     40
24B8 3E14       05350           LD      A,20            ;Init "GAT Read Error"
24BA C21E24     05360           JP      NZ,IOERR
24BD CD932A     05370           CALL    CKPAWS          ;<BREAK> hit ?
                05380 ;
                05390 ;
                05400 ;         Calculate the FREE space on the disk
                05410 ;
                05420 ;
24C0 110000     05430           LD      DE,0            ;DE = Gran count
24C3 2ECC       05440           LD      L,0CCH          ;HL => GAT + X'CC'
24C5 7E         05450           LD      A,(HL)          ;P/u excess cyl byte
24C6 C623       05460           ADD     A,35            ;Cyl excess of 35
24C8 47         05470           LD      B,A             ;Set loop counter
24C9 6A         05480           LD      L,D             ;HL => GAT + X'00'
24CA C5         05490           PUSH    BC              ;Save cyl count in B
                05500 ;
                05510 ;         HL => GAT, B = # of cyls, DE = Gran count
                05520 ;
24CB 7E         05530 FS1       LD      A,(HL)          ;P/u a GAT byte & set
24CC 37         05540 FS2       SCF                     ;Carry so bit 7 stays 1
                05550 ;
                05560 ;         Is the granule in use ?
                05570 ;
24CD 1F         05580           RRA                     ;Shift gran bit -> carry
24CE 3801       05590           JR      C,FS3           ;Don't inc if in use
                05600 ;
                05610 ;         Free Granule - bump Free Granule Count
                05620 ;
24D0 13         05630           INC     DE              ;Another spare gran
24D1 FEFF       05640 FS3       CP      0FFH            ;Fin with this GAT byte?
24D3 20F7       05650           JR      NZ,FS2          ;Loop if not
                05660 ;
                05670 ;         Finished with GAT byte, advance to next
                05680 ;
24D5 2C         05690           INC     L               ;Advance to next byte
24D6 10F3       05700           DJNZ    FS1             ;B cylinders to check
                05710 ;
                05720 ;         DE = Free Grans, Calculate # Grans/cyl
                05730 ;
24D8 C1         05740           POP     BC              ;B = # of cylinders
24D9 FD7E08     05750           LD      A,(IY+8)        ;P/u DCT+8
24DC 07         05760           RLCA                    ;Move Grans/Cyl into
24DD 07         05770           RLCA                    ;Bits 0-2
24DE 07         05780           RLCA
24DF E607       05790           AND     7
24E1 3C         05800           INC     A               ;A = Grans/Cylinder
24E2 FDCB046E   05810           BIT     5,(IY+4)        ;Double-bit set ?
24E6 2801       05820           JR      Z,NOTDUB        ;No - don't double
24E8 87         05830           ADD     A,A             ;Double grans/cylinder
                05840 ;
                05850 ;         A = # Grans/Cyl, Calculate Total # of Grans
```

LBDIRA - Mainline Program

```
                    05860 ;
24E9 210000         05870 NOTDUB  LD      HL,0            ;Init HL = 0
24EC D5             05880         PUSH    DE              ;Save Free Grans
24ED 54             05890         LD      D,H             ;Set DE = # cyls
24EE 58             05900         LD      E,B
24EF 47             05910         LD      B,A             ;B = Grans/Cyl
                    05920 ;
                    05930 ;       Multiply Grans/Cyl (B) x # Cyls (DE)
                    05940 ;
24F0 19             05950 GPCLOOP ADD     HL,DE           ;Add cylinder count
24F1 10FD           05960         DJNZ    GPCLOOP         ;Grans/cyl times
                    05970 ;
                    05980 ;       HL = # of grans/disk, Is this a hard drive ?
                    05990 ;
24F3 FDCB035E       06000         BIT     3,(IY+3)        ;Hard Drive ?
24F7 201E           06010         JR      NZ,SKIPLOC      ;Yes-don't check lockout
                    06020 ;
                    06030 ;       Floppy disk - check for locked out cylinders
                    06040 ;
24F9 43             06050         LD      B,E             ;B = cylinder count
24FA EB             06060         EX      DE,HL           ;Save total cnt in DE
24FB 21602E         06070         LD      HL,GAT+60H      ;HL => Lockout table
24FE 0E00           06080         LD      C,0             ;C = Locked out cyl count
2500 F5             06090         PUSH    AF              ;Save Grans/Cyl in A
                    06100 ;
                    06110 ;       Loop to count up Locked out cylinders in C
                    06120 ;
2501 3E01           06130 LKLOOP  LD      A,1             ;Init cyl checker
2503 A6             06140         AND     (HL)            ;Locked out ?
2504 2801           06150         JR      Z,GOODCYL       ;No - good cylinder
2506 0C             06160         INC     C               ;Bump locked out count
2507 2C             06170 GOODCYL INC     L               ;Bump ptr
2508 10F7           06180         DJNZ    LKLOOP          ;B cylinders
                    06190 ;
                    06200 ;       Multiply Cylinders (BC) x Grans/Cyl
                    06210 ;
250A F1             06220         POP     AF              ;A = Grans/Cyl
250B F5             06230         PUSH    AF              ;Save it
250C 60             06240         LD      H,B             ;Init HL = 0
250D 68             06250         LD      L,B
                    06260 ;
250E 09             06270 GTUSED  ADD     HL,BC           ;Add cylinder count
250F 3D             06280         DEC     A               ;Grans/cyl times
2510 20FC           06290         JR      NZ,GTUSED
2512 F1             06300         POP     AF              ;A = Grans/Cyl
                    06310 ;
                    06320 ;       Subtract # of Grans locked out from total
                    06330 ;
2513 B7             06340         OR      A               ;Clear carry
2514 EB             06350         EX      DE,HL
2515 ED52           06360         SBC     HL,DE           ;HL = Grans possible
2517 D1             06370 SKIPLOC POP     DE              ;Rcvr # of Free Grans
                    06380 ;
                    06390 ;       HL = # Grans possible, DE = # Grans Free
                    06400 ;
2518 E5             06410         PUSH    HL              ;Save Grans used
2519 21302C         06420         LD      HL,KFREE        ;Convert Grans Free
251C CDE829         06430         CALL    CALCK           ;  to ASCII K & stuff
251F D1             06440         POP     DE              ;  into string.
```

LBDIRA - Mainline Program

```
                    06450 ;
                    06460 ;       Calculate # of K used & stuff into header
                    06470 ;
2520 213C2C         06480         LD      HL,KPOSS        ;Pt to where to stuff
2523 CDE829         06490         CALL    CALCK           ;Calculate K & stuff
                    06500 ;
                    06510 ;       Transfer Diskette Name into string buffer
                    06520 ;
2526 21D02E         06530         LD      HL,GAT+0D0H     ;HL => Diskette Name
2529 11112C         06540         LD      DE,NAME         ;Move pack name -> header
252C 0E08           06550         LD      C,8             ;BC = 8 chars to xfer
252E EDB0           06560         LDIR                    ;Xfer into buff
                    06570 ;
                    06580 ;       Clear out Date buffer
                    06590 ;
2530 114D2C         06600         LD      DE,DATBUF       ;DE => Start of buffer
2533 3E20           06610         LD      A,' '           ;Space
2535 0609           06620         LD      B,9             ;9 chars to clear
2537 12             06630 CLRLP   LD      (DE),A          ;Stuff in space
2538 13             06640         INC     DE              ;Bump
2539 10FC           06650         DJNZ    CLRLP
                    06660 ;
                    06670 ;       HL => Date in mm/dd/yy format - p/u month
                    06680 ;
253B 7E             06690         LD      A,(HL)          ;P/u month
253C D630           06700         SUB     '0'             ;Convert tens to binary
253E 4F             06710         LD      C,A             ;Save in C
                    06720 ;
                    06730 ;       Multiply first digit of month x 10
                    06740 ;
253F 87             06750         ADD     A,A             ;X 2
2540 87             06760         ADD     A,A             ;X 4
2541 81             06770         ADD     A,C             ;X 5
2542 87             06780         ADD     A,A             ;X 10
2543 4F             06790         LD      C,A             ;Stuff in C
                    06800 ;
                    06810 ;       Pick up second digit of month & add to 10's
                    06820 ;
2544 23             06830         INC     HL              ;Bump to ones
2545 7E             06840         LD      A,(HL)          ;P/u ones of month
2546 D630           06850         SUB     '0'             ;Convert to binary
2548 81             06860         ADD     A,C             ;A = Month (1-12)
2549 282E           06870         JR      Z,ILLDATE       ;Abort if NO DATE
254B FE0D           06880         CP      13              ;Legal Month ?
254D 302A           06890         JR      NC,ILLDATE      ;No - illegal date
                    06900 ;
                    06910 ;       Legal Month - Mult x 3 & pt to month string
                    06920 ;
254F 4F             06930         LD      C,A             ;Xfer month to C
2550 87             06940         ADD     A,A             ;X 2
2551 81             06950         ADD     A,C             ;X 3
2552 4F             06960         LD      C,A             ;BC = offset
2553 E5             06970         PUSH    HL              ;Save date pointer
2554 21AB2D         06980         LD      HL,MONTBL-3     ;HL => Month String table
2557 09             06990         ADD     HL,BC           ;HL => Month String
                    07000 ;
                    07010 ;       HL => Month String, Stuff into Buffer
                    07020 ;
2558 11502C         07030         LD      DE,DATBUF+3     ;DE => Destination
```

LBDIRA - Mainline Program

```
255B ØEØ3     Ø7Ø4Ø          LD      C,3             ;BC = 3 chars to xfer
255D EDBØ     Ø7Ø5Ø          LDIR                    ;Xfer date to buffer
              Ø7Ø6Ø ;
              Ø7Ø7Ø ;        Transfer Day (ØØ-31) into date buffer
              Ø7Ø8Ø ;
255F E1       Ø7Ø9Ø          POP     HL              ;Recover ptr
256Ø 23       Ø71ØØ          INC     HL              ;Bump
2561 23       Ø711Ø          INC     HL              ;HL => Day of month
2562 114D2C   Ø712Ø          LD      DE,DATBUF       ;DE => date buffer
2565 ØE Ø2    Ø713Ø          LD      C,2             ;Xfer into buffer
2567 EDBØ     Ø714Ø          LDIR
              Ø715Ø ;
              Ø716Ø ;        Transfer Year into buffer
              Ø717Ø ;
2569 23       Ø718Ø          INC     HL              ;HL => Year (8Ø-87)
256A ØE Ø2    Ø719Ø          LD      C,2             ;2 chars to xfer
256C 11542C   Ø72ØØ          LD      DE,DATBUF+7     ;DE => Destination
256F EDBØ     Ø721Ø          LDIR                    ;Xfer into buffer
              Ø722Ø ;
              Ø723Ø ;        Stuff "-" after day & month
              Ø724Ø ;
2571 3E2D     Ø725Ø          LD      A,'-'           ;Separator
2573 324F2C   Ø726Ø          LD      (DATBUF+2),A
2576 32532C   Ø727Ø          LD      (DATBUF+6),A
              Ø728Ø ;
              Ø729Ø ;        Display the files in the directory
              Ø73ØØ ;        Init DIR rec ptr = mem start, count = Ø
              Ø731Ø ;
2579 3C       Ø732Ø ILLDATE  INC     A               ;Set flag
257A 32D82D   Ø733Ø          LD      (FILFLAG),A     ;Set file alr disp flag
257D 21ØØ3Ø   Ø734Ø          LD      HL,MEMORY       ;Init DIRPTR to start
258Ø 22D92D   Ø735Ø          LD      (DIRPTR),HL     ;  of available memory
2583 AF       Ø736Ø          XOR     A               ;Set File display
2584 ED62     Ø737Ø          SBC     HL,HL           ;Set HL = Ø
2586 22F725   Ø738Ø          LD      (TFILES+1),HL   ;Total Files = Ø
2589 22D625   Ø739Ø          LD      (COUNT+1),HL    ;Count = Ø
258C 22Ø226   Ø74ØØ          LD      (TOTGRNS+1),HL  ;Total Grans = Ø
              Ø741Ø ;
              Ø742Ø ;        Read in the HIT of the disk
              Ø743Ø ;
258F C1       Ø744Ø          POP     BC              ;Recover Drive # in C
259Ø FD56Ø9   Ø745Ø          LD      D,(IY+9)        ;P/u directory cylinder
2593 1EØ1     Ø746Ø          LD      E,1             ;Pt to HIT sector
2595 21ØØ2E   Ø747Ø          LD      HL,HIT          ;HL => I/O buffer
2598          Ø748Ø          @@RDSSC                 ;Read System Sector
2598 3E55     ØØØ16          LD      A,85
259A EF       ØØØ17          RST     4Ø
259B 3E16     Ø749Ø          LD      A,16H           ;"HIT read error"?
259D C21E24   Ø75ØØ          JP      NZ,IOERR        ;Jump if read error
25AØ CD932A   Ø751Ø          CALL    CKPAWS          ;<BREAK> hit ?
25A3 C34526   Ø752Ø $JPØ     JP      CKHIT5          ;Jump into middle of loop
              Ø753Ø ;
              Ø754Ø ;        Loop to Process HIT entries
              Ø755Ø ;
25A6 E1       Ø756Ø CKHIT    POP     HL
25A7 C1       Ø757Ø CKHIT1   POP     BC              ;Recover HIT pointer lo
              Ø758Ø ;
              Ø759Ø ;        Point HL => Last HIT entry
              Ø76ØØ ;
```

LBDIRA - Mainline Program

```
25A8 262E     Ø761Ø         LD      H,HIT<-8        ;Set H = hi byte of HIT
25AA 68       Ø762Ø         LD      L,B             ;HL => Last HIT entry
              Ø763Ø ;
              Ø764Ø ;       Position to next entry of the Record
              Ø765Ø ;
25AB 7D       Ø766Ø CKHIT2  LD      A,L             ;P/u current entry
25AC C62Ø     Ø767Ø         ADD     A,32            ;Add 32 (bytes/entry)
25AE 6F       Ø768Ø         LD      L,A             ;HL => Next entry
25AF 3ØF2     Ø769Ø         JR      NC,$JPØ         ;Go to next record ?
              Ø77ØØ ;
              Ø771Ø ;       Position to entry zero of next record
              Ø772Ø ;
25B1 2C       Ø773Ø         INC     L               ;Posn to next record
25B2 CB6D     Ø774Ø         BIT     5,L             ;Done with drive ?
25B4 28ED     Ø775Ø         JR      Z,$JPØ          ;No - process entry
              Ø776Ø ;
              Ø777Ø ;       Finished with drive - Sort data unless (O=N)
              Ø778Ø ;
25B6 3AF926   Ø779Ø         LD      A,(SORTPRM+1)   ;If sort requested,
25B9 B7       Ø78ØØ         OR      A               ;  then need to output
25BA C4DF2A   Ø781Ø         CALL    NZ,SORTIT       ;  the sorted data
              Ø782Ø ;
              Ø783Ø ;       Were there any files displayed ?
              Ø784Ø ;
25BD 2AD625   Ø785Ø         LD      HL,(COUNT+1)    ;P/u displayed file count
25CØ 7C       Ø786Ø         LD      A,H             ;Any entered ?
25C1 B5       Ø787Ø         OR      L
25C2 2ØØB     Ø788Ø         JR      NZ,FILES        ;Yes - dsp under if (A)
              Ø789Ø ;
              Ø79ØØ ;       Display Title & line feed
              Ø791Ø ;
25C4 21Ø72C   Ø792Ø         LD      HL,DSTRING      ;HL => Title
25C7 CDB929   Ø793Ø         CALL    LINOUT          ;Display title
25CA CD652A   Ø794Ø         CALL    CKPAGE          ;Check for scroll
25CD 184A     Ø795Ø         JR      NOTAP           ;Get next drive
              Ø796Ø ;
              Ø797Ø ;       Get next drive # if the A parm was specified
              Ø798Ø ;
25CF 3AØ628   Ø799Ø FILES   LD      A,(APARM+1)     ;Don't display if A
25D2 B7       Ø8ØØØ         OR      A
25D3 2844     Ø8Ø1Ø         JR      Z,NOTAP         ;Not A - Output C/R
              Ø8Ø2Ø ;
              Ø8Ø3Ø ;       Were there any files shown in directory ?
              Ø8Ø4Ø ;
25D5 21ØØØØ   Ø8Ø5Ø COUNT   LD      HL,$-$          ;P/u count
25D8 7C       Ø8Ø6Ø         LD      A,H             ;Any files shown ?
25D9 B5       Ø8Ø7Ø         OR      L
25DA 284E     Ø8Ø8Ø         JR      Z,TERMDRV       ;No - get next drive
              Ø8Ø9Ø ;
              Ø81ØØ ;       Display Line of equal signs "="
              Ø811Ø ;
25DC Ø64F     Ø812Ø         LD      B,79            ;Output 79 "="
25DE 3E3D     Ø813Ø D79EQ   LD      A,'='
25EØ CDCB29   Ø814Ø         CALL    BYTOUT          ;Output "="
25E3 1ØF9     Ø815Ø         DJNZ    D79EQ
              Ø816Ø ;
              Ø817Ø ;       End line & check for scroll
              Ø818Ø ;
25E5 3EØD     Ø819Ø         LD      A,CR            ;End line with C/R
```

LBDIRA - Mainline Program

```
25E7 CDCB29   08200          CALL    BYTOUT
25EA CD652A   08210          CALL    CKPAGE
              08220 ;
              08230 ;        Stuff # of files used into footer string
              08240 ;
25ED C5       08250          PUSH    BC              ;Save Drive #
25EE 11E92C   08260          LD      DE,FDISP        ;DE => Destination
25F1 3E20     08270          LD      A,' '           ;Stuff # of files
25F3 CD382A   08280          CALL    CVD100          ;  used in string.
              08290 ;
              08300 ;        Pick up # of used files & stuff in string
              08310 ;
25F6 210000   08320 TFILES   LD      HL,$-$          ;P/u total files used
25F9 11FA2C   08330          LD      DE,FUSED        ;DE => Destination
25FC 3E20     08340          LD      A,' '
25FE CD382A   08350          CALL    CVD100          ;Stuff in string
              08360 ;
              08370 ;        P/u Total # of Grans & stuff into string
              08380 ;
2601 110000   08390 TOTGRNS  LD      DE,$-$          ;P/u total # of Grans
2604 21102D   08400          LD      HL,SPUSED       ;HL => Destination
2607 CDE829   08410          CALL    CALCK           ;Stuff into string
260A C1       08420          POP     BC              ;C = drive #
              08430 ;
              08440 ;        Display Footer String
              08450 ;
260B 21DE2C   08460          LD      HL,FSTRING      ;HL => Files disp string
260E CDB929   08470          CALL    LINOUT          ;Display line
2611 CD652A   08480          CALL    CKPAGE          ;Check for title
2614 CD652A   08490          CALL    CKPAGE
2617 1811     08500          JR      TERMDRV         ;Get next drive
              08510 ;
              08520 ;        A parm not spec'd, was a header displayed ?
              08530 ;
2619 3AD82D   08540 NOTAP    LD      A,(FILFLAG)     ;Was a header displayed ?
261C B7       08550          OR      A
261D 200B     08560          JR      NZ,TERMDRV      ;No - get next drive
              08570 ;
              08580 ;        Output a C/R if a full line wasn't displayed
              08590 ;
261F 3A0E28   08600          LD      A,(DONAM9+1)    ;Full line ?
2622 FE04     08610          CP      4
2624 C41928   08620          CALL    NZ,ENDLINE      ;End line
2627 CD1928   08630          CALL    ENDLINE         ;Do a blank line
              08640 ;
              08650 ;        Position to next drive - or exit if finished
              08660 ;
262A 3E00     08670 TERMDRV  LD      A,$-$           ;P/u term drive
262C 0C       08680          INC     C               ;Bump current drive #
262D B9       08690          CP      C               ;Done ?
262E D23324   08700          JP      NC,DIR4         ;Loop if in range
2631 C32924   08710          JP      EXIT            ;Exit if NZ
              08720 ;
              08730 ;        Get next drive unless drivespec specified
              08740 ;
2634 C1       08750 CKHIT4   POP     BC              ;Get drive # in C
2635 3E00     08760 SPECIF   LD      A,$-$           ;P/u specific flag
2637 B7       08770          OR      A
2638 210000   08780          LD      HL,0            ;Init in case exit
```

LBDIRA - Mainline Program

```
263B C8       Ø879Ø          RET     Z               ;Not global
              Ø88ØØ ;
              Ø881Ø ;        Bump Drive number
              Ø882Ø ;
263C 3A2B26   Ø883Ø          LD      A,(TERMDRV+1)   ;P/u term drive #
263F ØC       Ø884Ø          INC     C               ;Bump
2640 B9       Ø885Ø          CP      C               ;Finished ?
2641 D23324   Ø886Ø          JP      NC,DIR4         ;Loop if more
2644 C9       Ø887Ø          RET                     ;  else return
              Ø888Ø ;
              Ø889Ø ;        Is the HIT entry in use ?
              Ø89ØØ ;
2645 7E       Ø891Ø CKHIT5   LD      A,(HL)          ;P/u HIT entry
2646 B7       Ø892Ø          OR      A               ;In use ?
2647 CAAB25   Ø893Ø          JP      Z,CKHIT2        ;No - get next entry
              Ø894Ø ;
              Ø895Ø ;        HIT entry in use - Point HL to that entry
              Ø896Ø ;
264A 45       Ø897Ø          LD      B,L             ;Save DEC in B
264B C5       Ø898Ø          PUSH    BC              ;  & to stack
264C 7D       Ø899Ø          LD      A,L             ;Point L to Entry posn
264D E6EØ     Ø9ØØØ          AND     ØEØH
264F 6F       Ø9Ø1Ø          LD      L,A
              Ø9Ø2Ø ;
              Ø9Ø3Ø ;        Do we need to Read in another sector ?
              Ø9Ø4Ø ;
2650 A8       Ø9Ø5Ø          XOR     B               ;Done with 8 entries ?
2651 FEFF     Ø9Ø6Ø CKHIT6   CP      ØFFH
2653 281Ø     Ø9Ø7Ø          JR      Z,CKDIR1        ;No - check out entry
              Ø9Ø8Ø ;
              Ø9Ø9Ø ;        Read in the next directory sector
              Ø91ØØ ;
2655 325226   Ø911Ø          LD      (CKHIT6+1),A    ;Stuff in last entry posn
2658          Ø912Ø          @@DIRRD                 ;  & read it into buffer
2658 3E57     ØØØ18          LD      A,87
265A EF       ØØØ19          RST     4Ø
265B C21E24   Ø913Ø          JP      NZ,IOERR        ;Jump on read error
265E 7C       Ø914Ø          LD      A,H             ;P/u high byte
265F 326626   Ø915Ø          LD      (CKDIR1+1),A    ;  and save
2662 328629   Ø916Ø          LD      (SBUFFER+1),A   ;  for later
              Ø917Ø ;
              Ø918Ø ;        Valid File (Alive & FPDE) ?
              Ø919Ø ;
2665 26ØØ     Ø92ØØ CKDIR1   LD      H,$-$           ;P/u high byte
2667 CB66     Ø921Ø          BIT     4,(HL)          ;Alive ?
2669 CAA725   Ø922Ø          JP      Z,CKHIT1        ;No - get next entry
266C CB7E     Ø923Ø          BIT     7,(HL)          ;FPDE ?
266E C2A725   Ø924Ø          JP      NZ,CKHIT1       ;No - get next entry
              Ø925Ø ;
              Ø926Ø ;        Alive FPDE - Bump Total File counter
              Ø927Ø ;
2671 E5       Ø928Ø          PUSH    HL              ;Save ptr
2672 2AF725   Ø929Ø          LD      HL,(TFILES+1)   ;HL => Total Files
2675 23       Ø93ØØ          INC     HL              ;Bump total files
2676 22F725   Ø931Ø          LD      (TFILES+1),HL
2679 E1       Ø932Ø          POP     HL
              Ø933Ø ;
              Ø934Ø ;        Is this a SYStem File ?
              Ø935Ø ;
```

LBDIRA - Mainline Program

```
267A CB76    Ø936Ø          BIT     6,(HL)          ;SYS file ?
267C 28ØA    Ø937Ø          JR      Z,CKDIR3        ;No - continue
             Ø938Ø ;
             Ø939Ø ;        SYS file - don't check unless S parm entered
             Ø94ØØ ;
267E 11ØØØØ  Ø941Ø SPARM    LD      DE,$-$          ;P/u S-parm
2681 7A      Ø942Ø          LD      A,D             ;Specified ?
2682 B3      Ø943Ø          OR      E
2683 CAA725  Ø944Ø          JP      Z,CKHIT1        ;No - don't check it
2686 18ØC    Ø945Ø          JR      CKMOD           ;Skip INV check
             Ø946Ø ;
             Ø947Ø ;        Non-SYS file - Is the file Visible ?
             Ø948Ø ;
2688 CB5E    Ø949Ø CKDIR3   BIT     3,(HL)          ;Visible ?
268A 28Ø8    Ø95ØØ          JR      Z,CKMOD         ;Yes - skip I check
             Ø951Ø ;
             Ø952Ø ;        File is invisible - was INV (I) specified ?
             Ø953Ø ;
268C 11ØØØØ  Ø954Ø IPARM    LD      DE,$-$          ;I-parm
268F 7A      Ø955Ø          LD      A,D             ;Ignore if I-parm not
2690 B3      Ø956Ø          OR      E               ;  entered as this file
2691 CAA725  Ø957Ø          JP      Z,CKHIT1        ;  is invisible
             Ø958Ø ;
             Ø959Ø ;        Was the MOD parm entered ?
             Ø96ØØ ;
2694 11ØØØØ  Ø961Ø CKMOD    LD      DE,$-$          ;P/u mod parm
2697 7A      Ø962Ø          LD      A,D             ;Was it entered ?
2698 B3      Ø963Ø          OR      E
2699 28Ø7    Ø964Ø          JR      Z,CKNAM         ;Go if MOD not entered
             Ø965Ø ;
             Ø966Ø ;        MOD parm entered - was this file modified ?
             Ø967Ø ;
269B 2C      Ø968Ø          INC     L               ;HL => DIR + 1
269C CB76    Ø969Ø          BIT     6,(HL)          ;Was the file modified ?
269E CAA725  Ø97ØØ          JP      Z,CKHIT1        ;No - get next entry
26A1 2D      Ø971Ø          DEC     L               ;Adjust back to start
             Ø972Ø ;
             Ø973Ø ;        Attributes match - check if filespec matches
             Ø974Ø ;
26A2 E5      Ø975Ø CKNAM    PUSH    HL              ;Save ptr to record
26A3 7D      Ø976Ø          LD      A,L             ;Pt to filename in dir
26A4 C6Ø5    Ø977Ø          ADD     A,5
26A6 6F      Ø978Ø          LD      L,A             ;HL => DIR filename
26A7 113A2D  Ø979Ø          LD      DE,BLANKS       ;DE => Partspec input
26AA ØEØB    Ø98ØØ          LD      B,11            ;Ck name/ext (11-chars)
             Ø981Ø ;
             Ø982Ø ;        Loop to check if partspec matches dir name
             Ø983Ø ;
26AC 1A      Ø984Ø CKNAM1   LD      A,(DE)          ;P/u partspec
26AD FE24    Ø985Ø          CP      '$'             ;Wild char?
26AF 28Ø7    Ø986Ø          JR      Z,CKNAM2        ;Yes - match
             Ø987Ø ;
             Ø988Ø ;        Does Directory char match partspec char ?
             Ø989Ø ;
26B1 BE      Ø99ØØ          CP      (HL)            ;Not global, char match?
26B2 2Ø4     Ø991Ø          JR      Z,CKNAM2        ;Ck more if match
             Ø992Ø ;
             Ø993Ø ;        Chars don't match - Dir char a space ?
             Ø994Ø ;
```

LBDIRA - Mainline Program

```
26B4 FE2Ø     Ø995Ø          CP      ' '             ;Blank = end of ck
26B6 2ØØB     Ø996Ø          JR      NZ,MFLG         ;If not blank, no match
              Ø997Ø ;
              Ø998Ø ;        Bump Dir ptr & Partspec ptr & continue loop
              Ø999Ø ;
26B8 23       1ØØØØ CKNAM2   INC     HL              ;Bump pointers
26B9 13       1ØØ1Ø          INC     DE
26BA 1ØFØ     1ØØ2Ø          DJNZ    CKNAM1          ;Loop for 11 chars
              1ØØ3Ø ;
              1ØØ4Ø ;        Entries Match - Was the "-" Exclude given ?
              1ØØ5Ø ;
26BC 3AC426   1ØØ6Ø          LD      A,(MFLG+1)      ;P/u flag
26BF FE2D     1ØØ7Ø          CP      '-'             ; - exclude given ?
26C1 18Ø3     1ØØ8Ø          JR      CK2HIT          ;Yes - get next entry
              1ØØ9Ø ;
              1Ø1ØØ ;        Entries Don't match - Was exclude given ?
              1Ø11Ø ;
26C3 3EØØ     1Ø12Ø MFLG     LD      A,$-$           ;P/u Exclude flag
26C5 B7       1Ø13Ø          OR      A               ;If no exclude given
26C6 CAA625   1Ø14Ø CK2HIT   JP      Z,CKHIT         ;  get next entry
              1Ø15Ø ;
              1Ø16Ø ;        Recover DIR+Ø pointer
              1Ø17Ø ;
26C9 E1       1Ø18Ø CKNAM2A  POP     HL              ;Rcvr ptr to DIR+Ø
26CA E5       1Ø19Ø          PUSH    HL              ;Save
              1Ø2ØØ ;
              1Ø21Ø ;        Unpack Date of Directory entry
              1Ø22Ø ;
26CB 23       1Ø23Ø          INC     HL              ;HL => DIR+1
26CC CD192A   1Ø24Ø          CALL    UNPACK          ;Unpack date
              1Ø25Ø ;
              1Ø26Ø ;        Use Dates before user-specified date ?
              1Ø27Ø ;
26CF 3AD72D   1Ø28Ø          LD      A,(FTFLG)       ;P/u From/To flag
26D2 Ø7       1Ø29Ø          RLCA                    ;Tst fm bit
26D3 3ØØF     1Ø3ØØ          JR      NC,CKNAM2B      ;No - check to
              1Ø31Ø ;
              1Ø32Ø ;        "FROM" flag set - does file have a date ?
              1Ø33Ø ;
26D5 7A       1Ø34Ø          LD      A,D             ;Ignore if no date
26D6 B3       1Ø35Ø          OR      E               ;  in DIR for file
26D7 CAA625   1Ø36Ø          JP      Z,CKHIT         ;No date - get next entry
              1Ø37Ø ;
              1Ø38Ø ;        Is the Specified date >= the file's date ?
              1Ø39Ø ;
26DA 2ADD2D   1Ø4ØØ          LD      HL,(FMPAKD)     ;P/u user date entry
26DD EB       1Ø41Ø          EX      DE,HL
26DE CD132A   1Ø42Ø          CALL    CPHLDE          ;Compare HL to DE
26E1 EB       1Ø43Ø          EX      DE,HL           ;File date < User date ?
26E2 3811     1Ø44Ø          JR      C,$JP1          ;Yes - get next entry
              1Ø45Ø ;
              1Ø46Ø ;        Use Dates after user-specified Date ?
              1Ø47Ø ;
26E4 3AD72D   1Ø48Ø CKNAM2B  LD      A,(FTFLG)       ;P/u FROM/TO flag
26E7 ØF       1Ø49Ø          RRCA                    ;Test TO bit
26E8 3ØØE     1Ø5ØØ          JR      NC,SORTPRM      ;Go if no TOPARM
              1Ø51Ø ;
              1Ø52Ø ;        "TO" Flag set - Does file have a date ?
              1Ø53Ø ;
```

LBDIRA - Mainline Program

```
26EA 7A       1Ø54Ø          LD      A,D             ;File have a valid date ?
26EB B3       1Ø55Ø          OR      E
26EC CAA625   1Ø56Ø          JP      Z,CKHIT         ;No - get next entry
              1Ø57Ø ;
              1Ø58Ø ;        File has a date - Is spec'd date less ?
              1Ø59Ø ;
26EF 2ADF2D   1Ø6ØØ          LD      HL,(TOPAKD)     ;P/u user's packed date
26F2 CD132A   1Ø61Ø          CALL    CPHLDE          ;User date < File date ?
26F5 DAA625   1Ø62Ø $JP1     JP      C,CKHIT         ;Yes - get next entry
              1Ø63Ø ;
              1Ø64Ø ;        Was the Sort Parameter turned off ?
              1Ø65Ø ;
26F8 11FFFF   1Ø66Ø SORTPRM  LD      DE,-1           ;P/u default parm
26FB E1       1Ø67Ø          POP     HL              ;HL => DIR+Ø
26FC 7A       1Ø68Ø          LD      A,D             ;Default to SORT=ON
26FD B3       1Ø69Ø          OR      E
26FE 282D     1Ø7ØØ          JR      Z,DODSP         ;Go display if no sort
              1Ø71Ø ;
              1Ø72Ø ;        SORT = ON --- Calculate allocation & extents
              1Ø73Ø ;
2700 E5       1Ø74Ø          PUSH    HL              ;Save DIR + Ø ptr
2701 CD5B29   1Ø75Ø          CALL    ALLØ9A          ;Calc alloc & extents
2704 E1       1Ø76Ø          POP     HL              ;Recover DIR+Ø ptr
              1Ø77Ø ;
              1Ø78Ø ;        Overwrite FPDE's 22-25 with # Grans & # exts
              1Ø79Ø ;
2705 E5       1Ø8ØØ          PUSH    HL              ;Point IX = DIR+22
2706 DDE1     1Ø81Ø          POP     IX
2708 DD7316   1Ø82Ø          LD      (IX+22),E       ;Stuff in # Grans
270B DD7217   1Ø83Ø          LD      (IX+23),D
270E DD7118   1Ø84Ø          LD      (IX+24),C       ;Stuff in # Extents
2711 DD7Ø19   1Ø85Ø          LD      (IX+25),B
              1Ø86Ø ;
              1Ø87Ø ;        Transfer Record into Memory For Sort
              1Ø88Ø ;
2714 ED5BD92D 1Ø89Ø          LD      DE,(DIRPTR)     ;P/u last used mem addr
2718 E5       1Ø9ØØ          PUSH    HL              ;Save current DIR ptr
2719 Ø12ØØØ   1Ø91Ø          LD      BC,32           ;Move record to buffer
271C EDBØ     1Ø92Ø          LDIR                    ;Xfer
271E ED53D92D 1Ø93Ø          LD      (DIRPTR),DE     ;Update the pointer
              1Ø94Ø ;
              1Ø95Ø ;        Is there an overflow of available memory ?
              1Ø96Ø ;
2722 2ADB2D   1Ø97Ø          LD      HL,(MAXMEM)     ;P/u approximate hi-mem
2725 ED52     1Ø98Ø          SBC     HL,DE           ;Did it overflow ?
2727 D2A625   1Ø99Ø          JP      NC,CKHIT        ;No - get next entry
272A C3ØD24   11ØØØ          JP      NOMEM           ;Insuf mem for sort buff
              11Ø1Ø ;
              11Ø2Ø ;        Display A Filename
              11Ø3Ø ;
272D CD3327   11Ø4Ø DODSP    CALL    MATCH           ;Display entry
2730 C3A725   11Ø5Ø          JP      CKHIT1          ;Loop to next DIR entry
2733          ØØ29Ø *GET LBDIRB:3
              11Ø6Ø ;LBDIRB/ASM - Display Filespec & attributes
2733          11Ø7Ø          SUBTTL  '<LBDIRC - File Attribute Output>'
```

LBDIRC - File Attribute Output

```
2733            11Ø8Ø          PAGE
                11Ø9Ø ;
                11ØØ ;         MATCH - Display a File's Name and Extension
                1111Ø ;
2733 E5         1112Ø MATCH    PUSH    HL              ;Save HIT posn
2734 21D625     1113Ø          LD      HL,COUNT+1      ;Bump file count
2737 34         1114Ø          INC     (HL)
                1115Ø ;
                1116Ø ;        Was the Drive Header Displayed ?
                1117Ø ;
2738 21D82D     1118Ø          LD      HL,FILFLAG      ;HL => File Header flag
273B AF         1119Ø          XOR     A               ;If (HL) is Non-Zero
273C BE         112ØØ          CP      (HL)            ;  then the header has not
273D 77         1121Ø          LD      (HL),A          ;  printed.
273E C4B12A     1122Ø          CALL    NZ,CKTITL       ;Display title if NZ
                1123Ø ;
                1124Ø ;        Position HL to Directory Entry Filename
                1125Ø ;
2741 E1         1126Ø ALRPRT   POP     HL              ;Recover DEC
2742 7D         1127Ø          LD      A,L             ;P/u DEC
2743 E6EØ       1128Ø          AND     ØEØH            ;Posn to entry
2745 C6Ø5       1129Ø          ADD     A,5             ;Pt to start of filename
2747 6F         113ØØ          LD      L,A             ;HL => Filename field
                1131Ø ;
                1132Ø ;        Init B=8 chars for filename, C=19 to col
                1133Ø ;
2748 ØE13       1134Ø          LD      C,19            ;Chars to next column
274A Ø6Ø8       1135Ø          LD      B,8             ;Filename
                1136Ø ;
                1137Ø ;        Loop to Output the Filename
                1138Ø ;
274C 7E         1139Ø DONAM1   LD      A,(HL)          ;P/u character
274D 23         114ØØ          INC     HL              ;Bump DIR ptr
274E FE2Ø       1141Ø          CP      ' '             ;Space ?
275Ø 28Ø7       1142Ø          JR      Z,DONAM2        ;Yes - done with filename
2752 CDCA29     1143Ø          CALL    BYTOUT2         ;No - output char
2755 1ØF5       1144Ø          DJNZ    DONAM1          ;Field loop
2757 18Ø4       1145Ø          JR      DONAM3          ;Bypass ext calculation
                1146Ø ;
                1147Ø ;        Filename has < 8 chars, Pt to extension
                1148Ø ;
2759 7D         1149Ø DONAM2   LD      A,L             ;P/u low byte
275A 8Ø         115ØØ          ADD     A,B             ;Add # of chars left
275B 3D         1151Ø          DEC     A               ;Back one
275C 6F         1152Ø          LD      L,A             ;HL => Extension
                1153Ø ;
                1154Ø ;        Does this file have an extension ?
                1155Ø ;
275D 7E         1156Ø DONAM3   LD      A,(HL)          ;P/u first char
275E FE2Ø       1157Ø          CP      ' '             ;Blank
276Ø 2812       1158Ø          JR      Z,DONAM5        ;Yes - no extension
                1159Ø ;
                116ØØ ;        Output a "/" & Set up for Extension loop
                1161Ø ;
2762 3E2F       1162Ø          LD      A,'/'           ;Display slash
2764 CDCA29     1163Ø          CALL    BYTOUT2
2767 Ø6Ø3       1164Ø          LD      B,3             ;3 chars max for EXT
                1165Ø ;
                1166Ø ;        Loop to output the extension
                1167Ø ;
```

LBDIRC - File Attribute Output

```
2769 7E       11680 DONAM4  LD      A,(HL)          ;P/u char
276A 23       11690         INC     HL              ;Bump ptr
276B FE20     11700         CP      ' '             ;Space ?
276D 2805     11710         JR      Z,DONAM5        ;Exit on 1st blank
276F CDCA29   11720         CALL    BYTOUT2         ;Else display the char
2772 10F5     11730         DJNZ    DONAM4          ;Loop 3 chars
              11740 ;
              11750 ;       Was the (A) parameter specified ?
              11760 ;
2774 3A0628   11770 DONAM5  LD      A,(APARM+1)     ;A parm specified ?
2777 B7       11780         OR      A
2778 2813     11790         JR      Z,DONAM5A       ;No - continue
              11800 ;
              11810 ;       (A) parameter specified - Tab to column 14
              11820 ;
277A 79       11830         LD      A,C             ;P/u chars left to col 20
277B D606     11840         SUB     6               ;Adjust to column 14
277D 47       11850         LD      B,A             ;Stuff into B for DJNZ
277E CDE029   11860         CALL    OUTSPC          ;Output B spaces
              11870 ;
              11880 ;       Output mod flag (if modified) & tab to 19
              11890 ;
2781 7D       11900         LD      A,L             ;Pt HL => DIR+Ø
2782 E6E0     11910         AND     0E0H
2784 6F       11920         LD      L,A
2785 CD3729   11930         CALL    OUTMOD          ;Output "+" if mod
2788 0603     11940         LD      B,3             ;Output 3 spaces
278A CDE029   11950         CALL    OUTSPC          ;Output B spaces
              11960 ;
              11970 ;       Display the File's Attributes
              11980 ;
278D 0601     11990 DONAM5A LD      B,1             ;Set B=1 space
278F CDE029   12000         CALL    OUTSPC          ;After filespec.
              12010 ;
              12020 ;       Point HL => DIR+Ø (Attributes)
              12030 ;
2792 7D       12040         LD      A,L             ;Pt to 1st byte of
2793 E6E0     12050         AND     0E0H            ;Directory record
2795 6F       12060         LD      L,A
              12070 ;
              12080 ;       Display "?" if File OPEN bit set
              12090 ;
2796 3E3F     12100         LD      A,'?'           ;"?" character
2798 23       12110         INC     HL              ;HL => DIR + 1
2799 CB6E     12120         BIT     5,(HL)          ;File Open ?
279B 2B       12130         DEC     HL              ;HL => DIR + Ø
279C C4CA29   12140         CALL    NZ,BYTOUT2      ;Yes - output byte
              12150 ;
              12160 ;       Display an "*" if this is a PDS file
              12170 ;
279F 46       12180         LD      B,(HL)          ;P/u attributes byte
27A0 3E2A     12190         LD      A,'*'           ;Init for PDS display
27A2 CB68     12200         BIT     5,B
27A4 C4CA29   12210         CALL    NZ,BYTOUT2      ;Display if PDS
              12220 ;
              12230 ;       Display an "S" if file is a SYS file
              12240 ;
27A7 CB70     12250         BIT     6,B             ;Is it a SYS file?
27A9 3E53     12260         LD      A,'S'
```

LBDIRC - File Attribute Output

```
27AB C4CA29   12270          CALL    NZ,BYTOUT2      ;Display S if so
              12280 ;
              12290 ;        Display an "I" if file is invisible
              12300 ;
27AE CB58     12310          BIT     3,B             ;Is it an INV file?
27BØ 3E49     12320          LD      A,'I'
27B2 C4CA29   12330          CALL    NZ,BYTOUT2      ;Display I if so
              12340 ;
              12350 ;        Point HL => Password Hash (DIR+16)
              12360 ;
27B5 E5       12370          PUSH    HL              ;Save ptr to 1st dir byte
27B6 7D       12380          LD      A,L             ;Pt to owner password
27B7 C61Ø     12390          ADD     A,16
27B9 6F       12400          LD      L,A             ;HL => DIR+16
              12410 ;
              12420 ;        Pick up Password in DE
              12430 ;
27BA 5E       12440          LD      E,(HL)          ;P/u in password in DE
27BB 2C       12450          INC     L
27BC 56       12460          LD      D,(HL)
              12470 ;
              12480 ;        Is this a password protected File ?
              12490 ;
27BD E5       12500          PUSH    HL              ;Save ptr to user psw
27BE 219642   12510          LD      HL,BLKHASH      ;Init to blanks hash
27C1 ED52     12520          SBC     HL,DE           ;Is password blanks?
27C3 E1       12530          POP     HL
27C4 2814     12540          JR      Z,DONAM6        ;Blanks - no "P"assword
              12550 ;
              12560 ;        Password - Display "P" if access <> ALL
              12570 ;
27C6 78       12580          LD      A,B             ;P/u attributes byte
27C7 E6Ø7     12590          AND     7               ;Get protection level
27C9 3E5Ø     12600          LD      A,'P'           ;Init for protected
27CB 2ØØF     12610          JR      NZ,DONAM7       ;Stuff the 'P' if prot
              12620 ;
              12630 ;        Access = ALL, Pick up USER password in DE
              12640 ;
27CD 2C       12650          INC     L               ;HL => User Password
27CE 5E       12660          LD      E,(HL)          ;P/u in DE
27CF 2C       12670          INC     L
27DØ 56       12680          LD      D,(HL)
              12690 ;
              12700 ;        Set A = "P" is User Password <> blank
              12710 ;
27D1 219642   12720          LD      HL,BLKHASH      ;Init to blanks hash
27D4 ED52     12730          SBC     HL,DE
27D6 3E5Ø     12740          LD      A,'P'           ;Init to Protected
27D8 2ØØ2     12750          JR      NZ,DONAM7       ;Stuff 'P' if acc<>blank
27DA 3E2Ø     12760 DONAM6   LD      A,' '           ;  else stuff a blank
              12770 ;
              12780 ;        Set Password flag if protected & display "P"
              12790 ;
27DC 322728   12800 DONAM7   LD      (ALLØ2+1),A     ;Stuff 'P' or blank
27DF FE2Ø     12810          CP      ' '             ;Space ?
27E1 C4CA29   12820          CALL    NZ,BYTOUT2      ;Display char if needed
27E4 E1       12830          POP     HL              ;HL => DIR+Ø
              12840 ;
              12850 ;        Display a "C" if the file was Created
```

LBDIRC - File Attribute Output

```
                    12860 ;
27E5 23             12870           INC     HL              ;HL => DIR+1
27E6 7E             12880           LD      A,(HL)          ;P/u attributes
27E7 2B             12890           DEC     HL              ;HL => DIR+Ø
27E8 Ø7             12900           RLCA                    ;Created ?
27E9 3E43           12910           LD      A,'C'           ;"C"reate character
27EB DCCA29         12920           CALL    C,BYTOUT2       ;Yes - output byte
                    12930 ;
                    12940 ;         Display Mod flag here if (A) not specified
                    12950 ;
27EE 3AØ628         12960           LD      A,(APARM+1)     ;P/u A-parm
27F1 B7             12970           OR      A
27F2 F5             12980           PUSH    AF              ;Save condition
27F3 CC3729         12990           CALL    Z,OUTMOD        ;Output mod flag if -A
27F6 F1             13000           POP     AF              ;NZ - (A) parm
                    13010 ;
                    13020 ;         If (A) parameter given - then tab to col 26
                    13030 ;
27F7 2804           13040           JR      Z,DONAM8        ;Not A - go to 2Ø
27F9 3E06           13050           LD      A,6             ;Add 6 to column #
27FB 81             13060           ADD     A,C
27FC 4F             13070           LD      C,A             ;C = # of spaces
                    13080 ;
                    13090 ;         Position to Next designated column
                    13100 ;
27FD 3E2Ø           13110 DONAM8    LD      A,' '           ;Write a space
27FF CDCB29         13120           CALL    BYTOUT          ;Output byte
2802 ØD             13130           DEC     C               ;Dec column counter
2803 2ØF8           13140           JR      NZ,DONAM8       ;Display trailing spaces
                    13150 ;
                    13160 ;         Display other things if (A) parm set
                    13170 ;
2805 11FFFF         13180 APARM     LD      DE,-1           ;P/u (A) parm
2808 7A             13190           LD      A,D             ;Specified ?
2809 B3             13200           OR      E
280A C42528         13210           CALL    NZ,ALLØ1        ;Full info if A-parm
                    13220 ;
                    13230 ;         Check for end of line
                    13240 ;
280D 3EØØ           13250 DONAM9    LD      A,Ø             ;Count down 4-across
280F 3D             13260           DEC     A
2810 32ØE28         13270           LD      (DONAM9+1),A    ;Update count
2813 CØ             13280           RET     NZ              ;Loop if more to go
2814 3E04           13290           LD      A,4             ;  else re-init to 4/line
2816 32ØE28         13300           LD      (DONAM9+1),A
                    13310 ;
                    13320 ;         Finished with one line - end with C/R
                    13330 ;
2819 3EØD           13340 ENDLINE   LD      A,CR            ;End line
281B CDCB29         13350           CALL    BYTOUT
281E CD652A         13360           CALL    CKPAGE          ;Check for page pause
2821 CD932A         13370           CALL    CKPAWS          ;Scan pause or break
2824 C9             13380           RET                     ;Loop
                    13390 ;
                    13400 ;         ALLØ1 - Display Full Allocation of a file
                    13410 ;
2825 E5             13420 ALLØ1     PUSH    HL              ;Save pointer to 1st byte
2826 3EØØ           13430 ALLØ2     LD      A,Ø             ;Bypass if not
2828 D62Ø           13440           SUB     2ØH             ;  password protected
```

LBDIRC - File Attribute Output

```
282A 28Ø3      13450          JR      Z,ALLØ3
282C 7E        13460          LD      A,(HL)          ;Get prot level &
282D E6Ø7      13470          AND     7               ;  multiply by 4
282F Ø7        13480 ALLØ3    RLCA                    ;  to index string array
2830 Ø7        13490          RLCA
2831 4F        13500          LD      C,A
2832 Ø6ØØ      13510          LD      B,Ø
2834 21462D    13520          LD      HL,PROTS$       ;Pt to 4-char abbrevs
2837 Ø9        13530          ADD     HL,BC           ;Pt to proper one
2838 11A72C    13540          LD      DE,PLEVEL       ;Move into output line
283B ØEØ4      13550          LD      C,4
283D EDBØ      13560          LDIR
283F E1        13570          POP     HL              ;Recover pointer to
2840 E5        13580          PUSH    HL              ;  1st byte of dir record
2841 2C        13590          INC     L
2842 2C        13600          INC     L
2843 2C        13610          INC     L
               13620 ;
               13630 ;        Pick up EOF offset byte & Stuff for later
               13640 ;
2844 7E        13650          LD      A,(HL)          ;P/u EOF offset byte
2845 329628    13660          LD      (EOFBYTE+1),A   ;Stuff into LD DE,$-$
               13670 ;
               13680 ;        calculate EOF record according to the formula:
               13690 ;        EOFREC= ((ERN-1)*256+EOF+LRL-1)/LRL if ERN<>Ø
               13700 ;        EOFREC= Ø if ERN=Ø
               13710 ;
2848 7E        13720          LD      A,(HL)          ;P/u EOF offset byte
2849 F5        13730          PUSH    AF              ;  & save it
284A 2C        13740          INC     L               ;Pt to LRL
284B 7E        13750          LD      A,(HL)          ;P/u LRL
284C 326A28    13760          LD      (ALLØ4+1),A     ;  & stuff it
               13770 ;
               13780 ;        get LRL into message
               13790 ;
284F E5        13800          PUSH    HL              ;Save ptr
2850 6F        13810          LD      L,A             ;Transfer LRL to HL
2851 2600      13820          LD      H,Ø
2853 B7        13830          OR      A               ;Test for <> 256
2854 2001      13840          JR      NZ,$+3
2856 24        13850          INC     H               ;Show 256
2857 11AE2C    13860          LD      DE,LRL          ;DE => LRL destination
285A 3E20      13870          LD      A,' '           ;Init the ASCII byte
285C CD382A    13880          CALL    CVD100          ;Cvrt to msg field
285F E1        13890          POP     HL
               13900 ;
               13910 ;        continue to calculate EOF
               13920 ;
2860 7D        13930          LD      A,L             ;Pt to ERN
2861 C610      13940          ADD     A,16
2863 6F        13950          LD      L,A
2864 5E        13960          LD      E,(HL)          ;P/u into reg DE
2865 2C        13970          INC     L
2866 56        13980          LD      D,(HL)
2867 C1        13990          POP     BC              ;Rcvr EOF byte in reg B
2868 EB        14000          EX      DE,HL           ;Xfer EOFREC -> reg HL
2869 3E00      14010 ALLØ4    LD      A,Ø             ;P/u LRL
286B B7        14020          OR      A
286C 2818      14030          JR      Z,TSTSIZ        ;Go use ERN if LRL=Ø
```

LBDIRC - File Attribute Output

```
286E 5F       14Ø4Ø          LD      E,A             ;Xfer LRL to reg E
286F Ø4       14Ø5Ø          INC     B               ;Test EOF
287Ø Ø5       14Ø6Ø          DEC     B
2871 28Ø1     14Ø7Ø          JR      Z,DONTDEC       ;Don't dec ERN if EOF=Ø
2873 2B       14Ø8Ø          DEC     HL              ;Reduce ERN for Ø offset
2874 CD3Ø29   14Ø9Ø DONTDEC  CALL    DIVIDE
2877 4D       141ØØ          LD      C,L
2878 54       1411Ø          LD      D,H
2879 67       1412Ø          LD      H,A
287A 68       1413Ø          LD      L,B             ;P/u EOF
287B 7B       1414Ø          LD      A,E
287C CD3Ø29   1415Ø          CALL    DIVIDE
287F 61       1416Ø          LD      H,C
288Ø B7       1417Ø          OR      A
2881 28Ø1     1418Ø          JR      Z,DONTINC
2883 23       1419Ø          INC     HL              ;Round up partial record
2884 7A       142ØØ DONTINC  LD      A,D             ;Ck if overflow
2885 B7       1421Ø          OR      A
2886 28ØD     1422Ø TSTSIZ   JR      Z,EOFBYTE       ;Use calc'd ERN if not
              1423Ø ;
              1424Ø ;        Overflow in # of Records - use "*****"
              1425Ø ;
2888 21D22D   1426Ø          LD      HL,STARS
288B 11B32C   1427Ø          LD      DE,RECORDS      ;DE => Destination
288E Ø1Ø5ØØ   1428Ø          LD      BC,5
2891 EDBØ     1429Ø          LDIR
2893 181B     143ØØ          JR      DIR_Ø
              1431Ø ;
              1432Ø ;        If # Records = Ø then set EOF = Ø
              1433Ø ;
2895 11ØØØØ   1434Ø EOFBYTE  LD      DE,ØØ           ;P/u EOF offset byte
2898 7C       1435Ø          LD      A,H             ;# Records = Ø ?
2899 B5       1436Ø          OR      L
289A 2ØØ2     1437Ø          JR      NZ,KEEPEOF      ;No - keep EOF
289C 1EØ1     1438Ø          LD      E,1             ;Set EOF=1 (gets DECed)
289E E5       1439Ø KEEPEOF  PUSH    HL              ;Save # Records
289F 21BB2C   144ØØ          LD      HL,OFFSET       ;HL => Destination
28A2 1D       1441Ø          DEC     E               ;DE = EOF byte
28A3 EB       1442Ø          EX      DE,HL           ;Swap for conversion
28A4 3E2Ø     1443Ø          LD      A,' '           ;Init
28A6 CD382A   1444Ø          CALL    CVD1ØØ          ;Convert
              1445Ø ;
              1446Ø ;        Stuff # of Records used into string
              1447Ø ;
28A9 E1       1448Ø          POP     HL              ;Recover # of Records
28AA 11B32C   1449Ø          LD      DE,RECORDS      ;DE => Destination
28AD CD2A2A   145ØØ          CALL    CVDDEC          ;  & stuff into message
              1451Ø ;
              1452Ø ;        Get # of extents & Granules used
              1453Ø ;
28BØ E1       1454Ø DIR_Ø    POP     HL              ;Rcvr ptr to 1st byte
28B1 E5       1455Ø          PUSH    HL
28B2 CD4529   1456Ø          CALL    ALLØ9           ;Get total grans in use
28B5 D5       1457Ø          PUSH    DE
28B6 69       1458Ø          LD      L,C             ;Xfer # extents
28B7 6Ø       1459Ø          LD      H,B
28B8 11CD2C   146ØØ          LD      DE,EXTENTS
28BB 3E2Ø     1461Ø          LD      A,' '
28BD CD3E2A   1462Ø          CALL    CVD1Ø           ;Cvrt to decimal & stuff
```

LBDIRC - File Attribute Output

```
28CØ D1        14630          POP     DE
               14640 ;
               14650 ;        DE = # Grans used - Add to Grans Counter
               14660 ;
28C1 2AØ226    14670          LD      HL,(TOTGRNS+1)  ;P/u total grans
28C4 19        14680          ADD     HL,DE           ;Add this file's count
28C5 22Ø226    14690          LD      (TOTGRNS+1),HL  ;  & stuff into counter.
               14700 ;
28C8 21CØ2C    14710          LD      HL,KSIZE        ;Pt to where to stuff
28CB CDE829    14720          CALL    CALCK           ;Cvrt to K
28CE 21D32C    14730          LD      HL,DATEFLD-1    ;Blank out day-mo-yr
28D1 11D42C    14740          LD      DE,DATEFLD
28D4 Ø1Ø9ØØ    14750          LD      BC,9
28D7 EDBØ      14760          LDIR
28D9 E1        14770          POP     HL              ;Rcvr ptr to DIR+Ø
28DA 11D42C    14780          LD      DE,DATEFLD
28DD 23        14790          INC     HL
28DE 23        14800          INC     HL              ;Advance to date field
28DF 7E        14810          LD      A,(HL)
28EØ B7        14820          OR      A
28E1 2841      14830          JR      Z,ALLØ8         ;Ignore if no date saved
28E3 ØF        14840          RRCA                    ;Has date, get day
28E4 ØF        14850          RRCA
28E5 ØF        14860          RRCA
28E6 E61F      14870          AND     1FH
28E8 Ø62F      14880          LD      B,2FH           ;Convert day to decimal
28EA Ø4        14890 ALLØ6    INC     B               ;By counting # of 1Ø's
28EB D6ØA      14900          SUB     1Ø              ;Sub 1Ø from day #
28ED 3ØFB      14910          JR      NC,ALLØ6
28EF C63A      14920          ADD     A,3AH           ;Cvrt lo order to ASCII
28F1 F5        14930          PUSH    AF              ;Save day low order
28F2 78        14940          LD      A,B             ;Stuff day hi order
28F3 12        14950          LD      (DE),A
28F4 13        14960          INC     DE              ;Bump
28F5 F1        14970          POP     AF              ;Rcvr lo order day #
28F6 12        14980          LD      (DE),A          ;Stuff low order
28F7 13        14990          INC     DE              ;Bump pointer to msg
28F8 13        15000          INC     DE              ;  to pt to month field
28F9 E5        15010          PUSH    HL              ;Save DIR ptr
28FA 2B        15020          DEC     HL              ;Pt to DIR+1 (month+)
28FB 7E        15030          LD      A,(HL)          ;P/u month etc
28FC E6ØF      15040          AND     ØFH             ;Strip off flags
28FE 3D        15050          DEC     A               ;(mon-1)*3 to index
28FF 4F        15060          LD      C,A             ;String conversion table
29ØØ Ø7        15070          RLCA
29Ø1 81        15080          ADD     A,C
29Ø2 4F        15090          LD      C,A
29Ø3 Ø6ØØ      15100          LD      B,Ø
29Ø5 21AE2D    15110          LD      HL,MONTBL
29Ø8 Ø9        15120          ADD     HL,BC           ;Add offset to table
29Ø9 ØE03      15130          LD      C,3
29ØB EDBØ      15140          LDIR                    ;Move 3-char month
29ØD 13        15150          INC     DE              ;Advance to year field
29ØE 3E38      15160          LD      A,'8'           ;Stuff 8 of 198Ø
291Ø 12        15170          LD      (DE),A
2911 13        15180          INC     DE              ;Bump msg ptr
2912 E1        15190          POP     HL              ;Rcvr DIR+2
2913 E5        15200          PUSH    HL
2914 7E        15210          LD      A,(HL)          ;P/u year field
```

LBDIRC - File Attribute Output

```
2915 E6Ø7     1522Ø          AND     7               ;Remove day
2917 C63Ø     1523Ø          ADD     A,'Ø'           ;Cvrt to ASCII
2919 12       1524Ø          LD      (DE),A          ;Stuff -> msg
291A E1       1525Ø          POP     HL              ;Rcvr DIR+2
291B 2B       1526Ø          DEC     HL              ;B/u to DIR+1
291C 3E2D     1527Ø          LD      A,'-'           ;Else change to not cur
291E 32D62C   1528Ø          LD      (DATEFLD+2),A   ;Stuff indicator
2921 32DA2C   1529Ø          LD      (DATEFLD+6),A   ;  between mo&day, day&yr
2924 21A72C   153ØØ ALLØ8    LD      HL,PLEVEL       ;Pt to start of message
2927 CDB929   1531Ø          CALL    LINOUT          ;  & output entire string
292A 3EØ1     1532Ø          LD      A,1             ;Show only one entry
292C 32ØE28   1533Ø          LD      (DONAM9+1),A    ;  per line if A-parm
292F C9       1534Ø          RET
              1535Ø ;
              1536Ø ;        DIVIDE - Divide HL by A
              1537Ø ;
2930 C5       1538Ø DIVIDE   PUSH    BC              ;Save BC
2931 4F       1539Ø          LD      C,A             ;Xfer Divisor in C
2932          154ØØ          @@DIV16                 ;Divide HL / C
2932 3E5E     ØØØ2Ø          LD      A,94
2934 EF       ØØØ21          RST     4Ø
2935 C1       1541Ø          POP     BC              ;Restore BC
2936 C9       1542Ø          RET
              1543Ø ;
              1544Ø ;        OUTMOD - Output a "+" if file has been modified
              1545Ø ;
2937 23       1546Ø OUTMOD   INC     HL              ;HL => DIR+1
2938 3E2Ø     1547Ø          LD      A,' '           ;Default to no mod
293A CB76     1548Ø          BIT     6,(HL)          ;Test MOD flag
293C 28Ø2     1549Ø          JR      Z,OUTCHR        ;Output space
293E 3E2B     155ØØ          LD      A,'+'           ;Mod flag char
2940 CDCA29   1551Ø OUTCHR   CALL    BYTOUT2         ;Display '+' if MOD
2943 2B       1552Ø          DEC     HL              ;Repoint to 1st byte
2944 C9       1553Ø          RET                     ;Done
              1554Ø ;
              1555Ø ;
              1556Ø ;         routine calculates total # of grans in use
              1557Ø ;
2945 3AF926   1558Ø ALLØ9    LD      A,(SORTPRM+1)   ;If sorted, then data
2948 B7       1559Ø          OR      A               ;  already calculated
2949 281Ø     156ØØ          JR      Z,ALLØ9A        ;Go if not sorted
294B E5       1561Ø          PUSH    HL
294C DDE1     1562Ø          POP     IX              ;P/u the saved data
294E DD5E16   1563Ø          LD      E,(IX+22)
2951 DD5617   1564Ø          LD      D,(IX+23)       ;P/u Space used
2954 DD4E18   1565Ø          LD      C,(IX+24)
2957 DD4619   1566Ø          LD      B,(IX+25)       ;P/u # of extents
295A C9       1567Ø          RET
              1568Ø ;
              1569Ø ;        ALLØ9A - Calculate space allocated to a file
              157ØØ ;        HL => DIR+Ø of an FPDE
              1571Ø ;        BC <= # of Extents in the file
              1572Ø ;        DE <= # of Grans allocated to the file
              1573Ø ;
295B 11ØØØØ   1574Ø ALLØ9A   LD      DE,Ø            ;Init gran counter to Ø
295E 43       1575Ø          LD      B,E             ;Init extent ctr to Ø
295F 4B       1576Ø          LD      C,E
              1577Ø ;
              1578Ø ;        Point to First Extent of a directory entry
```

LBDIRC - File Attribute Output

```
                15790 ;
296Ø 7D         15800 ALL1Ø   LD      A,L             ;P/u low byte
2961 C616       15810 ALL11   ADD     A,22
2963 6F         15820         LD      L,A             ;HL => DIR + 22
                15830 ;
                15840 ;       Is the Extent Field in Use ?
                15850 ;
2964 7E         15860 ALL14   LD      A,(HL)          ;P/u cylinder
2965 2C         15870         INC     L               ;Bump to alloc info
2966 FEFE       15880         CP      ØFEH            ;Another extent or done ?
2968 3ØØD       15890         JR      NC,ALL15        ;Either X'FE' or X'FF'
                15900 ;
                15910 ;       Extent Field is in use - Get allocation info
                15920 ;
296A Ø3         15930         INC     BC              ;Bump extent counter
296B 7E         15940         LD      A,(HL)          ;P/u alloc info
296C 2C         15950         INC     L               ;Bump ptr to next extent
296D E61F       15960         AND     1FH             ;Keep # of grans
296F 3C         15970         INC     A               ;Adj for zero offset
                15980 ;
                15990 ;       A = # of contig grans, add to gran counter
                16000 ;
297Ø 83         16010         ADD     A,E             ;Accumulate # of grans
2971 5F         16020         LD      E,A
2972 3ØFØ       16030         JR      NC,ALL14        ;Forget hi if no carry
2974 14         16040         INC     D               ;Bump hi
2975 18ED       16050         JR      ALL14           ;Get next extent field
                16060 ;
                16070 ;       P/u DEC if (X'FE') or RET if done (X'FF')
                16080 ;
2977 CØ         16090 ALL15   RET     NZ              ;Ret if not extended
2978 7E         16100         LD      A,(HL)          ;P/u DEC of FXDE
                16110 ;
                16120 ;       Point HL => Extended Directory Entry posn
                16130 ;
2979 E61F       16140         AND     1FH             ;Get dir sector of DEC
297B F5         16150         PUSH    AF              ;Save it
297C AE         16160         XOR     (HL)            ;Get dir record of FXDE
297D 6F         16170         LD      L,A             ;Save dir record position
297E F1         16180         POP     AF              ;Recover DEC of FXDE
                16190 ;
                16200 ;       Is the Dir Sector with FXDE already in mem ?
                16210 ;
297F E5         16220         PUSH    HL              ;Save ptr to 1st extent
2980 215226     16230         LD      HL,CKHIT6+1     ;Do we have this dir
2983 BE         16240         CP      (HL)            ;  sector in core?
2984 E1         16250         POP     HL              ;Restore ptr
2985 2600       16260 SBUFFER LD      H,ØØ            ;Buffer hi order
2987 28D7       16270         JR      Z,ALL1Ø         ;Jump if we have it
                16280 ;
                16290 ;       Dir Sector not res - Is Ext buf resident ?
                16300 ;
2989 FEØØ       16310 ALL16   CP      Ø               ;Same as extended area?
298B 262F       16320         LD      H,BUF2<-8       ;Pt to extended buf area
298D 28D1       16330         JR      Z,ALL1Ø         ;Jump if we have it there
298F 328A29     16340         LD      (ALL16+1),A     ;  else upd the test byte
                16350 ;
                16360 ;       Set B = Directory Entry Code of FXDE
                16370 ;
```

LBDIRC - File Attribute Output

```
2992 C5        16380          PUSH    BC              ;Save Gran counter
2993 D5        16390          PUSH    DE              ;  & Extent counter
2994 B5        16400          OR      L               ;Combine sector & record
2995 47        16410          LD      B,A             ;  pointers to retrieve DEC
               16420 ;
               16430 ;        Set C = Logical Drive #, D = Directory Cyl
               16440 ;
2996 3AØE2C    16450          LD      A,(DRIVE)       ;P/u ASCII drive #
2999 D63Ø      16460          SUB     'Ø'             ;Adjust to binary
299B 4F        16470          LD      C,A             ;Save in C
299C FD56Ø9    16480          LD      D,(IY+9)        ;P/u Directory cyl in D
               16490 ;
               16500 ;        Set E = FXDE's Dir Sector, HL => I/O buffer
               16510 ;
299F 78        16520          LD      A,B             ;P/u DEC
29AØ E61F      16530          AND     1FH             ;Get sector #
29A2 C6Ø2      16540          ADD     A,2             ;Adj for GAT & HIT
29A4 5F        16550          LD      E,A             ;Stuff in E
29A5 21ØØ2F    16560          LD      HL,BUF2         ;HL => I/O Buffer
               16570 ;
               16580 ;        Read in the FXDE's Directory Sector
               16590 ;
29A8           16600          @@RDSEC                 ;Read a sector
29A8 3E31      ØØØ22          LD      A,49
29AA EF        ØØØ23          RST     4Ø
29AB FEØ6      16610          CP      6               ;Expecting Error #6
29AD 3E11      16620          LD      A,11H           ;Read error?
29AF C21E24    16630          JP      NZ,IOERR        ;Jump if got error
               16640 ;
               16650 ;        Set A = offset into Sector of entry
               16660 ;
29B2 78        16670          LD      A,B             ;P/u FXDE DEC
29B3 E6EØ      16680          AND     ØEØH            ;Pt to dir record
29B5 D1        16690          POP     DE              ;Restore counters
29B6 C1        16700          POP     BC
29B7 18A8      16710          JR      ALL11           ;Loop through extents
               16720 ;
               16730 ;        LINOUT - Output line to *DO/*PR
               16740 ;        HL => Buffer to output
               16750 ;
29B9           16760 LINOUT   @@DSPLY                 ;Output line to *DO
               ØØØ24          IFEQ    ØØH,1
               ØØØ25          LD      HL,
               ØØØ26          ENDIF
29B9 3EØA      ØØØ27          LD      A,1Ø
29BB EF        ØØØ28          RST     4Ø
29BC 2ØØ8      16770          JR      NZ,IOER1        ;NZ - Abort
29BE 3AD329    16780          LD      A,(PPARM+1)     ;Ck P-parm
29C1 B7        16790          OR      A
29C2 C8        16800          RET     Z               ;Not spec'd - don't print
29C3           16810          @@PRINT                 ;Output line to *PR
               ØØØ29          IFEQ    ØØH,1
               ØØØ3Ø          LD      HL,
               ØØØ31          ENDIF
29C3 3EØE      ØØØ32          LD      A,14
29C5 EF        ØØØ33          RST     4Ø
29C6 C21E24    16820 IOER1    JP      NZ,IOERR        ;NZ - Abort
29C9 C9        16830          RET
               16840 ;
```

LBDIRC - File Attribute Output

```
                16850 ;       BYTOUT - Output a byte to *DO/*PR
                16860 ;       A = Character to output
                16870 ;
29CA ØD         16880 BYTOUT2 DEC     C               ;Decrement col #
29CB C5         16890 BYTOUT  PUSH    BC              ;Save BC
29CC 4F         16900         LD      C,A             ;Save char in C
29CD            16910         @@DSP                   ;Display char
29CD 3EØ2       00034         LD      A,2
29CF EF         00035         RST     40
29DØ 20F4       16920         JR      NZ,IOER1        ;NZ - Abort
29D2 110000     16930 PPARM   LD      DE,Ø            ;P/u P-parm
29D5 1C         16940         INC     E               ;Specified ?
29D6 2006       16950         JR      NZ,NOPRT        ;No - don't print
29D8            16960         @@PRT                   ;Output byte
29D8 3E06       00036         LD      A,6
29DA EF         00037         RST     40
29DB 20E9       16970         JR      NZ,IOER1        ;NZ - Abort
29DD 79         16980         LD      A,C             ;Get back char
29DE C1         16990 NOPRT   POP     BC              ;Restore BC
29DF C9         17000         RET                     ;And return
                17010 ;
                17020 ;       OUTSPC - Output B spaces
                17030 ;
29EØ 3E20       17040 OUTSPC  LD      A,' '           ;Space char
29E2 CDCA29     17050         CALL    BYTOUT2         ;Output space
29E5 10F9       17060         DJNZ    OUTSPC
29E7 C9         17070         RET                     ;RETurn
29E8            00300 *GET LBDIRC:3
                17080 ;LBDIRC/ASM - DIR math, strings, & buffers
29E8            17090         SUBTTL  '<LBDIRC - Math Routines>'
```

LBDIRC - Math Routines

```
29E8            17100           PAGE
                17110 ;
                17120 ;         CALCK - Calculate the # of K given # of Grans
                17130 ;         DE => # of Granules
                17140 ;         HL => Destination of #K ASCII string
                17150 ;
29E8 22FD29     17160 CALCK     LD      (CALCK2+1),HL   ;Stuff dest address
                17170 ;
                17180 ;         Calc # of Free Sects (Sectors/Gran x Grans)
                17190 ;
29EB EB         17200           EX      DE,HL           ;HL = # of Free Grans
29EC ØE00       17210 CALCK1    LD      C,$-$           ;C = Sectors/Gran
29EE            17220           @@MUL16                 ;Mult HL x C
29EE 3E5B       00038           LD      A,91
29F0 EF         00039           RST     40
                17230 ;
                17240 ;         LA = Total # of Sectors - Divide by 4 for K
                17250 ;
29F1 F5         17260           PUSH    AF              ;Save offset
29F2 65         17270           LD      H,L             ;Set HL = LA
29F3 6F         17280           LD      L,A
29F4 CB3C       17290           SRL     H               ;Divide HL / 4
29F6 CB1D       17300           RR      L
29F8 CB3C       17310           SRL     H
29FA CB1D       17320           RR      L
                17330 ;
                17340 ;         P/u dest address & stuff in # of FULL K
                17350 ;
29FC 110000     17360 CALCK2    LD      DE,$-$          ;P/u destination address
29FF CD2A2A     17370           CALL    CVDDEC          ;Stuff in message
2A02 13         17380           INC     DE              ;DE => Hundredths
                17390 ;
                17400 ;         Stuff hundredths value into string
                17410 ;
2A03 F1         17420           POP     AF              ;Rcvr offset to
2A04 E603       17430           AND     3               ;Get offset
2A06 87         17440           ADD     A,A
2A07 0600       17450           LD      B,0
2A09 4F         17460           LD      C,A             ;BC = offset
2A0A 21722D     17470           LD      HL,HUNDTAB      ;HL => Hundredths table
2A0D 09         17480           ADD     HL,BC           ;HL => Hundredths offset
2A0E 0E02       17490           LD      C,2             ;BC = 2 characters
2A10 EDB0       17500           LDIR                    ;Transfer to DE
2A12 C9         17510           RET
                17520 ;
                17530 ;         CPHLDE - Compare HL to DE
                17540 ;
2A13 7C         17550 CPHLDE    LD      A,H             ;P/u high byte
2A14 BA         17560           CP      D               ;Same ?
2A15 C0         17570           RET     NZ              ;No - Return C or NC
2A16 7D         17580           LD      A,L             ;P/u low byte
2A17 BB         17590           CP      E               ;Less than or greater ?
2A18 C9         17600           RET                     ;Return - C, NC, or Z
                17610 ;
                17620 ;         UNPACK - Unpack the Date from a directory entry
                17630 ;         HL => DIR+1
                17640 ;         DE <= Date in DATE$ format
                17650 ;
2A19 7E         17660 UNPACK    LD      A,(HL)          ;P/u DIR+1
2A1A E60F       17670           AND     0FH             ;Bits 3-0 contain month
```

LBDIRC - Math Routines

```
2A1C 57       17680          LD      D,A             ;Save month in D
              17690 ;
              17700 ;        D = Month (1-12)
              17710 ;
2A1D 23       17720          INC     HL              ;HL => DIR+2
2A1E 7E       17730          LD      A,(HL)          ;P/u Day/Year byte
2A1F E6F8     17740          AND     ØF8H            ;Mask off the year
2A21 5F       17750          LD      E,A             ;Save day in E
2A22 7E       17760          LD      A,(HL)          ;P/u Day/Year byte
2A23 AB       17770          XOR     E               ;Strip off day
2A24 ØF       17780          RRCA                    ;Year to bits 5-7
2A25 ØF       17790          RRCA
2A26 ØF       17800          RRCA
2A27 B2       17810          OR      D               ;  and merge with month
2A28 57       17820          LD      D,A
2A29 C9       17830          RET
              17840 ;
              17850 ;        conversion of grans to decimal
              17860 ;
2A2A 3E2Ø     17870 CVDDEC   LD      A,' '
2A2C Ø11Ø27   17880          LD      BC,1ØØØØ        ;Init 1st power or 1Ø
2A2F CD4A2A   17890          CALL    CVDØ            ;Make ASCII
2A32 Ø1E8Ø3   17900          LD      BC,1ØØØ         ;Continue until 1s
2A35 CD4A2A   17910          CALL    CVDØ
2A38 Ø164ØØ   17920 CVD1ØØ   LD      BC,1ØØ
2A3B CD4A2A   17930          CALL    CVDØ
2A3E Ø1ØAØØ   17940 CVD1Ø    LD      BC,1Ø
2A41 CD4A2A   17950          CALL    CVDØ
2A44 7D       17960          LD      A,L
2A45 C63Ø     17970          ADD     A,3ØH
2A47 12       17980          LD      (DE),A
2A48 13       17990          INC     DE
2A49 C9       18ØØØ          RET
2A4A D5       18Ø1Ø CVDØ     PUSH    DE
2A4B 5F       18Ø2Ø          LD      E,A             ;Fill character
2A4C 16FF     18Ø3Ø          LD      D,ØFFH          ;Init to Ø
2A4E AF       18Ø4Ø          XOR     A
2A4F 14       18Ø5Ø CVD1     INC     D               ;Inc digit counter
2A5Ø ED42     18Ø6Ø          SBC     HL,BC
2A52 3ØFB     18Ø7Ø          JR      NC,CVD1         ;Sub power/1Ø til carry
2A54 Ø9       18Ø8Ø          ADD     HL,BC           ;Add back the carry
2A55 7B       18Ø9Ø          LD      A,E
2A56 42       18ıØØ          LD      B,D             ;Get count this power
2A57 D1       1811Ø          POP     DE
2A58 12       1812Ø          LD      (DE),A          ;Default char to buffer
2A59 Ø4       1813Ø          INC     B
2A5A Ø5       1814Ø          DEC     B               ;Was count = Ø?
2A5B 28Ø6     1815Ø          JR      Z,CVD2          ;Go if so
2A5D 78       1816Ø          LD      A,B
2A5E C63Ø     1817Ø          ADD     A,3ØH           ;  else make count ASCII
2A6Ø 12       1818Ø          LD      (DE),A          ;  and store in buffer
2A61 3E3Ø     1819Ø          LD      A,3ØH
2A63 13       182ØØ CVD2     INC     DE              ;Advance buffer
2A64 C9       1821Ø          RET
              1822Ø ;
              1823Ø ;        CKPAGE - Check for Page Pause
              1824Ø ;
2A65 3EØØ     1825Ø CKPAGE   LD      A,$-$           ;Ck for display pause
2A67 3D       1826Ø          DEC     A               ;Count down
```

LBDIRC - Math Routines

```
2A68 32662A   18270          LD      (CKPAGE+1),A    ;Update
2A6B C0       18280          RET     NZ              ;Ret if not yet full
              18290 ;
              18300 ;        Displayed a full page - Reset Counter
              18310 ;
2A6C 3E16     18320          LD      A,22            ;Reset to max lines/page
2A6E 32662A   18330          LD      (CKPAGE+1),A
              18340 ;
              18350 ;        Don't pause if NOPAUSE (N) parm entered
              18360 ;
2A71 110000   18370 NPARM    LD      DE,0            ;P/u NOPAUSE parm
2A74 7B       18380          LD      A,E             ;Specified ?
2A75 B2       18390          OR      D
2A76 C0       18400          RET     NZ              ;Nonstop if non-zero
              18410 ;
              18420 ;        Non-Stop if <DO> in effect
              18430 ;
2A77 3E00     18440 SFLAG    LD      A,$-$           ;P/u SFLAG$
2A79 E620     18450          AND     20H             ;Strip all but <DO> bit
2A7B C0       18460          RET     NZ              ;Return if do in effect
              18470 ;
              18480 ;        There isn't a <DO> in effect - Wait for key
              18490 ;
2A7C          18500          @@KEY                   ;Wait for key entry
2A7C 3E01     00040          LD      A,1
2A7E EF       00041          RST     40
2A7F C21E24   18510 IOERR5   JP      NZ,IOERR
              18520 ;
              18530 ;        Clear Screen
              18540 ;
2A82          18550          @@CLS                   ;Clear Screen
2A82 3E69     00042          LD      A,105
2A84 EF       00043          RST     40
2A85 20F8     18560          JR      NZ,IOERR5
              18570 ;
              18580 ;        If the NOTITLE flag is set - don't display
              18590 ;
2A87 3E00     18600 NOTITLE  LD      A,$-$           ;P/u flag
2A89 B7       18610          OR      A               ;No title ?
2A8A C0       18620          RET     NZ              ;Then RETurn
              18630 ;
              18640 ;        Display a title if there were matching files
              18650 ;
2A8B 3AD82D   18660          LD      A,(FILFLAG)     ;Was a matching file
2A8E B7       18670          OR      A               ;  displayed ?
2A8F C4B12A   18680          CALL    NZ,CKTITL       ;Yes - display title
2A92 C9       18690          RET                     ;Return
              18700 ;
              18710 ;        CKPAWS - Check for <SHIFT><@> or <BREAK>
              18720 ;
              18730 CKPAWS
              18740 ;
              18750 ;        Was the <BREAK> key hit ?
              18760 ;
2A93 3A0000   18770 KFLAG    LD      A,($-$)         ;P/u KFLAG$
2A96 0F       18780          RRCA                    ;<BREAK> hit ?
2A97 DA1724   18790          JP      C,ABORT         ;Yes - cease DIR
              18800 ;
              18810 ;        Is the <PAUSE> bit set ?
```

LBDIRC - Math Routines

```
                18820 ;
2A9A ØF         18830           RRCA                    ;<PAUSE> bit set ?
2A9B DØ         18840           RET     NC              ;Ret if not pause
                18850 ;
                18860 ;         The <PAUSE> bit is set - Wait for Char
                18870 ;
2A9C            18880 CKPAW1    @@KEY                   ;Scan keyboard
2A9C 3EØ1       ØØØ44           LD      A,1
2A9E EF         ØØØ45           RST     4Ø
                18890 ;
                18900 ;         Character entered - Ignore it if <SHIFT><@>
                18910 ;
2A9F FE6Ø       18920 CKPAW2    CP      6ØH             ;<SHIFT><@> ?
2AA1 28F9       18930           JR      Z,CKPAW1        ;Yes - get another char
2AA3 FE8Ø       18940           CP      BREAK
2AA5 CA1724     18950           JP      Z,ABORT
                18960 ;
                18970 ;         Reset <PAUSE> & <ENTER> bits
                18980 ;
2AA8 3AØØØØ     18990 RESKFL    LD      A,($-$)         ;P/u KFLAG$
2AAB E6F9       19000           AND     ØF9H            ;Reset <PAUSE> & <ENTER>
2AAD 32ØØØØ     19010 KFLAG1    LD      ($-$),A         ;Stuff into KFLAG$
2ABØ C9         19020           RET                     ;  & RETurn
                19030 ;
                19040 ;         CKTITL - Display Title
                19050 ;
                19060 ;         Display Disk type Header
                19070 ;
2AB1 21Ø72C     19080 CKTITL    LD      HL,DSTRING      ;HL => Heading
2AB4 CDB929     19090           CALL    LINOUT          ;Output line
2AB7 CD652A     19100           CALL    CKPAGE          ;Bump line count
2ABA CD652A     19110           CALL    CKPAGE          ;  twice.
                19120 ;
                19130 ;         Display Attributes header if A parm spec'd
                19140 ;
2ABD 3AØ628     19150           LD      A,(APARM+1)     ;Was the A parm spec'd
2ACØ B7         19160           OR      A
2AC1 3EØD       19170           LD      A,CR            ;Output a CR if A
2AC3 CACB29     19180           JP      Z,BYTOUT        ;  not specified.
                19190 ;
2AC6 21572C     19200           LD      HL,HEADING      ;HL => Attr heading
2AC9 CDB929     19210           CALL    LINOUT          ;Output line
                19220 ;
                19230 ;         Display Underline
                19240 ;
2ACC C5         19250           PUSH    BC              ;Save BC
2ACD Ø64F       19260           LD      B,79            ;Display underline
2ACF 3E2D       19270 D79L      LD      A,'-'
2AD1 CDCB29     19280           CALL    BYTOUT          ;Output byte
2AD4 1ØF9       19290           DJNZ    D79L            ;  79 times
2AD6 C1         19300           POP     BC              ;Restore BC
2AD7 3EØD       19310           LD      A,CR            ;One CR between
2AD9 CDCB29     19320           CALL    BYTOUT
2ADC C3652A     19330           JP      CKPAGE          ;Check page pause & RET
                19340 ;
2ADF            19350           SUBTTL  '<LBDIRC - Sort Code>'
                19360 ;
                19370 ;         SORTIT - Set up Directory Records for Shell Sort
                19380 ;
```

LBDIRC - Sort Code

```
2ADF 2AD92D   19390 SORTIT  LD      HL,(DIRPTR)     ;Calculate # of records
2AE2 11ØØ3Ø   19400         LD      DE,MEMORY       ;Point to buf start
2AE5 73       19410         LD      (HL),E          ;Prime the 1st index
2AE6 23       19420         INC     HL              ;  in case there is
2AE7 72       19430         LD      (HL),D          ;  only one record
2AE8 2B       19440         DEC     HL              ;  to sort
2AE9 AF       19450         XOR     A
2AEA ED52     19460         SBC     HL,DE           ;PTREND - PTRBGN
2AEC C8       19470         RET     Z               ;Ret if nothing
              19480 ;
              19490 ;       Set HL = # of directory entries
              19500 ;
2AED Ø6Ø5     19510         LD      B,5             ;Divide by
2AEF CB3C     19520 SORT1   SRL     H               ;  32 bytes/record
2AF1 CB1D     19530         RR      L
2AF3 1ØFA     19540         DJNZ    SORT1
              19550 ;
              19560 ;       Set B = # of entries & init count
              19570 ;
2AF5 45       19580         LD      B,L             ;Set loop counter
2AF6 C5       19590         PUSH    BC              ;Save it for printing
2AF7 22332B   19600         LD      (COUNTM1),HL    ;Init the count
              19610 ;
              19620 ;       Skip sort if # of entries = Ø
              19630 ;
2AFA 7C       19640         LD      A,H             ;If length = Ø
2AFB B5       19650         OR      L               ;  then no need to sort
2AFC 2821     19660         JR      Z,SORT2A
2AFE 29       19670         ADD     HL,HL           ;Make sure enuff room
2AFF EB       19680         EX      DE,HL
2BØØ 2ADB2D   19690         LD      HL,(MAXMEM)
2BØ3 AF       19700         XOR     A
2BØ4 ED52     19710         SBC     HL,DE
2BØ6 DAØD24   19720         JP      C,NOMEM
2BØ9 2AD92D   19730         LD      HL,(DIRPTR)     ;Set up the index array
2BØC 11ØØ3Ø   19740         LD      DE,MEMORY       ;Starting record pointer
2BØF 73       19750 SORT2   LD      (HL),E          ;Place record pointers
2B1Ø 23       19760         INC     HL              ;  into index array
2B11 72       19770         LD      (HL),D
2B12 23       19780         INC     HL
2B13 7B       19790         LD      A,E             ;Increment pointer by 32
2B14 C62Ø     19800         ADD     A,32
2B16 5F       19810         LD      E,A
2B17 3ØØ1     19820         JR      NC,$+3          ;Go if no overflow
2B19 14       19830         INC     D               ;  else bump high order
2B1A 1ØF3     19840         DJNZ    SORT2           ;Loop for all records
2B1C CD322B   19850         CALL    SHELL           ;Sort the dir records
2B1F C1       19860 SORT2A  POP     BC              ;Recover loop counter
2B2Ø 2AD92D   19870         LD      HL,(DIRPTR)     ;P/u starting record
2B23 5E       19880 SORT3   LD      E,(HL)          ;Grab its address
2B24 23       19890         INC     HL
2B25 56       19900         LD      D,(HL)
2B26 23       19910         INC     HL
2B27 E5       19920         PUSH    HL              ;Save index pointer
2B28 C5       19930         PUSH    BC              ;Save loop counter
2B29 EB       19940         EX      DE,HL           ;Record address -> HL
2B2A CD3327   19950         CALL    MATCH           ;Display the record
2B2D C1       19960         POP     BC              ;Rcvr loop counter
2B2E E1       19970         POP     HL              ;Rcvr index pointer
```

LBDIRC - Sort Code

```
2B2F 1ØF2      19980          DJNZ    SORT3
2B31 C9        19990          RET
               20000 ;
               20010 ;        SHELL - Shell Sort Routine
               20020 ;
2B32 21ØØØØ   20030 SHELL    LD      HL,$-$          ;P/u count minus 1
2B33           20040 COUNTM1  EQU     $-2
2B35 22392B    20050          LD      (STORM),HL
               20060 ;
               20070 ;        Start Select & Compare
               20080 ;
2B38 11ØØØØ   20090 CYCLE    LD      DE,Ø            ;M = M / 2
2B39           20100 STORM    EQU     $-2
2B3B CB3A      20110          SRL     D
2B3D CB1B      20120          RR      E
2B3F 7A        20130          LD      A,D             ;Return when M=Ø
2B4Ø B3        20140          OR      E
2B41 C8        20150          RET     Z
2B42 ED53392B  20160          LD      (STORM),DE
2B46 2A332B    20170          LD      HL,(COUNTM1)    ;K = N - M
2B49 ED52      20180          SBC     HL,DE
2B4B 22BA2B    20190          LD      (STORK),HL
2B4E 21ØØØØ   20200          LD      HL,Ø            ;J = Ø
2B51 22552B    20210          LD      (STORJ),HL
2B54 21ØØØØ   20220 AGAIN    LD      HL,$-$          ;I = J
2B55           20230 STORJ    EQU     $-2
2B57 225B2B    20240          LD      (STORI),HL
2B5A 21ØØØØ   20250 REPEAT   LD      HL,$-$          ;L = I + M
2B5B           20260 STORI    EQU     $-2
2B5D ED5B392B  20270          LD      DE,(STORM)
2B61 19        20280          ADD     HL,DE
2B62 29        20290          ADD     HL,HL           ;L * 2 -> regHL
2B63 E5        20300          PUSH    HL              ;Save L
2B64 2A5B2B    20310          LD      HL,(STORI)      ;I * 2 -> regHL
2B67 29        20320          ADD     HL,HL
2B68 ED4BD92D  20330          LD      BC,(DIRPTR)     ;P/u string parm ptr
2B6C Ø9        20340          ADD     HL,BC           ;Pt to A$(I) parm
2B6D EB        20350          EX      DE,HL           ;Ptr -> DE
2B6E E1        20360          POP     HL              ;Pt to A$(L) parm
2B6F Ø9        20370          ADD     HL,BC           ;Ptr -> HL
2B7Ø E5        20380          PUSH    HL              ;Save ptr to A$(L)
2B71 D5        20390          PUSH    DE              ;Save ptr to A$(I)
2B72 Ø6ØB      20400          LD      B,11            ;Set compare length
2B74 C5        20410          PUSH    BC              ;Save cpr len & flag
2B75 7E        20420          LD      A,(HL)          ;P/u string2 ptr
2B76 23        20430          INC     HL
2B77 66        20440          LD      H,(HL)
2B78 6F        20450          LD      L,A
2B79 Ø1Ø5ØØ   20460          LD      BC,5            ;Key is 5 bytes in
2B7C Ø9        20470          ADD     HL,BC
2B7D EB        20480          EX      DE,HL           ;String2 ptr -> rDE
2B7E 7E        20490          LD      A,(HL)          ;P/u string1 ptr
2B7F 23        20500          INC     HL
2B8Ø 66        20510          LD      H,(HL)
2B81 6F        20520          LD      L,A
2B82 Ø9        20530          ADD     HL,BC           ;Key is 5 bytes in
2B83 C1        20540          POP     BC              ;Rcvr len & flag
2B84 1A        20550 BACK     LD      A,(DE)          ;Go swap if str1>str2
2B85 96        20560          SUB     (HL)
```

LBDIRC - Sort Code

```
2B86 38Ø8      2Ø57Ø          JR      C,POP
2B88 2Ø25      2Ø58Ø          JR      NZ,FINIS        ;Next str if str2>str1
2B8A 13        2Ø59Ø          INC     DE              ;Loop if this matches
2B8B 23        2Ø6ØØ          INC     HL
2B8C 1ØF6      2Ø61Ø          DJNZ    BACK
2B8E 181F      2Ø62Ø          JR      FINIS           ;None really should match
2B9Ø D1        2Ø63Ø POP      POP     DE              ;Else swap
2B91 E1        2Ø64Ø          POP     HL
2B92 Ø6Ø2      2Ø65Ø          LD      B,2             ;Swap 2-byte
2B94 4E        2Ø66Ø SWAP     LD      C,(HL)          ;String pointer
2B95 EB        2Ø67Ø          EX      DE,HL
2B96 7E        2Ø68Ø          LD      A,(HL)
2B97 71        2Ø69Ø          LD      (HL),C
2B98 EB        2Ø7ØØ          EX      DE,HL
2B99 77        2Ø71Ø          LD      (HL),A
2B9A 23        2Ø72Ø          INC     HL
2B9B 13        2Ø73Ø          INC     DE
2B9C 1ØF6      2Ø74Ø          DJNZ    SWAP
2B9E 2A392B    2Ø75Ø          LD      HL,(STORM)      ;P/u M
2BA1 EB        2Ø76Ø          EX      DE,HL
2BA2 2A5B2B    2Ø77Ø          LD      HL,(STORI)      ;P/u I
2BA5 AF        2Ø78Ø          XOR     A
2BA6 ED52      2Ø79Ø          SBC     HL,DE
2BA8 225B2B    2Ø8ØØ          LD      (STORI),HL      ;I = I - M
2BAB 3ØAD      2Ø81Ø          JR      NC,REPEAT       ;Repeat if I => Ø
2BAD 18Ø2      2Ø82Ø          JR      EXITSRT         ;Else exit the loop
2BAF D1        2Ø83Ø FINIS    POP     DE
2BBØ E1        2Ø84Ø          POP     HL
2BB1 2A552B    2Ø85Ø EXITSRT  LD      HL,(STORJ)
2BB4 23        2Ø86Ø          INC     HL              ;J = J + 1
2BB5 22552B    2Ø87Ø          LD      (STORJ),HL
2BB8 AF        2Ø88Ø          XOR     A
2BB9 11ØØØØ    2Ø89Ø          LD      DE,$-$
2BBA           2Ø9ØØ STORK    EQU     $-2
2BBC ED52      2Ø91Ø          SBC     HL,DE           ;J - K
2BBE D2382B    2Ø92Ø          JP      NC,CYCLE        ;Cycle if J => K *
2BC1 C3542B    2Ø93Ø          JP      AGAIN           ;Else again
               2Ø94Ø ;
2BC4           2Ø95Ø          SUBTTL  '<LBDIRC - Data>'
               2Ø96Ø ;
2BC4 8Ø        2Ø97Ø PRMTBL$  DB      8ØH             ;6.x parameters
               2Ø98Ø ;
               2Ø99Ø ;        A - Flag input only
               21ØØØ ;
2BC5 41        21Ø1Ø          DB      FLAG!1
2BC6 41        21Ø2Ø          DB      'A'
2BC7 ØØ        21Ø3Ø          DB      Ø
2BC8 Ø628      21Ø4Ø          DW      APARM+1
               21Ø5Ø ;
               21Ø6Ø ;        INV (I) - Flag input only
               21Ø7Ø ;
2BCA 53        21Ø8Ø          DB      FLAG!ABB!3
2BCB 49        21Ø9Ø          DB      'INV'
     4E 56
2BCE ØØ        211ØØ          DB      Ø
2BCF 8D26      2111Ø          DW      IPARM+1
               2112Ø ;
               2113Ø ;        P - Flag input only
               2114Ø ;
```

LBDIRC - Data

```
2BD1 41        21150         DB      FLAG!1
2BD2 50        21160         DB      'P'
2BD3 ØØ        21170         DB      Ø
2BD4 D329      21180         DW      PPARM+1
               21190 ;
               21200 ;       SYS (S) - Flag input only
               21210 ;
2BD6 53        21220         DB      FLAG!ABB!3
2BD7 53        21230         DB      'SYS'
     59 53
2BDA ØØ        21240         DB      Ø
2BDB 7F26      21250         DW      SPARM+1
               21260 ;
               21270 ;       N - Flag input only
               21280 ;
2BDD 41        21290         DB      FLAG!1
2BDE 4E        21300         DB      'N'
2BDF ØØ        21310         DB      Ø
2BEØ 722A      21320         DW      NPARM+1
               21330 ;
               21340 ;       DATE (D) - Flag or String input
               21350 ;
2BE2 74        21360         DB      FLAG!STR!ABB!4
2BE3 44        21370         DB      'DATE'
     41 54 45
2BE7 ØØ        21380 DRESP   DB      Ø
2BE8 982E      21390         DW      DATPRM+1
               21400 ;
               21410 ;       MOD (M) - Flag input only
               21420 ;
2BEA 53        21430         DB      FLAG!ABB!3
2BEB 4D        21440         DB      'MOD'
     4F 44
2BEE ØØ        21450         DB      Ø
2BEF 9526      21460         DW      CKMOD+1
               21470 ;
               21480 ;       SORT (O) - Flag input only
               21490 ;
2BF1 44        21500         DB      FLAG!4
2BF2 53        21510         DB      'SORT'
     4F 52 54
2BF6 ØØ        21520         DB      Ø
2BF7 F926      21530         DW      SORTPRM+1
               21540 ;
2BF9 41        21550         DB      FLAG!1
2BFA 4F        21560         DB      'O'
2BFB ØØ        21570         DB      Ø
2BFC F926      21580         DW      SORTPRM+1
               21590 ;
               21600 ;
2BFE ØØ        21610         DB      Ø
               21620 ;
2BFF 78        21630 DEN     DB      'xDEN'
     44 45 4E
2CØ3 48        21640 HARD    DB      'Hard'
     61 72 64
               21650 ;
2CØ7 44        21660 DSTRING DB      'Drive :'
     72 69 76 65 2Ø 3A
```

LBDIRC - Data

```
2CØE 64          21670 DRIVE   DB      'd  '
     2Ø 2Ø
2C11 64          21680 NAME    DB      'diskname  '
     69 73 6B 6E 61 6D 65 2Ø
     2Ø
2C1B 2Ø          21690 CYLCNT  DB      '    Cyl, '
     2Ø 2Ø 2Ø 43 79 6C 2C 2Ø
2C24 6E          21700 DENSITY DB      'nDEN, Free ='
     44 45 4E 2C 2Ø 46 72 65
     65 2Ø 3D
2C3Ø 2Ø          21710 KFREE   DB      '     . K / '
     2Ø 2Ø 2Ø 2Ø 2E 2Ø 2Ø 4B
     2Ø 2F 2Ø
2C3C 2Ø          21720 KPOSS   DB      '     . K,  Date '
     2Ø 2Ø 2Ø 2Ø 2E 2Ø 2Ø 4B
     2C 2Ø 2Ø 44 61 74 65 2Ø
2C4D 64          21730 DATBUF  DB      'dd-mmm-yy',CR
     64 2D 6D 6D 6D 2D 79 79
     ØD
                 21740 ;
2C57 46          21750 HEADING DB      'Filespec    MOD   Attr   Prot   LRL'
     69 6C 65 73 7Ø 65 63 2Ø
     2Ø 2Ø 2Ø 4D 4F 44 2Ø 2Ø
     2Ø 41 74 74 72 2Ø 2Ø 2Ø
     5Ø 72 6F 74 2Ø 2Ø 2Ø 4C
     52 4C
2C7A 2Ø          21760         DB      '  #Recs   EOF  File Size   Ext     Mod '
     2Ø 23 52 65 63 73 2Ø 2Ø
     2Ø 45 4F 46 2Ø 2Ø 46 69
     6C 65 2Ø 53 69 7A 65 2Ø
     2Ø 2Ø 45 78 74 2Ø 2Ø 2Ø
     2Ø 2Ø 4D 6F 64 2Ø
2CA1 44          21770         DB      'Date ',CR
     61 74 65 2Ø ØD
                 21780 ;
2CA7 5Ø          21790 PLEVEL  DB      'Prot   '
     72 6F 74 2Ø 2Ø 2Ø
2CAE 2Ø          21800 LRL     DB      '     '
     2Ø 2Ø 2Ø 2Ø
2CB3 2Ø          21810 RECORDS DB      '        '
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
2CBB 2Ø          21820 OFFSET  DB      '     '
     2Ø 2Ø 2Ø 2Ø
2CCØ 2Ø          21830 KSIZE   DB      '     . K    '
     2Ø 2Ø 2Ø 2Ø 2E 2Ø 2Ø 4B
     2Ø 2Ø 2Ø 2Ø
2CCD 2Ø          21840 EXTENTS DB      '       '
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
2CD4 64          21850 DATEFLD DB      'dd-mmm-yy',ETX
     64 2D 6D 6D 6D 2D 79 79
     Ø3
                 21860 ;
2CDE 2Ø          21870 FSTRING DB      '           '
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø
2CE9 2Ø          21880 FDISP   DB      '    files out of '
     2Ø 2Ø 2Ø 66 69 6C 65 73
     2Ø 6F 75 74 2Ø 6F 66 2Ø
2CFA 2Ø          21890 FUSED   DB      '    selected, Space = '
```

LBDIRC - Data

```
     2Ø 2Ø 2Ø 73 65 6C 65 63
     74 65 64 2C 2Ø 53 7Ø 61
     63 65 2Ø 3D 2Ø
2D1Ø 2Ø       21ØØØ SPUSED  DB      '     . K',LF,CR
     2Ø 2Ø 2Ø 2Ø 2E 2Ø 2Ø 4B
     ØA ØD
              21Ø1Ø ;
2D1B 44       21Ø2Ø NODISK  DB      'Drive :'
     72 69 76 65 2Ø 3A
2D22 6E       21Ø3Ø NDRIVE  DB      'n  [No  Disk]',LF,CR
     2Ø 2Ø 5B 4E 6F 2Ø 2Ø 44
     69 73 6B 5D ØA ØD
              21Ø4Ø ;
2D31 6D       21Ø5Ø TDATE   DB      'mm/dd/yy"'
     6D 2F 64 64 2F 79 79 22
2D3A 2Ø       21Ø6Ø BLANKS  DB      '            '
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø 2Ø
2D46 46       21Ø7Ø PROTS$  DB      'FULLREMVNAMEWRITUPDTREADEXECNO  '
     55 4C 4C 52 45 4D 56 4E
     41 4D 45 57 52 49 54 55
     5Ø 44 54 52 45 41 44 45
     58 45 43 4E 4F 2Ø 2Ø
2D66 1F       21Ø8Ø MAXDAYS DB      31,28,31,3Ø,31,3Ø,31,31,3Ø,31,3Ø,31
     1C 1F 1E 1F 1E 1F 1F 1E
     1F 1E 1F
2D72 3Ø       21Ø9Ø HUNDTAB DB      'ØØ255Ø75'
     3Ø 32 35 35 3Ø 37 35
2D7A 49       22ØØØ NOMEM$  DB      'Insufficient memory for SORT buffer',CR
     6E 73 75 66 66 69 63 69
     65 6E 74 2Ø 6D 65 6D 6F
     72 79 2Ø 66 6F 72 2Ø 53
     4F 52 54 2Ø 62 75 66 66
     65 72 ØD
2D9E 42       22Ø1Ø BADFMT$ DB      'Bad date format',CR
     61 64 2Ø 64 61 74 65 2Ø
     66 6F 72 6D 61 74 ØD
2DAE 4A       22Ø2Ø MONTBL  DB      'JanFebMarAprMayJunJulAugSepOctNovDec'
     61 6E 46 65 62 4D 61 72
     41 7Ø 72 4D 61 79 4A 75
     6E 4A 75 6C 41 75 67 53
     65 7Ø 4F 63 74 4E 6F 76
     44 65 63
2DD2 2A       22Ø3Ø STARS   DB      '*****'
     2A 2A 2A 2A
2DD7 ØØ       22Ø4Ø FTFLG   DB      Ø
2DD8 ØØ       22Ø5Ø FILFLAG DB      Ø
2DD9          22Ø6Ø DIRPTR  EQU     $
2DDB          22Ø7Ø MAXMEM  EQU     DIRPTR+2
2DDD          22Ø8Ø FMPAKD  EQU     MAXMEM+2
2DDF          22Ø9Ø TOPAKD  EQU     FMPAKD+2
2DE1          221ØØ LILBUF$ EQU     TOPAKD+2
              2211Ø ;
              2212Ø ;
              2213Ø ;
2EØØ          2214Ø GAT     EQU     LILBUF$+3<-8+1<+8
2EØØ          2215Ø HIT     EQU     GAT
2FØØ          2216Ø BUF2    EQU     GAT+256
3ØØØ          2217Ø MEMORY  EQU     GAT+512
```

LBDIRC - Data

```
                22180 ;
2DD9            22190           SUBTTL  '<LBDIRC - Initialization Code>'
```

LBDIRC - Initialization Code

```
2DD9            22200           PAGE
                22210 ;
                22220 ;         DIR Entry Point - Initialization code
                22230 ;
                22240 DIR
2DD9            22250           @@CKBRKC                ;Check for break
2DD9 3E6A       00046           LD      A,106
2DDB EF         00047           RST     40
2DDC 2804       22260           JR      Z,DIRA          ;If not go
2DDE 21FFFF     22270           LD      HL,-1           ;  else abort
2DE1 C9         22280           RET
                22290 ;
                22300 DIRA
2DE2 ED732D24   22310           LD      (SAVESP+1),SP   ;Save SP address
2DE6 E5         22320           PUSH    HL              ;Save command ptr
                22330 ;
                22340 ;         Pick up Flag Table base Address
                22350 ;
2DE7            22360           @@FLAGS                 ;IY => System Flag table
2DE7 3E65       00048           LD      A,101
2DE9 EF         00049           RST     40
2DEA FDE5       22370           PUSH    IY              ;Xfer to DE too
2DEC D1         22380           POP     DE
                22390 ;
                22400 ;         Calculate KFLAG$ address & stuff away
                22410 ;
2DED 210A00     22420           LD      HL,KFLAG$       ;KFLAG$ offset
2DF0 19         22430           ADD     HL,DE           ;HL => KFLAG$
2DF1 22942A     22440           LD      (KFLAG+1),HL    ;Save for later testing
2DF4 22A92A     22450           LD      (RESKFL+1),HL
2DF7 22AE2A     22460           LD      (KFLAG1+1),HL
                22470 ;
2DFA CDA82A     22480           CALL    RESKFL          ;Reset bits 0-2 of KFLAG$
2DFD E1         22490           POP     HL              ;Rvr command ptr
                22500 ;
                22510 ;         Pick up SFLAG
                22520 ;
2DFE FD7E12     22530           LD      A,(IY+'S'-'A')  ;Get SFLAG
2E01 32782A     22540           LD      (SFLAG+1),A     ;Save for later testing
                22550 ;
                22560 ;         Find parameter entry if existent
                22570 ;
2E04 E5         22580           PUSH    HL              ;Save command ptr
2E05 7E         22590 FPLP      LD      A,(HL)          ;P/u character
2E06 FE28       22600           CP      '('             ;Parameter(s) ?
2E08 2807       22610           JR      Z,GETPRM        ;Yes - go get 'em
2E0A FE0D       22620           CP      CR              ;End of line ?
2E0C 2809       22630           JR      Z,RESTPTR       ;Yes - restore ptr
2E0E 23         22640           INC     HL              ;No - bump til end
2E0F 18F4       22650           JR      FPLP            ;Do til eol or "("
                22660 ;
                22670 ;         Process any parameters entered
                22680 ;
2E11 11C42B     22690 GETPRM    LD      DE,PRMTBL$      ;DE => Parameter table
2E14            22700           @@PARAM                 ;@PARAM
2E14 3E11       00050           LD      A,17
2E16 EF         00051           RST     40
2E17 E1         22710 RESTPTR   POP     HL              ;Recover ptr
2E18 C21E24     22720           JP      NZ,IOERR        ;NZ - "Parameter Error"
                22730 ;
```

LBDIRC - Initialization Code

```
                22740 ;       If first character is a "8" or "9" abort
                22750 ;
2E1B 7E         22760         LD      A,(HL)          ;Is this a "8" or "9" ?
2E1C FE0D       22770         CP      CR              ;If CR, then global
2E1E 2834       22780         JR      Z,DIR2
2E20 FE38       22790         CP      '8'             ;If so - Illegal drive #
2E22 2804       22800         JR      Z,ILLDRV
2E24 FE39       22810         CP      '9'
2E26 2003       22820         JR      NZ,CKITOUT      ;Must be a partspec
                22830 ;
                22840 ;       Illegal Drive Number
                22850 ;
2E28 C31C24     22860 ILLDRV  JP      ERR32           ;Go to I/O error handler
                22870 ;
                22880 ;       Pick up Drive # Range field if any
                22890 ;
2E2B E5         22900 CKITOUT PUSH    HL              ;Save source ptr
2E2C CDE72E     22910         CALL    CKDSPEC         ;Legal Drive range ?
2E2F D1         22920         POP     DE              ;Save source ptr in DE
2E30 2832       22930         JR      Z,DIR3          ;Legal - use HL
                22940 ;
                22950 ;       Point DE => Partspec match field, B=8 chars
                22960 ;
2E32 EB         22970         EX      DE,HL           ;Illegal - use DE
2E33 7E         22980         LD      A,(HL)          ;P/u first char
2E34 23         22990         INC     HL              ;  and bump to next
2E35 113A2D     23000 DIR0    LD      DE,BLANKS       ;DE => Partspec area
2E38 0608       23010         LD      B,8             ;B = 8 chars/filename
                23020 ;
                23030 ;       Was the NOT switch entered ?
                23040 ;
2E3A FE2D       23050         CP      '-'             ;NOT ?
2E3C 2005       23060         JR      NZ,DIR1         ;No - continue
                23070 ;
                23080 ;       NOT "-" entered - set flag & bump cmd ptr
                23090 ;
2E3E 32C426     23100         LD      (MFLG+1),A      ;Stuff "-" in flag
2E41 7E         23110         LD      A,(HL)          ;P/u next char & bump
2E42 23         23120         INC     HL              ;  command ptr
                23130 ;
                23140 ;       Transfer Filename to Filespec buffer
                23150 ;
2E43 CD2A2F     23160 DIR1    CALL    PRSPC           ;Parse 8 chars
2E46 FE2F       23170         CP      '/'             ;Extension ?
2E48 200A       23180         JR      NZ,DIR2         ;No - don't check
                23190 ;
                23200 ;       Transfer Extension to Filespec buffer
                23210 ;
2E4A 11422D     23220         LD      DE,BLANKS+8     ;DE => Extension field
2E4D 0603       23230         LD      B,3             ;Max 3 chars
2E4F 7E         23240         LD      A,(HL)          ;P/u next character
2E50 23         23250         INC     HL              ;Bump
2E51 CD2A2F     23260         CALL    PRSPC           ;Xfer extension
                23270 ;
                23280 ;       Was a drivespec entered ?
                23290 ;
2E54 FE3A       23300 DIR2    CP      ':'             ;Drive entered?
2E56 010700     23310         LD      BC,7            ;St = 0, terminating = 7
2E59 2009       23320         JR      NZ,DIR3         ;No - use drive 0
```

LBDIRC - Initialization Code

```
                    23330 ;
                    23340 ;       Check if char following is a legal drive #
                    23350 ;
2E5B CDE72E         23360         CALL    CKDSPEC         ;Legal Drive field ?
2E5E 20C8           23370         JR      NZ,ILLDRV       ;Illegal - abort
2E60 FE08           23380         CP      8               ;Trap DIR :8
2E62 28C4           23390         JR      Z,ILLDRV
                    23400 ;
                    23410 ;       B = Start drv #, C = Term drv # - save 'em
                    23420 ;
2E64 78             23430 DIR3    LD      A,B             ;Save starting drive
2E65 32DE2E         23440         LD      (DIR3A+1),A
2E68 91             23450         SUB     C               ;Set Specific Drive flag
2E69 323626         23460         LD      (SPECIF+1),A
2E6C 79             23470         LD      A,C             ;Save term drive
2E6D 322B26         23480         LD      (TERMDRV+1),A
                    23490 ;
                    23500 ;       Command line parsed - check available mem
                    23510 ;
2E70 FDCB024E       23520         BIT     1,(IY+CFLAG$)   ;Called from @CMNDR?
2E74 210000         23530         LD      HL,0            ;Set SORT (O) parm = 0
2E77 2803           23540         JR      Z,GETHI         ;No - fine
                    23550 ;
                    23560 ;       Executing from @CMNDR - Turn off SORT
                    23570 ;
2E79 22F926         23580         LD      (SORTPRM+1),HL
                    23590 ;
                    23600 ;       Pick up Current HIGH$, & set max mem to use
                    23610 ;
2E7C 45             23620 GETHI   LD      B,L             ;B=0
2E7D                23630         @@HIGH$
2E7D 3E64           00052         LD      A,100
2E7F EF             00053         RST     40
2E80 11DFFF         23640         LD      DE,-33          ;Subtract 33 from it
2E83 19             23650         ADD     HL,DE
2E84 22DB2D         23660         LD      (MAXMEM),HL     ;Stuff in maximum memory
                    23670 ;
                    23680 ;       Turn on N parm if P parm specified
                    23690 ;
2E87 2AD329         23700         LD      HL,(PPARM+1)    ;P/u P-parm
2E8A 7C             23710         LD      A,H             ;Specified ?
2E8B B5             23720         OR      L
2E8C 2803           23730         JR      Z,GTDATE        ;No - don't change N
2E8E 22722A         23740         LD      (NPARM+1),HL    ;Turn on N-parm
                    23750 ;
                    23760 ;       Was the DATE parameter specified ?
                    23770 ;
2E91 3AE72B         23780 GTDATE  LD      A,(DRESP)       ;Check out response
2E94 B7             23790         OR      A               ;Any response ?
2E95 2846           23800         JR      Z,DIR3A         ;None entered - no date
                    23810 ;
                    23820 ;       Something was specified - Check type
                    23830 ;
2E97 210000         23840 DATPRM  LD      HL,$-$          ;P/u date
2E9A CB77           23850         BIT     6,A             ;Flag input ?
2E9C 280C           23860         JR      Z,CHKSTR        ;No - must be string
                    23870 ;
                    23880 ;       Flag input - if YES, then use today's date
                    23890 ;
```

LBDIRC - Initialization Code

```
2E9E 7C       23900          LD      A,H             ;DATE = OFF ?
2E9F B5       23910          OR      L
2EAØ 283B     23920          JR      Z,DIR3A         ;Yes - ignore it
              23930 ;
              23940 ;        DATE parameter entered - get today's date
              23950 ;
2EA2 21312D   23960          LD      HL,TDATE        ;HL => Todays Date
2EA5 E5       23970          PUSH    HL              ;Save position
2EA6          23980          @@DATE                  ;Get today's date
2EA6 3E12     ØØØ54          LD      A,18
2EA8 EF       ØØØ55          RST     4Ø
2EA9 E1       23990          POP     HL              ;HL => Today's Date
              24000 ;
              24010 ;        Display dates before "-mm/dd/yy" ?
              24020 ;
2EAA 7E       24030 CHKSTR   LD      A,(HL)          ;P/u first char
2EAB FE2D     24040          CP      '-'             ;"to-" ?
2EAD 2815     24050          JR      Z,CKTO          ;Yes - do it
              24060 ;
              24070 ;        Not before - set flag accordingly
              24080 ;
2EAF 3E8Ø     24090          LD      A,8ØH           ;Set from bit
2EB1 32D72D   24100          LD      (FTFLG),A       ;Note from entered
              24110 ;
              24120 ;        Pack Date entry
              24130 ;
2EB4 CD492F   24140          CALL    PAKDAT          ;Pack the date entry
2EB7 ED43DD2D 24150          LD      (FMPAKD),BC     ;Stuff away date
              24160 ;
              24170 ;        End of first date ?
              24180 ;
2EBB 7E       24190          LD      A,(HL)          ;P/u terminator
2EBC FE22     24200          CP      '"'             ;End of date ?
2EBE 2811     24210          JR      Z,FRCTO         ;Yes - use spec'd date
              24220 ;
              24230 ;        Is there a to "-" symbol following date ?
              24240 ;
2ECØ FE2D     24250          CP      '-'             ;Check for "-to"
2EC2 2Ø19     24260          JR      NZ,DIR3A        ;No - check if legal
              24270 ;
              24280 ;        Is there a date following ?
              24290 ;
2EC4 23       24300 CKTO     INC     HL              ;Bypass the '-'
2EC5 7E       24310          LD      A,(HL)          ;P/u next char
2EC6 FE22     24320          CP      '"'             ;End of parm ?
2EC8 2813     24330          JR      Z,DIR3A         ;Yes - use that date
              24340 ;
2ECA FEØD     24350          CP      CR              ;End of parm ?
2ECC 28ØF     24360          JR      Z,DIR3A         ;Yes - use that date
              24370 ;
              24380 ;        Something following - parse date
              24390 ;
2ECE CD492F   24400          CALL    PAKDAT          ;Pack Date
              24410 ;
              24420 ;        Stuff in "TO" packed date & set TO flag
              24430 ;
2ED1 3AD72D   24440 FRCTO    LD      A,(FTFLG)       ;P/u From-To Flag
2ED4 F6Ø1     24450          OR      1               ;Set TO bit
2ED6 32D72D   24460          LD      (FTFLG),A       ;Stuff in flag
```

LBDIRC - Initialization Code

```
2ED9 ED43DF2D 24470         LD      (TOPAKD),BC     ;Stuff for later
              24480 ;
              24490 ;       P/u starting drive #, & init page counter
              24500 ;
2EDD ØEØØ     24510 DIR3A   LD      C,$-$           ;P/u starting drive
2EDF 3E16     24520         LD      A,22            ;Max lines to dsply
2EE1 32662A   24530         LD      (CKPAGE+1),A    ;Stuff in counter
2EE4 C33324   24540         JP      DIR4            ;Directory Start
              24550 ;
              24560 ;       CKDSPEC - Check if a drive spec field is legal
              24570 ;       HL => Drive specification Field
              24580 ;       Z - Set if Drive spec Field is Legal
              24590 ;       B <= Starting Drive # (Ø-7)
              24600 ;       C <= Terminating Drive # (Ø-7)
              24610 ;
2EE7 7E       24620 CKDSPEC LD      A,(HL)          ;P/u first character
2EE8 FE2D     24630         CP      '-'             ;"TO" or "NOT" ?
2EEA 2ØØC     24640         JR      NZ,NOTDASH      ;No - check if drive #
              24650 ;
              24660 ;       Char is a "-" ---- Could be "TO" or "NOT"
              24670 ;
2EEC CD222F   24680         CALL    LEGDRV          ;Legal Drive Number ?
2EEF D8       24690         RET     C               ;No - RETurn NZ
              24700 ;
              24710 ;       Legal Drive # - Next char must be a term
              24720 ;
2EFØ 4F       24730         LD      C,A             ;C = Terminating Drive
2EF1 23       24740         INC     HL              ;HL => Following char
2EF2 CDØB2F   24750         CALL    TERM            ;Does a term follow ?
2EF5 Ø6ØØ     24760         LD      B,Ø             ;B  default start Ø
2EF7 C9       24770         RET                     ;RETurn Z or NZ
              24780 ;
              24790 ;       Is the First character a legal drive # ?
              24800 ;
2EF8 CD232F   24810 NOTDASH CALL    LEGDRV1         ;Legal drive (Ø-7) ?
2EFB D8       24820         RET     C               ;No - RETurn NZ (ex 8)
2EFC 47       24830         LD      B,A             ;Set B = Starting Drive
2EFD 4F       24840         LD      C,A             ;Set C = Terminator
              24850 ;
              24860 ;       Legal Drive - a "-" or term MUST follow
              24870 ;
2EFE 23       24880         INC     HL              ;Bump to next char
2EFF 7E       24890         LD      A,(HL)          ;If next char is not a
2FØØ FE2D     24900         CP      '-'             ;  "-", RETurn Z or NZ
2FØ2 2811     24910         JR      Z,CKTDRIV       ;  depending on next char.
2FØ4 CDØB2F   24920         CALL    TERM            ;Legal terminator ?
2FØ7 C2282E   24930         JP      NZ,ILLDRV       ;No - Illegal Drive #
2FØA C9       24940         RET                     ;Yes - Return
              24950 ;
              24960 ;       Is the character a terminator ?
              24970 ;
2FØB 7E       24980 TERM    LD      A,(HL)          ;P/u char
2FØC FE2Ø     24990         CP      ' '             ;Space is legal
2FØE C8       25000         RET     Z               ;RETurn Z if space
2FØF FEØD     25010         CP      CR              ;CR is legal
2F11 C8       25020         RET     Z               ;RETurn Z if CR
2F12 FE28     25030         CP      '('             ;Paren is legal
2F14 C9       25040         RET                     ;RETurn w/ condition
              25050 ;
```

LBDIRC - Initialization Code

```
                25Ø6Ø ;       Next char must be a valid drive # or term
                25Ø7Ø ;
2F15 CD222F     25Ø8Ø CKTDRIV CALL    LEGDRV          ;Legal Drive # ?
2F18 ØEØ7       25Ø9Ø         LD      C,7             ;C = Default term drive 7
2F1A 38EF       251ØØ         JR      C,TERM          ;Not drv # - ck for term
                2511Ø ;
                2512Ø ;       Make sure Term Drive # > or = Start Drive #
                2513Ø ;
2F1C 4F         2514Ø         LD      C,A             ;Set C = Term drive #
2F1D B8         2515Ø         CP      B               ;> or = start drive # ?
2F1E D8         2516Ø         RET     C               ;Less - Return
                2517Ø ;
                2518Ø ;       Drive span range good - make sure term legal
                2519Ø ;
2F1F 23         252ØØ         INC     HL              ;Bump ptr
2F2Ø 18E9       2521Ø         JR      TERM            ;RETurn Z or NZ
                2522Ø ;
                2523Ø ;       LEGDRV - Is a character a legal drive #
                2524Ø ;       HL => One before Character to check
                2525Ø ;       HL <= Character in question
                2526Ø ;       A  <= Drive Number (Ø-7)
                2527Ø ;       CF <= Set if Character is not a legal drive #
                2528Ø ;
2F22 23         2529Ø LEGDRV  INC     HL              ;Bump to next
2F23 7E         253ØØ LEGDRV1 LD      A,(HL)          ;P/u char
2F24 D63Ø       2531Ø         SUB     'Ø'             ;Convert to binary
2F26 FEØ8       2532Ø         CP      7+1             ;Greater than "7" ?
2F28 3F         2533Ø         CCF                     ;C - Illegal
2F29 C9         2534Ø         RET                     ;RETurn with condition
                2535Ø ;
                2536Ø ;       PRSPC - Parse a line and stuff in buffer
                2537Ø ;       HL => Source Buffer
                2538Ø ;       DE => Destination of converted field
                2539Ø ;       B = # of characters to parse
                254ØØ ;
2F2A FE24       2541Ø PRSPC   CP      '$'             ;Wild character?
2F2C 2814       2542Ø         JR      Z,PS2           ;Yes - stuff in buff
2F2E FE41       2543Ø         CP      'A'             ;Alphabetic ?
2F3Ø 3ØØ6       2544Ø         JR      NC,PS1          ;Maybe - convert to U/C
                2545Ø ;
                2546Ø ;       Is the character a numeric value (Ø-9) ?
                2547Ø ;
2F32 FE3A       2548Ø         CP      '9'+1           ;Greater than "9" ?
2F34 DØ         2549Ø         RET     NC              ;Yes - return
2F35 FE3Ø       255ØØ         CP      'Ø'             ;Less than "Ø" ?
2F37 D8         2551Ø         RET     C               ;Yes - return
                2552Ø ;
                2553Ø ;       Convert character to Upper Case
                2554Ø ;
2F38 FE61       2555Ø PS1     CP      'a'             ;Lower case alpha ?
2F3A 38Ø6       2556Ø         JR      C,PS2           ;No - stuff in buffer
2F3C FE7B       2557Ø         CP      'z'+1
2F3E 3ØØ2       2558Ø         JR      NC,PS2
2F4Ø CBAF       2559Ø         RES     5,A             ;Convert to U/C
                256ØØ ;
                2561Ø ;       Put char in buffer, & bump cmd & buffer ptrs
                2562Ø ;
2F42 12         2563Ø PS2     LD      (DE),A          ;Stuff in buffer
2F43 13         2564Ø PS3     INC     DE              ;Bump
```

LBDIRC - Initialization Code

```
2F44 7E       25650          LD      A,(HL)          ;P/u command buff char
2F45 23       25660          INC     HL              ;Bump
2F46 10E2     25670          DJNZ    PRSPC           ;  B times
2F48 C9       25680          RET
              25690 ;
              25700 ;        PAKDAT - Pack Date & Stuff into buffer
              25710 ;        HL => Buffer containing Date string
              25720 ;        BC <= Packed Date in lsb,msb format
              25730 ;
2F49 7E       25740 PAKDAT   LD      A,(HL)          ;P/u character
2F4A 0E2F     25750          LD      C,'/'           ;Init separator
              25760 ;
              25770 ;        Is the date a valid entry ?
              25780 ;
2F4C CD8E2F   25790          CALL    PARSDAT         ;Parse entry
2F4F C21124   25800          JP      NZ,BADFMT       ;Abort on format error
              25810 ;
              25820 ;        If year = 1980 or 84 then set FEB = 29 days
              25830 ;
2F52 EB       25840          EX      DE,HL           ;Save command ptr
2F53 3AE12D   25850          LD      A,(LILBUF$)     ;P/u year (80-87)
2F56 E603     25860          AND     3               ;Mask off bits 7-2
2F58 21672D   25870          LD      HL,MAXDAYS+1    ;Set Feb to have 29 days
2F5B 2001     25880          JR      NZ,NOTLEAP      ;No - don't inc it
2F5D 34       25890          INC     (HL)            ;Leap year - inc max days
              25900 ;
              25910 ;        Check Range of month - must be 1-12
              25920 ;
2F5E 3AE32D   25930 NOTLEAP  LD      A,(LILBUF$+2)   ;P/u month
2F61 3D       25940          DEC     A               ;Set month = 1-11
2F62 FE0C     25950          CP      12              ;Valid month ?
2F64 D21124   25960          JP      NC,BADFMT       ;Abort if 0 or >12
              25970 ;
              25980 ;        Valid month - point HL to max days/month
              25990 ;
2F67 2B       26000          DEC     HL              ;Point before JAN entry
2F68 85       26010          ADD     A,L             ;Add the month
2F69 6F       26020          LD      L,A             ;HL => max days for month
2F6A 3001     26030          JR      NC,NOINC        ;Bump H if C set
2F6C 24       26040          INC     H
              26050 ;
              26060 ;        Check if day entry is valid
              26070 ;
2F6D 3AE22D   26080 NOINC    LD      A,(LILBUF$+1)   ;P/u day entry
2F70 3D       26090          DEC     A               ;Reduce for test (0->FF)
2F71 BE       26100          CP      (HL)            ;More than max days ?
2F72 D21124   26110          JP      NC,BADFMT       ;Go if too large (or 0)
              26120 ;
              26130 ;        Pick up month from buffer
              26140 ;
2F75 21E32D   26150          LD      HL,LILBUF$+2    ;HL => Month
2F78 7E       26160          LD      A,(HL)          ;P/u month
2F79 47       26170          LD      B,A             ;Save month
              26180 ;
              26190 ;        Transfer Day to Bit positions 3-7
              26200 ;
2F7A 2B       26210          DEC     HL              ;HL => Day
2F7B 7E       26220          LD      A,(HL)          ;P/u day
2F7C 2B       26230          DEC     HL              ;HL => Year
```

LBDIRC - Initialization Code

```
2F7D Ø7       26240          RLCA                    ;Shift day to 3-7
2F7E Ø7       26250          RLCA
2F7F Ø7       26260          RLCA
2FØØ 4F       26270          LD      C,A             ;Save in C
              26280 ;
              26290 ;        Pick up year & convert to binary (Ø-7)
              26300 ;
2F81 7E       26310          LD      A,(HL)          ;P/u year
2F82 D65Ø     26320          SUB     8Ø              ;Adjust for offset
2F84 3ØØ1     26330          JR      NC,GDATE        ;If entry < 198Ø,
2F86 AF       26340          XOR     A               ;  then use 198Ø
              26350 ;
              26360 ;        Shift year into positions 7-5
              26370 ;
2F87 ØF       26380 GDATE    RRCA                    ;Shift into bits 7-5
2F88 ØF       26390          RRCA
2F89 ØF       26400          RRCA
              26410 ;
              26420 ;        Merge with month & Return
              26430 ;
2F8A BØ       26440          OR      B               ;  & merge with month
2F8B 47       26450          LD      B,A             ;Stuff in B
2F8C EB       26460          EX      DE,HL           ;HL => Buffer
2F8D C9       26470          RET                     ;RETurn
              26480 ;
              26490 ;        PARSDAT - Parse TIME/DATE string entry
              26500 ;        HL => Buffer containing string to parse
              26510 ;        C  => Delimiter ("/" = DATE, ":" = TIME)
              26520 ;        LILBUF$-LILBUF$+2 <= Data in compressed format
              26530 ;        Z  - Set if successful
              26540 ;
2F8E 11E32D   26550 PARSDAT  LD      DE,LILBUF$+2    ;Point to buf end
2F91 Ø6Ø3     26560          LD      B,3             ;Process 3 fields
              26570 ;
              26580 ;        Parse a field - Return NZ if bad
              26590 ;
2F93 D5       26600 PRS1     PUSH    DE              ;Save pointer
2F94 CDA32F   26610          CALL    PRS2            ;Get a digit pair
2F97 D1       26620          POP     DE              ;Recover pointer
2F98 CØ       26630          RET     NZ              ;Ret if bad digit pair
              26640 ;
              26650 ;        Good field - Stuff in buff, dec ptr, & count
              26660 ;
2F99 12       26670          LD      (DE),A          ;  else stuff the value
2F9A 1B       26680          DEC     DE              ;Backup the pointer
2F9B Ø5       26690          DEC     B               ;Loop countdown
2F9C C8       26700          RET     Z               ;Do for 3 fields
              26710 ;
              26720 ;        Parsed a field - is the separator valid ?
              26730 ;
2F9D 7E       26740          LD      A,(HL)          ;P/u separator
2F9E 23       26750          INC     HL              ;Bump pointer
2F9F B9       26760          CP      C               ;Correct ?
2FAØ 28F1     26770          JR      Z,PRS1          ;Yes - continue
2FA2 C9       26780          RET                     ;No - RET NZ
              26790 ;
              26800 ;        PRS2 - Parse a digit pair at HL
              26810 ;
2FA3 CDB92F   26820 PRS2     CALL    PRS4            ;Get a digit
```

LBDIRC - Initialization Code

```
2FA6 3ØØF     26830         JR      NC,PRS3         ;Illegal - clr stc & RET
              26840 ;
              26850 ;       Legal Digit - Multiply by 1Ø
              26860 ;
2FA8 5F       26870         LD      E,A             ;Multiply by ten
2FA9 Ø7       26880         RLCA                    ;X 2
2FAA Ø7       26890         RLCA                    ;X 4
2FAB 83       26900         ADD     A,E             ;X 5
2FAC Ø7       26910         RLCA                    ;X 1Ø
2FAD 5F       26920         LD      E,A             ;Stuff in E
              26930 ;
              26940 ;       Get another digit
              26950 ;
2FAE CDB92F   26960         CALL    PRS4            ;Get ones digit
2FB1 3ØØ4     26970         JR      NC,PRS3         ;Bad - return NZ
              26980 ;
              26990 ;       Legal digit - Add to tens digit & set Z flag
              27000 ;
2FB3 83       27010         ADD     A,E             ;Accumulate new digit
2FB4 5F       27020         LD      E,A             ;Save 2-digit value
2FB5 BF       27030         CP      A               ;Clear flags
2FB6 C9       27040         RET                     ;Return Z
              27050 ;
              27060 ;       Force NZ & Return
              27070 ;
2FB7 B7       27080 PRS3    OR      A               ;Set NZ
2FB8 C9       27090         RET                     ;RETurn
              27100 ;
              27110 ;       Pick up a digit and convert to binary
              27120 ;
2FB9 7E       27130 PRS4    LD      A,(HL)          ;P/u a digit &
2FBA 23       27140         INC     HL              ;  bump ptr
2FBB D63Ø     27150         SUB     'Ø'             ;Convert to binary
2FBD FEØA     27160         CP      1Ø              ;Legal ?
2FBF C9       27170         RET                     ;C - legal, NC - illegal
              27180 ;
2FCØ          27190 ENDMEM  EQU     $
              ØØ310 ;
              ØØ320 ;       Bytes Free =
              ØØ330 ;
ØØ4Ø          ØØ340 FREE$   EQU     3ØØØH-ENDMEM
              ØØ350 ;
              ØØ360         IFGT    $,2FFFH
              ØØ370         ERR     'LIB memory region overflow
              ØØ380         ENDIF
              ØØ390 ;
2FCØ          ØØ400         SUBTTL  <>
24ØØ          ØØ410         END     ENTRY
```

$JPØ	25A3 $JP1	26F5 @@1	ØØØØ
@@2	ØØØØ @@3	ØØØØ @@4	ØØØØ
@MOD2	ØØØØ @MOD4	FFFF ABB	ØØ1Ø
ABORT	2417 ABORT3	242F AGAIN	2B54
ALLØ1	2825 ALLØ2	2826 ALLØ3	282F
ALLØ4	2869 ALLØ6	28EA ALLØ8	2924
ALLØ9	2945 ALLØ9A	295B ALL1Ø	296Ø
ALL11	2961 ALL14	2964 ALL15	2977
ALL16	2989 ALRPRT	2741 AP	ØØ27
APARM	28Ø5 BACK	2B84 BADFMT	2411
BADFMT$	2D9E BLANKS	2D3A BLKHASH	4296
BREAK	ØØ8Ø BS	ØØØ8 BUF2	2FØØ
BYTOUT	29CB BYTOUT2	29CA CALCK	29E8
CALCK1	29EC CALCK2	29FC CATBGN	24Ø3
CFLAG$	ØØØ2 CHKSTR	2EAA CK2HIT	26C6
CKDIR1	2665 CKDIR3	2688 CKDSPEC	2EE7
CKHIT	25A6 CKHIT1	25A7 CKHIT2	25AB
CKHIT4	2634 CKHIT5	2645 CKHIT6	2651
CKITOUT	2E2B CKMOD	2694 CKNAM	26A2
CKNAM1	26AC CKNAM2	26B8 CKNAM2A	26C9
CKNAM2B	26E4 CKPAGE	2A65 CKPAW1	2A9C
CKPAW2	2A9F CKPAWS	2A93 CKTDRIV	2F15
CKTITL	2AB1 CKTO	2EC4 CLRLP	2537
COUNT	25D5 COUNTM1	2B33 CPHLDE	2A13
CR	ØØØD CVDØ	2A4A CVD1	2A4F
CVD1Ø	2A3E CVD1ØØ	2A38 CVD2	2A63
CVDDEC	2A2A CYCLE	2B38 CYLCNT	2C1B
D79EQ	25DE D79L	2ACF DATBUF	2C4D
DATEFLD	2CD4 DATPRM	2E97 DEN	2BFF
DENSITY	2C24 DFLAG$	ØØØ3 DIR	2DD9
DIRØ	2E35 DIR1	2E43 DIR2	2E54
DIR3	2E64 DIR3A	2EDD DIR4	2433
DIRA	2DE2 DIRPTR	2DD9 DIR_Ø	28BØ
DIVIDE	293Ø DODSP	272D DOLDIR	24AA
DONAM1	274C DONAM2	2759 DONAM3	275D
DONAM4	2769 DONAM5	2774 DONAM5A	278D
DONAM6	27DA DONAM7	27DC DONAM8	27FD
DONAM9	28ØD DONTDEC	2874 DONTINC	2884
DRESP	2BE7 DRIVE	2CØE DSTRING	2CØ7
DUBDEN	249D ENDLINE	2819 ENDMEM	2FCØ
ENTRY	24ØØ EOFBYTE	2895 ERR32	241C
ETX	ØØØ3 EXIT	2429 EXITSRT	2BB1
EXTENTS	2CCD FDISP	2CE9 FILES	25CF
FILFLAG	2DD8 FINIS	2BAF FLAG	ØØ4Ø
FMPAKD	2DDD FPLP	2EØ5 FRCTO	2ED1
FREE$	ØØ4Ø FS1	24CB FS2	24CC
FS3	24D1 FSTRING	2CDE FTFLG	2DD7
FUSED	2CFA GAT	2EØØ GDATE	2F87
GDCKDRV	2475 GETHI	2E7C GETPRM	2E11
GOODCYL	25Ø7 GPCLOOP	24FØ GTDATE	2E91
GTUSED	25ØE HARD	2CØ3 HEADING	2C57
HIT	2EØØ HUNDTAB	2D72 ILLDATE	2579
ILLDRV	2E28 IOER1	29C6 IOERR	241E
IOERR5	2A7F IPARM	268C KEEPEOF	289E
KFLAG	2A93 KFLAG$	ØØØA KFLAG1	2AAD
KFREE	2C3Ø KPOSS	2C3C KSIZE	2CCØ
LEGDRV	2F22 LEGDRV1	2F23 LF	ØØØA
LILBUF$	2DE1 LINOUT	29B9 LKLOOP	25Ø1

LRL	2CAE	MATCH	2733	MAXDAYS	2D66
MAXMEM	2DDB	MEMORY	3ØØØ	MFLG	26C3
MONTBL	2DAE	NAME	2C11	NDRIVE	2D22
NEXTDRV	2472	NODISK	2D1B	NOINC	2F6D
NOMEM	24ØD	NOMEM$	2D7A	NOPRT	29DE
NOTAP	2619	NOTDASH	2EF8	NOTDUB	24E9
NOTITLE	2A87	NOTLEAP	2F5E	NO_DISK	245F
NPARM	2A71	NUM	ØØ8Ø	OFFSET	2CBB
OUTCHR	294Ø	OUTMOD	2937	OUTSPC	29EØ
PAKDAT	2F49	PARSDAT	2F8E	PAR_ERR	ØØ2C
PLEVEL	2CA7	POP	2B9Ø	PPARM	29D2
PRMTBL$	2BC4	PROTS$	2D46	PRS1	2F93
PRS2	2FA3	PRS3	2FB7	PRS4	2FB9
PRSPC	2F2A	PS1	2F38	PS2	2F42
PS3	2F43	RECORDS	2CB3	REPEAT	2B5A
RESKFL	2AA8	RESTPTR	2E17	SAVESP	242C
SBUFFER	2985	SFLAG	2A77	SFLAG$	ØØ12
SHELL	2B32	SKIPLOC	2517	SORT1	2AEF
SORT2	2BØF	SORT2A	2B1F	SORT3	2B23
SORTIT	2ADF	SORTPRM	26F8	SPARM	267E
SPECIF	2635	SPUSED	2D1Ø	STARS	2DD2
STORI	2B5B	STORJ	2B55	STORK	2BBA
STORM	2B39	STR	ØØ2Ø	SWAP	2B94
TAB	ØØØ9	TDATE	2D31	TERM	2FØB
TERMDRV	262A	TFILES	25F6	TOPAKD	2DDF
TOTGRNS	26Ø1	TSTSIZ	2886	UNPACK	2A19
VFLAG$	ØØ15	@@ABORT	6E3D	@@ADTSK	6EDØ
@@BANK	73E8	@@BKSP	7ØC8	@@BREAK	73FE
@@CHNIO	6E28	@@CKBRKC	744C	@@CKDRV	6F24
@@CKEOF	7ØDD	@@CKTSK	6EBB	@@CLOSE	7ØB3
@@CLS	7436	@@CMNDI	6E67	@@CMNDR	6E7C
@@CTL	6C8C	@@DATE	6DFE	@@DCSTAT	6F63
@@DEBUG	6EA6	@@DECHEX	7368	@@DIRRD	72D5
@@DIRWR	72EA	@@DIV16	7353	@@DIV8	733E
@@DODIR	6F39	@@DSP	6C5Ø	@@DSPLY	6CFØ
@@ERROR	6E91	@@EXIT	6E52	@@FEXT	7242
@@FLAGS	73D2	@@FNAME	7257	@@FSPEC	722D
@@GATRD	72CØ	@@GATWR	72FF	@@GET	6C64
@@GTDCB	7281	@@GTDCT	726C	@@GTMOD	7296
@@HDFMT	7ØØB	@@HEX16	73A7	@@HEX8	7392
@@HEXDEC	737D	@@HIGH$	73BC	@@INIT	7Ø89
@@KBD	6CC8	@@KEY	6C3C	@@KEYIN	6CDC
@@KLTSK	6FØF	@@LOAD	72Ø3	@@LOC	7ØF2
@@LOF	71Ø7	@@LOGER	6D27	@@LOGOT	6D3C
@@MSG	6D73	@@MUL16	7329	@@MUL8	7314
@@OPEN	7Ø9E	@@PARAM	6DE9	@@PAUSE	6DD4
@@PEOF	711C	@@POSN	7131	@@PRINT	6D88
@@PRT	6CAØ	@@PUT	6C78	@@RAMDIR	6F4E
@@RDSEC	6FE1	@@RDSSC	72AB	@@READ	7146
@@REMOV	7Ø74	@@RENAM	7Ø5F	@@REW	715B
@@RMTSK	6EE5	@@RPTSK	6EFA	@@RREAD	717Ø
@@RSLCT	6FCC	@@RSTOR	6F8D	@@RUN	7218
@@RWRIT	7185	@@SEEK	6FB7	@@SEEKSC	719A
@@SKIP	71AF	@@SLCT	6F78	@@STEPI	6FA2
@@TIME	6E13	@@VDCTL	6DBF	@@VER	71C4
@@VRSEC	6FF6	@@WEOF	71D9	@@WHERE	6CB4
@@WRITE	71EE	@@WRSEC	7Ø2Ø	@@WRSSC	7Ø35
@@WRTRK	7Ø4A				

24ØØ is the transfer address

ØØØØØ Total errors

NOTES:

NOTES:

Command: DO

Library: SYS6/SYS

ISAM # : 91H

```
                00100 ;LBDO/ASM - Library 'DO' command
0000            00110         TITLE   <DO - LS-DOS 6.2>
                00120 ;
00C0            00130 JFCB$   EQU     0C0H            ;Low core EQU*
                00140 ;
                00150 ;
0000            00160 SMALL   EQU     0
000D            00170 CR      EQU     13
0000            00180 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00190 ;
2400            00200         ORG     2400H
                00210 ;
2400            00220 DO      EQU     $
                00230 ;
                00240 ;       Note: The first 80 bytes (until PARSINP) are
                00250 ;       Used as a line buffer during processing.
                00260 ;
2400            00270 JCLBUF2 EQU     $
2400 ED737929   00280         LD      (SPSAV+1),SP    ;Save stack pointer
                00290 ;
                00300         IF      SMALL
                00310         JR      NOCPLS          ;No compile if Small
                00320         ENDIF
2404 226D24     00330         LD      (INBUF+1),HL    ;Save start of command
                00340 ;
                00350 *LIST   OFF
                00370 *LIST   ON
2407            00380         @@FLAGS                 ;Get flag table pointer
2407 3E65       00001         LD      A,101
2409 EF         00002         RST     40
240A 7E         00390         LD      A,(HL)
240B FE2A       00400         CP      '*'             ;Execute last DO file?
240D CAF024     00410         JP      Z,NOCPL2
2410 FE3D       00420         CP      '='             ;Execute without compile?
2412 CAD924     00430         JP      Z,NOCPL
2415 FE24       00440         CP      '$'             ;Compile only?
2417 200A       00450         JR      NZ,GETSPEC
2419 32B424     00460         LD      (NOEXEC?+1),A
241C 23         00470         INC     HL
241D 7E         00480         LD      A,(HL)
241E FE20       00490         CP      ' '             ;Bypass space separator
2420 2001       00500         JR      NZ,GETSPEC      ;  if present
2422 23         00510         INC     HL
2423 117C29     00520 GETSPEC LD      DE,DOFCB        ;Get DO filespec
2426            00530         @@FSPEC
2426 3E4E       00003         LD      A,78
2428 EF         00004         RST     40
2429 C24829     00540         JP      NZ,SPCREQ       ;Go if bad/missing filespec
242C E5         00550         PUSH    HL              ;Save INBUF$ pointer
242D 21D729     00560         LD      HL,SYSJCL+7     ;Default ext to "/JCL"
2430            00570         @@FEXT
2430 3E4F       00005         LD      A,79
2432 EF         00006         RST     40
2433 21002C     00580         LD      HL,INPBUF       ;Open DO file
2436 45         00590         LD      B,L             ;LRL=256
2437 FDCB12C6   00600         SET     0,(IY+'S'-'A')  ;Inhibit file open bit
243B            00610         @@OPEN
243B 3E3B       00007         LD      A,59
```

```
243D EF       00008          RST     40
243E C23D29   00620          JP      NZ,IOERR        ;Jump on open error
2441 CDF524   00630          CALL    MOVFCB          ;Move SYSTEM/JCL into FCB
2444 11C000   00640          LD      DE,JFCB$        ;Init FCB pointer
2447 21002D   00650          LD      HL,OUTBUF
244A          00660          @@INIT
244A 3E3A     00009          LD      A,58
244C EF       00010          RST     40
244D C25C29   00670          JP      NZ,DSKFUL       ;Jump on error
2450 E1       00680          POP     HL              ;Rcvr pointer to INBUF$
              00690 ;
              00700 ;        Routine to parse a command line
              00710 ;
2451 7E       00720 PARSINP  LD      A,(HL)          ;P/u line char
2452 FE0D     00730          CP      CR              ;End of line?
2454 CA4825   00740          JP      Z,TSTLBL
2457 23       00750          INC     HL              ;Bump pointer
2458 CD2D29   00760          CALL    CKSPCOM         ;Ignore spaces & commas
245B 28F4     00770          JR      Z,PARSINP
245D FE28     00780          CP      '('             ;Beginning of parms?
245F CA0125   00790          JP      Z,PARAM
2462 FE3B     00800          CP      ';'             ;Line continuation?
2464 C26029   00810          JP      NZ,PRMERR
2467 0E3F     00820          LD      C,'?'           ;Prompt for line continue
2469          00830          @@DSP
2469 3E02     00011          LD      A,2
246B EF       00012          RST     40
246C 210000   00840 INBUF    LD      HL,$-$          ;Input continuation line
246F 2D       00850          DEC     L               ;Backup to start
2470 2D       00860          DEC     L
2471 01004F   00870          LD      BC,79<8         ;Max 79 chars input
2474          00880          @@KEYIN
2474 3E09     00013          LD      A,9
2476 EF       00014          RST     40
2477 DA6029   00890          JP      C,PRMERR        ;Jump if break
247A          00900          @@LOGER                 ;Log the line
247A 3E0B     00015          LD      A,11
247C EF       00016          RST     40
247D 18D2     00910          JR      PARSINP         ;Go parse it
              00920 ;
              00930 ;        Routine to move to higher level nest
              00940 ;
247F 2ABA2A   00950 UNNEST   LD      HL,(NESTPTR)    ;Shift the last nest's
2482 2B       00960          DEC     HL              ;  FCB into FCB area
2483 119B29   00970          LD      DE,DOFCB+31
2486 012000   00980          LD      BC,32
2489 EDB8     00990          LDDR
248B 23       01000          INC     HL
248C 22BA2A   01010          LD      (NESTPTR),HL    ;Reset current FCB ptr
248F 117C29   01020          LD      DE,DOFCB        ;Reread last sector of
2492          01030          @@RREAD                 ;  nested FCB
2492 3E45     00017          LD      A,69
2494 EF       00018          RST     40
2495 C23D29   01040          JP      NZ,IOERR
2498 C9       01050          RET
              01060 ;
2499 2ABA2A   01070 CKNEST   LD      HL,(NESTPTR)    ;P/u current FCB pointer
249C 11BC2A   01080          LD      DE,NESTFCB      ;Is it the first nest?
249F AF       01090          XOR     A
24A0 ED52     01100          SBC     HL,DE
24A2 2806     01110          JR      Z,CPLFIN        ;Jump if so & exit
```

```
24A4 CD7F24    01120          CALL    UNNEST          ;  processing
24A7 C37E25    01130          JP      CPLJCL
               01140 ;
               01150 ;        Finished compilation - Close 'er up
               01160 ;
24AA 11C000    01170 CPLFIN   LD      DE,JFCB$        ;Close SYSTEM/JCL file
24AD           01180          @@CLOSE
24AD 3E3C      00019          LD      A,60
24AF EF        00020          RST     40
24B0 C23D29    01190          JP      NZ,IOERR
24B3 3E00      01200 NOEXEC?  LD      A,0             ;Set to non-zero on
24B5 B7        01210          OR      A               ;  compile only
24B6 210000    01220          LD      HL,0
24B9 C0        01230          RET     NZ              ;Exit on compile only
               01240          ENDIF
               01250 *LIST    ON
               01260 ;
24BA 11C000    01270 CPLFIN1  LD      DE,JFCB$        ;Point to SYSTEM/JCL FCB
24BD 210000    01280          LD      HL,0            ;Correct bufptr later
24C0 45        01290          LD      B,L             ;LRL=256
24C1 FDCB12C6  01300          SET     0,(IY+'S'-'A')  ;Inhibit file open bit
24C5           01310          @@OPEN                  ;Open it up
24C5 3E3B      00021          LD      A,59
24C7 EF        00022          RST     40
24C8 C23D29    01320          JP      NZ,IOERR        ;Jump on error
24CB ED4BC600  01330          LD      BC,(JFCB$+6)    ;Get SBUFF$
24CF           01340          @@DIRRD
24CF 3E57      00023          LD      A,87
24D1 EF        00024          RST     40
24D2 7C        01350          LD      A,H             ;Stuff high order to
24D3 32C400    01360          LD      (JFCB$+4),A     ;  use for JFCB$ buffer
24D6 3E9D      01370          LD      A,9DH           ;Call SYS11, entry 1
24D8 EF        01380          RST     28H
               01390 ;
               01400 ;        Process execution without compilation
               01410 ;
24D9 23        01420 NOCPL    INC     HL
24DA 7E        01430 NOCPLS   LD      A,(HL)          ;Bypass space separator
24DB FE20      01440          CP      ' '             ;  if present
24DD 28FA      01450          JR      Z,NOCPL
24DF 11C000    01460 NOCPL1   LD      DE,JFCB$        ;Fetch DO filespec
24E2           01470          @@FSPEC
24E2 3E4E      00025          LD      A,78
24E4 EF        00026          RST     40
24E5 C24829    01480          JP      NZ,SPCREQ       ;Jump on error
24E8 21D729    01490          LD      HL,SYSJCL+7     ;Default to /JCL
24EB           01500          @@FEXT
24EB 3E4F      00027          LD      A,79
24ED EF        00028          RST     40
24EE 18CA      01510          JR      CPLFIN1         ;Go execute file
               01520 ;
               01530 *LIST    OFF
               01550 *LIST    ON
24F0 CDF524    01560 NOCPL2   CALL    MOVFCB          ;Execute SYSTEM/JCL
24F3 18C5      01570          JR      CPLFIN1         ;  file
               01580 ;
24F5 21D029    01590 MOVFCB   LD      HL,SYSJCL       ;Move SYSTEM/JCL into
24F8 11C000    01600          LD      DE,JFCB$        ;  FCB area
24FB 012000    01610          LD      BC,32
24FE EDB0      01620          LDIR
2500 C9        01630          RET
```

```
                    01640 ;
                    01650 ;       Found a parm entered
                    01660 ;
2501 CDE727         01670 PARAM   CALL    PARSNAM         ;Parse symbol -> current
2504 2014           01680         JR      NZ,PARAM1       ;Jump if bad symbol
2506 F5             01690         PUSH    AF              ;Save separator char
2507 3E00           01700 FNDLBL  LD      A,0             ;Test if a label
2509 B7             01710         OR      A               ;  was found
250A 2029           01720         JR      NZ,MOVLBL
250C CDB128         01730         CALL    FINDSYM         ;Search symbol table
250F CA6429         01740         JP      Z,MULDEF        ;Multiply defined if in
2512 CD9428         01750         CALL    MOVNAME         ;Add symbol to table
2515 F1             01760         POP     AF              ;Recover separator
2516 FE3D           01770         CP      '='             ;Assignment?
2518 2811           01780         JR      Z,PARAM2
251A CD2D29         01790 PARAM1  CALL    CKSPCOM         ;Ck space or comma
251D 28E2           01800         JR      Z,PARAM
251F FE29           01810         CP      ')'             ;Exit parm scan on
2521 CA5124         01820         JP      Z,PARSINP       ;  closing paren
2524 FE0D           01830         CP      CR              ;Also accept closing CR
2526 2820           01840         JR      Z,TSTLBL
2528 C36029         01850         JP      PRMERR          ;Else parm error
                    01860 ;
252B CDF227         01870 PARAM2  CALL    PARSVAL         ;Parse value into buf
252E F5             01880         PUSH    AF              ;Save separator char
252F CDA628         01890         CALL    MOVALUE         ;Symbol value into table
2532 F1             01900 GETSEP  POP     AF              ;Recover separator
2533 18E5           01910         JR      PARAM1          ;Loop
                    01920 ;
2535 E5             01930 MOVLBL  PUSH    HL
2536 219C29         01940         LD      HL,CURSYM       ;Pt to current sym buf
2539 11C529         01950         LD      DE,LBLSAV       ;  & save label for
253C 010800         01960         LD      BC,8            ;  later testing
253F EDB0           01970         LDIR
2541 AF             01980         XOR     A               ;Turn off "found label"
2542 320825         01990         LD      (FNDLBL+1),A
2545 E1             02000         POP     HL              ;Rcvr line ptr
2546 18EA           02010         JR      GETSEP          ;Back for more
                    02020 ;
                    02030 ;       Got to end of JCL command line
                    02040 ;
2548 3A5528         02050 TSTLBL  LD      A,(GOTLBL+1)    ;Was @LABEL a parm?
254B B7             02060         OR      A
254C 2830           02070         JR      Z,CPLJCL        ;If not, don't look
                    02080 ;
                    02090 ;       Find the procedure block named @LABEL
                    02100 ;
254E CD6126         02110 FINDLBL CALL    RDJCL           ;Read JCL line
2551 2811           02120         JR      Z,GOTLIN        ;Go if line read
2553 2ABA2A         02130         LD      HL,(NESTPTR)    ;See if nested
2556 11BC2A         02140         LD      DE,NESTFCB      ;  in an Include file
2559 AF             02150         XOR     A
255A ED52           02160         SBC     HL,DE
255C CA5429         02170         JP      Z,NOFIND        ;If not, lable not found
255F CD7F24         02180         CALL    UNNEST          ;  else continue search
2562 18EA           02190         JR      FINDLBL
                    02200 ;
2564 218C2B         02210 GOTLIN  LD      HL,JCLBUF1      ;Pt to start
2567 7E             02220         LD      A,(HL)          ;Is 1st char a label
2568 FE40           02230         CP      '@'             ;  indicator?
256A 20E2           02240         JR      NZ,FINDLBL      ;Back for more if not
```

```
                02250 ;
                02260 ;       Found a label - is it the one needed?
                02270 ;
256C 23         02280         INC     HL              ;Pt to 1st char
256D EB         02290         EX      DE,HL           ;Ptr to DE
256E 21C529     02300         LD      HL,LBLSAV
2571 010808     02310         LD      BC,808H         ;Symbol & field len =8
2574 CDDD28     02320         CALL    FNDPRM          ;A match?
2577 20D5       02330         JR      NZ,FINDLBL      ;No match? look for next
2579 1803       02340         JR      CPLJCL          ;  else you're the one
                02350 ;
257B CD9226     02360 CONDCPL CALL    TSTCOND
257E CD6126     02370 CPLJCL  CALL    RDJCL           ;Read line from JCL file
2581 C29924     02380         JP      NZ,CKNEST       ;Exit on end of file
2584 218C2B     02390         LD      HL,JCLBUF1      ;Parse the line just read
2587 110024     02400         LD      DE,JCLBUF2
258A 7E         02410         LD      A,(HL)
258B 23         02420         INC     HL
258C FE40       02430         CP      '@'             ;End procedure if found
258E CA9924     02440         JP      Z,CKNEST        ;  another label
2591 FE2F       02450         CP      '/'             ;Slash?
2593 2004       02460         JR      NZ,CPLJCL1
2595 BE         02470         CP      (HL)            ;Double slash?
2596 CA3326     02480         JP      Z,MACRO         ;Jump on double slash
                02490 CPLJCL1
                02500 ;
                02510 ;       Modification for HEX parsing
                02520 ;
2599 FE23       02530         CP      '#'             ;Substitution?
259B 2825       02540         JR      Z,CPLJCL4
259D FE25       02550         CP      '%'             ;Hex value?
259F 2017       02560         JR      NZ,CPLJCL2      ;Back to take char if not
25A1 CDA625     02570         CALL    CPLJCL7         ;Go test double %
25A4 1818       02580         JR      CPLJCL3
25A6 BE         02590 CPLJCL7 CP      (HL)            ;Double %?
25A7 2821       02600         JR      Z,CPLJCL6
25A9 CDCE25     02610         CALL    CVRTHEX         ;Convert digit
25AC 23         02620         INC     HL              ;Bump to next char
25AD 07         02630         RLCA
25AE 07         02640         RLCA
25AF 07         02650         RLCA
25B0 07         02660         RLCA                    ;Rotate into left nybble
25B1 4F         02670         LD      C,A             ;Save for now
25B2 CDCE25     02680         CALL    CVRTHEX         ;Convert 2nd digit
25B5 B1         02690         OR      C               ;Merge left nybble
25B6 1812       02700         JR      CPLJCL6
25B8 12         02710 CPLJCL2 LD      (DE),A          ;Nothing special, xfer
25B9 13         02720         INC     DE
25BA FE0D       02730         CP      CR
25BC 28BD       02740         JR      Z,CONDCPL       ;Exit on end of line
25BE 7E         02750 CPLJCL3 LD      A,(HL)          ;Grab next input char
25BF 23         02760         INC     HL
25C0 18D7       02770         JR      CPLJCL1         ;  & loop
25C2 CDC725     02780 CPLJCL4 CALL    CPLJCL5         ;Ck on double '#'
25C5 18F7       02790         JR      CPLJCL3         ;Substitute if not ##
25C7 BE         02800 CPLJCL5 CP      (HL)            ;Double #?
25C8 2015       02810         JR      NZ,SUBSYM       ;Jump to substitute if
25CA 23         02820 CPLJCL6 INC     HL              ;  only single #
25CB 12         02830         LD      (DE),A          ;  else xfer the char
25CC 13         02840         INC     DE
25CD C9         02850         RET
```

```
                    02860 ;
25CE 7E             02870 CVRTHEX LD      A,(HL)          ;P/u the digit
25CF D630           02880         SUB     30H             ;Start conversion
25D1 380A           02890         JR      C,CVRTHE1       ;Error if < 0
25D3 FE0A           02900         CP      10
25D5 D8             02910         RET     C               ;Go if 0-9
25D6 CBAF           02920         RES     5,A             ;In case l/c
25D8 D607           02930         SUB     7               ;Adjust A-F -> 10-15
25DA FE10           02940         CP      16
25DC D8             02950         RET     C               ;Go if 10-15
25DD 183F           02960 CVRTHE1 JR      BADHDR
                    02970 ;
                    02980 ;       Symbol substitution routine
                    02990 ;
25DF E5             03000 SUBSYM  PUSH    HL
25E0 D5             03010         PUSH    DE
25E1 CDE727         03020         CALL    PARSNAM         ;Parse symbol
25E4 FE23           03030         CP      '#'             ;Must have closing #
25E6 2036           03040         JR      NZ,BADHDR       ;Bad JCL format if not
25E8 E3             03050         EX      (SP),HL         ;Put new posn on stack
25E9 E5             03060         PUSH    HL              ;  and get HL=start posn
25EA CDB128         03070         CALL    FINDSYM         ;Get symbol value
25ED 200F           03080         JR      NZ,SUBSYM1      ;Bypass if not in table
25EF 1A             03090         LD      A,(DE)          ;Get symbol length
25F0 B7             03100         OR      A
25F1 280B           03110         JR      Z,SUBSYM1       ;Bypass if zero length
25F3 0600           03120         LD      B,0
25F5 4F             03130         LD      C,A
25F6 13             03140         INC     DE              ;Point to 1st symbol char
25F7 E1             03150         POP     HL              ;Rcvr where we need to
25F8 EB             03160         EX      DE,HL           ;  substitute then move
25F9 EDB0           03170         LDIR                    ;  symbol value into pos
25FB E1             03180         POP     HL
25FC F1             03190         POP     AF
25FD C9             03200         RET
                    03210 ;
25FE D1             03220 SUBSYM1 POP     DE              ;Symbol not in table, so
25FF F1             03230         POP     AF              ;  leave as is in the DO
2600 E1             03240         POP     HL              ;  file.
2601 3E23           03250         LD      A,'#'           ;Starting #
2603 12             03260 SUBSYM2 LD      (DE),A
2604 13             03270         INC     DE              ;Inc buffer
2605 7E             03280         LD      A,(HL)          ;Get a char from line
2606 23             03290         INC     HL
2607 FE0D           03300         CP      CR              ;If a CR before closing #
2609 2813           03310         JR      Z,BADHDR        ;  abort
260B FE23           03320         CP      '#'             ;End of substitution?
260D 20F4           03330         JR      NZ,SUBSYM2      ;Get more if not
260F 12             03340         LD      (DE),A
2610 13             03350         INC     DE
2611 C9             03360         RET
                    03370 ;
                    03380 ;       Check if conditional is at top level
                    03390 ;
2612 ED5B5C2B       03400 CKCOND  LD      DE,(CONDPTR)    ;P/u conditional pointer
2616 215E2B         03410         LD      HL,CONDFLG      ;Test if still on 1st one
2619 AF             03420         XOR     A
261A ED52           03430         SBC     HL,DE
261C EB             03440         EX      DE,HL           ;Pointer back to HL
261D C0             03450         RET     NZ              ;Ok if nested else error
                    03460 ;
```

```
                03470 ;       Output invalid JCL format message
                03480 ;
261E 11832B     03490 BADHDR  LD      DE,BADHDR$+5    ;Show bad JCL line found
2621 2ACE29     03500         LD      HL,(LINENO)     ;Put decimal line #
2624            03510         @@HEXDEC                ;  into message
2624 3E61       00029         LD      A,97
2626 EF         00030         RST     40
2627 217E2B     03520         LD      HL,BADHDR$      ;Display bad line #
262A            03530         @@LOGOT
                00031         IFEQ    00H,1
                00032         LD      HL,
                00033         ENDIF
262A 3E0C       00034         LD      A,12
262C EF         00035         RST     40
262D 217A2A     03540 BADH1   LD      HL,BADJCL$      ;  and abort message
2630 C36729     03550         JP      EXTERR
                03560 ;
                03570 ;       Compile "//" line
                03580 ;
2633 23         03590 MACRO   INC     HL
2634 CDE727     03600         CALL    PARSNAM         ;Get symbol name
2637 2015       03610         JR      NZ,MACRO2       ;Go if not JCL macro
2639 CDC228     03620         CALL    CK4COND         ;Ck for IF, ELSE, END
263C D5         03630         PUSH    DE              ;Stack the routine entry
263D C8         03640         RET     Z               ;  & branch if found
263E D1         03650         POP     DE              ;  else remove RET &...
                03660 ;
                03670 ;       Test the conditional logic state
                03680 ;
263F ED5B5C2B   03690         LD      DE,(CONDPTR)    ;P/u conditional pointer
2643 1A         03700         LD      A,(DE)          ;  & conditional state
2644 B7         03710         OR      A
2645 C27E25     03720         JP      NZ,CPLJCL       ;Jump if logic FALSE
2648 CDCB28     03730         CALL    CK4ASSN         ;Test for SET, RESET,
                03740                                 ;  ASSIGN, INCLUDE, QUIT
264B D5         03750         PUSH    DE              ;Stack the routine entry
264C C8         03760         RET     Z               ;  & branch if found
264D D1         03770         POP     DE
264E 118C2B     03780 MACRO2  LD      DE,JCLBUF1      ;Point to where we left
2651 AF         03790         XOR     A               ;  off and continue to
2652 ED52       03800         SBC     HL,DE           ;  parse the input line
2654 44         03810         LD      B,H             ;  from the JCL file
2655 4D         03820         LD      C,L
2656 218C2B     03830         LD      HL,JCLBUF1
2659 110024     03840         LD      DE,JCLBUF2
265C EDB0       03850         LDIR
265E C3BE25     03860         JP      CPLJCL3
                03870 ;
                03880 ;       Read a line from the JCL file
                03890 ;
2661 2ACE29     03900 RDJCL   LD      HL,(LINENO)     ;Bump line counter
2664 23         03910         INC     HL
2665 22CE29     03920         LD      (LINENO),HL
2668 218C2B     03930         LD      HL,JCLBUF1      ;Point to line buffer
266B 117C29     03940         LD      DE,DOFCB        ;Point to FCB
266E 0650       03950         LD      B,80            ;Permit only 80 chars
2670            03960 RDJCL1  @@GET                   ;Get a char
2670 3E03       00036         LD      A,3
2672 EF         00037         RST     40
2673 2014       03970         JR      NZ,RDJCL2       ;Jump on error
2675 B7         03980         OR      A
```

```
2676 2816     03990          JR      Z,RDJCL3        ;Bypass on null byte
2678 77       04000          LD      (HL),A          ;Xfer byte to line buf
2679 23       04010          INC     HL
267A FE0D     04020          CP      CR              ;End of line?
267C C8       04030          RET     Z
267D 10F1     04040          DJNZ    RDJCL1          ;Loop if not
              04050 ;
              04060 ;        If falls through, line too long
              04070 ;
267F 360D     04080          LD      (HL),CR         ;Stuff CR & provide
2681 21EE29   04090          LD      HL,LINLNG$      ;  error log message
2684 222E26   04100          LD      (BADH1+1),HL
2687 1895     04110          JR      BADHDR
              04120 ;
2689 FE1C     04130 RDJCL2   CP      1CH             ;EOF?
268B C23D29   04140          JP      NZ,IOERR        ;Jump on any other error
268E 3E1C     04150 RDJCL3   LD      A,1CH
2690 B7       04160          OR      A
2691 C9       04170          RET
              04180 ;
              04190 ;        Act on JCL line if conditional state = TRUE
              04200 ;
2692 2A5C2B   04210 TSTCOND  LD      HL,(CONDPTR)    ;Grab conditional pointer
2695 7E       04220          LD      A,(HL)          ;Grab conditional state
2696 B7       04230          OR      A
2697 C0       04240          RET     NZ              ;Return if logic FALSE
2698 210024   04250          LD      HL,JCLBUF2      ;Point to processed line
269B 11C000   04260          LD      DE,JFCB$        ;SYSTEM/JCL FCB
269E 7E       04270          LD      A,(HL)          ;Ck on double /
269F FE2F     04280          CP      '/'
26A1 2010     04290          JR      NZ,WRCPLD       ;Done if not /
26A3 23       04300          INC     HL
26A4 BE       04310          CP      (HL)            ;Check for double /
26A5 2B       04320          DEC     HL
26A6 200B     04330          JR      NZ,WRCPLD       ;Jump if not //
26A8 3A0224   04340          LD      A,(JCLBUF2+2)   ;Ck on comment
26AB FE2E     04350          CP      '.'             ;//. ?
26AD 2004     04360          JR      NZ,WRCPLD       ;Bypass if not comment
26AF          04370          @@DSPLY                 ;Else display the comment
              00038          IFEQ    00H,1
              00039          LD      HL,
              00040          ENDIF
26AF 3E0A     00041          LD      A,10
26B1 EF       00042          RST     40
26B2 C9       04380          RET
              04390 ;
              04400 ;        Write compiled line to SYSTEM/JCL
              04410 ;
26B3 4E       04420 WRCPLD   LD      C,(HL)          ;P/u a char
26B4          04430          @@PUT                   ;Put it out
26B4 3E04     00043          LD      A,4
26B6 EF       00044          RST     40
26B7 C23D29   04440          JP      NZ,IOERR        ;Jump on error
26BA 7E       04450          LD      A,(HL)          ;Grab again to test
26BB 23       04460          INC     HL              ;Bump pointer
26BC FE0D     04470          CP      CR              ;End of line?
26BE 20F3     04480          JR      NZ,WRCPLD       ;Loop if not
26C0 C9       04490          RET
              04500 ;
              04510 ;        Parameter tables
              04520 ;
```

```
26C1 49 46 2Ø 2Ø 2Ø   Ø453Ø CONDTBL DB   'IF   '
26C6 ØA27             Ø454Ø          DW   IFØ1
26C8 45 4C 53 45 2Ø   Ø455Ø          DB   'ELSE '
26CD 3327             Ø456Ø          DW   ELSE1
26CF 45 4E 44 2Ø 2Ø   Ø457Ø          DB   'END  '
26D4 3E27             Ø458Ø          DW   END1
26D6 ØØ               Ø459Ø          NOP
26D7 53 45 54 2Ø 2Ø 2Ø 2Ø Ø46ØØ ASSNTBL DB   'SET     '
26DF 6D27             Ø461Ø          DW   SET1
26E1 52 45 53 45 54 2Ø 2Ø 2Ø Ø462Ø  DB   'RESET   '
26E9 7C27             Ø463Ø          DW   RESET1
26EB 41 53 53 49 47 4E 2Ø 2Ø Ø464Ø  DB   'ASSIGN  '
26F3 8E27             Ø465Ø          DW   ASSIGN
26F5 49 4E 43 4C 55 44 45 2Ø Ø466Ø  DB   'INCLUDE '
26FD A927             Ø467Ø          DW   INCLUD
26FF 51 55 49 54 2Ø 2Ø 2Ø 2Ø Ø468Ø  DB   'QUIT    '
2707 E127             Ø469Ø          DW   QUIT
2709 ØØ               Ø47ØØ          NOP
                      Ø471Ø ;
                      Ø472Ø ;        Process IF command
                      Ø473Ø ;
270A CD4727           Ø474Ø IFØ1     CALL IFØ5          ;Parse expression
270D 2814             Ø475Ø          JR   Z,IFØ2         ;Z=true, NZ=false
270F FEØD             Ø476Ø          CP   CR             ;False & end of line?
2711 2813             Ø477Ø          JR   Z,IFØ3
2713 FE2B             Ø478Ø          CP   '+'            ;Logical OR?
2715 28F3             Ø479Ø          JR   Z,IFØ1
                      Ø48ØØ ;
                      Ø481Ø ;        Test for FALSE and logical AND (&)
                      Ø482Ø ;
2717 FE26             Ø483Ø          CP   '&'            ;Separator AND?
2719 2Ø55             Ø484Ø          JR   NZ,BADHDRØ     ;Invalid format if not
271B 23               Ø485Ø IFØ1A    INC  HL             ;Ignore rest of line
271C 7E               Ø486Ø          LD   A,(HL)
271D FEØD             Ø487Ø          CP   CR
271F 2ØFA             Ø488Ø          JR   NZ,IFØ1A
2721 18Ø3             Ø489Ø          JR   IFØ3
2723 AF               Ø49ØØ IFØ2     XOR  A              ;Logic = true
2724 18Ø2             Ø491Ø          JR   IFØ4
2726 3EFF             Ø492Ø IFØ3     LD   A,ØFFH         ;Logic = false
2728 2A5C2B           Ø493Ø IFØ4     LD   HL,(CONDPTR)   ;Get conditional pointer
272B B6               Ø494Ø          OR   (HL)           ;Set logic state
272C 23               Ø495Ø          INC  HL             ;Bump pointer
272D 77               Ø496Ø          LD   (HL),A         ;Stuff state result
272E 225C2B           Ø497Ø          LD   (CONDPTR),HL   ;Save pointer
2731 1846             Ø498Ø          JR   GOJCL
                      Ø499Ø ;
                      Ø5ØØØ ;        Process ELSE command
                      Ø5Ø1Ø ;
2733 CD1226           Ø5Ø2Ø ELSE1    CALL CKCOND         ;Ck nest of conditional
2736 7E               Ø5Ø3Ø          LD   A,(HL)         ;Flip state of flag based
2737 2F               Ø5Ø4Ø          CPL                 ;  on previous test
2738 2B               Ø5Ø5Ø          DEC  HL
```

```
2739 B6       Ø5Ø6Ø          OR      (HL)            ;OR in previous state
273A 23       Ø5Ø7Ø          INC     HL
273B 77       Ø5Ø8Ø          LD      (HL),A          ;Store new value
273C 183B     Ø5Ø9Ø          JR      GOJCL
              Ø51ØØ ;
              Ø511Ø ;        Process END command
              Ø512Ø ;
273E CD1226   Ø513Ø END1     CALL    CKCOND          ;Ck nest level
2741 2B       Ø514Ø          DEC     HL              ;Backup conditional one
2742 225C2B   Ø515Ø          LD      (CONDPTR),HL    ;  level & reset pointer
2745 1832     Ø516Ø          JR      GOJCL
              Ø517Ø ;
              Ø518Ø ;        Parse conditional expression logic
              Ø519Ø ;
2747 CD5127   Ø52ØØ IFØ5     CALL    IFØ6            ;Get if symbol is true
274A CØ       Ø521Ø          RET     NZ              ;  or false & ret if false
274B FE26     Ø522Ø          CP      '&'             ;Logical AND separator?
274D 28F8     Ø523Ø          JR      Z,IFØ5          ;If TRUE AND -> ck next
274F AF       Ø524Ø          XOR     A               ;True and not AND,
2750 C9       Ø525Ø          RET                     ;  ret true
2751 7E       Ø526Ø IFØ6     LD      A,(HL)
2752 FE2D     Ø527Ø          CP      '-'             ;Logical NOT?
2754 2ØØA     Ø528Ø          JR      NZ,IFØ8
2756 23       Ø529Ø          INC     HL              ;Bypass '-'
2757 CD6Ø27   Ø53ØØ          CALL    IFØ8            ;Grab symbol logic state
275A 2ØØ1     Ø531Ø          JR      NZ,IFØ7         ;Z=true, NZ=false
275C F6       Ø532Ø          DB      ØF6H            ;Was true, not => false
275D AF       Ø533Ø IFØ7     XOR     A               ;Was false, not => true
275E 78       Ø534Ø          LD      A,B             ;Rcvr separator
275F C9       Ø535Ø          RET
2760 CDE727   Ø536Ø IFØ8     CALL    PARSNAM         ;Get symbol name into buf
2763 CØ       Ø537Ø          RET     NZ              ;Ret if bad symbol
2764 F5       Ø538Ø          PUSH    AF
2765 E5       Ø539Ø          PUSH    HL
2766 CDB128   Ø54ØØ          CALL    FINDSYM         ;Find symbol in table
2769 E1       Ø541Ø          POP     HL
276A C1       Ø542Ø          POP     BC
276B 78       Ø543Ø          LD      A,B             ;Put zero in A & use flag
276C C9       Ø544Ø          RET                     ;From search
              Ø545Ø ;
              Ø546Ø ;        Process SET command
              Ø547Ø ;
276D CDE727   Ø548Ø SET1     CALL    PARSNAM         ;Parse symbol name
2770 C21E26   Ø549Ø BADHDRØ  JP      NZ,BADHDR       ;Jump if bad symbol
2773 CDB128   Ø55ØØ          CALL    FINDSYM         ;Find in table
2776 C49428   Ø551Ø          CALL    NZ,MOVNAME      ;Move name into table
2779 C37E25   Ø552Ø GOJCL    JP      CPLJCL
              Ø553Ø ;
              Ø554Ø ;        Process RESET command
              Ø555Ø ;
277C CDE727   Ø556Ø RESET1   CALL    PARSNAM         ;Parse symbol name
277F 2ØEF     Ø557Ø          JR      NZ,BADHDRØ
2781 CDB128   Ø558Ø          CALL    FINDSYM         ;Find symbol in table
2784 2ØF3     Ø559Ø          JR      NZ,GOJCL        ;No problem if not there
2786 21F8FF   Ø56ØØ          LD      HL,-8           ;Point to start of name
2789 19       Ø561Ø          ADD     HL,DE           ;  & put in a blank
278A 362Ø     Ø562Ø          LD      (HL),' '        ;  to remove symbol
278C 18EB     Ø563Ø          JR      GOJCL
              Ø564Ø ;
              Ø565Ø ;        Process ASSIGN command
              Ø566Ø ;
```

```
278E CDE727   Ø567Ø ASSIGN  CALL    PARSNAM          ;Parse symbol name
2791 2ØDD     Ø568Ø         JR      NZ,BADHDRØ       ;Jump on bad name
2793 F5       Ø569Ø         PUSH    AF               ;Save separator char
2794 CDB128   Ø57ØØ         CALL    FINDSYM          ;Find in table
2797 C49428   Ø571Ø         CALL    NZ,MOVNAME       ;Add to table if not in
279A F1       Ø572Ø         POP     AF               ;Recover separator
279B FE3D     Ø573Ø         CP      '='              ;Error if not =
279D 2ØD1     Ø574Ø         JR      NZ,BADHDRØ
279F CDF227   Ø575Ø         CALL    PARSVAL          ;Parse value of symbol
27A2 2ØCC     Ø576Ø         JR      NZ,BADHDRØ
27A4 CDA628   Ø577Ø         CALL    MOVALUE          ;Place value into table
27A7 18DØ     Ø578Ø         JR      GOJCL
              Ø579Ø ;
              Ø58ØØ ;       Process INCLUDE command
              Ø581Ø ;
27A9 E5       Ø582Ø INCLUD  PUSH    HL
27AA ED5BBA2A Ø583Ø         LD      DE,(NESTPTR)     ;Point to next FCB save
27AE 215C2B   Ø584Ø         LD      HL,NESTEND       ;  area & check if room
27B1 AF       Ø585Ø         XOR     A                ;  to store another FCB
27B2 ED52     Ø586Ø         SBC     HL,DE
27B4 CA4C29   Ø587Ø         JP      Z,NESTS          ;Error if 5 nests already
27B7 217C29   Ø588Ø         LD      HL,DOFCB         ;Shift current FCB into
27BA Ø12ØØØ   Ø589Ø         LD      BC,32            ;  INCLUDE FCB save area
27BD EDBØ     Ø59ØØ         LDIR
27BF ED53BA2A Ø591Ø         LD      (NESTPTR),DE     ;Update new nest pointer
27C3 E1       Ø592Ø         POP     HL
27C4 117C29   Ø593Ø         LD      DE,DOFCB         ;Point to FCB
27C7          Ø594Ø         @@FSPEC                  ;Fetch included file
27C7 3E4E     ØØØ45         LD      A,78
27C9 EF       ØØØ46         RST     4Ø
27CA 2ØA4     Ø595Ø         JR      NZ,BADHDRØ       ;Jump on error
27CC 21D729   Ø596Ø         LD      HL,SYSJCL+7      ;Default to /JCL
27CF          Ø597Ø         @@FEXT
27CF 3E4F     ØØØ47         LD      A,79
27D1 EF       ØØØ48         RST     4Ø
27D2 21ØØ2C   Ø598Ø         LD      HL,INPBUF        ;Open the included file
27D5 45       Ø599Ø         LD      B,L
27D6 FDCB12C6 Ø6ØØØ         SET     Ø,(IY+'S'-'A')   ;Inhibit file open bit
27DA          Ø6Ø1Ø         @@OPEN
27DA 3E3B     ØØØ49         LD      A,59
27DC EF       ØØØ5Ø         RST     4Ø
27DD 2Ø91     Ø6Ø2Ø         JR      NZ,BADHDRØ
27DF 1898     Ø6Ø3Ø         JR      GOJCL
              Ø6Ø4Ø ;
              Ø6Ø5Ø ;       Process QUIT command
              Ø6Ø6Ø ;
27E1 218C2B   Ø6Ø7Ø QUIT    LD      HL,JCLBUF1       ;Log the //QUIT command
27E4 C36729   Ø6Ø8Ø         JP      EXTERR
              Ø6Ø9Ø ;
              Ø61ØØ ;       Parse symbol name
              Ø611Ø ;       A <= separator char
              Ø612Ø ;       Z  = ok, NZ = bad symbol char
              Ø613Ø ;
27E7 D5       Ø614Ø PARSNAM PUSH    DE
27E8 Ø6Ø8     Ø615Ø         LD      B,8              ;8 chars max
27EA 119C29   Ø616Ø         LD      DE,CURSYM        ;Symbol buffer area
27ED CD2E28   Ø617Ø         CALL    PARSER           ;Parse it
27FØ D1       Ø618Ø         POP     DE
27F1 C9       Ø619Ø         RET
              Ø62ØØ ;
              Ø621Ø ;       Parse a symbol value
```

```
                    Ø622Ø ;
27F2 D5             Ø623Ø PARSVAL PUSH    DE
27F3 Ø62Ø           Ø624Ø         LD      B,32            ;32 chars max
27F5 11A529         Ø625Ø         LD      DE,VALBUF       ;Value buffer
27F8 CD1128         Ø626Ø         CALL    XFRSTR          ;Transfer from input
27FB F5             Ø627Ø         PUSH    AF
27FC E5             Ø628Ø         PUSH    HL
27FD EB             Ø629Ø         EX      DE,HL           ;Calculate length of
27FE 11A529         Ø63ØØ         LD      DE,VALBUF       ;  the string
28Ø1 AF             Ø631Ø         XOR     A
28Ø2 ED52           Ø632Ø         SBC     HL,DE
28Ø4 7D             Ø633Ø         LD      A,L
28Ø5 FE21           Ø634Ø         CP      33
28Ø7 D25Ø29         Ø635Ø         JP      NC,TOOLNG       ;Jump if > 32 chars
28ØA 32A429         Ø636Ø         LD      (STRLEN),A      ;Stuff string length
28ØD E1             Ø637Ø         POP     HL
28ØE F1             Ø638Ø         POP     AF
28ØF D1             Ø639Ø         POP     DE
281Ø C9             Ø64ØØ         RET
                    Ø641Ø ;
                    Ø642Ø ;       Transfer a string field
                    Ø643Ø ;
2811 CD2E28         Ø644Ø XFRSTR  CALL    PARSER          ;Xfer max of 32 chars
2814 CD2D29         Ø645Ø XFRSTR1 CALL    CKSPCOM         ;Return on space
2817 C8             Ø646Ø         RET     Z               ;  or comma
2818 FEØD           Ø647Ø         CP      CR
281A C8             Ø648Ø         RET     Z               ;Ret on end of line
281B FE3D           Ø649Ø         CP      '='
281D C8             Ø65ØØ         RET     Z               ;Ret on =
281E FE28           Ø651Ø         CP      '('
282Ø C8             Ø652Ø         RET     Z               ;Ret on left paren
2821 FE29           Ø653Ø         CP      ')'
2823 C8             Ø654Ø         RET     Z               ;Ret on right paren
2824 FE23           Ø655Ø         CP      '#'
2826 2ØE9           Ø656Ø         JR      NZ,XFRSTR       ;Loop if not #
2828 CDC725         Ø657Ø         CALL    CPLJCL5         ;Ck on substitution
282B 7E             Ø658Ø         LD      A,(HL)
282C 18E6           Ø659Ø         JR      XFRSTR1         ;Then loop
                    Ø66ØØ ;
                    Ø661Ø ;       Parse a field
                    Ø662Ø ;
282E 78             Ø663Ø PARSER  LD      A,B             ;Set max length of field
282F 329Ø28         Ø664Ø         LD      (PAR6+1),A
2832 Ø4             Ø665Ø         INC     B
2833 7E             Ø666Ø PAR2    LD      A,(HL)          ;P/u entry char
2834 FEØ3           Ø667Ø         CP      3               ;ETX?
2836 284C           Ø668Ø         JR      Z,PAR5
2838 FEØD           Ø669Ø         CP      CR
283A 2848           Ø67ØØ         JR      Z,PAR5
283C 23             Ø671Ø         INC     HL              ;Not ending char, bump
283D FE22           Ø672Ø         CP      '"'             ;Ck on string quote
283F 2ØØ7           Ø673Ø         JR      NZ,NOTQT
2841 EE22           Ø674Ø         XOR     '"'             ;Ck if opening or closing
2842                Ø675Ø STUFQT  EQU     $-1
2843 324228         Ø676Ø         LD      (STUFQT),A
2846 18EB           Ø677Ø         JR      PAR2            ;Loop until terminator
2848 4F             Ø678Ø NOTQT   LD      C,A             ;Save char & test if
2849 3A4228         Ø679Ø         LD      A,(STUFQT)      ;  within quoted string
284C B7             Ø68ØØ         OR      A
284D 79             Ø681Ø         LD      A,C             ;Get back the char
284E 2826           Ø682Ø         JR      Z,PAR3          ;Allow all within "..."
```

```
285Ø FE4Ø     Ø683Ø          CP      '@'             ;Start of label?
2852 2ØØD     Ø684Ø          JR      NZ,NOLBL
2854 D6ØØ     Ø685Ø GOTLBL   SUB     Ø               ;Make sure only one
2856 CA5829   Ø686Ø          JP      Z,LBLERR
2859 325528   Ø687Ø          LD      (GOTLBL+1),A    ;Stuff '&' into test
285C 32Ø825   Ø688Ø          LD      (FNDLBL+1),A    ;  & also for check
285F 18D2     Ø689Ø          JR      PAR2            ;Loop through start
2861 FE2E     Ø69ØØ NOLBL    CP      '.'             ;Accept (., /, Ø-9, :)
2863 381F     Ø691Ø          JR      C,PAR5
2865 FE3B     Ø692Ø          CP      ':'+1
2867 38ØD     Ø693Ø          JR      C,PAR3
2869 FE41     Ø694Ø          CP      'A'             ;Test for A-Z
286B 3817     Ø695Ø          JR      C,PAR5
286D FE5B     Ø696Ø          CP      'Z'+1
286F 38Ø5     Ø697Ø          JR      C,PAR3
2871 CD3329   Ø698Ø          CALL    CKLCA2Z         ;Test for a-z
2874 38ØE     Ø699Ø          JR      C,PAR5
2876 Ø5       Ø7ØØØ PAR3     DEC     B               ;Char count down
2877 28Ø8     Ø7Ø1Ø          JR      Z,PAR4
2879 12       Ø7Ø2Ø          LD      (DE),A          ;Save the char
287A AF       Ø7Ø3Ø          XOR     A               ;Show we found at
287B 329Ø28   Ø7Ø4Ø          LD      (PAR6+1),A      ;  least one valid char
287E 13       Ø7Ø5Ø          INC     DE              ;Bump receiving buffer
287F 18B2     Ø7Ø6Ø          JR      PAR2            ;Loop
2881 Ø4       Ø7Ø7Ø PAR4     INC     B               ;Ignore trailing chars
2882 18AF     Ø7Ø8Ø          JR      PAR2            ;  past max length
2884 4F       Ø7Ø9Ø PAR5     LD      C,A             ;Found char out of range
2885 D5       Ø71ØØ          PUSH    DE              ;Save current end of buf
2886 18Ø4     Ø711Ø          JR      PAR5B
2888 3E2Ø     Ø712Ø PAR5A    LD      A,' '           ;Fill out remaining field
288A 12       Ø713Ø          LD      (DE),A          ;  with blanks
288B 13       Ø714Ø          INC     DE
288C 1ØFA     Ø715Ø PAR5B    DJNZ    PAR5A
288E D1       Ø716Ø          POP     DE              ;Recover pointer to last
288F 3EØØ     Ø717Ø PAR6     LD      A,Ø             ;Char xfered, get max len
2891 B7       Ø718Ø          OR      A               ;Note if we found a char
2892 79       Ø719Ø          LD      A,C             ;Xfer separator char
2893 C9       Ø72ØØ          RET
              Ø721Ø ;
              Ø722Ø ;        Xfer symbol name to table & init value
              Ø723Ø ;
2894 E5       Ø724Ø MOVNAME  PUSH    HL
2895 219C29   Ø725Ø          LD      HL,CURSYM       ;Current symbol buffer
2898 Ø1Ø8ØØ   Ø726Ø          LD      BC,8            ;8 chars to move
289B EDBØ     Ø727Ø          LDIR
289D AF       Ø728Ø          XOR     A               ;Zero accumulator
289E 12       Ø729Ø          LD      (DE),A          ;Show symbol length=Ø
289F 2121ØØ   Ø73ØØ          LD      HL,33           ;Point to 1st byte
28A2 19       Ø731Ø          ADD     HL,DE           ;  of next symbol pos and
28A3 77       Ø732Ø          LD      (HL),A          ;  show it spare
28A4 E1       Ø733Ø          POP     HL
28A5 C9       Ø734Ø          RET
              Ø735Ø ;
              Ø736Ø ;        Place symbol value into table
              Ø737Ø ;
28A6 E5       Ø738Ø MOVALUE  PUSH    HL
28A7 21A429   Ø739Ø          LD      HL,STRLEN       ;Current value buffer
28AA Ø121ØØ   Ø74ØØ          LD      BC,33           ;Length & value
28AD EDBØ     Ø741Ø          LDIR
28AF E1       Ø742Ø          POP     HL
28BØ C9       Ø743Ø          RET
```

```
                    07440 ;
                    07450 ;       Find symbol in table
                    07460 ;
28B1 E5             07470 FINDSYM PUSH    HL
28B2 119C29         07480         LD      DE,CURSYM       ;Symbol buffer
28B5 2100 2E        07490         LD      HL,SYMTAB       ;Start of table
28B8 012908         07500         LD      BC,8<8!41       ;CP8, field (8,1,32)
28BB CDDD28         07510         CALL    FNDPRM          ;Search in progress
28BE 54             07520         LD      D,H             ;Xfer pointer of symbol
28BF 5D             07530         LD      E,L             ;  or to spare slot
28C0 E1             07540         POP     HL
28C1 C9             07550         RET
                    07560 ;
                    07570 ;       Routine to check for IF, ELSE, END
                    07580 ;
28C2 E5             07590 CK4COND PUSH    HL
28C3 21C126         07600         LD      HL,CONDTBL      ;Parm table
28C6 010705         07610         LD      BC,5<8!7        ;5 chars, 7-char field
28C9 1807           07620         JR      CK4AS1
                    07630 ;
                    07640 ;       Check on SET, RESET, ASSIGN, INCLUDE, QUIT
                    07650 ;
28CB E5             07660 CK4ASSN PUSH    HL
28CC 21D726         07670         LD      HL,ASSNTBL      ;Parm table
28CF 010A08         07680         LD      BC,8<8!10       ;Parm length, field len
28D2 119C29         07690 CK4AS1  LD      DE,CURSYM       ;Buffer area
28D5 CDDD28         07700         CALL    FNDPRM          ;Ck for match
28D8 5E             07710         LD      E,(HL)          ;Xfer vector address
28D9 23             07720         INC     HL
28DA 56             07730         LD      D,(HL)
28DB E1             07740         POP     HL
28DC C9             07750         RET
                    07760 ;
                    07770 ;       Scan parm table for match
                    07780 ;
28DD 7E             07790 FNDPRM  LD      A,(HL)          ;End of parm table?
28DE B7             07800         OR      A
28DF 2002           07810         JR      NZ,FND1         ;Jump if not
28E1 3C             07820         INC     A               ;  else show not found
28E2 C9             07830         RET
28E3 1A             07840 FND1    LD      A,(DE)          ;Char match?
28E4 CD3329         07850         CALL    CKLCA2Z         ;Convert a-z to A-Z
28E7 BE             07860         CP      (HL)
28E8 2807           07870         JR      Z,FND3          ;Jump if 1st matches
28EA C5             07880 FND2    PUSH    BC              ;  else bypass complete
28EB 0600           07890         LD      B,0             ;  field & go to next one
28ED 09             07900         ADD     HL,BC
28EE C1             07910         POP     BC
28EF 18EC           07920         JR      FNDPRM
28F1 E5             07930 FND3    PUSH    HL              ;1st matches, ck rest
28F2 D5             07940         PUSH    DE
28F3 C5             07950         PUSH    BC
28F4 05             07960         DEC     B               ;Adj for 1st match
28F5 13             07970 FND4    INC     DE
28F6 23             07980         INC     HL
28F7 1A             07990         LD      A,(DE)
28F8 FE20           08000         CP      ' '
28FA 2827           08010         JR      Z,FND7          ;Stop checking on space
28FC FE0D           08020         CP      CR
28FE 2823           08030         JR      Z,FND7          ;Or end of line
2900 CD3329         08040         CALL    CKLCA2Z         ;Ck & convert a-z to A-Z
```

```
2903 BE       08050          CP      (HL)            ;Compare remaining chars
2904 200D     08060          JR      NZ,FND6         ;Jump on mismatch
2906 10ED     08070          DJNZ    FND4            ;Loop to count
2908 C1       08080 FND5     POP     BC              ;Must have matched
2909 D1       08090          POP     DE              ;Bypass remaining part
290A E1       08100          POP     HL              ;  of field and point to
290B C5       08110          PUSH    BC              ;  address vector of parm
290C 48       08120          LD      C,B             ;  in parm table
290D 0600     08130          LD      B,0
290F 09       08140          ADD     HL,BC
2910 C1       08150          POP     BC
2911 AF       08160          XOR     A
2912 C9       08170          RET
2913 FE30     08180 FND6     CP      '0'             ;No match, is it ASCII?
2915 380C     08190          JR      C,FND7
2917 FE3A     08200          CP      '9'+1           ;0-9?
2919 380D     08210          JR      C,FND8
291B FE41     08220          CP      'A'             ;A-Z?
291D 3804     08230          JR      C,FND7
291F FE5B     08240          CP      'Z'+1
2921 3805     08250          JR      C,FND8
2923 7E       08260 FND7     LD      A,(HL)          ;If table entry also a
2924 FE20     08270          CP      ' '             ;  space, we have a match
2926 28E0     08280          JR      Z,FND5
2928 C1       08290 FND8     POP     BC
2929 D1       08300          POP     DE
292A E1       08310          POP     HL
292B 18BD     08320          JR      FND2
              08330          ENDIF
              08340 ;
              08350 ;        Routine to ck on space or comma
              08360 ;
292D FE20     08370 CKSPCOM  CP      ' '
292F C8       08380          RET     Z
2930 FE2C     08390          CP      ','
2932 C9       08400          RET
              08410 ;
              08420 ;        Routine to convert a-z to A-Z & set C-flag
              08430 ;
2933 FE61     08440 CKLCA2Z  CP      'a'             ;Back with C-flag if
2935 D8       08450          RET     C               ;  not a-z
2936 FE7B     08460          CP      'z'+1
2938 3F       08470          CCF
2939 D8       08480          RET     C
293A EE20     08490          XOR     20H             ;Make U/C & reset CF
293C C9       08500          RET
              08510 *LIST    ON
              08520 ;
              08530 ;
              08540 ;        Error processing
              08550 ;
293D 6F       08560 IOERR    LD      L,A             ;Xfer errnum to HL
293E 2600     08570          LD      H,0
2940 F6C0     08580          OR      0C0H            ;Set brief, return
2942 4F       08590          LD      C,A
2943          08600          @@ERROR                 ;Display error
2943 3E1A     00051          LD      A,26
2945 EF       00052          RST     40
2946 1825     08610          JR      ERREXIT
              08620 ;
2948 21DB29   08630 SPCREQ   LD      HL,SPCREQ$      ;"filespec required"
```

```
                    08640 ;
                    08650 *LIST   OFF
                    08670 *LIST   ON
294B DD             08680         DB      ØDDH
294C 21A12A         08690 NESTS   LD      HL,NESTS$
294F DD             08700         DB      ØDDH
2950 21FC29         08710 TOOLNG  LD      HL,TOOLNG$      ;"symbol too long..
2953 DD             08720         DB      ØDDH
2954 21132A         08730 NOFIND  LD      HL,NOFIND$      ;"proc not found..
2957 DD             08740         DB      ØDDH
2958 21272A         08750 LBLERR  LD      HL,LBLERR$      ;"too many proc labels..
295B DD             08760         DB      ØDDH
295C 213C2A         08770 DSKFUL  LD      HL,DSKFUL$      ;"can't create SYS/JCL"
295F DD             08780         DB      ØDDH
2960 216A2A         08790 PRMERR  LD      HL,PRMERR$      ;"parameter error"
2963 DD             08800         DB      ØDDH
2964 21592A         08810 MULDEF  LD      HL,MULDEF$      ;"multiply defined
                    08820         ENDIF
                    08830 *LIST   ON
                    08840 ;
2967                08850 EXTERR  @@LOGOT
                    00053         IFEQ    ØØH,1
                    00054         LD      HL,
                    00055         ENDIF
2967 3EØC           00056         LD      A,12
2969 EF             00057         RST     4Ø
296A 21FFFF         08860         LD      HL,-1           ;Set error exit
296D                08870 ERREXIT EQU     $
296D 11CØØØ         08880         LD      DE,JFCB$        ;If the output JCL file
2970 1A             08890         LD      A,(DE)          ;  is open, then we need
2971 CB7F           08900         BIT     7,A             ;  to close it
2973 28Ø3           08910         JR      Z,SPSAV
2975                08920         @@CLOSE
2975 3E3C           00058         LD      A,6Ø
2977 EF             00059         RST     4Ø
2978 31ØØØØ         08930 SPSAV   LD      SP,$-$
297B C9             08940         RET
                    08950 ;
                    08960 *LIST   OFF
                    08980 *LIST   ON
ØØ2Ø                08990 DOFCB   DS      32
ØØØ8                09000 CURSYM  DS      8
ØØØ1                09010 STRLEN  DS      1
ØØ2Ø                09020 VALBUF  DS      32
ØØØ8                09030 LBLSAV  DS      8
29CD ØØ             09040         NOP                     ;Must be zero
                    09050         ENDIF
                    09060 ;
                    09070 *LIST   ON
29CE ØØØØ           09080 LINENO  DW      Ø               ;JCL line #
29DØ 53             09090 SYSJCL  DB      'SYSTEM/JCL',3
     59 53 54 45 4D 2F 4A 43
     4C Ø3
29DB 46             09100 SPCREQ$ DB      'File spec required',CR
     69 6C 65 2Ø 73 7Ø 65 63
     2Ø 72 65 71 75 69 72 65
     64 ØD
                    09110 *LIST   OFF
                    09130 *LIST   ON
29EE 4C             09140 LINLNG$ DB      'Line too long',CR
     69 6E 65 2Ø 74 6F 6F 2Ø
```

```
     6C 6F 6E 67 ØD
29FC 53       Ø915Ø TOOLNG$ DB      'Symbol string too long',CR
     79 6D 62 6F 6C 2Ø 73 74
     72 69 6E 67 2Ø 74 6F 6F
     2Ø 6C 6F 6E 67 ØD
2A13 5Ø       Ø916Ø NOFIND$ DB      'Procedure not found',CR
     72 6F 63 65 64 75 72 65
     2Ø 6E 6F 74 2Ø 66 6F 75
     6E 64 ØD
2A27 54       Ø917Ø LBLERR$ DB      'Too many Proc labels',CR
     6F 6F 2Ø 6D 61 6E 79 2Ø
     5Ø 72 6F 63 2Ø 6C 61 62
     65 6C 73 ØD
2A3C 43       Ø918Ø DSKFUL$ DB      'Can''t create SYSTEM/JCL file',CR
     61 6E 27 74 2Ø 63 72 65
     61 74 65 2Ø 53 59 53 54
     45 4D 2F 4A 43 4C 2Ø 66
     69 6C 65 ØD
2A59 4D       Ø919Ø MULDEF$ DB      'Multiply defined '     ;Follow with PRMERR$
     75 6C 74 69 7Ø 6C 79 2Ø
     64 65 66 69 6E 65 64 2Ø
2A6A 5Ø       Ø92ØØ PRMERR$ DB      'Parameter error',CR
     61 72 61 6D 65 74 65 72
     2Ø 65 72 72 6F 72 ØD
2A7A 49       Ø921Ø BADJCL$ DB      'Invalid JCL format, processing aborted',CR
     6E 76 61 6C 69 64 2Ø 4A
     43 4C 2Ø 66 6F 72 6D 61
     74 2C 2Ø 7Ø 72 6F 63 65
     73 73 69 6E 67 2Ø 61 62
     6F 72 74 65 64 ØD
2AA1 54       Ø922Ø NESTS$  DB      'Too many nested INCLUDEs',CR
     6F 6F 2Ø 6D 61 6E 79 2Ø
     6E 65 73 74 65 64 2Ø 49
     4E 43 4C 55 44 45 73 ØD
2ABA BC2A     Ø923Ø NESTPTR DW      NESTFCB         ;Pointer to nest FCB
ØØAØ          Ø924Ø NESTFCB DS      32*5            ;Space for 5 levels
2B5C          Ø925Ø NESTEND EQU     $               ;Ck for too many includes
2B5C 5E2B     Ø926Ø CONDPTR DW      CONDFLG         ;Conditional pointer
2B5E ØØ       Ø927Ø CONDFLG DB      Ø               ;Init 1st state to TRUE
ØØ1F          Ø928Ø         DS      31              ;32 conditional levels
2B7E 4C       Ø929Ø BADHDR$ DB      'Line xxxxx -->'
     69 6E 65 2Ø 78 78 78 78
     78 2Ø 2D 2D 3E
ØØ5Ø          Ø93ØØ JCLBUF1 DS      8Ø
2CØØ          Ø931Ø         ORG     $<-8+1<+8
Ø1ØØ          Ø932Ø INPBUF  DS      256
Ø1ØØ          Ø933Ø OUTBUF  DS      256
2EØØ ØØ       Ø934Ø SYMTAB  DB      Ø
              Ø935Ø         ENDIF
              Ø936Ø *LIST   ON
2EØ1          Ø937Ø CORE$   DEFL    $
              Ø938Ø ;
24ØØ          Ø939Ø         END     DO
```

@@1	ØØØØ	@@2	ØØØØ	@@3	ØØØØ	@@4	ØØØØ
@MOD2	ØØØØ	@MOD4	FFFF	ASSIGN	278E	ASSNTBL	26D7
BADH1	262D	BADHDR	261E	BADHDR$	2B7E	BADHDRØ	277Ø
BADJCL$	2A7A	CK4AS1	28D2	CK4ASSN	28CB	CK4COND	28C2
CKCOND	2612	CKLCA2Z	2933	CKNEST	2499	CKSPCOM	292D
CONDCPL	257B	CONDFLG	2B5E	CONDPTR	2B5C	CONDTBL	26C1
CORE$	2EØ1	CPLFIN	24AA	CPLFIN1	24BA	CPLJCL	257E
CPLJCL1	2599	CPLJCL2	25B8	CPLJCL3	25BE	CPLJCL4	25C2
CPLJCL5	25C7	CPLJCL6	25CA	CPLJCL7	25A6	CR	ØØØD
CURSYM	299C	CVRTHE1	25DD	CVRTHEX	25CE	DO	24ØØ
DOFCB	297C	DSKFUL	295C	DSKFUL$	2A3C	ELSE1	2733
END1	273E	ERREXIT	296D	EXTERR	2967	FINDLBL	254E
FINDSYM	28B1	FND1	28E3	FND2	28EA	FND3	28F1
FND4	28F5	FND5	29Ø8	FND6	2913	FND7	2923
FND8	2928	FNDLBL	25Ø7	FNDPRM	28DD	GETSEP	2532
GETSPEC	2423	GOJCL	2779	GOTLBL	2854	GOTLIN	2564
IFØ1	27ØA	IFØ1A	271B	IFØ2	2723	IFØ3	2726
IFØ4	2728	IFØ5	2747	IFØ6	2751	IFØ7	275D
IFØ8	276Ø	INBUF	246C	INCLUD	27A9	INPBUF	2CØØ
IOERR	293D	JCLBUF1	2B8C	JCLBUF2	24ØØ	JFCB$	ØØCØ
LBLERR	2958	LBLERR$	2A27	LBLSAV	29C5	LINENO	29CE
LINLNG$	29EE	MACRO	2633	MACRO2	264E	MOVALUE	28A6
MOVFCB	24F5	MOVLBL	2535	MOVNAME	2894	MULDEF	2964
MULDEF$	2A59	NESTEND	2B5C	NESTFCB	2ABC	NESTPTR	2ABA
NESTS	294C	NESTS$	2AA1	NOCPL	24D9	NOCPL1	24DF
NOCPLS	24DA	NOEXEC?	24B3	NOFIND	2954	NOFIND$	2A13
NOLBL	2861	NOTQT	2848	OUTBUF	2DØØ	PAR2	2833
PAR3	2876	PAR4	2881	PAR5	2884	PAR5A	2888
PAR5B	288C	PAR6	288F	PARAM	25Ø1	PARAM1	251A
PARAM2	252B	PARSER	282E	PARSINP	2451	PARSNAM	27E7
PARSVAL	27F2	PRMERR	296Ø	PRMERR$	2A6A	QUIT	27E1
RDJCL	2661	RDJCL1	267Ø	RDJCL2	2689	RDJCL3	268E
RESET1	277C	SET1	276D	SMALL	ØØØØ	SPCREQ	2948
SPCREQ$	29DB	SPSAV	2978	STRLEN	29A4	STUFQT	2842
SUBSYM	25DF	SUBSYM1	25FE	SUBSYM2	26Ø3	SYMTAB	2EØØ
SYSJCL	29DØ	TOOLNG	295Ø	TOOLNG$	29FC	TSTCOND	2692
TSTLBL	2548	UNNEST	247F	VALBUF	29A5	WRCPLD	26B3
XFRSTR	2811	XFRSTR1	2814	@@ABORT	BF67	@@ADTSK	BFFA
@@BANK	C512	@@BKSP	C1F2	@@BREAK	C528	@@CHNIO	BF52
@@CKBRKC	C576	@@CKDRV	CØ4E	@@CKEOF	C2Ø7	@@CKTSK	BFE5
@@CLOSE	C1DD	@@CLS	C56Ø	@@CMNDI	BF91	@@CMNDR	BFA6
@@CTL	BDB6	@@DATE	BF28	@@DCSTAT	CØ8D	@@DEBUG	BFDØ
@@DECHEX	C492	@@DIRRD	C3FF	@@DIRWR	C414	@@DIV16	C47D
@@DIV8	C468	@@DODIR	CØ63	@@DSP	BD7A	@@DSPLY	BE1A
@@ERROR	BFBB	@@EXIT	BF7C	@@FEXT	C36C		

```
@@FLAGS     C4FC @@FNAME    C381 @@FSPEC    C357
@@GATRD     C3EA @@GATWR    C429 @@GET      BD8E
@@GTDCB     C3AB @@GTDCT    C396 @@GTMOD    C3CØ
@@HDFMT     C135 @@HEX16    C4D1 @@HEX8     C4BC
@@HEXDEC    C4A7 @@HIGH$    C4E6 @@INIT     C1B3
@@KBD       BDF2 @@KEY      BD66 @@KEYIN    BEØ6
@@KLTSK     CØ39 @@LOAD     C32D @@LOC      C21C
@@LOF       C231 @@LOGER    BE51 @@LOGOT    BE66
@@MSG       BE9D @@MUL16    C453 @@MUL8     C43E
@@OPEN      C1C8 @@PARAM    BF13 @@PAUSE    BEFE
@@PEOF      C246 @@POSN     C25B @@PRINT    BEB2
@@PRT       BDCA @@PUT      BDA2 @@RAMDIR   CØ78
@@RDSEC     C1ØB @@RDSSC    C3D5 @@READ     C27Ø
@@REMOV     C19E @@RENAM    C189 @@REW      C285
@@RMTSK     CØØF @@RPTSK    CØ24 @@RREAD    C29A
@@RSLCT     CØF6 @@RSTOR    CØB7 @@RUN      C342
@@RWRIT     C2AF @@SEEK     CØE1 @@SEEKSC   C2C4
@@SKIP      C2D9 @@SLCT     CØA2 @@STEPI    CØCC
@@TIME      BF3D @@VDCTL    BEE9 @@VER      C2EE
@@VRSEC     C12Ø @@WEOF     C3Ø3 @@WHERE    BDDE
@@WRITE     C318 @@WRSEC    C14A @@WRSSC    C15F
@@WRTRK     C174
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

Command: DUMP

Library: SYS7/SYS

ISAM # : 71H

```
                00100 ;LBDUMP/ASM - DUMP Command
0000            00110         TITLE   <DUMP - LS-DOS 6.2>
                00120 ;
0000            00130 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00140 ;
000D            00150 CR      EQU     13
002C            00160 PAR_ERR EQU     44              ;"Parameter Error" #
3000            00170 SINIT   EQU     3000H
3000            00180 EINIT   EQU     SINIT
                00190 ;
2400            00200         ORG     2400H
                00210 ;
2400            00220 DUMP    @@CKBRKC                ;Break key down?
2400 3E6A       00001         LD      A,106
2402 EF         00002         RST     40
2403 2804       00230         JR      Z,BEGINA        ;Ok if not
2405 21FFFF     00240         LD      HL,-1           ;  else abort
2408 C9         00250         RET
                00260 ;
2409 111C26     00270 BEGINA  LD      DE,FCB1         ;Fetch the filespec
240C            00280         @@FSPEC
240C 3E4E       00003         LD      A,78
240E EF         00004         RST     40
240F C26E25     00290         JP      NZ,SPCREQ       ;Jump on error
2412 1A         00300         LD      A,(DE)          ;Cannot be a device
2413 FE2A       00310         CP      '*'
2415 CA6E25     00320         JP      Z,SPCREQ        ;Quit if device
2418            00330         @@FLAGS                 ;Get system flag table
2418 3E65       00005         LD      A,101
241A EF         00006         RST     40
241B E5         00340         PUSH    HL              ;Save cmdline ptr
241C FD661A     00350         LD      H,(IY+26)       ;P/u SVC table MSB
241F 2E2C       00360         LD      L,22*2          ;  & point to @EXIT entry
2421 7E         00370         LD      A,(HL)          ;Get @EXIT LSB
2422 2C         00380         INC     L
2423 66         00390         LD      H,(HL)          ;Get @EXIT MSB
2424 6F         00400         LD      L,A
2425 224025     00410         LD      (TPARM+1),HL    ;Init transfer to @EXIT
2428 E1         00420         POP     HL
                00430 ;
                00440 ;       Search for parameters
                00450 ;
2429 11BD25     00460         LD      DE,PRMTBL$      ;Get the parms
242C            00470         @@PARAM
242C 3E11       00007         LD      A,17
242E EF         00008         RST     40
242F C25A25     00480         JP      NZ,PRMERR       ;Jump on parm error
2432 2AE024     00490         LD      HL,(EPARM+1)    ;Ck on end > start
2435 ED4BDA24   00500         LD      BC,(SPARM+1)
2439 AF         00510         XOR     A
243A ED42       00520         SBC     HL,BC
243C DA6A25     00530         JP      C,ENLTST        ;Jump on start < end
243F 213003     00540         LD      HL,SINIT        ;Pt to lowest possible
2442 2B         00550         DEC     HL              ;Reduce for compare
2443 ED42       00560         SBC     HL,BC           ;Ck on start > minimum
2445 D26625     00570         JP      NC,STLT30       ;Jump if start < minimum
2448 010000     00580 APARM   LD      BC,0            ;ASCII txt or code cim
244B 78         00590         LD      A,B
```

```
244C B1        ØØ6ØØ           OR      C
244D 2Ø2B      ØØ61Ø           JR      NZ,DUMPTXT      ;Go if ASCII
244F 111626    ØØ62Ø           LD      DE,NAMFLD       ;Get up to a 6-character
2452 211C26    ØØ63Ø           LD      HL,FCB1         ;Filename to stuff
2455 Ø6Ø6      ØØ64Ø           LD      B,6             ;As file header
2457 7E        ØØ65Ø $?1       LD      A,(HL)
2458 FE3Ø      ØØ66Ø           CP      'Ø'             ;Stop on non-alpha
245A 3813      ØØ67Ø           JR      C,$?3
245C FE3A      ØØ68Ø           CP      '9'+1           ;Use if Ø-9
245E 38Ø8      ØØ69Ø           JR      C,$?2
246Ø FE41      ØØ7ØØ           CP      'A'             ;Ck on A-Z
2462 38ØB      ØØ71Ø           JR      C,$?3
2464 FE5B      ØØ72Ø           CP      'Z'+1
2466 3ØØ7      ØØ73Ø           JR      NC,$?3          ;Exit if not A-Z
2468 12        ØØ74Ø $?2       LD      (DE),A          ;Xfer this char
2469 23        ØØ75Ø           INC     HL              ;Bump input ptr
246A 13        ØØ76Ø           INC     DE              ;  & output pointer
246B 1ØEA      ØØ77Ø           DJNZ    $?1             ;Loop 6-chars max
246D 18Ø6      ØØ78Ø           JR      $?4
246F 3E2Ø      ØØ79Ø $?3       LD      A,' '           ;Place blanks to
2471 12        ØØ8ØØ           LD      (DE),A          ;Fill out to 6 chars
2472 13        ØØ81Ø           INC     DE
2473 1ØFA      ØØ82Ø           DJNZ    $?3
2475 21B725    ØØ83Ø $?4       LD      HL,LMFEXT       ;Use /LMF extension
2478 18Ø3      ØØ84Ø           JR      DUMPCIM
247A 21BA25    ØØ85Ø DUMPTXT   LD      HL,TXTEXT       ;Use /TXT extension
247D 111C26    ØØ86Ø DUMPCIM   LD      DE,FCB1         ;Default the EXT
2480           ØØ87Ø           @@FEXT
2480 3E4F      ØØØØ9           LD      A,79
2482 EF        ØØØ1Ø           RST     4Ø
2483 Ø1ØØØØ    ØØ88Ø LPARM     LD      BC,Ø            ;P/u LRL
2486 78        ØØ89Ø           LD      A,B             ;Test for > 256
2487 B7        ØØ9ØØ           OR      A               ;If hi-order = Ø,
2488 28Ø8      ØØ91Ø           JR      Z,LP1           ;Just use lo-order
248A 3D        ØØ92Ø           DEC     A               ;Test hi-order = 1
248B C25A25    ØØ93Ø           JP      NZ,PRMERR
248E B1        ØØ94Ø           OR      C               ;P/u lo-order
248F C25A25    ØØ95Ø           JP      NZ,PRMERR       ;Lo-order must be Ø
2492 B1        ØØ96Ø LP1       OR      C               ;Merge lo-order
2493 47        ØØ97Ø           LD      B,A
2494 21ØØ27    ØØ98Ø           LD      HL,BUFFER       ;Pt to buffer
2497           ØØ99Ø           @@INIT                  ;Init the file
2497 3E3A      ØØØ11           LD      A,58
2499 EF        ØØØ12           RST     4Ø
249A C25C25    Ø1ØØØ           JP      NZ,IOERR        ;Quit on init error
               Ø1Ø1Ø ;
               Ø1Ø2Ø ;         Display the filespec being dumped
               Ø1Ø3Ø ;
249D ED4B2226  Ø1Ø4Ø           LD      BC,(FCB1+6)     ;P/u DEC & drive
24A1 11F625    Ø1Ø5Ø           LD      DE,FCB2         ;Point to FCB area
24A4 D5        Ø1Ø6Ø           PUSH    DE
24A5           Ø1Ø7Ø           @@FNAME                 ;Fetch the name
24A5 3E5Ø      ØØØ13           LD      A,8Ø
24A7 EF        ØØØ14           RST     4Ø
24A8 E1        Ø1Ø8Ø           POP     HL
24A9 C25C25    Ø1Ø9Ø           JP      NZ,IOERR        ;Quit on error
24AC 3E2Ø      Ø11ØØ           LD      A,2ØH           ;Scan until ETX char
24AE 23        Ø111Ø FNLP      INC     HL
24AF BE        Ø112Ø           CP      (HL)
24BØ 38FC      Ø113Ø           JR      C,FNLP
24B2 36ØD      Ø114Ø           LD      (HL),CR         ;Replace with CR
```

```
24B4            01150          @@LOGOT DUMP$           ;Display "Dumping...
                00015          IFEQ    01H,1
24B4 21ED25     00016          LD      HL,DUMP$
                00017          ENDIF
24B7 3E0C       00018          LD      A,12
24B9 EF         00019          RST     40
                01160 ;
24BA 111C26     01170          LD      DE,FCB1         ;Get dump FCB
24BD 3A4924     01180          LD      A,(APARM+1)     ;Ck if ASCII parm used
24C0 B7         01190          OR      A
24C1 2016       01200          JR      NZ,SPARM        ;  and go if so
24C3 3E05       01210          LD      A,5             ;Name header
24C5 CD5325     01220          CALL    PUTOUT
24C8 3E06       01230          LD      A,6             ;Name length
24CA CD5325     01240          CALL    PUTOUT
24CD 0606       01250          LD      B,6             ;Init loop
24CF 211626     01260          LD      HL,NAMFLD
24D2 7E         01270 $?5      LD      A,(HL)
24D3 23         01280          INC     HL
24D4 CD5325     01290          CALL    PUTOUT          ;Output the filename
24D7 10F9       01300          DJNZ    $?5
                01310 ;
24D9 210030     01320 SPARM    LD      HL,SINIT        ;P/u starting addr
24DC E5         01330 $?7      PUSH    HL              ;Ck on write of
24DD 44         01340          LD      B,H             ;  last byte written
24DE 4D         01350          LD      C,L
24DF 21FF2F     01360 EPARM    LD      HL,EINIT-1      ;Where to end
24E2 23         01370          INC     HL
24E3 AF         01380          XOR     A
24E4 ED42       01390          SBC     HL,BC
24E6 282E       01400          JR      Z,$?10          ;Go if at end
24E8 06FE       01410          LD      B,254           ;254-byte blocks
24EA 7C         01420          LD      A,H             ;A full sector left
24EB B7         01430          OR      A               ;To write?
24EC 2006       01440          JR      NZ,$?8
24EE 7D         01450          LD      A,L
24EF FEFF       01460          CP      0FFH
24F1 3001       01470          JR      NC,$?8          ;If less than full,
24F3 45         01480          LD      B,L             ;  reset len
24F4 E1         01490 $?8      POP     HL
24F5 3A4924     01500          LD      A,(APARM+1)
24F8 B7         01510          OR      A
24F9 2012       01520          JR      NZ,$?9          ;Bypass if TXT
24FB 3C         01530          INC     A               ;Init start of block
24FC CD5325     01540          CALL    PUTOUT
24FF 78         01550          LD      A,B             ;Get block length
2500 C602       01560          ADD     A,2             ;Add 2 for load address
2502 CD5325     01570          CALL    PUTOUT          ;  & put it out
2505 7D         01580          LD      A,L
2506 CD5325     01590          CALL    PUTOUT          ;Lo-order load address
2509 7C         01600          LD      A,H
250A CD5325     01610          CALL    PUTOUT          ;Hi-order load address
250D 7E         01620 $?9      LD      A,(HL)          ;Write a load block
250E 23         01630          INC     HL
250F CD5325     01640          CALL    PUTOUT
2512 10F9       01650          DJNZ    $?9
2514 18C6       01660          JR      $?7             ;Loop for more
                01670 ;
2516 E1         01680 $?10     POP     HL              ;Stack integrity
2517 3A4924     01690          LD      A,(APARM+1)     ;No TRAADR if TXT
251A B7         01700          OR      A               ;  or TRAADR if not TXT
```

```
251B 2818      Ø171Ø          JR      Z,TRAADR
251D Ø1Ø3ØØ    Ø172Ø ETXPARM  LD      BC,3            ;P/u etx character
252Ø 79        Ø173Ø          LD      A,C
2521 21E925    Ø174Ø          LD      HL,ETXRESP
2524 CB7E      Ø175Ø          BIT     7,(HL)          ;Value input means
2526 2Ø1F      Ø176Ø          JR      NZ,PUTETX       ;  put the ETX given
2528 ØA        Ø177Ø          LD      A,(BC)          ;In case string
2529 CB6E      Ø178Ø          BIT     5,(HL)          ;String input puts the
252B 2Ø1A      Ø179Ø          JR      NZ,PUTETX       ;  entered char
252D CB76      Ø18ØØ          BIT     6,(HL)          ;Flag input gives ETX=3
252F 2819      Ø181Ø          JR      Z,CLSFIL        ;  if ETX=on
2531 3EØ3      Ø182Ø          LD      A,3
2533 1812      Ø183Ø          JR      PUTETX
2535 3EØ2      Ø184Ø TRAADR   LD      A,2             ;Put traadr header
2537 CD5325    Ø185Ø          CALL    PUTOUT
253A 3EØ2      Ø186Ø          LD      A,2
253C CD5325    Ø187Ø          CALL    PUTOUT
253F 21ØØØØ    Ø188Ø TPARM    LD      HL,$-$          ;P/u transfer address
2542 7D        Ø189Ø          LD      A,L
2543 CD5325    Ø19ØØ          CALL    PUTOUT          ;Tra lo-order
2546 7C        Ø191Ø          LD      A,H
2547 CD5325    Ø192Ø PUTETX   CALL    PUTOUT          ;Tra hi-order or ETX
254A           Ø193Ø CLSFIL   @@CLOSE                 ;Close 'er up
254A 3E3C      ØØØ2Ø          LD      A,6Ø
254C EF        ØØØ21          RST     4Ø
254D 21ØØØØ    Ø194Ø          LD      HL,Ø
255Ø C8        Ø195Ø          RET     Z               ;Back on no error
2551 18Ø9      Ø196Ø          JR      IOERR           ;Go on error
               Ø197Ø ;
2553 4F        Ø198Ø PUTOUT   LD      C,A             ;Xfer the char
2554           Ø199Ø          @@PUT                   ;Test each byte transfer
2554 3EØ4      ØØØ22          LD      A,4
2556 EF        ØØØ23          RST     4Ø
2557 C8        Ø2ØØØ          RET     Z               ;Back if no error
2558 E1        Ø2Ø1Ø          POP     HL              ;Pop the RET addr
2559 21        Ø2Ø2Ø          DB      21H             ;Skip LD A,## instruction
255A 3E2C      Ø2Ø3Ø PRMERR   LD      A,PAR_ERR       ;"Parameter Error"
255C 6F        Ø2Ø4Ø IOERR    LD      L,A             ;Error code to HL
255D 26ØØ      Ø2Ø5Ø          LD      H,Ø
255F F6CØ      Ø2Ø6Ø          OR      ØCØH            ;Abbrev & return
2561 4F        Ø2Ø7Ø          LD      C,A
2562           Ø2Ø8Ø          @@ERROR                 ;Show the error
2562 3E1A      ØØØ24          LD      A,26
2564 EF        ØØØ25          RST     4Ø
2565 C9        Ø2Ø9Ø          RET
               Ø21ØØ ;
               Ø211Ø ;        Internal error routine
               Ø212Ø ;
2566 213C25    Ø213Ø STLT3Ø   LD      HL,STLT3Ø$
2569 DD        Ø214Ø          DB      ØDDH
256A 217825    Ø215Ø ENLTST   LD      HL,ENLTST$
256D DD        Ø216Ø          DB      ØDDH
256E 21A425    Ø217Ø SPCREQ   LD      HL,SPCREQ$
2571           Ø218Ø          @@LOGOT
               ØØØ26          IFEQ    ØØH,1
               ØØØ27          LD      HL,
               ØØØ28          ENDIF
2571 3EØC      ØØØ29          LD      A,12
2573 EF        ØØØ3Ø          RST     4Ø
2574 21FFFF    Ø219Ø          LD      HL,-1
2577 C9        Ø22ØØ          RET
```

```
                    02210 ;
2578 53             02220 ENLTST$ DB      'START or END error ',CR
     54 41 52 54 20 6F 72 20
     45 4E 44 20 65 72 72 6F
     72 20 0D
258C 53             02230 STLT30$ DB      'Start less than X''3000''',CR
     74 61 72 74 20 6C 65 73
     73 20 74 68 61 6E 20 58
     27 33 30 30 30 27 0D
25A4 46             02240 SPCREQ$ DB      'File spec required',CR
     69 6C 65 20 73 70 65 63
     20 72 65 71 75 69 72 65
     64 0D
25B7 4C             02250 LMFEXT  DB      'LMF'
     4D 46
25BA 54             02260 TXTEXT  DB      'TXT'
     58 54
                    02270 ;
0080                02280 VAL     EQU     80H
0040                02290 SW      EQU     40H
0020                02300 STR     EQU     20H
0010                02310 SGL     EQU     10H
                    02320 ;
25BD 80             02330 PRMTBL$ DB      80H
25BE 95             02340         DB      VAL!SGL!5,'START',0
     53 54 41 52 54 00
25C5 DA24           02350         DW      SPARM+1
25C7 93             02360         DB      VAL!SGL!3,'END',0
     45 4E 44 00
25CC E024           02370         DW      EPARM+1
25CE 93             02380         DB      VAL!SGL!3,'TRA',0
     54 52 41 00
25D3 4025           02390         DW      TPARM+1
25D5 55             02400         DB      SW!SGL!5,'ASCII',0
     41 53 43 49 49 00
25DC 4924           02410         DW      APARM+1
25DE 93             02420         DB      VAL!SGL!3,'LRL',0
     4C 52 4C 00
25E3 8424           02430         DW      LPARM+1
25E5 83             02440         DB      VAL!3,'ETX',0
     45 54 58 00
25E9                02450 ETXRESP EQU     $-1
25EA 1E25           02460         DW      ETXPARM+1
25EC 00             02470         NOP
                    02480 ;
25ED 44             02490 DUMP$   DB      'Dumping: '     ;FCB2 must follow
     75 6D 70 69 6E 67 3A 20
0020                02500 FCB2    DS      32
0006                02510 NAMFLD  DS      6
0020                02520 FCB1    DS      32
2700                02530         ORG     $<-8+1<+8
0100                02540 BUFFER  DS      256
27FF                02550 LAST    EQU     $-1
                    02560 ;
2400                02570         END     DUMP
```

$?1	2457	$?10	2516	$?2	2468
$?3	246F	$?4	2475	$?5	24D2
$?7	24DC	$?8	24F4	$?9	250D
@@1	0000	@@2	0000	@@3	0000
@@4	0000	@MOD2	0000	@MOD4	FFFF
APARM	2448	BEGINA	2409	BUFFER	2700
CLSFIL	254A	CR	000D	DUMP	2400
DUMP$	25ED	DUMPCIM	247D	DUMPTXT	247A
EINIT	3000	ENLTST	256A	ENLTST$	2578
EPARM	24DF	ETXPARM	251D	ETXRESP	25E9
FCB1	261C	FCB2	25F6	FNLP	24AE
IOERR	255C	LAST	27FF	LMFEXT	25B7
LP1	2492	LPARM	2483	NAMFLD	2616
PAR_ERR	002C	PRMERR	255A	PRMTBL$	25BD
PUTETX	2547	PUTOUT	2553	SGL	0010
SINIT	3000	SPARM	24D9	SPCREQ	256E
SPCREQ$	25A4	STLT30	2566	STLT30$	258C
STR	0020	SW	0040	TPARM	253F
TRAADR	2535	TXTEXT	25BA	VAL	0080
@@ABORT	8160	@@ADTSK	81F3	@@BANK	870B
@@BKSP	83EB	@@BREAK	8721	@@CHNIO	814B
@@CKBRKC	876F	@@CKDRV	8247	@@CKEOF	8400
@@CKTSK	81DE	@@CLOSE	83D6	@@CLS	8759
@@CMNDI	818A	@@CMNDR	819F	@@CTL	7FAF
@@DATE	8121	@@DCSTAT	8286	@@DEBUG	81C9
@@DECHEX	868B	@@DIRRD	85F8	@@DIRWR	860D
@@DIV16	8676	@@DIV8	8661	@@DODIR	825C
@@DSP	7F73	@@DSPLY	8013	@@ERROR	81B4
@@EXIT	8175	@@FEXT	8565	@@FLAGS	86F5
@@FNAME	857A	@@FSPEC	8550	@@GATRD	85E3
@@GATWR	8622	@@GET	7F87	@@GTDCB	85A4
@@GTDCT	858F	@@GTMOD	85B9	@@HDFMT	832E
@@HEX16	86CA	@@HEX8	86B5	@@HEXDEC	86A0
@@HIGH$	86DF	@@INIT	83AC	@@KBD	7FEB
@@KEY	7F5F	@@KEYIN	7FFF	@@KLTSK	8232
@@LOAD	8526	@@LOC	8415	@@LOF	842A
@@LOGER	804A	@@LOGOT	805F	@@MSG	8096
@@MUL16	864C	@@MUL8	8637	@@OPEN	83C1
@@PARAM	810C	@@PAUSE	80F7	@@PEOF	843F
@@POSN	8454	@@PRINT	80AB	@@PRT	7FC3
@@PUT	7F9B	@@RAMDIR	8271	@@RDSEC	8304
@@RDSSC	85CE	@@READ	8469	@@REMOV	8397
@@RENAM	8382	@@REW	847E	@@RMTSK	8208
@@RPTSK	821D	@@RREAD	8493	@@RSLCT	82EF
@@RSTOR	82B0	@@RUN	853B	@@RWRIT	84A8
@@SEEK	82DA	@@SEEKSC	84BD	@@SKIP	84D2
@@SLCT	829B	@@STEPI	82C5	@@TIME	8136
@@VDCTL	80E2	@@VER	84E7	@@VRSEC	8319
@@WEOF	84FC	@@WHERE	7FD7	@@WRITE	8511
@@WRSEC	8343	@@WRSSC	8358	@@WRTRK	836D

2400 is the transfer address
00000 Total errors

NOTES:

Command: FORMS

Library: SYS8/SYS

ISAM # : B1H

```
                00100 ;LBFORMS/ASM - Set Line Printer Values
0000            00110         TITLE   <FORMS - LS-DOS 6.2>
                00120 ;
                00130 ;
002C            00140 PAR_ERR EQU     44              ;Parameter Error Code
0007            00150 FLAGBT  EQU     7               ;Flag byte offset
0007            00160 ADDLF   EQU     7               ;Add Line Feed = Bit 0
0007            00170 FFHARD  EQU     7               ;Form Feed Hard = Bit 1
0007            00180 TABV    EQU     7               ;Tab Expansion = Bit 2
0008            00190 CHARS   EQU     8               ;Characters per line
0006            00200 INDENT  EQU     6               ;Indent after wrap-around
0002            00210 LINES   EQU     2               ;Maximum Lines to Print
0009            00220 MARGIN  EQU     9               ;Left hand margin value
0000            00230 PAGE    EQU     0               ;Maximum Lines per page
0004            00240 XLATEF  EQU     4               ;Xlate From
0005            00250 XLATET  EQU     5               ;Xlate To
                00260 ;
0042            00270 PDEF    EQU     66              ;Page Default = 66
0042            00280 LDEF    EQU     66              ;Line Default = 66
                00290 ;
000E            00300 CURON   EQU     0EH             ;Cursor on
000F            00310 CUROFF  EQU     0FH             ;Cursor off
00DD            00320 SKIP    EQU     0DDH            ;Skip 3 byte instruction
                00330 ;
0000            00340 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
0000            00350 *GET    VALUES:3                ;Misc. equates
                03920 ;VALUES/ASM - Version 6
                03930 *LIST OFF
                04200 *LIST ON
                00360 ;
2400            00370         ORG     2400H
                00380 ;
                00390 START
2400 ED732324   00400         LD      (SAVESP+1),SP   ;Save SP loc
2404 CD2924     00410         CALL    FORMS           ;Execute Form Code
2407 210000     00420 EXIT    LD      HL,0            ;Set no error
240A 1816       00430         JR      SAVESP          ;Exit
                00440 ;
                00450 ;       I/O Error Handling
                00460 ;
240C 3E2C       00470 PRMERR  LD      A,PAR_ERR       ;Parameter Error
240E 6F         00480 IOERR   LD      L,A             ;Xfer error # to HL
240F 2600       00490         LD      H,0
2411 F6C0       00500         OR      0C0H            ;Abbrev, return
2413 4F         00510         LD      C,A             ;Xfer to C
2414            00520         @@ERROR                 ;Display error
2414 3E1A       00001         LD      A,26
2416 EF         00002         RST     40
2417 1809       00530         JR      SAVESP          ;Go to exit routine
                00540 ;
                00550 ;       Internal Error Message Handling
                00560 ;
2419 217928     00570 NOPF    LD      HL,NOPF$        ;No filter present
241C            00580         @@LOGOT                 ;Log Message
                00003         IFEQ    00H,1
                00004         LD      HL,
                00005         ENDIF
241C 3E0C       00006         LD      A,12
```

```
241E EF          00007          RST     40
241F 21FFFF      00590 ABORT    LD      HL,-1           ;Set abort code
2422 310000      00600 SAVESP   LD      SP,$-$          ;P/u original SP
2425             00610          @@CKBRKC                ;Clear any <BREAK>
2425 3E6A        00008          LD      A,106
2427 EF          00009          RST     40
2428 C9          00620          RET                     ;And RETurn to DOS
                 00630 ;
                 00640 ;        FORMS - Process the Forms Filter Parameters
                 00650 ;
                 00660 FORMS
2429 CD0027      00670          CALL    DOINIT          ;Do initialization
                 00680 ;
                 00690 ;        Ignore Leading Spaces
                 00700 ;
242C 2B          00710          DEC     HL
242D 23          00720 IGSPCS   INC     HL              ;Bump cmdline ptr
242E 7E          00730          LD      A,(HL)          ;Skip leading spaces
242F FE20        00740          CP      ' '
2431 28FA        00750          JR      Z,IGSPCS
                 00760 ;
                 00770 ;        Any Parameters Entered ?
                 00780 ;
2433 FE0E        00790          CP      CR+1            ;End of line ?
2435 3007        00800          JR      NC,GETPRM       ;Go if not
                 00810 ;
                 00820 ;        Display current parameter settings
                 00830 ;
2437 CD8924      00840 DISPFRM  CALL    DSFORMS         ;Create default string
243A CDF726      00850          CALL    DSPLY           ;Display defaults
243D C9          00860          RET                     ;  and RETurn
                 00870 ;
                 00880 ;        Display "Paramter Error" if Illegal input
                 00890 ;
243E 116227      00900 GETPRM   LD      DE,PRMTBL$      ;Any Paramters ?
2441             00910          @@PARAM
2441 3E11        00010          LD      A,17
2443 EF          00011          RST     40
2444 C20C24      00920          JP      NZ,PRMERR       ;NZ - "Parameter Error"
                 00930 ;
                 00940 ;        Create Xlate From Data Area
                 00950 ;
2447 3ABA27      00960          LD      A,(XTRESP)      ;P/u xlate TO response
244A 32DF27      00970          LD      (XFRESP),A      ;Xfer to FROM response
244D 21D627      00980          LD      HL,XTPARM+1     ;HL => XLATE To
2450 7E          00990          LD      A,(HL)          ;P/u value
2451 3600        01000          LD      (HL),0          ;Set Xlate To msb = 0
2453 32E227      01010          LD      (XFPARM),A      ;Xfer to From parm
                 01020 ;
                 01030 ;        Over-ride all other parms if <D>efault
                 01040 ;
2456 3AC527      01050          LD      A,(DRESP)       ;<D>efault Parm entered ?
2459 B7          01060          OR      A
245A 2817        01070          JR      Z,CHECKQ        ;No - check for <Q>uery
                 01080 ;
                 01090 ;        Overwrite $FF data area with default values
                 01100 ;
245C ED5BD125    01110          LD      DE,(DATAREA+2)  ;DE => Data area start
2460 010A00      01120          LD      BC,10           ;BC = 10 bytes in table
2463 215827      01130          LD      HL,DEFTAB       ;HL => Default Table
2466 E5          01140          PUSH    HL              ;Save regs
```

```
2467 C5        01150          PUSH    BC
2468 EDB0      01160          LDIR                    ;Xfer to $FF data area
246A C1        01170          POP     BC
246B E1        01180          POP     HL
246C 11E329    01190          LD      DE,DUPDA
246F EDB0      01200          LDIR                    ;Xfer to duplicate DA
2471 18C4      01210          JR      DISPFRM         ;Display forms & exit
               01220 ;
               01230 ;        Prompt for any parms not entered & stuff
               01240 ;
2473           01250 CHECKQ   EQU     $
2473 CD1926    01260          CALL    INITVAL         ;Init parm values = def's
2476 CD8924    01270          CALL    DSFORMS         ;Create string
2479 3AAA27    01280          LD      A,(QRESP)       ;<Q>uery parm used?
247C B7        01290          OR      A
247D CCFC24    01300          CALL    Z,CKCOMM        ;Check cmdline values if not
2480 C42A25    01310          CALL    NZ,PROMPT       ;Prompt if "Q"
2483 CDCC25    01320 STUFFIN  CALL    STFPRMS         ;Stuff parms in $FF data
2486 C30724    01330          JP      EXIT            ;Good exit
               01340 ;
               01350 ;        Display Current FORMS value settings
               01360 ;
2489 DDCB0746  01370 DSFORMS  BIT     0,(IX+ADDLF)    ;Add line feed ?
248D 11E128    01380          LD      DE,SADDLF       ;DE => addlf dsply msg
2490 C4E826    01390          CALL    NZ,XFERON       ;Put "ON" in message
               01400 ;
               01410 ;        Display "OFF" if zero, or value if <> zero
               01420 ;
2493 DD7E08    01430          LD      A,(IX+CHARS)    ;CHARS value if it wasn't
2496 B7        01440          OR      A               ;OFF ?
2497 11BA28    01450          LD      DE,SCHARS       ;DE => Chars msg
249A C42127    01460          CALL    NZ,HEXDEC       ;Convert value to dec ASCII
249D 2006      01470          JR      NZ,DOFFHRD      ;Go if Char parm used
               01480 ;
249F 217628    01490 CHAROFF  LD      HL,OFFSTR       ;  else xfer "OFF" into
24A2 CDEB26    01500          CALL    XFER            ;  Chars message
               01510 ;
               01520 ;        FFHARD specified ?
               01530 ;
24A5 DDCB074E  01540 DOFFHRD  BIT     1,(IX+FFHARD)   ;FFHARD parm used?
24A9 11EE28    01550          LD      DE,SFFHARD      ;DE => Ffhard msg
24AC C4E826    01560          CALL    NZ,XFERON       ;Xfer "ON" if set
               01570 ;
               01580 ;        Xfer INDENT value into string
               01590 ;
24AF DD7E06    01600          LD      A,(IX+INDENT)   ;Default value
24B2 11D428    01610          LD      DE,SINDENT      ;DE=> Indent msg
24B5 CD2127    01620          CALL    HEXDEC          ;Convert to decimal ASCII
               01630 ;
               01640 ;        Xfer LINES value into string
               01650 ;
24B8 DD7E02    01660          LD      A,(IX+LINES)    ;P/u LINES value
24BB 11AD28    01670          LD      DE,SLINES       ;Pt to Lines msg
24BE CD2127    01680          CALL    HEXDEC          ;Convert to decimal ASCII
               01690 ;
               01700 ;        Xfer MARGIN value into string
               01710 ;
24C1 DD7E09    01720          LD      A,(IX+MARGIN)   ;P/u MARGIN value
24C4 11C728    01730          LD      DE,SMARGIN      ;DE => Margin msg
24C7 CD2127    01740          CALL    HEXDEC          ;Convert to decimal ASCII
               01750 ;
```

```
                    01760 ;       Xfer PAGE value into string
                    01770 ;
24CA DD7E00         01780         LD      A,(IX+PAGE)     ;P/u page value
24CD 11A028         01790         LD      DE,SPAGE        ;DE => Page msg
24D0 CD2127         01800         CALL    HEXDEC          ;Convert to decimal ASCII
                    01810 ;
                    01820 ;       Xfer "ON" into string if Tab set
                    01830 ;
24D3 DDCB0756       01840         BIT     2,(IX+TABV)     ;Check Tab bit
24D7 11FB28         01850         LD      DE,STAB         ;DE => Tab msg
24DA C4E826         01860         CALL    NZ,XFERON       ;Xfer "ON" if set
                    01870 ;
                    01880 ;       Is Xlate FROM = Xlate TO ?
                    01890 ;
24DD DD7E04         01900         LD      A,(IX+XLATEF)   ;P/u FROM byte
24E0 DD4605         01910         LD      B,(IX+XLATET)   ;P/u TO byte
24E3 B8             01920         CP      B               ;Same ?
24E4 2812           01930         JR      Z,NOSHOW        ;Yes - no show
                    01940 ;
                    01950 ;       Two distinct values - convert to Hex
                    01960 ;
24E6 21FF28         01970         LD      HL,DOXLATE      ;Change CR to LF
24E9 360A           01980         LD      (HL),LF         ;  so msg will dsply
                    01990 ;
24EB 210B29         02000         LD      HL,SXLFROM      ;"From" message
24EE CD3C27         02010         CALL    HEX8            ;Convert A to Hex @ HL
24F1 78             02020         LD      A,B             ;P/u TO
24F2 211429         02030         LD      HL,SXLTO        ;"To" message
24F5 CD3C27         02040         CALL    HEX8            ;Convert A to Hex @ HL
                    02050 ;
                    02060 ;       Point HL to string & RETurn
                    02070 ;
24F8 219728         02080 NOSHOW  LD      HL,VALUES       ;HL => Default val string
24FB C9             02090         RET                     ;RETurn
                    02100 ;
                    02110 ;
                    02120 ;       CKCOMM - Check command line parameter values
                    02130 ;
                    02140 ;
24FC 1E0A           02150 CKCOMM  LD      E,10            ;10 values to check
24FE FD210E28       02160         LD      IY,STRTAB       ;IY => Response table
                    02170 ;
2502 FD6E01         02180 CKCOMML LD      L,(IY+1)        ;P/u address of response
2505 FD6602         02190         LD      H,(IY+2)
                    02200 ;
                    02210 ;       Set BC = Parameter Response
                    02220 ;
2508 7E             02230         LD      A,(HL)          ;Was anything entered ?
2509 B7             02240         OR      A
250A 23             02250         INC     HL              ;Parm addr follows resp
250B 4E             02260         LD      C,(HL)          ;Set HL = (HL)
250C 23             02270         INC     HL
250D 66             02280         LD      H,(HL)
250E 69             02290         LD      L,C
250F 4E             02300         LD      C,(HL)          ;P/u response value
2510 23             02310         INC     HL              ;  into BC
2511 46             02320         LD      B,(HL)
                    02330 ;
                    02340 ;       Call routine to Range check parm entry
                    02350 ;
2512 FD6E05         02360         LD      L,(IY+5)        ;P/u address of routine
```

```
2515 FD6606   02370          LD      H,(IY+6)          ;  to check value validity.
2518 221C25   02380          LD      (CALLINS+1),HL    ;Stuff addr to CALL instr
251B C40000   02390 CALLINS  CALL    NZ,$-$            ;BC = response, A = type
251E C20C24   02400          JP      NZ,PRMERR         ;NZ - "Parameter Error"
              02410 ;
              02420 ;        Position to next table entry
              02430 ;
2521 010900   02440          LD      BC,9              ;Pos to next STRTAB entry
2524 FD09     02450          ADD     IY,BC
2526 1D       02460          DEC     E                 ;Done ?
2527 20D9     02470          JR      NZ,CKCOMML
2529 C9       02480          RET                       ;Yes - RETurn
              02490 ;
              02500 ;
              02510 ;        PROMPT - for any vals not entered in parm line
              02520 ;
252A 060A     02530 PROMPT   LD      B,10              ;Eight normal + 2 Xlates
252C FD210E28 02540          LD      IY,STRTAB         ;Prompt, response table
              02550 ;
              02560 ;        P/u type byte from table & set length = 1
              02570 ;
2530 FD7E00   02580 PROMPTL  LD      A,(IY)            ;P/u type byte
2533 3C       02590          INC     A                 ;Merge length = 1
2534 326928   02600          LD      (FAKETAB+1),A     ;Store new type byte
              02610 ;
              02620 ;        P/u address of response byte
              02630 ;
2537 FD5E01   02640 REINPUT  LD      E,(IY+1)          ;P/u address
253A FD5602   02650          LD      D,(IY+2)          ;  in DE
              02660 ;
              02670 ;        Pick up Prompt string address & display it
              02680 ;
253D FD6E03   02690 DOPRMPT  LD      L,(IY+3)          ;P/u address in HL
2540 FD6604   02700          LD      H,(IY+4)
2543 CD5825   02710          CALL    DISPROM           ;P/u default & display
              02720 ;
              02730 ;        Input response & stuff into Parm table
              02740 ;
2546 CD4327   02750          CALL    INPUT             ;Input value
2549 C5       02760          PUSH    BC                ;Save count
254A C4A325   02770          CALL    NZ,STUFVAL        ;Stuff in valid input
254D C1       02780          POP     BC                ;Restore count
254E 20E7     02790          JR      NZ,REINPUT        ;Re-input if bad value
              02800 ;
              02810 ;        Position to next table entry
              02820 ;
2550 110900   02830 NEXTPR   LD      DE,9              ;9 bytes per entry
2553 FD19     02840          ADD     IY,DE
2555 10D9     02850          DJNZ    PROMPTL           ;B prompts
2557 C9       02860          RET                       ;Done
              02870 ;
              02880 ;
              02890 ;        DISPROM - Display Prompt
              02900 ;
2558 D5       02910 DISPROM  PUSH    DE                ;Save regs
2559 C5       02920          PUSH    BC
255A 0E0F     02930          LD      C,CUROFF          ;Turn off cursor
255C CDF126   02940          CALL    DSP
255F 0620     02950          LD      B,32              ;Space padding base
              02960 ;
2561 4E       02970 PRLP     LD      C,(HL)            ;P/u character
```

```
2562 23       02980          INC     HL              ;Pos to next
2563 05       02990          DEC     B               ;Dec count
2564 CDF126   03000          CALL    DSP             ;Output byte
2567 79       03010          LD      A,C             ;P/u char
2568 FE7B     03020          CP      '{'             ;Bracket ?
256A 20F5     03030          JR      NZ,PRLP         ;No - go til bracket
256C CD8225   03040          CALL    STUFDEF         ;Display default
256F 78       03050          LD      A,B             ;P/u base #
2570 81       03060          ADD     A,C             ;  & calculate # of
2571 47       03070          LD      B,A             ;Spaces to print
2572 0E20     03080          LD      C,' '
              03090 ;
2574 CDF126   03100 SPLP     CALL    DSP             ;Output spaces
2577 10FB     03110          DJNZ    SPLP
              03120 ;
2579 219328   03130          LD      HL,ENDPROM      ;End of prompt
257C CDF726   03140          CALL    DSPLY
257F C1       03150          POP     BC              ;Recover regs
2580 D1       03160          POP     DE
2581 C9       03170          RET                     ;  and RETurn
              03180 ;
              03190 ;
              03200 ;        STUFDEF - Stuff default value in prompt
              03210 ;
2582 FD6E07   03220 STUFDEF  LD      L,(IY+7)        ;P/u default string
2585 FD6608   03230          LD      H,(IY+8)        ;  address
2588 0E05     03240          LD      C,5             ;5 chars max
258A 7E       03250 PNLP     LD      A,(HL)
258B 23       03260          INC     HL              ;Bump source
258C FE0A     03270          CP      LF              ;Done ?
258E 280A     03280          JR      Z,DUNLP
2590 FE20     03290          CP      ' '             ;Leading space ?
2592 28F6     03300          JR      Z,PNLP          ;Yes - ignore it
2594 CD9C25   03310          CALL    DISPA           ;Output A
              03320 ;
2597 0D       03330 PNLP2    DEC     C               ;Dec count
2598 20F0     03340          JR      NZ,PNLP
              03350 ;
259A 3E7D     03360 DUNLP    LD      A,'}'           ;Output end bracket
259C C5       03370 DISPA    PUSH    BC              ;Save count in C
259D 4F       03380          LD      C,A             ;Xfer char to C
259E CDF126   03390          CALL    DSP             ;Output byte
25A1 C1       03400          POP     BC              ;Recover C
25A2 C9       03410          RET                     ;  and RETurn
              03420 ;
              03430 ;
              03440 ;        STUFVAL - Stuff values into Parm Table
              03450 ;
25A3 D5       03460 STUFVAL  PUSH    DE              ;DE => Response Byte
25A4 21FF27   03470          LD      HL,FAKEPRM      ;HL => Fake Parm Entry
25A7 116828   03480          LD      DE,FAKETAB      ;DE => Fake Parm Table
25AA          03490          @@PARAM                 ;Parse entry
25AA 3E11     00012          LD      A,17
25AC EF       00013          RST     40
25AD E1       03500          POP     HL              ;HL => Response
25AE C0       03510          RET     NZ              ;NZ - Re-input
              03520 ;
              03530 ;        Stuff response into Parameter Table
              03540 ;
25AF E5       03550          PUSH    HL              ;Save response dest
25B0 3A6B28   03560          LD      A,(FAKERES)     ;P/u response
```

```
25B3 010000    03570 VALUE   LD      BC,$-$          ;P/u value
25B6 23        03580         INC     HL              ;HL => Parm Address
25B7 5E        03590         LD      E,(HL)          ;P/u parm address
25B8 23        03600         INC     HL
25B9 56        03610         LD      D,(HL)
25BA EB        03620         EX      DE,HL           ;HL => Parm lsb
25BB 71        03630         LD      (HL),C          ;Stuff response in table
               03640 ;
               03650 ;       CALL range checking routine
               03660 ;
25BC 21C725    03670         LD      HL,RETADR       ;Put RET addr on stack
25BF E5        03680         PUSH    HL
25C0 FD6E05    03690         LD      L,(IY+5)        ;P/u addr of range
25C3 FD6606    03700         LD      H,(IY+6)        ;Checking in HL
25C6 E9        03710         JP      (HL)            ;Routine sets Z for stat
25C7 E1        03720 RETADR  POP     HL              ;HL => Response byte
25C8 C0        03730         RET     NZ              ;Don't change if NZ
25C9 3680      03740         LD      (HL),80H        ;  else stuff non-zero
25CB C9        03750         RET                     ;  value for response
               03760 ;
               03770 ;
               03780 ;       STFPRMS - Stuff Numeric & Flag Parms into $FF
               03790 ;
               03800 ;
               03810 ;       Pt HL => Response byte addr & offset Table
               03820 ;
25CC 21E427    03830 STFPRMS LD      HL,RESPTAB      ;HL => Response Table
25CF DD210000  03840 DATAREA LD      IX,$-$          ;P/u Data Area pointer
25D3 0607      03850         LD      B,7             ;7 numeric values
               03860 ;
               03870 ;       P/u response byte & offset byte to $FF data
               03880 ;
25D5 5E        03890 STUFLP  LD      E,(HL)          ;P/u response address
25D6 23        03900         INC     HL
25D7 56        03910         LD      D,(HL)
25D8 23        03920         INC     HL              ;HL => $FF data offset
25D9 4E        03930         LD      C,(HL)          ;P/u offset in data area
25DA 23        03940         INC     HL
               03950 ;
25DB 1A        03960         LD      A,(DE)          ;P/u response
25DC B7        03970         OR      A               ;Parm entered ?
               03980 ;
               03990 ;       Parm entered - calculate Parm's Location
               04000 ;
25DD 13        04010         INC     DE              ;DE => Parameter Dest
25DE EB        04020         EX      DE,HL           ;Xfer to HL
25DF 7E        04030         LD      A,(HL)          ;Set HL = (HL)
25E0 23        04040         INC     HL
25E1 66        04050         LD      H,(HL)
25E2 6F        04060         LD      L,A
               04070 ;
               04080 ;       Stuff parm response into $FF data region
               04090 ;
25E3 79        04100         LD      A,C             ;Xfer offset to A
25E4 4E        04110         LD      C,(HL)          ;P/u lsb of Parm response
25E5 EB        04120         EX      DE,HL           ;Recover HL (Table ptr)
25E6 2806      04130 NOPOUT  JR      Z,NOPARM        ;No - don't stuff
25E8 32ED25    04140         LD      (IXINST+2),A    ;Modify offset in IX inst
25EB DD7100    04150 IXINST  LD      (IX+$-$),C      ;Xfer parm resp to $FF
               04160 ;
25EE 10E5      04170 NOPARM  DJNZ    STUFLP          ;Next entry
```

```
                    Ø418Ø ;
                    Ø419Ø ;       Set Flag bits in $FF data area if parms set
                    Ø42ØØ ;
25FØ Ø6Ø3           Ø421Ø GETFLAG LD      B,3               ;3 flag values
25F2 5E             Ø422Ø FLOOP   LD      E,(HL)            ;P/u response address
25F3 23             Ø423Ø         INC     HL
25F4 56             Ø424Ø         LD      D,(HL)
25F5 23             Ø425Ø         INC     HL
25F6 1A             Ø426Ø         LD      A,(DE)            ;Entered ?
25F7 B7             Ø427Ø         OR      A
25F8 281C           Ø428Ø         JR      Z,NEXTFLG         ;No - get next one
                    Ø429Ø ;
                    Ø43ØØ ;       Response - If true (SET), False (RES)
                    Ø431Ø ;
25FA 13             Ø432Ø         INC     DE                ;Pos to parm address
25FB EB             Ø433Ø         EX      DE,HL             ;P/u Parm
25FC 7E             Ø434Ø         LD      A,(HL)            ;Set HL = (HL)
25FD 23             Ø435Ø         INC     HL
25FE 66             Ø436Ø         LD      H,(HL)
25FF 6F             Ø437Ø         LD      L,A
26ØØ EB             Ø438Ø         EX      DE,HL             ;Put into DE
2601 ØE86           Ø439Ø         LD      C,1ØØØØ11ØB       ;Default = Reset bit inst
2603 1A             Ø44ØØ         LD      A,(DE)            ;P/u lsb of parm
2604 B7             Ø441Ø         OR      A                 ;Set ?
2605 2802           Ø442Ø         JR      Z,SKIPSET         ;No - skip SET inst
2607 CBF1           Ø443Ø         SET     6,C               ;Change to Set bit inst
                    Ø444Ø ;
                    Ø445Ø ;       Create Post opcode for IX instruction
                    Ø446Ø ;
2609 78             Ø447Ø SKIPSET LD      A,B               ;P/u bit # (Ø-2)
260A 3D             Ø448Ø         DEC     A
260B Ø7             Ø449Ø         RLCA                      ;Move to bits 3-5
260C Ø7             Ø45ØØ         RLCA
260D Ø7             Ø451Ø         RLCA
260E B1             Ø452Ø         OR      C                 ;Post op code
260F 321526         Ø453Ø         LD      (IXINST2+3),A     ;Change RES b,(IX+nn) ins
2612 DDCBØ786       Ø454Ø IXINST2 RES     $-$,(IX+FLAGBT)   ;Set/Reset bit B in $FF
2616 1ØDA           Ø455Ø NEXTFLG DJNZ    FLOOP             ;Get next flag
2618 C9             Ø456Ø         RET                       ;Done - RETurn
                    Ø457Ø ;
                    Ø458Ø ;
                    Ø459Ø ;       INITVAL - Initial Parm values
                    Ø46ØØ ;
                    Ø461Ø ;
2619 Ø6Ø5           Ø462Ø INITVAL LD      B,5               ;5 values to stuff
261B 21E427         Ø463Ø         LD      HL,RESPTAB        ;HL => Response & offsets
                    Ø464Ø ;
261E 5E             Ø465Ø SDLP    LD      E,(HL)            ;P/u response byte addr
261F 23             Ø466Ø         INC     HL
2620 56             Ø467Ø         LD      D,(HL)
2621 23             Ø468Ø         INC     HL
2622 1A             Ø469Ø         LD      A,(DE)            ;P/u response byte
                    Ø47ØØ ;
                    Ø471Ø ;       Get parm table address - DE = (DE)
                    Ø472Ø ;
2623 EB             Ø473Ø         EX      DE,HL
2624 23             Ø474Ø         INC     HL                ;Parm address after resp
2625 4E             Ø475Ø         LD      C,(HL)            ;P/u lsb
2626 23             Ø476Ø         INC     HL
2627 66             Ø477Ø         LD      H,(HL)            ;P/u msb
2628 69             Ø478Ø         LD      L,C               ;HL = (HL)
```

```
2629 EB       04790          EX      DE,HL           ;Get back to DE
              04800 ;
              04810 ;        P/u default value from $FF data area
              04820 ;
262A D5       04830          PUSH    DE              ;Save HL & DE
262B E5       04840          PUSH    HL
262C 5E       04850          LD      E,(HL)          ;P/u offset
262D 1600     04860          LD      D,0             ;DE = offset to default
262F 2AD125   04870          LD      HL,(DATAREA+2)  ;HL => Data Area
2632 19       04880          ADD     HL,DE
2633 4E       04890          LD      C,(HL)          ;P/u default value
2634 E1       04900          POP     HL              ;Restore regs
2635 D1       04910          POP     DE
              04920 ;
              04930 ;        If parm wasn't entered - stuff default value
              04940 ;
2636 23       04950          INC     HL              ;Posn to next entry
2637 B7       04960          OR      A               ;Parm entered ?
2638 2806     04970          JR      Z,STFDEF        ;No - stuff default
263A 3AAA27   04980          LD      A,(QRESP)       ;<Q>uery parm used?
263D B7       04990          OR      A
263E 2802     05000          JR      Z,PRMENT        ;No - don't stuff
2640 79       05010 STFDEF   LD      A,C             ;Yes - stuff default
2641 12       05020          LD      (DE),A
2642 10DA     05030 PRMENT   DJNZ    SDLP
2644 C9       05040          RET                     ;Done
              05050 ;
              05060 ;
              05070 ;        Range Checking Code of Values
              05080 ;
              05090 ;
              05100 ;        Is the Page length valid ?
              05110 ;
2645 CDD126   05120 RPAGE    CALL    MORE0?          ;Number between 1 - 255 ?
2648 C0       05130          RET     NZ              ;No - NZ
2649 3ACF27   05140          LD      A,(LPARM)       ;P/u LINES value
264C 3D       05150          DEC     A
264D B9       05160          CP      C               ;LINES > PAGE ?
264E F5       05170          PUSH    AF              ;Save status
264F 3AAA27   05180          LD      A,(QRESP)       ;<Q>uery parm used?
2652 B7       05190          OR      A
2653 2003     05200          JR      NZ,PQUERY       ;Go if so
2655 F1       05210          POP     AF              ;No
2656 1872     05220          JR      VALID2?         ;Return NZ if L>P
2658 F1       05230 PQUERY   POP     AF              ;L > P ?
2659 380D     05240          JR      C,SETZ          ;No - Set Z flag
265B 79       05250          LD      A,C             ;Yes - Set LINES = PAGE
265C 32E529   05260          LD      (DUPDA+LINES),A
265F 328F27   05270          LD      (LRESP),A       ;Pretend that LINES was
2662 32CF27   05280          LD      (LPARM),A       ;  responded to
2665 CD8924   05290          CALL    DSFORMS         ;Reset defaults
2668 BF       05300 SETZ     CP      A               ;Set Z flag
2669 C9       05310          RET
              05320 ;
              05330 ;        Is the lines printed per page valid ?
              05340 ;
266A CDD126   05350 RLINES   CALL    MORE0?          ;Number between 1 - 255 ?
266D C0       05360          RET     NZ              ;No - NZ
266E 3D       05370          DEC     A
266F 21D327   05380          LD      HL,PPARM        ;HL => Page length
2672 1855     05390          JR      VALID1?         ;Set status accordingly
```

```
                    Ø54ØØ ;
                    Ø541Ø ;       Is the Characters printed per line valid ?
                    Ø542Ø ;
2674 CB77           Ø543Ø RCHARS  BIT     6,A             ;Flag response ?
2676 2ØFØ           Ø544Ø         JR      NZ,SETZ         ;Yes - Set Z
2678 CDD126         Ø545Ø         CALL    MOREØ?          ;No - More than zero ?
267B CØ             Ø546Ø         RET     NZ              ;No - NZ
267C 3AAA27         Ø547Ø         LD      A,(QRESP)       ;<Q>uery parm used?
267F B7             Ø548Ø         OR      A
2680 C8             Ø549Ø         RET     Z               ;Return if not
                    Ø55ØØ ;
                    Ø551Ø ;       <Q>uery - Make sure CHARS > INDENT+MARGIN
                    Ø552Ø ;
2681 21D127         Ø553Ø         LD      HL,MPARM        ;HL => Margin value
2684 3ACD27         Ø554Ø         LD      A,(IPARM)       ;A = Indent value
2687 86             Ø555Ø         ADD     A,(HL)          ;A = Indent + Margin
2688 B9             Ø556Ø         CP      C               ;Less than CHARS ?
2689 38DD           Ø557Ø         JR      C,SETZ          ;Yes - Set Z
268B AF             Ø558Ø         XOR     A               ;Reset INDENT & MARGIN=Ø
268C 32E929         Ø559Ø         LD      (DUPDA+INDENT),A
268F 32EC29         Ø56ØØ         LD      (DUPDA+MARGIN),A
2692 32CD27         Ø561Ø         LD      (IPARM),A
2695 32D127         Ø562Ø         LD      (MPARM),A
2698 3C             Ø563Ø         INC     A               ;Pretend that INDENT &
2699 329927         Ø564Ø         LD      (MRESP),A       ;  MARGIN were responded
269C 328627         Ø565Ø CHNGIND LD      (IRESP),A       ;  to
269F CD8924         Ø566Ø         CALL    DSFORMS         ;Change defaults
26A2 AF             Ø567Ø         XOR     A               ;Set Z & RETurn
26A3 C9             Ø568Ø         RET
                    Ø569Ø ;
                    Ø57ØØ ;       Is Margin less than Characters/Line ?
                    Ø571Ø ;
26A4 CDDA26         Ø572Ø RMARGIN CALL    NUMERIC         ;Number between Ø - 255 ?
26A7 CØ             Ø573Ø         RET     NZ              ;No - NZ
26A8 CDC626         Ø574Ø         CALL    VALID?          ;Yes - less than CHARS ?
26AB CØ             Ø575Ø         RET     NZ              ;No - RETurn NZ
26AC 3ACD27         Ø576Ø         LD      A,(IPARM)       ;P/u INDENT
26AF 81             Ø577Ø         ADD     A,C             ;Add to MARGIN
26BØ CDC626         Ø578Ø         CALL    VALID?          ;M + I < CHARS ?
26B3 C8             Ø579Ø         RET     Z               ;Yes - RETurn Z
26B4 AF             Ø58ØØ         XOR     A               ;No - Set INDENT default
26B5 32CD27         Ø581Ø         LD      (IPARM),A       ;Equal to Zero
26B8 32E929         Ø582Ø         LD      (DUPDA+INDENT),A
26BB 3C             Ø583Ø         INC     A               ;Pretend I was responded
26BC 18DE           Ø584Ø         JR      CHNGIND         ;  to
                    Ø585Ø ;
                    Ø586Ø ;       Is Margin + Indent less than chars/line ?
                    Ø587Ø ;
26BE CDDA26         Ø588Ø RINDENT CALL    NUMERIC         ;Number between Ø - 255 ?
26C1 CØ             Ø589Ø         RET     NZ              ;No - NZ
26C2 3AD127         Ø59ØØ         LD      A,(MPARM)       ;P/u MARGIN val
26C5 81             Ø591Ø         ADD     A,C             ;A = MARGIN + INDENT
26C6 21CB27         Ø592Ø VALID?  LD      HL,CPARM        ;HL => Characters/Line
26C9 BE             Ø593Ø VALID1? CP      (HL)            ;Response > (HL) ?
26CA 3ØØ2           Ø594Ø VALID2? JR      NC,SETNZ        ;Yes - Reset Z flag
26CC BF             Ø595Ø         CP      A               ;No - Set Z flag
26CD C9             Ø596Ø         RET
26CE AF             Ø597Ø SETNZ   XOR     A               ;Reset Z flag
26CF 3C             Ø598Ø         INC     A
26DØ C9             Ø599Ø         RET
                    Ø6ØØØ ;
```

```
                    Ø6Ø1Ø ;       Is the response a number between 1-255 ?
                    Ø6Ø2Ø ;
26D1 CDDA26         Ø6Ø3Ø MOREØ?  CALL    NUMERIC          ;Is the response a number
26D4 CØ             Ø6Ø4Ø         RET     NZ               ;Between Ø - 255 ?
26D5 B7             Ø6Ø5Ø         OR      A                ;Is the response zero ?
26D6 28F6           Ø6Ø6Ø         JR      Z,SETNZ          ;Yes - reset Z flag
26D8 BF             Ø6Ø7Ø         CP      A                ;No - set Z flag
26D9 C9             Ø6Ø8Ø         RET
                    Ø6Ø9Ø ;
                    Ø61ØØ ;       Is the response a 1 byte number ?
                    Ø611Ø ;
26DA E68Ø           Ø612Ø NUMERIC AND     8ØH              ;Bit 7 is set if the
26DC EE8Ø           Ø613Ø         XOR     8ØH              ;Response is numeric.
26DE CØ             Ø614Ø         RET     NZ               ;NZ <= if Bit is reset
26DF Ø4             Ø615Ø         INC     B                ;Is the response only
26EØ Ø5             Ø616Ø         DEC     B                ;1 byte (msb = Ø) ?
26E1 79             Ø617Ø         LD      A,C              ;Set A = response
26E2 C9             Ø618Ø         RET                      ;Yes (Z), no (NZ)
                    Ø619Ø ;
                    Ø62ØØ ;       Is the response a flag (ON/YES, OFF/NO) ?
                    Ø621Ø ;
26E3 E64Ø           Ø622Ø FLAG?   AND     4ØH              ;Bit 6 is set if the
26E5 EE4Ø           Ø623Ø         XOR     4ØH              ;Response is a flag.
26E7 C9             Ø624Ø         RET                      ;Yes (Z), no (NZ)
                    Ø625Ø ;
                    Ø626Ø ;
                    Ø627Ø ;       XFER - Xfer string @ HL to DE
                    Ø628Ø ;       XFERON - Xfer "ON" string to DE
                    Ø629Ø ;
                    Ø63ØØ ;
26E8 217328         Ø631Ø XFERON  LD      HL,ONSTR         ;HL => "ON"
26EB Ø1Ø3ØØ         Ø632Ø XFER    LD      BC,3             ;3 chars to xfer
26EE EDBØ           Ø633Ø         LDIR
26FØ C9             Ø634Ø         RET
                    Ø635Ø ;
                    Ø636Ø ;
                    Ø637Ø ;       DSP - Display a byte
                    Ø638Ø ;
                    Ø639Ø ;
26F1 D5             Ø64ØØ DSP     PUSH    DE               ;Save DE
26F2                Ø641Ø         @@DSP                    ;Output byte
26F2 3EØ2           ØØØ14         LD      A,2
26F4 EF             ØØØ15         RST     4Ø
26F5 18Ø4           Ø642Ø         JR      EXDSP
                    Ø643Ø ;
                    Ø644Ø ;
                    Ø645Ø ;       DSPLY - Display a string
                    Ø646Ø ;
                    Ø647Ø ;
26F7 D5             Ø648Ø DSPLY   PUSH    DE               ;Save DE
26F8                Ø649Ø         @@DSPLY                  ;Display it
                    ØØØ16         IFEQ    ØØH,1
                    ØØØ17         LD      HL,
                    ØØØ18         ENDIF
26F8 3EØA           ØØØ19         LD      A,1Ø
26FA EF             ØØØ2Ø         RST     4Ø
26FB D1             Ø65ØØ EXDSP   POP     DE
26FC C8             Ø651Ø         RET     Z                ;Return if good
26FD C3ØE24         Ø652Ø         JP      IOERR            ;NZ - I/O Error
                    Ø653Ø ;
                    Ø654Ø ;
```

```
                    Ø655Ø ;       DOINIT - Sign on message & Get Data area
                    Ø656Ø ;
                    Ø657Ø ;
2700 E5             Ø658Ø DOINIT  PUSH    HL              ;Save command ptr
2701                Ø659Ø         @@FLAGS                 ;Get system flags
2701 3E65           ØØØ21         LD      A,101
2703 EF             ØØØ22         RST     40
                    Ø66ØØ ;
                    Ø661Ø ;       Point IX to Filter Data area
                    Ø662Ø ;
2704 116F28         Ø663Ø         LD      DE,$FF          ;DE => "$FF"
2707                Ø664Ø         @@GTMOD                 ;Find start
2707 3E53           ØØØ23         LD      A,83
2709 EF             ØØØ24         RST     40
270A C21924         Ø665Ø         JP      NZ,NOPF         ;Abort if Forms/Flt missing
                    Ø666Ø ;
270D EB             Ø667Ø         EX      DE,HL           ;HL => Data Area
270E Ø1Ø4ØØ         Ø668Ø         LD      BC,4            ;Add 4 to ptr
2711 Ø9             Ø669Ø         ADD     HL,BC
2712 22D125         Ø67ØØ         LD      (DATAREA+2),HL  ;Save $FF data pointer
2715 11E329         Ø671Ø         LD      DE,DUPDA        ;DE => Duplicate D area
2718 D5             Ø672Ø         PUSH    DE              ;Save ptr
2719 ØEØA           Ø673Ø         LD      C,10            ;BC = 10 bytes to xfer
271B EDBØ           Ø674Ø         LDIR
271D DDE1           Ø675Ø         POP     IX              ;IX pts to data area
271F E1             Ø676Ø         POP     HL              ;Recover cmdline ptr
2720 C9             Ø677Ø         RET                     ;  and RETurn
                    Ø678Ø ;
                    Ø679Ø ;
                    Ø68ØØ ;
                    Ø681Ø ;       HEXDEC - Convert Hex Number to Decimal ASCII
                    Ø682Ø ;       A => 8-bit Hex Number to Convert
                    Ø683Ø ;       DE => Destination of ASCII characters
                    Ø684Ø ;
                    Ø685Ø ;
2721 C5             Ø686Ø HEXDEC  PUSH    BC              ;Save regs
2722 E5             Ø687Ø         PUSH    HL
2723 F5             Ø688Ø         PUSH    AF
                    Ø689Ø ;
                    Ø69ØØ ;       Transfer ASCII chars into temporary buffer
                    Ø691Ø ;
2724 D5             Ø692Ø         PUSH    DE              ;Save real destination
2725 11DE29         Ø693Ø         LD      DE,TEMBUF       ;DE => Temporary buffer
2728 26ØØ           Ø694Ø         LD      H,Ø             ;Xfer # to HL
272A 6F             Ø695Ø         LD      L,A
272B                Ø696Ø         @@HEXDEC                ;Convert to ASCII
272B 3E61           ØØØ25         LD      A,97
272D EF             ØØØ26         RST     40
272E 1B             Ø697Ø         DEC     DE              ;Pos to 3-byte field
272F 1B             Ø698Ø         DEC     DE
2730 1B             Ø699Ø         DEC     DE
2731 E1             Ø7ØØØ         POP     HL              ;Recover user buffer
2732 EB             Ø7Ø1Ø         EX      DE,HL           ;HL to #, DE to user buff
2733 Ø1Ø3ØØ         Ø7Ø2Ø         LD      BC,3
2736 EDBØ           Ø7Ø3Ø         LDIR                    ;Move the ASCII number
                    Ø7Ø4Ø ;
2738 F1             Ø7Ø5Ø         POP     AF              ;Recover #
2739 E1             Ø7Ø6Ø         POP     HL              ;  and other regs
273A C1             Ø7Ø7Ø         POP     BC
273B C9             Ø7Ø8Ø         RET
                    Ø7Ø9Ø ;
```

```
                    07100 ;
                    07110 ;       HEX8 - Convert HEX Number in A to HEX @ HL
                    07120 ;
                    07130 ;
273C C5             07140 HEX8    PUSH    BC              ;Save regs
273D 4F             07150         LD      C,A             ;Xfer char to C
273E                07160         @@HEX8                  ;Do it
273E 3E62           00027         LD      A,98
2740 EF             00028         RST     40
2741 C1             07170         POP     BC
2742 C9             07180         RET                     ;  and RETurn
                    07190 ;
                    07200 ;
                    07210 ;
                    07220 ;       INPUT - Input a string into INBUFF$
                    07230 ;
                    07240 ;
2743 E5             07250 INPUT   PUSH    HL              ;Save regs
2744 D5             07260         PUSH    DE
2745 C5             07270         PUSH    BC
                    07280 ;
2746 010003         07290         LD      BC,3<8          ;3 chars max
2749 210228         07300         LD      HL,INBUFF$      ;Key input buffer
274C                07310         @@KEYIN                 ;Input line
274C 3E09           00029         LD      A,9
274E EF             00030         RST     40
274F DA1F24         07320         JP      C,ABORT         ;Abort if <BREAK>
                    07330 ;
2752 04             07340         INC     B               ;Set Z flag if
2753 05             07350         DEC     B               ;  no input
                    07360 ;
2754 C1             07370         POP     BC              ;Restore regs
2755 D1             07380         POP     DE
2756 E1             07390         POP     HL
2757 C9             07400         RET                     ;  & RETurn with condition
                    07410 ;
                    07420 ;       Default Value Table
                    07430 ;
2758 42             07440 DEFTAB  DB      PDEF,0,LDEF,0,0,0,0,00000100B,0,0
     00 42 00 00 00 00 04 00
     00
                    07450 ;
                    07460 ;       Parameter table
                    07470 ;
2762 80             07480 PRMTBL$ DB      80H             ;6.2 @PARAM
                    07490 ;
                    07500 ;       ADDLF (A) - Flag Input Only
                    07510 ;
2763 55             07520         DB      FLAG!ABB!5
2764 41             07530         DB      'ADDLF'
     44 44 4C 46
2769 00             07540 ARESP   DB      0
276A D727           07550         DW      APARM
                    07560 ;
                    07570 ;       CHARS (C) - Accept Numeric or Flag input
                    07580 ;
276C D5             07590         DB      FLAG!ABB!NUM!5
276D 43             07600         DB      'CHARS'
     48 41 52 53
2772 00             07610 CRESP   DB      0
2773 CB27           07620         DW      CPARM
```

```
                    07630 ;
                    07640 ;       FFHARD (F) - Accept Flag input only
                    07650 ;
2775 56             07660         DB      FLAG!ABB!6
2776 46             07670         DB      'FFHARD'
     46 48 41 52 44
277C ØØ             07680 FRESP   DB      Ø
277D D927           07690         DW      FPARM
                    07700 ;
                    07710 ;       INDENT (I) - Accept Numeric Input only
                    07720 ;
277F 96             07730         DB      NUM!ABB!6
2780 49             07740         DB      'INDENT'
     4E 44 45 4E 54
2786 ØØ             07750 IRESP   DB      Ø
2787 CD27           07760         DW      IPARM
                    07770 ;
                    07780 ;       LINES (L) - Accept Numeric Input only
                    07790 ;
2789 95             07800         DB      NUM!ABB!5
278A 4C             07810         DB      'LINES'
     49 4E 45 53
278F ØØ             07820 LRESP   DB      Ø
2790 CF27           07830         DW      LPARM
                    07840 ;
                    07850 ;       MARGIN (M) - Accept Numeric Input only
                    07860 ;
2792 96             07870         DB      NUM!ABB!6
2793 4D             07880         DB      'MARGIN'
     41 52 47 49 4E
2799 ØØ             07890 MRESP   DB      Ø
279A D127           07900         DW      MPARM
                    07910 ;
                    07920 ;       PAGE (P) - Accept Numeric Input only
                    07930 ;
279C 94             07940         DB      NUM!ABB!4
279D 50             07950         DB      'PAGE'
     41 47 45
27A1 ØØ             07960 PRESP   DB      Ø
27A2 D327           07970         DW      PPARM
                    07980 ;
                    07990 ;       QUERY (Q) - Accept Flag Input Only
                    08000 ;
27A4 55             08010         DB      FLAG!ABB!5
27A5 51             08020         DB      'QUERY'
     55 45 52 59
27AA ØØ             08030 QRESP   DB      Ø
27AB C927           08040         DW      QPARM
                    08050 ;
                    08060 ;       TAB (T) - Accept Flag input only
                    08070 ;
27AD 53             08080         DB      FLAG!ABB!3
27AE 54             08090         DB      'TAB'
     41 42
27B1 ØØ             08100 TRESP   DB      Ø
27B2 DB27           08110         DW      TPARM
                    08120 ;
                    08130 ;       XLATE (X) - Accept Numeric input only
                    08140 ;
27B4 95             08150         DB      NUM!ABB!5
27B5 58             08160         DB      'XLATE'
```

```
     4C 41 54 45
27BA ØØ        Ø817Ø XTRESP  DB      Ø
27BB D527      Ø818Ø         DW      XTPARM
               Ø819Ø ;
               Ø82ØØ ;       DEFAULT (D) - Accept Flag input only
               Ø821Ø ;
27BD 57        Ø822Ø         DB      FLAG!ABB!7
27BE 44        Ø823Ø         DB      'DEFAULT'
     45 46 41 55 4C 54
27C5 ØØ        Ø824Ø DRESP   DB      Ø
27C6 DD27      Ø825Ø         DW      DPARM
               Ø826Ø ;
27C8 ØØ        Ø827Ø         DB      Ø
               Ø828Ø ;
27C9 ØØØØ      Ø829Ø QPARM   DW      Ø
27CB ØØØØ      Ø83ØØ CPARM   DW      Ø
27CD ØØØØ      Ø831Ø IPARM   DW      Ø
27CF ØØØØ      Ø832Ø LPARM   DW      Ø
27D1 ØØØØ      Ø833Ø MPARM   DW      Ø
27D3 ØØØØ      Ø834Ø PPARM   DW      Ø
27D5 ØØØØ      Ø835Ø XTPARM  DW      Ø
               Ø836Ø ;
27D7 ØØØØ      Ø837Ø APARM   DW      Ø
27D9 ØØØØ      Ø838Ø FPARM   DW      Ø
27DB ØØØØ      Ø839Ø TPARM   DW      Ø
27DD ØØØØ      Ø84ØØ DPARM   DW      Ø
               Ø841Ø ;
27DF ØØ        Ø842Ø XFRESP  DB      Ø
27EØ E227      Ø843Ø         DW      XFPARM
27E2 ØØØØ      Ø844Ø XFPARM  DW      Ø
               Ø845Ø ;
               Ø846Ø ;
               Ø847Ø ;       Response Table - Response Addr, $FF Offset
               Ø848Ø ;
               Ø849Ø ;
               Ø85ØØ ;       8-bit Numeric Responses
               Ø851Ø ;
27E4 7227      Ø852Ø RESPTAB DW      CRESP
27E6 Ø8        Ø853Ø         DB      CHARS
               Ø854Ø ;
27E7 8627      Ø855Ø         DW      IRESP
27E9 Ø6        Ø856Ø         DB      INDENT
               Ø857Ø ;
27EA 8F27      Ø858Ø         DW      LRESP
27EC Ø2        Ø859Ø         DB      LINES
               Ø86ØØ ;
27ED 9927      Ø861Ø         DW      MRESP
27EF Ø9        Ø862Ø         DB      MARGIN
               Ø863Ø ;
27FØ A127      Ø864Ø         DW      PRESP
27F2 ØØ        Ø865Ø         DB      PAGE
               Ø866Ø ;
27F3 BA27      Ø867Ø         DW      XTRESP
27F5 Ø5        Ø868Ø         DB      XLATET
               Ø869Ø ;
27F6 DF27      Ø87ØØ         DW      XFRESP
27F8 Ø4        Ø871Ø         DB      XLATEF
               Ø872Ø ;
               Ø873Ø ;       Flag Response Table
               Ø874Ø ;
27F9 B127      Ø875Ø         DW      TRESP
```

```
27FB 7C27      Ø876Ø          DW      FRESP
27FD 6927      Ø877Ø          DW      ARESP
               Ø878Ø ;
               Ø879Ø ;
27FF 28        Ø88ØØ FAKEPRM  DB      '(F='
     46 3D
ØØØC           Ø881Ø INBUFF$  DS      12
               Ø882Ø ;
               Ø883Ø ;
               Ø884Ø ;        STRTAB - 1Ø entries each with 9 bytes:
               Ø885Ø ;
               Ø886Ø ;        1 byte : Type of expected response - flag or numeric
               Ø887Ø ;        2 bytes: Address of response byte
               Ø888Ø ;        2 bytes: Address of prompt string
               Ø889Ø ;        2 bytes: Address of routine to range check response
               Ø89ØØ ;        2 bytes: Address of default value string
               Ø891Ø ;
               Ø892Ø ;
               Ø893Ø ;
28ØE           Ø894Ø STRTAB   EQU     $
28ØE 8Ø        Ø895Ø          DB      NUM             ;PAGE
28ØF A127      Ø896Ø          DW      PRESP,PPROMPT,RPAGE,SPAGE
     A229 4526 A028
2817 8Ø        Ø897Ø          DB      NUM             ;LINES
2818 8F27      Ø898Ø          DW      LRESP,LPROMPT,RLINES,SLINES
     7A29 6A26 AD28
282Ø 8Ø        Ø899Ø          DB      NUM             ;CHARS
2821 7227      Ø9ØØØ          DW      CRESP,CPROMPT,RCHARS,SCHARS
     3229 7426 BA28
2829 8Ø        Ø9Ø1Ø          DB      NUM             ;MARGIN
282A 9927      Ø9Ø2Ø          DW      MRESP,MPROMPT,RMARGIN,SMARGIN
     9229 A426 C728
2832 8Ø        Ø9Ø3Ø          DB      NUM             ;INDENT
2833 8627      Ø9Ø4Ø          DW      IRESP,IPROMPT,RINDENT,SINDENT
     6Ø29 BE26 D428
283B 4Ø        Ø9Ø5Ø          DB      FLAG            ;ADDLF
283C 6927      Ø9Ø6Ø          DW      ARESP,APROMPT,FLAG?,SADDLF
     1929 E326 E128
2844 4Ø        Ø9Ø7Ø          DB      FLAG            ;FFHARD
2845 7C27      Ø9Ø8Ø          DW      FRESP,FPROMPT,FLAG?,SFFHARD
     4F29 E326 EE28
284D 4Ø        Ø9Ø9Ø          DB      FLAG            ;TAB
284E B127      Ø91ØØ          DW      TRESP,TPROMPT,FLAG?,STAB
     B829 E326 FB28
2856 8Ø        Ø911Ø          DB      NUM             ;XLATE From
2857 DF27      Ø912Ø          DW      XFRESP,XPROMF,NUMERIC,SXLFROM-2
     C729 DA26 Ø929
285F 8Ø        Ø913Ø          DB      NUM             ;XLATE To
2860 BA27      Ø914Ø          DW      XTRESP,XPROMT,NUMERIC,SXLTO-2
     D329 DA26 1229
               Ø915Ø ;
               Ø916Ø ;        Fake Parameter Table for prompts (QUERY)
               Ø917Ø ;
2868 8Ø        Ø918Ø FAKETAB  DB      8ØH             ;6.2 @ PARAM
2869 ØØ        Ø919Ø          DB      Ø               ;Type byte
286A 46        Ø92ØØ          DB      'F'
286B ØØ        Ø921Ø FAKERES  DB      Ø
286C B425      Ø922Ø          DW      VALUE+1         ;Destination
286E ØØ        Ø923Ø          DB      Ø
               Ø924Ø ;
               Ø925Ø ;
```

```
286F 24          Ø926Ø $FF      DB      '$FF',ETX
     46 46 Ø3
2873 2Ø          Ø927Ø ONSTR    DB      ' ON'
     4F 4E
2876 4F          Ø928Ø OFFSTR   DB      'OFF'
     46 46
                 Ø929Ø ;
2879 46          Ø93ØØ NOPF$    DB      'Forms Filter not Resident',CR
     6F 72 6D 73 2Ø 46 69 6C
     74 65 72 2Ø 6E 6F 74 2Ø
     52 65 73 69 64 65 6E 74
     ØD
                 Ø931Ø ;
2893 3F          Ø932Ø ENDPROM  DB      '? ',CURON,ETX
     2Ø ØE Ø3
2897 5Ø          Ø933Ø VALUES   DB      'PAGE   = '
     41 47 45 2Ø 2Ø 2Ø 3D 2Ø
28AØ 2Ø          Ø934Ø SPAGE    DB      ' 66',LF,'LINES  = '
     36 36 ØA 4C 49 4E 45 53
     2Ø 2Ø 3D 2Ø
28AD 2Ø          Ø935Ø SLINES   DB      ' 66',LF,'CHARS  = '
     36 36 ØA 43 48 41 52 53
     2Ø 2Ø 3D 2Ø
28BA 4F          Ø936Ø SCHARS   DB      'OFF',LF,'MARGIN = '
     46 46 ØA 4D 41 52 47 49
     4E 2Ø 3D 2Ø
28C7 2Ø          Ø937Ø SMARGIN  DB      '  Ø',LF,'INDENT = '
     2Ø 3Ø ØA 49 4E 44 45 4E
     54 2Ø 3D 2Ø
28D4 2Ø          Ø938Ø SINDENT  DB      '  Ø',LF,'ADDLF  = '
     2Ø 3Ø ØA 41 44 44 4C 46
     2Ø 2Ø 3D 2Ø
28E1 4F          Ø939Ø SADDLF   DB      'OFF',LF,'FFHARD = '
     46 46 ØA 46 46 48 41 52
     44 2Ø 3D 2Ø
28EE 4F          Ø94ØØ SFFHARD  DB      'OFF',LF,'TAB    = '
     46 46 ØA 54 41 42 2Ø 2Ø
     2Ø 2Ø 3D 2Ø
28FB 4F          Ø941Ø STAB     DB      'OFF',LF
     46 46 ØA
28FF ØD          Ø942Ø DOXLATE  DB      CR,'XLATE  = X',AP
     58 4C 41 54 45 2Ø 2Ø 3D
     2Ø 58 27
29ØB 3Ø          Ø943Ø SXLFROM  DB      'ØØ',AP,' => X',AP
     3Ø 27 2Ø 3D 3E 2Ø 58 27
2914 3Ø          Ø944Ø SXLTO    DB      'ØØ',AP,LF,CR
     3Ø 27 ØA ØD
                 Ø945Ø ;
                 Ø946Ø ;
2919 41          Ø947Ø APROMPT  DB      'Add Line Feed after C/R {'
     64 64 2Ø 4C 69 6E 65 2Ø
     46 65 65 64 2Ø 61 66 74
     65 72 2Ø 43 2F 52 2Ø 7B
2932 4D          Ø948Ø CPROMPT  DB      'Maximum Characters per Line {'
     61 78 69 6D 75 6D 2Ø 43
     68 61 72 61 63 74 65 72
     73 2Ø 7Ø 65 72 2Ø 4C 69
     6E 65 2Ø 7B
294F 52          Ø949Ø FPROMPT  DB      'Real Form Feeds {'
     65 61 6C 2Ø 46 6F 72 6D
     2Ø 46 65 65 64 73 2Ø 7B
```

```
296Ø 49          Ø95ØØ IPROMPT DB      'Indent after Wrap-around {'
     6E 64 65 6E 74 2Ø 61 66
     74 65 72 2Ø 57 72 61 7Ø
     2D 61 72 6F 75 6E 64 2Ø
     7B
297A 4C          Ø951Ø LPROMPT DB      'Lines Printed per Page {'
     69 6E 65 73 2Ø 5Ø 72 69
     6E 74 65 64 2Ø 7Ø 65 72
     2Ø 5Ø 61 67 65 2Ø 7B
2992 4D          Ø952Ø MPROMPT DB      'Margin Setting {'
     61 72 67 69 6E 2Ø 53 65
     74 74 69 6E 67 2Ø 7B
29A2 5Ø          Ø953Ø PPROMPT DB      'Physical Page Length {'
     68 79 73 69 63 61 6C 2Ø
     5Ø 61 67 65 2Ø 4C 65 6E
     67 74 68 2Ø 7B
29B8 54          Ø954Ø TPROMPT DB      'Tab Expansion {'
     61 62 2Ø 45 78 7Ø 61 6E
     73 69 6F 6E 2Ø 7B
29C7 58          Ø955Ø XPROMF  DB      'Xlate From {'
     6C 61 74 65 2Ø 46 72 6F
     6D 2Ø 7B
29D3 58          Ø956Ø XPROMT  DB      'Xlate To {'
     6C 61 74 65 2Ø 54 6F 2Ø
     7B
                 Ø957Ø ;
                 Ø958Ø ;
29DD 2Ø          Ø959Ø         DB      ' '
ØØØ5             Ø96ØØ TEMBUF  DS      5
ØØØA             Ø961Ø DUPDA   DS      1Ø
                 Ø962Ø ;
24ØØ             Ø963Ø         END     START
```

$FF	286F	@@1	ØØØØ	@@2	ØØØØ
@@3	ØØØØ	@@4	ØØØØ	@MOD2	ØØØØ
@MOD4	FFFF	ABB	ØØ1Ø	ABORT	241F
ADDLF	ØØØ7	AP	ØØ27	APARM	27D7
APROMPT	2919	ARESP	2769	BREAK	ØØ8Ø
BS	ØØØ8	CALLINS	251B	CFLAG$	ØØØ2
CHAROFF	249F	CHARS	ØØØ8	CHECKQ	2473
CHNGIND	269C	CKCOMM	24FC	CKCOMML	25Ø2
CPARM	27CB	CPROMPT	2932	CR	ØØØD
CRESP	2772	CUROFF	ØØØF	CURON	ØØØE
DATAREA	25CF	DEFTAB	2758	DFLAG$	ØØØ3
DISPA	259C	DISPFRM	2437	DISPROM	2558
DOFFHRD	24A5	DOINIT	27ØØ	DOPRMPT	253D
DOXLATE	28FF	DPARM	27DD	DRESP	27C5
DSFORMS	2489	DSP	26F1	DSPLY	26F7
DUNLP	259A	DUPDA	29E3	ENDPROM	2893
ETX	ØØØ3	EXDSP	26FB	EXIT	24Ø7
FAKEPRM	27FF	FAKERES	286B	FAKETAB	2868
FFHARD	ØØØ7	FLAG	ØØ4Ø	FLAG?	26E3
FLAGBT	ØØØ7	FLOOP	25F2	FORMS	2429
FPARM	27D9	FPROMPT	294F	FRESP	277C
GETFLAG	25FØ	GETPRM	243E	HEX8	273C
HEXDEC	2721	IGSPCS	242D	INBUFF$	28Ø2
INDENT	ØØØ6	INITVAL	2619	INPUT	2743
IOERR	24ØE	IPARM	27CD	IPROMPT	296Ø
IRESP	2786	IXINST	25EB	IXINST2	2612
KFLAG$	ØØØA	LDEF	ØØ42	LF	ØØØA
LINES	ØØØ2	LPARM	27CF	LPROMPT	297A
LRESP	278F	MARGIN	ØØØ9	MOREØ?	26D1
MPARM	27D1	MPROMPT	2992	MRESP	2799
NEXTFLG	2616	NEXTPR	255Ø	NOPARM	25EE
NOPF	2419	NOPF$	2879	NOPOUT	25E6
NOSHOW	24F8	NUM	ØØ8Ø	NUMERIC	26DA
OFFSTR	2876	ONSTR	2873	PAGE	ØØØØ
PAR_ERR	ØØ2C	PDEF	ØØ42	PNLP	258A
PNLP2	2597	PPARM	27D3	PPROMPT	29A2
PQUERY	2658	PRESP	27A1	PRLP	2561
PRMENT	2642	PRMERR	24ØC	PRMTBL$	2762
PROMPT	252A	PROMPTL	253Ø	QPARM	27C9
QRESP	27AA	RCHARS	2674	REINPUT	2537
RESPTAB	27E4	RETADR	25C7	RINDENT	26BE
RLINES	266A	RMARGIN	26A4	RPAGE	2645
SADDLF	28E1	SAVESP	2422	SCHARS	28BA
SDLP	261E	SETNZ	26CE	SETZ	2668
SFFHARD	28EE	SFLAG$	ØØ12	SINDENT	28D4
SKIP	ØØDD	SKIPSET	26Ø9	SLINES	28AD
SMARGIN	28C7	SPAGE	28AØ	SPLP	2574
STAB	28FB	START	24ØØ	STFDEF	264Ø
STFPRMS	25CC	STR	ØØ2Ø	STRTAB	28ØE
STUFDEF	2582	STUFFIN	2483	STUFLP	25D5
STUFVAL	25A3	SXLFROM	29ØB	SXLTO	2914
TAB	ØØØ9	TABV	ØØØ7	TEMBUF	29DE
TPARM	27DB	TPROMPT	29B8	TRESP	27B1
VALID1?	26C9	VALID2?	26CA	VALID?	26C6
VALUE	25B3	VALUES	2897	VFLAG$	ØØ15
XFER	26EB	XFERON	26E8	XFPARM	27E2
XFRESP	27DF	XLATEF	ØØØ4	XLATET	ØØØ5
XPROMF	29C7	XPROMT	29D3	XTPARM	27D5
XTRESP	27BA	@@ABORT	BD3C	@@ADTSK	BDCF
@@BANK	C2E7	@@BKSP	BFC7	@@BREAK	C2FD

```
@@CHNIO    BD27 @@CKBRKC   C34B @@CKDRV    BE23
@@CKEOF    BFDC @@CKTSK    BDBA @@CLOSE    BFB2
@@CLS      C335 @@CMNDI    BD66 @@CMNDR    BD7B
@@CTL      BB8B @@DATE     BCFD @@DCSTAT   BE62
@@DEBUG    BDA5 @@DECHEX   C267 @@DIRRD    C1D4
@@DIRWR    C1E9 @@DIV16    C252 @@DIV8     C23D
@@DODIR    BE38 @@DSP      BB4F @@DSPLY    BBEF
@@ERROR    BD9Ø @@EXIT     BD51 @@FEXT     C141
@@FLAGS    C2D1 @@FNAME    C156 @@FSPEC    C12C
@@GATRD    C1BF @@GATWR    C1FE @@GET      BB63
@@GTDCB    C18Ø @@GTDCT    C16B @@GTMOD    C195
@@HDFMT    BFØA @@HEX16    C2A6 @@HEX8     C291
@@HEXDEC   C27C @@HIGH$    C2BB @@INIT     BF88
@@KBD      BBC7 @@KEY      BB3B @@KEYIN    BBDB
@@KLTSK    BEØE @@LOAD     C1Ø2 @@LOC      BFF1
@@LOF      CØØ6 @@LOGER    BC26 @@LOGOT    BC3B
@@MSG      BC72 @@MUL16    C228 @@MUL8     C213
@@OPEN     BF9D @@PARAM    BCE8 @@PAUSE    BCD3
@@PEOF     CØ1B @@POSN     CØ3Ø @@PRINT    BC87
@@PRT      BB9F @@PUT      BB77 @@RAMDIR   BE4D
@@RDSEC    BEEØ @@RDSSC    C1AA @@READ     CØ45
@@REMOV    BF73 @@RENAM    BF5E @@REW      CØ5A
@@RMTSK    BDE4 @@RPTSK    BDF9 @@RREAD    CØ6F
@@RSLCT    BECB @@RSTOR    BE8C @@RUN      C117
@@RWRIT    CØ84 @@SEEK     BEB6 @@SEEKSC   CØ99
@@SKIP     CØAE @@SLCT     BE77 @@STEPI    BEA1
@@TIME     BD12 @@VDCTL    BCBE @@VER      CØC3
@@VRSEC    BEF5 @@WEOF     CØD8 @@WHERE    BBB3
@@WRITE    CØED @@WRSEC    BF1F @@WRSSC    BF34
@@WRTRK    BF49
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: FREE

Library: SYS7/SYS

ISAM # : 22H

```
                00100 ;LBFREE/ASM - FREE Command
0000            00110         TITLE   <FREE - LS-DOS 6.2>
                00120 ;
0008            00130 TPL     EQU     8               ;Tracks per Line = 8
0002            00140 @DSP    EQU     2               ;@DSP SVC #
0006            00150 @PRT    EQU     6               ;@PRT SVC #
000A            00160 @DSPLY  EQU     10              ;@DSPLY SVC #
000E            00170 @PRINT  EQU     14              ;@PRINT SVC #
0001            00180 @KEY    EQU     1               ;@KEY SVC #
                00190 ;
0000            00200 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
0000            00210 *GET    VALUES:3                ;Misc. equates
                03920 ;VALUES/ASM - Version 6
                03930 *LIST OFF
                04200 *LIST ON
                00220 ;
2400            00230         ORG     2400H
                00240 ;
2400            00250 FREEMAP EQU     $
                00260 ;
2400            00270         @@CKBRKC                ;See if break down
2400 3E6A       00001         LD      A,106
2402 EF         00002         RST     40
2403 2804       00280         JR      Z,BEGINA        ;Ok if not,
2405 21FFFF     00290         LD      HL,-1           ;  else abort
2408 C9         00300         RET
                00310 ;
                00320 ;       <BREAK> not hit - execute module
                00330 ;
                00340 BEGINA
2409 ED731424   00350         LD      (SAVESP+1),SP   ;Save SP address for exit
240D CD2524     00360         CALL    FREE            ;Show Free Space
                00370 ;
                00380 ;       Finished - Clear out <BREAK> & return
                00390 ;
2410 210000     00400 EXIT    LD      HL,0            ;HL = 0 (normal exit)
2413 310000     00410 SAVESP  LD      SP,$-$          ;P/u old SP address
2416            00420         @@CKBRKC                ;Clear <BREAK>
2416 3E6A       00003         LD      A,106
2418 EF         00004         RST     40
2419 C9         00430         RET
                00440 ;
                00450 ;       Error Handler - Display message & Abort
                00460 ;
241A 6F         00470 IOERR   LD      L,A             ;Set HL = Error #
241B 2600       00480         LD      H,0
241D F6C0       00490         OR      0C0H            ;Short error - & RETurn
241F 4F         00500         LD      C,A             ;Stuff error # in C
2420            00510         @@ERROR                 ;Display error
2420 3E1A       00005         LD      A,26
2422 EF         00006         RST     40
2423 18EE       00520         JR      SAVESP          ;Exit
                00530 ;
                00540 ;       FREE - Display Free Disk Space
                00550 ;
                00560 FREE
2425            00570         @@FLAGS                 ;IY => System Flags
2425 3E65       00007         LD      A,101
```

```
2427 EF          00008          RST     40
                 00580 ;
                 00590 ;        Stuff Address of SFLAG$ into routine
                 00600 ;
2428 111200      00610          LD      DE,SFLAG$       ;DE => Offset to SFLAG$
242B FD19        00620          ADD     IY,DE           ;IY => SFLAG$
242D FD226D26    00630          LD      (SFLAG1+1),IY   ;Save for later test
                 00640 ;
                 00650 ;        Position to parameters or end of line
                 00660 ;
2431 E5          00670          PUSH    HL              ;Save command ptr
2432 7E          00680 SKPLP    LD      A,(HL)          ;P/u char
2433 FE28        00690          CP      '('             ;Parameter(s) ?
2435 2807        00700          JR      Z,GETPRMS       ;Yes - get parameters
2437 FE0D        00710          CP      CR              ;End of line ?
2439 2809        00720          JR      Z,GETHL         ;Recover command ptr
243B 23          00730          INC     HL              ;Bump ptr
243C 18F4        00740          JR      SKPLP           ;No - go til terminator
                 00750 ;
                 00760 ;        Process any parameters if entered
                 00770 ;
243E 116828      00780 GETPRMS  LD      DE,PRMTBL$      ;DE => Parameter table
2441             00790          @@PARAM                 ;@PARAM
2441 3E11        00009          LD      A,17
2443 EF          00010          RST     40
2444 E1          00800 GETHL    POP     HL              ;Recover cmdline ptr
2445 C21A24      00810          JP      NZ,IOERR        ;NZ - parameter error
                 00820 ;
                 00830 ;        Anything after FREE command entered ?
                 00840 ;
2448 7E          00850          LD      A,(HL)          ;P/u first character
2449 FE29        00860          CP      '('+1           ;End of line ?
244B 3813        00870          JR      C,FREE0         ;Display lines
                 00880 ;
                 00890 ;        P/u next character if character is a colon
                 00900 ;
244D FE3A        00910          CP      ':'             ;Drivspec ?
244F 2002        00920          JR      NZ,CKIFDRV      ;No - check if numeric
2451 23          00930          INC     HL              ;Yes - p/u next char
2452 7E          00940          LD      A,(HL)
                 00950 ;
                 00960 ;        Convert drive # to binary (if legal) & save
                 00970 ;
2453 D630        00980 CKIFDRV  SUB     '0'             ;Legal drive Number ?
2455 3804        00990          JR      C,ILDRNUM       ;No - illegal drive #
2457 FE08        01000          CP      7+1
2459 383B        01010          JR      C,MAP           ;Less than 8 - good
                 01020 ;
                 01030 ;        Illegal Drive Number - display & Abort
                 01040 ;
245B 3E20        01050 ILDRNUM  LD      A,32            ;"Illegal Drive Number"
245D C31A24      01060          JP      IOERR           ;Display & Abort
                 01070 ;
                 01080 ;        Output a C/R to *PR if output is to *PR
                 01090 ;
2460 2A4C26      01100 FREE0    LD      HL,(PPARM+1)    ;P/u P parm
2463 2C          01110          INC     L               ;Specified ?
2464 200A        01120          JR      NZ,FREE0A       ;No - don't PRINT
2466 0E20        01130          LD      C,' '           ;Output space
2468 CD8B27      01140          CALL    PRT             ;  to printer
246B 0E0D        01150          LD      C,CR            ;Output C/R
```

```
246D CD8B27   01160          CALL    PRT             ;  to printer
2470 0E00     01170 FREE0A   LD      C,0             ;Init drive # to 0
              01180 ;
              01190 ;        Is there a disk in the drive ?
              01200 ;
2472 C5       01210 FREE1    PUSH    BC              ;Save drive #
2473          01220          @@GTDCT                 ;IY => DCT
2473 3E51     00011          LD      A,81
2475 EF       00012          RST     40
2476 FD7E00   01230          LD      A,(IY)          ;Drive on line ?
2479 FEC3     01240          CP      0C3H
247B 2012     01250          JR      NZ,NXTDRV       ;No - get next drive
247D          01260          @@CKDRV                 ;Disk in drive ?
247D 3E21     00013          LD      A,33
247F EF       00014          RST     40
              01270 ;
              01280 ;        Check if the <BREAK> was hit
              01290 ;
2480 F5       01300          PUSH    AF              ;Save @CKDRV condition
2481 CD9226   01310          CALL    CKBREAK         ;Check <BREAK>
2484 F1       01320          POP     AF              ;Recover @CKDRV cond
              01330 ;
              01340 ;        Display <No Disk> if @CKDRV fails
              01350 ;
2485 2805     01360          JR      Z,DOINF         ;Disk in - use header
2487 CD5426   01370          CALL    NO_DISK         ;No - display <No Disk>
248A 1803     01380          JR      NXTDRV          ;Get next drive
              01390 ;
              01400 ;        Create Header String & display if successful
              01410 ;
248C CD1425   01420 DOINF    CALL    GETINFO         ;Display Header string
              01430 ;
              01440 ;        Get next drive number
              01450 ;
248F C1       01460 NXTDRV   POP     BC              ;C = Drive #
2490 0C       01470          INC     C               ;Inc it
2491 CB59     01480          BIT     3,C             ;Finished ?
2493 28DD     01490          JR      Z,FREE1         ;No - get next drive
2495 C9       01500          RET                     ;Finished - RETurn
              01510 ;
              01520 ;        MAP - Display Free Space Map
              01530 ;
              01540 ;        Log in diskette if possible
              01550 ;
2496 4F       01560 MAP      LD      C,A             ;Xfer drive # to C
2497          01570          @@GTDCT                 ;IY => DCT + 0
2497 3E51     00015          LD      A,81
2499 EF       00016          RST     40
249A FD7E00   01580          LD      A,(IY)          ;P/u enable/disable
249D FEC3     01590          CP      0C3H            ;Drive enabled ?
249F C25B24   01600          JP      NZ,ILDRNUM      ;No - Illegal Drive #
24A2          01610          @@CKDRV                 ;Disk in drive ?
24A2 3E21     00017          LD      A,33
24A4 EF       00018          RST     40
24A5 2806     01620          JR      Z,DISKIN        ;Good - Disk in Drive
              01630 ;
              01640 ;        No Disk in Drive - Display message & Abort
              01650 ;
24A7 CD5426   01660          CALL    NO_DISK         ;Display <No Disk>
24AA C31024   01670          JP      EXIT            ;Go to exit routine
              01680 ;
```

```
                    01690 ;       Create header/footer strings & output header
                    01700 ;
24AD CD9226         01710 DISKIN  CALL    CKBREAK         ;Check for <BREAK>
24BØ CD1425         01720         CALL    GETINFO         ;Get GAT, create header
24B3 CD2527         01730         CALL    DISPUND         ;Display underline
24B6 CD8426         01740         CALL    CLRLN           ;Clear Line buffer
                    01750 ;
                    01760 ;       Transfer "  Ø-  7" string to line buffer
                    01770 ;
24B9 21A427         01780         LD      HL,MTRK         ;Initial track #s
24BC 11ØØ2B         01790         LD      DE,LINBUF
24BF Ø1Ø7ØØ         01800         LD      BC,7            ;Len track # display
24C2 EDBØ           01810         LDIR
                    01820 ;
                    01830 ;       Pt HL => GAT+Ø, C = cylinder -1 (gets INCed)
                    01840 ;
24C4 21ØØ29         01850         LD      HL,GAT          ;Pt to stored GAT
24C7 ØD             01860         DEC     C               ;Init Cyl = -1
                    01870 ;
                    01880 ;       Loop to Display each line of Cylinders
                    01890 ;
24C8 Ø6Ø8           01900 NEXTLIN LD      B,TPL           ;Max track per line count
24CA DD21Ø82B       01910         LD      IX,LINBUF+8     ;Pt to display buffer
                    01920 ;
                    01930 ;       Bump cylinder number & display Gran info
                    01940 ;
24CE ØC             01950 DSPSC   INC     C               ;Current cylinder
24CF CDE326         01960         CALL    DFRE            ;Display Free grans
24D2 23             01970         INC     HL              ;Pt to next track
                    01980 ;
                    01990 ;       Finished Displaying all the cylinders ?
                    02000 ;
24D3 FD7EØ6         02010         LD      A,(IY+6)        ;P/u max cylinder
24D6 B9             02020         CP      C               ;Finished ?
24D7 281D           02030         JR      Z,ENDRET        ;Yes - display footer
                    02040 ;
                    02050 ;       Calculate offset (9-Grans/cyl) to next track
                    02060 ;
24D9 3AE526         02070         LD      A,(GRANS+1)     ;P/u Grans/Cyl
24DC ED44           02080         NEG
24DE C6Ø9           02090         ADD     A,TPL+1         ;A = offset to next
                    02100 ;
                    02110 ;       Add offset to Line buffer pointer (IX)
                    02120 ;
24EØ 16ØØ           02130         LD      D,Ø             ;Stuff in DE
24E2 5F             02140         LD      E,A
24E3 DD19           02150         ADD     IX,DE           ;Where to dsp next track
24E5 1ØE7           02160         DJNZ    DSPSC           ;Loop current 6 trks
                    02170 ;
                    02180 ;       Finished 8 cylinders - display line
                    02190 ;
24E7 E5             02200         PUSH    HL              ;Save buffer loc'n
24E8 C5             02210         PUSH    BC              ;Save current cyl
24E9 CD4226         02220         CALL    DSPLINE         ;Display current line
                    02230 ;
                    02240 ;       Clear granule display line buffer
                    02250 ;
24EC CD8426         02260         CALL    CLRLN           ;Clear Line buffer
24EF C1             02270         POP     BC              ;Recover cylinder # in C
                    02280 ;
                    02290 ;       Change cylinder numbers in line buffer
```

```
                    02300 ;
24F0 CD2D27         02310           CALL    DSPTRK          ;Calc new track #'s
24F3 E1             02320           POP     HL              ;Restore GAT pointer
24F4 18D2           02330           JR      NEXTLIN         ;Get next line
                    02340 ;
                    02350 ;         Finished with drive - Display current line
                    02360 ;
24F6 CD4226         02370 ENDRET    CALL    DSPLINE         ;Display tracks in buffer
24F9 CD2527         02380           CALL    DISPUND         ;Display underline
                    02390 ;
                    02400 ;         Display Footer Message
                    02410 ;
24FC 21FB27         02420           LD      HL,FOOTER       ;HL => Footer Message
24FF CD4526         02430           CALL    DSPMSG          ;Display footer string
                    02440 ;
                    02450 ;         If footer will cause a scroll - wait for key
                    02460 ;
2502 3A6126         02470           LD      A,(CKPAGE+1)    ;P/u # of lines left
2505 FE02           02480           CP      2               ;At least 2 lines left?
2507 3003           02490           JR      NC,FRET         ;Lprint, free to return
2509 CD7F27         02500           CALL    KEY             ;Wait for char
250C 360D           02510 FRET      LD      (HL),CR         ;Scroll
250E CD4526         02520           CALL    DSPMSG          ;Display line
2511 C31024         02530           JP      EXIT            ;Go to normal exit
                    02540 ;
                    02550 ;         Stuff Drive Number into String Header
                    02560 ;
2514 79             02570 GETINFO   LD      A,C             ;P/u drive #
2515 C630           02580           ADD     A,'0'           ;Convert to ASCII
2517 32B227         02590           LD      (HDRIVE),A      ;Stuff into header string
                    02600 ;
                    02610 ;         Read in the diskette's GAT
                    02620 ;
251A                02630           @@GTDCT                 ;IY => DCT+0
251A 3E51           00019           LD      A,81
251C EF             00020           RST     40
251D FD5609         02640           LD      D,(IY+9)        ;P/u Directory cylinder
2520 1E00           02650           LD      E,0             ;Sector Zero
2522 210029         02660           LD      HL,GAT          ;HL => GAT I/O buffer
2525                02670           @@RDSSC                 ;Read System Sector
2525 3E55           00021           LD      A,85
2527 EF             00022           RST     40
2528 3E14           02680           LD      A,14H           ;Init to "GAT Read Error"
252A C21A24         02690 DERR      JP      NZ,IOERR        ;Jump on GAT read error
252D CD9226         02700           CALL    CKBREAK         ;Check for <BREAK>
                    02710 ;
                    02720 ;         Read in the diskette's HIT
                    02730 ;
2530 1C             02740           INC     E               ;Sector 1
2531 24             02750           INC     H               ;HL => HIT I/O buffer
2532                02760           @@RDSSC                 ;Read System Sector
2532 3E55           00023           LD      A,85
2534 EF             00024           RST     40
2535 3E16           02770           LD      A,16H           ;Init to "HIT Read Error"
2537 20F1           02780           JR      NZ,DERR         ;Go to Error handler
2539 CD9226         02790           CALL    CKBREAK         ;Check for <BREAK>
                    02800 ;
                    02810 ;         Pick up quantity of Sectors/Granule
                    02820 ;
253C FD7E08         02830           LD      A,(IY+8)        ;Bits 4-0 contain #
253F E61F           02840           AND     1FH             ;  of Sectors/Granule.
```

```
2541 3C          02850          INC     A               ;Adjust for zero offset
                 02860 ;
                 02870 ;        Convert Sectors/Gran to K & stuff in string
                 02880 ;
2542 F5          02890          PUSH    AF              ;Save Sectors/Granule
2543 210000      02900          LD      HL,0            ;Set HLA = # Sec/Gran
2546 114428      02910          LD      DE,FGRAN        ;DE => Destination
2549 019C26      02920          LD      BC,CVT2D        ;Only 2 digits possible
254C CD5C27      02930          CALL    CALCK2          ;Convert to K
254F F1          02940          POP     AF              ;A = Sectors/Granule
2550 5F          02950          LD      E,A             ;Xfer to E
                 02960 ;
                 02970 ;        Pick up number of cylinders in HL
                 02980 ;
2551 FD6E06      02990          LD      L,(IY+6)        ;P/u # of cylinders
2554 2600        03000          LD      H,0             ;Msb = 0
2556 23          03010          INC     HL              ;Relative to zero
                 03020 ;
                 03030 ;        Calculate quantity of Granules/Cylinder
                 03040 ;
2557 F5          03050          PUSH    AF              ;Save # of sectors/gran
2558 FD7E08      03060          LD      A,(IY+8)        ;Bits 7-5 contain
255B E6E0        03070          AND     0E0H            ;  # of Granules/cylinder.
255D 07          03080          RLCA                    ;& shift to bits 0-2
255E 07          03090          RLCA
255F 07          03100          RLCA
2560 3C          03110          INC     A               ;Adjust for zero offset
2561 FDCB046E    03120          BIT     5,(IY+4)        ;Double sided?
2565 2801        03130          JR      Z,FREE2         ;Bypass if one-sided
2567 87          03140          ADD     A,A             ;Else double the count
2568 32E526      03150 FREE2    LD      (GRANS+1),A     ;Save # Grans/Cyl
256B 4F          03160          LD      C,A             ;Save in C for @MULT8
                 03170 ;
                 03180 ;        Calculate quantity of Sectors/Cylinder
                 03190 ;
256C             03200          @@MUL8                  ;Mult E x C
256C 3E5A        00025          LD      A,90
256E EF          00026          RST     40
                 03210 ;
                 03220 ;        A = quantity of Sectors per cylinder
                 03230 ;
256F F5          03240          PUSH    AF              ;Save # Sectors/Cyl
2570 E5          03250          PUSH    HL              ;Save # Cylinders
                 03260 ;
                 03270 ;        File slots avail = 256 if more than 32 secs
                 03280 ;
2571 210001      03290          LD      HL,256          ;256 files maximum
2574 D602        03300          SUB     2               ;Set A = # secs in dir
2576 FE20        03310          CP      20H             ;  Greater than 32 ?
2578 3006        03320          JR      NC,FREE3        ;Yes - use default of 256
                 03330 ;
                 03340 ;        Calculate number of directory entries avail
                 03350 ;
257A 87          03360          ADD     A,A             ;Multiply # of Sectors
257B 87          03370          ADD     A,A             ;  in directory by 8
257C 87          03380          ADD     A,A             ;  to get # of slots.
257D 6F          03390          LD      L,A             ;Stuff in HL
257E 2600        03400          LD      H,0
2580 221D26      03410 FREE3    LD      (FREE7+1),HL    ;File slots to test later
                 03420 ;
                 03430 ;        Stuff # of entries (HL) into header string
```

```
                    03440 ;
2583 11F727         03450           LD      DE,HPOSSF       ;DE => Destination
2586 CDA026         03460           CALL    CVT3D           ;Cvt HL to ASCII @ DE
2589 E1             03470           POP     HL              ;Recover # of cylinders
258A F1             03480           POP     AF              ;Rcvr # of sectors/cyl
                    03490 ;
                    03500 ;         Calculate total # of sectors HL x A
                    03510 ;
258B 4F             03520           LD      C,A             ;Set C = Sec/cyl
258C B7             03530           OR      A
258D 2803           03540           JR      Z,SKIPMUL       ;Don't multiply if zero
258F                03550           @@MUL16                 ;Multiply HL x C
258F 3E5B           00027           LD      A,91
2591 EF             00028           RST     40
                    03560 ;
                    03570 ;         Convert # of sectors to K & stuff in string
                    03580 ;
2592 11E027         03590 SKIPMUL   LD      DE,HPOSSK       ;DE => Destination
2595 CD5927         03600           CALL    CALCK           ;Stuff into string
                    03610 ;
                    03620 ;         Transfer Diskette Name from GAT into string
                    03630 ;
2598 21D029         03640           LD      HL,GAT+0D0H     ;HL => Pack Name
259B 11B527         03650           LD      DE,HNAME        ;DE => String destination
259E 0E08           03660           LD      C,8             ;8 chars to xfer
25A0 EDB0           03670           LDIR                    ;Xfer into string
                    03680 ;
                    03690 ;         Transfer Diskette Date from GAT into string
                    03700 ;
25A2 11BF27         03710           LD      DE,HDATE        ;DE => Destination
25A5 0E08           03720           LD      C,8             ;8 chars to xfer
25A7 EDB0           03730           LDIR                    ;Xfer to string
                    03740 ;
                    03750 ;         Pt HL => GAT, DE = Free Gran cnt, B = cyls
                    03760 ;
25A9 210029         03770           LD      HL,GAT          ;Pt to start of GAT
25AC 110000         03780           LD      DE,0            ;Init gran counter
25AF 3ACC29         03790           LD      A,(GAT+0CCH)    ;P/u cyl excess
25B2 C623           03800           ADD     A,35            ;Add base
25B4 47             03810           LD      B,A             ;Set loop counter
                    03820 ;
                    03830 ;         Calculate quantity of Free granules left
                    03840 ;
25B5 7E             03850 FREE4     LD      A,(HL)          ;P/u GAT byte & set
25B6 37             03860 FREE5     SCF                     ;  carry so bit 7 stays 1
                    03870 ;
                    03880 ;         Is the granule in use ?
                    03890 ;
25B7 1F             03900           RRA                     ;Slide gran bit to carry
25B8 3801           03910           JR      C,FREE6         ;Ignore if in use
                    03920 ;
                    03930 ;         Free Granule - Bump Free Granule count
                    03940 ;
25BA 13             03950           INC     DE              ;Free, bump gran counter
25BB FEFF           03960 FREE6     CP      0FFH            ;End of byte?
25BD 20F7           03970           JR      NZ,FREE5        ;Loop if not
                    03980 ;
                    03990 ;         Finished with one cylinder, advance to next
                    04000 ;
25BF 2C             04010           INC     L               ;Bump GAT byte pointer
25C0 10F3           04020           DJNZ    FREE4           ;Loop for # cyls
```

```
                    04030 ;
                    04040 ;       Multiply # Grans (DE) by Sectors/Gran
                    04050 ;
25C2 EB             04060         EX      DE,HL           ;Xfer # Grans to HL
25C3 F1             04070         POP     AF              ;Rcvr # of sectors/gran
25C4 4F             04080         LD      C,A             ;Put in C for @MUL16
25C5                04090         @@MUL16                 ;Multiply HL x C
25C5 3E5B           00029         LD      A,91
25C7 EF             00030         RST     40
                    04100 ;
                    04110 ;       Cvt # of Free Grans to K & stuff in string
                    04120 ;
25C8 11D627         04130         LD      DE,HFREEK       ;Cvrt to decimal
25CB CD5927         04140         CALL    CALCK           ;Cvrt to ASCII & stuff
                    04150 ;
                    04160 ;       Build Footer String in case of map
                    04170 ;
25CE 3E35           04180         LD      A,'5'           ;Init 5"/8" media
25D0 FDCB036E       04190         BIT     5,(IY+3)        ;Test DCT for size
25D4 2802           04200         JR      Z,FIVEIN        ;Go if 5"
25D6 3E38           04210         LD      A,'8'           ;Else reset to 8"
25D8 320428         04220 FIVEIN  LD      (FSIZE),A       ;Stuff size into header
                    04230 ;
                    04240 ;       P/u # of heads from DCT & stuff into footer
                    04250 ;
25DB FD7E07         04260         LD      A,(IY+7)        ;Bits 7-5 = # heads
25DE 07             04270         RLCA                    ;Shift to 0-2
25DF 07             04280         RLCA
25E0 07             04290         RLCA
25E1 E607           04300         AND     7               ;Mask off other junk
25E3 3C             04310         INC     A               ;Relative to zero
25E4 F630           04320         OR      '0'             ;Make it ASCII
25E6 321928         04330         LD      (FHEADS),A      ;Stuff into header
                    04340 ;
                    04350 ;       If this is a hard drive - ignore sides check
                    04360 ;
25E9 FDCB035E       04370         BIT     3,(IY+3)        ;Check if hard
25ED 2810           04380         JR      Z,DOSIDES       ;Not hard - check sides
                    04390 ;
                    04400 ;       Hard Drive - overwrite Floppy in footer
                    04410 ;
25EF 219E27         04420         LD      HL,HARD         ;HL => "Hard  "
25F2 110728         04430         LD      DE,FTYPE        ;DE => Dest in footer
25F5 010600         04440         LD      BC,6            ;BC = 6 chars to xfer
25F8 EDB0           04450         LDIR                    ;Transfer to footer
25FA 219827         04460         LD      HL,RIGID        ;HL => "RIGID"
25FD 1815           04470         JR      D3              ;Xfer "RIGID" to footer
                    04480 ;
                    04490 ;       Floppy disk - Stuff # of sides into footer
                    04500 ;
25FF 3E31           04510 DOSIDES LD      A,'1'           ;Init # of sides
2601 FDCB046E       04520         BIT     5,(IY+4)        ;Test DCT for sides
2605 2801           04530         JR      Z,ONESIDE       ;Go if 1-sides
2607 3C             04540         INC     A               ;Else bump to 2
2608 321928         04550 ONESIDE LD      (FHEADS),A      ;Stuff into header
                    04560 ;
                    04570 ;       If floppy is double density pt HL to string
                    04580 ;
260B FDCB0376       04590         BIT     6,(IY+3)        ;Test SDEN/DDEN
260F 280B           04600         JR      Z,FREE7         ;Single - that's default
2611 219227         04610         LD      HL,MDDEN        ;Density MSG - Double
```

```
                    04620 ;
                    04630 ;       Xfer "Single, Double, or Rigid" to footer
                    04640 ;
2614 112728         04650 D3      LD      DE,FDENS        ;Density MSG dsp pos
2617 010600         04660         LD      BC,6            ;6 chars to xfer
261A EDB0           04670         LDIR                    ;Move Double to cover
                    04680 ;
                    04690 ;       Calculate # of Free HIT positions available
                    04700 ;
261C 110000         04710 FREE7   LD      DE,$-$          ;P/u # of poss entries
261F 21002A         04720         LD      HL,HIT          ;HL => HIT + 0
2622 1B             04730 FREE8   DEC     DE              ;Dec count in case of SYS
                    04740 ;
                    04750 ;       Check SYS slots if this is a data disk
                    04760 ;
2623 3ACD29         04770         LD      A,(GAT+0CDH)    ;Bit 7 set if Data disk
2626 07             04780         RLCA
2627 3805           04790         JR      C,DATDISK       ;Set - ignore SYS check
                    04800 ;
                    04810 ;       Is this a SYS slot - 00-07 or 20-27 ?
                    04820 ;
2629 7D             04830         LD      A,L             ;P/u HIT offset
262A E6D8           04840         AND     0D8H            ;Reserved slot ?
262C 2805           04850         JR      Z,FREE9         ;Yes - can't use it
                    04860 ;
                    04870 ;       Not reserved - is the HIT posn in use ?
                    04880 ;
262E 7E             04890 DATDISK LD      A,(HL)          ;File in use?
262F B7             04900         OR      A
2630 2001           04910         JR      NZ,FREE9        ;Yes - don't bump count
                    04920 ;
                    04930 ;       Slot not in use - bump free slot count
                    04940 ;
2632 13             04950         INC     DE              ;Bump free count
2633 2C             04960 FREE9   INC     L               ;Bump HIT pointer
2634 20EC           04970         JR      NZ,FREE8        ;Loop if not through
                    04980 ;
                    04990 ;       Stuff available files into string
                    05000 ;
2636 EB             05010         EX      DE,HL           ;Available files to HL
2637 11F327         05020         LD      DE,HFREEF       ;Cvrt to ASCII into msg
263A CDA026         05030         CALL    CVT3D           ;Convert & RETurn
                    05040 ;
                    05050 ;       Display Header String & RETurn
                    05060 ;
263D 21AB27         05070         LD      HL,HEADER       ;HL => Header string
2640 1803           05080         JR      DSPMSG          ;Display header & RETurn
                    05090 ;
2642 21002B         05100 DSPLINE LD      HL,LINBUF       ;Fall into Display & RET
                    05110 ;
                    05120 ;       DSPMSG - Display a message pointed to by HL
                    05130 ;
2645 CD9226         05140 DSPMSG  CALL    CKBREAK         ;Check for <BREAK>
2648 CD8527         05150         CALL    DSPLY           ;Display message to video
264B 110000         05160 PPARM   LD      DE,$-$          ;P/u P parm
264E 1C             05170         INC     E               ;Was it entered ?
264F 200F           05180         JR      NZ,CKPAGE       ;No - Check page pause
2651 C38827         05190         JP      PRINT           ;Output line to *PR
                    05200 ;
2654 79             05210 NO_DISK LD      A,C             ;P/u drive #
2655 C630           05220         ADD     A,'0'           ;Cvt to ASCII
```

```
2657 325228   05230          LD      (NODISKN),A     ;Stuff in string
265A 214B28   05240          LD      HL,NODISK       ;HL => Message
265D C34526   05250          JP      DSPMSG          ;Display Mess & RETurn
              05260 ;
              05270 ;        Decrement Lines printed count
              05280 ;
2660 3E16     05290 CKPAGE   LD      A,22            ;Ck for display pause
2662 3D       05300          DEC     A               ;Count down
2663 326126   05310          LD      (CKPAGE+1),A    ;Update
2666 C0       05320          RET     NZ              ;Ret if not yet full
              05330 ;
              05340 ;        Printed a full page - Reset to count to max
              05350 ;
2667 3E17     05360          LD      A,23            ;Max lines to print
2669 326126   05370          LD      (CKPAGE+1),A    ;Reset to max
              05380 ;
              05390 ;        Do not stop if a <DO> is in effect
              05400 ;
266C 3A0000   05410 SFLAG1   LD      A,($-$)         ;P/u SFLAG$
266F E620     05420          AND     20H             ;Do in effect ?
2671 C0       05430          RET     NZ              ;Yes - RETurn
              05440 ;
              05450 ;        Wait for key - then clear screen
              05460 ;
2672 CD7F27   05470          CALL    KEY             ;Wait for key entry
2675          05480 DISPHDR  @@CLS                   ;Clear Screen
2675 3E69     00031          LD      A,105
2677 EF       00032          RST     40
2678 C21A24   05490          JP      NZ,IOERR        ;Abort if I/O Error
              05500 ;
              05510 ;        Display Map header
              05520 ;
267B 21AB27   05530          LD      HL,HEADER       ;Point to the header
267E CD4526   05540          CALL    DSPMSG          ;  & display it
2681 CD2527   05550          CALL    DISPUND         ;Display underline
              05560 ;
              05570 ;        CLRLN - Clear line buffer
              05580 ;
2684 3E20     05590 CLRLN    LD      A,' '           ;Clear buffer
2686 21002B   05600 BUFSTUF  LD      HL,LINBUF       ;Point to buffer
2689 064F     05610          LD      B,79            ;Length of buffer
268B 77       05620 CLRLN1   LD      (HL),A          ;Stuff with char given
268C 23       05630          INC     HL
268D 10FC     05640          DJNZ    CLRLN1
268F 360D     05650          LD      (HL),CR         ;End line with C/R
2691 C9       05660          RET
              05670 ;
              05680 ;        CKBREAK - Check if the <BREAK> was pressed
              05690 ;
2692          05700 CKBREAK  EQU     $
2692          05710          @@CKBRKC                ;<BREAK> hit ?
2692 3E6A     00033          LD      A,106
2694 EF       00034          RST     40
2695 C8       05720          RET     Z               ;No - RETurn
2696 21FFFF   05730 ABORT    LD      HL,-1           ;<BREAK> hit - abort
2699 C31324   05740          JP      SAVESP
              05750 ;
              05760 ;
              05770 ;        CVTDEC - Convert Hex Number to Decimal ASCII
              05780 ;        CVD2D - CVD4D  - Convert to 2,3, or 4 digits
              05790 ;
```

```
                    05800 ;       HL => Hex Number to Convert
                    05810 ;       DE => Buffer to receive characters
                    05820 ;
269C 3E20           05830 CVT2D   LD      A,' '
269E 181C           05840         JR      CVT10
26A0 3E20           05850 CVT3D   LD      A,' '
26A2 1812           05860         JR      CVT100
26A4 3E20           05870 CVD4D   LD      A,' '
26A6 1808           05880         JR      CVT1000
26A8 3E20           05890 CVTDEC  LD      A,' '
                    05900 ;
26AA 011027         05910         LD      BC,10000
26AD CDC826         05920         CALL    CVD1
26B0 01E803         05930 CVT1000 LD      BC,1000
26B3 CDC826         05940         CALL    CVD1
26B6 016400         05950 CVT100  LD      BC,100
26B9 CDC826         05960         CALL    CVD1
26BC 010A00         05970 CVT10   LD      BC,10
26BF CDC826         05980         CALL    CVD1
26C2 7D             05990         LD      A,L
26C3 C630           06000         ADD     A,'0'
26C5 12             06010         LD      (DE),A
26C6 13             06020         INC     DE
26C7 C9             06030         RET
                    06040 ;
26C8 D5             06050 CVD1    PUSH    DE
26C9 5F             06060         LD      E,A
26CA 16FF           06070         LD      D,0FFH
26CC AF             06080         XOR     A
26CD 14             06090 CVD2    INC     D
26CE ED42           06100         SBC     HL,BC
26D0 30FB           06110         JR      NC,CVD2
26D2 09             06120         ADD     HL,BC
26D3 7B             06130         LD      A,E
26D4 42             06140         LD      B,D
26D5 D1             06150         POP     DE
26D6 12             06160         LD      (DE),A
26D7 04             06170         INC     B
26D8 05             06180         DEC     B
26D9 2806           06190         JR      Z,CVD3
26DB 78             06200         LD      A,B
26DC C630           06210         ADD     A,'0'
26DE 12             06220         LD      (DE),A
26DF 3E30           06230         LD      A,'0'
26E1 13             06240 CVD3    INC     DE
26E2 C9             06250         RET
                    06260 ;
                    06270 ;       DFRE - Stuff a cylinder's Gran symbols in buffer
                    06280 ;
                    06290 ;       IX => Buffer to receive characters
                    06300 ;       HL => GAT cylinder byte to use
                    06310 ;
26E3 C5             06320 DFRE    PUSH    BC              ;Save C, cur cyl loc
26E4 0600           06330 GRANS   LD      B,$-$           ;P/u Grans/Cylinder
                    06340 ;
                    06350 ;       Is this cylinder the directory ?
                    06360 ;
26E6 FD7E09         06370         LD      A,(IY+9)        ;P/u dir cyl from DCT
26E9 B9             06380         CP      C               ;Directory ?
26EA 282F           06390         JR      Z,DDIR          ;Yes - use "D"'s
                    06400 ;
```

```
                    06410 ;       Not the directory cyl - use "x", "." & "*"
                    06420 ;
26EC DDE5           06430         PUSH    IX              ;Save buffer pointer
26EE C5             06440         PUSH    BC              ;Save Grans/Cyl
                    06450 ;
                    06460 ;       Is the Granule in use ?
                    06470 ;
26EF CBØE           06480 DF1     RRC     (HL)            ;P/u next Granule
26F1 3E78           06490         LD      A,'x'           ;Init "in use"
26F3 38Ø2           06500         JR      C,DF2           ;Set - use "x"
                    06510 ;
                    06520 ;       Granule isn't in use - stuff a "." in buffer
                    06530 ;
26F5 3E2E           06540         LD      A,'.'           ;Else free
26F7 DD77ØØ         06550 DF2     LD      (IX+Ø),A        ;Stuff char
                    06560 ;
                    06570 ;       Bump buffer pointer & decrement G/C count
                    06580 ;
26FA DD23           06590         INC     IX              ;Next display loc
26FC 1ØF1           06600         DJNZ    DF1             ;Loop thru all grans
                    06610 ;
                    06620 ;       Recover Buff ptr, Grans/Cyl
                    06630 ;
26FE C1             06640         POP     BC              ;B = Grans per Cyl
26FF DDE1           06650         POP     IX              ;IX to start of track
                    06660 ;
                    06670 ;       Position HL to Lockout table
                    06680 ;
2701 E5             06690         PUSH    HL              ;Save Cyl ptr
2702 116ØØØ         06700         LD      DE,6ØH          ;Offset to lockout table
2705 19             06710         ADD     HL,DE           ;Point to lockout
                    06720 ;
                    06730 ;       Go through lockout & overwrite if locked out
                    06740 ;
2706 FDCBØ35E       06750 LO1     BIT     3,(IY+3)        ;If hard drive, there's
270A 2ØØ8           06760         JR      NZ,LO2          ;  no lockout
                    06770 ;
                    06780 ;       Diskette is a floppy - Is gran locked out ?
                    06790 ;
270C CBØE           06800         RRC     (HL)            ;Gran locked out ?
270E 3ØØ4           06810         JR      NC,LO2          ;No - bump buff ptr
2710 DD36ØØ2A       06820         LD      (IX+Ø),'*'      ;Asterisk = lockout
                    06830 ;
                    06840 ;       Bump buffer pointer & loop til done
                    06850 ;
2714 DD23           06860 LO2     INC     IX              ;Next gran dsp loc
2716 1ØEE           06870         DJNZ    LO1             ;B grans/cyl
                    06880 ;
                    06890 ;       Recover ptrs & RETurn
                    06900 ;
2718 E1             06910         POP     HL
2719 C1             06920         POP     BC
271A C9             06930         RET
                    06940 ;
                    06950 ;       DDIR - Use "D"'s for Directory instead of "x"
                    06960 ;
271B DD36ØØ44       06970 DDIR    LD      (IX+Ø),'D'      ;Stuff "D" char
271F DD23           06980         INC     IX              ;Loop thru all DIR grans
2721 1ØF8           06990         DJNZ    DDIR
2723 C1             07000         POP     BC
2724 C9             07010         RET
```

```
                        07020 ;
                        07030 ;       DISPUND - Display a line of "-"
                        07040 ;
2725 3E2D               07050 DISPUND LD      A,'-'           ;Character to underline
2727 CD8626             07060         CALL    BUFSTUF         ;Stuff line & display
272A C34226             07070         JP      DSPLINE         ;Display line
                        07080 ;
                        07090 ;       DSPTRK - Stuff cylinder numbers in line buffer
                        07100 ;
272D D5                 07110 DSPTRK  PUSH    DE              ;Save registers used
272E C5                 07120         PUSH    BC
                        07130 ;
                        07140 ;       Stuff starting cylinder # in line buffer
                        07150 ;
272F 11002B             07160         LD      DE,LINBUF       ;Display buffer
2732 0C                 07170         INC     C               ;Bump to next cylinder
2733 CD5027             07180         CALL    DSPTK4          ;Display cylinder number
                        07190 ;
                        07200 ;       Is this the only cylinder in the line ?
                        07210 ;
2736 FD7E06             07220         LD      A,(IY+6)        ;P/u maximum # cyls
2739 B9                 07230         CP      C               ;Are we at the top?
273A 2811               07240         JR      Z,DSPTK3        ;Go if yes
                        07250 ;
                        07260 ;       More than 1 cyl - stuff "-" in line buffer
                        07270 ;
273C F5                 07280         PUSH    AF              ;Save max cylinder
273D 3E2D               07290         LD      A,'-'           ;Stuff dash in buffer
273F 12                 07300         LD      (DE),A
2740 F1                 07310         POP     AF              ;Recover max cylinder
                        07320 ;
                        07330 ;       Get ending cylinder # for this line in C
                        07340 ;
2741 13                 07350         INC     DE              ;Position to next avail
2742 0607               07360         LD      B,TPL-1         ;Need 7 more on a line
                        07370 ;
2744 0C                 07380 DSPTK1  INC     C               ;Bump until 7 or max
2745 B9                 07390         CP      C
2746 2802               07400         JR      Z,DSPTK2
2748 10FA               07410         DJNZ    DSPTK1
                        07420 ;
                        07430 ;       Stuff ending cylinder # into line buffer
                        07440 ;
274A CD5027             07450 DSPTK2  CALL    DSPTK4          ;Stuff ending cyl on line
274D C1                 07460 DSPTK3  POP     BC              ;Recover registers
274E D1                 07470         POP     DE
274F C9                 07480         RET                     ;RETurn
                        07490 ;
                        07500 ;       Convert cylinder # (C) to ASCII at DE
                        07510 ;
2750 69                 07520 DSPTK4  LD      L,C             ;Xfer cylinder # to HL
2751 2600               07530         LD      H,0
2753 C5                 07540         PUSH    BC              ;Save cylinder #
2754 CDA026             07550         CALL    CVT3D           ;Convert cyl# to ASCII
2757 C1                 07560         POP     BC              ;C = cylinder #
2758 C9                 07570         RET                     ;RETurn
                        07580 ;
                        07590 ;       CALCK - Calculate Number of K & stuff in string
                        07600 ;       HLA => Total # of Sectors
                        07610 ;       DE => Destination of String
                        07620 ;
```

```
2759 01A826    07630 CALCK   LD      BC,CVTDEC       ;4 digit default
275C ED436C27  07640 CALCK2  LD      (CONVERT+1),BC
2760 65        07650 DIVHLA  LD      H,L             ;Per disk pack
2761 6F        07660         LD      L,A
2762 CB3C      07670         SRL     H               ;Divide total sectors
2764 CB1D      07680         RR      L               ;  by 4 to calculate
2766 CB3C      07690         SRL     H               ;  space in K
2768 CB1D      07700         RR      L
               07710 ;
               07720 ;       Convert K free (HL) to ASCII & put in string
               07730 ;
276A F5        07740         PUSH    AF              ;Save offset
276B CD0000    07750 CONVERT CALL    $-$             ;Cvt HL to ASCII @ DE
276E F1        07760         POP     AF              ;Recover offset
               07770 ;
               07780 ;       Stuff hundredths value into string
               07790 ;
276F 13        07800         INC     DE              ;Go past decimal point
2770 E603      07810         AND     3               ;Modulo 4
2772 87        07820         ADD     A,A             ;Multiply by 2
2773 0600      07830         LD      B,0             ;  to position to
2775 4F        07840         LD      C,A             ;  hundredths string
2776 216028    07850         LD      HL,HUNDTAB      ;HL => Table base
2779 09        07860         ADD     HL,BC           ;HL => Hundredths string
277A 0E02      07870         LD      C,2             ;2 chars to xfer
277C EDB0      07880         LDIR                    ;Stuff in string
277E C9        07890         RET                     ;RETurn
               07900 ;
               07910 ;       KEY/DSP/DSPLY/PRT/PRINT - SVC routines
               07920 ;
277F 3E01      07930 KEY     LD      A,@KEY          ;Wait for key
2781 11        07940         DB      11H
               07950 ;
2782 3E02      07960 DSP     LD      A,@DSP          ;Display byte
2784 11        07970         DB      11H             ;LD DE,nnnn
               07980 ;
2785 3E0A      07990 DSPLY   LD      A,@DSPLY        ;Display line
2787 11        08000         DB      11H
               08010 ;
2788 3E0E      08020 PRINT   LD      A,@PRINT        ;Print line
278A 11        08030         DB      11H
               08040 ;
278B 3E06      08050 PRT     LD      A,@PRT          ;Print byte
               08060 ;
278D EF        08070 DO_OUT  RST     40              ;Do SVC & check error
278E C8        08080         RET     Z               ;RETurn if good
278F C31A24    08090         JP      IOERR           ;NZ - I/O Error
               08100 ;
2792 44        08110 MDDEN   DB      'DOUBLE'
     4F 55 42 4C 45
2798 52        08120 RIGID   DB      'RIGID '
     49 47 49 44 20
279E 48        08130 HARD    DB      'Hard  '
     61 72 64 20 20
               08140 ;
27A4 20        08150 MTRK    DB      '  0- 7'
     20 30 2D 20 20 37
               08160 ;
27AB 44        08170 HEADER  DB      'Drive :'
     72 69 76 65 20 3A
27B2 64        08180 HDRIVE  DB      'd  '
```

```
     20 20
27B5 64          08190 HNAME   DB      'diskname  '
     69 73 6B 6E 61 6D 65 20
     20
27BF 64          08200 HDATE   DB      'dd/mm/yy   Free Space ='
     64 2F 6D 6D 2F 79 79 20
     20 20 46 72 65 65 20 53
     70 61 63 65 20 3D
27D6 6E          08210 HFREEK  DB      'nnnnn.nnK/'
     6E 6E 6E 6E 2E 6E 6E 4B
     2F
27E0 6E          08220 HPOSSK  DB      'nnnnn.nnK  Files = '
     6E 6E 6E 6E 2E 6E 6E 4B
     20 20 46 69 6C 65 73 20
     3D 20
27F3 64          08230 HFREEF  DB      'ddd/'
     64 64 2F
27F7 64          08240 HPOSSF  DB      'ddd',CR
     64 64 0D
                 08250 ;
27FB 54          08260 FOOTER  DB      'Type =>  '
     79 70 65 20 3D 3E 20 20
2804 73          08270 FSIZE   DB      's" '
     22 20
2807 46          08280 FTYPE   DB      'Floppy    Heads = '
     6C 6F 70 70 79 20 20 20
     20 48 65 61 64 73 20 3D
     20
2819 6E          08290 FHEADS  DB      'n   Density = '
     20 20 20 44 65 6E 73 69
     74 79 20 3D 20
2827 53          08300 FDENS   DB      'SINGLE   Note - 1 Position = '
     49 4E 47 4C 45 20 20 20
     4E 6F 74 65 20 2D 20 31
     20 50 6F 73 69 74 69 6F
     6E 20 3D 20
2844 6E          08310 FGRAN   DB      'nn.nnK'
     6E 2E 6E 6E 4B
284A 0D          08320 ENDFOOT DB      CR
                 08330 ;
284B 44          08340 NODISK  DB      'Drive :'
     72 69 76 65 20 3A
2852 64          08350 NODISKN DB      'd  [No  Disk]',CR
     20 20 5B 4E 6F 20 20 44
     69 73 6B 5D 0D
                 08360 ;
2860 30          08370 HUNDTAB DB      '00255075'
     30 32 35 35 30 37 35
                 08380 ;
                 08390 ;       Parameter Table
                 08400 ;
2868 80          08410 PRMTBL$ DB      80H
2869 41          08420         DB      FLAG!1
286A 50          08430         DB      'P'
286B 00          08440         DB      0
286C 4C26        08450         DW      PPARM+1
286E 00          08460         NOP
                 08470 ;
2900             08480         ORG     $<-8+1<+8
0100             08490 GAT     DS      256
0100             08500 HIT     DS      256
```

```
2BØØ            Ø851Ø LINBUF   EQU     $
                Ø852Ø ;
24ØØ            Ø853Ø          END     FREEMAP
```

@@1	ØØØØ	@@2	ØØØØ	@@3	ØØØØ	@@4	ØØØØ
@DSP	ØØØ2	@DSPLY	ØØØA	@KEY	ØØØ1	@MOD2	ØØØØ
@MOD4	FFFF	@PRINT	ØØØE	@PRT	ØØØ6	ABB	ØØ1Ø
ABORT	2696	AP	ØØ27	BEGINA	24Ø9	BREAK	ØØ8Ø
BS	ØØØ8	BUFSTUF	2686	CALCK	2759	CALCK2	275C
CFLAG$	ØØØ2	CKBREAK	2692	CKIFDRV	2453	CKPAGE	266Ø
CLRLN	2684	CLRLN1	268B	CONVERT	276B	CR	ØØØD
CVD1	26C8	CVD2	26CD	CVD3	26E1	CVD4D	26A4
CVT1Ø	26BC	CVT1ØØ	26B6	CVT1ØØØ	26BØ	CVT2D	269C
CVT3D	26AØ	CVTDEC	26A8	D3	2614	DATDISK	262E
DDIR	271B	DERR	252A	DF1	26EF	DF2	26F7
DFLAG$	ØØØ3	DFRE	26E3	DISKIN	24AD	DISPHDR	2675
DISPUND	2725	DIVHLA	276Ø	DOINF	248C	DOSIDES	25FF
DO_OUT	278D	DSP	2782	DSPLINE	2642	DSPLY	2785
DSPMSG	2645	DSPSC	24CE	DSPTK1	2744	DSPTK2	274A
DSPTK3	274D	DSPTK4	275Ø	DSPTRK	272D	ENDFOOT	284A
ENDRET	24F6	ETX	ØØØ3	EXIT	241Ø	FDENS	2827
FGRAN	2844	FHEADS	2819	FIVEIN	25D8	FLAG	ØØ4Ø
FOOTER	27FB	FREE	2425	FREEØ	246Ø	FREEØA	247Ø
FREE1	2472	FREE2	2568	FREE3	258Ø	FREE4	25B5
FREE5	25B6	FREE6	25BB	FREE7	261C	FREE8	2622
FREE9	2633	FREEMAP	24ØØ	FRET	25ØC	FSIZE	28Ø4
FTYPE	28Ø7	GAT	29ØØ	GETHL	2444	GETINFO	2514
GETPRMS	243E	GRANS	26E4	HARD	279E	HDATE	27BF
HDRIVE	27B2	HEADER	27AB	HFREEF	27F3	HFREEK	27D6
HIT	2AØØ	HNAME	27B5	HPOSSF	27F7	HPOSSK	27EØ
HUNDTAB	286Ø	ILDRNUM	245B	IOERR	241A	KEY	277F
KFLAG$	ØØØA	LF	ØØØA	LINBUF	2BØØ	LO1	27Ø6
LO2	2714	MAP	2496	MDDEN	2792	MTRK	27A4
NEXTLIN	24C8	NODISK	284B	NODISKN	2852	NO_DISK	2654
NUM	ØØ8Ø	NXTDRV	248F	ONESIDE	26Ø8	PAR_ERR	ØØ2C
PPARM	264B	PRINT	2788	PRMTBL$	2868	PRT	278B
RIGID	2798	SAVESP	2413	SFLAG$	ØØ12	SFLAG1	266C
SKIPMUL	2592	SKPLP	2432	STR	ØØ2Ø	TAB	ØØØ9
TPL	ØØØ8	VFLAG$	ØØ15	@@ABORT	B8C9	@@ADTSK	B95C
@@BANK	BE74	@@BKSP	BB54	@@BREAK	BE8A	@@CHNIO	B8B4
@@CKBRKC	BED8	@@CKDRV	B9BØ	@@CKEOF	BB69	@@CKTSK	B947
@@CLOSE	BB3F	@@CLS	BEC2	@@CMNDI	B8F3	@@CMNDR	B9Ø8
@@CTL	B718	@@DATE	B88A	@@DCSTAT	B9EF	@@DEBUG	B932
@@DECHEX	BDF4	@@DIRRD	BD61	@@DIRWR	BD76	@@DIV16	BDDF
@@DIV8	BDCA	@@DODIR	B9C5	@@DSP	B6DC	@@DSPLY	B77C
@@ERROR	B91D	@@EXIT	B8DE	@@FEXT	BCCE	@@FLAGS	BE5E
@@FNAME	BCE3	@@FSPEC	BCB9	@@GATRD	BD4C	@@GATWR	BD8B
@@GET	B6FØ	@@GTDCB	BDØD	@@GTDCT	BCF8	@@GTMOD	BD22
@@HDFMT	BA97	@@HEX16	BE33	@@HEX8	BE1E	@@HEXDEC	BEØ9

```
@@HIGH$    BE48 @@INIT    BB15 @@KBD     B754
@@KEY      B6C8 @@KEYIN   B768 @@KLTSK   B99B
@@LOAD     BC8F @@LOC     BB7E @@LOF     BB93
@@LOGER    B7B3 @@LOGOT   B7C8 @@MSG     B7FF
@@MUL16    BDB5 @@MUL8    BDAØ @@OPEN    BB2A
@@PARAM    B875 @@PAUSE   B86Ø @@PEOF    BBA8
@@POSN     BBBD @@PRINT   B814 @@PRT     B72C
@@PUT      B7Ø4 @@RAMDIR  B9DA @@RDSEC   BA6D
@@RDSSC    BD37 @@READ    BBD2 @@REMOV   BBØØ
@@RENAM    BAEB @@REW     BBE7 @@RMTSK   B971
@@RPTSK    B986 @@RREAD   BBFC @@RSLCT   BA58
@@RSTOR    BA19 @@RUN     BCA4 @@RWRIT   BC11
@@SEEK     BA43 @@SEEKSC  BC26 @@SKIP    BC3B
@@SLCT     BAØ4 @@STEPI   BA2E @@TIME    B89F
@@VDCTL    B84B @@VER     BC5Ø @@VRSEC   BA82
@@WEOF     BC65 @@WHERE   B74Ø @@WRITE   BC7A
@@WRSEC    BAAC @@WRSSC   BAC1 @@WRTRK   BAD6

24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

NOTES:

Command: LIB, CLS, TOF

Library: SYS6/SYS

ISAM # : 19H, 24H, 25H

```
                00100 ;LBLIB/ASM - LIB/CLS/TOF Commands
0000            00110         TITLE   <LIB - LS-DOS 6.2>
0000            00120 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00130 ;
000D            00140 CR      EQU     13
000A            00150 LF      EQU     10
1E00            00160 OVLAY$  EQU     1E00H
                00170 ;
2400            00180         ORG     2400H
                00190 ;
                00200 ;
                00210 ;       LIB command entry point
                00220 ;
2400 C32402     00230 LIB     JP      JPLIB           ;Skip around to 'LIB'
                00240 ;
                00250 ;       CLS command entry point
                00260 ;
2403 C31924     00270         JP      CLS             ;Skip around to 'CLS'
                00280 ;
                00290 ;       TOF command entry point
                00300 ;
                00310 TOF
2406 0E0C       00320         LD      C,12            ;Load TOF character
2408            00330         @@PRT                   ;Do @PRT SVC
2408 3E06       00001         LD      A,6
240A EF         00002         RST     40
240B 210000     00340         LD      HL,0            ;Init to no error
240E C8         00350         RET     Z               ;  and back if ok
240F 6F         00360 TOFERR  LD      L,A             ;Error code to HL
2410 2600       00370         LD      H,0
2412 F6C0       00380         OR      0C0H            ;Abbrev, return
2414 4F         00390         LD      C,A
2415            00400         @@ERROR
2415 3E1A       00003         LD      A,26
2417 EF         00004         RST     40
2418 C9         00410         RET
                00420 ;
2419 3E69       00430 CLS     LD      A,105           ;@CLS svc
241B EF         00440         RST     40              ;  and do it
241C AF         00450         XOR     A               ;Ret without error
241D 6F         00460         LD      L,A
241E 67         00470         LD      H,A
241F C9         00480         RET
                00490 ;
2420 0603       00500 JPLIB   LD      B,3             ;Init for 3 libraries
2422 C5         00510 LIB0    PUSH    BC
2423            00520         @@DSPLY LIBMSG
                00005         IFEQ    01H,1
2423 211024     00006         LD      HL,LIBMSG
                00007         ENDIF
2426 3E0A       00008         LD      A,10
2428 EF         00009         RST     40
2429 360A       00530         LD      (HL),LF         ;Put LF at start of
                00540                                 ;  string for next time
242B 2A021E     00550         LD      HL,(OVLAY$+2)   ;PU table pointer
242E 0602       00560 LIB1    LD      B,2             ;Dsply 1st command
2430 CD8824     00570         CALL    TAB             ;  tabbed in 2 spaces
2433 0E07       00580         LD      C,7             ;Init for 7-across
```

```
2435 7E       ØØ59Ø LIB2   LD      A,(HL)
2436 B7       ØØ6ØØ        OR      A
2437 2837     ØØ61Ø        JR      Z,LIB4          ;Jump on end
2439 E5       ØØ62Ø        PUSH    HL
243A 11Ø7ØØ   ØØ63Ø        LD      DE,7            ;Index to lib #
243D 19       ØØ64Ø        ADD     HL,DE
243E 3E8Ø     ØØ65Ø LIBX   LD      A,8ØH           ;Init for LIB-A
244Ø BE       ØØ66Ø        CP      (HL)            ;Is this command in
2441 28Ø4     ØØ67Ø        JR      Z,LIB2A         ;  the current library?
2443 F1       ØØ68Ø        POP     AF              ;Is not, skip past it
2444 23       ØØ69Ø        INC     HL
2445 18EE     ØØ7ØØ        JR      LIB2
              ØØ71Ø ;
2447 E1       ØØ72Ø LIB2A  POP     HL              ;Get 1st char of command
2448 C5       ØØ73Ø        PUSH    BC              ;Save reg C
2449 4E       ØØ74Ø        LD      C,(HL)          ;  and display in upper
244A          ØØ75Ø        @@DSP
244A 3EØ2     ØØØ1Ø        LD      A,2
244C EF       ØØØ11        RST     4Ø
244D Ø6Ø5     ØØ76Ø        LD      B,5             ;Write 6-char LIB word
244F 23       ØØ77Ø LIB3   INC     HL              ;Point to next char
245Ø 7E       ØØ78Ø        LD      A,(HL)
2451 FE2Ø     ØØ79Ø        CP      ' '             ;If space, don't lower
2453 28Ø2     ØØ8ØØ        JR      Z,$+4           ;  case the char
2455 EE2Ø     ØØ81Ø        XOR     2ØH
2457 4F       ØØ82Ø        LD      C,A             ;Xfer to C & display it
2458          ØØ83Ø        @@DSP
2458 3EØ2     ØØØ12        LD      A,2
245A EF       ØØØ13        RST     4Ø
245B 1ØF2     ØØ84Ø        DJNZ    LIB3
              ØØ85Ø ;
245D Ø6Ø5     ØØ86Ø        LD      B,5             ;Move over 5 spaces
245F CD8824   ØØ87Ø        CALL    TAB             ;  for start of next command
2462 23       ØØ88Ø        INC     HL              ;Bypass LIB parm vector
2463 23       ØØ89Ø        INC     HL
2464 23       ØØ9ØØ        INC     HL
2465 C1       ØØ91Ø        POP     BC              ;Get across counter
2466 ØD       ØØ92Ø        DEC     C               ;  & decrement
2467 2ØCC     ØØ93Ø        JR      NZ,LIB2         ;Loop on < 7
2469 ØEØD     ØØ94Ø        LD      C,CR            ;Write a new line
246B          ØØ95Ø        @@DSP
246B 3EØ2     ØØØ14        LD      A,2
246D EF       ØØØ15        RST     4Ø
246E 18BE     ØØ96Ø        JR      LIB1            ;Loop
              ØØ97Ø ;
247Ø ØEØD     ØØ98Ø LIB4   LD      C,CR            ;End with new line
2472          ØØ99Ø        @@DSP
2472 3EØ2     ØØØ16        LD      A,2
2474 EF       ØØØ17        RST     4Ø
2475 219A24   Ø1ØØØ        LD      HL,LIBMSG+1Ø
2478 34       Ø1Ø1Ø        INC     (HL)            ;Bump to next lib
2479 3A3F24   Ø1Ø2Ø        LD      A,(LIBX+1)      ;Advance RST code also
247C C62Ø     Ø1Ø3Ø        ADD     A,2ØH
247E 323F24   Ø1Ø4Ø        LD      (LIBX+1),A
2481 C1       Ø1Ø5Ø        POP     BC              ;Recover Lib count
2482 1Ø9E     Ø1Ø6Ø        DJNZ    LIBØ            ;Loop until all done
2484 21ØØØØ   Ø1Ø7Ø        LD      HL,Ø            ;Set no error
2487 C9       Ø1Ø8Ø        RET
              Ø1Ø9Ø ;
2488 ØE2Ø     Ø11ØØ TAB    LD      C,' '           ;Display spaces,
248A          Ø111Ø        @@DSP
```

```
248A 3E02      00018          LD      A,2
248C EF        00019          RST     40
248D 10F9      01120          DJNZ    TAB             ;  reg B has count
248F C9        01130          RET
               01140 ;
2490 00        01150 LIBMSG   DB      0,'Library <A>',CR
     4C 69 62 72 61 72 79 20
     3C 41 3E 0D
2400           01160          END     LIB
```

```
@@1             ØØØØ @@2             ØØØØ @@3             ØØØØ
@@4             ØØØØ @MOD2           ØØØØ @MOD4           FFFF
CLS             2419 CR              ØØØD JPLIB           242Ø
LF              ØØØA LIB             24ØØ LIBØ            2422
LIB1            242E LIB2            2435 LIB2A           2447
LIB3            244F LIB4            247Ø LIBMSG          249Ø
LIBX            243E OVLAY$          1EØØ TAB             2488
TOF             24Ø6 TOFERR          24ØF @@ABORT         748C
@@ADTSK         751F @@BANK          7A37 @@BKSP          7717
@@BREAK         7A4D @@CHNIO         7477 @@CKBRKC        7A9B
@@CKDRV         7573 @@CKEOF         772C @@CKTSK         75ØA
@@CLOSE         77Ø2 @@CLS           7A85 @@CMNDI         74B6
@@CMNDR         74CB @@CTL           72DB @@DATE          744D
@@DCSTAT        75B2 @@DEBUG         74F5 @@DECHEX        79B7
@@DIRRD         7924 @@DIRWR         7939 @@DIV16         79A2
@@DIV8          798D @@DODIR         7588 @@DSP           729F
@@DSPLY         733F @@ERROR         74EØ @@EXIT          74A1
@@FEXT          7891 @@FLAGS         7A21 @@FNAME         78A6
@@FSPEC         787C @@GATRD         79ØF @@GATWR         794E
@@GET           72B3 @@GTDCB         78DØ @@GTDCT         78BB
@@GTMOD         78E5 @@HDFMT         765A @@HEX16         79F6
@@HEX8          79E1 @@HEXDEC        79CC @@HIGH$         7AØB
@@INIT          76D8 @@KBD           7317 @@KEY           728B
@@KEYIN         732B @@KLTSK         755E @@LOAD          7852
@@LOC           7741 @@LOF           7756 @@LOGER         7376
@@LOGOT         738B @@MSG           73C2 @@MUL16         7978
@@MUL8          7963 @@OPEN          76ED @@PARAM         7438
@@PAUSE         7423 @@PEOF          776B @@POSN          778Ø
@@PRINT         73D7 @@PRT           72EF @@PUT           72C7
@@RAMDIR        759D @@RDSEC         763Ø @@RDSSC         78FA
@@READ          7795 @@REMOV         76C3 @@RENAM         76AE
@@REW           77AA @@RMTSK         7534 @@RPTSK         7549
@@RREAD         77BF @@RSLCT         761B @@RSTOR         75DC
@@RUN           7867 @@RWRIT         77D4 @@SEEK          76Ø6
@@SEEKSC        77E9 @@SKIP          77FE @@SLCT          75C7
@@STEPI         75F1 @@TIME          7462 @@VDCTL         74ØE
@@VER           7813 @@VRSEC         7645 @@WEOF          7828
@@WHERE         73Ø3 @@WRITE         783D @@WRSEC         766F
@@WRSSC         7684 @@WRTRK         7699
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: LINK

Library: SYS6/SYS

ISAM # : 62H

```
                00100 ;LBLINK/ASM - LINK Command
0000            00110         TITLE   <LINK - LS-DOS 6.2>
                00120 ;
000D            00130 CR      EQU     13
0000            00140 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00150 ;
2400            00160         ORG     2400H
                00170 ;
2400 11AA24     00180 LINK    LD      DE,FCB1         ;Fetch source spec
2403            00190         @@FSPEC
2403 3E4E       00001         LD      A,78
2405 EF         00002         RST     40
2406 2010       00200         JR      NZ,SPCERR1      ;Exit if bad name
2408 1A         00210         LD      A,(DE)          ;Must be a device
2409 FE2A       00220         CP      '*'
240B 207E       00230         JR      NZ,SPCERR
                00240 ;
                00250 ;       Fetch the second device spec
                00260 ;
240D 11AD24     00270         LD      DE,FCB2         ;Fetch destination spec
2410            00280         @@FSPEC
2410 3E4E       00003         LD      A,78
2412 EF         00004         RST     40
2413 2076       00290         JR      NZ,SPCERR       ;Exit if bad name
2415 1A         00300         LD      A,(DE)
2416 FE2A       00310         CP      '*'             ;Must also be a device
2418 2071       00320 SPCERR1 JR      NZ,SPCERR
                00330 ;
                00340 ;       Make sure source <> destination
                00350 ;
241A 2AAB24     00360         LD      HL,(FCB1+1)     ;If devices are the same,
241D ED5BAE24   00370         LD      DE,(FCB2+1)     ;  then quit
2421 ED52       00380         SBC     HL,DE
2423 2866       00390         JR      Z,SPCERR
                00400 ;
                00410 ;       Locate a spare DCB for the link
                00420 ;
2425 110000     00430         LD      DE,0
2428            00440         @@GTDCB
2428 3E52       00005         LD      A,82
242A EF         00006         RST     40
242B 3E21       00450         LD      A,33            ;Init "No device space...
242D 2052       00460         JR      NZ,IOERR
242F 224A24     00470         LD      (LINKDCB+1),HL  ;Save pointer
                00480 ;
                00490 ;       Locate destination DCB address
                00500 ;
2432 ED5BAE24   00510         LD      DE,(FCB2+1)     ;Grab DCB name
2436            00520         @@GTDCB                 ;Locate its address
2436 3E52       00007         LD      A,82
2438 EF         00008         RST     40
2439 2046       00530         JR      NZ,IOERR        ;Jump if not found
243B 226824     00540         LD      (DSTDCB+1),HL   ;Save destination
                00550 ;
                00560 ;       Locate source DCB address
                00570 ;
243E ED5BAB24   00580         LD      DE,(FCB1+1)     ;Get 1st DCB name
2442            00590         @@GTDCB                 ;Locate in device tables
```

```
2442 3E52      00009          LD      A,82
2444 EF        00010          RST     40
2445 203A      00600          JR      NZ,IOERR        ;Jump if not found
2447 E5        00610          PUSH    HL              ;Save pointer we used
2448 F3        00620          DI                      ;Can't interrupt
               00630 ;
               00640 ;        Save the old device vector while stuffing new
               00650 ;
2449 010000    00660 LINKDCB  LD      BC,$-$          ;P/u link DCB address
244C 2C        00670          INC     L               ;Bump to vector
244D 7E        00680          LD      A,(HL)          ;Save what's there
244E 71        00690          LD      (HL),C          ;Stuff link address
244F 4F        00700          LD      C,A             ;  into DCB of source
2450 2C        00710          INC     L               ;  while saving old
2451 7E        00720          LD      A,(HL)          ;  vector for storage
2452 70        00730          LD      (HL),B          ;  (could be a FCB)
2453 47        00740          LD      B,A
               00750 ;
               00760 ;        Now set LINK bit and rest of LINK DCB block
               00770 ;
2454 E1        00780          POP     HL              ;Rcvr ptr to source DCB+0
2455 7E        00790          LD      A,(HL)          ;Init the LINK bit
2456 F5        00800          PUSH    AF              ;Save old TYPE byte
2457 E607      00810          AND     7               ;Strip flags
2459 F620      00820          OR      20H             ;Set Link bit
245B 77        00830          LD      (HL),A          ;Show source is linked
245C 2A4A24    00840          LD      HL,(LINKDCB+1)  ;P/u link DCB address
245F F1        00850          POP     AF              ;Rcvr source TYPE
2460 77        00860          LD      (HL),A          ;New LINK TYPE
2461 2C        00870          INC     L
2462 71        00880          LD      (HL),C          ;Stuff source vector
2463 2C        00890          INC     L
2464 70        00900          LD      (HL),B
2465 2C        00910          INC     L               ;Bypass dest TYPE
2466 2C        00920          INC     L
2467 010000    00930 DSTDCB   LD      BC,$-$          ;P/u destination DCB addr
246A 71        00940          LD      (HL),C          ;  & stuff into link DCB
246B 2C        00950          INC     L
246C 70        00960          LD      (HL),B
246D 2C        00970          INC     L
246E E5        00980          PUSH    HL              ;Save name field pointer
246F 114C2F    00990          LD      DE,'/L'         ;Let's find a link name
2472 14        01000 NAMLP    INC     D               ;Bump "2nd" character
2473           01010          @@GTDCB                 ;If we find this name
2473 3E52      00011          LD      A,82
2475 EF        00012          RST     40
2476 28FA      01020          JR      Z,NAMLP         ;  look for another
2478 E1        01030          POP     HL              ;Get name pointer
2479 73        01040          LD      (HL),E          ;  & stuff in the
247A 2C        01050          INC     L               ;  selected link name
247B 72        01060          LD      (HL),D
247C FB        01070          EI                      ;Start tasks again
247D 210000    01080          LD      HL,0            ;Show no error
2480 C9        01090          RET
               01100 ;
               01110 ;        Error processing
               01120 ;
2481 6F        01130 IOERR    LD      L,A             ;Move error # into HL
2482 2600      01140          LD      H,0
2484 F6C0      01150          OR      0C0H            ;Abbrev & return
2486 4F        01160          LD      C,A
```

```
2487                01170          @@ERROR
2487 3E1A           00013          LD      A,26
2489 EF             00014          RST     40
248A C9             01180          RET
248B 219524         01190 SPCERR   LD      HL,SPCERR$      ;Bad devspec found
248E                01200          @@LOGOT
                    00015          IFEQ    00H,1
                    00016          LD      HL,
                    00017          ENDIF
248E 3E0C           00018          LD      A,12
2490 EF             00019          RST     40
2491 21FFFF         01210          LD      HL,-1
2494 C9             01220          RET
2495 44             01230 SPCERR$ DB      'Device spec required',CR
     65 76 69 63 65 20 73 70
     65 63 20 72 65 71 75 69
     72 65 64 0D
0003                01240 FCB1     DS      3               ;Only 3-bytes needed
0020                01250 FCB2     DS      32
2400                01260          END     LINK
```

```
@@1      ØØØØ @@2      ØØØØ @@3      ØØØØ
@@4      ØØØØ @MOD2    ØØØØ @MOD4    FFFF
CR       ØØØD DSTDCB   2467 FCB1     24AA
FCB2     24AD IOERR    2481 LINK     24ØØ
LINKDCB  2449 NAMLP    2472 SPCERR   248B
SPCERR$  2495 SPCERR1  2418 @@ABORT  7678
@@ADTSK  77ØB @@BANK   7C23 @@BKSP   79Ø3
@@BREAK  7C39 @@CHNIO  7663 @@CKBRKC 7C87
@@CKDRV  775F @@CKEOF  7918 @@CKTSK  76F6
@@CLOSE  78EE @@CLS    7C71 @@CMNDI  76A2
@@CMNDR  76B7 @@CTL    74C7 @@DATE   7639
@@DCSTAT 779E @@DEBUG  76E1 @@DECHEX 7BA3
@@DIRRD  7B1Ø @@DIRWR  7B25 @@DIV16  7B8E
@@DIV8   7B79 @@DODIR  7774 @@DSP    748B
@@DSPLY  752B @@ERROR  76CC @@EXIT   768D
@@FEXT   7A7D @@FLAGS  7CØD @@FNAME  7A92
@@FSPEC  7A68 @@GATRD  7AFB @@GATWR  7B3A
@@GET    749F @@GTDCB  7ABC @@GTDCT  7AA7
@@GTMOD  7AD1 @@HDFMT  7846 @@HEX16  7BE2
@@HEX8   7BCD @@HEXDEC 7BB8 @@HIGH$  7BF7
@@INIT   78C4 @@KBD    75Ø3 @@KEY    7477
@@KEYIN  7517 @@KLTSK  774A @@LOAD   7A3E
@@LOC    792D @@LOF    7942 @@LOGER  7562
@@LOGOT  7577 @@MSG    75AE @@MUL16  7B64
@@MUL8   7B4F @@OPEN   78D9 @@PARAM  7624
@@PAUSE  76ØF @@PEOF   7957 @@POSN   796C
@@PRINT  75C3 @@PRT    74DB @@PUT    74B3
@@RAMDIR 7789 @@RDSEC  781C @@RDSSC  7AE6
@@READ   7981 @@REMOV  78AF @@RENAM  789A
@@REW    7996 @@RMTSK  772Ø @@RPTSK  7735
@@RREAD  79AB @@RSLCT  78Ø7 @@RSTOR  77C8
@@RUN    7A53 @@RWRIT  79CØ @@SEEK   77F2
@@SEEKSC 79D5 @@SKIP   79EA @@SLCT   77B3
@@STEPI  77DD @@TIME   764E @@VDCTL  75FA
@@VER    79FF @@VRSEC  7831 @@WEOF   7A14
@@WHERE  74EF @@WRITE  7A29 @@WRSEC  785B
@@WRSSC  787Ø @@WRTRK  7885
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: LIST

Library: SYS6/SYS

ISAM # : 41H

```
                00100 ;LBLIST/ASM - List Command
0000            00110         TITLE   <LIST - LS-DOS 6.2>
                00120 ;
0000            00130 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
0000            00140 *GET    VALUES:3                ;Misc. equates
                03920 ;VALUES/ASM - Version 6
                03930 *LIST OFF
                04200 *LIST ON
                00150 ;
0006            00160 @PRT    EQU     6
0002            00170 @DSP    EQU     2
                00180 ;
2400            00190         ORG     2400H
                00200 ;
                00210 START
2400            00220         @@CKBRKC                ;Check break
2400 3E6A       00001         LD      A,106
2402 EF         00002         RST     40
2403 2804       00230         JR      Z,LISTA         ;Continue if not
2405 21FFFF     00240         LD      HL,-1           ;  else abort
2408 C9         00250         RET
                00260 ;
                00270 ;       <BREAK> not hit, Execute Module
                00280 ;
                00290 LISTA
2409 ED731124   00300         LD      (SAVESP+1),SP   ;Save SP
240D CD2D24     00310         CALL    LIST            ;List a file
2410 310000     00320 SAVESP  LD      SP,$-$          ;P/u SP address
2413            00330         @@CKBRKC                ;Reset if user pressed break
2413 3E6A       00003         LD      A,106
2415 EF         00004         RST     40
2416 C9         00340         RET                     ;Abort
                00350 ;
                00360 ;       I/O Error Processing - Display & Abort
                00370 ;
2417 2600       00380 IOERR   LD      H,0             ;Set HL = Error #
2419 6F         00390         LD      L,A             ;
241A F6C0       00400         OR      0C0H            ;Short error message
241C 4F         00410         LD      C,A             ;Stuff error # in C
241D            00420         @@ERROR                 ;Display error
241D 3E1A       00005         LD      A,26
241F EF         00006         RST     40
2420 18EE       00430         JR      SAVESP          ;Abort
                00440 ;
                00450 ;       Internal Error Message Display Routine
                00460 ;
2422 210727     00470 SPCREQ  LD      HL,SPCREQ$      ;File spec Required
2425            00480         @@LOGOT                 ;Log error message
                00007         IFEQ    00H,1
                00008         LD      HL,
                00009         ENDIF
2425 3E0C       00010         LD      A,12
2427 EF         00011         RST     40
2428 21FFFF     00490 ABORT   LD      HL,-1           ;Internal error = -1
242B 18E3       00500         JR      SAVESP          ;P/u stack & return
                00510 ;
                00520 ;       LIST - List a file in hex or ASCII
                00530 ;
```

```
242D CD9E26   00540 LIST    CALL    RESKFL          ;Reset pause, & enter
              00550 ;
              00560 ;       Find parameter entries if existent
              00570 ;
2430 E5       00580         PUSH    HL              ;Save command ptr
2431 7E       00590 FPLP    LD      A,(HL)          ;P/u character
2432 FE28     00600         CP      '('             ;Parameter(s) ?
2434 2807     00610         JR      Z,GETPRM        ;Yes - go get 'em
2436 FE0E     00620         CP      CR+1            ;End of line ?
2438 380C     00630         JR      C,RESTPTR       ;Yes - restore ptr
243A 23       00640         INC     HL              ;No - bump til end
243B 18F4     00650         JR      FPLP            ;Do til eol or "("
              00660 ;
              00670 ;       Process any parameters entered
              00680 ;
243D 113227   00690 GETPRM  LD      DE,PRMTBL$      ;DE => Parameter table
2440          00700         @@PARAM                 ;@PARAM
2440 3E11     00012         LD      A,17
2442 EF       00013         RST     40
2443 C21724   00710         JP      NZ,IOERR        ;NZ - "Parameter Error"
2446 E1       00720 RESTPTR POP     HL              ;Recover ptr
              00730 ;
              00740 ;       Skip command line blanks
              00750 ;
2447 7E       00760 IGSPC   LD      A,(HL)          ;P/u character
2448 23       00770         INC     HL              ;Bump ptr
2449 FE20     00780         CP      ' '             ;Space ?
244B 28FA     00790         JR      Z,IGSPC         ;Yes - ignore them
244D 2B       00800         DEC     HL              ;HL => First non-space
              00810 ;
              00820 ;       Check if the filespec is legal in format
              00830 ;
244E 118527   00840         LD      DE,FCB1         ;Fetch file spec
2451          00850         @@FSPEC                 ;Filespec legal ?
2451 3E4E     00014         LD      A,78
2453 EF       00015         RST     40
2454 C22224   00860         JP      NZ,SPCREQ       ;No - filespec required
              00870 ;
              00880 ;       If this is a device - don't LIST
              00890 ;
2457 1A       00900         LD      A,(DE)          ;Is this a device ?
2458 FE2A     00910         CP      '*'             ;
245A CA2224   00920         JP      Z,SPCREQ        ;Yes - Filespec req
              00930 ;
              00940 ;       Save the original filespec
              00950 ;
245D EB       00960         EX      DE,HL           ;HL => Source FCB
245E 11A527   00970         LD      DE,FCB2         ;DE => Duplicate FCB
2461 012000   00980         LD      BC,32           ;32 bytes tm xfer
2464 E5       00990         PUSH    HL              ;Save source ptr
2465 EDB0     01000         LDIR                    ;Xfer
2467 D1       01010         POP     DE              ;DE => Source FCB
              01020 ;
              01030 ;       Stuff default extension of /TXT to source
              01040 ;
2468 212F27   01050         LD      HL,TXTEXT       ;HL => TXT
246B          01060         @@FEXT                  ;Stuff TXT if no ext
246B 3E4F     00016         LD      A,79
246D EF       00017         RST     40
              01070 ;
              01080 ;       Open the file with LRL of 256
```

```
                    01090 ;
246E 0600           01100           LD      B,0             ;Set LRL (B) = 256
2470 CDF926         01110           CALL    OPEN            ;Open file with LRL=256
2473 2816           01120           JR      Z,INITLRL       ;Good - LIST it
                    01130 ;
                    01140 ;         Error - Was it a "File not Found" Error ?
                    01150 ;
2475 FE18           01160           CP      24              ;"File not Found" ?
2477 C21724         01170           JP      NZ,IOERR        ;No - I/O Error
                    01180 ;
                    01190 ;         OPEN original filespec instead of TXT
                    01200 ;
247A 0E20           01210           LD      C,32            ;BC = 32 bytes to xfer
247C D5             01220           PUSH    DE              ;DE => FCB1
247D E5             01230           PUSH    HL              ;Save I/O buff ptr
247E 21A527         01240           LD      HL,FCB2         ;HL => Filespec
2481 EDB0           01250           LDIR                    ;Xfer original filespec
2483 E1             01260           POP     HL              ;HL => I/O buffer
2484 D1             01270           POP     DE              ;DE => unopen FCB
                    01280 ;
                    01290 ;         Open Original Filespec without extension
                    01300 ;
2485 CDF926         01310           CALL    OPEN            ;Open the file
2488 C21724         01320           JP      NZ,IOERR        ;I/O error - abort
                    01330 ;
                    01340 ;         Pick up the DEC from the FCB
                    01350 ;
248B D5             01360 INITLRL   PUSH    DE              ;Xfer fcb to IX
248C DDE1           01370           POP     IX
248E DD4607         01380           LD      B,(IX+7)        ;P/u DEC
2491 DD4E06         01390           LD      C,(IX+6)        ;  & drive #
2494                01400           @@DIRRD                 ;Read in directory record
2494 3E57           00018           LD      A,87
2496 EF             00019           RST     40
                    01410 ;
                    01420 ;         Was the LRL parm specified ?
                    01430 ;
2497 3A6427         01440           LD      A,(LRESP)       ;P/u response
249A B7             01450           OR      A
249B 2008           01460           JR      NZ,SKIPLRL      ;Go if entered
                    01470 ;
                    01480 ;         No LRL parm, get from directory record
                    01490 ;
249D 7D             01500           LD      A,L             ;Position to DIR+4
249E C604           01510           ADD     A,4
24A0 6F             01520           LD      L,A             ;HL => LRL byte (DIR+4)
                    01530 ;
                    01540 ;         Pick up LRL & stuff into LRL parameter
                    01550 ;
24A1 7E             01560           LD      A,(HL)          ;P/u LRL
24A2 329E25         01570           LD      (LPARM+1),A     ;Set dir LRL into parm
                    01580 ;
                    01590 ;         Pick up LRL & stuff into FCB
                    01600 ;
24A5 3A9E25         01610 SKIPLRL   LD      A,(LPARM+1)     ;P/u possible new LRL
24A8 DD7709         01620           LD      (IX+9),A        ;Put into FCB
24AB DDCB01FE       01630           SET     7,(IX+1)        ;Mark for byte I/O
                    01640 ;
                    01650 ;         Check if TAB (T) parm is flag or numeric
                    01660 ;
24AF 3A7027         01670           LD      A,(TRESP)       ;P/u response byte
```

```
24B2 E640      01680          AND     40H             ;Flag response ?
24B4 3A3425    01690          LD      A,(TPARM+1)     ;P/u value
24B7 2805      01700          JR      Z,NOTFLG        ;No - numeric
               01710 ;
               01720 ;        Flag Response - Is it ON (T=8) or OFF (T=1)
               01730 ;
24B9 3C        01740          INC     A               ;Tab=OFF (NZ) or ON (Z) ?
24BA 2002      01750          JR      NZ,NOTFLG       ;Tab = OFF ---> TABEXP=1
24BC 3E08      01760          LD      A,8             ;Tab = ON ---> TABEXP=8
               01770 ;
               01780 ;        P/u TAB (T) parm value & stuff into routine
               01790 ;
24BE 323E25    01800 NOTFLG   LD      (TABEXP+1),A    ;Stuff away
24C1 3D        01810          DEC     A               ;Range can
24C2 FE20      01820          CP      32              ;Only be between 1-32.
24C4 3E2C      01830          LD      A,PAR_ERR       ;Greater - Parm Error
24C6 D21724    01840          JP      NC,IOERR
               01850 ;
               01860 ;        Was the P (print) parameter entered ?
               01870 ;
24C9 010000    01880 PPARM    LD      BC,$-$          ;P/u P parameter
24CC 78        01890          LD      A,B             ;Was it specified ?
24CD B1        01900          OR      C
24CE 2805      01910          JR      Z,HPARM         ;No - use @DSP
               01920 ;
               01930 ;        Stuff @PRT SVC # in output routine
               01940 ;
24D0 3E06      01950          LD      A,@PRT          ;@PRT SVC Number
24D2 325B26    01960          LD      (PUTOUT1+1),A   ;Overwrite @DSP
               01970 ;
               01980 ;        Hex Parameter Entered ?
               01990 ;
24D5 010000    02000 HPARM    LD      BC,$-$          ;P/u Hex parm
24D8 78        02010          LD      A,B             ;Entered ?
24D9 B1        02020          OR      C
24DA C27E25    02030          JP      NZ,RPARM        ;Yes - check out Records
               02040 ;
               02050 ;        Routine to LIST a file in ASCII
               02060 ;
24DD 327427    02070          LD      (BYTCTR),A      ;Init counter to 0
               02080 ;
24E0 010100    02090 LINPRM   LD      BC,1            ;P/u start line parm
24E3 0B        02100          DEC     BC              ;Count down for start
24E4 78        02110          LD      A,B             ;Ready to list ?
24E5 B1        02120          OR      C
24E6 2812      02130          JR      Z,BGNLIN        ;Go list/print if ready
24E8 118527    02140          LD      DE,FCB1         ;DE => FCB
               02150 ;
               02160 ;        Ignore all lines until specified start posn
               02170 ;
24EB           02180 FND1ST   @@GET                   ;Get a character
24EB 3E03      00020          LD      A,3
24ED EF        00021          RST     40
24EE C21724    02190          JP      NZ,IOERR        ;Error - abort
24F1 FE0D      02200          CP      CR              ;End of line?
24F3 20F6      02210          JR      NZ,FND1ST       ;Keep reading
               02220 ;
               02230 ;        Finished with line - decrement LINE count
               02240 ;
24F5 0B        02250          DEC     BC              ;Dec line counter
24F6 78        02260          LD      A,B             ;Finished
```

```
24F7 B1       02270          OR      C               ;Ignoring lines ?
24F8 20F1     02280          JR      NZ,FND1ST       ;No - loop until LINES=0
              02290 ;
              02300 ;        Start LISTing File
              02310 ;
24FA 2AE124   02320 BGNLIN   LD      HL,(LINPRM+1)   ;P/u original line #
24FD 011A27   02330          LD      BC,VARDOT       ;Convert to Decimal ASCII
2500 CDB226   02340          CALL    CVTDEC          ;  and stuff into string
              02350 ;
              02360 ;        Read in a character from the file
              02370 ;
2503 118527   02380 GETCHR   LD      DE,FCB1         ;DE => FCB
2506          02390          @@GET                   ;Get a byte
2506 3E03     00022          LD      A,3
2508 EF       00023          RST     40
2509 205D     02400          JR      NZ,GOTERR       ;Go if read error
250B 322125   02410          LD      (PUCHAR+1),A    ;Hang on to char
              02420 ;
              02430 ;        Test if NUM parameter was entered
              02440 ;
250E 010000   02450 NPARM    LD      BC,$-$          ;P/u N parameter
2511 78       02460          LD      A,B             ;Was it entered ?
2512 B1       02470          OR      C
2513 280B     02480          JR      Z,PUCHAR        ;No - don't print line #
              02490 ;
              02500 ;        N parm entered - print line # & Increment it
              02510 ;
2515 211A27   02520          LD      HL,VARDOT       ;HL => Buffer with line #
2518 E5       02530          PUSH    HL              ;Save ptr
2519 CD7926   02540          CALL    PUTLINE         ;Output line @ HL
251C E1       02550          POP     HL              ;Restore line # ptr
251D CDE826   02560          CALL    INCNUM          ;Increment line #
              02570 ;
              02580 ;        Pick up character and Check if high bit set
              02590 ;
2520 3E00     02600 PUCHAR   LD      A,$-$           ;P/u character
2522 07       02610 DLOOP    RLCA                    ;Get HB into Carry flag
              02620 ;
              02630 ;        Reset High bit unless A8 parameter entered
              02640 ;
2523 110000   02650 A8PARM   LD      DE,$-$          ;P/u A8 parm
2526 14       02660          INC     D               ;Was it entered ?
2527 15       02670          DEC     D
2528 2003     02680          JR      NZ,A8BIT        ;Yes - don't change byte
252A CB3F     02690          SRL     A               ;Reset Bit 7
252C 1E       02700          DB      1EH             ;LD E,nn instruction
252D 0F       02710 A8BIT    RRCA                    ;Use all 8 bits
              02720 ;
              02730 ;        Is the character a Tab ?
              02740 ;
252E F5       02750          PUSH    AF              ;Save C flag
252F FE09     02760          CP      TAB             ;Was it a tab ?
2531 201B     02770          JR      NZ,NOTTAB       ;No - don't check TAB
              02780 ;
              02790 ;        Character is a Tab - Was T=N specified ?
              02800 ;
2533 110800   02810 TPARM    LD      DE,0008H        ;P/u TAB parm (Default=8)
2536 1C       02820          INC     E               ;TAB = N ?
2537 1D       02830          DEC     E
2538 2814     02840          JR      Z,NOTTAB        ;Yes - don't expand
              02850 ;
```

```
                    Ø286Ø ;       P/u column # & calculate # of spaces to pad
                    Ø287Ø ;
253A 3A7427         Ø288Ø         LD      A,(BYTCTR)      ;P/u column number
253D ØEØ8           Ø289Ø TABEXP  LD      C,8             ;P/u TAB expansion #
253F 91             Ø29ØØ CLOOP   SUB     C
2540 30FD           Ø291Ø         JR      NC,CLOOP
2542 ED44           Ø292Ø         NEG                     ;Subtract A from Col #
                    Ø293Ø ;
                    Ø294Ø ;       Output A blank spaces for tab expansion
                    Ø295Ø ;
2544 47             Ø296Ø         LD      B,A             ;Put # spaces in B
2545 3E2Ø           Ø297Ø TP1     LD      A,' '           ;Pad with a space
2547 CD5826         Ø298Ø         CALL    PUTOUT          ;Output character
254A 1ØF9           Ø299Ø         DJNZ    TP1             ;B spaces to output
254C 18Ø3           Ø3ØØØ         JR      WASTAB          ;Check byte counter
                    Ø3Ø1Ø ;
                    Ø3Ø2Ø ;       Character was not a tab, display it
                    Ø3Ø3Ø ;
254E CD5826         Ø3Ø4Ø NOTTAB  CALL    PUTOUT          ;Print the character
2551 3A7427         Ø3Ø5Ø WASTAB  LD      A,(BYTCTR)      ;P/u byte counter
2554 B7             Ø3Ø6Ø         OR      A               ;If C/R printed
2555 CC8626         Ø3Ø7Ø         CALL    Z,CKPAWS        ;Then check for <PAUSE>
                    Ø3Ø8Ø ;
                    Ø3Ø9Ø ;       Check for <PAUSE> if hi bit set on character
                    Ø31ØØ ;
2558 F1             Ø311Ø         POP     AF              ;Get back char
2559 DC8626         Ø312Ø         CALL    C,CKPAWS        ;If high bit was set
                    Ø313Ø ;
                    Ø314Ø ;       If character = C/R then read in another line
                    Ø315Ø ;
255C FEØD           Ø316Ø         CP      CR              ;Was it C/R ?
255E 28A3           Ø317Ø         JR      Z,GETCHR        ;Yes - get new line
                    Ø318Ø ;
                    Ø319Ø ;       Get another byte from file
                    Ø32ØØ ;
2560 118527         Ø321Ø         LD      DE,FCB1         ;DE => FCB
2563                Ø322Ø         @@GET                   ;Get a byte
2563 3EØ3           ØØØ24         LD      A,3
2565 EF             ØØØ25         RST     4Ø
2566 28BA           Ø323Ø         JR      Z,DLOOP         ;Good - check character
                    Ø324Ø ;
                    Ø325Ø ;       I/O Error on @GET - Output a Carriage Return
                    Ø326Ø ;
2568 F5             Ø327Ø GOTERR  PUSH    AF              ;Save error code
2569 3EØD           Ø328Ø         LD      A,CR            ;Write end of line
256B CD5826         Ø329Ø         CALL    PUTOUT
256E F1             Ø33ØØ         POP     AF              ;Rcvr error code
                    Ø331Ø ;
                    Ø332Ø ;       If End of File Error - Exit normally
                    Ø333Ø ;
256F 21ØØØØ         Ø334Ø         LD      HL,Ø            ;Set HL = Ø (normal exit)
2572 FE1C           Ø335Ø         CP      1CH             ;EOF?
2574 CA1Ø24         Ø336Ø GTBK    JP      Z,SAVESP        ;Exit if so
2577 FE1D           Ø337Ø         CP      1DH             ;NRN > ERN?
2579 28F9           Ø338Ø         JR      Z,GTBK          ;Yes - leave
257B C31724         Ø339Ø         JP      IOERR           ;Other - Abort
                    Ø34ØØ ;
                    Ø341Ø ;       LIST a file in HEX format
                    Ø342Ø ;
257E Ø1ØØØØ         Ø343Ø RPARM   LD      BC,$-$          ;P/u starting Record #
2581 118527         Ø344Ø         LD      DE,FCB1         ;DE => FCB
```

```
2584            03450          @@POSN                  ;Position to Record BC
2584 3E42       00026          LD      A,66
2586 EF         00027          RST     40
2587 C21724     03460          JP      NZ,IOERR        ;Abort on position error
                03470 ;
                03480 ;        Reset byte counter to Zero
                03490 ;
258A AF         03500 DOHEX    XOR     A               ;Init byte counter to 0
258B 32A225     03510          LD      (DOHEX1+1),A
                03520 ;
                03530 ;        Stuff Record Number in Line Number buffer
                03540 ;
258E ED5B7F25   03550          LD      DE,(RPARM+1)    ;P/u Record Number
2592 212327     03560          LD      HL,VARCLN+1     ;HL => Hex ASCII dest
2595            03570          @@HEX16                 ;Convert DE to ASCII
2595 3E63       00028          LD      A,99
2597 EF         00029          RST     40
                03580 ;
                03590 ;        Bump Record Number & stuff into RPARM
                03600 ;
2598 13         03610          INC     DE              ;Bump by one
2599 ED537F25   03620          LD      (RPARM+1),DE    ;  & store for next time
                03630 ;
259D 010000     03640 LPARM    LD      BC,$-$          ;P/u LRL
                03650 ;
                03660 ;        Convert Byte counter to Hex & stuff in buff
                03670 ;
25A0 C5         03680 DISBYTE  PUSH    BC              ;Save bytes left in Rec
25A1 0E00       03690 DOHEX1   LD      C,$-$           ;P/u byte counter
25A3 212827     03700          LD      HL,VAREQU       ;HL => Hex ASCII dest
25A6            03710          @@HEX8                  ;Cvrt C to ASCII @ HL
25A6 3E62       00030          LD      A,98
25A8 EF         00031          RST     40
25A9 C1         03720          POP     BC              ;BC = Bytes left in Rec
                03730 ;
                03740 ;        Display Record Number/Starting byte string
                03750 ;
25AA 212227     03760          LD      HL,VARCLN       ;HL => Display buffer
25AD CD7926     03770          CALL    PUTLINE         ;Display Rec # byte
                03780 ;
                03790 ;        P/u byte counter & add 16 (BPL) & stuff away
                03800 ;
25B0 3AA225     03810          LD      A,(DOHEX1+1)    ;P/u byte counter
25B3 0610       03820          LD      B,16            ;Set B = 16 bytes
25B5 80         03830          ADD     A,B             ;Add 16 to byte count
25B6 32A225     03840          LD      (DOHEX1+1),A    ;  & stuff into LD C inst
                03850 ;
25B9 217527     03860          LD      HL,LINBUF       ;HL => Line buffer
                03870 ;
                03880 ;        Get a byte from the File
                03890 ;
25BC 118527     03900 DOHEX2   LD      DE,FCB1         ;DE => FCB
25BF            03910          @@GET                   ;Get a byte
25BF 3E03       00032          LD      A,3
25C1 EF         00033          RST     40
25C2 280B       03920          JR      Z,DOHEX4        ;Good - stuff byte
                03930 ;
                03940 ;        End of File Error ?
                03950 ;
25C4 F5         03960          PUSH    AF              ;Save error code
25C5 FE1C       03970          CP      1CH             ;EOF?
```

```
25C7 28Ø5      Ø398Ø          JR      Z,DOHEX3        ;Yes - Okay
               Ø399Ø ;
               Ø4ØØØ ;        Past End of File Error ?
               Ø4Ø1Ø ;
25C9 FE1D      Ø4Ø2Ø          CP      1DH             ;NRN>ERN?
25CB C21724    Ø4Ø3Ø          JP      NZ,IOERR        ;Another error
               Ø4Ø4Ø ;
               Ø4Ø5Ø ;        Recover Flags & check type of Error
               Ø4Ø6Ø ;
25CE F1        Ø4Ø7Ø DOHEX3   POP     AF              ;Recover error code
25CF 2Ø18      Ø4Ø8Ø DOHEX4   JR      NZ,DOHEX5       ;Bypass if at end of file
               Ø4Ø9Ø ;
               Ø41ØØ ;        Stuff Character in buffer & bump
               Ø411Ø ;
25D1 77        Ø412Ø          LD      (HL),A          ;Stuff byte in buffer
25D2 23        Ø413Ø          INC     HL              ;Bump ptr
               Ø414Ø ;
               Ø415Ø ;        Output byte in Hex & follow with a space
               Ø416Ø ;
25D3 CD4726    Ø417Ø          CALL    CVTHEX          ;Output the byte in hex
25D6 3E2Ø      Ø418Ø          LD      A,' '           ;  followed by a space
25D8 CD5826    Ø419Ø          CALL    PUTOUT
               Ø42ØØ ;
               Ø421Ø ;        Output an extra space if halfway in line
               Ø422Ø ;
25DB 78        Ø423Ø          LD      A,B             ;P/u byte counter
25DC FEØ9      Ø424Ø          CP      9               ;Halfway yet?
25DE CC7526    Ø425Ø          CALL    Z,WR1SPA        ;Yes - display space
               Ø426Ø ;
               Ø427Ø ;        Dec Chars/Line & # of chars left in Rec
               Ø428Ø ;
25E1 Ø5        Ø429Ø          DEC     B               ;Count down
25E2 ØD        Ø43ØØ          DEC     C               ;Count down the LRL
25E3 28Ø4      Ø431Ø          JR      Z,DOHEX5        ;Done - get next record
               Ø432Ø ;
               Ø433Ø ;        Finished with Line ?
               Ø434Ø ;
25E5 78        Ø435Ø          LD      A,B             ;Done with line ?
25E6 B7        Ø436Ø          OR      A
25E7 2ØD3      Ø437Ø          JR      NZ,DOHEX2       ;No - do til 16 chars
               Ø438Ø ;
               Ø439Ø ;        Finished with Line or Logical Record
               Ø44ØØ ;
25E9 F5        Ø4411 DOHEX5   PUSH    AF              ;End the line
25EA 78        Ø4421          LD      A,B             ;P/u byte counter
25EB FE1Ø      Ø4431          CP      16              ;Done with this line ?
25ED 2826      Ø4441          JR      Z,PRTLIN2       ;Yes - get another record
               Ø4451 ;
               Ø4461 ;        Display ASCII equivalent of line
               Ø4471 ;
25EF C5        Ø4481 PRTLIN   PUSH    BC              ;Save counters
               Ø4491 ;
               Ø45ØØ ;        Multiply # of chars not printed by three
               Ø4511 ;
25FØ 78        Ø4521          LD      A,B             ;P/u # chars not printed
25F1 87        Ø4531          ADD     A,A             ;X 2
25F2 8Ø        Ø4541          ADD     A,B             ;X 3
25F3 47        Ø4551          LD      B,A             ;Stuff in B
               Ø4561 ;
               Ø4571 ;        Add two extra spaces if more than halfway
               Ø4581 ;
```

```
25F4 FE1B   04590          CP      27              ;Need extra space if
25F6 3F     04600          CCF                     ;Not halfway
25F7 3E01   04610          LD      A,1             ;Plus 1 more space
25F9 88     04620          ADC     A,B             ;Carry set for 3 spaces
            04630 ;
            04640 ;        Position to ASCII portion of line
            04650 ;
25FA 47     04660          LD      B,A             ;Set loop counter
25FB CD6F26 04670          CALL    WRSPA           ;Output B spaces
25FE C1     04680          POP     BC              ;Recover the counters
            04690 ;
            04700 ;        Calculate # of characters to print
            04710 ;
25FF 3E10   04720          LD      A,16            ;Get # to print
2601 90     04730          SUB     B
2602 47     04740          LD      B,A             ;Xfer to B for DJNZ
2603 217527 04750          LD      HL,LINBUF       ;HL => Line buffer
2606 C5     04760          PUSH    BC              ;Save C
2607 0E08   04770          LD      C,8             ;Space after 8
            04780 ;
            04790 ;        Display ASCII part of line
            04800 ;
2609 7E     04810 PRTLIN1  LD      A,(HL)          ;P/u character
260A 23     04820          INC     HL              ;Bump ptr
260B CD3B26 04830          CALL    CVTDOT          ;Output each char
260E 0D     04840          DEC     C               ;Space yet ?
260F CC7526 04850          CALL    Z,WR1SPA        ;Yes - display space
2612 10F5   04860          DJNZ    PRTLIN1         ;Output line
2614 C1     04870          POP     BC              ;Recover C
            04880 ;
            04890 ;        End of Line - Output C/R & check for EOF
            04900 ;
2615 3E0D   04910 PRTLIN2  LD      A,CR            ;Output C/R
2617 CD5826 04920          CALL    PUTOUT
261A F1     04930          POP     AF              ;Recover @GET ret code
261B 3E1C   04940          LD      A,1CH           ;Init to EOF
261D C26825 04950          JP      NZ,GOTERR       ;End of file - Abort
            04960 ;
2620 CD8626 04970          CALL    CKPAWS          ;<PAUSE> or <BREAK> ?
            04980 ;
            04990 ;        Are we done with the Record ?
            05000 ;
2623 79     05010          LD      A,C             ;P/u # of bytes left
2624 B7     05020          OR      A               ;Finished ?
2625 C2A025 05030          JP      NZ,DISBYTE      ;No - get next line
            05040 ;
            05050 ;        Finished with record - Output space & C/R
            05060 ;
2628 3E20   05070          LD      A,' '           ;Space
262A CD5826 05080          CALL    PUTOUT
262D 3E0D   05090          LD      A,CR            ;Carriage Return
262F CD5826 05100          CALL    PUTOUT
            05110 ;
            05120 ;        Increment Line Number
            05130 ;
2632 212227 05140          LD      HL,VARCLN       ;HL => Line Number
2635 CDE826 05150          CALL    INCNUM          ;Increment
2638 C38A25 05160          JP      DOHEX           ;Loop for more
            05170 ;
            05180 ;        CVTDOT - Output chars & convert non-printables
            05190 ;
```

```
263B FE20      05200 CVTDOT  CP      ' '             ;Don't print controls
263D 3804      05210         JR      C,CVTDOT1
263F FE7F      05220         CP      7FH             ;Print X'20' thru X'7E'
2641 3815      05230         JR      C,PUTOUT
2643 3E2E      05240 CVTDOT1 LD      A,'.'           ;Otherwise change to "."
2645 1811      05250         JR      PUTOUT
               05260 ;
               05270 ;       CVTHEX - Convert A to Hex ASCII & Output it
               05280 ;
2647 F5        05290 CVTHEX  PUSH    AF              ;Save Lower digit
2648 0F        05300         RRCA                    ;Get most sig nibble
2649 0F        05310         RRCA
264A 0F        05320         RRCA
264B 0F        05330         RRCA
264C CD5026    05340         CALL    CVTH1           ;Output upper nibble
264F F1        05350         POP     AF              ;Recover #
               05360 ;
2650 E60F      05370 CVTH1   AND     0FH             ;Mask off upper nibble
2652 C690      05380         ADD     A,90H           ;Convert to Hex digit
2654 27        05390         DAA
2655 CE40      05400         ADC     A,40H
2657 27        05410         DAA                     ;Fall into Output byte
               05420 ;
               05430 ;       PUTOUT - Put out a byte to *DO/*PR
               05440 ;
2658 C5        05450 PUTOUT  PUSH    BC              ;Save BC
2659 4F        05460         LD      C,A             ;Xfer char to C
               05470 ;
               05480 ;       Output byte to *DO or *PR
               05490 ;
265A 3E02      05500 PUTOUT1 LD      A,@DSP          ;@DSP or @PRT SVC #
265C EF        05510         RST     28H
265D C21724    05520         JP      NZ,IOERR        ;NZ - I/O Error
               05530 ;
               05540 ;       Increment byte counter
               05550 ;
2660 E5        05560         PUSH    HL              ;Save HL
2661 217427    05570         LD      HL,BYTCTR       ;HL => Byte counter
2664 34        05580         INC     (HL)            ;Bump counter
2665 79        05590         LD      A,C             ;P/u byte
2666 FE0D      05600         CP      CR              ;End of line ?
2668 2002      05610         JR      NZ,NOTCR        ;No - rest regs & RETurn
               05620 ;
266A 3600      05630         LD      (HL),0          ;Reset byte counter
266C E1        05640 NOTCR   POP     HL              ;Restore registers
266D C1        05650         POP     BC
266E C9        05660         RET                     ;  & RETurn
               05670 ;
               05680 ;
               05690 ;       Output B spaces to Display or Printer
               05700 ;
266F CD7526    05710 WRSPA   CALL    WR1SPA          ;Write a space
2672 10FB      05720         DJNZ    WRSPA           ;Do it B times
2674 C9        05730         RET                     ;RETurn
               05740 ;
               05750 ;       Output a space to Display or Printer
               05760 ;
2675 3E20      05770 WR1SPA  LD      A,' '           ;Space Character
2677 18DF      05780         JR      PUTOUT          ;Output byte
               05790 ;
               05800 ;       PUTLINE - Output a line to the video or printer
```

```
                    Ø581Ø ;       HL => Line of data to output
                    Ø582Ø ;
2679 7E             Ø583Ø PUTLINE LD      A,(HL)          ;P/u byte
267A 23             Ø584Ø         INC     HL              ;Prepare for next
267B FEØ3           Ø585Ø         CP      ETX             ;Check if done none
267D C8             Ø586Ø         RET     Z               ;  return if so
267E CD5826         Ø587Ø         CALL    PUTOUT          ;Char OK output it
2681 FEØD           Ø588Ø         CP      CR              ;Check for CR
2683 C8             Ø589Ø         RET     Z               ;  return if so
2684 18F3           Ø59ØØ         JR      PUTLINE
                    Ø591Ø ;
                    Ø592Ø ;       CKPAWS - Check for Pause (SHIFT @)
                    Ø593Ø ;
2686                Ø594Ø CKPAWS  @@FLAGS                 ;IY => System flags
2686 3E65           ØØØ34         LD      A,1Ø1
2688 EF             ØØØ35         RST     4Ø
                    Ø595Ø ;
                    Ø596Ø ;       Was the <BREAK> key pressed ?
                    Ø597Ø ;
2689                Ø598Ø         @@CKBRKC                ;<BREAK> hit ?
2689 3E6A           ØØØ36         LD      A,1Ø6
268B EF             ØØØ37         RST     4Ø
268C 2Ø1C           Ø599Ø         JR      NZ,GOTBRK       ;Quit if so
                    Ø6ØØØ ;
                    Ø6Ø1Ø ;       Was the <SHIFT><@> pressed ?
                    Ø6Ø2Ø ;
268E FDCBØA4E       Ø6Ø3Ø         BIT     1,(IY+KFLAG$)   ;Is the <PAUSE> bit set ?
2692 C8             Ø6Ø4Ø         RET     Z               ;Return if not <SHIFT><@>
                    Ø6Ø5Ø ;
                    Ø6Ø6Ø ;       Pause - Wait for key to continue
                    Ø6Ø7Ø ;
2693                Ø6Ø8Ø CKWAIT  @@KEY                   ;Wait for key press
2693 3EØ1           ØØØ38         LD      A,1
2695 EF             ØØØ39         RST     4Ø
2696 FE6Ø           Ø6Ø9Ø CKWAIT1 CP      6ØH             ;Was key a <SHIFT @>?
2698 28F9           Ø61ØØ         JR      Z,CKWAIT        ;Ignore if it was
269A FE8Ø           Ø611Ø         CP      8ØH             ;Was key a Break?
269C 28ØC           Ø612Ø         JR      Z,GOTBRK        ;Quit if so
                    Ø613Ø ;
                    Ø614Ø ;       Reset <PAUSE> & <ENTER> bits & RETurn
                    Ø615Ø ;
269E                Ø616Ø RESKFL  @@FLAGS                 ;IY => Flag Table
269E 3E65           ØØØ4Ø         LD      A,1Ø1
26AØ EF             ØØØ41         RST     4Ø
26A1 FD7EØA         Ø617Ø         LD      A,(IY+KFLAG$)   ;P/u KFLAG$
26A4 E6F9           Ø618Ø         AND     ØF9H            ;Reset Pause and Enter
26A6 FD77ØA         Ø619Ø         LD      (IY+KFLAG$),A   ;Stuff into KFLAG$
26A9 C9             Ø62ØØ         RET
                    Ø621Ø ;
                    Ø622Ø ;       <BREAK> hit - Display C/R & Abort
                    Ø623Ø ;
26AA 3EØD           Ø624Ø GOTBRK  LD      A,CR            ;Send end of line
26AC CD5826         Ø625Ø         CALL    PUTOUT          ;Output byte
26AF C32824         Ø626Ø         JP      ABORT           ;  and abort due to BREAK
                    Ø627Ø ;
                    Ø628Ø ;       CVTDEC - Convert HL to Decimal & stuff in BC
                    Ø629Ø ;
26B2 111Ø27         Ø63ØØ CVTDEC  LD      DE,1ØØØØ        ;Divide by 1ØØØØ
26B5 CDCD26         Ø631Ø         CALL    CVD1
26B8 11E8Ø3         Ø632Ø         LD      DE,1ØØØ         ;Divide by 1ØØØ
26BB CDCD26         Ø633Ø         CALL    CVD1
```

```
26BE 116400    06340          LD      DE,100          ;Divide by 100
26C1 CDCD26    06350          CALL    CVD1
26C4 110A00    06360          LD      DE,10           ;Divide by 10
26C7 CDCD26    06370          CALL    CVD1
26CA 110100    06380          LD      DE,1            ;Divide by 1
               06390 ;
               06400 ;        Divide Quotient in HL by value in DE
               06410 ;
26CD AF        06420 CVD1     XOR     A               ;Clear carry set A=0
26CE ED52      06430 CVD2     SBC     HL,DE           ;Subtract Divisor
26D0 3803      06440          JR      C,CVD3          ;Done - add back divisor
26D2 3C        06450          INC     A               ;Bump counter
26D3 18F9      06460          JR      CVD2            ;Subtract until a Carry
               06470 ;
               06480 ;        Add divisor to neg rem & cvrt A to ASCII
               06490 ;
26D5 19        06500 CVD3     ADD     HL,DE           ;HL = New quotient
26D6 C630      06510          ADD     A,'0'           ;A = ASCII numeric digit
26D8 FE30      06520          CP      '0'             ;Zero ?
26DA 2009      06530          JR      NZ,CVD4         ;No - stuff in buff (BC)
               06540 ;
               06550 ;        Char is a Zero - use space if leading zero
               06560 ;
26DC 0B        06570          DEC     BC              ;Backspace buff ptr
26DD 0A        06580          LD      A,(BC)          ;P/u last char
26DE 03        06590          INC     BC              ;Bump to current
26DF FE20      06600          CP      ' '             ;Last char a space ?
26E1 2802      06610          JR      Z,CVD4          ;Yes - don't use lead 0
26E3 3E30      06620          LD      A,'0'           ;No - use zero
               06630 ;
               06640 ;        Stuff Numeric ASCII character into buffer
               06650 ;
26E5 02        06660 CVD4     LD      (BC),A          ;Stuff char in buff
26E6 03        06670          INC     BC              ;Bump ptr
26E7 C9        06680          RET                     ;RETurn
               06690 ;
               06700 ;        INCNUM - Increment Line number in buffer (HL)
               06710 ;
26E8 23        06720 INCNUM   INC     HL              ;Point to lo-order digit
26E9 23        06730          INC     HL
26EA 23        06740          INC     HL
26EB 23        06750          INC     HL
               06760 ;
               06770 ;        Loop to Increment digit and return if done
               06780 ;
26EC 7E        06790 INCNUM1  LD      A,(HL)          ;P/u digit
26ED F630      06800          OR      '0'             ;Start with possible 0
26EF 3C        06810          INC     A               ;Add 1
26F0 77        06820          LD      (HL),A          ;Restuff
26F1 D63A      06830          SUB     '9'+1           ;See if went to 10
26F3 D8        06840          RET     C               ;Ret if not
26F4 3630      06850          LD      (HL),'0'        ;Make it 0
26F6 2B        06860          DEC     HL              ;B/u one
26F7 18F3      06870          JR      INCNUM1         ;  & loop
               06880 ;
               06890 ;        OPEN - Open a file
               06900 ;
26F9           06910 OPEN     @@FLAGS                 ;IY => System Flags
26F9 3E65      00042          LD      A,101
26FB EF        00043          RST     40
26FC FDCB12C6  06920          SET     0,(IY+SFLAG$)   ;P/u SFLAG$
```

```
2700 210028   06930          LD      HL,IOBUFF       ;HL => I/O buffer
2703          06940          @@OPEN                  ;Open the file
2703 3E3B     00044          LD      A,59
2705 EF       00045          RST     40
2706 C9       06950          RET                     ;  & RETurn with condition
              06960 ;
2707 46       06970 SPCREQ$ DB      'File spec required',CR
     69 6C 65 20 73 70 65 63
     20 72 65 71 75 69 72 65
     64 0D
              06980 ;
              06990 ;
271A 20       07000 VARDOT  DB      '     . ',ETX
     20 20 20 20 2E 20 03
2722 20       07010 VARCLN  DB      '     :'
     20 20 20 20 3A
2728 58       07020 VAREQU  DB      'XX = · ',ETX
     58 20 3D 20 20 03
272F 54       07030 TXTEXT  DB      'TXT'
     58 54
              07040 ;
              07050 ;
              07060 ;       PARAMETER TABLE
              07070 ;
2732 80       07080 PRMTBL$ DB      80H
              07090 ;
              07100 ;        ASCII8 (A8) - Flag input only
              07110 ;
2733 46       07120          DB      FLAG!6
2734 41       07130          DB      'ASCII8'
     53 43 49 49 38
273A 00       07140          DB      0
273B 2425     07150          DW      A8PARM+1
              07160 ;
273D 42       07170          DB      FLAG!2
273E 41       07180          DB      'A8'
     38
2740 00       07190          DB      0
2741 2425     07200          DW      A8PARM+1
              07210 ;
              07220 ;        LINE - Numeric input only
              07230 ;
2743 84       07240          DB      NUM!4
2744 4C       07250          DB      'LINE'
     49 4E 45
2748 00       07260          DB      0
2749 E124     07270          DW      LINPRM+1
              07280 ;
              07290 ;        NUM (N) - Flag input only
              07300 ;
274B 53       07310          DB      FLAG!ABB!3
274C 4E       07320          DB      'NUM'
     55 4D
274F 00       07330          DB      0
2750 0F25     07340          DW      NPARM+1
              07350 ;
              07360 ;        HEX (H) - Flag input only
              07370 ;
2752 53       07380          DB      FLAG!ABB!3
2753 48       07390          DB      'HEX'
     45 58
```

```
2756 ØØ          Ø74ØØ           DB      Ø
2757 D624        Ø741Ø           DW      HPARM+1
                 Ø742Ø ;
                 Ø743Ø ;         REC (R) - Numeric input only
                 Ø744Ø ;
2759 93          Ø745Ø           DB      NUM!ABB!3
275A 52          Ø746Ø           DB      'REC'
     45 43
275D ØØ          Ø747Ø           DB      Ø
275E 7F25        Ø748Ø           DW      RPARM+1
                 Ø749Ø ;
                 Ø75ØØ ;         LRL (L) - Numeric input only
                 Ø751Ø ;
276Ø 93          Ø752Ø           DB      NUM!ABB!3
2761 4C          Ø753Ø           DB      'LRL'
     52 4C
2764 ØØ          Ø754Ø LRESP     DB      Ø
2765 9E25        Ø755Ø           DW      LPARM+1
                 Ø756Ø ;
                 Ø757Ø ;         P - Flag input only
                 Ø758Ø ;
2767 41          Ø759Ø           DB      FLAG!1
2768 5Ø          Ø76ØØ           DB      'P'
2769 ØØ          Ø761Ø           DB      Ø
276A CA24        Ø762Ø           DW      PPARM+1
                 Ø763Ø ;
                 Ø764Ø ;         TAB (T) - Flag input only
                 Ø765Ø ;
276C 53          Ø766Ø           DB      FLAG!ABB!3
276D 54          Ø767Ø           DB      'TAB'
     41 42
277Ø ØØ          Ø768Ø TRESP     DB      Ø
2771 3425        Ø769Ø           DW      TPARM+1
                 Ø77ØØ ;
2773 ØØ          Ø771Ø           DB      Ø
                 Ø772Ø ;
                 Ø773Ø ;Buffer Area
                 Ø774Ø ;
ØØØ1             Ø775Ø BYTCTR    DS      1
ØØ1Ø             Ø776Ø LINBUF    DS      16
ØØ2Ø             Ø777Ø FCB1      DS      32
ØØ2Ø             Ø778Ø FCB2      DS      32
                 Ø779Ø ;
28ØØ             Ø78ØØ           ORG     $<-8+1<+8
Ø1ØØ             Ø781Ø IOBUFF    DS      256
                 Ø782Ø ;
24ØØ             Ø783Ø           END     START
```

@@1	ØØØØ	@@2	ØØØØ	@@3	ØØØØ	@@4	ØØØØ
@DSP	FFFF	@MOD2	ØØØ2	@MOD4	FFFF	@PRT	ØØØ6
A8BIT	252D	A8PARM	2523	ABB	ØØ1Ø	ABORT	2428
AP	ØØ27	BGNLIN	24FA	BREAK	ØØ8Ø	BS	ØØØ8
BYTCTR	2774	CFLAG$	ØØØ2	CKPAWS	2686	CKWAIT	2693
CKWAIT1	2696	CLOOP	253F	CR	ØØØD	CVD1	26CD
CVD2	26CE	CVD3	26D5	CVD4	26E5	CVTDEC	26B2
CVTDOT	263B	CVTDOT1	2643	CVTH1	265Ø	CVTHEX	2647
DFLAG$	ØØØ3	DISBYTE	25AØ	DLOOP	2522	DOHEX	258A
DOHEX1	25A1	DOHEX2	25BC	DOHEX3	25CE	DOHEX4	25CF
DOHEX5	25E9	ETX	ØØØ3	FCB1	2785	FCB2	27A5
FLAG	ØØ4Ø	FND1ST	24EB	FPLP	2431	GETCHR	25Ø3
GETPRM	243D	GOTBRK	26AA	GOTERR	2568	GTBK	2574
HPARM	24D5	IGSPC	2447	INCNUM	26E8	INCNUM1	26EC
INITLRL	248B	IOBUFF	28ØØ	IOERR	2417	KFLAG$	ØØØA
LF	ØØØA	LINBUF	2775	LINPRM	24EØ	LIST	242D
LISTA	24Ø9	LPARM	259D	LRESP	2764	NOTCR	266C
NOTFLG	24BE	NOTTAB	254E	NPARM	25ØE	NUM	ØØ8Ø
OPEN	26F9	PAR_ERR	ØØ2C	PPARM	24C9	PRMTBL$	2732
PRTLIN	25EF	PRTLIN1	26Ø9	PRTLIN2	2615	PUCHAR	252Ø
PUTLINE	2679	PUTOUT	2658	PUTOUT1	265A	RESKFL	269E
RESTPTR	2446	RPARM	257E	SAVESP	241Ø	SFLAG$	ØØ12
SKIPLRL	24A5	SPCREQ	2422	SPCREQ$	27Ø7	START	24ØØ
STR	ØØ2Ø	TAB	ØØØ9	TABEXP	253D	TP1	2545
TPARM	2533	TRESP	277Ø	TXTEXT	272F	VARCLN	2722
VARDOT	271A	VAREQU	2728	VFLAG$	ØØ15	WASTAB	2551
WR1SPA	2675	WRSPA	266F	@@ABORT	AF29	@@ADTSK	AFBC
@@BANK	B4D4	@@BKSP	B1B4	@@BREAK	B4EA	@@CHNIO	AF14
@@CKBRKC	B538	@@CKDRV	BØ1Ø	@@CKEOF	B1C9	@@CKTSK	AFA7
@@CLOSE	B19F	@@CLS	B522	@@CMNDI	AF53	@@CMNDR	AF68
@@CTL	AD78	@@DATE	AEEA	@@DCSTAT	BØ4F	@@DEBUG	AF92
@@DECHEX	B454	@@DIRRD	B3C1	@@DIRWR	B3D6	@@DIV16	B43F
@@DIV8	B42A	@@DODIR	BØ25	@@DSP	AD3C	@@DSPLY	ADDC
@@ERROR	AF7D	@@EXIT	AF3E	@@FEXT	B32E	@@FLAGS	B4BE
@@FNAME	B343	@@FSPEC	B319	@@GATRD	B3AC	@@GATWR	B3EB
@@GET	AD5Ø	@@GTDCB	B36D	@@GTDCT	B358	@@GTMOD	B382
@@HDFMT	BØF7	@@HEX16	B493	@@HEX8	B47E	@@HEXDEC	B469
@@HIGH$	B4A8	@@INIT	B175	@@KBD	ADB4	@@KEY	AD28
@@KEYIN	ADC8	@@KLTSK	AFFB	@@LOAD	B2EF	@@LOC	B1DE
@@LOF	B1F3	@@LOGER	AE13	@@LOGOT	AE28	@@MSG	AE5F
@@MUL16	B415	@@MUL8	B4ØØ	@@OPEN	B18A	@@PARAM	AED5
@@PAUSE	AECØ	@@PEOF	B2Ø8	@@POSN	B21D	@@PRINT	AE74
@@PRT	AD8C	@@PUT	AD64	@@RAMDIR	BØ3A	@@RDSEC	BØCD
@@RDSSC	B397	@@READ	B232	@@REMOV	B16Ø	@@RENAM	B14B
@@REW	B247	@@RMTSK	AFD1	@@RPTSK	AFE6	@@RREAD	B25C

```
@@RSLCT     BØB8 @@RSTOR    BØ79 @@RUN      B3Ø4
@@RWRIT     B271 @@SEEK     BØA3 @@SEEKSC   B286
@@SKIP      B29B @@SLCT     BØ64 @@STEPI    BØ8E
@@TIME      AEFF @@VDCTL    AEAB @@VER      B2BØ
@@VRSEC     BØE2 @@WEOF     B2C5 @@WHERE    ADAØ
@@WRITE     B2DA @@WRSEC    B1ØC @@WRSSC    B121
@@WRTRK     B136
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: LOAD, RUN

Library: SYS6/SYS

ISAM # : 81H, 82H

```
                00100 ;LBLOAD/ASM - LOAD & RUN Commands
0000            00110         TITLE   <LOAD/RUN - LS-DOS 6.2>
                00120 ;
000D            00130 CR      EQU     13
004D            00140 @RUN    EQU     77
0028            00150 RST28   EQU     28H
0000            00160 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00170 ;
2400            00180         ORG     2400H
                00190 ;
                00200 ;       RUN entry point
                00210 ;
2400 C32924     00220 RUN     JP      RUN0            ;RUN entry point
                00230 ;
                00240 ;       LOAD entry point
                00250 ;
2403 CDC924     00260 LOAD    CALL    COMMON          ;Parse parms & filespec
2406 203F       00270         JR      NZ,IOERR
2408 3AFC24     00280         LD      A,(XPARM+1)     ;If not (X), go to it
240B B7         00290         OR      A
240C 200A       00300         JR      NZ,NEEDPR
240E 119425     00310         LD      DE,FCB          ;Pt to fcb
2411            00320         @@LOAD
2411 3E4C       00001         LD      A,76
2413 EF         00002         RST     40
2414 2031       00330         JR      NZ,IOERR        ;Go on error
2416 1844       00340         JR      EXIT            ;  or just exit
                00350 ;
                00360 ;       Need to prompt for the system disk
                00370 ;
2418 CD0F25     00380 NEEDPR  CALL    LOADIT          ;Load the file
241B 202A       00390         JR      NZ,IOERR        ;Exit if error
241D 210000     00400         LD      HL,0            ;Init no error
                00410 ;
                00420 ;       Get the system disk back in system drive
                00430 ;
2420 E5         00440 PMTSYS  PUSH    HL              ;Save cmd line ptr
2421 213A25     00450         LD      HL,PMTSYS$
2424 CD6324     00460         CALL    FLASH
2427 E1         00470         POP     HL              ;Rcvr cmnd line ptr
2428 C9         00480         RET
                00490 ;
                00500 ;       RUN command entry
                00510 ;
2429 CDC924     00520 RUN0    CALL    COMMON          ;Parse common args
242C 2019       00530         JR      NZ,IOERR        ;Go on error
242E 3AFC24     00540         LD      A,(XPARM+1)
2431 B7         00550         OR      A
2432 119425     00560         LD      DE,FCB
2435 2005       00570         JR      NZ,RUN1         ;Prompt if (X)
2437 3E4D       00580         LD      A,@RUN          ;RUN SVC number
2439 C32800     00590         JP      RST28
                00600 ;
243C E5         00610 RUN1    PUSH    HL              ;Save cmnd line ptr
243D 119425     00620         LD      DE,FCB
2440 CD0F25     00630         CALL    LOADIT
2443 E3         00640         EX      (SP),HL         ;Get cmd ptr & save ept
2444 28DA       00650         JR      Z,PMTSYS        ;Run if prog OK or
```

```
2446 E1        ØØ66Ø          POP     HL              ;  pop TRAADR & error
               ØØ67Ø ;
               ØØ68Ø ;        Error handling
               ØØ69Ø ;
2447 FE3F      ØØ7ØØ IOERR    CP      63              ;If extended error
2449 28ØA      ØØ71Ø          JR      Z,EXTERR        ;  handle by @LOGOT
244B 6F        ØØ72Ø          LD      L,A             ;Put error # into HL
244C 26ØØ      ØØ73Ø          LD      H,Ø
244E F6CØ      ØØ74Ø          OR      ØCØH            ;Set short error and ret
245Ø 4F        ØØ75Ø          LD      C,A
2451           ØØ76Ø          @@ERROR
2451 3E1A      ØØØØ3          LD      A,26
2453 EF        ØØØØ4          RST     4Ø
2454 C9        ØØ77Ø          RET
2455           ØØ78Ø EXTERR   @@LOGOT
               ØØØØ5          IFEQ    ØØH,1
               ØØØØ6          LD      HL,
               ØØØØ7          ENDIF
2455 3EØC      ØØØØ8          LD      A,12
2457 EF        ØØØØ9          RST     4Ø
2458 21FFFF    ØØ79Ø          LD      HL,-1
245B C9        ØØ8ØØ          RET
245C 21ØØØØ    ØØ81Ø EXIT     LD      HL,Ø
245F C9        ØØ82Ø          RET
               ØØ83Ø ;
               ØØ84Ø ;        Flash the prompt & await reply
               ØØ85Ø ;
246Ø CDB924    ØØ86Ø FLASHØ   CALL    RESKFLG         ;Reset 3-bit field
2463 Ø1FD41    ØØ87Ø FLASH    LD      BC,16893        ;Delay for 25Ø ms
2466           ØØ88Ø          @@PAUSE
2466 3E1Ø      ØØØ1Ø          LD      A,16
2468 EF        ØØØ11          RST     4Ø
2469 FD7EØA    ØØ89Ø          LD      A,(IY+'K'-'A')
246C E6Ø5      ØØ9ØØ          AND     4!1             ;Wait for no ENTER!BRK
246E 2ØFØ      ØØ91Ø          JR      NZ,FLASHØ
247Ø CDB924    ØØ92Ø          CALL    RESKFLG         ;Reset in case BREAK
2473           ØØ93Ø FLS1     @@DSPLY                 ;Display the message
               ØØØ12          IFEQ    ØØH,1
               ØØØ13          LD      HL,
               ØØØ14          ENDIF
2473 3EØA      ØØØ15          LD      A,1Ø
2475 EF        ØØØ16          RST     4Ø
2476 C24724    ØØ94Ø          JP      NZ,IOERR        ;Abort on error
2479 Ø1ØØ4Ø    ØØ95Ø          LD      BC,4ØØØH
247C CD9Ø24    ØØ96Ø          CALL    FLS2            ;Blink start
247F 2Ø42      ØØ97Ø          JR      NZ,GOTBRK       ;Handle BREAK
2481 ØE1E      ØØ98Ø          LD      C,1EH           ;Cursor erase to EOL
2483 CD2Ø25    ØØ99Ø          CALL    DSP
2486 Ø13333    Ø1ØØØ          LD      BC,3333H        ;Wait
2489 CD9Ø24    Ø1Ø1Ø          CALL    FLS2            ;Wait & ck enter
248C 2Ø35      Ø1Ø2Ø          JR      NZ,GOTBRK       ;Handle BREAK
248E 18E3      Ø1Ø3Ø          JR      FLS1            ;Loop until ENTER
               Ø1Ø4Ø ;
               Ø1Ø5Ø ;        FLS2 - Delay a while & ck on <BREAK/ENTER>
               Ø1Ø6Ø ;
2490           Ø1Ø7Ø FLS2     @@CKBRKC                ;<BREAK> hit ?
2490 3E6A      ØØØ17          LD      A,1Ø6
2492 EF        ØØØ18          RST     4Ø
2493 2808      Ø1Ø8Ø          JR      Z,CKENT         ;No - check <ENTER>
2495 ØE1E      Ø1Ø9Ø          LD      C,1EH           ;Erase Line
2497 CD2Ø25    Ø11ØØ          CALL    DSP             ;Output byte
```

```
249A AF          01110          XOR     A               ;Set NZ
249B 3C          01120          INC     A               ;
249C C9          01130          RET                     ;And RETurn
                 01140 ;
249D FDCB00A56   01150 CKENT    BIT     2,(IY+'K'-'A')  ;Ck ENTER bit
24A1 2006        01160          JR      NZ,FLS4         ;Go on ENTER down
24A3 0B          01170          DEC     BC              ;Count down
24A4 78          01180          LD      A,B
24A5 B1          01190          OR      C
24A6 20E8        01200          JR      NZ,FLS2
24A8 C9          01210          RET                     ;Return with Z-flag
                 01220 ;
                 01230 ;        ENTER condition found
                 01240 ;
24A9 F1          01250 FLS4     POP     AF              ;Pop return code
24AA             01260 FLS5     @@KBD                   ;Clear type ahead buffer
24AA 3E08        00019          LD      A,8
24AC EF          00020          RST     40
24AD 28FB        01270          JR      Z,FLS5          ;Loop if have character
24AF 0E1E        01280          LD      C,1EH           ;Wipe line
24B1 CD2025      01290          CALL    DSP
24B4 0E0E        01300          LD      C,14            ;Cursor on
24B6 CD2025      01310          CALL    DSP
24B9 FD7E0A      01320 RESKFLG  LD      A,(IY+'K'-'A')  ;Reset 3-bit field
24BC E6F8        01330          AND     0F8H
24BE FD770A      01340          LD      (IY+'K'-'A'),A
24C1 AF          01350          XOR     A               ;Set Z-flag
24C2 C9          01360          RET
                 01370 ;
24C3 217825      01380 GOTBRK   LD      HL,STOP$        ;Point to error message
24C6 3E3F        01390          LD      A,63            ;Init extended error
24C8 C9          01400          RET                     ;  & return NZ
                 01410 ;
                 01420 ;        Common initialization routine
                 01430 ;
24C9 118A25      01440 COMMON   LD      DE,PRMTBL       ;Parm of X?
24CC             01450          @@PARAM
24CC 3E11        00021          LD      A,17
24CE EF          00022          RST     40
24CF C0          01460          RET     NZ              ;Ret with error code
24D0             01470 COMM1    @@FLAGS                 ;Get flag table pointer
24D0 3E65        00023          LD      A,101
24D2 EF          00024          RST     40
24D3 7E          01480 COMM1A   LD      A,(HL)          ;Skip past spaces
24D4 FE20        01490          CP      ' '
24D6 2003        01500          JR      NZ,COMM2
24D8 23          01510          INC     HL
24D9 18F8        01520          JR      COMM1A
24DB 119425      01530 COMM2    LD      DE,FCB          ;Get filespec
24DE             01540          @@FSPEC
24DE 3E4E        00025          LD      A,78
24E0 EF          00026          RST     40
24E1 2005        01550          JR      NZ,COMM3        ;Go on error
24E3 1A          01560          LD      A,(DE)          ;Device specs not allowed
24E4 FE2A        01570          CP      '*'
24E6 2007        01580          JR      NZ,COMM4        ;Go if OK
24E8 212725      01590 COMM3    LD      HL,SPCREQ$      ;Point to error message
24EB 3E3F        01600          LD      A,63            ;Init extended error
24ED B7          01610          OR      A               ;Set NZ condition
24EE C9          01620          RET
                 01630 ;
```

```
24EF E5          Ø164Ø COMM4   PUSH    HL              ;Save cmdline ptr
24FØ 219125      Ø165Ø         LD      HL,CMDEXT       ;Default to CMD
24F3             Ø166Ø         @@FEXT
24F3 3E4F        ØØØ27         LD      A,79
24F5 EF          ØØØ28         RST     4Ø
24F6 CDØC25      Ø167Ø         CALL    GOSYS2          ;Get SYS2 for open
24F9 E1          Ø168Ø         POP     HL              ;Pop the INBUF$ pointer
24FA CØ          Ø169Ø         RET     NZ
24FB 11ØØØØ      Ø17ØØ XPARM   LD      DE,Ø            ;Ck on X parm
24FE 7A          Ø171Ø         LD      A,D
24FF B3          Ø172Ø         OR      E
25ØØ C8          Ø173Ø         RET     Z               ;Back on no (X)
25Ø1 E5          Ø174Ø         PUSH    HL              ;Save pointer
25Ø2 215925      Ø175Ø         LD      HL,PMTSRC$      ;Init prompt
25Ø5 CD6324      Ø176Ø         CALL    FLASH           ;Prompt for source disk
25Ø8 D1          Ø177Ø         POP     DE              ;Pointer to DE
25Ø9 CØ          Ø178Ø         RET     NZ              ;Back on error in HL
25ØA EB          Ø179Ø         EX      DE,HL           ;If no error, pointer
25ØB C9          Ø18ØØ         RET                     ;  back to HL
                 Ø181Ø ;
                 Ø182Ø ;       Call SYS2 for open routine
                 Ø183Ø ;
25ØC 3E84        Ø184Ø GOSYS2  LD      A,84H           ;Load sys2
25ØE EF          Ø185Ø         RST     28H
                 Ø186Ø ;
                 Ø187Ø ;       Loading routine
                 Ø188Ø ;
25ØF 119425      Ø189Ø LOADIT  LD      DE,FCB
2512 FDCB12D6    Ø19ØØ         SET     2,(IY+'S'-'A')  ;Turn on RUN flag
2516             Ø191Ø         @@LOAD                  ;Load the file
2516 3E4C        ØØØ29         LD      A,76
2518 EF          ØØØ3Ø         RST     4Ø
2519 C8          Ø192Ø         RET     Z
251A F5          Ø193Ø         PUSH    AF              ;Save error ret code
251B CD2Ø24      Ø194Ø         CALL    PMTSYS          ;Get system disk back
251E F1          Ø195Ø         POP     AF              ;Rcvr error ret code
251F C9          Ø196Ø         RET
                 Ø197Ø ;
2520             Ø198Ø DSP     @@DSP                   ;Display byte
2520 3E02        ØØ31          LD      A,2
2522 EF          ØØ32          RST     4Ø
2523 C8          Ø199Ø         RET     Z               ;Return if OK
2524 C34724      Ø2ØØØ         JP      IOERR
                 Ø2Ø1Ø ;
                 Ø2Ø2Ø ;
2527 46          Ø2Ø3Ø SPCREQ$ DB      'File spec required',CR
     69 6C 65 2Ø 73 7Ø 65 63
     2Ø 72 65 71 75 69 72 65
     64 ØD
253A ØF          Ø2Ø4Ø PMTSYS$ DB      15,29,3Ø,'Insert SYSTEM disk <ENTER>',29,3
     1D 1E 49 6E 73 65 72 74
     2Ø 53 59 53 54 45 4D 2Ø
     64 69 73 6B 2Ø 3C 45 4E
     54 45 52 3E 1D Ø3
2559 ØF          Ø2Ø5Ø PMTSRC$ DB      15,29,3Ø,'Insert SOURCE disk <ENTER>',29,3
     1D 1E 49 6E 73 65 72 74
     2Ø 53 4F 55 52 43 45 2Ø
     64 69 73 6B 2Ø 3C 45 4E
     54 45 52 3E 1D Ø3
2578 ØE          Ø2Ø6Ø STOP$   DB      14,29,'Command aborted',CR
     1D 43 6F 6D 6D 61 6E 64
```

```
     2Ø 61 62 6F 72 74 65 64
     ØD
                Ø2Ø7Ø ;
258A 8Ø         Ø2Ø8Ø PRMTBL   DB       8ØH,41H,'X',Ø
     41 58 ØØ
258E FC24       Ø2Ø9Ø          DW       XPARM+1
259Ø ØØ         Ø21ØØ          NOP
                Ø211Ø ;
2591 43         Ø212Ø CMDEXT   DB       'CMD'
     4D 44
2594 ØØ         Ø213Ø FCB      DB       Ø
ØØ1F            Ø214Ø          DS       31
                Ø215Ø ;
24Ø3            Ø216Ø          END      LOAD
```

```
@@1             ØØØØ @@2             ØØØØ @@3             ØØØØ
@@4             ØØØØ @MOD2           ØØØØ @MOD4           FFFF
@RUN            ØØ4D CKENT           249D CMDEXT          2591
COMM1           24DØ COMM1A          24D3 COMM2           24DB
COMM3           24E8 COMM4           24EF COMMON          24C9
CR              ØØØD DSP             252Ø EXIT            245C
EXTERR          2455 FCB             2594 FLASH           2463
FLASHØ          246Ø FLS1            2473 FLS2            249Ø
FLS4            24A9 FLS5            24AA GOSYS2          25ØC
GOTBRK          24C3 IOERR           2447 LOAD            24Ø3
LOADIT          25ØF NEEDPR          2418 PMTSRC$         2559
PMTSYS          242Ø PMTSYS$         253A PRMTBL          258A
RESKFLG         24B9 RST28           ØØ28 RUN             24ØØ
RUNØ            2429 RUN1            243C SPCREQ$         2527
STOP$           2578 XPARM           24FB @@ABORT         7D1F
@@ADTSK         7DB2 @@BANK          82CA @@BKSP          7FAA
@@BREAK         82EØ @@CHNIO         7DØA @@CKBRKC        832E
@@CKDRV         7EØ6 @@CKEOF         7FBF @@CKTSK         7D9D
@@CLOSE         7F95 @@CLS           8318 @@CMNDI         7D49
@@CMNDR         7D5E @@CTL           7B6E @@DATE          7CEØ
@@DCSTAT        7E45 @@DEBUG         7D88 @@DECHEX        824A
@@DIRRD         81B7 @@DIRWR         81CC @@DIV16         8235
@@DIV8          822Ø @@DODIR         7E1B @@DSP           7B32
@@DSPLY         7BD2 @@ERROR         7D73 @@EXIT          7D34
@@FEXT          8124 @@FLAGS         82B4 @@FNAME         8139
@@FSPEC         81ØF @@GATRD         81A2 @@GATWR         81E1
@@GET           7B46 @@GTDCB         8163 @@GTDCT         814E
@@GTMOD         8178 @@HDFMT         7EED @@HEX16         8289
@@HEX8          8274 @@HEXDEC        825F @@HIGH$         829E
@@INIT          7F6B @@KBD           7BAA @@KEY           7B1E
@@KEYIN         7BBE @@KLTSK         7DF1 @@LOAD          8ØE5
@@LOC           7FD4 @@LOF           7FE9 @@LOGER         7CØ9
@@LOGOT         7C1E @@MSG           7C55 @@MUL16         82ØB
@@MUL8          81F6 @@OPEN          7F8Ø @@PARAM         7CCB
@@PAUSE         7CB6 @@PEOF          7FFE @@POSN          8Ø13
@@PRINT         7C6A @@PRT           7B82 @@PUT           7B5A
@@RAMDIR        7E3Ø @@RDSEC         7EC3 @@RDSSC         818D
@@READ          8Ø28 @@REMOV         7F56 @@RENAM         7F41
@@REW           8Ø3D @@RMTSK         7DC7 @@RPTSK         7DDC
@@RREAD         8Ø52 @@RSLCT         7EAE @@RSTOR         7E6F
@@RUN           8ØFA @@RWRIT         8Ø67 @@SEEK          7E99
@@SEEKSC        8Ø7C @@SKIP          8Ø91 @@SLCT          7E5A
@@STEPI         7E84 @@TIME          7CF5 @@VDCTL         7CA1
@@VER           8ØA6 @@VRSEC         7ED8 @@WEOF          8ØBB
@@WHERE         7B96 @@WRITE         8ØDØ @@WRSEC         7FØ2
@@WRSSC         7F17 @@WRTRK         7F2C
24Ø3 is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: MEMORY

Library: SYS6/SYS

ISAM # : 1EH

```
                00100 ;LBMEMORY/ASM - MEMORY Command
0000            00110         TITLE   <MEMORY - LS-DOS 6.2>
                00120 ;
0000            00130 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
0000            00140 *GET    VALUES:3                ;Misc. equates
                03920 ;VALUES/ASM - Version 6
                03930 *LIST OFF
                04200 *LIST ON
                00150 ;
002C            00160 PAR_ERR EQU     44              ;Parameter Error
2600            00170 LOWEST  EQU     2600H
                00180 ;
2400            00190         ORG     2400H
                00200 ;
2400            00210 MEMORY  @@CKBRKC                ;Break key down?
2400 3E6A       00001         LD      A,106
2402 EF         00002         RST     40
2403 2804       00220         JR      Z,BEGINA        ;Ok if not
2405 21FFFF     00230         LD      HL,-1           ;  else abort
2408 C9         00240         RET
                00250 ;
2409 ED73F725   00260 BEGINA  LD      (EXIT+1),SP     ;Save SP Address
                00270 ;
                00280 ;       Process any Parameters entered
                00290 ;
240D 112326     00300         LD      DE,PRMTBL$      ;DE => Parameter Table
2410 D5         00310         PUSH    DE              ;Save Parm table start
2411            00320         @@PARAM                 ;Process parameters
2411 3E11       00003         LD      A,17
2413 EF         00004         RST     40
2414 E1         00330         POP     HL              ;HL => Parm table start
2415 C2B525     00340 PERR    JP      NZ,PRMERR       ;NZ - Parameter Error
                00350 ;
                00360 ;       Legal input - were the entries acceptable ?
                00370 ;
2418 CDC325     00380 GDPARMS CALL    CKPARM          ;Valid entries ?
241B 20F8       00390         JR      NZ,PERR         ;No - Parameter error
                00400 ;
241D            00410         @@FLAGS                 ;IY => System Flags
241D 3E65       00005         LD      A,101
241F EF         00006         RST     40
                00420 ;
                00430 ;       Was the CLEAR (C) parameter entered ?
                00440 ;
2420 0100FF     00450 CPARM   LD      BC,0FF00H       ;C = Clear byte
2423 3A5626     00460         LD      A,(CRESP)       ;P/u response
2426 B7         00470         OR      A               ;Any response ?
2427 283C       00480         JR      Z,NEXTPRM       ;No - get next parm
2429 FDCB0246   00490         BIT     0,(IY+CFLAG$)   ;If memory frozen,
242D C2E925     00500         JP      NZ,NOMEM        ;  can't do it
                00510 ;
                00520 ;       CLEAR (C) parm entered - is this a flag ?
                00530 ;
2430 CB77       00540         BIT     6,A             ;Is this a flag ?
2432 2805       00550         JR      Z,ISNUMER       ;No - check if numeric
                00560 ;
                00570 ;       Response is a FLAG - Is it NO, OFF, or N ?
                00580 ;
```

```
2434 04       00590          INC     B               ;Yes or No ?
2435 202E     00600          JR      NZ,NEXTPRM      ;No - get next parm
2437 180F     00610          JR      FILLMEM         ;Yes - Fill mem with 0's
              00620 ;
              00630 ;        Response is not a FLAG, check if numeric
              00640 ;
2439 215E24   00650 ISNUMER  LD      HL,FILLBYT+1    ;HL => Byte to Fill
243C CB7F     00660          BIT     7,A             ;Numeric response ?
243E 71       00670          LD      (HL),C          ;Stuff byte in LD (HL),nn
243F 2007     00680          JR      NZ,FILLMEM      ;Fill mem with char
              00690 ;
              00700 ;        Response must be a string - Is length = 1 ?
              00710 ;
2441 E60F     00720          AND     00001111B       ;P/u length
2443 3D       00730          DEC     A               ;Better be one
2444 20CF     00740          JR      NZ,PERR         ;Not - parameter error
              00750 ;
              00760 ;        Pick up character at address & stuff away
              00770 ;
2446 0A       00780          LD      A,(BC)          ;P/u character to fill
2447 77       00790          LD      (HL),A          ;Stuff in LD (HL),nn
              00800 ;
              00810 ;        Set HL => HIGH$ (if DOS) or LOW$ (@CMNDR)
              00820 ;
2448 CDA025   00830 FILLMEM  CALL    GETHILO         ;HIGH$ (HL), LOW$ (DE)
244B FDCB024E 00840          BIT     1,(IY+CFLAG$)   ;Executing @CMNDR ?
244F 2801     00850          JR      Z,USEHIGH       ;No - use HIGH$
2451 EB       00860          EX      DE,HL           ;@CMNDR - use LOW$
2452 110026   00870 USEHIGH  LD      DE,MEMORY+200H  ;Start clearing here
              00880 ;
              00890 ;        Calculate amount of memory to fill
              00900 ;
2455 AF       00910          XOR     A               ;Clear carry
2456 ED52     00920          SBC     HL,DE           ;Get # to fill
2458 44       00930          LD      B,H             ;Xfer to BC
2459 4D       00940          LD      C,L
              00950 ;
              00960 ;        Fill user area - HIGH$/LOW$ with spec'd byte
              00970 ;
245A 62       00980          LD      H,D             ;HL => LOW$
245B 6B       00990          LD      L,E
245C 13       01000          INC     DE              ;DE => LOW$ + 1
245D 3600     01010 FILLBYT  LD      (HL),$-$        ;Stuff in fill byte
245F EDB0     01020          LDIR                    ;Fill memory
2461 210000   01030 GOODEX   LD      HL,0            ;Good Exit
2464 C9       01040          RET
              01050 ;
              01060 ;        Was the ADDR (A) parameter specified ?
              01070 ;
2465 3A3726   01080 NEXTPRM  LD      A,(ARESP)       ;P/u Address response
2468 B7       01090          OR      A               ;Specified ?
2469 CA0C25   01100          JP      Z,HICHECK       ;No - get next parm
              01110 ;
              01120 ;         Check for NUMERIC entry
              01130 ;
246C CB7F     01140          BIT     7,A             ;Check NUM bit
246E 201C     01150          JR      NZ,APARM        ;  go if set
              01160 ;
              01170 ;        Response must be a string - Is length = 1 ?
              01180 ;
2470 E60F     01190          AND     00001111B       ;P/u length
```

```
2472 3D       01200          DEC     A               ;Better be one
2473 20A0     01210          JR      NZ,PERR         ;Not - parameter error
              01220 ;
              01230 ;        P/u character representation of 'FLAG'
              01240 ;
2475 2A8D24   01250          LD      HL,(APARM+1)    ;P/u pointer to char
2478 7E       01260          LD      A,(HL)          ;P/u char
2479 CBAF     01270          RES     5,A             ;  and make UC
247B D641     01280          SUB     41H             ;Normalize for flag#
247D FE1A     01290          CP      26              ;Check on range
247F D2ED25   01300          JP      NC,RANGER       ;Error if too high
2482 FDE5     01310          PUSH    IY
2484 E1       01320          POP     HL              ;P/u base of flags
2485 1600     01330          LD      D,0
2487 5F       01340          LD      E,A             ;Put offset in DE
2488 19       01350          ADD     HL,DE           ;Add them together
2489 228D24   01360          LD      (APARM+1),HL    ;Set value in response word
              01370 ;
              01380 ;        P/u addr, cvrt to Hex ASCII & put in string
              01390 ;
248C 110000   01400 APARM    LD      DE,$-$          ;P/u the address
248F 217A26   01410          LD      HL,HEXADD       ;HL => Destination
2492          01420          @@HEX16                 ;Cvrt DE to ASCII @ HL
2492 3E63     00007          LD      A,99
2494 EF       00008          RST     40
              01430 ;
              01440 ;        Convert DE to Decimal ASCII & put in string
              01450 ;
2495 EB       01460          EX      DE,HL           ;Set HL = Address
2496 E5       01470          PUSH    HL              ;Save addr ptr
2497 118226   01480          LD      DE,DECADD       ;DE => Destination
249A          01490          @@HEXDEC                ;Cvrt HL to ASCII @ DE
249A 3E61     00009          LD      A,97
249C EF       00010          RST     40
249D E1       01500          POP     HL              ;HL = Address
              01510 ;
              01520 ;        P/u word & byte at that address
              01530 ;
249E E5       01540          PUSH    HL              ;Save address ptr
249F 4E       01550          LD      C,(HL)          ;P/u byte
24A0 51       01560          LD      D,C             ;P/u the word value
24A1 23       01570          INC     HL              ;  & stuff into the
24A2 5E       01580          LD      E,(HL)          ;  message in hex
              01590 ;
              01600 ;        Convert Byte to Hex ASCII & stuff in string
              01610 ;
24A3 21A026   01620          LD      HL,OLDBYTE      ;HL => Destination
24A6          01630          @@HEX8                  ;Convert C to ASCII @ HL
24A6 3E62     00011          LD      A,98
24A8 EF       00012          RST     40
              01640 ;
              01650 ;        Convert Word to Hex ASCII & stuff in string
              01660 ;
24A9 218C26   01670          LD      HL,OLDWORD      ;HL => Destination
24AC          01680          @@HEX16                 ;Cvt DE to ASCII @ HL
24AC 3E63     00013          LD      A,99
24AE EF       00014          RST     40
24AF DDE1     01690          POP     IX              ;Recover Address ptr
              01700 ;
              01710 ;        Was WORD or BYTE parameter entered ?
              01720 ;
```

```
24B1 214726   01730         LD      HL,BRESP        ;HL => Byte Response
24B4 113F26   01740         LD      DE,WRESP        ;DE => Word Response
24B7 1A       01750         LD      A,(DE)          ;Anything entered ?
24B8 B6       01760         OR      (HL)
24B9 2010     01770         JR      NZ,WHICH        ;Yes - which one
              01780 ;
              01790 ;       Neither Entered - Modify string
              01800 ;
24BB 219126   01810         LD      HL,OLDWORD+5    ;End string
24BE 3629     01820         LD      (HL),')'        ;Don't display new word
24C0 23       01830         INC     HL
24C1 3620     01840         LD      (HL),' '
24C3 23       01850         INC     HL
24C4 3620     01860         LD      (HL),' '
24C6 23       01870         INC     HL
24C7 3603     01880         LD      (HL),ETX
24C9 182C     01890         JR      DSPSTR          ;Display string
              01900 ;
              01910 ;       One or Both was entered - ensure both not
              01920 ;
24CB 1A       01930 WHICH   LD      A,(DE)          ;Word entered ?
24CC B7       01940         OR      A
24CD 2816     01950         JR      Z,BPARM         ;No - get byte
              01960 ;
              01970 ;       Word Entered - Make sure byte wasn't entered
              01980 ;
24CF 7E       01990         LD      A,(HL)          ;Entered ?
24D0 B7       02000         OR      A
24D1 C2B525   02010         JP      NZ,PRMERR       ;Yes - parameter error
              02020 ;
              02030 ;       Pick up Word Value & Stuff into Memory
              02040 ;
24D4 110000   02050 WPARM   LD      DE,$-$          ;P/u word
24D7 DD7200   02060         LD      (IX),D          ;Stuff lsb
24DA DD7301   02070         LD      (IX+1),E        ;Stuff msb
              02080 ;
              02090 ;       Cvt Word/Byte to Hex ASCII & put in string
              02100 ;
24DD 219726   02110 STUFVAL LD      HL,NEWWORD      ;HL => Destination
24E0          02120         @@HEX16                 ;Cvrt DE to hex in (HL)
24E0 3E63     00015         LD      A,99
24E2 EF       00016         RST     40
24E3 1812     02130         JR      DSPSTR          ;Display String
              02140 ;
              02150 ;       Stuff byte into mem if between 0-255
              02160 ;
24E5 11FF00   02170 BPARM   LD      DE,00FFH        ;P/u byte
24E8 7A       02180         LD      A,D             ;Hi-order must = 0
24E9 B7       02190         OR      A
24EA C2B525   02200         JP      NZ,PRMERR       ;No - Parm error
24ED DD7300   02210         LD      (IX),E          ;Stuff LSB into string
              02220 ;
              02230 ;       Convert byte to Hex ASCII & stuff in string
              02240 ;
24F0 4B       02250         LD      C,E             ;Set C = new byte
24F1 21A926   02260         LD      HL,NEWBYTE      ;HL => New byte
24F4          02270         @@HEX8                  ;Cvt C to ASCII @ HL
24F4 3E62     00017         LD      A,98
24F6 EF       00018         RST     40
              02280 ;
              02290 ;       Display address string
```

```
                    Ø23ØØ ;
24F7 217826         Ø231Ø DSPSTR  LD      HL,ADDMSG       ;HL => Message
24FA CDBØ25         Ø232Ø         CALL    DSPLY           ;Display it
                    Ø233Ø ;
                    Ø234Ø ;       Display Word/Byte string
                    Ø235Ø ;
24FD 218C26         Ø236Ø         LD      HL,OLDWORD      ;HL => "Word string"
25ØØ 3A4726         Ø237Ø         LD      A,(BRESP)       ;Byte response
25Ø3 B7             Ø238Ø         OR      A               ;Specified
25Ø4 28Ø3           Ø239Ø         JR      Z,DSPSTR2       ;No - display Word string
25Ø6 21AØ26         Ø24ØØ         LD      HL,OLDBYTE      ;Yes - use Byte string
25Ø9 CDBØ25         Ø241Ø DSPSTR2 CALL    DSPLY           ;Display string
                    Ø242Ø ;
                    Ø243Ø ;       HIGH$ & LOW$ check - Was HIGH$ entered ?
                    Ø244Ø ;
25ØC CDAØ25         Ø245Ø HICHECK CALL    GETHILO         ;HIGH$ (HL), LOW$ (DE)
25ØF 22E125         Ø246Ø         LD      (OLDHI),HL      ;Put old HIGH$ in header
2512 3A2926         Ø247Ø         LD      A,(HRESP)       ;HIGH$ = value ?
2515 B7             Ø248Ø         OR      A
2516 281E           Ø249Ø         JR      Z,LOCHECK       ;No - check LOW$
2518 FDCBØ246       Ø25ØØ         BIT     Ø,(IY+CFLAG$)   ;If memory frozen,
251C C2E925         Ø251Ø         JP      NZ,NOMEM        ;  can't do it
                    Ø252Ø ;
                    Ø253Ø ;       HIGH$ entered, p/u value & check range
                    Ø254Ø ;
251F Ø1ØØØØ         Ø255Ø HPARM   LD      BC,$-$          ;P/u requested new HIGH$
2522 ED42           Ø256Ø         SBC     HL,BC           ;New HIGH$ > old HIGH$ ?
2524 DAED25         Ø257Ø         JP      C,RANGER        ;Yes - Range error
                    Ø258Ø ;
                    Ø259Ø ;       Create Header String & establish true HIGH$
                    Ø26ØØ ;
2527 D5             Ø261Ø         PUSH    DE              ;Save DE
2528 5Ø             Ø262Ø         LD      D,B             ;Set DE = HIGH$ + 1
2529 59             Ø263Ø         LD      E,C
252A 13             Ø264Ø         INC     DE
252B 21E525         Ø265Ø         LD      HL,HD_ADD       ;Convert DE to Hex ASCII
252E                Ø266Ø         @@HEX16                 ;  at HL (BC is preserved)
252E 3E63           ØØØ19         LD      A,99
253Ø EF             ØØØ2Ø         RST     4Ø
2531 21F6FF         Ø267Ø         LD      HL,-HLEN        ;Set HL = actual HIGH$
2534 Ø9             Ø268Ø         ADD     HL,BC           ;  including header size
2535 D1             Ø269Ø         POP     DE              ;Restore LOW$
                    Ø27ØØ ;
                    Ø271Ø ;       Was LOW$ entered ?
                    Ø272Ø ;
2536 3A3Ø26         Ø273Ø LOCHECK LD      A,(LRESP)       ;P/u response
2539 B7             Ø274Ø         OR      A               ;Entered ?
253A 2816           Ø275Ø         JR      Z,CHKBOTH       ;No - check range
253C FDCBØ246       Ø276Ø         BIT     Ø,(IY+CFLAG$)   ;If memory frozen,
254Ø C2E925         Ø277Ø         JP      NZ,NOMEM        ;  can't do it
                    Ø278Ø ;
                    Ø279Ø ;       LOW$ entered - cannot be below LOWEST
                    Ø28ØØ ;
2543 11ØØØØ         Ø281Ø LPARM   LD      DE,$-$          ;P/u value entered
2546 Ø1ØØ26         Ø282Ø         LD      BC,LOWEST       ;BC = lowest possible
2549 EB             Ø283Ø         EX      DE,HL           ;  memory location
254A E5             Ø284Ø         PUSH    HL
254B ED42           Ø285Ø         SBC     HL,BC           ;In range ?
254D E1             Ø286Ø         POP     HL
254E EB             Ø287Ø         EX      DE,HL
254F DAED25         Ø288Ø         JP      C,RANGER        ;No - display range error
```

```
                    Ø289Ø ;
                    Ø29ØØ ;       HL = HIGH$, DE = LOW$ - do they overlap ?
                    Ø291Ø ;
2552 E5             Ø292Ø CHKBOTH PUSH    HL              ;HIGH$ must be
2553 B7             Ø293Ø         OR      A               ;  greater than or equal
2554 ED52           Ø294Ø         SBC     HL,DE           ;  to LOW$.
2556 E1             Ø295Ø         POP     HL
2557 DAED25         Ø296Ø         JP      C,RANGER        ;Less - range error
                    Ø297Ø ;
                    Ø298Ø ;       LOW$ and HIGH$ are both valid - set LOW$
                    Ø299Ø ;
255A EB             Ø3ØØØ         EX      DE,HL           ;Pt DE => HI$, HL => LOW$
255B Ø6Ø1           Ø3Ø1Ø         LD      B,1
255D                Ø3Ø2Ø         @@HIGH$                 ;Set LOW$
255D 3E64           ØØØ21         LD      A,1ØØ
255F EF             ØØØ22         RST     4Ø
                    Ø3Ø3Ø ;
                    Ø3Ø4Ø ;       Was the HIGH parameter specified ?
                    Ø3Ø5Ø ;
256Ø 3A2926         Ø3Ø6Ø         LD      A,(HRESP)       ;Was HIGH specified ?
2563 B7             Ø3Ø7Ø         OR      A
2564 2813           Ø3Ø8Ø         JR      Z,DSPHI         ;No - don't alter it
                    Ø3Ø9Ø ;
                    Ø31ØØ ;       Yes - Change Exit message to include header
                    Ø311Ø ;
2566 3EØA           Ø312Ø         LD      A,LF            ;Change C/R to a L/F
2568 32BØ26         Ø313Ø         LD      (HEADMES),A
                    Ø314Ø ;
                    Ø315Ø ;       Xfer header into high memory
                    Ø316Ø ;
256B D5             Ø317Ø         PUSH    DE              ;Save HIGH$ ptr
256C 13             Ø318Ø         INC     DE              ;Pt to header destination
256D 21DF25         Ø319Ø         LD      HL,HEADER       ;HL => Header
257Ø Ø1ØAØØ         Ø32ØØ         LD      BC,HLEN         ;BC = header length
2573 EDBØ           Ø321Ø         LDIR                    ;Xfer to mem
2575 E1             Ø322Ø         POP     HL              ;HL => HIGH$ (BC = Ø)
2576                Ø323Ø         @@HIGH$                 ;Set HIGH$ (func. Ø)
2576 3E64           ØØØ23         LD      A,1ØØ
2578 EF             ØØØ24         RST     4Ø
                    Ø324Ø ;
                    Ø325Ø ;       P/u HIGH$/LOW$, cvt to Hex, & put in string
                    Ø326Ø ;
2579 CDAØ25         Ø327Ø DSPHI   CALL    GETHILO         ;HIGH$ (HL), LOW$ (DE)
257C E5             Ø328Ø         PUSH    HL              ;Save HIGH$ ptr
257D 217226         Ø329Ø         LD      HL,LOWIS1       ;HL => Destination
258Ø                Ø33ØØ         @@HEX16                 ;Cvt DE to ASCII @ HL
258Ø 3E63           ØØØ25         LD      A,99
2582 EF             ØØØ26         RST     4Ø
2583 D1             Ø331Ø         POP     DE              ;DE = HIGH$
2584 216326         Ø332Ø         LD      HL,HIGHIS1      ;HL => Destination
2587                Ø333Ø         @@HEX16                 ;Cvt DE to ASCII @ HL
2587 3E63           ØØØ27         LD      A,99
2589 EF             ØØØ28         RST     4Ø
                    Ø334Ø ;
                    Ø335Ø ;       Display HIGH$ = nnnn & LOW$ = nnnn string
                    Ø336Ø ;
258A 215A26         Ø337Ø         LD      HL,HIGHIS       ;HL => HIGH$/LOW$ string
258D                Ø338Ø         @@LOGOT                 ;Log & display
                    ØØØ29         IFEQ    ØØH,1
                    ØØØ3Ø         LD      HL,
                    ØØØ31         ENDIF
```

```
258D 3E0C     00032          LD      A,12
258F EF       00033          RST     40
2590 21B026   03390          LD      HL,HEADMES      ;Display Header message
2593          03400          @@LOGOT                 ;  if one was inserted
              00034          IFEQ    00H,1
              00035          LD      HL,
              00036          ENDIF
2593 3E0C     00037          LD      A,12
2595 EF       00038          RST     40
              03410 ;
              03420 ;        Was a Go Parameter Entered ?
              03430 ;
2596 3A4D26   03440 GO       LD      A,(GRESP)       ;GO entered ?
2599 B7       03450          OR      A
259A CA6124   03460          JP      Z,GOODEX        ;No - RETurn HL = 0
259D C30000   03470 GPARM    JP      $-$             ;Yes - Jump to address
              03480 ;
              03490 ;        GETHILO - Get HIGH$ in HL, & LOW$ in DE
              03500 ;
25A0 210000   03510 GETHILO  LD      HL,0            ;P/u LOW$
25A3 54       03520          LD      D,H             ;Set DE = 0
25A4 5D       03530          LD      E,L
25A5 0601     03540          LD      B,1
25A7          03550          @@HIGH$
25A7 3E64     00039          LD      A,100
25A9 EF       00040          RST     40
25AA EB       03560          EX      DE,HL           ;Save in DE
25AB 45       03570          LD      B,L             ;Set B = 0
25AC          03580          @@HIGH$                 ;Get HIGH$ & RETurn
25AC 3E64     00041          LD      A,100
25AE EF       00042          RST     40
25AF C9       03590          RET                     ;RETurn
              03600 ;
              03610 ;        DSPLY - Display a line to the video
              03620 ;
25B0          03630 DSPLY    @@DSPLY                 ;Display line
              00043          IFEQ    00H,1
              00044          LD      HL,
              00045          ENDIF
25B0 3E0A     00046          LD      A,10
25B2 EF       00047          RST     40
25B3 C8       03640          RET     Z               ;Z - RETurn
25B4 21       03650          DB      21H             ;Skip LD A,## instruction
              03660 ;
              03670 ;        IOERR - Fatal Error Handler
              03680 ;
25B5 3E2C     03690 PRMERR   LD      A,PAR_ERR       ;Parameter Error
25B7 6F       03700 IOERR    LD      L,A             ;Error # to HL
25B8 2600     03710          LD      H,0
25BA F6C0     03720          OR      0C0H            ;Short error message
25BC 4F       03730          LD      C,A             ;Save in C
25BD          03740          @@ERROR                 ;Display error
25BD 3E1A     00048          LD      A,26
25BF EF       00049          RST     40
25C0 1834     03750          JR      EXIT            ;Go to exit routine
              03760 ;
              03770 ;        CKPARM - Check if Parameter types are legal
              03780 ;        HL => Beginning of Parameter Table
              03790 ;        Z  <= Set if Parameters entered were legal
              03800 ;
25C2 D1       03810 GOODPRM  POP     DE              ;Clear stack
```

```
25C3 23        Ø382Ø CKPARM  INC     HL              ;Bump past 8ØH
25C4 7E        Ø383Ø         LD      A,(HL)          ;P/u type byte
25C5 47        Ø384Ø         LD      B,A             ;Save in B
25C6 E6ØF      Ø385Ø         AND     ØFH             ;Get length
25C8 C8        Ø386Ø         RET     Z               ;Zero - finished
               Ø387Ø ;
               Ø388Ø ;       Position HL to response byte
               Ø389Ø ;
25C9 23        Ø39ØØ         INC     HL              ;HL => Parameter Name
25CA E5        Ø391Ø         PUSH    HL              ;Save start of name
25CB 5F        Ø392Ø         LD      E,A             ;Set DE = name
25CC 16ØØ      Ø393Ø         LD      D,Ø             ;  length.
25CE 19        Ø394Ø         ADD     HL,DE
               Ø395Ø ;
               Ø396Ø ;       Pick up response, mask off junk, & xfer in D
               Ø397Ø ;
25CF 7E        Ø398Ø         LD      A,(HL)          ;P/u response
25DØ E6EØ      Ø399Ø         AND     ØEØH            ;Bits 7-5 = response
25D2 57        Ø4ØØØ         LD      D,A             ;Save in D
               Ø4Ø1Ø ;
               Ø4Ø2Ø ;       Was the response bit acceptable by type ?
               Ø4Ø3Ø ;
25D3 78        Ø4Ø4Ø         LD      A,B             ;P/u type byte
25D4 A2        Ø4Ø5Ø         AND     D               ;Mask off bits 4-Ø
25D5 AA        Ø4Ø6Ø         XOR     D               ;Result = Z if both set
               Ø4Ø7Ø ;
               Ø4Ø8Ø ;       Position HL to next parameter entry
               Ø4Ø9Ø ;
25D6 23        Ø41ØØ         INC     HL              ;Go past word
25D7 23        Ø411Ø         INC     HL
25D8 28E8      Ø412Ø         JR      Z,GOODPRM       ;Z - good entry
               Ø413Ø ;
               Ø414Ø ;       Illegal Entry - Recover Name start & RETurn
               Ø415Ø ;
25DA 4B        Ø416Ø         LD      C,E             ;Set BC = length
25DB Ø6ØØ      Ø417Ø         LD      B,Ø
25DD E1        Ø418Ø         POP     HL              ;HL => Parameter Name
25DE C9        Ø419Ø         RET                     ;RETurn NZ
               Ø42ØØ ;
               Ø421Ø ;       6.2 Memory Header
               Ø422Ø ;
25DF 18Ø8      Ø423Ø HEADER  JR      MEMSTRT         ;JR to start of module
25E1 ØØØØ      Ø424Ø OLDHI   DW      Ø               ;HIGH$ before this module
25E3 Ø5        Ø425Ø         DB      5,'&'           ;Use "&" to denote addr.
     26
25E5 6E        Ø426Ø HD_ADD  DB      'nnnn'          ;Hex ASCII address
     6E 6E 6E
25E9           Ø427Ø MEMSTRT EQU     $               ;Length byte
ØØØA           Ø428Ø HLEN    EQU     $-HEADER        ;Length of Header
               Ø429Ø ;
               Ø43ØØ ;
               Ø431Ø ;       Messages
               Ø432Ø ;
               Ø433Ø ;
25E9 21FD25    Ø434Ø NOMEM   LD      HL,NOMEM$
25EC DD        Ø435Ø         DB      ØDDH
25ED 211726    Ø436Ø RANGER  LD      HL,RANGER$
               Ø437Ø ;
               Ø438Ø ;       Display & Log message, & RETurn
               Ø439Ø ;
25FØ           Ø44ØØ         @@LOGOT                 ;Display/Log message
```

```
                00050          IFEQ    00H,1
                00051          LD      HL,
                00052          ENDIF
25F0 3E0C       00053          LD      A,12
25F2 EF         00054          RST     40
25F3 21FFFF     04410 ABORT    LD      HL,-1           ;Internal Error
25F6 310000     04420 EXIT     LD      SP,$-$          ;P/u old SP address
25F9            04430          @@CKBRKC                ;Clear any Break
25F9 3E6A       00055          LD      A,106
25FB EF         00056          RST     40
25FC C9         04440          RET                     ;RETurn
                04450 ;
25FD 4E         04460 NOMEM$   DB      'No memory space available',CR
     6F 20 6D 65 6D 6F 72 79
     20 73 70 61 63 65 20 61
     76 61 69 6C 61 62 6C 65
     0D
2617 52         04470 RANGER$ DB       'Range error',CR
     61 6E 67 65 20 65 72 72
     6F 72 0D
                04480 ;
                04490 ;        PARAMETER TABLE
                04500 ;
2623 80         04510 PRMTBL$ DB       80H             ;6.x type @PARAM
                04520 ;
                04530 ;        HIGH (H) - Accept Numeric input only
                04540 ;
2624 94         04550          DB      NUM!ABB!4
2625 48         04560          DB      'HIGH'
     49 47 48
2629 00         04570 HRESP    DB      0
262A 2025       04580          DW      HPARM+1
                04590 ;
                04600 ;        LOW (L) - Accept numeric input only
                04610 ;
262C 93         04620          DB      NUM!ABB!3
262D 4C         04630          DB      'LOW'
     4F 57
2630 00         04640 LRESP    DB      0
2631 4425       04650          DW      LPARM+1
                04660 ;
                04670 ;        ADD (A) - Accept numeric, string input
                04680 ;
2633 B3         04690          DB      NUM!STR!ABB!3
2634 41         04700          DB      'ADD'
     44 44
2637 00         04710 ARESP    DB      0
2638 8D24       04720          DW      APARM+1
                04730 ;
                04740 ;        WORD (W) - Accept Numeric input only
                04750 ;
263A 94         04760          DB      NUM!ABB!4
263B 57         04770          DB      'WORD'
     4F 52 44
263F 00         04780 WRESP    DB      0
2640 D524       04790          DW      WPARM+1
                04800 ;
                04810 ;        BYTE (B) - Accept Numeric input only
                04820 ;
2642 94         04830          DB      NUM!ABB!4
2643 42         04840          DB      'BYTE'
```

```
     59 54 45
2647 ØØ        Ø485Ø BRESP   DB      Ø
2648 E624      Ø486Ø         DW      BPARM+1
               Ø487Ø ;
               Ø488Ø ;       GO (G) - Accept Numeric input only
               Ø489Ø ;
264A 92        Ø49ØØ         DB      NUM!ABB!2
264B 47        Ø491Ø         DB      'GO'
     4F
264D ØØ        Ø492Ø GRESP   DB      Ø
264E 9E25      Ø493Ø         DW      GPARM+1
               Ø494Ø ;
               Ø495Ø ;       CLEAR (C) - Accept Flag, Numeric or string input
               Ø496Ø ;
265Ø F5        Ø497Ø         DB      FLAG!NUM!STR!ABB!5
2651 43        Ø498Ø         DB      'CLEAR'
     4C 45 41 52
2656 ØØ        Ø499Ø CRESP   DB      Ø
2657 2124      Ø5ØØØ         DW      CPARM+1
               Ø5Ø1Ø ;
2659 ØØ        Ø5Ø2Ø         NOP
               Ø5Ø3Ø ;
265A 48        Ø5Ø4Ø HIGHIS  DB      'High = X',AP
     69 67 68 2Ø 3D 2Ø 58 27
2663 78        Ø5Ø5Ø HIGHIS1 DB      'xxxx',AP,'  Low = X',AP
     78 78 78 27 2Ø 2Ø 4C 6F
     77 2Ø 3D 2Ø 58 27
2672 78        Ø5Ø6Ø LOWIS1  DB      'xxxx',AP,ETX
     78 78 78 27 Ø3
               Ø5Ø7Ø ;
2678 58        Ø5Ø8Ø ADDMSG  DB      'X',AP
     27
267A 6E        Ø5Ø9Ø HEXADD  DB      'nnnn',AP,' = '
     6E 6E 6E 27 2Ø 3D 2Ø
2682 64        Ø51ØØ DECADD  DB      'ddddd (X',AP,ETX
     64 64 64 64 2Ø 28 58 27
     Ø3
               Ø511Ø ;
268C 6E        Ø512Ø OLDWORD DB      'nnnn',AP,' => X',AP
     6E 6E 6E 27 2Ø 3D 3E 2Ø
     58 27
2697 6E        Ø513Ø NEWWORD DB      'nnnn',AP,')  ',ETX
     6E 6E 6E 27 29 2Ø 2Ø Ø3
               Ø514Ø ;
26AØ 6E        Ø515Ø OLDBYTE DB      'nn',AP,' => X',AP
     6E 27 2Ø 3D 3E 2Ø 58 27
26A9 6E        Ø516Ø NEWBYTE DB      'nn',AP,')  ',ETX
     6E 27 29 2Ø 2Ø Ø3
               Ø517Ø ;
26BØ ØD        Ø518Ø HEADMES DB      CR,'Note : Memory Header Inserted',CR
     4E 6F 74 65 2Ø 3A 2Ø 4D
     65 6D 6F 72 79 2Ø 48 65
     61 64 65 72 2Ø 49 6E 73
     65 72 74 65 64 ØD
               Ø519Ø ;
24ØØ           Ø52ØØ         END     MEMORY
```

@@1	ØØØØ	@@2	ØØØØ	@@3	ØØØØ		
@@4	ØØØØ	@MOD2	ØØØØ	@MOD4	FFFF		
ABB	ØØ1Ø	ABORT	25F3	ADDMSG	2678		
AP	ØØ27	APARM	248C	ARESP	2637		
BEGINA	24Ø9	BPARM	24E5	BREAK	ØØ8Ø		
BRESP	2647	BS	ØØØ8	CFLAG$	ØØØ2		
CHKBOTH	2552	CKPARM	25C3	CPARM	242Ø		
CR	ØØØD	CRESP	2656	DECADD	2682		
DFLAG$	ØØØ3	DSPHI	2579	DSPLY	25BØ		
DSPSTR	24F7	DSPSTR2	25Ø9	ETX	ØØØ3		
EXIT	25F6	FILLBYT	245D	FILLMEM	2448		
FLAG	ØØ4Ø	GDPARMS	2418	GETHILO	25AØ		
GO	2596	GOODEX	2461	GOODPRM	25C2		
GPARM	259D	GRESP	264D	HD_ADD	25E5		
HEADER	25DF	HEADMES	26BØ	HEXADD	267A		
HICHECK	25ØC	HIGHIS	265A	HIGHIS1	2663		
HLEN	ØØØA	HPARM	251F	HRESP	2629		
IOERR	25B7	ISNUMER	2439	KFLAG$	ØØØA		
LF	ØØØA	LOCHECK	2536	LOWEST	26ØØ		
LOWIS1	2672	LPARM	2543	LRESP	263Ø		
MEMORY	24ØØ	MEMSTRT	25E9	NEWBYTE	26A9		
NEWWORD	2697	NEXTPRM	2465	NOMEM	25E9		
NOMEM$	25FD	NUM	ØØ8Ø	OLDBYTE	26AØ		
OLDHI	25E1	OLDWORD	268C	PAR_ERR	ØØ2C		
PERR	2415	PRMERR	25B5	PRMTBL$	2623		
RANGER	25ED	RANGER$	2617	SFLAG$	ØØ12		
STR	ØØ2Ø	STUFVAL	24DD	TAB	ØØØ9		
USEHIGH	2452	VFLAG$	ØØ15	WHICH	24CB		
WPARM	24D4	WRESP	263F	@@ABORT	99FB		
@@ADTSK	9A8E	@@BANK	9FA6	@@BKSP	9C86		
@@BREAK	9FBC	@@CHNIO	99E6	@@CKBRKC	AØØA		
@@CKDRV	9AE2	@@CKEOF	9C9B	@@CKTSK	9A79		
@@CLOSE	9C71	@@CLS	9FF4	@@CMNDI	9A25		
@@CMNDR	9A3A	@@CTL	984A	@@DATE	99BC		
@@DCSTAT	9B21	@@DEBUG	9A64	@@DECHEX	9F26		
@@DIRRD	9E93	@@DIRWR	9EA8	@@DIV16	9F11		
@@DIV8	9EFC	@@DODIR	9AF7	@@DSP	98ØE		
@@DSPLY	98AE	@@ERROR	9A4F	@@EXIT	9A1Ø		
@@FEXT	9EØØ	@@FLAGS	9F9Ø	@@FNAME	9E15		
@@FSPEC	9DEB	@@GATRD	9E7E	@@GATWR	9EBD		
@@GET	9822	@@GTDCB	9E3F	@@GTDCT	9E2A		
@@GTMOD	9E54	@@HDFMT	9BC9	@@HEX16	9F65		
@@HEX8	9F5Ø	@@HEXDEC	9F3B	@@HIGH$	9F7A		
@@INIT	9C47	@@KBD	9886	@@KEY	97FA		
@@KEYIN	989A	@@KLTSK	9ACD	@@LOAD	9DC1		
@@LOC	9CBØ	@@LOF	9CC5	@@LOGER	98E5		
@@LOGOT	98FA	@@MSG	9931	@@MUL16	9EE7		
@@MUL8	9ED2	@@OPEN	9C5C	@@PARAM	99A7		
@@PAUSE	9992	@@PEOF	9CDA	@@POSN	9CEF		
@@PRINT	9946	@@PRT	985E	@@PUT	9836		
@@RAMDIR	9BØC	@@RDSEC	9B9F	@@RDSSC	9E69		
@@READ	9DØ4	@@REMOV	9C32	@@RENAM	9C1D		
@@REW	9D19	@@RMTSK	9AA3	@@RPTSK	9AB8		
@@RREAD	9D2E	@@RSLCT	9B8A	@@RSTOR	9B4B		
@@RUN	9DD6	@@RWRIT	9D43	@@SEEK	9B75		
@@SEEKSC	9D58	@@SKIP	9D6D	@@SLCT	9B36		
@@STEPI	9B6Ø	@@TIME	99D1	@@VDCTL	997D		
@@VER	9D82	@@VRSEC	9BB4	@@WEOF	9D97		
@@WHERE	9872	@@WRITE	9DAC	@@WRSEC	9BDE		
@@WRSSC	9BF3	@@WRTRK	9CØ8				

24ØØ is the transfer address
ØØØØØ Total errors

NOTES:

Command: PURGE

Library: SYS7/SYS

ISAM # : 72H

```
                00100 ;LBPURGE/ASM - PURGE Command
0000            00110         TITLE   <PURGE - LS-DOS 6.2>
                00120 ;
000A            00130 LF      EQU     10
000D            00140 CR      EQU     13
002C            00150 PAR_ERR EQU     44              ;Parameter Error
42E0            00160 PASSWORD        EQU     42E0H
                00170 ;
0000            00180 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00190 ;
2400            00200         ORG     2400H
                00210 ;
2400            00220 PURGE   @@CKBRKC                ;Break key down?
2400 3E6A       00001         LD      A,106
2402 EF         00002         RST     40
2403 2804       00230         JR      Z,BEGINA        ;Ok if not
2405 21FFFF     00240         LD      HL,-1           ;  else abort
2408 C9         00250         RET
                00260 ;
2409 ED73F327   00270 BEGINA  LD      (SAVESP+1),SP   ;Save stack pointer
240D 7E         00280 PURGE1  LD      A,(HL)          ;Bypass cmd line blanks
240E 23         00290         INC     HL
240F FE20       00300         CP      ' '
2411 28FA       00310         JR      Z,PURGE1
2413 11B828     00320         LD      DE,BLANKS       ;Pt to filespec area
2416 0608       00330         LD      B,8             ;Init for file name
2418 FE2D       00340         CP      '-'             ;If -, set up flag
241A 2005       00350         JR      NZ,PUR0
241C 323E25     00360         LD      (MFLG+1),A
241F 7E         00370         LD      A,(HL)
2420 23         00380         INC     HL
2421 CD2727     00390 PUR0    CALL    PRSPEC
2424 FE2F       00400         CP      '/'             ;Ck on file EXT entered
2426 200A       00410         JR      NZ,PUR1         ;Jump if no extension
2428 11C028     00420         LD      DE,BLANKS+8     ;Point to ext field
242B 0603       00430         LD      B,3             ;Max 3 chars
242D 7E         00440         LD      A,(HL)
242E 23         00450         INC     HL
242F CD2727     00460         CALL    PRSPEC          ;Ck on EXT
2432 FE3A       00470 PUR1    CP      ':'             ;Drive entered?
2434 0E00       00480         LD      C,0             ;Init to drive 0
2436 201A       00490         JR      NZ,PRMERRA      ;Quit if no drive #
2438 7E         00500         LD      A,(HL)          ;P/u drive #
2439 23         00510         INC     HL              ;Bump to next field
243A CD2F29     00520         CALL    PATCH1          ;Ck drive range 0-7
243D 79         00530 PUR2    LD      A,C             ;Xfer drive to regA
243E 327C26     00540         LD      (TSTMPW+1),A    ;  & stuff for later
2441            00550         @@CKDRV                 ;Ck if drive available
2441 3E21       00003         LD      A,33
2443 EF         00004         RST     40
2444 3E20       00560         LD      A,32            ;"drive not avail...
2446 CD3829     00570         CALL    PATCH2          ;Ck WP or missing disk
2449            00580         @@GTDCT                 ;DCT to reg IY
2449 3E51       00005         LD      A,81
244B EF         00006         RST     40
244C 11E728     00590         LD      DE,PRMTBL$      ;Get parms
244F            00600         @@PARAM
244F 3E11       00007         LD      A,17
```

```
2451 EF       00008          RST     40
2452 C2DB27   00610 PRMERRA  JP      NZ,PRMERR       ;Jump on error
2455 210000   00620 DATPRM   LD      HL,0            ;P/u date="from-to"
2458 7C       00630          LD      A,H
2459 B5       00640          OR      L
245A 2835     00650          JR      Z,PUR3          ;Bypass if not entered
245C 7E       00660          LD      A,(HL)          ;Check for "-to"
245D FE2D     00670          CP      '-'
245F 2818     00680          JR      Z,CKTO
2461 3E80     00690          LD      A,80H           ;Set from bit
2463 32C328   00700          LD      (FTFLG),A       ;Note from entered
2466 CD5727   00710          CALL    PAKDAT          ;Pack the date entry
2469 C2DD27   00720          JP      NZ,IOERR        ;Quit if bad date
246C ED43C428 00730          LD      (FMPAKD),BC
2470 7E       00740          LD      A,(HL)
2471 FE22     00750          CP      '"'
2473 2810     00760          JR      Z,FRCTO
2475 FE2D     00770          CP      '-'             ;Check for "-to"
2477 2018     00780          JR      NZ,PUR3
2479 23       00790 CKTO     INC     HL              ;Bypass the '-'
247A 7E       00800          LD      A,(HL)          ;Ck for end of parm
247B FE22     00810          CP      '"'
247D 2812     00820          JR      Z,PUR3          ;Go on parm end
247F CD5727   00830          CALL    PAKDAT          ;  else pack the date
2482 C2DD27   00840          JP      NZ,IOERR        ;Quit on bad date
2485 3AC328   00850 FRCTO    LD      A,(FTFLG)
2488 F601     00860          OR      1               ;Set TO bit
248A 32C328   00870          LD      (FTFLG),A
248D ED43C628 00880          LD      (TOPAKD),BC     ;Stuff for later
2491 3AB025   00890 PUR3     LD      A,(QPARM+1)     ;Query parm used?
2494 B7       00900          OR      A
2495 2806     00910          JR      Z,DOEVER        ;Go if not
2497 CD1527   00920          CALL    CKINDO          ;Invalid command during
249A C2DD27   00930          JP      NZ,IOERR        ;  <DO> processing
249D CD7B26   00940 DOEVER   CALL    TSTMPW          ;Ck on master password
24A0 C2DD27   00950          JP      NZ,IOERR        ;Go if worng
24A3 3A7C26   00960          LD      A,(TSTMPW+1)    ;P/u drive
24A6 4F       00970          LD      C,A
24A7 FD5609   00980          LD      D,(IY+9)        ;Get DIR cylinder
24AA 1E01     00990          LD      E,1             ;Pt to HIT sector
24AC 21002C   01000          LD      HL,HITBUF
24AF          01010          @@FLAGS                 ;Pt IY => Flags
24AF 3E65     00009          LD      A,101
24B1 EF       00010          RST     40
24B2          01020          @@RDSSC                 ;Read the HIT
24B2 3E55     00011          LD      A,85
24B4 EF       00012          RST     40
24B5 3E16     01030          LD      A,16H           ;Init "HIT read error...
24B7 C2DD27   01040          JP      NZ,IOERR        ;Abort on read error
24BA 1818     01050          JR      SCNH3
              01060 ;
              01070 ;        Major loop to scan HIT for files
              01080 ;
24BC E1       01090 SCNHIT   POP     HL
24BD C1       01100 SCNH1    POP     BC              ;Rcvr HIT ptr DEC
24BE 262C     01110          LD      H,HITBUF<-8     ;Pt to hi-order buffer
24C0 68       01120          LD      L,B             ;Set lo-order DEC
24C1 7D       01130 SCNH2    LD      A,L
24C2 C620     01140          ADD     A,32            ;Pt to next one in
24C4 6F       01150          LD      L,A             ;Same dir sector
24C5 300D     01160          JR      NC,SCNH3        ;Jump if still in sector
```

```
24C7 2C       Ø117Ø          INC     L               ;Bump to next dir sector
24C8 FE1F     Ø118Ø          CP      1FH             ;End of the line?
24CA 2ØØ8     Ø119Ø          JR      NZ,SCNH3        ;Loop if not
24CC ØEØD     Ø12ØØ          LD      C,CR
24CE          Ø121Ø          @@DSP                   ;Write new line & exit
24CE 3EØ2     ØØØ13          LD      A,2
24DØ EF       ØØØ14          RST     4Ø
24D1 C3F827   Ø122Ø          JP      EXIT
              Ø123Ø ;
              Ø124Ø ;        Routine to check on dir record in use
              Ø125Ø ;
24D4 7D       Ø126Ø SCNH3    LD      A,L             ;Ignore BOOT & DIR
24D5 E6FE     Ø127Ø          AND     ØFEH
24D7 28E8     Ø128Ø          JR      Z,SCNH2
24D9 7E       Ø129Ø          LD      A,(HL)          ;P/u HIT hash byte
24DA B7       Ø13ØØ          OR      A
24DB 28E4     Ø131Ø          JR      Z,SCNH2         ;Ignore if spare
24DD 45       Ø132Ø          LD      B,L             ;Save DEC
24DE C5       Ø133Ø          PUSH    BC
24DF 7D       Ø134Ø          LD      A,L             ;Get record # in L
24EØ E6EØ     Ø135Ø          AND     ØEØH
24E2 6F       Ø136Ø          LD      L,A
24E3 A8       Ø137Ø          XOR     B               ;Get sector # in A
24E4 FEFF     Ø138Ø SCNH3A   CP      ØFFH            ;Same as what's in core?
24E6 28ØD     Ø139Ø          JR      Z,SCNH4         ;Bypass if same
24E8 32E524   Ø14ØØ          LD      (SCNH3A+1),A    ;Update indicator byte
24EB          Ø141Ø          @@DIRRD                 ;Read this directory
24EB 3E57     ØØØ15          LD      A,87
24ED EF       ØØØ16          RST     4Ø
24EE C2DD27   Ø142Ø          JP      NZ,IOERR        ;Quit on read error
24F1 7C       Ø143Ø          LD      A,H             ;Set SBUFF pointer
24F2 32F624   Ø144Ø          LD      (SCNH4+1),A
24F5 26ØØ     Ø145Ø SCNH4    LD      H,Ø             ;Pt to dir buf hi-order
24F7 7E       Ø146Ø          LD      A,(HL)          ;L set to lo-order
24F8 CB67     Ø147Ø          BIT     4,A             ;Ignore if not assigned
24FA 28C1     Ø148Ø          JR      Z,SCNH1
24FC CB7F     Ø149Ø          BIT     7,A             ;Ignore if it's an
24FE 2ØBD     Ø15ØØ          JR      NZ,SCNH1        ;  extended dir record
25ØØ CB77     Ø151Ø          BIT     6,A             ;Jump if not a SYS file
25Ø2 28ØA     Ø152Ø          JR      Z,CKINV
25Ø4 11ØØØØ   Ø153Ø SPARM    LD      DE,Ø            ;P/u S-parm
25Ø7 7A       Ø154Ø          LD      A,D
25Ø8 B3       Ø155Ø          OR      E               ;Ignore this one if
25Ø9 CABD24   Ø156Ø          JP      Z,SCNH1         ;  S-parm not entered
25ØC 18ØC     Ø157Ø          JR      CKNAM
              Ø158Ø ;
              Ø159Ø ;        Non-SYS file
              Ø16ØØ ;
25ØE CB5F     Ø161Ø CKINV    BIT     3,A             ;Jump if visible
251Ø 28Ø8     Ø162Ø          JR      Z,CKNAM
2512 11ØØØØ   Ø163Ø IPARM    LD      DE,Ø            ;I-parm
2515 7A       Ø164Ø          LD      A,D             ;Ignore if I-parm not
2516 B3       Ø165Ø          OR      E               ;  entered as this file
2517 CABD24   Ø166Ø          JP      Z,SCNH1         ;  is invisible
              Ø167Ø ;
              Ø168Ø ;        Parms match, grab filename & check class
              Ø169Ø ;
251A E5       Ø17ØØ CKNAM    PUSH    HL              ;Save ptr to record
251B 7D       Ø171Ø          LD      A,L             ;Pt to filename in dir
251C C6Ø5     Ø172Ø          ADD     A,5
251E 6F       Ø173Ø          LD      L,A
```

```
251F 11B828   Ø174Ø          LD      DE,BLANKS       ;Pt to parsed input
2522 Ø6ØB     Ø175Ø          LD      B,11            ;Ck name/ext (11-chars)
2524 1A       Ø176Ø SCNH5    LD      A,(DE)
2525 FE24     Ø177Ø          CP      '$'             ;Wild char?
2527 28Ø7     Ø178Ø          JR      Z,SCNH6         ;Always a match
2529 BE       Ø179Ø          CP      (HL)            ;Not global, char match?
252A 28Ø4     Ø18ØØ          JR      Z,SCNH6         ;Ck more if match
252C FE2Ø     Ø181Ø          CP      ' '             ;Blank = end of ck
252E 2ØØD     Ø182Ø          JR      NZ,MFLG         ;If not blank, no match
253Ø 23       Ø183Ø SCNH6    INC     HL              ;Bump pointers
2531 13       Ø184Ø          INC     DE
2532 1ØFØ     Ø185Ø          DJNZ    SCNH5           ;Loop for 11 chars
2534 3A3E25   Ø186Ø          LD      A,(MFLG+1)      ;Bypass if a match but
2537 B7       Ø187Ø          OR      A               ;  - exclude given
2538 C2BC24   Ø188Ø          JP      NZ,SCNHIT
253B 18Ø6     Ø189Ø          JR      SCNH6A
253D 3EØØ     Ø19ØØ MFLG     LD      A,Ø             ;Ignore if no match &
253F B7       Ø191Ø          OR      A               ;  no exclude given
2540 CABC24   Ø192Ø          JP      Z,SCNHIT
2543 E1       Ø193Ø SCNH6A   POP     HL              ;Rcvr ptr to DIR+Ø
2544 E5       Ø194Ø          PUSH    HL
              Ø195Ø ;
              Ø196Ø ;        Now check if date matches
              Ø197Ø ;
2545 23       Ø198Ø          INC     HL              ;Pt to date field
2546 CD4627   Ø199Ø          CALL    UNPACK          ;Alter date for cpr
2549 3AC328   Ø2ØØØ          LD      A,(FTFLG)
254C Ø7       Ø2Ø1Ø          RLCA                    ;Tst fm bit
254D 3Ø1Ø     Ø2Ø2Ø          JR      NC,SCNH6B
254F 7A       Ø2Ø3Ø          LD      A,D             ;Ignore if no date
2550 B3       Ø2Ø4Ø          OR      E               ;  in DIR for file
2551 CABC24   Ø2Ø5Ø          JP      Z,SCNHIT
2554 2AC428   Ø2Ø6Ø          LD      HL,(FMPAKD)     ;P/u user entry
2557 EB       Ø2Ø7Ø          EX      DE,HL
2558 CDD527   Ø2Ø8Ø          CALL    CPHLDE          ;HL-DE
255B EB       Ø2Ø9Ø          EX      DE,HL
255C DABC24   Ø21ØØ          JP      C,SCNHIT        ;Go if out of range
255F 3AC328   Ø211Ø SCNH6B   LD      A,(FTFLG)
2562 ØF       Ø212Ø          RRCA                    ;Tst TO bit
2563 3ØØE     Ø213Ø          JR      NC,MATCHES      ;Go if no TOPARM
2565 7A       Ø214Ø          LD      A,D             ;  else ck if file is dated
2566 B3       Ø215Ø          OR      E
2567 CABC24   Ø216Ø          JP      Z,SCNHIT        ;Go if no dir date
256A 2AC628   Ø217Ø          LD      HL,(TOPAKD)     ;P/u user's packed date
256D CDD527   Ø218Ø          CALL    CPHLDE          ;HL-DE
2570 DABC24   Ø219Ø          JP      C,SCNHIT        ;Go if out of range
2573 E1       Ø22ØØ MATCHES  POP     HL              ;Rcvr pointer to DIRREC
2574 E5       Ø221Ø DONAM    PUSH    HL
2575 7D       Ø222Ø          LD      A,L             ;  & point to file name
2576 C6Ø5     Ø223Ø          ADD     A,5
2578 6F       Ø224Ø          LD      L,A
2579 11CC28   Ø225Ø          LD      DE,FCB1$        ;Pt to name/ext buffer
257C Ø6Ø8     Ø226Ø          LD      B,8             ;Max 8-char name
257E 7E       Ø227Ø DONAM1   LD      A,(HL)          ;Move filename into
257F FE2Ø     Ø228Ø          CP      ' '             ;  buffer until space
2581 28Ø5     Ø229Ø          JR      Z,DONAME2       ;  or 8 characters
2583 12       Ø23ØØ          LD      (DE),A
2584 23       Ø231Ø          INC     HL
2585 13       Ø232Ø          INC     DE
2586 1ØF6     Ø233Ø          DJNZ    DONAM1
2588 7D       Ø234Ø DONAME2  LD      A,L             ;Point to file ext
```

```
2589 8Ø        Ø235Ø           ADD     A,B
258A 6F        Ø236Ø           LD      L,A
258B 7E        Ø237Ø           LD      A,(HL)          ;Is there an extension?
258C FE2Ø      Ø238Ø           CP      ' '
258E 281Ø      Ø239Ø           JR      Z,DONAM5        ;Bypass if not
259Ø 3E2F      Ø24ØØ           LD      A,'/'
2592 12        Ø241Ø           LD      (DE),A          ;Stuff ext separator
2593 13        Ø242Ø           INC     DE
2594 Ø6Ø3      Ø243Ø           LD      B,3             ;Init 3-char ext max
2596 7E        Ø244Ø DONAM4    LD      A,(HL)          ;Transfer up to space
2597 FE2Ø      Ø245Ø           CP      ' '             ;  or 3 chars
2599 28Ø5      Ø246Ø           JR      Z,DONAM5
259B 12        Ø247Ø           LD      (DE),A
259C 23        Ø248Ø           INC     HL
259D 13        Ø249Ø           INC     DE
259E 1ØF6      Ø25ØØ           DJNZ    DONAM4
25AØ 3E3A      Ø251Ø DONAM5    LD      A,':'           ;Add the drivespec
25A2 12        Ø252Ø           LD      (DE),A
25A3 13        Ø253Ø           INC     DE
25A4 3A7C26    Ø254Ø           LD      A,(TSTMPW+1)    ;P/u drivespec
25A7 F63Ø      Ø255Ø           OR      'Ø'             ;Make it ASCII & stuff
25A9 12        Ø256Ø           LD      (DE),A
25AA 13        Ø257Ø           INC     DE
25AB 3EØ3      Ø258Ø           LD      A,3             ;Terminate with ETX
25AD 12        Ø259Ø           LD      (DE),A
25AE D5        Ø26ØØ           PUSH    DE              ;Save pointer
25AF 11FFFF    Ø261Ø QPARM     LD      DE,-1           ;Query each file?
25B2 7A        Ø262Ø           LD      A,D
25B3 B3        Ø263Ø           OR      E
25B4 CA3D26    Ø264Ø           JP      Z,NOPRMPT       ;Not if not Q=N
               Ø265Ø ;
25B7           Ø266Ø           @@DSPLY PRGFIL$         ;"Purge file?...
               ØØØ17           IFEQ    Ø1H,1
25B7 217728    ØØØ18           LD      HL,PRGFIL$
               ØØØ19           ENDIF
25BA 3EØA      ØØØ2Ø           LD      A,1Ø
25BC EF        ØØØ21           RST     4Ø
25BD D1        Ø267Ø           POP     DE              ;Rcvr ptr to file buf ETX
25BE E1        Ø268Ø           POP     HL              ;Rcvr ptr to 1st dir byte
25BF D5        Ø269Ø           PUSH    DE
25CØ 23        Ø27ØØ           INC     HL              ;Pt to MOD bit
25C1 CB76      Ø271Ø           BIT     6,(HL)          ;Test MOD flag
25C3 28Ø8      Ø272Ø           JR      Z,SCDAT1        ;Go if not mod'ed
25C5 3E2Ø      Ø273Ø           LD      A,' '           ;Put a space
25C7 12        Ø274Ø           LD      (DE),A
25C8 13        Ø275Ø           INC     DE
25C9 3E2B      Ø276Ø           LD      A,'+'           ;  and the mod sign
25CB 12        Ø277Ø           LD      (DE),A
25CC 13        Ø278Ø           INC     DE
25CD 3E2Ø      Ø279Ø SCDAT1    LD      A,' '           ;Write a space
25CF 12        Ø28ØØ           LD      (DE),A
25DØ 13        Ø281Ø           INC     DE
25D1 23        Ø282Ø           INC     HL              ;Advance to date field
25D2 EB        Ø283Ø           EX      DE,HL
25D3 367B      Ø284Ø           LD      (HL),'{'        ;Stuff left brace
25D5 23        Ø285Ø           INC     HL
25D6 EB        Ø286Ø           EX      DE,HL
25D7 7E        Ø287Ø           LD      A,(HL)
25D8 B7        Ø288Ø           OR      A
25D9 283D      Ø289Ø           JR      Z,SCDAT4        ;Ignore if no date saved
25DB ØF        Ø29ØØ           RRCA                    ;Has date, get day
```

```
25DC ØF      Ø291Ø          RRCA
25DD ØF      Ø292Ø          RRCA
25DE E61F    Ø293Ø          AND     1FH
25EØ Ø62F    Ø294Ø          LD      B,2FH          ;Convert day to decimal
25E2 Ø4      Ø295Ø SCDAT2   INC     B              ;  by counting # of 1Ø's
25E3 D6ØA    Ø296Ø          SUB     1Ø             ;Sub 1Ø from day #
25E5 3ØFB    Ø297Ø          JR      NC,SCDAT2
25E7 C63A    Ø298Ø          ADD     A,3AH          ;Cvrt lo order to ASCII
25E9 F5      Ø299Ø          PUSH    AF             ;Save day low order
25EA 78      Ø3ØØØ          LD      A,B            ;Stuff day hi order
25EB 12      Ø3Ø1Ø          LD      (DE),A
25EC 13      Ø3Ø2Ø          INC     DE             ;Bump
25ED F1      Ø3Ø3Ø          POP     AF             ;Rcvr lo order day #
25EE 12      Ø3Ø4Ø          LD      (DE),A         ;Stuff low order
25EF 13      Ø3Ø5Ø          INC     DE             ;Bump pointer to msg
25FØ 3E2D    Ø3Ø6Ø          LD      A,'-'          ;Init seperator
25F2 12      Ø3Ø7Ø          LD      (DE),A         ;  and stuff in buffer
25F3 13      Ø3Ø8Ø          INC     DE             ;Pt to month field
25F4 E5      Ø3Ø9Ø          PUSH    HL             ;Save DIR ptr
25F5 2B      Ø31ØØ          DEC     HL             ;Pt to DIR+1 (month+)
25F6 7E      Ø311Ø          LD      A,(HL)         ;P/u month etc
25F7 E6ØF    Ø312Ø          AND     ØFH            ;Strip off flags
25F9 3D      Ø313Ø          DEC     A              ;(mon-1)*3 indexes string
25FA 4F      Ø314Ø          LD      C,A            ;  conversion table
25FB Ø7      Ø315Ø          RLCA
25FC 81      Ø316Ø          ADD     A,C
25FD 4F      Ø317Ø          LD      C,A
25FE Ø6ØØ    Ø318Ø          LD      B,Ø
26ØØ 219428  Ø319Ø          LD      HL,MONTBL
2603 Ø9      Ø32ØØ          ADD     HL,BC          ;Add offset to tbl start
2604 ØE03    Ø321Ø          LD      C,3
2606 EDBØ    Ø322Ø          LDIR                   ;Move 3-char month
2608 3E2D    Ø323Ø          LD      A,'-'          ;Suff separator char
260A 12      Ø324Ø          LD      (DE),A
260B 13      Ø325Ø          INC     DE             ;Advance to year field
260C 3E38    Ø326Ø          LD      A,'8'          ;Stuff 8 of 198Ø
260E 12      Ø327Ø          LD      (DE),A
260F 13      Ø328Ø          INC     DE             ;Bump msg ptr
2610 E1      Ø329Ø          POP     HL             ;Rcvr DIR+2
2611 7E      Ø33ØØ          LD      A,(HL)         ;P/u year field
2612 E6Ø7    Ø331Ø          AND     7              ;Remove day
2614 C63Ø    Ø332Ø          ADD     A,'Ø'          ;Cvrt to ASCII
2616 12      Ø333Ø          LD      (DE),A         ;Stuff -> msg
2617 13      Ø334Ø          INC     DE
2618 3E03    Ø335Ø SCDAT4   LD      A,3            ;Show etx for display
261A 12      Ø336Ø          LD      (DE),A
261B         Ø337Ø          @@DSPLY FCB1$          ;Display filename
             ØØØ22          IFEQ    Ø1H,1
261B 21CC28  ØØØ23          LD      HL,FCB1$
             ØØØ24          ENDIF
261E 3EØA    ØØØ25          LD      A,1Ø
2620 EF      ØØØ26          RST     4Ø
2621         Ø338Ø          @@DSPLY QMARK$         ;Display ???
             ØØØ27          IFEQ    Ø1H,1
2621 218428  ØØØ28          LD      HL,QMARK$
             ØØØ29          ENDIF
2624 3EØA    ØØØ3Ø          LD      A,1Ø
2626 EF      ØØØ31          RST     4Ø
2627 21C828  Ø339Ø          LD      HL,LILBUF$     ;Get response y,n
262A Ø1ØØØ3  Ø34ØØ          LD      BC,3<8         ;For Yes, No
262D         Ø341Ø          @@KEYIN
```

```
262D 3E09      00032          LD      A,9
262F EF        00033          RST     40
2630 DAEC27    03420          JP      C,BREAK         ;Abort on <BREAK>
2633 7E        03430          LD      A,(HL)          ;P/u response
2634 CBAF      03440          RES     5,A             ;Strip l/c if entered
2636 FE59      03450          CP      'Y'             ;Is it yes?
2638 C2BC24    03460          JP      NZ,SCNHIT       ;Bypass if not
263B E3        03470          EX      (SP),HL         ;Place dummy HL below
263C E5        03480          PUSH    HL              ;  pointer
263D FDCB0A46  03490 NOPRMPT  BIT     0,(IY+'K'-'A')  ;Ck if BREAK bit in
2641 C2EC27    03500          JP      NZ,BREAK        ;  KFLAG is active
2644           03510          @@LOGOT PURGE$          ;Dsply "Purging: "
               00034          IFEQ    01H,1
2644 218A28    00035          LD      HL,PURGE$
               00036          ENDIF
2647 3E0C      00037          LD      A,12
2649 EF        00038          RST     40
264A E1        03520          POP     HL              ;Get pointer where ETX is
264B 360D      03530          LD      (HL),CR         ;  & replace with CR
264D           03540          @@LOGOT FCB1$           ;Dsply filename
               00039          IFEQ    01H,1
264D 21CC28    00040          LD      HL,FCB1$
               00041          ENDIF
2650 3E0C      00042          LD      A,12
2652 EF        00043          RST     40
2653 E1        03550          POP     HL              ;Pop dummy or DIRREC ptr
2654 C1        03560          POP     BC              ;Get drive & DEC
2655 C5        03570          PUSH    BC
2656 78        03580          LD      A,B             ;P/u the DEC
2657 321629    03590          LD      (FCB+7),A       ;  & stuff
265A 3A7C26    03600          LD      A,(TSTMPW+1)    ;P/u drive
265D 321529    03610          LD      (FCB+6),A       ;  & stuff
2660 3E01      03620          LD      A,1             ;Set up FCB for remove
2662 321029    03630          LD      (FCB+1),A
2665 3E80      03640          LD      A,80H           ;Show FCB as open
2667 320F29    03650          LD      (FCB),A
266A 110F29    03660          LD      DE,FCB          ;Remove the file
266D           03670          @@REMOV
266D 3E39      00044          LD      A,57
266F EF        00045          RST     40
2670 C2DD27    03680          JP      NZ,IOERR        ;Jump on error
2673 3EFF      03690          LD      A,0FFH          ;Show we don't have the
2675 32E524    03700          LD      (SCNH3A+1),A    ;  latest dir record
2678 C3BD24    03710          JP      SCNH1           ;Loop
               03720 ;
               03730 ;        Routine to get the master password & match it
               03740 ;
267B 0E00      03750 TSTMPW   LD      C,$-$           ;Init to drive requested
267D CD0427    03760          CALL    GATRD           ;Read GAT into GATBUF
2680 C0        03770          RET     NZ              ;Back on error
2681 2ACE2B    03780          LD      HL,(GATBUF+0CEH)
2684 11E042    03790          LD      DE,PASSWORD     ;Password="PASSWORD" ?
2687 AF        03800          XOR     A
2688 ED52      03810          SBC     HL,DE
268A C8        03820          RET     Z               ;Back if PASSWORD
               03830 ;
               03840 ;        MPW is not "PASSWORD" - check entry match
               03850 ;
268B 110000    03860 PWPARM   LD      DE,0            ;P/u MPW string addr
268E 212B28    03870          LD      HL,MPW$         ;Init prompt
2691 CDA226    03880          CALL    GETMPW          ;Hash parm or entry
```

```
2694 C0       03890          RET     NZ
2695 EB       03900          EX      DE,HL           ;Xfer hashed MPW to DE
2696 2ACE2B   03910          LD      HL,(GATBUF+0CEH)        ;Grab pack MPW &
2699 AF       03920          XOR     A               ;  check if user entered
269A ED52     03930          SBC     HL,DE           ;  the pack MPW
269C 214328   03940          LD      HL,BADMPW$      ;Init error pointer
269F 3E3F     03950          LD      A,63            ;Set extended error
26A1 C9       03960          RET                     ;Z or NZ
              03970 ;
              03980 ;        Routine to get 8-char password
              03990 ;
26A2 CDA926   04000 GETMPW   CALL    GMPW1           ;Test if user entered MPW
26A5 C0       04010          RET     NZ
26A6 3EE4     04020          LD      A,0E4H          ;Hash password (DE) to HL
26A8 EF       04030          RST     28H             ;Ret to what called
26A9 7A       04040 GMPW1    LD      A,D             ;Test if user entered MPW
26AA B3       04050          OR      E
26AB 281D     04060          JR      Z,GMPW3         ;Prompt if not
26AD 3C       04070          INC     A               ;  or no operand
26AE 281A     04080          JR      Z,GMPW3
              04090 ;
              04100 ;        Place entered password into buffer
              04110 ;
26B0 21002A   04120          LD      HL,BUFFER
26B3 E5       04130          PUSH    HL
26B4 0608     04140          LD      B,8             ;Max entry of 8 chars
26B6 1A       04150 GMPW2    LD      A,(DE)          ;P/u pswd char
26B7 FE0D     04160          CP      CR              ;End of the line?
26B9 282F     04170          JR      Z,GMPW4         ;Space out if so
26BB FE2C     04180          CP      ','             ;Comma separator?
26BD 282B     04190          JR      Z,GMPW4
26BF FE22     04200          CP      '"'             ;Closing quote?
26C1 2827     04210          JR      Z,GMPW4
26C3 13       04220          INC     DE
26C4 77       04230          LD      (HL),A          ;Xfer the char
26C5 23       04240          INC     HL
26C6 10EE     04250          DJNZ    GMPW2           ;Loop for 8
26C8 1825     04260          JR      GMPW5
              04270 ;
              04280 ;        Not entered as parm, grab from keyboard
              04290 ;
26CA CD1527   04300 GMPW3    CALL    CKINDO          ;Can't prompt in <DO>
26CD C0       04310          RET     NZ
26CE          04320          @@DSPLY                 ;Display request
              00046          IFEQ    00H,1
              00047          LD      HL,
              00048          ENDIF
26CE 3E0A     00049          LD      A,10
26D0 EF       00050          RST     40
26D1 C0       04330          RET     NZ
26D2 010008   04340          LD      BC,8<8          ;Max 8 chars input
26D5 21002A   04350          LD      HL,BUFFER       ;Pt to buffer
26D8 E5       04360          PUSH    HL
26D9          04370          @@KEYIN                 ;Get parm input
26D9 3E09     00051          LD      A,9
26DB EF       00052          RST     40
26DC DAEC27   04380          JP      C,BREAK         ;Exit on Break
26DF EB       04390          EX      DE,HL           ;Buf start to DE
26E0 2600     04400          LD      H,0             ;Buf len to HL
26E2 68       04410          LD      L,B
26E3 19       04420          ADD     HL,DE           ;Pt to 1st unused pos
```

```
26E4 3E08      04430          LD      A,8             ;Calculate spaces needed
26E6 90        04440          SUB     B
26E7 2806      04450          JR      Z,GMPW5         ;Ret if none needed
26E9 47        04460          LD      B,A             ;Set counter for spaces
26EA 3620      04470 GMPW4    LD      (HL),' '        ;  & put them in
26EC 23        04480          INC     HL
26ED 10FB      04490          DJNZ    GMPW4
               04500 ;
               04510 ;        Convert (SP) through (SP)+7 to upper case
               04520 ;
26EF E1        04530 GMPW5    POP     HL              ;Rcvr pointer to buf
26F0 E5        04540          PUSH    HL
26F1 0608      04550          LD      B,8             ;Loop through field
26F3 7E        04560 GMPW6    LD      A,(HL)
26F4 FE61      04570          CP      'a'
26F6 3806      04580          JR      C,GMPW7
26F8 FE7B      04590          CP      'z'+1
26FA 3002      04600          JR      NC,GMPW7
26FC CBAE      04610          RES     5,(HL)          ;L/c -> U/C
26FE 23        04620 GMPW7    INC     HL
26FF 10F2      04630          DJNZ    GMPW6
2701 D1        04640          POP     DE              ;Rcvr ptr to start
2702 AF        04650          XOR     A               ;Indicate no error
2703 C9        04660          RET
               04670 ;
               04680 ;        Routine to read the granule allocation table
               04690 ;
2704 D5        04700 GATRD    PUSH    DE
2705 E5        04710          PUSH    HL
2706 FD5609    04720          LD      D,(IY+9)        ;Dir cylinder
2709 21002B    04730          LD      HL,GATBUF
270C 5D        04740          LD      E,L             ;Set to sector 0
270D           04750          @@RDSSC
270D 3E55      00053          LD      A,85
270F EF        00054          RST     40
2710 E1        04760          POP     HL
2711 D1        04770          POP     DE
2712 3E14      04780          LD      A,14H           ;Init "GAT read error
2714 C9        04790          RET                     ;Z or NZ
               04800 ;
               04810 ;        Routine to check if <DO> active
               04820 ;
2715 FDE5      04830 CKINDO   PUSH    IY
2717           04840          @@FLAGS
2717 3E65      00055          LD      A,101
2719 EF        00056          RST     40
271A FDCB126E  04850          BIT     5,(IY+'S'-'A')  ;Set if DO active
271E FDE1      04860          POP     IY
2720 C8        04870          RET     Z
2721 210428    04880          LD      HL,NOINDO$
2724 3E3F      04890          LD      A,63
2726 C9        04900          RET
               04910 ;
               04920 ;        Parse file name or ext on command line
               04930 ;
2727 FE24      04940 PRSPEC   CP      '$'             ;Wild character?
2729 2814      04950          JR      Z,PRS2          ;Always a match
272B FE41      04960          CP      'A'             ;Ck on filename entry
272D 3006      04970          JR      NC,PRS1         ;Jump if possible alpha
272F FE3A      04980          CP      '9'+1           ;Ck on 0-9
2731 D0        04990          RET     NC              ;Bad if > 9 and < A
```

```
2732 FE3Ø      Ø5ØØØ          CP      'Ø'
2734 D8        Ø5Ø1Ø          RET     C               ;Bad if < Ø
2735 FE61      Ø5Ø2Ø PRS1     CP      'a'             ;Cvrt to UC if needed
2737 3806      Ø5Ø3Ø          JR      C,PRS2
2739 FE7B      Ø5Ø4Ø          CP      'z'+1
273B 3ØØ2      Ø5Ø5Ø          JR      NC,PRS2
273D CBAF      Ø5Ø6Ø          RES     5,A
273F 12        Ø5Ø7Ø PRS2     LD      (DE),A          ;Xfer char to buffer
2740 13        Ø5Ø8Ø          INC     DE              ;Bump dest ptr
2741 7E        Ø5Ø9Ø          LD      A,(HL)          ;Get next char
2742 23        Ø51ØØ          INC     HL              ;Bump source ptr
2743 1ØE2      Ø511Ø          DJNZ    PRSPEC          ;Loop 8 max
2745 C9        Ø512Ø          RET
               Ø513Ø ;
               Ø514Ø ;        Routine to extract date from directory
               Ø515Ø ;
2746 7E        Ø516Ø UNPACK   LD      A,(HL)          ;P/u DIR+1
2747 E6ØF      Ø517Ø          AND     ØFH             ;Remove flags
2749 57        Ø518Ø          LD      D,A             ;Save month
274A 23        Ø519Ø          INC     HL              ;Pt to DIR+2
274B 7E        Ø52ØØ          LD      A,(HL)
274C E6F8      Ø521Ø          AND     ØF8H            ;Strip year
274E 5F        Ø522Ø          LD      E,A             ;Save day in E
274F 7E        Ø523Ø          LD      A,(HL)
275Ø AB        Ø524Ø          XOR     E               ;Strip day fm year
2751 ØF        Ø525Ø          RRCA                    ;Shift year to 5-7
2752 ØF        Ø526Ø          RRCA
2753 ØF        Ø527Ø          RRCA
2754 B2        Ø528Ø          OR      D               ;Merge with month
2755 57        Ø529Ø          LD      D,A
2756 C9        Ø53ØØ          RET
               Ø531Ø ;
               Ø532Ø ;        Pack user date string
               Ø533Ø ;
2757 7E        Ø534Ø PAKDAT   LD      A,(HL)
2758 ØE2F      Ø535Ø          LD      C,'/'           ;Init separator
275A CDA227    Ø536Ø          CALL    PARSDAT         ;Parse entry
275D 2Ø3C      Ø537Ø          JR      NZ,BADFMT       ;Jump on format error
275F EB        Ø538Ø          EX      DE,HL
2760 3AC828    Ø539Ø          LD      A,(LILBUF$)     ;Is year a leap year?
2763 E6Ø3      Ø54ØØ          AND     3
2765 215C28    Ø541Ø          LD      HL,MAXDAYS+1    ;Set Feb to have 29 days
2768 2ØØ1      Ø542Ø          JR      NZ,$+3          ;  if so
276A 34        Ø543Ø          INC     (HL)
276B 3ACA28    Ø544Ø          LD      A,(LILBUF$+2)   ;P/u month
276E 3D        Ø545Ø          DEC     A               ;Range check
276F FEØC      Ø546Ø          CP      12
2771 3Ø28      Ø547Ø          JR      NC,BADFMT       ;Go if Ø or >12
2773 2B        Ø548Ø          DEC     HL              ;Point to Jan entry
2774 85        Ø549Ø          ADD     A,L             ;Index the month
2775 6F        Ø55ØØ          LD      L,A
2776 7C        Ø551Ø          LD      A,H
2777 CEØØ      Ø552Ø          ADC     A,Ø
2779 67        Ø553Ø          LD      H,A
277A 3AC928    Ø554Ø          LD      A,(LILBUF$+1)   ;P/u day entry
277D 3D        Ø555Ø          DEC     A               ;Reduce for test (Ø->FF)
277E BE        Ø556Ø          CP      (HL)
277F 3Ø1A      Ø557Ø          JR      NC,BADFMT       ;Go if too large (or Ø)
2781 21CA28    Ø558Ø          LD      HL,LILBUF$+2    ;Pt to month
2784 7E        Ø559Ø          LD      A,(HL)          ;P/u month
2785 2B        Ø56ØØ          DEC     HL              ;Pt to day
```

```
2786 47       Ø5610          LD      B,A             ;Save month in B
2787 7E       Ø5620          LD      A,(HL)          ;P/u day
2788 2B       Ø5630          DEC     HL              ;Pt to year
2789 Ø7       Ø5640          RLCA                    ;Shift day to 3-7
278A Ø7       Ø5650          RLCA
278B Ø7       Ø5660          RLCA
278C 4F       Ø5670          LD      C,A
278D 7E       Ø5680          LD      A,(HL)          ;P/u year
278E D65Ø     Ø5690          SUB     8Ø              ;Adjust for offset
279Ø 3ØØ1     Ø57ØØ          JR      NC,$+3          ;If entry < 198Ø,
2792 AF       Ø5710          XOR     A               ;  then use 198Ø
2793 ØF       Ø5720          RRCA                    ;Shift into bits 5-7
2794 ØF       Ø5730          RRCA
2795 ØF       Ø5740          RRCA
2796 BØ       Ø5750          OR      B               ;  & merge with month
2797 47       Ø5760          LD      B,A
2798 EB       Ø5770          EX      DE,HL
2799 AF       Ø5780          XOR     A               ;Set Z, no error
279A C9       Ø5790          RET
              Ø58ØØ ;
279B 216728   Ø5810 BADFMT   LD      HL,BADFMT$      ;Init error pointer
279E 3E3F     Ø5820          LD      A,63            ;Set extended error
27AØ B7       Ø5830          OR      A
27A1 C9       Ø5840          RET
              Ø5850 ;
              Ø5860 ;        Routine to parse DATE/TIME entry
              Ø5870 ;
27A2 11CA28   Ø5880 PARSDAT  LD      DE,LILBUF$+2    ;Point to buf end
27A5 Ø6Ø3     Ø5890          LD      B,3             ;Process 3 fields
27A7 D5       Ø59ØØ PRSD1    PUSH    DE              ;Save pointer
27A8 CDB727   Ø5910          CALL    PRSD2           ;Get a digit pair
27AB D1       Ø5920          POP     DE              ;Recover pointer
27AC CØ       Ø5930          RET     NZ              ;Ret if bad digit pair
27AD 12       Ø5940          LD      (DE),A          ;  else stuff the value
27AE 1B       Ø5950          DEC     DE              ;Backup the pointer
27AF Ø5       Ø5960          DEC     B               ;Loop countdown
27BØ C8       Ø5970          RET     Z
27B1 7E       Ø5980          LD      A,(HL)          ;Ck for valid separator
27B2 23       Ø5990          INC     HL              ;Bump pointer
27B3 B9       Ø6ØØØ          CP      C               ;Separator char required
27B4 28F1     Ø6Ø1Ø          JR      Z,PRSD1         ;Loop if match
27B6 C9       Ø6Ø2Ø          RET                     ;  else ret bad (NZ)
              Ø6Ø3Ø ;
              Ø6Ø4Ø ;        Routine to parse a digit pair
              Ø6Ø5Ø ;
27B7 CDCE27   Ø6Ø6Ø PRSD2    CALL    PRS4            ;Get a digit
27BA 3Ø1Ø     Ø6Ø7Ø          JR      NC,PRSD3        ;Jump if bad digit
27BC 5F       Ø6Ø8Ø          LD      E,A             ;Multiply by ten
27BD Ø7       Ø6Ø9Ø          RLCA
27BE Ø7       Ø61ØØ          RLCA
27BF 83       Ø611Ø          ADD     A,E
27CØ Ø7       Ø612Ø          RLCA
27C1 5F       Ø613Ø          LD      E,A
27C2 CDCE27   Ø614Ø          CALL    PRS4            ;Get another digit
27C5 3ØØ5     Ø615Ø          JR      NC,PRSD3        ;Jump on bad digit
27C7 83       Ø616Ø          ADD     A,E             ;Accumulate new digit
27C8 5F       Ø617Ø          LD      E,A             ;Save 2-digit value
27C9 AF       Ø618Ø          XOR     A               ;Clear flags
27CA 7B       Ø619Ø          LD      A,E             ;Xfer field value
27CB C9       Ø62ØØ          RET
              Ø621Ø ;
```

```
27CC B7        Ø622Ø PRSD3   OR      A               ;Set NZ
27CD C9        Ø623Ø         RET
27CE 7E        Ø624Ø PRS4    LD      A,(HL)          ;P/u a digit &
27CF 23        Ø625Ø         INC     HL              ;  convert to binary
27DØ D63Ø      Ø626Ø         SUB     'Ø'
27D2 FEØA      Ø627Ø         CP      1Ø
27D4 C9        Ø628Ø         RET
               Ø629Ø ;
               Ø63ØØ ;       Routine to compare DE to HL
               Ø631Ø ;
27D5 7C        Ø632Ø CPHLDE  LD      A,H
27D6 92        Ø633Ø         SUB     D
27D7 CØ        Ø634Ø         RET     NZ
27D8 7D        Ø635Ø         LD      A,L
27D9 93        Ø636Ø         SUB     E
27DA C9        Ø637Ø         RET
               Ø638Ø ;
               Ø639Ø ;       Error processing
               Ø64ØØ ;
27DB 3E2C      Ø641Ø PRMERR  LD      A,PAR_ERR       ;Parameter Error
27DD FE3F      Ø642Ø IOERR   CP      63              ;Extended error?
27DF 281E      Ø643Ø         JR      Z,EXTERR
27E1 6F        Ø644Ø         LD      L,A
27E2 26ØØ      Ø645Ø         LD      H,Ø
27E4 F6CØ      Ø646Ø         OR      ØCØH            ;Abbrev & return
27E6 4F        Ø647Ø         LD      C,A
27E7           Ø648Ø         @@ERROR
27E7 3E1A      ØØØ57         LD      A,26
27E9 EF        ØØØ58         RST     4Ø
27EA 18Ø6      Ø649Ø         JR      SAVESP
               Ø65ØØ ;
               Ø651Ø ;       BREAK handler routine
               Ø652Ø ;
27EC           Ø653Ø BREAK   @@CKBRKC                ;Clear Break Bit
27EC 3E6A      ØØØ59         LD      A,1Ø6
27EE EF        ØØØ6Ø         RST     4Ø
27EF 21FFFF    Ø654Ø ERREXIT LD      HL,-1
27F2 31ØØØØ    Ø655Ø SAVESP  LD      SP,$-$          ;Restore the stack
27F5 22F927    Ø656Ø         LD      (RETCOD),HL
27F8           Ø657Ø EXIT    EQU     $               ;Exit clears Break
27F8 21ØØØØ    Ø658Ø         LD      HL,Ø
27F9           Ø659Ø RETCOD  EQU     $-2
27FB           Ø66ØØ         @@CKBRKC
27FB 3E6A      ØØØ61         LD      A,1Ø6
27FD EF        ØØØ62         RST     4Ø
27FE C9        Ø661Ø         RET
               Ø662Ø ;
27FF           Ø663Ø EXTERR  @@LOGOT
               ØØØ63         IFEQ    ØØH,1
               ØØØ64         LD      HL,
               ØØØ65         ENDIF
27FF 3EØC      ØØØ66         LD      A,12
2801 EF        ØØØ67         RST     4Ø
2802 18EB      Ø664Ø         JR      ERREXIT
2804 49        Ø665Ø NOINDO$ DB      'Invalid command during <DO> '
     6E 76 61 6C 69 64 2Ø 63
     6F 6D 6D 61 6E 64 2Ø 64
     75 72 69 6E 67 2Ø 3C 44
     4F 3E 2Ø
282Ø 7Ø        Ø666Ø         DB      'processing',CR
     72 6F 63 65 73 73 69 6E
```

```
     67 ØD
282B 4D        Ø667Ø MPW$    DB      'Master password ?      ',3
     61 73 74 65 72 2Ø 7Ø 61
     73 73 77 6F 72 64 2Ø 3F
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø Ø3
2843 49        Ø668Ø BADMPW$ DB      'Invalid master password',CR
     6E 76 61 6C 69 64 2Ø 6D
     61 73 74 65 72 2Ø 7Ø 61
     73 73 77 6F 72 64 ØD
285B 1F        Ø669Ø MAXDAYS DB      31,28,31,3Ø,31,3Ø,31,31,3Ø,31,3Ø,31
     1C 1F 1E 1F 1E 1F 1F 1E
     1F 1E 1F
2867 42        Ø67ØØ BADFMT$ DB      'Bad date format',CR
     61 64 2Ø 64 61 74 65 2Ø
     66 6F 72 6D 61 74 ØD
               Ø671Ø ;
2877 5Ø        Ø672Ø PRGFIL$ DB      'Purge file: ',3
     75 72 67 65 2Ø 66 69 6C
     65 3A 2Ø Ø3
2884 7D        Ø673Ø QMARK$  DB      '} ?  ',3
     2Ø 3F 2Ø 2Ø Ø3
288A 5Ø        Ø674Ø PURGE$  DB      'Purging: ',3
     75 72 67 69 6E 67 3A 2Ø
     Ø3
2894 4A        Ø675Ø MONTBL  DM      'JanFebMarAprMayJunJulAugSepOctNovDec'
     61 6E 46 65 62 4D 61 72
     41 7Ø 72 4D 61 79 4A 75
     6E 4A 75 6C 41 75 67 53
     65 7Ø 4F 63 74 4E 6F 76
     44 65 63
28B8 2Ø        Ø676Ø BLANKS  DM      '                '
     2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
     2Ø 2Ø
28C3 ØØ        Ø677Ø FTFLG   DB      Ø
ØØØ2           Ø678Ø FMPAKD  DS      2
ØØØ2           Ø679Ø TOPAKD  DS      2
ØØØ4           Ø68ØØ LILBUF$ DS      4
ØØ1B           Ø681Ø FCB1$   DS      27
               Ø682Ø ;
               Ø683Ø ;       Parameter table
               Ø684Ø ;
28E7 8Ø        Ø685Ø PRMTBL$ DB      8ØH
ØØ8Ø           Ø686Ø VAL     EQU     8ØH
ØØ4Ø           Ø687Ø SW      EQU     4ØH
ØØ2Ø           Ø688Ø STR     EQU     2ØH
ØØ1Ø           Ø689Ø SGL     EQU     1ØH
28E8 53        Ø69ØØ         DB      SW!SGL!3,'INV',Ø
     49 4E 56 ØØ
28ED 1325      Ø691Ø         DW      IPARM+1
28EF 53        Ø692Ø         DB      SW!SGL!3,'SYS',Ø
     53 59 53 ØØ
28F4 Ø525      Ø693Ø         DW      SPARM+1
28F6 73        Ø694Ø         DB      SW!STR!SGL!3,'MPW',Ø
     4D 5Ø 57 ØØ
28FB 8C26      Ø695Ø         DW      PWPARM+1
28FD 55        Ø696Ø         DB      SW!SGL!5,'QUERY',Ø
     51 55 45 52 59 ØØ
29Ø4 BØ25      Ø697Ø         DW      QPARM+1
29Ø6 34        Ø698Ø         DB      STR!SGL!4,'DATE',Ø
     44 41 54 45 ØØ
29ØC 5624      Ø699Ø         DW      DATPRM+1
```

```
290E ØØ       Ø7ØØØ          NOP
              Ø7Ø1Ø ;
ØØ2Ø          Ø7Ø2Ø FCB      DS      32
292F D63Ø     Ø7Ø3Ø PATCH1   SUB     'Ø'             ;Cvrt to binary
2931 FEØ8     Ø7Ø4Ø          CP      7+1
2933 D2DB27   Ø7Ø5Ø          JP      NC,PRMERR
2936 4F       Ø7Ø6Ø          LD      C,A
2937 C9       Ø7Ø7Ø          RET
              Ø7Ø8Ø ;
2938 C2DD27   Ø7Ø9Ø PATCH2   JP      NZ,IOERR        ;Go on CKDRV error
293B 3EØF     Ø71ØØ          LD      A,15            ;Init WP error
293D DADD27   Ø711Ø          JP      C,IOERR         ;Exit if WP
294Ø C9       Ø712Ø          RET
              Ø713Ø ;
2AØØ          Ø714Ø          ORG     $<-8+1<+8
Ø1ØØ          Ø715Ø BUFFER   DS      256
Ø1ØØ          Ø716Ø GATBUF   DS      256
Ø1ØØ          Ø717Ø HITBUF   DS      256
2CFF          Ø718Ø LAST     EQU     $-1
              Ø719Ø ;
24ØØ          Ø72ØØ          END     PURGE
```

@@1	ØØØØ	@@2	ØØØØ	@@3	ØØØØ	@@4	ØØØØ
@MOD2	FFFF	BADFMT	279B	BADFMT$	2867	BADMPW$	2843
BEGINA	24Ø9	BLANKS	28B8	BREAK	27EC	BUFFER	2AØØ
CKINDO	2715	CKINV	25ØE	CKNAM	251A	CKTO	2479
CPHLDE	27D5	CR	ØØØD	DATPRM	2455	DOEVER	249D
DONAM	2574	DONAM1	257E	DONAM4	2596	DONAM5	25AØ
DONAME2	2588	ERREXIT	27EF	EXIT	27F8	EXTERR	27FF
FCB	29ØF	FCB1$	28CC	FMPAKD	28C4	FRCTO	2485
FTFLG	28C3	GATBUF	2BØØ	GATRD	27Ø4	GETMPW	26A2
GMPW1	26A9	GMPW2	26B6	GMPW3	26CA	GMPW4	26EA
GMPW5	26EF	GMPW6	26F3	GMPW7	26FE	HITBUF	2CØØ
IOERR	27DD	IPARM	2512	LAST	2CFF	LF	ØØØA
LILBUF$	28C8	MATCHES	2573	MAXDAYS	285B	MFLG	253D
MONTBL	2894	MPW$	282B	NOINDO$	28Ø4	NOPRMPT	263D
PAKDAT	2757	PARSDAT	27A2	PAR_ERR	ØØ2C	PASSWORD	42EØ
PATCH1	292F	PATCH2	2938	PRGFIL$	2877	PRMERR	27DB
PRMERRA	2452	PRMTBL$	28E7	PRS1	2735	PRS2	273F
PRS4	27CE	PRSD1	27A7	PRSD2	27B7	PRSD3	27CC
PRSPEC	2727	PURØ	2421	PUR1	2432	PUR2	243D
PUR3	2491	PURGE	24ØØ	PURGE$	288A	PURGE1	24ØD
PWPARM	268B	QMARK$	2884	QPARM	25AF	RETCOD	27F9
SAVESP	27F2	SCDAT1	25CD	SCDAT2	25E2	SCDAT4	2618
SCNH1	24BD	SCNH2	24C1	SCNH3	24D4	SCNH3A	24E4
SCNH4	24F5	SCNH5	2524	SCNH6	253Ø	SCNH6A	2543
SCNH6B	255F	SCNHIT	24BC	SGL	ØØ1Ø	SPARM	25Ø4
STR	ØØ2Ø	SW	ØØ4Ø	TOPAKD	28C6	TSTMPW	267B
UNPACK	2746	VAL	ØØ8Ø	@@ABORT	AB4A	@@ADTSK	ABDD
@@BANK	BØF5	@@BKSP	ADD5	@@BREAK	B1ØB	@@CHNIO	AB35
@@CKBRKC	B159	@@CKDRV	AC31	@@CKEOF	ADEA	@@CKTSK	ABC8
@@CLOSE	ADCØ	@@CLS	B143	@@CMNDI	AB74	@@CMNDR	AB89
@@CTL	A999	@@DATE	ABØB	@@DCSTAT	AC7Ø	@@DEBUG	ABB3
@@DECHEX	BØ75	@@DIRRD	AFE2	@@DIRWR	AFF7	@@DIV16	BØ6Ø
@@DIV8	BØ4B	@@DODIR	AC46	@@DSP	A95D	@@DSPLY	A9FD
@@ERROR	AB9E	@@EXIT	AB5F	@@FEXT	AF4F	@@FLAGS	BØDF
@@FNAME	AF64	@@FSPEC	AF3A	@@GATRD	AFCD	@@GATWR	BØØC
@@GET	A971	@@GTDCB	AF8E	@@GTDCT	AF79	@@GTMOD	AFA3
@@HDFMT	AD18	@@HEX16	BØB4	@@HEX8	BØ9F	@@HEXDEC	BØ8A
@@HIGH$	BØC9	@@INIT	AD96	@@KBD	A9D5	@@KEY	A949
@@KEYIN	A9E9	@@KLTSK	AC1C	@@LOAD	AF1Ø	@@LOC	ADFF
@@LOF	AE14	@@LOGOT	AA49	@@LOGER	AA34	@@MSG	AA8Ø
@@MUL16	BØ36	@@MUL8	BØ21	@@OPEN	ADAB	@@PARAM	AAF6
@@PAUSE	AAE1	@@PEOF	AE29	@@POSN	AE3E	@@PRINT	AA95
@@PRT	A9AD	@@PUT	A985	@@RAMDIR	AC5B	@@RDSEC	ACEE
@@RDSSC	AFB8	@@READ	AE53	@@REMOV	AD81	@@RENAM	AD6C
@@REW	AE68	@@RMTSK	ABF2	@@RPTSK	ACØ7		

```
@@RREAD          AE7D @@RSLCT          ACD9 @@RSTOR          AC9A
@@RUN            AF25 @@RWRIT          AE92 @@SEEK           ACC4
@@SEEKSC         AEA7 @@SKIP           AEBC @@SLCT           AC85
@@STEPI          ACAF @@TIME           AB2Ø @@VDCTL          AACC
@@VER            AED1 @@VRSEC          ADØ3 @@WEOF           AEE6
@@WHERE          A9C1 @@WRITE          AEFB @@WRSEC          AD2D
@@WRSSC          AD42 @@WRTRK          AD57
2400 is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: REMOVE

Library: SYS6/SYS

ISAM # : 18H

```
                ØØ1ØØ ;LBREMOVE/ASM - REMOVE command
ØØØØ            ØØ11Ø          TITLE   <REMOVE - LS-DOS 6.2>
                ØØ12Ø ;
ØØØØ            ØØ13Ø *GET     SVCMAC:3                ;SVC Macro equivalents
                ØØØ1Ø ;SVCMAC/ASM - LS-DOS Version VI
                ØØØ2Ø *LIST    OFF
                Ø39ØØ *LIST    ON
ØØØØ            ØØ14Ø *GET     VALUES:3                ;Misc. equates
                Ø392Ø ;VALUES/ASM - Version 6
                Ø393Ø *LIST OFF
                Ø42ØØ *LIST ON
                ØØ15Ø ;
24ØØ            ØØ16Ø          ORG     24ØØH
                ØØ17Ø ;
24ØØ            ØØ18Ø REMOVE   @@CKBRKC                ;Break key down?
24ØØ 3E6A       ØØØØ1          LD      A,1Ø6
24Ø2 EF         ØØØØ2          RST     4Ø
24Ø3 28Ø4       ØØ19Ø          JR      Z,BEGINA        ;Ok if not
24Ø5 21FFFF     ØØ2ØØ          LD      HL,-1           ;  else abort
24Ø8 C9         ØØ21Ø          RET
                ØØ22Ø ;
24Ø9 ED731B24   ØØ23Ø BEGINA   LD      (SAVESP+1),SP   ;Save Stack
24ØD CD3724     ØØ24Ø          CALL    REMOV           ;Do the removing
241Ø 18Ø5       ØØ25Ø          JR      EXIT
                ØØ26Ø ;
                ØØ27Ø ;        Exit Routine - Clear any <BREAK> & return
                ØØ28Ø ;
                ØØ29Ø BRKABT
2412 21FFFF     ØØ3ØØ          LD      HL,-1
2415 18Ø3       ØØ31Ø          JR      SAVESP          ;Abort with error
                ØØ32Ø ;
2417 21ØØØØ    ØØ33Ø EXIT     LD      HL,Ø            ;Set retcod if exiting
241A 31ØØØØ    ØØ34Ø SAVESP   LD      SP,$-$          ;P/u stack
241D            ØØ35Ø          @@CKBRKC                ;Clear <BREAK> if entered
241D 3E6A       ØØØØ3          LD      A,1Ø6
241F EF         ØØØØ4          RST     4Ø
242Ø C9         ØØ36Ø          RET                     ;  and Return
                ØØ37Ø ;
                ØØ38Ø ;        Special Message Exit
                ØØ39Ø ;
2421 212C25     ØØ4ØØ SPCREQ   LD      HL,SPCREQ$
2424            ØØ41Ø          @@LOGOT
                ØØØØ5          IFEQ    ØØH,1
                ØØØØ6          LD      HL,
                ØØØØ7          ENDIF
2424 3EØC       ØØØØ8          LD      A,12
2426 EF         ØØØØ9          RST     4Ø
2427 21FFFF     ØØ42Ø ERREXIT  LD      HL,-1
242A 18EE       ØØ43Ø          JR      SAVESP          ;Get stack & RET
                ØØ44Ø ;
                ØØ45Ø ;        I/O Error Exit
                ØØ46Ø ;
242C 6F         ØØ47Ø IOERR    LD      L,A             ;Error # to HL
242D 26ØØ       ØØ48Ø          LD      H,Ø
242F F6CØ       ØØ49Ø          OR      ØCØH            ;Abbrev, return
2431 4F         ØØ5ØØ          LD      C,A             ;Error code to reg C
2432            ØØ51Ø          @@ERROR
2432 3E1A       ØØØ1Ø          LD      A,26
2434 EF         ØØØ11          RST     4Ø
2435 18E3       ØØ52Ø          JR      SAVESP          ;P/u stack & return
                ØØ53Ø ;
```

```
                00540 ;       REMOV - Remove file(s) or device(s)
                00550 ;
2437 E5         00560 REMOV   PUSH    HL              ;Save INBUF$ pointer
2438 AF         00570 REMOV0  XOR     A               ;Set Z-flag for loop
2439 E1         00580 REMOV1  POP     HL
243A C22C24     00590         JP      NZ,IOERR
                00600 ;
                00610 ;       Skip leading spaces - exit on terminator
                00620 ;
243D            00630         @@CKBRKC                ;Check for break
243D 3E6A       00012         LD      A,106
243F EF         00013         RST     40
2440 20D0       00640         JR      NZ,BRKABT       ;  abort if so
                00650 ;
2442 2B         00660         DEC     HL
2443 23         00670 REMOV2  INC     HL              ;Skip past spaces
2444 7E         00680         LD      A,(HL)
2445 FE20       00690         CP      ' '
2447 28FA       00700         JR      Z,REMOV2
2449 D8         00710         RET     C               ;Return on terminator
                00720 ;
                00730 ;       Hit a non-space char - is this a filespec ?
                00740 ;
244A 115B25     00750 REMOV3  LD      DE,FCB          ;Fetch the filespec
244D            00760         @@FSPEC
244D 3E4E       00014         LD      A,78
244F EF         00015         RST     40
2450 C22124     00770         JP      NZ,SPCREQ       ;Jump on spec error
2453 E5         00780         PUSH    HL              ;Save INBUF$ pointer
                00790 ;
                00800 ;       Transfer Filespec to buff without password
                00810 ;
2454 D5         00820         PUSH    DE              ;Save FCB ptr
2455 EB         00830         EX      DE,HL           ;Pt HL => FCB
2456 114C25     00840         LD      DE,FILESPC      ;DE => Filespec
2459 CD0B25     00850         CALL    MOVSPC          ;Move filespec
245C 3E0D       00860         LD      A,CR            ;End spec with C/R
245E 12         00870         LD      (DE),A
245F D1         00880         POP     DE              ;Recover FCB
                00890 ;
                00900 ;       Are we Removing a Device ?
                00910 ;
2460 1A         00920         LD      A,(DE)          ;Is it a device?
2461 FE2A       00930         CP      '*'
2463 2843       00940         JR      Z,RMVDVC
                00950 ;
                00960 ;       Removing a File - Does it exist ?
                00970 ;
2465            00980         @@OPEN                  ;Open the file
2465 3E3B       00016         LD      A,59
2467 EF         00017         RST     40
2468 2804       00990         JR      Z,GOODOPN       ;Z - it does exist
246A FE2A       01000         CP      42              ;LRL Open Fault ?
246C 20CB       01010         JR      NZ,REMOV1       ;No - abort
                01020 ;
                01030 ;       Successful Open - Can we Remove the file ?
                01040 ;
246E 3A5C25     01050 GOODOPN LD      A,(FCB+1)       ;P/u access level
2471 E607       01060         AND     7
2473 FE02       01070         CP      2               ;REMOVE access ?
2475 3E25       01080         LD      A,37            ;No - "Illegal access
```

```
2477 D22C24   01090          JP      NC,IOERR        ;  attempted to ..."
              01100 ;
              01110 ;        Is there a drivespec in the filespec ?
              01120 ;
247A 214C25   01130          LD      HL,FILESPC      ;HL => Filespec
247D 7E       01140 FLOOP    LD      A,(HL)          ;P/u char
247E FE0E     01150          CP      CR+1            ;End of Filespec ?
2480 300E     01160          JR      NC,CHKDSPC
              01170 ;
              01180 ;        Drivespec wasn't specified - put it on
              01190 ;
2482 363A     01200          LD      (HL),':'        ;Append drivespec onto
2484 23       01210          INC     HL              ;  end of filespec
2485 3A6125   01220          LD      A,(FCB+6)       ;Xfer drive # to A
2488 C630     01230          ADD     A,'0'           ;Convert to ASCII
248A 77       01240          LD      (HL),A
248B 23       01250          INC     HL              ;Bump
248C 360D     01260          LD      (HL),CR         ;End of filespec
248E 1805     01270          JR      SHOWFIL         ;Dsply name & remove it
              01280 ;
              01290 ;        Stop when ":" hit or terminator
              01300 ;
2490 FE3A     01310 CHKDSPC  CP      ':'             ;Already have one ?
2492 23       01320          INC     HL
2493 20E8     01330          JR      NZ,FLOOP
              01340 ;
              01350 ;        Display "Removing Filespec" & remove it
              01360 ;
2495 CD9D24   01370 SHOWFIL  CALL    SHOWIT          ;"Removing:"
2498          01380          @@REMOV                 ;Remove the file
2498 3E39     00018          LD      A,57
249A EF       00019          RST     40
249B 189C     01390          JR      REMOV1          ;Loop for more
              01400 ;
              01410 ;        Display Filespec or Devspec
              01420 ;
249D 214225   01430 SHOWIT   LD      HL,SHOW$        ;Show what we are
24A0 D5       01440          PUSH    DE              ;Save FCB ptr
24A1          01450          @@DSPLY                 ;  removing
              00020          IFEQ    00H,1
              00021          LD      HL,
              00022          ENDIF
24A1 3E0A     00023          LD      A,10
24A3 EF       00024          RST     40
24A4 D1       01460          POP     DE
24A5 2092     01470          JR      NZ,REMOV1       ;NZ - abort
24A7 C9       01480          RET                     ;  else RETurn Z
              01490 ;
              01500 ;        Routine to Remove a device
              01510 ;
24A8 D5       01520 RMVDVC   PUSH    DE              ;Xfer FCB ptr to IX
24A9 DDE1     01530          POP     IX
24AB DD5E01   01540          LD      E,(IX+1)        ;P/u device name
24AE DD5602   01550          LD      D,(IX+2)
              01560 ;
              01570 ;        Routine to find a device in the device tables
              01580 ;
24B1          01590          @@GTDCB                 ;Find this DCB
24B1 3E52     00025          LD      A,82
24B3 EF       00026          RST     40
24B4 2083     01600          JR      NZ,REMOV1       ;Abort if not found
```

```
                01610 ;
                01620 ;       Found device in tables, is it killable?
                01630 ;
24B6 ED534025   01640 DEV4    LD      (RENDEV+1),DE   ;Try to rename it
24BA E5         01650         PUSH    HL              ;Save address pointer
24BB 213F25     01660         LD      HL,RENDEV       ;New & old names same
24BE 5D         01670         LD      E,L
24BF 54         01680         LD      D,H
24C0            01690         @@RENAM
24C0 3E38       00027         LD      A,56
24C2 EF         00028         RST     40
24C3 E1         01700         POP     HL              ;Rcvr address pointer
24C4 FE13       01710         CP      19              ;If not "illegal name"
24C6 C23924     01720         JP      NZ,REMOV1       ;  error, is protected
                01730 ;
                01740 ;       Device not system device - is it in use?
                01750 ;
24C9 CB5E       01760 CKNILL  BIT     3,(HL)          ;Test NIL device
24CB 2006       01770         JR      NZ,REMOVIT      ;OK if NIL (not in use)
24CD 3E27       01780         LD      A,27H           ;  else "device in use"
24CF B7         01790         OR      A
24D0 C33924     01800         JP      REMOV1          ;  and return error
                01810 ;
                01820 ;       Remove the device
                01830 ;
24D3 CD0325     01840 REMOVIT CALL    SCND4           ;Zero 8-byte DCB area
                01850 ;
                01860 ;       Any other device routes to the removed one?
                01870 ;
24D6 114B49     01880 SCNDCB  LD      DE,'IK'         ;Point to begin of area
24D9            01890         @@GTDCB
24D9 3E52       00029         LD      A,82
24DB EF         00030         RST     40
24DC CB66       01900 SCND1   BIT     4,(HL)          ;Routed device?
24DE 2811       01910         JR      Z,SCND2         ;Jump if not
24E0 E5         01920         PUSH    HL              ;Save DCB pointer
24E1 2C         01930         INC     L               ;Bypass TYPE code
24E2 7E         01940         LD      A,(HL)          ;P/u route vector
24E3 2C         01950         INC     L               ;  lo-order
24E4 66         01960         LD      H,(HL)          ;P/u vector hi-order
24E5 6F         01970         LD      L,A
24E6 7E         01980         LD      A,(HL)          ;P/u TYPE code of nest
24E7 E1         01990         POP     HL              ;Rcvr DCB pointer
24E8 B7         02000         OR      A               ;If TYPE=0, we killed
24E9 2006       02010         JR      NZ,SCND2        ;  the routed device
24EB 3E08       02020         LD      A,8
24ED B6         02030         OR      (HL)            ;Convert to NIL if so
24EE 77         02040         LD      (HL),A
24EF 18E5       02050         JR      SCNDCB
                02060 ;
                02070 ;       Point to next device
                02080 ;
24F1 7D         02090 SCND2   LD      A,L             ;Advance to next DCB area
24F2 C608       02100         ADD     A,8             ;Loop through all
24F4 6F         02110         LD      L,A             ;  devices while checking
24F5 20E5       02120         JR      NZ,SCND1        ;Loop until table end
                02130 ;
                02140 ;       Device table cleared, now zero the DCB/FCB
                02150 ;
24F7 DDE5       02160         PUSH    IX              ;Xfer vector to HL
24F9 E1         02170         POP     HL
```

```
24FA CD0325   02180          CALL    SCND4
24FD CD9D24   02190          CALL    SHOWIT          ;Show "Removing:"
2500 C33824   02200          JP      REMOV0          ;Next ...
              02210 ;
2503 0608     02220 SCND4    LD      B,8             ;Clear 8 bytes
2505 AF       02230          XOR     A
2506 77       02240 SCND5    LD      (HL),A          ;Go for it
2507 23       02250          INC     HL
2508 10FC     02260          DJNZ    SCND5
250A C9       02270          RET
              02280 ;
              02290 ;        Transfer a filespec from HL to DE
              02300 ;
250B 7E       02310 MOVSPC   LD      A,(HL)          ;P/u a spec character
250C FE2F     02320          CP      '/'             ;Extension ?
250E 2008     02330          JR      NZ,CKSPACE      ;No - check if space
2510 23       02340          INC     HL              ;Is the next character
2511 7E       02350          LD      A,(HL)          ;  valid ?
2512 FE41     02360          CP      'A'
2514 3802     02370          JR      C,CKSPACE       ;No - don't output it
2516 2B       02380          DEC     HL              ;Back one
2517 7E       02390          LD      A,(HL)          ;P/u slash
2518 FE20     02400 CKSPACE  CP      ' '
251A D8       02410          RET     C               ;Stop on space or less
251B FE2E     02420          CP      '.'             ;Password ?
251D 2009     02430          JR      NZ,MOVSPC1
251F 23       02440 SKIPPW   INC     HL
2520 7E       02450          LD      A,(HL)
2521 FE20     02460          CP      ' '
2523 D8       02470          RET     C               ;Back on terminator
2524 FE3A     02480          CP      ':'
2526 20F7     02490          JR      NZ,SKIPPW
2528 EDA0     02500 MOVSPC1  LDI                     ;Move the char
252A 18DF     02510          JR      MOVSPC
              02520 ;
252C 46       02530 SPCREQ$  DB      'File spec required',CR
     69 6C 65 20 73 70 65 63
     20 72 65 71 75 69 72 65
     64 0D
253F 2A       02540 RENDEV   DB      '*LS'
     4C 53
2542 52       02550 SHOW$    DB      'Removing: '
     65 6D 6F 76 69 6E 67 3A
     20
000F          02560 FILESPC  DS      15
255B 00       02570 FCB      DB      0
001F          02580          DS      31
              02590 ;
2400          02600          END     REMOVE
```

```
@@1      ØØØØ @@2      ØØØØ @@3      ØØØØ
@@4      ØØØØ @MOD2    ØØØØ @MOD4    FFFF
ABB      ØØ1Ø AP       ØØ27 BEGINA   24Ø9
BREAK    ØØ8Ø BRKABT   2412 BS       ØØØ8
CFLAG$   ØØØ2 CHKDSPC  249Ø CKNILL   24C9
CKSPACE  2518 CR       ØØØD DEV4     24B6
DFLAG$   ØØØ3 ERREXIT  2427 ETX      ØØØ3
EXIT     2417 FCB      255B FILESPC  254C
FLAG     ØØ4Ø FLOOP    247D GOODOPN  246E
IOERR    242C KFLAG$   ØØØA LF       ØØØA
MOVSPC   25ØB MOVSPC1  2528 NUM      ØØ8Ø
PAR_ERR  ØØ2C REMOV    2437 REMOVØ   2438
REMOV1   2439 REMOV2   2443 REMOV3   244A
REMOVE   24ØØ REMOVIT  24D3 RENDEV   253F
RMVDVC   24A8 SAVESP   241A SCND1    24DC
SCND2    24F1 SCND4    25Ø3 SCND5    25Ø6
SCNDCB   24D6 SFLAG$   ØØ12 SHOW$    2542
SHOWFIL  2495 SHOWIT   249D SKIPPW   251F
SPCREQ   2421 SPCREQ$  252C STR      ØØ2Ø
TAB      ØØØ9 VFLAG$   ØØ15 @@ABORT 817D
@@ADTSK  821Ø @@BANK   8728 @@BKSP   84Ø8
@@BREAK  873E @@CHNIO  8168 @@CKBRKC 878C
@@CKDRV  8264 @@CKEOF  841D @@CKTSK  81FB
@@CLOSE  83F3 @@CLS    8776 @@CMNDI  81A7
@@CMNDR  81BC @@CTL    7FCC @@DATE   813E
@@DCSTAT 82A3 @@DEBUG  81E6 @@DECHEX 86A8
@@DIRRD  8615 @@DIRWR  862A @@DIV16  8693
@@DIV8   867E @@DODIR  8279 @@DSP    7F9Ø
@@DSPLY  8Ø3Ø @@ERROR  81D1 @@EXIT   8192
@@FEXT   8582 @@FLAGS  8712 @@FNAME  8597
@@FSPEC  856D @@GATRD  86ØØ @@GATWR  863F
@@GET    7FA4 @@GTDCB  85C1 @@GTDCT  85AC
@@GTMOD  85D6 @@HDFMT  834B @@HEX16  86E7
@@HEX8   86D2 @@HEXDEC 86BD @@HIGH$  86FC
@@INIT   83C9 @@KBD    8ØØ8 @@KEY    7F7C
@@KEYIN  8Ø1C @@KLTSK  824F @@LOAD   8543
@@LOC    8432 @@LOF    8447 @@LOGER  8Ø67
@@LOGOT  8Ø7C @@MSG    8ØB3 @@MUL16  8669
@@MUL8   8654 @@OPEN   83DE @@PARAM  8129
@@PAUSE  8114 @@PEOF   845C @@POSN   8471
@@PRINT  8ØC8 @@PRT    7FEØ @@PUT    7FB8
@@RAMDIR 828E @@RDSEC  8321 @@RDSSC  85EB
@@READ   8486 @@REMOV  83B4 @@RENAM  839F
@@REW    849B @@RMTSK  8225 @@RPTSK  823A
@@RREAD  84BØ @@RSLCT  83ØC @@RSTOR  82CD
@@RUN    8558 @@RWRIT  84C5 @@SEEK   82F7
@@SEEKSC 84DA @@SKIP   84EF @@SLCT   82B8
@@STEPI  82E2 @@TIME   8153 @@VDCTL  8ØFF
@@VER    85Ø4 @@VRSEC  8336 @@WEOF   8519
@@WHERE  7FF4 @@WRITE  852E @@WRSEC  836Ø
@@WRSSC  8375 @@WRTRK  838A
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: RENAME

Library: SYS6/SYS

ISAM # : 53H

```
                00100 ;LBRENAME/ASM - RENAME Command
0000            00110         TITLE   <RENAME - LS-DOS 6.2>
                00120 ;
0000            00130 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
0000            00140 *GET    VALUES:3                ;Misc. equates
                03920 ;VALUES/ASM - Version 6
                03930 *LIST OFF
                04200 *LIST ON
                00150 ;
0000            00160 INH     EQU     0               ;Inhibit LRL Fault
                00170 ;
2400            00180         ORG     2400H
                00190 ;
2400            00200 RENAME  @@CKBRKC                ;Break key down?
2400 3E6A       00001         LD      A,106
2402 EF         00002         RST     40
2403 2804       00210         JR      Z,BEGINA        ;Ok if not
2405 21FFFF     00220         LD      HL,-1           ;  else abort
2408 C9         00230         RET
                00240 ;
2409 ED733224   00250 BEGINA  LD      (SAVESP+1),SP   ;Save SP
240D CD3824     00260         CALL    RENAM           ;Rename File/Device
2410 210000     00270         LD      HL,0            ;Init successful
2413 281C       00280         JR      Z,SAVESP        ;Z - successful rename
                00290 ;
                00300 ;       I/O Error Processing
                00310 ;
2415 6F         00320 IOERR   LD      L,A             ;Error # to HL
2416 2600       00330         LD      H,0
2418 F6C0       00340         OR      0C0H            ;Set to brief & return
241A 4F         00350         LD      C,A             ;Xfer error code
241B            00360         @@ERROR
241B 3E1A       00003         LD      A,26
241D EF         00004         RST     40
241E 1811       00370         JR      SAVESP          ;Restore stack & RET
                00380 ;
                00390 ;       Internal Message Error Processing
                00400 ;
2420 21C325     00410 SPCERR  LD      HL,SPCERR$
2423 DD         00420         DB      0DDH
2424 21D725     00430 DUPNAM  LD      HL,DUPNAM$
2427 DD         00440         DB      0DDH
2428 21EB25     00450 TOWHAT  LD      HL,TOWHAT$
242B            00460         @@LOGOT
                00005         IFEQ    00H,1
                00006         LD      HL,
                00007         ENDIF
242B 3E0C       00008         LD      A,12
242D EF         00009         RST     40
242E 21FFFF     00470         LD      HL,-1
                00480 ;
                00490 ;       Clean up stack & clear any pending <BREAK>s
                00500 ;
2431 310000     00510 SAVESP  LD      SP,$-$          ;P/u original SP
2434            00520         @@CKBRKC                ;Clear any <BREAK>
2434 3E6A       00010         LD      A,106
2436 EF         00011         RST     40
2437 C9         00530         RET
```

```
                    00540 ;
                    00550 ;       RENAM - Rename a filespec or devspec
                    00560 ;
2438 E5             00570 RENAM   PUSH    HL              ;Save cmd line ptr
2439 117526         00580         LD      DE,TEMPFCB      ;Xfer Filespec to buffer
243C                00590         @@FSPEC
243C 3E4E           00012         LD      A,78
243E EF             00013         RST     40
243F E1             00600         POP     HL              ;Ignore error
2440                00610         @@FLAGS                 ;IY => Flag Table
2440 3E65           00014         LD      A,101
2442 EF             00015         RST     40
2443 112326         00620         LD      DE,OLDFCB       ;Get filespec
2446                00630         @@FSPEC
2446 3E4E           00016         LD      A,78
2448 EF             00017         RST     40
2449 20D5           00640         JR      NZ,SPCERR       ;Quit if bad source name
244B 110326         00650         LD      DE,NEWFCB       ;Get new name
244E                00660         @@FSPEC
244E 3E4E           00018         LD      A,78
2450 EF             00019         RST     40
2451 C43C25         00670         CALL    NZ,CVRTUC       ;Cvrt partial spec to UC
2454 3A0326         00680 REN1    LD      A,(NEWFCB)      ;If new name starts out
2457 FE0F           00690         CP      CR+2            ;  with something less
2459 DA2824         00700         JP      C,TOWHAT        ;  than X'0E', to what ?
245C 212326         00710         LD      HL,OLDFCB
245F 110326         00720         LD      DE,NEWFCB
2462 7E             00730         LD      A,(HL)          ;Check on device rename
2463 FE2A           00740         CP      '*'
2465 CA0625         00750         JP      Z,DEVREN
2468 1A             00760         LD      A,(DE)          ;Old is file, new must
2469 FE2A           00770         CP      '*'             ;  be also
246B 28B3           00780         JR      Z,SPCERR
                    00790 ;
                    00800 ;       Renaming Files - Can we OPEN old file ?
                    00810 ;
246D 117526         00820         LD      DE,TEMPFCB      ;Can we OPEN it ?
2470 FDCB12C6       00830         SET     INH,(IY+SFLAG$) ;Inhibit open bit set
2474                00840         @@OPEN
2474 3E3B           00020         LD      A,59
2476 EF             00021         RST     40
2477 C0             00850         RET     NZ              ;NZ - "File not Found"
2478 ED4B7B26       00860         LD      BC,(TEMPFCB+6)  ;P/u drive #/DEC
                    00870 ;
                    00880 ;       Good Open - Is there a drivespec in string ?
                    00890 ;
247C E5             00900         PUSH    HL              ;Save ptr
247D 7E             00910 FLOOP   LD      A,(HL)          ;P/u char
247E FE0E           00920         CP      CR+1            ;End of Filespec ?
2480 300F           00930         JR      NC,CHKDSPC
                    00940 ;
                    00950 ;       Drivespec wasn't specified - put it on
                    00960 ;
2482 363A           00970         LD      (HL),':'        ;Append drivespec onto
2484 23             00980         INC     HL              ;  end of filespec
2485 79             00990         LD      A,C             ;Xfer drive # to A
2486 C630           01000         ADD     A,'0'           ;Convert to ASCII
2488 77             01010         LD      (HL),A
2489 32A924         01020         LD      (OLD_DRV+1),A   ;Self-modify NEW FCB
248C 23             01030         INC     HL              ;Bump
248D 360D           01040         LD      (HL),CR         ;End of filespec
```

```
248F 1809      01050          JR      DOMATCH         ;Get defaults
               01060 ;
               01070 ;        Stop when ":" hit or terminator
               01080 ;
2491 FE3A      01090 CHKDSPC  CP      ':'             ;Already have one ?
2493 23        01100          INC     HL
2494 20E7      01110          JR      NZ,FLOOP
2496 7E        01120          LD      A,(HL)          ;P/u drive #
2497 32A924    01130          LD      (OLD_DRV+1),A   ;Self-modify NEW FCB
249A E1        01140 DOMATCH  POP     HL              ;HL => Old FCB
249B 110326    01150          LD      DE,NEWFCB       ;DE => New FCB
249E CD6125    01160          CALL    MATCH
               01170 ;
               01180 ;        Make sure NEW drivespec is same as OLD one
               01190 ;
24A1 D5        01200          PUSH    DE              ;Save New
24A2 1A        01210 F2LOOP   LD      A,(DE)          ;Go until ":"
24A3 13        01220          INC     DE
24A4 FE3A      01230          CP      ':'
24A6 20FA      01240          JR      NZ,F2LOOP
24A8 3E00      01250 OLD_DRV  LD      A,$-$           ;P/u OLD drivespec
24AA 12        01260          LD      (DE),A          ;Overwrite
24AB D1        01270          POP     DE              ;Restore DE
               01280 ;
               01290 ;        Does the NEW filename already exist ?
               01300 ;
24AC E5        01310          PUSH    HL              ;Save OLD ptr
24AD D5        01320          PUSH    DE              ;Save NEW ptr
24AE EB        01330          EX      DE,HL
24AF 117526    01340          LD      DE,TEMPFCB      ;DE => Temp buffer
24B2 FDCB12C6  01350          SET     INH,(IY+SFLAG$)
24B6           01360          @@FSPEC                 ;Xfer filespec
24B6 3E4E      00022          LD      A,78
24B8 EF        00023          RST     40
24B9           01370          @@OPEN                  ;File already exist ?
24B9 3E3B      00024          LD      A,59
24BB EF        00025          RST     40
24BC CA2424    01380          JP      Z,DUPNAM        ;Error if so
24BF D1        01390          POP     DE              ;Restore ptrs
24C0 E1        01400          POP     HL
24C1 E5        01410 REN2     PUSH    HL              ;OLD Filename/Device
24C2 D5        01420          PUSH    DE              ;NEW Filename/Device
               01430 ;
               01440 ;        Xfer the OLD & NEW specs to SPEC$ minus PASSWORD
               01450 ;
24C3 114D26    01460          LD      DE,SPECS$
24C6 CDE524    01470          CALL    MOVSPC          ;Move the OLD spec
24C9 21FE25    01480          LD      HL,TO$
24CC 010400    01490          LD      BC,4
24CF EDB0      01500          LDIR                    ;Move ' to '
24D1 E1        01510          POP     HL              ;Recover NEW spec
24D2 E5        01520          PUSH    HL
24D3 CDE524    01530          CALL    MOVSPC          ;Move the NEW spec
24D6 3E0D      01540          LD      A,CR
24D8 12        01550          LD      (DE),A          ;Terminate with CR
24D9           01560          @@LOGOT RENAM$          ;Send names to video
               00026          IFEQ    01H,1
24D9 214326    00027          LD      HL,RENAM$
               00028          ENDIF
24DC 3E0C      00029          LD      A,12
24DE EF        00030          RST     40
```

```
24DF E1       01570          POP     HL              ;Recover new
24E0 D1       01580          POP     DE              ;Recover old
24E1          01590          @@RENAM                 ;Rename file
24E1 3E38     00031          LD      A,56
24E3 EF       00032          RST     40
24E4 C9       01600          RET                     ;Return with condition
              01610 ;
              01620 ;        MOVSPC - Create Secondary Spec
              01630 ;
24E5 7E       01640 MOVSPC   LD      A,(HL)          ;P/u a spec character
24E6 FE2F     01650          CP      '/'             ;Extension ?
24E8 2008     01660          JR      NZ,CKSPACE      ;No - check if space
24EA 23       01670          INC     HL              ;Is the next character
24EB 7E       01680          LD      A,(HL)          ;  valid ?
24EC FE41     01690          CP      'A'
24EE 3802     01700          JR      C,CKSPACE       ;No - don't output it
24F0 2B       01710          DEC     HL              ;Back one
24F1 7E       01720          LD      A,(HL)          ;P/u slash
24F2 FE20     01730 CKSPACE  CP      ' '
24F4 D8       01740          RET     C               ;Exit on terminator
24F5 FE2E     01750          CP      '.'             ;If password, ignore it
24F7 2009     01760          JR      NZ,MOVSPC1
24F9 23       01770 SKIPPW   INC     HL
24FA 7E       01780          LD      A,(HL)
24FB FE20     01790          CP      ' '
24FD D8       01800          RET     C               ;Back on terminator
24FE FE3A     01810          CP      ':'
2500 20F7     01820          JR      NZ,SKIPPW
2502 EDA0     01830 MOVSPC1  LDI                     ;Move the char
2504 18DF     01840          JR      MOVSPC
              01850 ;
              01860 ;        Routine to rename a device
              01870 ;
2506 1A       01880 DEVREN   LD      A,(DE)          ;Old was device, new must
2507 FE2A     01890          CP      '*'             ;  also be a device spec
2509 C22024   01900          JP      NZ,SPCERR       ;Abort if bad
              01910 ;
              01920 ;        Does the Source Devspec exist ?
              01930 ;
250C E5       01940          PUSH    HL              ;Save Old Device name
250D D5       01950          PUSH    DE              ;Save New Device name
250E 23       01960          INC     HL              ;Bump past "*"
250F 5E       01970          LD      E,(HL)          ;Set DE = Device name
2510 23       01980          INC     HL
2511 56       01990          LD      D,(HL)
2512          02000          @@GTDCB                 ;Does it exist ?
2512 3E52     00033          LD      A,82
2514 EF       00034          RST     40
2515 C21524   02010          JP      NZ,IOERR        ;NZ - "Dev not Available"
              02020 ;
              02030 ;        P/u the Job Log DCB Address (last DCB)
              02040 ;
2518 44       02050          LD      B,H             ;Save DCB ptr in BC
2519 4D       02060          LD      C,L
251A 114A4C   02070          LD      DE,'LJ'         ;Find *JL
251D          02080          @@GTDCB
251D 3E52     00035          LD      A,82
251F EF       00036          RST     40
2520 23       02090          INC     HL              ;Pt HL => Past Protected
2521 B7       02100          OR      A               ;  system Device table.
2522 ED42     02110          SBC     HL,BC           ;Protected Device ?
```

```
2524 3E28      Ø212Ø          LD      A,4Ø            ;Init errcode
2526 D21524    Ø213Ø          JP      NC,IOERR        ;Jump on error
               Ø214Ø ;
               Ø215Ø ;        Does the destination device already exist ?
               Ø216Ø ;
2529 E1        Ø217Ø          POP     HL              ;HL => New Devspec
252A E5        Ø218Ø          PUSH    HL
252B 23        Ø219Ø          INC     HL              ;Bump past "*"
252C 5E        Ø22ØØ          LD      E,(HL)          ;Set DE = Device name
252D 23        Ø221Ø          INC     HL
252E 56        Ø222Ø          LD      D,(HL)
252F           Ø223Ø          @@GTDCB                 ;Already Exist ?
252F 3E52      ØØØ37          LD      A,82
2531 EF        ØØØ38          RST     4Ø
2532 3E27      Ø224Ø          LD      A,39            ;Yes - Device in use
2534 CA1524    Ø225Ø          JP      Z,IOERR
2537 D1        Ø226Ø          POP     DE              ;Restore NEW & OLD ptrs
2538 E1        Ø227Ø          POP     HL
2539 C3C124    Ø228Ø          JP      REN2
               Ø229Ø ;
               Ø23ØØ ;        Routine xfers partial filespec & cvrts to UC
               Ø231Ø ;
253C 7E        Ø232Ø CVRTUC   LD      A,(HL)
253D FEØD      Ø233Ø          CP      CR
253F C8        Ø234Ø          RET     Z               ;Ret if no new name
254Ø 2B        Ø235Ø          DEC     HL              ;Backup to 1st separator
2541 7E        Ø236Ø COPØ     LD      A,(HL)
2542 23        Ø237Ø          INC     HL
2543 FE2Ø      Ø238Ø          CP      ' '             ;Skip past spaces
2545 28FA      Ø239Ø          JR      Z,COPØ
2547 2B        Ø24ØØ          DEC     HL
2548 Ø62Ø      Ø241Ø          LD      B,32            ;Max 32 chars
254A 7E        Ø242Ø COP1     LD      A,(HL)          ;Transfer the partial
254B FE61      Ø243Ø COP2     CP      'a'             ;Cvrt lc <a-z> to uc
254D 38Ø6      Ø244Ø          JR      C,COP3
254F FE7B      Ø245Ø          CP      'z'+1
2551 3ØØ2      Ø246Ø          JR      NC,COP3
2553 D62Ø      Ø247Ø          SUB     2ØH
2555 12        Ø248Ø COP3     LD      (DE),A          ;Filespec until paren
2556 FEØD      Ø249Ø          CP      CR              ;  or <ENTER>
2558 C8        Ø25ØØ          RET     Z
2559 FE28      Ø251Ø          CP      '('
255B C8        Ø252Ø          RET     Z
255C 23        Ø253Ø          INC     HL              ;  or end-of-line
255D 13        Ø254Ø          INC     DE              ;  or 32 chars max
255E 1ØEA      Ø255Ø          DJNZ    COP1
256Ø C9        Ø256Ø          RET
               Ø257Ø ;
               Ø258Ø ;        Match source & destination for defaults
               Ø259Ø ;
2561 D5        Ø26ØØ MATCH    PUSH    DE              ;Save NEW spec
2562 E5        Ø261Ø          PUSH    HL              ;Save OLD spec
2563 1A        Ø262Ø          LD      A,(DE)          ;P/u a dest character
2564 FE41      Ø263Ø          CP      'A'
2566 DCA325    Ø264Ø          CALL    C,MATCH7        ;Match if not a filename
2569 Ø62F      Ø265Ø          LD      B,'/'
256B CD7C25    Ø266Ø          CALL    MATCH2
256E Ø63A      Ø267Ø          LD      B,':'
257Ø CD7C25    Ø268Ø          CALL    MATCH2
2573 Ø62E      Ø269Ø          LD      B,'.'
2575 CD7C25    Ø27ØØ          CALL    MATCH2
```

```
2578 E1        Ø271Ø          POP     HL
2579 D1        Ø272Ø          POP     DE
257A C9        Ø273Ø          RET
               Ø274Ø ;
257B 13        Ø275Ø MATCH1   INC     DE
257C 1A        Ø276Ø MATCH2   LD      A,(DE)          ;Scan destination until
257D B8        Ø277Ø          CP      B               ;  the test character is
257E 28ØE      Ø278Ø          JR      Z,MATCH3        ;  found or until some
258Ø FE41      Ø279Ø          CP      'A'             ;  other special char
2582 3ØF7      Ø28ØØ          JR      NC,MATCH1       ;  is reached
2584 FE3Ø      Ø281Ø          CP      'Ø'             ;Loop on <Ø-9>
2586 38Ø8      Ø282Ø          JR      C,MATCH4
2588 FE3A      Ø283Ø          CP      '9'+1
258A 38EF      Ø284Ø          JR      C,MATCH1
258C 18Ø2      Ø285Ø          JR      MATCH4
258E 13        Ø286Ø MATCH3   INC     DE
258F C9        Ø287Ø          RET
               Ø288Ø ;
               Ø289Ø ;        Found some other special char - Need the field
               Ø29ØØ ;
2590 E5        Ø291Ø MATCH4   PUSH    HL              ;Save pointer to source
2591 7E        Ø292Ø MATCH5   LD      A,(HL)          ;Scan source until the
2592 23        Ø293Ø          INC     HL              ;  desired field is
2593 FEØ3      Ø294Ø          CP      ETX             ;  found (if it is
2595 28ØA      Ø295Ø          JR      Z,MATCH6        ;  supplied by the user)
2597 FEØD      Ø296Ø          CP      CR
2599 28Ø6      Ø297Ø          JR      Z,MATCH6
259B B8        Ø298Ø          CP      B
259C 2ØF3      Ø299Ø          JR      NZ,MATCH5
259E CDAF25    Ø3ØØØ          CALL    MATCH9          ;Move source field
25A1 E1        Ø3Ø1Ø MATCH6   POP     HL
25A2 C9        Ø3Ø2Ø          RET
               Ø3Ø3Ø ;
               Ø3Ø4Ø ;        Routines to move a source field to destination
               Ø3Ø5Ø ;
25A3 7E        Ø3Ø6Ø MATCH7   LD      A,(HL)          ;P/u source character
25A4 FE3Ø      Ø3Ø7Ø          CP      'Ø'             ;Back when out of range
25A6 D8        Ø3Ø8Ø          RET     C
25A7 FE3A      Ø3Ø9Ø          CP      '9'+1
25A9 38Ø3      Ø31ØØ          JR      C,MATCH8
25AB FE41      Ø311Ø          CP      'A'
25AD D8        Ø312Ø          RET     C
25AE 23        Ø313Ø MATCH8   INC     HL              ;Advance source ptr
25AF E5        Ø314Ø MATCH9   PUSH    HL              ;Save HL and make it
25BØ 62        Ø315Ø          LD      H,D             ;  the destination ptr
25B1 6B        Ø316Ø          LD      L,E
25B2 4E        Ø317Ø MATCH1Ø  LD      C,(HL)          ;Get char at destination
25B3 77        Ø318Ø          LD      (HL),A          ;  and put in new one
25B4 23        Ø319Ø          INC     HL              ;Next dest loc.
25B5 79        Ø32ØØ          LD      A,C             ;What was there?
25B6 FEØ3      Ø321Ø          CP      ETX             ;Go until ETX
25B8 28Ø4      Ø322Ø          JR      Z,MATCH11
25BA FEØD      Ø323Ø          CP      CR              ;  or end of line
25BC 2ØF4      Ø324Ø          JR      NZ,MATCH1Ø
25BE 77        Ø325Ø MATCH11  LD      (HL),A
25BF E1        Ø326Ø          POP     HL
25CØ 13        Ø327Ø          INC     DE
25C1 18EØ      Ø328Ø          JR      MATCH7
               Ø329Ø ;
25C3 53        Ø33ØØ SPCERR$  DB      'Specification error',CR
     7Ø 65 63 69 66 69 63 61
```

```
     74 69 6F 6E 2Ø 65 72 72
     6F 72 ØD
25D7 44        Ø331Ø DUPNAM$ DB      'Duplicate file name',CR
     75 7Ø 6C 69 63 61 74 65
     2Ø 66 69 6C 65 2Ø 6E 61
     6D 65 ØD
25EB 52        Ø332Ø TOWHAT$ DB      'Rename it to what?',CR
     65 6E 61 6D 65 2Ø 69 74
     2Ø 74 6F 2Ø 77 68 61 74
     3F ØD
25FE 2Ø        Ø333Ø TO$     DB      ' to ',ETX
     74 6F 2Ø Ø3
2603 ØD        Ø334Ø NEWFCB  DB      CR              ;Init to cr
ØØ1F           Ø335Ø         DS      31
ØØ2Ø           Ø336Ø OLDFCB  DS      32
2643 52        Ø337Ø RENAM$  DB      'Renaming: '
     65 6E 61 6D 69 6E 67 3A
     2Ø
ØØ28           Ø338Ø SPECS$  DS      4Ø
ØØ2Ø           Ø339Ø TEMPFCB DS      32
2695 ØØØØ      Ø34ØØ OLD_FIL DW      Ø
2697           Ø341Ø LAST    EQU     $
               Ø342Ø ;
24ØØ           Ø343Ø         END     RENAME
```

@@1	ØØØØ	@@2	ØØØØ	@@3	ØØØØ	@@4	ØØØØ
@MOD2	ØØØØ	@MOD4	FFFF	ABB	ØØ1Ø	AP	ØØ27
BEGINA	24Ø9	BREAK	ØØ8Ø	BS	ØØØ8	CFLAG$	ØØØ2
CHKDSPC	2491	CKSPACE	24F2	COPØ	2541	COP1	254A
COP2	254B	COP3	2555	CR	ØØØD	CVRTUC	253C
DEVREN	25Ø6	DFLAG$	ØØØ3	DOMATCH	249A	DUPNAM	2424
DUPNAM$	25D7	ETX	ØØØ3	F2LOOP	24A2	FLAG	ØØ4Ø
FLOOP	247D	INH	ØØØØ	IOERR	2415	KFLAG$	ØØØA
LAST	2697	LF	ØØØA	MATCH	2561	MATCH1	257B
MATCH1Ø	25B2	MATCH11	25BE	MATCH2	257C	MATCH3	258E
MATCH4	259Ø	MATCH5	2591	MATCH6	25A1	MATCH7	25A3
MATCH8	25AE	MATCH9	25AF	MOVSPC	24E5	MOVSPC1	25Ø2
NEWFCB	26Ø3	NUM	ØØ8Ø	OLDFCB	2623	OLD_DRV	24A8
OLD_FIL	2695	PAR_ERR	ØØ2C	REN1	2454	REN2	24C1
RENAM	2438	RENAM$	2643	RENAME	24ØØ	SAVESP	2431
SFLAG$	ØØ12	SKIPPW	24F9	SPCERR	242Ø	SPCERR$	25C3
SPECS$	264D	STR	ØØ2Ø	TAB	ØØØ9	TEMPFCB	2675
TO$	25FE	TOWHAT	2428	TOWHAT$	25EB	VFLAG$	ØØ15
@@ABORT	89ØE	@@ADTSK	89A1	@@BANK	8EB9	@@BKSP	8B99
@@BREAK	8ECF	@@CHNIO	88F9	@@CKBRKC	8F1D	@@CKDRV	89F5
@@CKEOF	8BAE	@@CKTSK	898C	@@CLOSE	8B84	@@CLS	8FØ7
@@CMNDI	8938	@@CMNDR	894D	@@CTL	875D	@@DATE	88CF
@@DCSTAT	8A34	@@DEBUG	8977	@@DECHEX	8E39	@@DIRRD	8DA6
@@DIRWR	8DBB	@@DIV16	8E24	@@DIV8	8EØF	@@DODIR	8AØA
@@DSP	8721	@@DSPLY	87C1	@@ERROR	8962	@@EXIT	8923
@@FEXT	8D13	@@FLAGS	8EA3	@@FNAME	8D28	@@FSPEC	8CFE
@@GATRD	8D91	@@GATWR	8DDØ	@@GET	8735	@@GTDCB	8D52
@@GTDCT	8D3D	@@GTMOD	8D67	@@HDFMT	8ADC	@@HEX16	8E78
@@HEX8	8E63	@@HEXDEC	8E4E	@@HIGH$	8E8D	@@INIT	8B5A
@@KBD	8799	@@KEY	87ØD	@@KEYIN	87AD	@@KLTSK	89EØ
@@LOAD	8CD4	@@LOC	8BC3	@@LOF	8BD8	@@LOGER	87F8
@@LOGOT	88ØD	@@MSG	8844	@@MUL16	8DFA	@@MUL8	8DE5
@@OPEN	8B6F	@@PARAM	88BA	@@PAUSE	88A5	@@PEOF	8BED
@@POSN	8CØ2	@@PRINT	8859	@@PRT	8771	@@PUT	8749
@@RAMDIR	8A1F	@@RDSEC	8AB2	@@RDSSC	8D7C	@@READ	8C17
@@REMOV	8B45	@@RENAM	8B3Ø	@@REW	8C2C	@@RMTSK	89B6
@@RPTSK	89CB	@@RREAD	8C41	@@RSLCT	8A9D	@@RSTOR	8A5E
@@RUN	8CE9	@@RWRIT	8C56	@@SEEK	8A88	@@SEEKSC	8C6B
@@SKIP	8C8Ø	@@SLCT	8A49	@@STEPI	8A73	@@TIME	88E4
@@VDCTL	889Ø	@@VER	8C95	@@VRSEC	8AC7	@@WEOF	8CAA
@@WHERE	8785	@@WRITE	8CBF	@@WRSEC	8AF1	@@WRSSC	8BØ6
@@WRTRK	8B1B						

24ØØ is the transfer address
ØØØØØ Total errors

NOTES:

Command: RESET

Library: SYS6/SYS

ISAM # : 63H

```
                00100 ;LBRESET/ASM - RESET Command
0000            00110         TITLE   <RESET - LS-DOS 6.2>
                00120 ;
000D            00130 CR      EQU     13
0000            00140 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00150 ;
2400            00160         ORG     2400H
                00170 ;
                00180 ;       Save stack & call reset code
                00190 ;
2400 ED732124   00200 RESET   LD      (SAVESP+1),SP   ;Save Stack ptr
2404 CD2724     00210         CALL    RESET1          ;Call Reset routine
2407 210000     00220         LD      HL,0            ;Init successful
240A 2814       00230         JR      Z,SAVESP        ;Z - Exit clean
                00240 ;
                00250 ;       I/O Error Processing
                00260 ;
240C 6F         00270 IOERR   LD      L,A             ;Error # to HL
240D 2600       00280         LD      H,0
240F F6C0       00290         OR      0C0H            ;Abbrev, return
2411 4F         00300         LD      C,A
2412            00310         @@ERROR
2412 3E1A       00001         LD      A,26
2414 EF         00002         RST     40
2415 1809       00320         JR      SAVESP          ;P/u stack & return
                00330 ;
                00340 ;       Internal Message Exit
                00350 ;
2417 21F824     00360 SPCREQ  LD      HL,SPCREQ$      ;"Devspec req...
241A            00370         @@LOGOT
                00003         IFEQ    00H,1
                00004         LD      HL,
                00005         ENDIF
241A 3E0C       00006         LD      A,12
241C EF         00007         RST     40
241D 21FFFF     00380         LD      HL,-1           ;Set abort code
                00390 ;
                00400 ;       P/u stack & clear any pending <BREAK>
                00410 ;
2420 310000     00420 SAVESP  LD      SP,$-$          ;P/u stack ptr
2423            00430         @@CKBRKC                ;Clear any Break
2423 3E6A       00008         LD      A,106
2425 EF         00009         RST     40
2426 C9         00440         RET
                00450 ;
                00460 ;       RESET1 - Reset a Filespec or Devspec
                00470 ;
2427 112D25     00480 RESET1  LD      DE,FCBDEV       ;Get file/device spec
242A            00490         @@FSPEC
242A 3E4E       00010         LD      A,78
242C EF         00011         RST     40
242D C21724     00500         JP      NZ,SPCREQ       ;Must reset something
2430 1A         00510         LD      A,(DE)          ;File reset used to
2431 FE2A       00520         CP      '*'             ;  reset the
2433 C26624     00530         JP      NZ,RESFIL       ;  "file open bit
2436 ED5B2E25   00540         LD      DE,(FCBDEV+1)   ;P/u the device name
243A            00550         @@GTDCB                 ;Find in device tables
243A 3E52       00012         LD      A,82
```

```
243C EF          00013          RST     40
243D C0          00560          RET     NZ              ;NZ - "Device not Avail"
243E E5          00570          PUSH    HL              ;Save pointer to table
243F CD8124      00580          CALL    CLSFILS         ;Reset routes to files
2442 E1          00590          POP     HL              ;Get DCB pointer
2443 C0          00600          RET     NZ              ;NZ - I/O Error
                 00610 ;
                 00620 ;        Unhook the device chain
                 00630 ;
2444 E5          00640          PUSH    HL
2445 F3          00650          DI                      ;Don't stop me now
2446 CD9D24      00660          CALL    FIXDCB          ;Fixup the DCB
2449 FB          00670          EI                      ;Now you can interrupt
244A 212D25      00680          LD      HL,FCBDEV       ;Determine if system
244D 54          00690          LD      D,H             ;  device by attempting
244E 5D          00700          LD      E,L             ;  to rename it
244F             00710          @@RENAM                 ;The error code will be
244F 3E38        00014          LD      A,56
2451 EF          00015          RST     40
2452 E1          00720          POP     HL              ;  either 19 or 40
2453 FE28        00730          CP      40              ;Protected system device?
2455 2804        00740          JR      Z,SYSDVC
2457 3608        00750          LD      (HL),8          ;Show device is NIL
2459 AF          00760          XOR     A               ;Set Z
245A C9          00770          RET                     ;Z - successful
                 00780 ;
                 00790 ;        RESET of system device
                 00800 ;
245B E5          00810 SYSDVC   PUSH    HL              ;Save DCB ptr
245C 2C          00820          INC     L               ;If DCB vector is X'0000'
245D 7E          00830          LD      A,(HL)          ;  then do NOT reset the
245E 2C          00840          INC     L               ;  NIL bit
245F B6          00850          OR      (HL)
2460 E1          00860          POP     HL
2461 C8          00870          RET     Z               ;Z - return
2462 CB9E        00880          RES     3,(HL)          ;Make sure NIL is off
2464 AF          00890          XOR     A               ;Set Z & Return
2465 C9          00900          RET
                 00910 ;
                 00920 ;        Reset the "file open bit" of a file
                 00930 ;
2466             00940 RESFIL   @@FLAGS                 ;Get flag table pointer
2466 3E65        00016          LD      A,101
2468 EF          00017          RST     40
2469 FDCB12C6    00950          SET     0,(IY+'S'-'A')  ;Inhibit file open bit
246D             00960          @@OPEN
246D 3E3B        00018          LD      A,59
246F EF          00019          RST     40
2470 C0          00970          RET     NZ              ;NZ - I/O Error
2471 7E          00980          LD      A,(HL)          ;Make sure access level
2472 E607        00990          AND     7               ;  is at least UPDATE
2474 FE05        01000          CP      5
2476 3E25        01010          LD      A,37            ;Init "Illegal access...
2478 C0          01020          RET     NZ              ;NZ - I/O error
2479 EB          01030          EX      DE,HL
247A CBF6        01040          SET     6,(HL)          ;Set "close authority
247C EB          01050          EX      DE,HL           ;  to reset dir bit
247D             01060          @@CLOSE
247D 3E3C        00020          LD      A,60
247F EF          00021          RST     40
2480 C9          01070          RET                     ;Return w/ condition
```

```
                    01080 ;
                    01090 ;       Find the last device route & close any open file
                    01100 ;
2481 CB66           01110 CLSFILS BIT     4,(HL)          ;Jump if no route
2483 2807           01120         JR      Z,CLSFIL1
2485 23             01130         INC     HL              ;Else p/u link address
2486 7E             01140         LD      A,(HL)          ;  and test that one
2487 23             01150         INC     HL              ;  for a chain
2488 66             01160         LD      H,(HL)
2489 6F             01170         LD      L,A
248A 18F5           01180         JR      CLSFILS
248C CB7E           01190 CLSFIL1 BIT     7,(HL)          ;A file?
248E C8             01200         RET     Z               ;Ret if not
248F 114D25         01210         LD      DE,FCBFIL       ;Pt to fcb area
2492 D5             01220         PUSH    DE
2493 012000         01230         LD      BC,32
2496 EDB0           01240         LDIR                    ;Fill from device vector
2498 D1             01250         POP     DE              ;Recover start
2499                01260         @@CLOSE                 ;Close the file
2499 3E3C           00022         LD      A,60
249B EF             00023         RST     40
249C C9             01270         RET                     ;Ret with Z, NZ status
                    01280 ;
                    01290 ;       Routine to fix up a system DCB
                    01300 ;
249D CB66           01310 FIXDCB  BIT     4,(HL)          ;If routed, recover the
249F 2810           01320         JR      Z,FIX1          ;  original data from
24A1 E5             01330         PUSH    HL              ;  DCB+3 to DCB+5
24A2 54             01340         LD      D,H             ;DCB start to DE
24A3 5D             01350         LD      E,L
24A4 2C             01360 FIX0    INC     L               ;Pt to old stored
24A5 2C             01370         INC     L               ;  information
24A6 2C             01380         INC     L
24A7 010300         01390         LD      BC,3
24AA EDB0           01400         LDIR                    ;Xfer to DCB
24AC E1             01410         POP     HL
24AD CBA6           01420         RES     4,(HL)          ;Reset Routed bit
24AF 18EC           01430         JR      FIXDCB
24B1 CB6E           01440 FIX1    BIT     5,(HL)          ;If linked, recover the
24B3 2819           01450         JR      Z,FIX2          ;  original data from
24B5 E5             01460         PUSH    HL              ;  the LINK DCB source &
24B6 2C             01470         INC     L               ;  clear the LINK DCB
24B7 5E             01480         LD      E,(HL)          ;P/u the LINK vector
24B8 2C             01490         INC     L
24B9 56             01500         LD      D,(HL)
24BA E1             01510         POP     HL              ;Recover DCB ptr
24BB E5             01520         PUSH    HL
24BC EB             01530         EX      DE,HL           ;Link to HL, DCB to DE
24BD E5             01540         PUSH    HL              ;Save for clearing
24BE 010300         01550         LD      BC,3
24C1 EDB0           01560         LDIR
24C3 E1             01570         POP     HL
24C4 0608           01580         LD      B,8
24C6 3600           01590 FIX1A   LD      (HL),0          ;Clear the LINK DCB
24C8 2C             01600         INC     L
24C9 10FB           01610         DJNZ    FIX1A
24CB E1             01620         POP     HL              ;Rcvr DCB pointer
24CC 18CF           01630         JR      FIXDCB
24CE CB76           01640 FIX2    BIT     6,(HL)          ;If filtered, recover the
24D0 C8             01650         RET     Z               ;  original data by
24D1 E5             01660         PUSH    HL              ;  swapping back the
```

```
24D2 54          Ø167Ø          LD      D,H
24D3 5D          Ø168Ø          LD      E,L
24D4 2C          Ø169Ø          INC     L               ;  1st three bytes with
24D5 7E          Ø17ØØ          LD      A,(HL)          ;  the FILTER DCB
24D6 2C          Ø171Ø          INC     L
24D7 66          Ø172Ø          LD      H,(HL)
24D8 6F          Ø173Ø          LD      L,A
24D9 Ø1Ø4ØØ      Ø174Ø          LD      BC,4            ;HL now points to the
24DC Ø9          Ø175Ø          ADD     HL,BC           ;  entry point. Get its
24DD 4E          Ø176Ø          LD      C,(HL)          ;  DCB address by peeking
24DE ØC          Ø177Ø          INC     C               ;  past the name field
24DF Ø9          Ø178Ø          ADD     HL,BC
24EØ 7E          Ø179Ø          LD      A,(HL)          ;Get low-order
24E1 23          Ø18ØØ          INC     HL
24E2 66          Ø181Ø          LD      H,(HL)          ;Get hi-order
24E3 6F          Ø182Ø          LD      L,A
24E4 E5          Ø183Ø          PUSH    HL              ;If DCB is itself, then
24E5 ED52        Ø184Ø          SBC     HL,DE           ;  bring in the NIL
24E7 E1          Ø185Ø          POP     HL
24E8 28BA        Ø186Ø          JR      Z,FIXØ
24EA Ø6Ø3        Ø187Ø          LD      B,3             ;  else swap the 1st three
24EC 4E          Ø188Ø FIX2A    LD      C,(HL)          ;  bytes of the DCBs
24ED 1A          Ø189Ø          LD      A,(DE)
24EE 77          Ø19ØØ          LD      (HL),A
24EF 79          Ø191Ø          LD      A,C
24FØ 12          Ø192Ø          LD      (DE),A
24F1 2C          Ø193Ø          INC     L
24F2 1C          Ø194Ø          INC     E
24F3 1ØF7        Ø195Ø          DJNZ    FIX2A
24F5 E1          Ø196Ø          POP     HL
24F6 18A5        Ø197Ø          JR      FIXDCB
                 Ø198Ø ;
                 Ø199Ø ;        Data area
                 Ø2ØØØ ;
24F8 44          Ø2Ø1Ø SPCREQ$ DB      'Device spec required',CR
     65 76 69 63 65 2Ø 73 7Ø
     65 63 2Ø 72 65 71 75 69
     72 65 64 ØD
25ØD ØØ          Ø2Ø2Ø          DC      32,Ø            ;Patch space
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ
252D ØØ          Ø2Ø3Ø FCBDEV  DB      Ø
ØØ1F             Ø2Ø4Ø          DS      31
254D ØØ          Ø2Ø5Ø FCBFIL  DB      Ø
ØØ1F             Ø2Ø6Ø          DS      31
                 Ø2Ø7Ø ;
24ØØ             Ø2Ø8Ø          END     RESET
```

```
@@1             ØØØØ @@2             ØØØØ @@3             ØØØØ
@@4             ØØØØ @MOD2           ØØØØ @MOD4           FFFF
CLSFIL1         248C CLSFILS         2481 CR              ØØØD
FCBDEV          252D FCBFIL          254D FIXØ            24A4
FIX1            24B1 FIX1A           24C6 FIX2            24CE
FIX2A           24EC FIXDCB          249D IOERR           24ØC
RESET           24ØØ RESET1          2427 RESFIL          2466
SAVESP          242Ø SPCREQ          2417 SPCREQ$         24F8
SYSDVC          245B @@ABORT         7D6A @@ADTSK         7DFD
@@BANK          8315 @@BKSP          7FF5 @@BREAK         832B
@@CHNIO         7D55 @@CKBRKC        8379 @@CKDRV         7E51
@@CKEOF         8ØØA @@CKTSK         7DE8 @@CLOSE         7FEØ
@@CLS           8363 @@CMNDI         7D94 @@CMNDR         7DA9
@@CTL           7BB9 @@DATE          7D2B @@DCSTAT        7E9Ø
@@DEBUG         7DD3 @@DECHEX        8295 @@DIRRD         82Ø2
@@DIRWR         8217 @@DIV16         828Ø @@DIV8          826B
@@DODIR         7E66 @@DSP           7B7D @@DSPLY         7C1D
@@ERROR         7DBE @@EXIT          7D7F @@FEXT          816F
@@FLAGS         82FF @@FNAME         8184 @@FSPEC         815A
@@GATRD         81ED @@GATWR         822C @@GET           7B91
@@GTDCB         81AE @@GTDCT         8199 @@GTMOD         81C3
@@HDFMT         7F38 @@HEX16         82D4 @@HEX8          82BF
@@HEXDEC        82AA @@HIGH$         82E9 @@INIT          7FB6
@@KBD           7BF5 @@KEY           7B69 @@KEYIN         7CØ9
@@KLTSK         7E3C @@LOAD          813Ø @@LOC           8Ø1F
@@LOF           8Ø34 @@LOGER         7C54 @@LOGOT         7C69
@@MSG           7CAØ @@MUL16         8256 @@MUL8          8241
@@OPEN          7FCB @@PARAM         7D16 @@PAUSE         7DØ1
@@PEOF          8Ø49 @@POSN          8Ø5E @@PRINT         7CB5
@@PRT           7BCD @@PUT           7BA5 @@RAMDIR        7E7B
@@RDSEC         7FØE @@RDSSC         81D8 @@READ          8Ø73
@@REMOV         7FA1 @@RENAM         7F8C @@REW           8Ø88
@@RMTSK         7E12 @@RPTSK         7E27 @@RREAD         8Ø9D
@@RSLCT         7EF9 @@RSTOR         7EBA @@RUN           8145
@@RWRIT         8ØB2 @@SEEK          7EE4 @@SEEKSC        8ØC7
@@SKIP          8ØDC @@SLCT          7EA5 @@STEPI         7ECF
@@TIME          7D4Ø @@VDCTL         7CEC @@VER           8ØF1
@@VRSEC         7F23 @@WEOF          81Ø6 @@WHERE         7BE1
@@WRITE         811B @@WRSEC         7F4D @@WRSSC         7F62
@@WRTRK         7F77
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

Command: ROUTE

Library: SYS6/SYS

ISAM # : 64H

```
                00100 ;LBROUTE/ASM - ROUTE Command
0000            00110         TITLE   <ROUTE - LS-DOS 6.2>
                00120 ;
000D            00130 CR      EQU     13
002C            00140 PAR_ERR EQU     44              ;Parameter Error
                00150 ;
0000            00160 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00170 ;
2400            00180         ORG     2400H
                00190 ;
                00200 ;       Save stack & call Route routine
                00210 ;
2400 ED732724   00220 ROUTE   LD      (SAVESP+1),SP   ;Save Stack
2404 CD2D24     00230         CALL    ROUTE1          ;Call route routine
2407 210000     00240 EXIT    LD      HL,0            ;Clean Exit
240A 181A       00250         JR      SAVESP
                00260 ;
                00270 ;       I/O Error Handling
                00280 ;
240C 3E2C       00290 PRMERR  LD      A,PAR_ERR       ;Parameter Error
240E 6F         00300 IOERR   LD      L,A
240F 2600       00310         LD      H,0
2411 F6C0       00320         OR      0C0H            ;Set abbrev & return
2413 4F         00330         LD      C,A
2414            00340         @@ERROR
2414 3E1A       00001         LD      A,26
2416 EF         00002         RST     40
2417 180D       00350         JR      SAVESP          ;P/u stack & return
                00360 ;
                00370 ;       Internal Error Message Handling
                00380 ;
2419 215525     00390 CANT    LD      HL,CANT$        ;"No mem space...
241C DD         00400         DB      0DDH
241D 216F25     00410 SPCERR  LD      HL,SPCERR$      ;"Devspec req...
2420            00420         @@LOGOT
                00003         IFEQ    00H,1
                00004         LD      HL,
                00005         ENDIF
2420 3E0C       00006         LD      A,12
2422 EF         00007         RST     40
2423 21FFFF     00430 ERREXIT LD      HL,-1           ;Set abort code
                00440 ;
                00450 ;       P/u stack & Clear any pending <BREAK>
                00460 ;
2426 310000     00470 SAVESP  LD      SP,$-$          ;P/u stack
2429            00480         @@CKBRKC                ;Clear any Break
2429 3E6A       00008         LD      A,106
242B EF         00009         RST     40
242C C9         00490         RET
                00500 ;
                00510 ;       ROUTE1 - Route spec to spec
                00520 ;
242D 11E525     00530 ROUTE1  LD      DE,FCBSRC       ;Fetch source spec
2430            00540         @@FSPEC
2430 3E4E       00010         LD      A,78
2432 EF         00011         RST     40
2433 2026       00550         JR      NZ,SPCER        ;Jump on error
2435 1A         00560         LD      A,(DE)
```

```
2436 FE2A     00570          CP      '*'             ;Must be a device
2438 2021     00580          JR      NZ,SPCER        ;Jump if not
243A 118425   00590          LD      DE,PRMTBL$      ;Get parameters
243D          00600          @@PARAM
243D 3E11     00012          LD      A,17
243F EF       00013          RST     40
2440 20CA     00610          JR      NZ,PRMERR       ;Jump on parm error
2442 ED5BE625 00620          LD      DE,(FCBSRC+1)   ;Stuff source name
2446 ED539F25 00630          LD      (RTENAM+3),DE
244A          00640          @@FLAGS                 ;Get flag table pointer
244A 3E65     00014          LD      A,101
244C EF       00015          RST     40
              00650 ;
              00660 ;        Test NIL parameter
              00670 ;
244D 010000   00680 NPARM    LD      BC,0            ;P/u NIL parm
2450 78       00690          LD      A,B
2451 B1       00700          OR      C
2452 C22825   00710          JP      NZ,NILDCB       ;Jump if NIL entered
              00720 ;
              00730 ;        Route to device/file - check which
              00740 ;
2455 11A525   00750          LD      DE,FCBDST       ;Fetch destination spec
2458          00760          @@FSPEC
2458 3E4E     00016          LD      A,78
245A EF       00017          RST     40
245B C21D24   00770 SPCER    JP      NZ,SPCERR       ;Jump on error
245E D5       00780          PUSH    DE
245F 118425   00790          LD      DE,PRMTBL$
2462          00800          @@PARAM                 ;Need in case REWIND
2462 3E11     00018          LD      A,17
2464 EF       00019          RST     40
2465 D1       00810          POP     DE
2466 20A4     00820          JR      NZ,PRMERR       ;Exit on parm error
2468 1A       00830          LD      A,(DE)
2469 FE2A     00840          CP      '*'             ;Test device/file
246B 2040     00850          JR      NZ,INITFCB      ;Jump on file
              00860 ;
              00870 ;        Destination spec is a device
              00880 ;
246D ED5BA625 00890          LD      DE,(FCBDST+1)   ;P/u device name
2471 2AE625   00900          LD      HL,(FCBSRC+1)   ;Make sure SRC<>DST
2474 ED52     00910          SBC     HL,DE           ;  CF is reset
2476 CA1D24   00920          JP      Z,SPCERR
2479          00930          @@GTDCB                 ;Find in tables
2479 3E52     00020          LD      A,82
247B EF       00021          RST     40
247C C2E024   00940          JP      NZ,IOERR        ;Jump if not found
247F E5       00950 CKDCBS   PUSH    HL              ;Save DCB address of dest
2480 CD0D25   00960          CALL    CKSRC           ;Locate source DCB
2483 C2E024   00970          JP      NZ,IOERR
2486          00980 CKDCB1   EQU     $
2486 F3       00990          DI
2487 C1       01000          POP     BC              ;Rcvr dest route vector
2488 E5       01010          PUSH    HL              ;Save DCB+0
              01020 ;
              01030 ;        Save the old device vector while stuffing new
              01040 ;
2489 2C       01050          INC     L               ;Bump to vector
248A 7E       01060          LD      A,(HL)          ;Save what's there
248B 71       01070          LD      (HL),C          ;Stuff dest route
```

```
248C 4F        Ø1Ø8Ø          LD      C,A             ;  into DCB of source
248D 2C        Ø1Ø9Ø          INC     L               ;  while saving old
248E 7E        Ø11ØØ          LD      A,(HL)          ;  vector for storage
248F 7Ø        Ø111Ø          LD      (HL),B          ;  (could be a FCB)
2490 47        Ø112Ø          LD      B,A
               Ø113Ø ;
               Ø114Ø ;        Now set ROUTE bit and rest of DCB block
               Ø115Ø ;
2491 E1        Ø116Ø          POP     HL              ;Rcvr ptr to DCB+Ø
2492 7E        Ø117Ø          LD      A,(HL)          ;Init the ROUTE bit
2493 F5        Ø118Ø          PUSH    AF              ;Save old TYPE byte
2494 E6Ø7      Ø119Ø          AND     7               ;Strip any flag bits
2496 F61Ø      Ø12ØØ          OR      1ØH
2498 77        Ø121Ø          LD      (HL),A          ;Show source is routed
2499 7D        Ø122Ø          LD      A,L
249A C6Ø7      Ø123Ø          ADD     A,7             ;Point to name field
249C 6F        Ø124Ø          LD      L,A
249D 72        Ø125Ø          LD      (HL),D          ;And stuff in the name
249E 2D        Ø126Ø          DEC     L               ;  in case this is a
249F 73        Ø127Ø          LD      (HL),E          ;  new DCB block
24AØ F1        Ø128Ø          POP     AF              ;P/u old TYPE byte &
24A1 CB67      Ø129Ø          BIT     4,A             ;  save old data if
24A3 2ØØ6      Ø13ØØ          JR      NZ,CKDCB2       ;  not already routed
24A5 2D        Ø131Ø          DEC     L
24A6 7Ø        Ø132Ø          LD      (HL),B          ;Stuff old vector
24A7 2D        Ø133Ø          DEC     L               ;  for reclamation
24A8 71        Ø134Ø          LD      (HL),C
24A9 2D        Ø135Ø          DEC     L
24AA 77        Ø136Ø          LD      (HL),A          ;Stuff old TYPE
24AB           Ø137Ø CKDCB2   EQU     $
24AB FB        Ø138Ø          EI
24AC C9        Ø139Ø          RET                     ;Successful
               Ø14ØØ ;
               Ø141Ø ;        Destination is file - init it & posn to end
               Ø142Ø ;
24AD D5        Ø143Ø INITFCB  PUSH    DE
24AE 119C25    Ø144Ø          LD      DE,RTENAM       ;See if space already
24B1           Ø145Ø          @@GTMOD                 ;  allocated for this
24B1 3E53      ØØ22           LD      A,83
24B3 EF        ØØ23           RST     4Ø
24B4 D1        Ø146Ø          POP     DE
24B5 2ØØC      Ø147Ø          JR      NZ,NOTRES       ;  device name
               Ø148Ø ;
               Ø149Ø ;        Space in memory, re-use it
               Ø15ØØ ;
24B7 23        Ø151Ø          INC     HL              ;Get last byte used
24B8 23        Ø152Ø          INC     HL              ;  into HL
24B9 7E        Ø153Ø          LD      A,(HL)
24BA 23        Ø154Ø          INC     HL
24BB 66        Ø155Ø          LD      H,(HL)
24BC 6F        Ø156Ø          LD      L,A
24BD AF        Ø157Ø          XOR     A               ;Set a Ø to show
24BE 32F724    Ø158Ø          LD      (CKIFRES+1),A   ;  already resident
24C1 18ØE      Ø159Ø          JR      SETBUF
               Ø16ØØ ;
               Ø161Ø ;        Not yet resident, get space
               Ø162Ø ;
24C3 FDCBØ246  Ø163Ø NOTRES   BIT     Ø,(IY+'C'-'A')  ;Can we alter HIGH$?
24C7 C21924    Ø164Ø          JP      NZ,CANT         ;Can't if frozen
24CA 21ØØØØ    Ø165Ø          LD      HL,Ø            ;Get high
24CD 45        Ø166Ø          LD      B,L
```

```
24CE            01670           @@HIGH$
24CE 3E64       00024           LD      A,100
24D0 EF         00025           RST     40
24D1 229925     01680 SETBUF    LD      (RTEDVR+2),HL   ;Stuff highest used
24D4 23         01690           INC     HL              ;Reserve a page for
24D5 25         01700           DEC     H               ;  the I/O buffer
24D6 E5         01710           PUSH    HL              ;Don't lose it
24D7 0600       01720           LD      B,0             ;LRL = 0
24D9            01730           @@INIT                  ;Init the file
24D9 3E3A       00026           LD      A,58
24DB EF         00027           RST     40
24DC 200D       01740           JR      NZ,INITF1       ;What? an error?
24DE 010000     01750 RPARM     LD      BC,0            ;Ck on rewind (no peof)
24E1 04         01760           INC     B               ;Keep file at start
24E2 2807       01770           JR      Z,INITF1        ;  if REWIND specified
24E4            01780           @@PEOF                  ;  else posn file
24E4 3E41       00028           LD      A,65
24E6 EF         00029           RST     40
24E7 2802       01790           JR      Z,INITF1        ;  to the end
24E9 FE1C       01800           CP      1CH             ;At End Of File?
24EB E1         01810 INITF1    POP     HL              ;Get back buffer pointer
24EC C2E424     01820           JP      NZ,IOERR        ;Any other error, JuMp
24EF 012E00     01830           LD      BC,32+14        ;Back up another 32
24F2 AF         01840           XOR     A               ;  for the FCB storage
24F3 ED42       01850           SBC     HL,BC           ;  + 14 for linkage
24F5 E5         01860           PUSH    HL              ;Save module start
                01870 ;
                01880 ;         Bypass HIGH$ stuff if "ISRES"
                01890 ;
24F6 F6FF       01900 CKIFRES   OR      -1              ;"OR 0" if "ISRES"
24F8 2804       01910           JR      Z,ISRES1
24FA 2B         01920           DEC     HL              ;Reset HIGH$ (B=0)
24FB            01930           @@HIGH$                 ;Stuff new high$
24FB 3E64       00030           LD      A,100
24FD EF         00031           RST     40
24FE D1         01940 ISRES1    POP     DE              ;Rcvr module pointer
24FF D5         01950           PUSH    DE
2500 219725     01960           LD      HL,RTEDVR       ;Move module to memory
2503 EDB0       01970           LDIR
2505 D1         01980           POP     DE              ;Now adjust to true
2506 210E00     01990           LD      HL,14           ;  FCB loc'n
2509 19         02000           ADD     HL,DE
250A C37F24     02010           JP      CKDCBS          ;Go check dcbs
                02020 ;
                02030 ;         Scan device tables for source device
                02040 ;
250D ED5BE625   02050 CKSRC     LD      DE,(FCBSRC+1)   ;P/u source device name
2511 D5         02060           PUSH    DE              ;  & save it for later
2512            02070           @@GTDCB                 ;Find device in table
2512 3E52       00032           LD      A,82
2514 EF         00033           RST     40
2515 280A       02080           JR      Z,CKSRC1        ;Use it if found
2517 110000     02090           LD      DE,0            ;  else find a spare
251A            02100           @@GTDCB                 ;  DCB block
251A 3E52       00034           LD      A,82
251C EF         00035           RST     40
251D 3E21       02110           LD      A,33            ;Init "No device space...
251F 2005       02120           JR      NZ,CKSRC2       ;Abort if no space
2521 E5         02130 CKSRC1    PUSH    HL
2522 CD3925     02140           CALL    CLSFILS         ;Close any existing
2525 E1         02150           POP     HL              ;  file routes
```

```
2526 D1       Ø216Ø CKSRC2  POP     DE              ;Recover source name
2527 C9       Ø217Ø         RET
              Ø218Ø ;
              Ø219Ø ;       NIL entered, close up any open file
              Ø22ØØ ;
2528 CDØD25   Ø221Ø NILDCB  CALL    CKSRC           ;Check on devqice
252B 7E       Ø222Ø         LD      A,(HL)          ;Get type byte
252C F6Ø8     Ø223Ø         OR      8
252E 77       Ø224Ø         LD      (HL),A          ;Show is NIL device
252F 7D       Ø225Ø         LD      A,L             ;Pt to name field
2530 C6Ø6     Ø226Ø         ADD     A,6
2532 6F       Ø227Ø         LD      L,A
2533 F3       Ø228Ø         DI
2534 73       Ø229Ø         LD      (HL),E          ;Stuff in our name
2535 2C       Ø23ØØ         INC     L               ;  in case it's a new
2536 72       Ø231Ø         LD      (HL),D          ;  DCB block
2537 FB       Ø232Ø         EI
2538 C9       Ø233Ø         RET                     ;Successful
              Ø234Ø ;
              Ø235Ø ;       Find the last device route & close any open file
              Ø236Ø ;
2539 CB66     Ø237Ø CLSFILS BIT     4,(HL)          ;Jump if no route
253B 28Ø7     Ø238Ø         JR      Z,CLSFIL1
253D 23       Ø239Ø         INC     HL              ;Else p/u link address
253E 7E       Ø24ØØ         LD      A,(HL)          ;  and test that one
253F 23       Ø241Ø         INC     HL              ;  for a chain
2540 66       Ø242Ø         LD      H,(HL)
2541 6F       Ø243Ø         LD      L,A
2542 18F5     Ø244Ø         JR      CLSFILS
2544 CB7E     Ø245Ø CLSFIL1 BIT     7,(HL)          ;A file?
2546 C8       Ø246Ø         RET     Z               ;Ret if not
2547 11C525   Ø247Ø         LD      DE,FCBFIL       ;Pt to fcb area
254A D5       Ø248Ø         PUSH    DE
254B Ø12ØØØ   Ø249Ø         LD      BC,32
254E EDBØ     Ø25ØØ         LDIR                    ;Fill from device vector
2550 D1       Ø251Ø         POP     DE              ;Recover start
2551          Ø252Ø         @@CLOSE                 ;Close the file
2551 3E3C     ØØØ36         LD      A,6Ø
2553 EF       ØØØ37         RST     4Ø
2554 C9       Ø253Ø         RET                     ;Ret with Z, NZ status
              Ø254Ø ;
              Ø255Ø ;       Messages
              Ø256Ø ;
2555 4E       Ø257Ø CANT$   DB      'No memory space available',CR
     6F 2Ø 6D 65 6D 6F 72 79
     2Ø 73 7Ø 61 63 65 2Ø 61
     76 61 69 6C 61 62 6C 65
     ØD
256F 44       Ø258Ø SPCERR$ DB      'Device spec required',CR
     65 76 69 63 65 2Ø 73 7Ø
     65 63 2Ø 72 65 71 75 69
     72 65 64 ØD
              Ø259Ø ;
2584 8Ø       Ø26ØØ PRMTBL$ DB      8ØH,53H,'NIL',Ø
     53 4E 49 4C ØØ
258A 4E24     Ø261Ø         DW      NPARM+1
258C 56       Ø262Ø         DB      56H,'REWIND',Ø
     52 45 57 49 4E 44 ØØ
2594 DF24     Ø263Ø         DW      RPARM+1
2596 ØØ       Ø264Ø         NOP
              Ø265Ø ;
```

```
2597 18FE      02660 RTEDVR  JR      $               ;No real jump
2599 0000      02670         DW      $-$             ;Stuff of high
259B 05        02680         DB      5
259C 52        02690 RTENAM  DB      'RTExx'
     54 45 78 78
25A1 0000      02700         DW      0,0
     0000
               02710 ;
25A5 00        02720 FCBDST  DB      0
001F           02730         DS      31
25C5 00        02740 FCBFIL  DB      0
001F           02750         DS      31
25E5 00        02760 FCBSRC  DB      0
001F           02770         DS      31
               02780 ;
2400           02790         END     ROUTE
```

```
@@1             0000 @@2             0000 @@3             0000
@@4             0000 @MOD2           0000 @MOD4           FFFF
CANT            2419 CANT$           2555 CKDCB1          2486
CKDCB2          24AB CKDCBS          247F CKIFRES         24F6
CKSRC           250D CKSRC1          2521 CKSRC2          2526
CLSFIL1         2544 CLSFILS         2539 CR              000D
ERREXIT         2423 EXIT            2407 FCBDST          25A5
FCBFIL          25C5 FCBSRC          25E5 INITF1          24EB
INITFCB         24AD IOERR           240E ISRES1          24FE
NILDCB          2528 NOTRES          24C3 NPARM           244D
PAR_ERR         002C PRMERR          240C PRMTBL$         2584
ROUTE           2400 ROUTE1          242D RPARM           24DE
RTEDVR          2597 RTENAM          259C SAVESP          2426
SETBUF          24D1 SPCER           245B SPCERR          241D
SPCERR$         256F @@ABORT         83D2 @@ADTSK         8465
@@BANK          897D @@BKSP          865D @@BREAK         8993
@@CHNIO         83BD @@CKBRKC        89E1 @@CKDRV         84B9
@@CKEOF         8672 @@CKTSK         8450 @@CLOSE         8648
@@CLS           89CB @@CMNDI         83FC @@CMNDR         8411
@@CTL           8221 @@DATE          8393 @@DCSTAT        84F8
@@DEBUG         843B @@DECHEX        88FD @@DIRRD         886A
@@DIRWR         887F @@DIV16         88E8 @@DIV8          88D3
@@DODIR         84CE @@DSP           81E5 @@DSPLY         8285
@@ERROR         8426 @@EXIT          83E7 @@FEXT          87D7
@@FLAGS         8967 @@FNAME         87EC @@FSPEC         87C2
@@GATRD         8855 @@GATWR         8894 @@GET           81F9
@@GTDCB         8816 @@GTDCT         8801 @@GTMOD         882B
@@HDFMT         85A0 @@HEX16         893C @@HEX8          8927
@@HEXDEC        8912 @@HIGH$         8951 @@INIT          861E
@@KBD           825D @@KEY           81D1 @@KEYIN         8271
@@KLTSK         84A4 @@LOAD          8798 @@LOC           8687
@@LOF           869C @@LOGER         82BC @@LOGOT         82D1
@@MSG           8308 @@MUL16         88BE @@MUL8          88A9
@@OPEN          8633 @@PARAM         837E @@PAUSE         8369
@@PEOF          86B1 @@POSN          86C6 @@PRINT         831D
@@PRT           8235 @@PUT           820D @@RAMDIR        84E3
@@RDSEC         8576 @@RDSSC         8840 @@READ          86DB
@@REMOV         8609 @@RENAM         85F4 @@REW           86F0
@@RMTSK         847A @@RPTSK         848F @@RREAD         8705
@@RSLCT         8561 @@RSTOR         8522 @@RUN           87AD
@@RWRIT         871A @@SEEK          854C @@SEEKSC        872F
@@SKIP          8744 @@SLCT          850D @@STEPI         8537
@@TIME          83A8 @@VDCTL         8354 @@VER           8759
@@VRSEC         858B @@WEOF          876E @@WHERE         8249
@@WRITE         8783 @@WRSEC         85B5 @@WRSSC         85CA
@@WRTRK         85DF
2400 is the transfer address
00000 Total errors
```

NOTES:

NOTES:

Command: SET, FILTER

Library: SYS6/SYS

ISAM # : 65H, 66H

```
                00100 ;LBSET/ASM - Set and Filter commands
0000            00110         TITLE   <SET/FILTER - LS-DOS 6.2>
                00120 ;
000D            00130 CR      EQU     13
0000            00140 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00150 ;
                00160 ;       ORG this up here to allow driver programs to
                00170 ;       load at X'2400' without clobbering this program
                00180 ;
2C00            00190         ORG     2C00H
                00200 ;
                00210 ;       FILTER entry point
                00220 ;
2C00 C3B22C     00230         JP      FILTER          ;Filter entry point
                00240 ;
                00250 ;       SET entry point
                00260 ;
2C03            00270 SET     @@FLAGS                 ;Flag table pointer
2C03 3E65       00001         LD      A,101
2C05 EF         00002         RST     40
2C06 FDCB0246   00280         BIT     0,(IY+'C'-'A')  ;Can't use if memory
2C0A C2272D     00290         JP      NZ,CANT         ;  is frozen
2C0D 11CD2D     00300         LD      DE,DEVFCB       ;Get filespec
2C10            00310         @@FSPEC
2C10 3E4E       00003         LD      A,78
2C12 EF         00004         RST     40
2C13 C2232D     00320         JP      NZ,DEVREQ       ;Quit if bad name
2C16 1A         00330         LD      A,(DE)          ;Ck on devicespec
2C17 FE2A       00340         CP      '*'
2C19 C2232D     00350         JP      NZ,DEVREQ       ;Must have device
2C1C 11ED2D     00360         LD      DE,PGMFCB       ;Get driver or filter
2C1F            00370         @@FSPEC                 ;  filespec
2C1F 3E4E       00005         LD      A,78
2C21 EF         00006         RST     40
2C22 C22B2D     00380         JP      NZ,SPCREQ       ;Must be entered
2C25 1A         00390         LD      A,(DE)          ;Target cannot be device
2C26 FE2A       00400         CP      '*'             ;  since this is SET
2C28 CA2B2D     00410         JP      Z,SPCREQ
2C2B E5         00420         PUSH    HL              ;Save INBUF$ pointer
2C2C D5         00430         PUSH    DE              ;  and FCB start
2C2D 21AD2D     00440         LD      HL,SAVSPEC      ;Save the filter/driver
2C30 EB         00450         EX      DE,HL           ;  filespec to try /DVR
2C31 012000     00460         LD      BC,32           ;  if /FLT is not found
2C34 EDB0       00470         LDIR
2C36 D1         00480         POP     DE              ;Recover FCB
2C37 210B2D     00490         LD      HL,FLTEXT       ;Default extension is FLT
2C3A            00500         @@FEXT                  ;Use default EXT if none
2C3A 3E4F       00007         LD      A,79
2C3C EF         00008         RST     40
2C3D E1         00510         POP     HL              ;Recover cmdline posn
                00520 ;
                00530 ;       Make sure device is not in system
                00540 ;
2C3E E5         00550         PUSH    HL              ;Save INBUF$ pointer
2C3F ED5BCE2D   00560         LD      DE,(DEVFCB+1)   ;P/u device name
2C43            00570         @@GTDCB                 ;Find device DCB address
2C43 3E52       00009         LD      A,82
2C45 EF         00010         RST     40
```

```
2C46 2016      00580          JR      NZ,NEWDCB       ;Go if not found
2C48 CB5E      00590          BIT     3,(HL)          ;  else check if NIL
2C4A 3E27      00600          LD      A,39            ;Init "Device in use...
2C4C 2840      00610          JR      Z,ERRPOP        ;Error if not NIL
               00620 ;
               00630 ;        Inhibit SETting any system device
               00640 ;
2C4E E5        00650          PUSH    HL              ;Save DCB pointer
2C4F 21CD2D    00660          LD      HL,DEVFCB       ;Determine if system
2C52 54        00670          LD      D,H             ;  device by attempting
2C53 5D        00680          LD      E,L             ;  to rename it
2C54           00690          @@RENAM                 ;The error code will be
2C54 3E38      00011          LD      A,56
2C56 EF        00012          RST     40
2C57 E1        00700          POP     HL              ;  either 19 or 40
2C58 FE28      00710          CP      40              ;Protected system device?
2C5A 2832      00720          JR      Z,ERRPOP
2C5C 180A      00730          JR      GOTDCB          ;  else we have it
               00740 ;
               00750 ;        Device not found - Locate spare DCB
               00760 ;
2C5E 110000    00770 NEWDCB   LD      DE,0            ;Find spare device
2C61           00780          @@GTDCB                 ;  table position
2C61 3E52      00013          LD      A,82
2C63 EF        00014          RST     40
2C64 3E21      00790          LD      A,33            ;"no device space avail
2C66 2026      00800          JR      NZ,ERRPOP       ;Exit on error
               00810 ;
               00820 ;        DCB available - Load the driver/filter
               00830 ;
2C68 E5        00840 GOTDCB   PUSH    HL              ;Save table address
2C69 FDCB12D6  00850          SET     2,(IY+'S'-'A')  ;Allow use with EXEC only
2C6D 11ED2D    00860          LD      DE,PGMFCB       ;Load the target file
2C70           00870          @@LOAD                  ;Transfer address in HL
2C70 3E4C      00015          LD      A,76
2C72 EF        00016          RST     40
2C73 281D      00880          JR      Z,LOADOK        ;Go if file found
2C75 E63F      00890          AND     3FH             ;Strip flags
2C77 FE1F      00900          CP      31              ;Program not found?
2C79 2012      00910          JR      NZ,LOADERR      ;Abort on any other
               00920 ;
               00930 ;        No FILTER found - Check on DRIVER
               00940 ;
2C7B 11AD2D    00950          LD      DE,SAVSPEC      ;Original filename
2C7E 210E2D    00960          LD      HL,DVREXT       ;Try with /DVR
2C81           00970          @@FEXT
2C81 3E4F      00017          LD      A,79
2C83 EF        00018          RST     40
2C84 FDCB12D6  00980          SET     2,(IY+'S'-'A')  ;Allow use with EXEC only
2C88           00990          @@LOAD
2C88 3E4C      00019          LD      A,76
2C8A EF        00020          RST     40
2C8B 2805      01000          JR      Z,LOADOK        ;Go if file found
2C8D E1        01010 LOADERR  POP     HL              ;Clean the stack
2C8E E1        01020 ERRPOP   POP     HL
2C8F C3112D    01030          JP      IOERR           ;Abort on load error
               01040 ;
               01050 ;        Move device name into string buffer
               01060 ;
2C92 D1        01070 LOADOK   POP     DE              ;Rcvr table address
2C93 D5        01080          PUSH    DE
```

```
2C94 F3          01090          DI                          ;Don't interrupt me
2C95 3E08        01100          LD       A,8                ;Set up as NIL first
2C97 12          01110          LD       (DE),A
2C98 1C          01120 NOSET    INC      E                  ;Transfer device name
2C99 1C          01130          INC      E                  ;  entered in command
2C9A 1C          01140          INC      E                  ;  to the device table
2C9B 3E08        01150          LD       A,8                ;Show RESET as NIL
2C9D 12          01160          LD       (DE),A
2C9E 1C          01170          INC      E
2C9F 1C          01180          INC      E                  ;Point to name field
2CA0 1C          01190          INC      E
2CA1 3ACE2D      01200          LD       A,(DEVFCB+1)       ;Move name to DCB
2CA4 12          01210          LD       (DE),A
2CA5 1C          01220          INC      E
2CA6 3ACF2D      01230          LD       A,(DEVFCB+2)
2CA9 12          01240          LD       (DE),A
2CAA FB          01250          EI                          ;Interrupts back on
2CAB D1          01260 GODOIT   POP      DE                 ;Recover DCB address
2CAC E3          01270          EX       (SP),HL            ;Stack prog's TRAADR
2CAD FDCB02DE    01280          SET      3,(IY+'C'-'A')     ;Set system request
2CB1 C9          01290          RET                         ;  & go to it
                 01300 ;
                 01310 ;        FILTER *dev *dev routine
                 01320 ;
2CB2 11CD2D      01330 FILTER   LD       DE,DEVFCB          ;Get first spec
2CB5             01340          @@FSPEC
2CB5 3E4E        00021          LD       A,78
2CB7 EF          00022          RST      40
2CB8 C2232D      01350          JP       NZ,DEVREQ          ;Quit on bad name
2CBB 1A          01360          LD       A,(DE)             ;Ck on devicespec
2CBC FE2A        01370          CP       '*'
2CBE C2232D      01380          JP       NZ,DEVREQ          ;Must have device
2CC1 11ED2D      01390          LD       DE,PGMFCB          ;Get filter device spec
2CC4             01400          @@FSPEC
2CC4 3E4E        00023          LD       A,78
2CC6 EF          00024          RST      40
2CC7 C22B2D      01410          JP       NZ,SPCREQ          ;Must be entered
2CCA 1A          01420          LD       A,(DE)             ;Target must be a device
2CCB FE2A        01430          CP       '*'                ;  since this is FILTER
2CCD C2232D      01440          JP       NZ,DEVREQ
2CD0 ED5BEE2D    01450          LD       DE,(PGMFCB+1)      ;Get filter DCB address
2CD4             01460          @@GTDCB
2CD4 3E52        00025          LD       A,82
2CD6 EF          00026          RST      40
2CD7 2038        01470          JR       NZ,IOERR           ;Quit if not found
2CD9 CB76        01480          BIT      6,(HL)             ;Must be a filter
2CDB 2842        01490          JR       Z,NOTFLT           ;Quit if not
                 01500 ;
                 01510 ;        FILTER must be inactive to use it
                 01520 ;
2CDD 54          01530          LD       D,H                ;Xfer FILTER DCB pointer
2CDE 5D          01540          LD       E,L                ;  to DE & locate the
2CDF 2C          01550          INC      L                  ;  DCB pointer in the
2CE0 7E          01560          LD       A,(HL)             ;  the FILTER module
2CE1 2C          01570          INC      L
2CE2 66          01580          LD       H,(HL)
2CE3 6F          01590          LD       L,A
2CE4 010400      01600          LD       BC,4               ;HL now points to the
2CE7 09          01610          ADD      HL,BC              ;  entry point. Get its
2CE8 4E          01620          LD       C,(HL)             ;  DCB address by peeking
2CE9 0C          01630          INC      C                  ;  past the name field
```

```
2CEA 09         01640           ADD     HL,BC
2CEB 7E         01650           LD      A,(HL)          ;Get low-order
2CEC 23         01660           INC     HL
2CED 66         01670           LD      H,(HL)          ;Get hi-order
2CEE 6F         01680           LD      L,A
                01690                                   ;If DCB is NOT pointing
2CEF ED52       01700           SBC     HL,DE           ;  to itself, then it
2CF1 2028       01710           JR      NZ,ACTFLT       ;  is an active filter
                01720 ;
                01730 ;         The filter DCB pointer points to its DCB
                01740 ;
2CF3 D5         01750           PUSH    DE              ;Save filter DCB
2CF4 ED5BCE2D   01760           LD      DE,(DEVFCB+1)   ;Find the device DCB
2CF8            01770           @@GTDCB
2CF8 3E52       00027           LD      A,82
2CFA EF         00028           RST     40
2CFB D1         01780           POP     DE
2CFC 2013       01790           JR      NZ,IOERR        ;Quit if not found
                01800 ;
                01810 ;         Swap the 1st three bytes of DCB & FILT DCB
                01820 ;
2CFE 0603       01830           LD      B,3
2D00 4E         01840 SWAP      LD      C,(HL)
2D01 1A         01850           LD      A,(DE)
2D02 77         01860           LD      (HL),A
2D03 79         01870           LD      A,C
2D04 12         01880           LD      (DE),A
2D05 2C         01890           INC     L
2D06 1C         01900           INC     E
2D07 10F7       01910           DJNZ    SWAP
2D09 182A       01920           JR      EXIT
                01930 ;
2D0B 46         01940 FLTEXT    DB      'FLT'
     4C 54
2D0E 44         01950 DVREXT    DB      'DVR'
     56 52
                01960 ;
2D11 6F         01970 IOERR     LD      L,A             ;Transfer error code
2D12 2600       01980           LD      H,0             ;  to HL
2D14 F6C0       01990           OR      0C0H            ;Set abbrev & return
2D16 4F         02000           LD      C,A
2D17            02010           @@ERROR
2D17 3E1A       00029           LD      A,26
2D19 EF         00030           RST     40
2D1A C9         02020           RET
                02030 ;
2D1B 21952D     02040 ACTFLT    LD      HL,ACTFLT$
2D1E DD         02050           DB      0DDH
2D1F 217E2D     02060 NOTFLT    LD      HL,NOTFLT$
2D22 DD         02070           DB      0DDH
2D23 214F2D     02080 DEVREQ    LD      HL,DEVREQ$
2D26 DD         02090           DB      0DDH
2D27 21642D     02100 CANT      LD      HL,CANT$
2D2A DD         02110           DB      0DDH
2D2B 213C2D     02120 SPCREQ    LD      HL,SPCREQ$
2D2E            02130           @@LOGOT
                00031           IFEQ    00H,1
                00032           LD      HL,
                00033           ENDIF
2D2E 3E0C       00034           LD      A,12
2D30 EF         00035           RST     40
```

```
2D31 21FFFF   02140         LD      HL,-1
2D34 C9       02150         RET
              02160 ;
2D35 210000   02170 EXIT    LD      HL,0
2D38          02180         @@CKBRKC                ;Clear out break
2D38 3E6A     00036         LD      A,106
2D3A EF       00037         RST     40
2D3B C9       02190         RET
              02200 ;
2D3C 46       02210 SPCREQ$ DB      'File spec required',CR
     69 6C 65 20 73 70 65 63
     20 72 65 71 75 69 72 65
     64 0D
2D4F 44       02220 DEVREQ$ DB      'Device spec required',CR
     65 76 69 63 65 20 73 70
     65 63 20 72 65 71 75 69
     72 65 64 0D
2D64 4E       02230 CANT$   DB      'No memory space available',CR
     6F 20 6D 65 6D 6F 72 79
     20 73 70 61 63 65 20 61
     76 61 69 6C 61 62 6C 65
     0D
2D7E 44       02240 NOTFLT$ DB      'Device is not a filter',CR
     65 76 69 63 65 20 69 73
     20 6E 6F 74 20 61 20 66
     69 6C 74 65 72 0D
2D95 46       02250 ACTFLT$ DB      'FILTER module is in use',CR
     49 4C 54 45 52 20 6D 6F
     64 75 6C 65 20 69 73 20
     69 6E 20 75 73 65 0D
              02260 ;
0020          02270 SAVSPEC DS      32
0020          02280 DEVFCB  DS      32              ;Device file control block
0020          02290 PGMFCB  DS      32              ;Driver/filter FCB
              02300 ;
2C03          02310         END     SET
```

```
@@1             ØØØØ @@2        ØØØØ @@3        ØØØØ
@@4             ØØØØ @MOD2      ØØØØ @MOD4      FFFF
ACTFLT          2D1B ACTFLT$    2D95 CANT       2D27
CANT$           2D64 CR         ØØØD DEVFCB     2DCD
DEVREQ          2D23 DEVREQ$    2D4F DVREXT     2DØE
ERRPOP          2C8E EXIT       2D35 FILTER     2CB2
FLTEXT          2DØB GODOIT     2CAB GOTDCB     2C68
IOERR           2D11 LOADERR    2C8D LOADOK     2C92
NEWDCB          2C5E NOSET      2C98 NOTFLT     2D1F
NOTFLT$         2D7E PGMFCB     2DED SAVSPEC    2DAD
SET             2CØ3 SPCREQ     2D2B SPCREQ$    2D3C
SWAP            2DØØ @@ABORT    8ØAØ @@ADTSK    8133
@@BANK          864B @@BKSP     832B @@BREAK    8661
@@CHNIO         8Ø8B @@CKBRKC   86AF @@CKDRV    8187
@@CKEOF         834Ø @@CKTSK    811E @@CLOSE    8316
@@CLS           8699 @@CMNDI    8ØCA @@CMNDR    8ØDF
@@CTL           7EEF @@DATE     8Ø61 @@DCSTAT   81C6
@@DEBUG         81Ø9 @@DECHEX   85CB @@DIRRD    8538
@@DIRWR         854D @@DIV16    85B6 @@DIV8     85A1
@@DODIR         819C @@DSP      7EB3 @@DSPLY    7F53
@@ERROR         8ØF4 @@EXIT     8ØB5 @@FEXT     84A5
@@FLAGS         8635 @@FNAME    84BA @@FSPEC    849Ø
@@GATRD         8523 @@GATWR    8562 @@GET      7EC7
@@GTDCB         84E4 @@GTDCT    84CF @@GTMOD    84F9
@@HDFMT         826E @@HEX16    86ØA @@HEX8     85F5
@@HEXDEC        85EØ @@HIGH$    861F @@INIT     82EC
@@KBD           7F2B @@KEY      7E9F @@KEYIN    7F3F
@@KLTSK         8172 @@LOAD     8466 @@LOC      8355
@@LOF           836A @@LOGER    7F8A @@LOGOT    7F9F
@@MSG           7FD6 @@MUL16    858C @@MUL8     8577
@@OPEN          83Ø1 @@PARAM    8Ø4C @@PAUSE    8Ø37
@@PEOF          837F @@POSN     8394 @@PRINT    7FEB
@@PRT           7FØ3 @@PUT      7EDB @@RAMDIR   81B1
@@RDSEC         8244 @@RDSSC    85ØE @@READ     83A9
@@REMOV         82D7 @@RENAM    82C2 @@REW      83BE
@@RMTSK         8148 @@RPTSK    815D @@RREAD    83D3
@@RSLCT         822F @@RSTOR    81FØ @@RUN      847B
@@RWRIT         83E8 @@SEEK     821A @@SEEKSC   83FD
@@SKIP          8412 @@SLCT     81DB @@STEPI    82Ø5
@@TIME          8Ø76 @@VDCTL    8Ø22 @@VER      8427
@@VRSEC         8259 @@WEOF     843C @@WHERE    7F17
@@WRITE         8451 @@WRSEC    8283 @@WRSSC    8298
@@WRTRK         82AD
2CØ3 is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: SETCOM

Library: SYS8/SYS

ISAM # : B2H

```
                00100 ;LBSETCOM/ASM - Set RS232 Parameters
0000            00110         TITLE   <SETCOM - LS-DOS 6.2>
                00120 ;
                00130 ;       Data area offsets
                00140 ;
0004            00150 OFFSET  EQU     4
0004            00160 MSMASK  EQU     OFFSET+0
0005            00170 UCIMAGE EQU     OFFSET+1
0006            00180 BAUDRT  EQU     OFFSET+2
0007            00190 BRK     EQU     OFFSET+3
                00200 ;
                00210 ;       Default settings
                00220 ;
0000            00230 DEFMS   EQU     000H
00A5            00240 DEFUC   EQU     0A5H
0055            00250 DEFBA   EQU     055H
0003            00260 DEFBR   EQU     003H
                00270 ;
                00280 ;       ASCII/init equivalences
                00290 ;
0002            00300 _INIT   EQU     02H             ;Ctl value to init driver
0003            00310 _ETX    EQU     03H
000D            00320 _CR     EQU     0DH
000A            00330 _LF     EQU     0AH
0027            00340 _APOS   EQU     27H
002C            00350 PAR_ERR EQU     44              ;Parameter Error
                00360 ;
                00370 ;       @PARAM evaluation types
                00380 ;
0040            00390 FLAG    EQU     01000000B
0010            00400 ABB     EQU     00010000B
0080            00410 NUM     EQU     10000000B
0020            00420 STR     EQU     00100000B
                00430 ;
0000            00440 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00450 ;
2400            00460         ORG     2400H
                00470 ;
2400            00480 BEGIN   EQU     $
2400 ED730B24   00490         LD      (SAVESP+1),SP   ;Save entry stack
2404 CD3A24     00500         CALL    PGRM            ;Perform operations
                00510 ;
                00520 ;       Set up exit condition
                00530 ;
2407 210000     00540 $EXIT   LD      HL,0            ;No errors
240A 310000     00550 SAVESP  LD      SP,$-$          ;P/u original SP
240D            00560         @@CKBRKC                ;Clear <BREAK>
240D 3E6A       00001         LD      A,106
240F EF         00002         RST     40
2410 C9         00570         RET
                00580 ;
                00590 ;       Abort program
                00600 ;
2411 21FFFF     00610 $ABORT  LD      HL,-1           ;Init error return
2414 18F4       00620         JR      SAVESP          ;Exit program
                00630 ;
                00640 ;       Display char in C
                00650 ;
```

```
2416          00660 $DSP    @@DSP                   ;Display
2416 3E02     00003         LD      A,2
2418 EF       00004         RST     40
2419 C8       00670         RET     Z               ;Return if NO error
241A 1804     00680         JR      IOERR           ;  else error
              00690 ;
              00700 ;       Display text @ (HL)
              00710 ;
241C          00720 $DSPLY  @@DSPLY                 ;Display
              00005         IFEQ    00H,1
              00006         LD      HL,
              00007         ENDIF
241C 3E0A     00008         LD      A,10
241E EF       00009         RST     40
241F C8       00730         RET     Z               ;Return if NO error
              00740 ;
              00750 ;       I/O error, display what
              00760 ;
2420 6F       00770 IOERR   LD      L,A             ;Save error code
2421 F6C0     00780         OR      0C0H            ;Set return+short message
2423 4F       00790         LD      C,A             ;Pass error code
2424          00800         @@ERROR                 ;Display error
2424 3E1A     00010         LD      A,26
2426 EF       00011         RST     40
              00810 ;
2427 2600     00820 ERROR$  LD      H,0             ;Set exit code
2429 18DF     00830         JR      SAVESP          ;Terminate program
              00840 ;
242B 21E126   00850 BADMOD  LD      HL,BADDCB       ;'COM/DVR not installed'
242E          00860         @@LOGOT                 ;Log error
              00012         IFEQ    00H,1
              00013         LD      HL,
              00014         ENDIF
242E 3E0C     00015         LD      A,12
2430 EF       00016         RST     40
2431 2E08     00870         LD      L,8             ;'device not available'
2433 18F2     00880         JR      ERROR$          ;Return error in HL
              00890 ;
              00900 ;       Parameter error
              00910 ;
2435 3E2C     00920 PRMERR  LD      A,PAR_ERR       ;"Parameter Error"
2437 C32024   00930         JP      IOERR           ;Abort
              00940 ;
              00950 ;       PGRM - Set up RS-232 Parameters
              00960 ;
243A          00970 PGRM    EQU     $
              00980 ;
              00990         IF      @MOD4
243A E5       01000         PUSH    HL              ;Save command line pntr
243B 11DD26   01010         LD      DE,MDNAME       ;$CL name header
243E          01020         @@GTMOD                 ;Find module header
243E 3E53     00017         LD      A,83
2440 EF       00018         RST     40
2441 20E8     01030         JR      NZ,BADMOD       ;Exit if not found
2443 D5       01040         PUSH    DE              ;Pass pointer to IX
2444 DDE1     01050         POP     IX              ;IX=>next byte after
2446 E1       01060         POP     HL              ;Rcvr cmdline ptr
              01070         ENDIF
              01080 ;
              01090 ;       Check if any parameters entered
              01100 ;
```

```
2447 2B       Ø111Ø         DEC     HL              ;Setup for immed INC
2448 23       Ø112Ø SPLP    INC     HL              ;Bump to next char
2449 7E       Ø113Ø         LD      A,(HL)          ;Fetch input char
244A FE2Ø     Ø114Ø         CP      ' '             ;Space?
244C 28FA     Ø115Ø         JR      Z,SPLP          ;Ignore if space
244E FE28     Ø116Ø         CP      '('             ;Param marker?
2450 C2E229   Ø117Ø         JP      NZ,SHOW         ;No, show settings
              Ø118Ø ;
2453 11Ø728   Ø119Ø         LD      DE,PRMTBL$      ;Parameter table
2456          Ø12ØØ         @@PARAM                 ;Evaluate user input
2456 3E11     ØØØ19         LD      A,17
2458 EF       ØØØ2Ø         RST     4Ø
2459 C23524   Ø121Ø         JP      NZ,PRMERR       ;Go if parameter error
              Ø122Ø ;
              Ø123Ø         IF      @MOD2
              Ø124Ø         LD      A,(PORTP)       ;Get port param
              Ø125Ø         OR      A               ;Anything input?
              Ø126Ø         LD      DE,MDNAME1      ;$CL
              Ø127Ø         JR      Z,PORTF         ;Go default
              Ø128Ø         LD      BC,(PORTD)      ;Get port value
              Ø129Ø         INC     B               ;Test msb
              Ø13ØØ         DEC     B
              Ø131Ø         JR      NZ,PRMERR       ;Param error if >256
              Ø132Ø         LD      A,C             ;Get lsb
              Ø133Ø         OR      A               ;Ø?
              Ø134Ø         JR      Z,PRMERR        ;Go if yes
              Ø135Ø         DEC     A               ;1?
              Ø136Ø         JR      Z,PORTF         ;Yes, $CL
              Ø137Ø         DEC     A               ;2?
              Ø138Ø         JR      NZ,PRMERR       ;Param error if not
              Ø139Ø         LD      DE,MDNAME2      ;$CM
245C          Ø14ØØ PORTF   @@GTMOD                 ;Locate it
              ØØ21          LD      A,83
              ØØ22          RST     4Ø
              Ø141Ø         JR      NZ,BADMOD       ;Go if not found
              Ø142Ø         PUSH    DE              ;Else pass to IX
              Ø143Ø         POP     IX              ;IX => block
              Ø144Ø         ENDIF
              Ø145Ø ;
              Ø146Ø ;       Check if DEFAULT param entered
              Ø147Ø ;
245C ED5B6627 Ø148Ø         LD      DE,(DFPARM)     ;Check D parm
2460 7A       Ø149Ø         LD      A,D
2461 B3       Ø15ØØ         OR      E
2462 C4CC26   Ø151Ø         CALL    NZ,DEFALT       ;Go if Default parm used
              Ø152Ø ;
2465 ED5B4C27 Ø153Ø         LD      DE,(QPARM)      ;Query parm used?
2469 7A       Ø154Ø         LD      A,D             ;Check input
246A B3       Ø155Ø         OR      E
246B CA2Ø25   Ø156Ø         JP      Z,CKPARM        ;Check input if not query
              Ø157Ø ;
              Ø158Ø ;       Prompt user for input not entered on command line
              Ø159Ø ;
              Ø16ØØ ;       Fetch BAUD
              Ø161Ø ;
246E 213825   Ø162Ø ASKBAUD LD      HL,DMSG2        ;BAUD param display
2471 111A28   Ø163Ø         LD      DE,BTYP         ;BAUD param
2474 CD5Ø26   Ø164Ø         CALL    GETIT           ;Get user input
2477 3ØØB     Ø165Ø         JR      NC,ASKWORD      ;Continue on ENTER
2479 CDC229   Ø166Ø         CALL    CKBAUD          ;Test response for valid
247C 28Ø6     Ø167Ø         JR      Z,ASKWORD       ;Move on if valid entry
```

```
247E AF        01680 BADBAUD XOR     A               ;  else clear
247F 321F28    01690         LD      (BRESP),A       ;  for re-try
2482 18EA      01700         JR      ASKBAUD         ;Bad entry - retry
               01710 ;
               01720 ;       Fetch WORD
               01730 ;
2484 216625    01740 ASKWORD LD      HL,DMSG3        ;WORD display
2487 112228    01750         LD      DE,WTYP         ;Param type
248A CD5026    01760         CALL    GETIT           ;Get user input
248D 3011      01770         JR      NC,ASKSTOP      ;Nil input, leave it
248F 3A5027    01780         LD      A,(WORDP)       ;Fetch input
2492 FE05      01790         CP      5               ;<5?
2494 3804      01800         JR      C,BADWORD       ;Bad entry
2496 FE09      01810         CP      8+1             ;>8?
2498 3806      01820         JR      C,ASKSTOP       ;Go if OK
249A AF        01830 BADWORD XOR     A               ;Clear param
249B 322728    01840         LD      (WRESP),A       ;Load flag
249E 18E4      01850         JR      ASKWORD         ;Re-try
               01860 ;
               01870 ;       Fetch STOP
               01880 ;
24A0 217F25    01890 ASKSTOP LD      HL,DMSG4        ;STOP
24A3 112A28    01900         LD      DE,STYP         ;=>type byte
24A6 CD5026    01910         CALL    GETIT           ;Get user input
24A9 3014      01920         JR      NC,ASKPAR       ;Nil input, leave it
24AB 3A5227    01930         LD      A,(STOPP)       ;Chk range
24AE B7        01940         OR      A               ;0?
24AF 2804      01950         JR      Z,BADSTOP       ;Invalid if yes
24B1 FE03      01960         CP      2+1             ;>2?
24B3 380A      01970         JR      C,ASKPAR        ;Go if OK
24B5 AF        01980 BADSTOP XOR     A               ;Load zero
24B6 322F28    01990         LD      (SRESP),A       ;Clear input param
24B9 18E5      02000         JR      ASKSTOP         ;Ask again
               02010 ;
               02020 ;       Fetch PARITY
               02030 ;
24BB AF        02040 BADPAR  XOR     A               ;Load zero
24BC 323928    02050         LD      (PRESP),A       ;Clear response byte
24BF 219125    02060 ASKPAR  LD      HL,DMSG5        ;PARITY display driver
24C2 113228    02070         LD      DE,PTYP         ;Param type
24C5 CD5026    02080         CALL    GETIT           ;Get user input
24C8 3017      02090         JR      NC,ASKBRK       ;Nil input, leave
               02100 ;
               02110 ;       Check if user entered 'O' or 'E' or ON/OFF
               02120 ;
24CA 3A3928    02130         LD      A,(PRESP)       ;Get response byte
24CD ED5B5427  02140         LD      DE,(PARITYP)    ;Get param pointer
24D1 CB77      02150         BIT     6,A             ;Switch?
24D3 200C      02160         JR      NZ,ASKBRK       ;Yes, continue
               02170 ;
               02180 ;       Check for ODD/EVEN input
               02190 ;
24D5 1A        02200         LD      A,(DE)          ;Get input
24D6 CDB929    02210         CALL    UCASE           ;Make upper case
24D9 FE45      02220         CP      'E'             ;Even?
24DB 2804      02230         JR      Z,ASKBRK        ;Continue if yes
24DD FE4F      02240         CP      'O'             ;Odd?
24DF 20DA      02250         JR      NZ,BADPAR       ;Invalid, re-prompt
               02260 ;
               02270 ;       Fetch BREAK
               02280 ;
```

```
24E1 21B025   02290 ASKBRK  LD      HL,DMSG6        ;Display driver
24E4 11A828   02300         LD      DE,BRTYP        ;Param type
24E7 CD5A26   02310         CALL    GETIT           ;Get user input
              02320 ;
              02330 ;       Fetch DTR
              02340 ;
24EA 21CB25   02350 ASKDTR  LD      HL,DMSG8        ;Display driver
24ED 114B28   02360         LD      DE,DTYP         ;Param type
24F0 CD5A26   02370         CALL    GETIT           ;Get user input
              02380 ;
              02390 ;       Fetch RTS
              02400 ;
24F3 21D725   02410 ASKRTS  LD      HL,DMSG9        ;Display driver
24F6 115228   02420         LD      DE,RTYP         ;Param type
24F9 CD5A26   02430         CALL    GETIT           ;Get user input
              02440 ;
              02450 ;       Fetch RI
              02460 ;
24FC 21F225   02470 ASKRI   LD      HL,DMSG11       ;Display driver
24FF 115928   02480         LD      DE,RITYP        ;Param type
2502 CD5A26   02490         CALL    GETIT           ;Get user input
              02500 ;
              02510 ;       Fetch DSR
              02520 ;
2505 21F925   02530 ASKDSR  LD      HL,DMSG12       ;Display driver
2508 116528   02540         LD      DE,DSTYP        ;Param type
250B CD5A26   02550         CALL    GETIT           ;Get user input
              02560 ;
              02570 ;       Fetch CD
              02580 ;
250E 210026   02590 ASKCD   LD      HL,DMSG13       ;Display driver
2511 115F28   02600         LD      DE,CDTYP        ;Param type
2514 CD5A26   02610         CALL    GETIT           ;Get user input
              02620 ;
              02630 ;       Fetch CTS
              02640 ;
2517 210726   02650 ASKCTS  LD      HL,DMSG14       ;Display driver
251A 116C28   02660         LD      DE,CTTYP        ;Param type
251D CD5A26   02670         CALL    GETIT           ;Get user input
              02680 ;
              02690 ;       Check params and issue INIT to device
              02700 ;
2520 CD7F28   02710 CKPARM  CALL    SETPARAM        ;Check entered params
2523 DD5E00   02720         LD      E,(IX+0)        ;Pickup DCB address lsb
2526 DD5601   02730         LD      D,(IX+1)        ;Pickup DCB address msb
2529 0E02     02740         LD      C,_INIT         ;Init code
252B          02750         @@CTL                   ;Setup channel for values
252B 3E05     00023         LD      A,5
252D EF       00024         RST     40
252E C9       02760         RET                     ;Completed
              02770 ;
              02780 ;       Display drivers for text
              02790 ;
252F 216C27   02800 DMSG1   LD      HL,MSG1         ;RS232 params:
2532 CD1C24   02810         CALL    $DSPLY          ;Display and return
2535 360A     02820         LD      (HL),_LF        ;Set for next time
2537 C9       02830         RET
              02840 ;
2538 218027   02850 DMSG2   LD      HL,MSG2         ;BAUD=
253B CD1C24   02860         CALL    $DSPLY          ;Display
253E DD7E06   02870         LD      A,(IX+BAUDRT)   ;Get baud rate
```

```
2541 E6ØF      Ø288Ø          AND     ØFH             ;Low 4 bits only
2543 87        Ø289Ø          ADD     A,A             ;*2
2544 21FA26    Ø29ØØ          LD      HL,BAUDTBL      ;Baud lookup table
2547 85        Ø291Ø          ADD     A,L             ;Add to lsb table
2548 6F        Ø292Ø          LD      L,A             ;Update lsb table
2549 3ØØ1      Ø293Ø          JR      NC,$+3          ;Go if no page cross
254B 24        Ø294Ø          INC     H               ;  else bump page
254C 7E        Ø295Ø          LD      A,(HL)          ;Get LSB baud
254D 23        Ø296Ø          INC     HL              ;Bump table pointer
254E 66        Ø297Ø          LD      H,(HL)          ;Get MSB baud
254F 6F        Ø298Ø          LD      L,A             ;HL = baud rate
2550 11F727    Ø299Ø          LD      DE,MSG2Ø        ;Text to load
2553           Ø3ØØØ          @@HEXDEC                ;Convert to decimal
2553 3E61      ØØØ25          LD      A,97
2555 EF        ØØØ26          RST     4Ø
2556 3EØ3      Ø3Ø1Ø          LD      A,_ETX          ;End text char
2558 12        Ø3Ø2Ø          LD      (DE),A          ;Terminate text
               Ø3Ø3Ø ;
               Ø3Ø4Ø ;        Strip leading zeroes from display
               Ø3Ø5Ø ;
2559 21F727    Ø3Ø6Ø          LD      HL,MSG2Ø        ;Text loaded
255C 3E2Ø      Ø3Ø7Ø          LD      A,' '           ;Test char
255E BE        Ø3Ø8Ø STRIP1   CP      (HL)            ;Leading zero?
255F 23        Ø3Ø9Ø          INC     HL              ;Bump pointer
2560 28FC      Ø31ØØ          JR      Z,STRIP1        ;Go till non-zero found
2562 2B        Ø311Ø          DEC     HL              ;Adjust pointer
2563 C31C24    Ø312Ø          JP      $DSPLY          ;Display remainder
               Ø313Ø ;
2566 218627    Ø314Ø DMSG3    LD      HL,MSG3         ;WORD=
2569 CD1C24    Ø315Ø          CALL    $DSPLY          ;Display header
256C DD7EØ5    Ø316Ø          LD      A,(IX+UCIMAGE)  ;Get data
256F Ø7        Ø317Ø          RLCA                    ;Align to low bits
2570 Ø7        Ø318Ø          RLCA
2571 Ø7        Ø319Ø          RLCA
2572 E6Ø3      Ø32ØØ          AND     3               ;2 bits only
2574 EA7925    Ø321Ø          JP      PE,$+5          ;Go if set
2577 EEØ3      Ø322Ø          XOR     3               ;Else reverse 6/7
2579 C635      Ø323Ø          ADD     A,5+'Ø'         ;Correct + ASCII
257B 4F        Ø324Ø          LD      C,A             ;Pass character
257C C31624    Ø325Ø          JP      $DSP            ;Display and return
               Ø326Ø ;
257F 218C27    Ø327Ø DMSG4    LD      HL,MSG4         ;STOP=
2582 CD1C24    Ø328Ø          CALL    $DSPLY          ;Display header
2585 ØE31      Ø329Ø          LD      C,'1'           ;Set one bit
2587 DDCBØ566  Ø33ØØ          BIT     4,(IX+UCIMAGE)  ;Get data
258B 28Ø1      Ø331Ø          JR      Z,$+3           ;Go if one stop
258D ØC        Ø332Ø          INC     C               ;Else bump to '2'
258E C31624    Ø333Ø          JP      $DSP            ;Display and return
               Ø334Ø ;
2591 219227    Ø335Ø DMSG5    LD      HL,MSG5         ;PARITY=
2594 CD1C24    Ø336Ø          CALL    $DSPLY          ;Display
2597 21E327    Ø337Ø          LD      HL,MSG16        ;OFF
259A DDCBØ55E  Ø338Ø          BIT     3,(IX+UCIMAGE)  ;Is off?
259E C21C24    Ø339Ø          JP      NZ,$DSPLY       ;Yes, go
25A1 21E727    Ø34ØØ          LD      HL,MSG17        ;ODD
25A4 DDCBØ57E  Ø341Ø          BIT     7,(IX+UCIMAGE)  ;Is odd?
25A8 28Ø3      Ø342Ø          JR      Z,$+5           ;Go if yes
25AA 21EB27    Ø343Ø          LD      HL,MSG18        ;EVEN
25AD C31C24    Ø344Ø          JP      $DSPLY          ;Display and return
               Ø345Ø ;
25BØ 219A27    Ø346Ø DMSG6    LD      HL,MSG6         ;BREAK=
```

```
25B3 CD1C24   03470          CALL    $DSPLY          ;Display header
25B6 DD4E07   03480          LD      C,(IX+BRK)      ;Get break char
25B9 21FF27   03490          LD      HL,MSG21A       ;Text to load
25BC          03500          @@HEX8                  ;Convert to ASCII
25BC 3E62     00027          LD      A,98
25BE EF       00028          RST     40
25BF 21FD27   03510          LD      HL,MSG21        ;Text start
25C2 C31C24   03520          JP      $DSPLY          ;Display and return
              03530 ;
25C5 21A127   03540 DMSG7    LD      HL,MSG7         ;Input:
25C8 C31C24   03550          JP      $DSPLY          ;Display and return
              03560 ;
25CB 21B327   03570 DMSG8    LD      HL,MSG8         ;DTR=
25CE CD1C24   03580          CALL    $DSPLY          ;Display
25D1 DDCB054E 03590          BIT     1,(IX+UCIMAGE)  ;Is on?
25D5 180A     03600          JR      DMSG89          ;Display
              03610 ;
25D7 21B827   03620 DMSG9    LD      HL,MSG9         ;RTS=
25DA CD1C24   03630          CALL    $DSPLY          ;Display header
25DD DDCB0546 03640          BIT     0,(IX+UCIMAGE)  ;Is on?
              03650 ;
25E1 21E027   03660 DMSG89   LD      HL,MSG15        ;ON?
25E4 2803     03670          JR      Z,$+5           ;Go if yes
25E6 21E327   03680          LD      HL,MSG16        ;OFF
25E9 C31C24   03690          JP      $DSPLY          ;Display and return
              03700 ;
25EC 21BD27   03710 DMSG10   LD      HL,MSG10        ;Output:
25EF C31C24   03720          JP      $DSPLY          ;Display and return
              03730 ;
25F2 21CE27   03740 DMSG11   LD      HL,MSG11        ;RI=
25F5 3E11     03750          LD      A,00010001B     ;Enable bits
25F7 1813     03760          JR      DMSG0           ;Go common
              03770 ;
25F9 21D227   03780 DMSG12   LD      HL,MSG12        ;DSR=
25FC 3E44     03790          LD      A,01000100B     ;Enable bits
25FE 180C     03800          JR      DMSG0           ;Go common
              03810 ;
2600 21D727   03820 DMSG13   LD      HL,MSG13        ;CD=
2603 3E22     03830          LD      A,00100010B     ;Enable bits
2605 1805     03840          JR      DMSG0           ;Go common
              03850 ;
2607 21DB27   03860 DMSG14   LD      HL,MSG14        ;CTS=
260A 3E88     03870          LD      A,10001000B     ;Enable bits
              03880 ;
260C F5       03890 DMSG0    PUSH    AF              ;Save bits
260D CD1C24   03900          CALL    $DSPLY          ;Display prefix
2610 F1       03910          POP     AF              ;Restore
2611 DDA604   03920          AND     (IX+MSMASK)     ;And with mask
2614 21F027   03930          LD      HL,MSG19        ;IGNORE
2617 2809     03940          JR      Z,DMSG0X        ;Go if yes
2619 21E027   03950          LD      HL,MSG15        ;ON
261C E22226   03960          JP      PO,DMSG0X       ;Go if yes
261F 21E327   03970          LD      HL,MSG16        ;OFF
2622 C31C24   03980 DMSG0X   JP      $DSPLY          ;Display and return
              03990 ;
2625 210328   04000 DMSG22   LD      HL,MSG22        ;CR
2628 C31C24   04010          JP      $DSPLY          ;Display and return
              04020 ;
262B 210428   04030 DMSG23   LD      HL,MSG23        ;', '
262E C31C24   04040          JP      $DSPLY          ;Display and return
              04050 ;
```

```
                    04060 ;       Display entire parameter set
                    04070 ;
2631 FDE5           04080 DSPALL  PUSH    IY              ;Save
2633 FD211A27       04090         LD      IY,DSPTBL       ;Display table
2637 FD6E00         04100 DSPSET  LD      L,(IY+0)        ;Get lsb vector
263A FD6601         04110         LD      H,(IY+1)        ;Get msb vector
263D 7C             04120         LD      A,H             ;Check for term
263E B5             04130         OR      L               ;HL = 0000?
263F 280C           04140         JR      Z,DSPDONE       ;Yes, go
2641 E5             04150         PUSH    HL              ;Save address
2642 214726         04160         LD      HL,DSPRET       ;Return vector
2645 E3             04170         EX      (SP),HL         ;Leave return, get vector
2646 E9             04180         JP      (HL)            ;Display
2647 FD23           04190 DSPRET  INC     IY              ;Bump table
2649 FD23           04200         INC     IY              ;2 byte entries
264B 18EA           04210         JR      DSPSET          ;Continue
264D FDE1           04220 DSPDONE POP     IY              ;Restore
264F C9             04230         RET                     ;Display complete
                    04240 ;
                    04250 ;       QUERYing for parameter
                    04260 ;
2650 E5             04270 GETIT   PUSH    HL              ;Save display driver
2651 D5             04280         PUSH    DE              ;Save type byte
2652 EB             04290         EX      DE,HL           ;Type byte to HL
2653 CD6326         04300         CALL    CKRSP           ;Check if response enterd
2656 D1             04310         POP     DE              ;Restore type byte
2657 E1             04320         POP     HL              ;Restore prompt
2658 C0             04330         RET     NZ              ;Already have this one
                    04340 ;
                    04350 ;       Setup for prompt display
                    04360 ;
2659 E5             04370         PUSH    HL              ;Save for repeat prompt
265A D5             04380         PUSH    DE
265B CD7426         04390         CALL    GETRSP          ;Get user response
265E D1             04400         POP     DE              ;Restore data
265F E1             04410         POP     HL
2660 20EE           04420         JR      NZ,GETIT        ;Invalid, ask again
2662 C9             04430         RET                     ;  else param loaded
                    04440 ;
                    04450 ;       Check if correct response entered
                    04460 ;
2663 22A026         04470 CKRSP   LD      (TTYP),HL       ;Save type position
                    04480 ;
2666 3E0F           04490 MVUP    LD      A,0FH           ;Low 4 bits for length
2668 4E             04500         LD      C,(HL)          ;Get type byte
2669 A1             04510         AND     C               ;Fetch length
266A 23             04520 UP      INC     HL              ;Bump pointer
266B 3D             04530         DEC     A               ;Less length
266C 20FC           04540         JR      NZ,UP           ;Go for length
266E 23             04550         INC     HL              ;Bump to next
266F 7E             04560         LD      A,(HL)          ;Get response byte
2670 E6E0           04570         AND     NUM!FLAG!STR    ;Response type bits
2672 A1             04580         AND     C               ;Compare to entered
2673 C9             04590         RET                     ;Return with status
                    04600 ;
                    04610 ;       Query on, param not entered, prompt for it
                    04620 ;
2674 E5             04630 GETRSP  PUSH    HL              ;Save driver address
2675 217A26         04640         LD      HL,GETRSPR      ;Return vector
2678 E3             04650         EX      (SP),HL         ;Leave get display driver
2679 E9             04660         JP      (HL)            ;Display prompt
```

```
                04670 ;
267A 0604       04680 GETRSPR LD      B,4             ;Command get cursor
267C            04690         @@VDCTL                 ;Fetch cursor
267C 3E0F       00029         LD      A,15
267E EF         00030         RST     40
267F 2E14       04700         LD      L,20            ;Column to position to
2681 0603       04710         LD      B,3             ;Command put cursor
2683            04720         @@VDCTL                 ;Setup new cursor
2683 3E0F       00031         LD      A,15
2685 EF         00032         RST     40
2686 21F726     04730         LD      HL,PROMPT       ;Prompt text
2689 CD1C24     04740         CALL    $DSPLY          ;Display prompt
268C AF         04750         XOR     A               ;Load zero
268D 32E829     04760         LD      (FRESP),A       ;Clear response byte
                04770 ;
2690 21EF29     04780         LD      HL,INBUF        ;Key input buffer
2693 01000A     04790         LD      BC,10<8+0       ;Max input length
2696            04800         @@KEYIN                 ;Get user input
2696 3E09       00033         LD      A,9
2698 EF         00034         RST     40
2699 DA1124     04810         JP      C,$ABORT        ;Terminate on BREAK
                04820 ;
                04830 ;       Evaluate user input
                04840 ;
269C 04         04850         INC     B               ;Check if nil input
269D 05         04860         DEC     B               ;B=0?
269E C8         04870         RET     Z               ;No input, go
                04880 ;
269F 210000     04890         LD      HL,0            ;Get param pointer
26A0            04900 TTYP    EQU     $-2
26A2 7E         04910         LD      A,(HL)          ;Get response type
26A3 E6F0       04920         AND     0F0H            ;  attribute only
26A5 F601       04930         OR      1               ;Single byte char
26A7 32E629     04940         LD      (FTYP),A        ;Pass to new block
                04950 ;
26AA 21EC29     04960         LD      HL,FSTR         ;Set command pointer
26AD 11E529     04970         LD      DE,PTBL2        ;Mini-param table
26B0            04980         @@PARAM                 ;Evaluate input
26B0 3E11       00035         LD      A,17
26B2 EF         00036         RST     40
26B3 C0         04990         RET     NZ              ;Invalid, go
                05000 ;
26B4 2AA026     05010         LD      HL,(TTYP)       ;Get response pointer
26B7 CD6626     05020         CALL    MVUP            ;Move to response byte
26BA 3AE829     05030         LD      A,(FRESP)       ;Get input response
26BD 77         05040         LD      (HL),A          ;  to param block
26BE 23         05050         INC     HL              ;Bump to word pointer
26BF CDB429     05060         CALL    GETHL           ;Get pointer
26C2 ED4B6A27   05070         LD      BC,(FPARM)      ;Get temp param
26C6 71         05080         LD      (HL),C          ;Load into param block
26C7 23         05090         INC     HL              ;Bump pointer
26C8 70         05100         LD      (HL),B          ;New param loaded
26C9 AF         05110         XOR     A               ;Set NO error
26CA 37         05120         SCF                     ;Carry = input
26CB C9         05130         RET                     ;Done
                05140 ;
                05150 ;       Init default settings
                05160 ;
26CC DD360400   05170 DEFALT  LD      (IX+MSMASK),DEFMS
26D0 DD3605A5   05180         LD      (IX+UCIMAGE),DEFUC
26D4 DD360655   05190         LD      (IX+BAUDRT),DEFBA
```

```
26D8 DD36Ø7Ø3 Ø52ØØ          LD      (IX+BRK),DEFBR
26DC C9       Ø521Ø          RET
              Ø522Ø ;
              Ø523Ø ;        Text area
              Ø524Ø ;
              Ø525Ø          IF      @MOD4
26DD 24       Ø526Ø MDNAME   DB      '$CL'
     43 4C
26EØ Ø3       Ø527Ø          DB      _ETX
              Ø528Ø          ENDIF
              Ø529Ø ;
              Ø53ØØ          IF      @MOD2
              Ø531Ø MDNAME1  DB      '$CL'
              Ø532Ø          DB      _ETX
              Ø533Ø MDNAME2  DB      '$CM'
              Ø534Ø          DB      _ETX
              Ø535Ø          ENDIF
              Ø536Ø ;
26E1 43       Ø537Ø BADDCB   DB      'COM/DVR not installed'
     4F 4D 2F 44 56 52 2Ø 6E
     6F 74 2Ø 69 6E 73 74 61
     6C 6C 65 64
26F6 ØD       Ø538Ø          DB      _CR
              Ø539Ø ;
26F7 3F       Ø54ØØ PROMPT   DB      '? '
     2Ø
26F9 Ø3       Ø541Ø          DB      _ETX
              Ø542Ø ;
              Ø543Ø ;        Valid baud rate lookup table
              Ø544Ø ;
26FA 32ØØ     Ø545Ø BAUDTBL  DW      5Ø,75,11Ø,135,15Ø,3ØØ,6ØØ,12ØØ,18ØØ
     4BØØ 6EØØ 87ØØ 96ØØ 2C Ø1 58Ø2 BØØ4 Ø8Ø7
270C DØØ7     Ø546Ø          DW      2ØØØ,24ØØ,36ØØ,48ØØ,72ØØ,96ØØ,192ØØ
     6ØØ9 1ØØE CØ12 2Ø1C 8Ø25 ØØ4B
              Ø547Ø ;
              Ø548Ø ;        Display driver lookup table
              Ø549Ø ;
271A 2F25     Ø55ØØ DSPTBL   DW      DMSG1           ;'RS232 params: '
271C 3825     Ø551Ø          DW      DMSG2           ;'BAUD='
271E 2B26     Ø552Ø          DW      DMSG23          ;', '
2720 6625     Ø553Ø          DW      DMSG3           ;'WORD='
2722 2B26     Ø554Ø          DW      DMSG23          ;', '
2724 7F25     Ø555Ø          DW      DMSG4           ;'STOP='
2726 2B26     Ø556Ø          DW      DMSG23          ;', '
2728 9125     Ø557Ø          DW      DMSG5           ;'PARITY='
272A 2B26     Ø558Ø          DW      DMSG23          ;', '
272C BØ25     Ø559Ø          DW      DMSG6           ;'BREAK='
              Ø56ØØ ;
272E C525     Ø561Ø          DW      DMSG7           ;'output: '
2730 CB25     Ø562Ø          DW      DMSG8           ;'DTR='
2732 2B26     Ø563Ø          DW      DMSG23          ;', '
2734 D725     Ø564Ø          DW      DMSG9           ;'RTS='
              Ø565Ø ;
2736 EC25     Ø566Ø          DW      DMSG1Ø          ;'input: '
2738 F225     Ø567Ø          DW      DMSG11          ;'RI='
273A 2B26     Ø568Ø          DW      DMSG23          ;', '
273C F925     Ø569Ø          DW      DMSG12          ;'DSR='
273E 2B26     Ø57ØØ          DW      DMSG23          ;', '
2740 ØØ26     Ø571Ø          DW      DMSG13          ;'CD='
2742 2B26     Ø572Ø          DW      DMSG23          ;', '
2744 Ø726     Ø573Ø          DW      DMSG14          ;'CTS='
```

```
2746 2526      05740          DW      DMSG22          ;_cr
2748 0000      05750          DW      0               ;Terminator
               05760 ;
               05770 ;        Parameter pointer table
               05780 ;
274A 0000      05790 BREAKP   DW      0
274C 0000      05800 QPARM    DW      0
274E 0000      05810 BAUDP    DW      0
2750 0000      05820 WORDP    DW      0
2752 0000      05830 STOPP    DW      0
2754 0000      05840 PARITYP  DW      0
2756 0000      05850 EVENP    DW      0
2758 0000      05860 ODDP     DW      0
275A 0000      05870 DTRP     DW      0
275C 0000      05880 RTSP     DW      0
275E 0000      05890 RIP      DW      0
2760 0000      05900 CDP      DW      0
2762 0000      05910 DSRP     DW      0
2764 0000      05920 CTSP     DW      0
2766 0000      05930 DFPARM   DW      0
2768 0000      05940 PORTD    DW      0
276A 0000      05950 FPARM    DW      0
               05960 ;
276C 00        05970 MSG1     DB      0               ;Will be LF after 1st DSP
276D 52        05980          DB      'RS232'
     53 32 33 32
               05990          IF      @MOD2
               06000 MSG1A    DB      'A'
               06010          ENDIF
2772 20        06020          DB      ' parameters: '
     70 61 72 61 6D 65 74 65
     72 73 3A 20
277F 03        06030          DB      _ETX
               06040 ;
2780 42        06050 MSG2     DB      'Baud='
     61 75 64 3D
2785 03        06060          DB      _ETX
               06070 ;
2786 57        06080 MSG3     DB      'Word='
     6F 72 64 3D
278B 03        06090          DB      _ETX
               06100 ;
278C 53        06110 MSG4     DB      'Stop='
     74 6F 70 3D
2791 03        06120          DB      _ETX
               06130 ;
2792 50        06140 MSG5     DB      'Parity='
     61 72 69 74 79 3D
2799 03        06150          DB      _ETX
               06160 ;
279A 42        06170 MSG6     DB      'Break='
     72 65 61 6B 3D
27A0 03        06180          DB      _ETX
               06190 ;
27A1 0A        06200 MSG7     DB      _LF
27A2 4F        06210          DB      'Output control: '
     75 74 70 75 74 20 63 6F
     6E 74 72 6F 6C 3A 20
27B2 03        06220          DB      _ETX
               06230 ;
27B3 44        06240 MSG8     DB      'DTR='
```

```
     54 52 3D
27B7 Ø3          Ø625Ø           DB      _ETX
                 Ø626Ø ;
27B8 52          Ø627Ø MSG9      DB      'RTS='
     54 53 3D
27BC Ø3          Ø628Ø           DB      _ETX
                 Ø629Ø ;
27BD ØA          Ø63ØØ MSG1Ø     DB      _LF
27BE 49          Ø631Ø           DB      'Input control: '
     6E 7Ø 75 74 2Ø 63 6F 6E
     74 72 6F 6C 3A 2Ø
27CD Ø3          Ø632Ø           DB      _ETX
                 Ø633Ø ;
27CE 52          Ø634Ø MSG11     DB      'RI='
     49 3D
27D1 Ø3          Ø635Ø           DB      _ETX
                 Ø636Ø ;
27D2 44          Ø637Ø MSG12     DB      'DSR='
     53 52 3D
27D6 Ø3          Ø638Ø           DB      _ETX
                 Ø639Ø ;
27D7 43          Ø64ØØ MSG13     DB      'CD='
     44 3D
27DA Ø3          Ø641Ø           DB      _ETX
                 Ø642Ø ;
27DB 43          Ø643Ø MSG14     DB      'CTS='
     54 53 3D
27DF Ø3          Ø644Ø           DB      _ETX
                 Ø645Ø ;
27EØ 4F          Ø646Ø MSG15     DB      'ON'
     4E
27E2 Ø3          Ø647Ø           DB      _ETX
                 Ø648Ø ;
27E3 4F          Ø649Ø MSG16     DB      'OFF'
     46 46
27E6 Ø3          Ø65ØØ           DB      _ETX
                 Ø651Ø ;
27E7 4F          Ø652Ø MSG17     DB      'ODD'
     44 44
27EA Ø3          Ø653Ø           DB      _ETX
                 Ø654Ø ;
27EB 45          Ø655Ø MSG18     DB      'EVEN'
     56 45 4E
27EF Ø3          Ø656Ø           DB      _ETX
                 Ø657Ø ;
27FØ 49          Ø658Ø MSG19     DB      'IGNORE'
     47 4E 4F 52 45
27F6 Ø3          Ø659Ø           DB      _ETX
                 Ø66ØØ ;
27F7 3Ø          Ø661Ø MSG2Ø     DB      'ØØØØØ'
     3Ø 3Ø 3Ø 3Ø
27FC Ø3          Ø662Ø           DB      _ETX
                 Ø663Ø ;
27FD 58          Ø664Ø MSG21     DB      'X',_APOS
     27
27FF 3Ø          Ø665Ø MSG21A    DB      'ØØ',_APOS
     3Ø 27
28Ø2 Ø3          Ø666Ø           DB      _ETX
                 Ø667Ø ;
28Ø3 ØD          Ø668Ø MSG22     DB      _CR
                 Ø669Ø ;
```

```
2804 2C          06700 MSG23   DB      ', '
     20
2806 03          06710         DB      _ETX
                 06720 ;
                 06730 ;       Parameter evaluation table
                 06740 ;
2807 80          06750 PRMTBL$ DB      80H             ;Extended param
                 06760 ;
2808 C5          06770 BRTYP   DB      5!FLAG!NUM
2809 42          06780         DB      'BREAK'
     52 45 41 4B
280E 00          06790 BRRESP  DB      0
280F 4A27        06800         DW      BREAKP
                 06810 ;
2811 55          06820         DB      5!ABB!FLAG
2812 51          06830         DB      'QUERY'
     55 45 52 59
2817 00          06840 QRESP   DB      0
2818 4C27        06850         DW      QPARM
                 06860 ;
281A 94          06870 BTYP    DB      4!ABB!NUM
281B 42          06880         DB      'BAUD'
     41 55 44
281F 00          06890 BRESP   DB      0
2820 4E27        06900         DW      BAUDP
                 06910 ;
2822 94          06920 WTYP    DB      4!ABB!NUM
2823 57          06930         DB      'WORD'
     4F 52 44
2827 00          06940 WRESP   DB      0
2828 5027        06950         DW      WORDP
                 06960 ;
282A 94          06970 STYP    DB      4!ABB!NUM
282B 53          06980         DB      'STOP'
     54 4F 50
282F 00          06990 SRESP   DB      0
2830 5227        07000         DW      STOPP
                 07010 ;
2832 76          07020 PTYP    DB      6!ABB!FLAG!STR
2833 50          07030         DB      'PARITY'
     41 52 49 54 59
2839 00          07040 PRESP   DB      0
283A 5427        07050         DW      PARITYP
                 07060 ;
283C 54          07070 ETYP    DB      4!ABB!FLAG
283D 45          07080         DB      'EVEN'
     56 45 4E
2841 00          07090 ERESP   DB      0
2842 5627        07100         DW      EVENP
                 07110 ;
2844 53          07120 OTYP    DB      3!ABB!FLAG
2845 4F          07130         DB      'ODD'
     44 44
2848 00          07140 ORESP   DB      0
2849 5827        07150         DW      ODDP
                 07160 ;
284B 43          07170 DTYP    DB      3!FLAG
284C 44          07180         DB      'DTR'
     54 52
284F 00          07190 DTRESP  DB      0
2850 5A27        07200         DW      DTRP
```

```
                    Ø7210 ;
2852 43             Ø7220 RTYP    DB      3!FLAG
2853 52             Ø7230         DB      'RTS'
     54 53
2856 ØØ             Ø7240 RTRESP  DB      Ø
2857 5C27           Ø7250         DW      RTSP
                    Ø7260 ;
2859 42             Ø7270 RITYP   DB      2!FLAG
285A 52             Ø7280         DB      'RI'
     49
285C ØØ             Ø7290 RIRESP  DB      Ø
285D 5E27           Ø7300         DW      RIP
                    Ø7310 ;
285F 42             Ø7320 CDTYP   DB      2!FLAG
286Ø 43             Ø7330         DB      'CD'
     44
2862 ØØ             Ø7340 CDRESP  DB      Ø
2863 6Ø27           Ø7350         DW      CDP
                    Ø7360 ;
2865 43             Ø7370 DSTYP   DB      3!FLAG
2866 44             Ø7380         DB      'DSR'
     53 52
2869 ØØ             Ø7390 DSRESP  DB      Ø
286A 6227           Ø7400         DW      DSRP
                    Ø7410 ;
286C 43             Ø7420 CTTYP   DB      3!FLAG
286D 43             Ø7430         DB      'CTS'
     54 53
287Ø ØØ             Ø7440 CTRESP  DB      Ø
2871 6427           Ø7450         DW      CTSP
                    Ø7460 ;
2873 57             Ø7470         DB      7!ABB!FLAG
2874 44             Ø7480         DB      'DEFAULT'
     45 46 41 55 4C 54
287B ØØ             Ø7490         DB      Ø
287C 6627           Ø7500         DW      DFPARM
                    Ø7510 ;
                    Ø7520         IF      @MOD2
                    Ø7530         DB      4!NUM
                    Ø7540         DB      'PORT'
                    Ø7550 PORTP   DB      Ø
                    Ø7560         DW      PORTD
                    Ø7570         ENDIF
                    Ø7580 ;
287E ØØ             Ø7590         DB      Ø               ;List terminator
                    Ø7600 ;
                    Ø7610 ;       Evaluate and setup user parameters
                    Ø7620 ;
                    Ø7630 SETPARAM
287F 213524         Ø7640         LD      HL,PRMERR       ;Param error exit vector
2882 E5             Ø7650         PUSH    HL              ;Leave for quick exits
                    Ø7660 ;
                    Ø7670 ;       Evaluate BAUD
                    Ø7680 ;
2883 3A1F28         Ø7690 SETBAUD LD      A,(BRESP)       ;Get baud response
2886 B7             Ø7700         OR      A               ;Anything?
2887 28Ø7           Ø7710         JR      Z,SETWORD       ;Nope, go
2889 CDC229         Ø7720         CALL    CKBAUD          ;Get baud setting
288C CØ             Ø7730         RET     NZ              ;Go if error
288D DD77Ø6         Ø7740         LD      (IX+BAUDRT),A   ;  else save new baud
                    Ø7750 ;
```

```
                    07760 ;       Evaluate WORD
                    07770 ;
2890 DD4E05         07780 SETWORD LD      C,(IX+UCIMAGE)  ;P/u current setings
2893 3A2728         07790         LD      A,(WRESP)       ;Get word response
2896 B7             07800         OR      A               ;Anything?
2897 281C           07810         JR      Z,SETSTOP       ;Nope, continue
2899 ED5B5027       07820         LD      DE,(WORDP)      ;Get parameter
289D 14             07830         INC     D               ;Check msb
289E 15             07840         DEC     D               ;D<>0?
289F C0             07850         RET     NZ              ;>256?
28A0 7B             07860         LD      A,E             ;Get lsb
28A1 D605           07870         SUB     5               ;Adjust 0 relative
28A3 D8             07880         RET     C               ;Go if out of range
28A4 FE04           07890         CP      4               ;5-8?
28A6 D0             07900         RET     NC              ;Out of range
28A7 B7             07910         OR      A               ;Clear carry, set PV
28A8 EAAD28         07920         JP      PE,$+5          ;Go if 5/8
28AB EE03           07930         XOR     3               ;Change 6=>7=>6
28AD 0F             07940         RRCA                    ;Align to bits 6/5
28AE 0F             07950         RRCA
28AF 0F             07960         RRCA
28B0 0660           07970         LD      B,01100000B     ;For bit setup
28B2 CDAB29         07980         CALL    SETBITS         ;Setup in C register
                    07990 ;
                    08000 ;       Evaluate STOP
                    08010 ;
28B5 3A2F28         08020 SETSTOP LD      A,(SRESP)       ;Get response byte
28B8 B7             08030         OR      A               ;Anything?
28B9 2817           08040         JR      Z,SETPAR        ;Nope, continue
28BB ED5B5227       08050         LD      DE,(STOPP)      ;Get input
28BF 14             08060         INC     D               ;Check for out of range
28C0 15             08070         DEC     D               ;D<>0?
28C1 C0             08080         RET     NZ              ;>256
28C2 7B             08090         LD      A,E             ;Get lsb
28C3 B7             08100         OR      A               ;0?
28C4 C8             08110         RET     Z               ;Invalid if not
28C5 FE03           08120         CP      3               ;1/2?
28C7 D0             08130         RET     NC              ;Invalid
28C8 3D             08140         DEC     A               ;Change 1|2 to 0|1
28C9 07             08150         RLCA                    ;Align result
28CA 07             08160         RLCA
28CB 07             08170         RLCA
28CC 07             08180         RLCA
28CD 0610           08190         LD      B,00010000B     ;Init mask
28CF CDAB29         08200         CALL    SETBITS         ;Setup in C register
                    08210 ;
                    08220 ;       Evaluate PARITY
                    08230 ;
28D2 3A3928         08240 SETPAR  LD      A,(PRESP)       ;Get parity response
28D5 B7             08250         OR      A               ;Any input?
28D6 2825           08260         JR      Z,SETPEO        ;Nope, check EVEN|ODD
                    08270 ;
                    08280 ;       Check if response was string or switch
                    08290 ;
28D8 ED5B5427       08300         LD      DE,(PARITYP)    ;Get user pointer
28DC CB77           08310         BIT     6,A             ;Switch?
28DE 2015           08320         JR      NZ,SETPOF       ;Yes, set ON/OFF
                    08330 ;
                    08340 ;       Response was string, check 'E'|'O'
                    08350 ;
28E0 1A             08360         LD      A,(DE)          ;Get input
```

```
28E1 CDB929    Ø837Ø          CALL    UCASE           ;Make upper case
28E4 FE4F      Ø838Ø          CP      'O'             ;Odd?
28E6 28Ø7      Ø839Ø          JR      Z,SETPO         ;Yes, go
28E8 FE45      Ø84ØØ          CP      'E'             ;Even?
28EA CØ        Ø841Ø          RET     NZ              ;Neither, invalid
28EB CBF9      Ø842Ø SETPE    SET     7,C             ;Set EVEN parity
28ED 18Ø2      Ø843Ø          JR      SETPAEO         ;Continue
28EF CBB9      Ø844Ø SETPO    RES     7,C             ;Set ODD parity
28F1 CB99      Ø845Ø SETPAEO  RES     3,C             ;Set ON parity
28F3 182C      Ø846Ø          JR      SETBRK          ;Continue
               Ø847Ø ;
               Ø848Ø ;        Response was switch, check ON|OFF
               Ø849Ø ;
28F5 7A        Ø85ØØ SETPOF   LD      A,D             ;Check input
28F6 B3        Ø851Ø          OR      E               ;=ØØØØ?
28F7 CBD9      Ø852Ø          SET     3,C             ;Set OFF
28F9 2826      Ø853Ø          JR      Z,SETBRK        ;Continue if yes
28FB CB99      Ø854Ø          RES     3,C             ;Set ON
               Ø855Ø ;
               Ø856Ø ;        Check for ODD|EVEN params
               Ø857Ø ;
28FD 3A4828    Ø858Ø SETPEO   LD      A,(ORESP)       ;Get odd response
2900 B7        Ø859Ø          OR      A               ;Anything?
2901 28ØC      Ø86ØØ          JR      Z,SETPAE        ;Nope, check EVEN
2903 ED5B5827  Ø861Ø          LD      DE,(ODDP)       ;Get user input
2907 7A        Ø862Ø          LD      A,D             ;Check for nil
2908 B3        Ø863Ø          OR      E               ;DE=ØØØØ?
2909 CBF9      Ø864Ø          SET     7,C             ;ODD=off?
290B 28Ø2      Ø865Ø          JR      Z,SETPAE        ;Go if yes
290D CBB9      Ø866Ø          RES     7,C             ;ODD=on
               Ø867Ø ;
290F 3A4128    Ø868Ø SETPAE   LD      A,(ERESP)       ;Get even response
2912 B7        Ø869Ø          OR      A               ;Anything?
2913 28ØC      Ø87ØØ          JR      Z,SETBRK        ;Nope, continue
2915 ED5B5627  Ø871Ø          LD      DE,(EVENP)      ;Get param
2919 7A        Ø872Ø          LD      A,D             ;Check for nil
291A B3        Ø873Ø          OR      E               ;DE=ØØØØ?
291B CBB9      Ø874Ø          RES     7,C             ;EVEN=off?
291D 28Ø2      Ø875Ø          JR      Z,SETBRK        ;Go if yes
291F CBF9      Ø876Ø          SET     7,C             ;EVEN=on
               Ø877Ø ;
               Ø878Ø ;        Evaluate BREAK
               Ø879Ø ;
2921 3AØE28    Ø88ØØ SETBRK   LD      A,(BRRESP)      ;Get user response
2924 B7        Ø881Ø          OR      A               ;Anything?
2925 2818      Ø882Ø          JR      Z,SETDTR        ;Nope, continue
2927 ED5B4A27  Ø883Ø          LD      DE,(BREAKP)     ;Get user input
292B CB77      Ø884Ø          BIT     6,A             ;Switch?
292D 2ØØ5      Ø885Ø          JR      NZ,SETBOF       ;Yes, check ON|OFF
               Ø886Ø ;
               Ø887Ø ;        Value entered, check if in range
               Ø888Ø ;
292F 14        Ø889Ø          INC     D               ;Check msb
2930 15        Ø89ØØ          DEC     D               ;D<>Ø?
2931 CØ        Ø891Ø          RET     NZ              ;>256?
2932 18Ø8      Ø892Ø          JR      UPDBRK          ;Continue
               Ø893Ø ;
               Ø894Ø ;        Switch entered, check if on/off
               Ø895Ø ;
2934 7A        Ø896Ø SETBOF   LD      A,D             ;Get input
2935 B3        Ø897Ø          OR      E               ;DE=ØØØØ?
```

```
2936 1E8Ø      Ø898Ø          LD      E,8ØH           ;Default
2938 28Ø2      Ø899Ø          JR      Z,UPDBRK        ;Go if off
293A 1EØ3      Ø9ØØØ          LD      E,Ø3H           ;Default
293C DD73Ø7    Ø9Ø1Ø UPDBRK   LD      (IX+BRK),E      ;Update data
               Ø9Ø2Ø ;
               Ø9Ø3Ø ;        Evaluate DTR
               Ø9Ø4Ø ;
293F 3A4F28    Ø9Ø5Ø SETDTR   LD      A,(DTRESP)      ;Get user response byte
2942 B7        Ø9Ø6Ø          OR      A               ;Anything?
2943 28ØC      Ø9Ø7Ø          JR      Z,SETRTS        ;Continue if not
2945 ED5B5A27  Ø9Ø8Ø          LD      DE,(DTRP)       ;Get user param
2949 7A        Ø9Ø9Ø          LD      A,D             ;Check if anything
294A B3        Ø91ØØ          OR      E               ;DE=ØØØØ?
294B CBC9      Ø911Ø          SET     1,C             ;Set OFF
294D 28Ø2      Ø912Ø          JR      Z,SETRTS        ;Go if yes
294F CB89      Ø913Ø          RES     1,C             ;Set ON
               Ø914Ø ;
               Ø915Ø ;        Evaluate RTS
               Ø916Ø ;
2951 3A5628    Ø917Ø SETRTS   LD      A,(RTRESP)      ;Get user response byte
2954 B7        Ø918Ø          OR      A               ;Anything?
2955 28ØC      Ø919Ø          JR      Z,UPDCI         ;Continue if not
2957 ED5B5C27  Ø92ØØ          LD      DE,(RTSP)       ;Get user param
295B 7A        Ø921Ø          LD      A,D             ;Check if Ø
295C B3        Ø922Ø          OR      E               ;DE=ØØØØ?
295D CBC1      Ø923Ø          SET     Ø,C             ;Set OFF
295F 28Ø2      Ø924Ø          JR      Z,UPDCI         ;Go if yes
2961 CB81      Ø925Ø          RES     Ø,C             ;Set ON
2963 CBD1      Ø926Ø UPDCI    SET     2,C             ;Enable transmit
2965 DD71Ø5    Ø927Ø          LD      (IX+UCIMAGE),C  ;Update new data
               Ø928Ø ;
               Ø929Ø ;        Evaluate RI
               Ø93ØØ ;
2968 215C28    Ø931Ø SETRI    LD      HL,RIRESP       ;Response byte
296B ØEØ1      Ø932Ø          LD      C,ØØØØØØØ1B     ;Bit to set
296D CD8A29    Ø933Ø          CALL    SETIT           ;Setup data
               Ø934Ø ;
               Ø935Ø ;        Evaluate DSR
               Ø936Ø ;
297Ø 216928    Ø937Ø SETDSR   LD      HL,DSRESP       ;Response byte
2973 ØEØ4      Ø938Ø          LD      C,ØØØØØ1ØØB     ;Bit mask
2975 CD8A29    Ø939Ø          CALL    SETIT           ;Setup data
               Ø94ØØ ;
               Ø941Ø ;        Evaluate CD
               Ø942Ø ;
2978 216228    Ø943Ø SETCD    LD      HL,CDRESP       ;Response byte
297B ØEØ2      Ø944Ø          LD      C,ØØØØØØ1ØB     ;Bit mask
297D CD8A29    Ø945Ø          CALL    SETIT           ;Setup data
               Ø946Ø ;
               Ø947Ø ;        Evaluate CTS
               Ø948Ø ;
298Ø 217Ø28    Ø949Ø SETCTS   LD      HL,CTRESP       ;Response byte
2983 ØEØ8      Ø95ØØ          LD      C,ØØØØ1ØØØB     ;Bit mask
2985 CD8A29    Ø951Ø          CALL    SETIT           ;Setup data
               Ø952Ø ;
               Ø953Ø ;        Evaluation complete
               Ø954Ø ;
2988 F1        Ø955Ø          POP     AF              ;Remove param error vect
2989 C9        Ø956Ø          RET                     ;Return from evaluate
               Ø957Ø ;
298A 34        Ø958Ø SETIT    INC     (HL)            ;Any response?
```

```
298B 35        Ø959Ø          DEC     (HL)              ;(HL) = ØØ?
298C C8        Ø96ØØ          RET     Z                 ;Back if no change
298D 23        Ø961Ø          INC     HL                ;Point to response data
298E CDB429    Ø962Ø          CALL    GETHL             ;Load HL with (HL)
2991 CDB429    Ø963Ø          CALL    GETHL             ;Get user response
2994 79        Ø964Ø          LD      A,C               ;Get response data ON
2995 Ø7        Ø965Ø          RLCA                      ;Align to upper 4 bits
2996 Ø7        Ø966Ø          RLCA
2997 Ø7        Ø967Ø          RLCA
2998 Ø7        Ø968Ø          RLCA
2999 B1        Ø969Ø          OR      C                 ;Combine
299A 47        Ø97ØØ          LD      B,A               ;Set response data OFF
299B 2F        Ø971Ø          CPL                       ;Reverse bits
299C DDA6Ø4    Ø972Ø          AND     (IX+MSMASK)       ;Drop desired bits
299F 5F        Ø973Ø          LD      E,A               ;Save new mask
29AØ 7C        Ø974Ø          LD      A,H               ;Check if OFF
29A1 B5        Ø975Ø          OR      L                 ;HL = ØØØØ?
29A2 78        Ø976Ø          LD      A,B               ;Get OFF bits
29A3 28Ø1      Ø977Ø          JR      Z,$+3             ;Go if off
29A5 79        Ø978Ø          LD      A,C               ;Get ON bits
29A6 B3        Ø979Ø          OR      E                 ;Combine with remainder
29A7 DD77Ø4    Ø98ØØ          LD      (IX+MSMASK),A     ;Update mask
29AA C9        Ø981Ø          RET                       ;Done
               Ø982Ø ;
               Ø983Ø ;        Set bits in C from A using B as mask
               Ø984Ø ;
29AB AØ        Ø985Ø SETBITS  AND     B                 ;Mask others
29AC F5        Ø986Ø          PUSH    AF                ;Save data
29AD 78        Ø987Ø          LD      A,B               ;Get mask
29AE 2F        Ø988Ø          CPL                       ;Reverse for mask off
29AF A1        Ø989Ø          AND     C                 ;Remove undesired bits
29BØ C1        Ø99ØØ          POP     BC                ;B = new bits
29B1 BØ        Ø991Ø          OR      B                 ;Combine
29B2 4F        Ø992Ø          LD      C,A               ;Update
29B3 C9        Ø993Ø          RET                       ;New bits set
               Ø994Ø ;
               Ø995Ø ;        Fetch HL from (HL)
               Ø996Ø ;
29B4 7E        Ø997Ø GETHL    LD      A,(HL)            ;Get lsb
29B5 23        Ø998Ø          INC     HL                ;Bump pointer
29B6 66        Ø999Ø          LD      H,(HL)            ;Get msb
29B7 6F        1ØØØØ          LD      L,A               ;HL = (HL)
29B8 C9        1ØØ1Ø          RET                       ;Done
               1ØØ2Ø ;
               1ØØ3Ø ;        Convert char in A to upper case
               1ØØ4Ø ;
29B9 FE61      1ØØ5Ø UCASE    CP      'a'               ;In range?
29BB D8        1ØØ6Ø          RET     C                 ;Nope, go
29BC FE7B      1ØØ7Ø          CP      'z'+1             ;In range?
29BE DØ        1ØØ8Ø          RET     NC                ;Nope, go
29BF E65F      1ØØ9Ø          AND     5FH               ;  else make upper case
29C1 C9        1Ø1ØØ          RET                       ;Done
               1Ø11Ø ;
               1Ø12Ø ;        Check for valid baud rate entered
               1Ø13Ø ;
29C2 ED5B4E27  1Ø14Ø CKBAUD   LD      DE,(BAUDP)        ;Baud parm (default 3ØØ)
29C6 21FA26    1Ø15Ø          LD      HL,BAUDTBL        ;Point to baud table
29C9 Ø1ØØ1Ø    1Ø16Ø          LD      BC,16<8+Ø         ;B=count, C=position
               1Ø17Ø ;
29CC 7E        1Ø18Ø BLOOP    LD      A,(HL)            ;Fetch LSB baud
29CD 23        1Ø19Ø          INC     HL                ;Bump table
```

```
29CE BB       10200          CP      E               ;Lsb match?
29CF 2004     10210          JR      NZ,NOMATB       ;Nope, go next entry
29D1 7E       10220          LD      A,(HL)          ;Get MSB baud
29D2 BA       10230          CP      D               ;Msb match?
29D3 2805     10240          JR      Z,MATCHB        ;Yes, baud found
              10250 ;
29D5 23       10260 NOMATB   INC     HL              ;Bump to next entry
29D6 0C       10270          INC     C               ;Bump position count
29D7 10F3     10280          DJNZ    BLOOP           ;Go for table length
29D9 C9       10290          RET                     ;Param error
              10300 ;
29DA 79       10310 MATCHB   LD      A,C             ;Pick up baud rate code
29DB 07       10320          RLCA                    ;Align to high bits
29DC 07       10330          RLCA
29DD 07       10340          RLCA
29DE 07       10350          RLCA
29DF B1       10360          OR      C               ;Use for xmit and rcv
29E0 BF       10370          CP      A               ;Z=good value
29E1 C9       10380          RET                     ;Return with BAUD setting
              10390 ;
              10400 ;        Nil params entered, display current settings
              10410 ;
29E2          10420 SHOW     EQU     $
              10430          IF      @MOD2
              10440          LD      A,-1            ;Set NO flag
              10450          LD      (SHOWF),A       ;Save flag
              10460          LD      DE,MDNAME1      ;CL1
              10470          LD      A,'A'           ;Comm A
              10480          LD      (MSG1A),A       ;To text
              10490          CALL    SHOW1           ;Display
              10500          LD      DE,MDNAME2      ;CL2
              10510          LD      A,'B'           ;Comm B
              10520          LD      (MSG1A),A       ;To text
              10530          CALL    SHOW1           ;Display
              10540          LD      A,-1            ;Get flag
              10550 SHOWF    EQU     $-1
              10560          OR      A               ;Any found?
              10570          JP      NZ,BADMOD       ;Not installed
              10580          RET                     ;  else OK
              10590 ;
29E2          10600 SHOW1    @@GTMOD                 ;Locate module
              00037          LD      A,83
              00038          RST     40
              10610          RET     NZ              ;Module not found
              10620          XOR     A               ;Set module found
              10630          LD      (SHOWF),A       ;Init flag
              10640          PUSH    DE              ;Pass to IX
              10650          POP     IX              ;IX => module
              10660          ENDIF
              10670 ;
29E2 C33126   10680          JP      DSPALL          ;Display all settings
              10690 ;
              10700 ;        Mini param block for 'query' evaluation
              10710 ;
29E5 80       10720 PTBL2    DB      80H             ;Extended param
              10730 ;
29E6 C1       10740 FTYP     DB      FLAG!NUM!1
29E7 46       10750          DB      'F'
29E8 00       10760 FRESP    DB      0
29E9 6A27     10770          DW      FPARM
              10780 ;
```

```
29EB ØØ         1Ø79Ø          DB      Ø
                1Ø8ØØ ;
                1Ø81Ø ;        Text string passed to @PARAM for query eval
                1Ø82Ø ;
29EC 28         1Ø83Ø FSTR     DB      '(F='
     46 3D
29EF            1Ø84Ø INBUF    EQU     $               ;Keyboard input buffer
                1Ø85Ø ;
24ØØ            1Ø86Ø          END     BEGIN
```

$ABORT	2411	$DSP	2416
$EXIT	24Ø7	@@1	ØØØØ
@@3	ØØØØ	@@4	ØØØØ
@MOD4	FFFF	ABB	ØØ1Ø
ASKBRK	24E1	ASKCD	25ØE
ASKDSR	25Ø5	ASKDTR	24EA
ASKRI	24FC	ASKRTS	24F3
ASKWORD	2484	BADBAUD	247E
BADMOD	242B	BADPAR	24BB
BADWORD	249A	BAUDP	274E
BAUDTBL	26FA	BEGIN	24ØØ
BREAKP	274A	BRESP	281F
BRRESP	28ØE	BRTYP	28Ø8
CDP	276Ø	CDRESP	2862
CKBAUD	29C2	CKPARM	252Ø
CTRESP	2870	CTSP	2764
DEFALT	26CC	DEFBA	ØØ55
DEFMS	ØØØØ	DEFUC	ØØA5
DMSGØ	26ØC	DMSGØX	2622
DMSG1Ø	25EC	DMSG11	25F2
DMSG13	26ØØ	DMSG14	26Ø7
DMSG22	2625	DMSG23	262B
DMSG4	257F	DMSG5	2591
DMSG7	25C5	DMSG8	25CB
DMSG9	25D7	DSPALL	2631
DSPRET	2647	DSPSET	2637
DSRESP	2869	DSRP	2762
DTRESP	284F	DTRP	275A
ERESP	2841	ERROR$	2427
EVENP	2756	FLAG	ØØ4Ø
FRESP	29E8	FSTR	29EC
GETHL	29B4	GETIT	265Ø
GETRSPR	267A	INBUF	29EF
MATCHB	29DA	MDNAME	26DD
MSG1Ø	27BD	MSG11	27CE
MSG13	27D7	MSG14	27DB
MSG16	27E3	MSG17	27E7
MSG19	27FØ	MSG2	278Ø
MSG21	27FD	MSG21A	27FF
MSG23	28Ø4	MSG3	2786
MSG5	2792	MSG6	279A
MSG8	27B3	MSG9	27B8
MVUP	2666	NOMATB	29D5
ODDP	2758	OFFSET	ØØØ4
OTYP	2844	PARITYP	2754
PGRM	243A	PORTD	2768
PRESP	2839	PRMERR	2435
PROMPT	26F7	PTBL2	29E5
QPARM	274C	QRESP	2817
RIRESP	285C	RITYP	2859
RTSP	275C	RTYP	2852
SETBAUD	2883	SETBITS	29AB
SETBRK	2921	SETCD	2978
SETDSR	297Ø	SETDTR	293F
SETPAE	29ØF	SETPAEO	28F1
SETPARAM	287F	SETPE	28EB
SETPO	28EF	SETPOF	28F5
SETRTS	2951	SETSTOP	28B5
SHOW	29E2	SHOW1	29E2
SRESP	282F	STOPP	2752

$DSPLY	241C
@@2	ØØØØ
@MOD2	ØØØØ
ASKBAUD	246E
ASKCTS	2517
ASKPAR	24BF
ASKSTOP	24AØ
BADDCB	26E1
BADSTOP	24B5
BAUDRT	ØØØ6
BLOOP	29CC
BRK	ØØØ7
BTYP	281A
CDTYP	285F
CKRSP	2663
CTTYP	286C
DEFBR	ØØØ3
DFPARM	2766
DMSG1	252F
DMSG12	25F9
DMSG2	2538
DMSG3	2566
DMSG6	25BØ
DMSG89	25E1
DSPDONE	264D
DSPTBL	271A
DSTYP	2865
DTYP	284B
ETYP	283C
FPARM	276A
FTYP	29E6
GETRSP	2674
IOERR	242Ø
MSG1	276C
MSG12	27D2
MSG15	27EØ
MSG18	27EB
MSG2Ø	27F7
MSG22	28Ø3
MSG4	278C
MSG7	27A1
MSMASK	ØØØ4
NUM	ØØ8Ø
ORESP	2848
PAR ERR	ØØ2C
PORTF	245C
PRMTBL$	28Ø7
PTYP	2832
RIP	275E
RTRESP	2856
SAVESP	24ØA
SETBOF	2934
SETCTS	298Ø
SETIT	298A
SETPAR	28D2
SETPEO	28FD
SETRI	2968
SETWORD	289Ø
SPLP	2448
STR	ØØ2Ø

```
STRIP1          255E STYP           282A TTYP           26AØ
UCASE           29B9 UCIMAGE        ØØØ5 UP             266A
UPDBRK          293C UPDCI          2963 WORDP          275Ø
WRESP           2827 WTYP           2822 _APOS          ØØ27
_CR             ØØØD _ETX           ØØØ3 _INIT          ØØØ2
_LF             ØØØA @@ABORT        C2EB @@ADTSK        C37E
@@BANK          C896 @@BKSP         C576 @@BREAK        C8AC
@@CHNIO         C2D6 @@CKBRKC       C8FA @@CKDRV        C3D2
@@CKEOF         C58B @@CKTSK        C369 @@CLOSE        C561
@@CLS           C8E4 @@CMNDI        C315 @@CMNDR        C32A
@@CTL           C13A @@DATE         C2AC @@DCSTAT       C411
@@DEBUG         C354 @@DECHEX       C816 @@DIRRD        C783
@@DIRWR         C798 @@DIV16        C8Ø1 @@DIV8         C7EC
@@DODIR         C3E7 @@DSP          CØFE @@DSPLY        C19E
@@ERROR         C33F @@EXIT         C3ØØ @@FEXT         C6FØ
@@FLAGS         C88Ø @@FNAME        C7Ø5 @@FSPEC        C6DB
@@GATRD         C76E @@GATWR        C7AD @@GET          C112
@@GTDCB         C72F @@GTDCT        C71A @@GTMOD        C744
@@HDFMT         C4B9 @@HEX16        C855 @@HEX8         C84Ø
@@HEXDEC        C82B @@HIGH$        C86A @@INIT         C537
@@KBD           C176 @@KEY          CØEA @@KEYIN        C18A
@@KLTSK         C3BD @@LOAD         C6B1 @@LOC          C5AØ
@@LOF           C5B5 @@LOGER        C1D5 @@LOGOT        C1EA
@@MSG           C221 @@MUL16        C7D7 @@MUL8         C7C2
@@OPEN          C54C @@PARAM        C297 @@PAUSE        C282
@@PEOF          C5CA @@POSN         C5DF @@PRINT        C236
@@PRT           C14E @@PUT          C126 @@RAMDIR       C3FC
@@RDSEC         C48F @@RDSSC        C759 @@READ         C5F4
@@REMOV         C522 @@RENAM        C5ØD @@REW          C6Ø9
@@RMTSK         C393 @@RPTSK        C3A8 @@RREAD        C61E
@@RSLCT         C47A @@RSTOR        C43B @@RUN          C6C6
@@RWRIT         C633 @@SEEK         C465 @@SEEKSC       C648
@@SKIP          C65D @@SLCT         C426 @@STEPI        C45Ø
@@TIME          C2C1 @@VDCTL        C26D @@VER          C672
@@VRSEC         C4A4 @@WEOF         C687 @@WHERE        C162
@@WRITE         C69C @@WRSEC        C4CE @@WRSSC        C4E3
@@WRTRK         C4F8
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: SETKI

Library: SYS8/SYS

ISAM # : B3H

```
                00100 ;LBSETKI/ASM - Set Keyboard Parameters
0000            00110         TITLE   <SETKI - LS-DOS 6.2>
                00120 ;
0004            00130 OFFSET  EQU     4               ;Length: end of name to data area
0006            00140 DELAY   EQU     OFFSET+2
0007            00150 RPEAT   EQU     OFFSET+3
000A            00160 DMIN    EQU     10
0001            00170 RMIN    EQU     1
0016            00180 DDFALT  EQU     22
0002            00190 RDFALT  EQU     2
0003            00200 ETX     EQU     03H
000D            00210 CR      EQU     0DH
000A            00220 LF      EQU     0AH
002C            00230 PAR_ERR EQU     44              ;Parameter Error
                00240 ;
0040            00250 FLAG    EQU     01000000B
0010            00260 ABB     EQU     00010000B
0080            00270 NUM     EQU     10000000B
                00280 ;
0000            00290 *GET    SVCMAC:3                ;SVC Macro equivalents
                00010 ;SVCMAC/ASM - LS-DOS Version VI
                00020 *LIST   OFF
                03900 *LIST   ON
                00300 ;
2400            00310         ORG     2400H
                00320 ;
2400            00330 BEGIN   EQU     $
2400 ED730B24   00340         LD      (SAVESP+1),SP
2404 CD2E24     00350         CALL    PGRM            ;Exit via RET
                00360 ;
                00370 ;       Set exit conditions
                00380 ;
2407 210000     00390 $EXIT   LD      HL,0            ;Init to no error
240A 310000     00400 SAVESP  LD      SP,$-$          ;P/u original SP
240D            00410         @@CKBRKC                ;Clear any <BREAK>
240D 3E6A       00001         LD      A,106
240F EF         00002         RST     40
2410 C9         00420         RET
2411 21FFFF     00430 $ABORT  LD      HL,-1           ;Set abort code
2414 18F4       00440         JR      SAVESP
                00450 ;
2416            00460 $DSP    @@DSP                   ;Display a character
2416 3E02       00003         LD      A,2
2418 EF         00004         RST     40
2419 C8         00470         RET     Z               ;Back if good
241A 1807       00480         JR      IOERR
241C            00490 $DSPLY  @@DSPLY                 ;Display a line
                00005         IFEQ    00H,1
                00006         LD      HL,
                00007         ENDIF
241C 3E0A       00008         LD      A,10
241E EF         00009         RST     40
241F C8         00500         RET     Z               ;Back if good
2420 21         00510         DB      21H             ;Skip LD A,##
2421 3E2C       00520 PRMERR  LD      A,PAR_ERR       ;Parameter Error
                00530 ;
                00540 ;       I/O Error Processing
                00550 ;
2423 2600       00560 IOERR   LD      H,0
2425 6F         00570         LD      L,A             ;Save error #
2426 F6C0       00580         OR      0C0H
```

```
2428 4F       00590          LD      C,A
2429          00600          @@ERROR                 ;Display message
2429 3E1A     00010          LD      A,26
242B EF       00011          RST     40
242C 18DC     00610          JR      SAVESP          ;Exit
              00620 ;
              00630 ;        PGRM - Set Keyboard Parameters
              00640 ;
242E E5       00650 PGRM     PUSH    HL              ;Save cmdline ptr
242F 113925   00660          LD      DE,MDNAME       ;Name of keyboard driver
2432          00670          @@GTMOD                 ;Find module header
2432 3E53     00012          LD      A,83
2434 EF       00013          RST     40
2435 3E08     00680          LD      A,8             ;Device not available
2437 C22324   00690          JP      NZ,IOERR        ;Exit if not found
243A D5       00700          PUSH    DE              ;Point to next byte
243B DDE1     00710          POP     IX              ;  after module name
              00720 ;
243D E1       00730          POP     HL              ;=>cmd line
243E CD1725   00740          CALL    SKSP            ;Move to non-space
2441 7E       00750          LD      A,(HL)          ;Char fm cmd line
2442 FE28     00760          CP      '('             ;Any params?
2444 CA6224   00770          JP      Z,GETNEW        ;Get/set new values
              00780 ;
              00790 ;        Display old values
              00800 ;
2447 CD5024   00810          CALL    SETMSG          ;Move old values for dsply
244A 211E25   00820          LD      HL,DMSG         ;Pt to dsply string
244D C31C24   00830          JP      $DSPLY          ;Dsply and exit
              00840 ;
              00850 ;        Set up values in string
              00860 ;
2450 DD7E06   00870 SETMSG   LD      A,(IX+DELAY)    ;P/u old Wait value
2453 112725   00880          LD      DE,DDELAY       ;=>buffer to receive
2456 CD6625   00890          CALL    HEXDEC          ;Convert to ASCII decimal
2459 DD7E07   00900          LD      A,(IX+RPEAT)    ;P/u old Rate value
245C 113525   00910          LD      DE,DRPEAT       ;Pt to dsply area
245F C36625   00920          JP      HEXDEC          ;Convert to decimal ASCII
              00930 ;
2462 114025   00940 GETNEW   LD      DE,PRMTBL$      ;Check user parameters
2465          00950          @@PARAM
2465 3E11     00014          LD      A,17
2467 EF       00015          RST     40
2468 C22124   00960          JP      NZ,PRMERR       ;Go on "Parm error
246B 110000   00970          LD      DE,$-$          ;P/u D parm
246C          00980 DFPARM   EQU     $-2
246E 7A       00990          LD      A,D             ;Default setting wanted?
246F B3       01000          OR      E
2470 2808     01010          JR      Z,SETSTR        ;Keep existing if not
2472 DD360616 01020          LD      (IX+DELAY),DDFALT       ;Stuff defaults
2476 DD360702 01030          LD      (IX+RPEAT),RDFALT
247A CD5024   01040 SETSTR   CALL    SETMSG          ;Put numbers in string
247D 110000   01050 QCHK     LD      DE,$-$
247E          01060 QPARM    EQU     $-2             ;Query parm used?
2480 7A       01070          LD      A,D
2481 B3       01080          OR      E
2482 2834     01090          JR      Z,CKPARM        ;Go if not
              01100 ;
2484 3A5925   01110          LD      A,(WRESP)       ;P/u Wait response byte
2487 E680     01120          AND     NUM             ;If a "W" number given,
2489 2013     01130          JR      NZ,CKR          ;  then ask for R only
```

```
248B 211E25   Ø1140 GETD    LD      HL,DMSG         ;Pt to Wait prompt msg
248E CDDE24   Ø1150         CALL    SHOW            ;Ask for W time
2491 CDAA25   Ø1160         CALL    GETIT           ;Get it (in DE)
2494 28Ø8     Ø1170         JR      Z,CKR           ;Don't change if CR only
2496 CDA125   Ø1180         CALL    DCHECK          ;Check if good value
2499 2ØFØ     Ø1190         JR      NZ,GETD         ;Don't change if not
249B DD77Ø6   Ø1200         LD      (IX+DELAY),A    ;Store new Wait
              Ø1210 ;
249E 3A5125   Ø1220 CKR     LD      A,(RRESP)       ;P/u Rate resopnse byte
24A1 E68Ø     Ø1230         AND     NUM             ;  and go if Rate
24A3 2Ø13     Ø1240         JR      NZ,CKPARM       ;  entered on cmdline
24A5 212C25   Ø1250         LD      HL,RMSG         ;Pt to Rate prompt msg
24A8 CDDE24   Ø1260         CALL    SHOW
24AB CDAA25   Ø1270         CALL    GETIT           ;Get new Rate
24AE 28Ø8     Ø1280         JR      Z,CKPARM        ;Don't change if CR only
24BØ CD9D25   Ø1290         CALL    RCHECK          ;Check range & don't
24B3 2ØE9     Ø1300         JR      NZ,CKR          ;  change if out of range
24B5 DD77Ø7   Ø1310         LD      (IX+RPEAT),A    ;Store new Rate
              Ø1320 ;
              Ø1330 ;       Check entries given on cmd line
              Ø1340 ;
24B8 3A5925   Ø1350 CKPARM  LD      A,(WRESP)
24BB E68Ø     Ø1360         AND     NUM             ;W parm response
24BD 28ØC     Ø1370         JR      Z,CKRP          ;Go if not entered
24BF 11ØØØØ   Ø1380         LD      DE,$-$          ;P/u Wait value
24CØ          Ø1390 WPARM   EQU     $-2
24C2 CDA125   Ø1400         CALL    DCHECK          ;Check value
24C5 C22124   Ø1410         JP      NZ,PRMERR       ;Go if bad
24C8 DD77Ø6   Ø1420         LD      (IX+DELAY),A    ;  else store it
              Ø1430 ;
24CB 3A5125   Ø1440 CKRP    LD      A,(RRESP)       ;Rate response
24CE E68Ø     Ø1450         AND     NUM
24DØ C8       Ø1460         RET     Z               ;Done if none
24D1 11ØØØØ   Ø1470         LD      DE,$-$          ;P/u Rate value
24D2          Ø1480 RPARM   EQU     $-2
24D4 CD9D25   Ø1490         CALL    RCHECK          ;Check value
24D7 C22124   Ø1500         JP      NZ,PRMERR       ;  and go if bad
24DA DD77Ø7   Ø1510         LD      (IX+RPEAT),A    ;  else store it
24DD C9       Ø1520         RET                     ;Done
              Ø1530 ;
24DE Ø6ØC     Ø1540 SHOW    LD      B,12            ;Col posn
24EØ 4E       Ø1550 SH2     LD      C,(HL)          ;Get char in C
24E1 CD1624   Ø1560         CALL    $DSP            ;Print byte fm string
24E4 Ø5       Ø1570         DEC     B               ;Dec chars to print
24E5 23       Ø1580         INC     HL              ;Bump string ptr
24E6 3E3D     Ø1590         LD      A,'='
24E8 B9       Ø16ØØ         CP      C
24E9 2ØF5     Ø1610         JR      NZ,SH2          ;Display up to =
24EB ØE2Ø     Ø1620         LD      C,' '           ;Then space
24ED CD1624   Ø1630         CALL    $DSP
24FØ CD1725   Ø1640         CALL    SKSP            ;Move to number
24F3 ØE7B     Ø1650         LD      C,'{'
24F5 CD1624   Ø1660         CALL    $DSP            ;Dsply opening brace
24F8 4E       Ø1670 SH3     LD      C,(HL)          ;P/u character
24F9 3E2F     Ø1680         LD      A,'Ø'-1         ;Check C for numeric value
24FB B9       Ø1690         CP      C
24FC 3ØØ7     Ø17ØØ         JR      NC,SH4          ;Go if not
24FE CD1624   Ø1710         CALL    $DSP            ;  else dsply,
25Ø1 Ø5       Ø1720         DEC     B               ;  dec chars remaining,
25Ø2 23       Ø1730         INC     HL              ;  pt to next char in string
25Ø3 18F3     Ø1740         JR      SH3             ;  and loop
```

```
25Ø5 ØE7D      Ø175Ø SH4      LD      C,'}'           ;Display closing brace
25Ø7 CD1624    Ø176Ø          CALL    $DSP
25ØA ØE2Ø      Ø177Ø          LD      C,' '
25ØC CD1624    Ø178Ø SH5      CALL    $DSP            ;Tab remaining distance
25ØF 1ØFB      Ø179Ø          DJNZ    SH5
2511 213D25    Ø18ØØ          LD      HL,ENDSTR       ;Print "?"
2514 C31C24    Ø181Ø          JP      $DSPLY
               Ø182Ø ;
2517 3E2Ø      Ø183Ø SKSP     LD      A,' '           ;Bypass all leading spaces
2519 BE        Ø184Ø SKP2     CP      (HL)
251A CØ        Ø185Ø          RET     NZ
251B 23        Ø186Ø          INC     HL
251C 18FB      Ø187Ø          JR      SKP2
               Ø188Ø ;
               Ø189Ø ;
               Ø19ØØ ;
251E 57        Ø191Ø DMSG     DB      'Wait   = '
     61 69 74 2Ø 2Ø 2Ø 3D 2Ø
2527 2Ø        Ø192Ø DDELAY   DB      '   , '
     2Ø 2Ø 2C 2Ø
252C 52        Ø193Ø RMSG     DB      'Rate   = '
     61 74 65 2Ø 2Ø 2Ø 3D 2Ø
2535 2Ø        Ø194Ø DRPEAT   DB      '   ',CR
     2Ø 2Ø ØD
2539 24        Ø195Ø MDNAME   DB      '$KI',ETX
     4B 49 Ø3
253D 3F        Ø196Ø ENDSTR   DB      '? ',ETX
     2Ø Ø3
               Ø197Ø ;
254Ø 8Ø        Ø198Ø PRMTBL$  DB      8ØH
               Ø199Ø ;
2541 57        Ø2ØØØ          DB      FLAG!ABB!7
2542 44        Ø2Ø1Ø          DB      'DEFAULT'
     45 46 41 55 4C 54
2549 ØØ        Ø2Ø2Ø          DB      Ø
254A 6C24      Ø2Ø3Ø          DW      DFPARM
               Ø2Ø4Ø ;
254C 94        Ø2Ø5Ø          DB      ABB!NUM!4
254D 52        Ø2Ø6Ø          DB      'RATE'          ;Repeat key rate
     41 54 45
2551 ØØ        Ø2Ø7Ø RRESP    DB      Ø
2552 D224      Ø2Ø8Ø          DW      RPARM
               Ø2Ø9Ø ;
2554 94        Ø21ØØ          DB      ABB!NUM!4
2555 57        Ø211Ø          DB      'WAIT'          ;Delay before repeat
     41 49 54
2559 ØØ        Ø212Ø WRESP    DB      Ø
255A CØ24      Ø213Ø          DW      WPARM
               Ø214Ø ;
255C 55        Ø215Ø          DB      ABB!FLAG!5
255D 51        Ø216Ø          DB      'QUERY'
     55 45 52 59
2562 ØØ        Ø217Ø          DB      Ø
2563 7E24      Ø218Ø          DW      QPARM
               Ø219Ø ;
2565 ØØ        Ø22ØØ          NOP                     ;Note end of parm table
               Ø221Ø ;
               Ø222Ø ;
               Ø223Ø ;        HEXDEC - Convert Hex Number to Decimal ASCII
               Ø224Ø ;        A => 8-bit Hex Number to Convert
               Ø225Ø ;        DE => Destination of ASCII characters
```

```
                    Ø226Ø ;
                    Ø227Ø ;
2566 C5             Ø228Ø HEXDEC  PUSH    BC              ;Save regs
2567 E5             Ø229Ø         PUSH    HL
2568 F5             Ø23ØØ         PUSH    AF
                    Ø231Ø ;
                    Ø232Ø ;       Xfer number to HL
                    Ø233Ø ;
2569 26ØØ           Ø234Ø         LD      H,Ø             ;Set HL = #
256B 6F             Ø235Ø         LD      L,A
                    Ø236Ø ;
256C 3E2Ø           Ø237Ø         LD      A,' '           ;Character if leading Ø
256E Ø164ØØ         Ø238Ø         LD      BC,1ØØ          ;Set 1Ø's power
2571 CD8225         Ø239Ø         CALL    CVD1            ;Convert to ASCII
2574 Ø1ØAØØ         Ø24ØØ         LD      BC,1Ø
2577 CD8225         Ø241Ø         CALL    CVD1
257A 7D             Ø242Ø         LD      A,L             ;Get remainder
257B C63Ø           Ø243Ø         ADD     A,'Ø'           ;Make ASCII and
257D 12             Ø244Ø         LD      (DE),A          ;  stuff in buffer
                    Ø245Ø ;
257E F1             Ø246Ø         POP     AF              ;Recover #
257F E1             Ø247Ø         POP     HL              ;And other regs
258Ø C1             Ø248Ø         POP     BC
2581 C9             Ø249Ø         RET
                    Ø25ØØ ;
2582 D5             Ø251Ø CVD1    PUSH    DE              ;Save user buffer
2583 5F             Ø252Ø         LD      E,A             ;Save pad character
2584 16FF           Ø253Ø         LD      D,ØFFH          ;Init digit count to -1
2586 AF             Ø254Ø         XOR     A
2587 14             Ø255Ø CVD2    INC     D               ;Inc digit count and
2588 ED42           Ø256Ø         SBC     HL,BC           ;  sub 1Ø' power untill
258A 3ØFB           Ø257Ø         JR      NC,CVD2         ;  underflow
258C Ø9             Ø258Ø         ADD     HL,BC           ;Add back last sub
258D 7B             Ø259Ø         LD      A,E             ;Recover pad char
258E 42             Ø26ØØ         LD      B,D             ;Count to B
258F D1             Ø261Ø         POP     DE              ;Recover buffer ptr
2590 12             Ø262Ø         LD      (DE),A          ;Tempy store pad char
2591 Ø4             Ø263Ø         INC     B               ;See if char a Ø
2592 Ø5             Ø264Ø         DEC     B
2593 28Ø6           Ø265Ø         JR      Z,CVD3          ;Go if so
2595 78             Ø266Ø         LD      A,B             ;  else make digit ASCII
2596 C63Ø           Ø267Ø         ADD     A,'Ø'
2598 12             Ø268Ø         LD      (DE),A          ;  and put in buffer
2599 3E3Ø           Ø269Ø         LD      A,'Ø'           ;Change the pad char
259B 13             Ø27ØØ CVD3    INC     DE              ;Bump buffer
259C C9             Ø271Ø         RET
                    Ø272Ø ;
259D 16Ø1           Ø273Ø RCHECK  LD      D,RMIN          ;Rate minimum value
259F 18Ø2           Ø274Ø         JR      CHECK
25A1 16ØA           Ø275Ø DCHECK  LD      D,DMIN          ;Wait minimum value
25A3 7B             Ø276Ø CHECK   LD      A,E             ;Get number
25A4 E67F           Ø277Ø         AND     7FH             ;Keep positive
25A6 BA             Ø278Ø         CP      D               ;Lowest allowed
25A7 D8             Ø279Ø         RET     C               ;Too low
25A8 BF             Ø28ØØ         CP      A               ;Set Z if good value
25A9 C9             Ø281Ø         RET
                    Ø282Ø ;
25AA 21D725         Ø283Ø GETIT   LD      HL,INBUF        ;Key buffer
25AD Ø1ØØØ3         Ø284Ø         LD      BC,3<8          ;3 chars max
25BØ                Ø285Ø         @@KEYIN
25BØ 3EØ9           ØØØ16         LD      A,9
```

```
25B2 EF      00017          RST     40
25B3 DA1124  02860          JP      C,$ABORT        ;Quit if Break
             02870 ;
             02880 ;        DECHEX - Decimal ASCII to Hex
             02890 ;        DE <= returns the Hex num
             02900 ;        HL => points to the start of Dec asc string
             02910 ;        A <= # of characters converted
             02920 ;
             02930 ;
25B6 0600    02940 DECHEX   LD      B,0             ;Init counter to 0
25B8 50      02950          LD      D,B             ;Init ret valu to 0
25B9 58      02960          LD      E,B
25BA 7E      02970 CVDEC    LD      A,(HL)          ;P/u a character
25BB D630    02980          SUB     30H             ;Make binary
25BD FE0A    02990          CP      10
25BF 3013    03000          JR      NC,DONECON      ;Leave when non-decimal found
25C1 E5      03010          PUSH    HL
25C2 62      03020          LD      H,D
25C3 6B      03030          LD      L,E             ;Prev. total to HL
25C4 29      03040          ADD     HL,HL           ;X2
25C5 29      03050          ADD     HL,HL           ;X4
25C6 19      03060          ADD     HL,DE           ;X5
25C7 29      03070          ADD     HL,HL           ;X10
25C8 EB      03080          EX      DE,HL           ;Result back to DE
25C9 83      03090          ADD     A,E             ;Add in newest digit
25CA 5F      03100          LD      E,A
25CB 3E00    03110          LD      A,0
25CD 8A      03120          ADC     A,D
25CE 57      03130          LD      D,A
25CF E1      03140          POP     HL              ;Get buffer posn
25D0 23      03150          INC     HL
25D1 04      03160          INC     B               ;Inc chars found
25D2 18E6    03170          JR      CVDEC           ;  and continue next
             03180 ;
25D4 78      03190 DONECON  LD      A,B
25D5 B7      03200          OR      A
25D6 C9      03210          RET
25D7         03220 INBUF    EQU     $
             03230 ;
2400         03240          END     BEGIN
```

$ABORT	2411	$DSP	2416	$DSPLY	241C		
$EXIT	24Ø7	@@1	ØØØØ	@@2	ØØØØ		
@@3	ØØØØ	@@4	ØØØØ	@MOD2	ØØØØ		
@MOD4	FFFF	ABB	ØØ1Ø	BEGIN	24ØØ		
CHECK	25A3	CKPARM	24B8	CKR	249E		
CKRP	24CB	CR	ØØØD	CVD1	2582		
CVD2	2587	CVD3	259B	CVDEC	25BA		
DCHECK	25A1	DDELAY	2527	DDFALT	ØØ16		
DECHEX	25B6	DELAY	ØØØ6	DFPARM	246C		
DMIN	ØØØA	DMSG	251E	DONECON	25D4		
DRPEAT	2535	ENDSTR	253D	ETX	ØØØ3		
FLAG	ØØ4Ø	GETD	248B	GETIT	25AA		
GETNEW	2462	HEXDEC	2566	INBUF	25D7		
IOERR	2423	LF	ØØØA	MDNAME	2539		
NUM	ØØ8Ø	OFFSET	ØØØ4	PAR_ERR	ØØ2C		
PGRM	242E	PRMERR	2421	PRMTBL$	254Ø		
QCHK	247D	QPARM	247E	RCHECK	259D		
RDFALT	ØØØ2	RMIN	ØØØ1	RMSG	252C		
RPARM	24D2	RPEAT	ØØØ7	RRESP	2551		
SAVESP	24ØA	SETMSG	245Ø	SETSTR	247A		
SH2	24EØ	SH3	24F8	SH4	25Ø5		
SH5	25ØC	SHOW	24DE	SKP2	2519		
SKSP	2517	WPARM	24CØ	WRESP	2559		
@@ABORT	8733	@@ADTSK	87C6	@@BANK	8CDE		
@@BKSP	89BE	@@BREAK	8CF4	@@CHNIO	871E		
@@CKBRKC	8D42	@@CKDRV	881A	@@CKEOF	89D3		
@@CKTSK	87B1	@@CLOSE	89A9	@@CLS	8D2C		
@@CMNDI	875D	@@CMNDR	8772	@@CTL	8582		
@@DATE	86F4	@@DCSTAT	8859	@@DEBUG	879C		
@@DECHEX	8C5E	@@DIRRD	8BCB	@@DIRWR	8BEØ		
@@DIV16	8C49	@@DIV8	8C34	@@DODIR	882F		
@@DSP	8546	@@DSPLY	85E6	@@ERROR	8787		
@@EXIT	8748	@@FEXT	8B38	@@FLAGS	8CC8		
@@FNAME	8B4D	@@FSPEC	8B23	@@GATRD	8BB6		
@@GATWR	8BF5	@@GET	855A	@@GTDCB	8B77		
@@GTDCT	8B62	@@GTMOD	8B8C	@@HDFMT	89Ø1		
@@HEX16	8C9D	@@HEX8	8C88	@@HEXDEC	8C73		
@@HIGH$	8CB2	@@INIT	897F	@@KBD	85BE		
@@KEY	8532	@@KEYIN	85D2	@@KLTSK	88Ø5		
@@LOAD	8AF9	@@LOC	89E8	@@LOF	89FD		
@@LOGER	861D	@@LOGOT	8632	@@MSG	8669		
@@MUL16	8C1F	@@MUL8	8CØA	@@OPEN	8994		
@@PARAM	86DF	@@PAUSE	86CA	@@PEOF	8A12		
@@POSN	8A27	@@PRINT	867E	@@PRT	8596		
@@PUT	856E	@@RAMDIR	8844	@@RDSEC	88D7		
@@RDSSC	8BA1	@@READ	8A3C	@@REMOV	896A		
@@RENAM	8955	@@REW	8A51	@@RMTSK	87DB		
@@RPTSK	87FØ	@@RREAD	8A66	@@RSLCT	88C2		
@@RSTOR	8883	@@RUN	8BØE	@@RWRIT	8A7B		
@@SEEK	88AD	@@SEEKSC	8A9Ø	@@SKIP	8AA5		
@@SLCT	886E	@@STEPI	8898	@@TIME	87Ø9		
@@VDCTL	86B5	@@VER	8ABA	@@VRSEC	88EC		
@@WEOF	8ACF	@@WHERE	85AA	@@WRITE	8AE4		
@@WRSEC	8916	@@WRSSC	892B	@@WRTRK	894Ø		

24ØØ is the transfer address
ØØØØØ Total errors

NOTES:

NOTES:

Command: SPOOL

Library: SYS8/SYS

ISAM # : A2H

```
                00100 ;LBSPOOL/ASM - Spool command
0000            00110         TITLE   <SPOOL - LS-DOS 6.2>
                00120 ;
0000            00130 *GET    LBSPOOLA:3
                00010 ;LBSPOOLA/ASM - SPOOL setup
                00020 ;
002C            00030 PAR_ERR EQU     44              ;Parameter Error
0003            00040 ETX     EQU     3
000D            00050 CR      EQU     13
0028            00060 RST28   EQU     28H
2300            00070 LOWBUF$ EQU     2300H           ;Low memory disk I/O buff
001F            00080 KITSK@  EQU     31              ;FLGTAB+31
                00090 ;
0000            00100 *GET    SVCMAC:3                ;SVC Macro equivalents
                00110 ;SVCMAC/ASM - LS-DOS Version VI
                00120 *LIST   OFF
                04000 *LIST   ON
                04020 *LIST   OFF                     ;Get LDOS60/EQU
                05330 *LIST   ON
                05340 ;
2400            05350         ORG     2400H
                05360 ;
                05370 ;       Save stack & call Spool routine
                05380 ;
2400 ED730B24   05390 SPOOL   LD      (SAVESP+1),SP   ;Save SP
2404 CD5924     05400         CALL    SPOOL1          ;Call SPOOL Code
2407 210000     05410 EXIT    LD      HL,0            ;Successful exit
                05420 ;
                05430 ;       P/u stack & clear any pending <BREAK>
                05440 ;
240A 310000     05450 SAVESP  LD      SP,$-$          ;P/u original stack
240D            05460         @@CKBRKC                ;Clear any <BREAK>
240D 3E6A       00001         LD      A,106
240F EF         00002         RST     40
2410 C9         05470         RET                     ;Ret to DOS
                05480 ;
                05490 ;       Informative Message Exit
                05500 ;
2411 217D29     05510 CLEARD  LD      HL,CLEARD$
2414 DD         05520         DB      0DDH
2415 21A829     05530 RESUMD  LD      HL,RESUMD$
2418 DD         05540         DB      0DDH
2419 219229     05550 PAUSED  LD      HL,PAUSED$
241C DD         05560         DB      0DDH
241D 213A2A     05570 SPLACT  LD      HL,SPLACT$
2420 DD         05580         DB      0DDH
2421 21D529     05590 SPLONM  LD      HL,SPLONM$
2424 DD         05600         DB      0DDH
2425 21BF29     05610 SPLOFM  LD      HL,SPLOFM$
2428 DD         05620         DB      0DDH
2429 21EF29     05630 SPLOPM  LD      HL,SPLOPM$
242C DD         05640         DB      0DDH
242D 21112A     05650 PRMBGM  LD      HL,PRMBGM$
2430            05660         @@LOGOT
                00003         IFEQ    00H,1
                00004         LD      HL,
                00005         ENDIF
2430 3E0C       00006         LD      A,12
2432 EF         00007         RST     40
2433 18D2       05670         JR      EXIT
                05680 ;
```

```
                    05690 ;       I/O Error Processor
                    05700 ;
2435 3E2C           05710 PRMERR  LD      A,PAR_ERR       ;Parameter Error
2437 6F             05720 IOERR   LD      L,A             ;Set HL = Error #
2438 2600           05730         LD      H,0
243A F6C0           05740         OR      0C0H            ;Set short, return
243C 4F             05750         LD      C,A             ;Error code to C
243D                05760         @@ERROR                 ;Dsply error
243D 3E1A           00008         LD      A,26
243F EF             00009         RST     40
2440 18C8           05770         JR      SAVESP          ;Exit
                    05780 ;
                    05790 ;       Internal Error Message Handler
                    05800 ;
2442 21C72A         05810 BADDCB  LD      HL,BADDCB$
2445 DD             05820         DB      0DDH
2446 216029         05830 NOFIND  LD      HL,NOFIND$
2449 DD             05840         DB      0DDH
244A 21972A         05850 INUSE   LD      HL,INUSE$
244D DD             05860         DB      0DDH
244E 21AD2A         05870 CANT    LD      HL,CANT$
2451                05880         @@LOGOT
                    00010         IFEQ    00H,1
                    00011         LD      HL,
                    00012         ENDIF
2451 3E0C           00013         LD      A,12
2453 EF             00014         RST     40
2454 21FFFF         05890 ERREXIT LD      HL,-1           ;Set abort code
2457 18B1           05900         JR      SAVESP          ;Exit
                    05910 ;
                    05920 ;       SPOOL1 - Set up system to Spool Output
                    05930 ;
                    05940 SPOOL1
2459 E5             05950         PUSH    HL              ;Save INBUF$ pointer
245A                05960         @@FLAGS                 ;Get flag table pointer
245A 3E65           00015         LD      A,101
245C EF             00016         RST     40
245D DD211D2D       05970         LD      IX,PRMSAV       ;Point to parm save area
2461 11142D         05980         LD      DE,SPOOL$       ;If already resident,
2464                05990         @@GTMOD                 ;  don't need to check
2464 3E53           00017         LD      A,83
2466 EF             00018         RST     40
2467 2032           06000         JR      NZ,SPL1         ;  if memory available
2469 ED53A428       06010         LD      (SVDCB),DE      ;Save DCB pointer
246D 22B228         06020         LD      (SVBGN),HL      ;Save module begin
2470 210400         06030         LD      HL,4            ;Get parm save pointer
2473 19             06040         ADD     HL,DE
2474 22DD28         06050         LD      (HIPARM),HL     ;Save for use by clear
2477 DDE5           06060         PUSH    IX              ;Also put address in DE
2479 D1             06070         POP     DE              ;Shift saved parms into
247A 010B00         06080         LD      BC,11           ;  spool driver front end
247D EDB0           06090         LDIR
247F DD5E04         06100         LD      E,(IX+4)        ;P/u previous MPARM
2482 DD5605         06110         LD      D,(IX+5)
2485 ED538D26       06120         LD      (MPARM+1),DE    ;Stuff as new default
2489 DD5E06         06130         LD      E,(IX+6)        ;P/u previous DPARM
248C DD5607         06140         LD      D,(IX+7)
248F ED531B26       06150         LD      (DPARM+1),DE    ;Stuff as new default
2493 DD7E0A         06160         LD      A,(IX+10)       ;P/u old bank #
2496 E67F           06170         AND     07FH            ;Strip hi bit
2498 32B625         06180         LD      (BPARM+1),A     ;  & save as default
```

```
249B E1        06190 SPL1    POP     HL              ;Get INBUF$ pointer
               06200 ;
               06210 ;       Fetch device spec to spool
               06220 ;
249C 11E22A    06230 SPL2    LD      DE,SPLDCB       ;Fetch device spec
249F           06240         @@FSPEC                 ;  to spool
249F 3E4E      00019         LD      A,78
24A1 EF        00020         RST     40
24A2 E5        06250         PUSH    HL              ;Save inbuf$ ptr
24A3 201E      06260         JR      NZ,USEPR        ;Use *PR if no entry
24A5 1A        06270         LD      A,(DE)          ;If a device spec not
24A6 FE2A      06280         CP      '*'             ;  found, maybe it was
24A8 2824      06290         JR      Z,SPL2A         ;  the spool file spec
               06300 ;
               06310 ;       File spec fetched - Move to spool FCB
               06320 ;
24AA EB        06330         EX      DE,HL           ;DCB to HL
24AB 11332F    06340         LD      DE,SPLFCB       ;Point to spool FCB
24AE D5        06350         PUSH    DE
24AF 012000    06360         LD      BC,32
24B2 EDB0      06370         LDIR                    ;Move it in
24B4 21092B    06380         LD      HL,PR$          ;Move in default spec
24B7 11E22A    06390         LD      DE,SPLDCB       ;  for spool device
24BA 010400    06400         LD      BC,4
24BD EDB0      06410         LDIR
24BF AF        06420         XOR     A
24C0 D1        06430         POP     DE              ;Point to SPLFCB
24C1 181B      06440         JR      SPL2B           ;Go - we have spec
24C3 21092B    06450 USEPR   LD      HL,PR$          ;Move in default spec
24C6 11E22A    06460         LD      DE,SPLDCB
24C9 010400    06470         LD      BC,4
24CC EDB0      06480         LDIR
               06490 ;
               06500 ;       Fetch file spec to use as disk spool buffer
               06510 ;
24CE E1        06520 SPL2A   POP     HL              ;Get inbuf$ ptr
24CF ED5BE32A  06530         LD      DE,(SPLDCB+1)   ;Stuff device name for
24D3 ED53022B  06540         LD      (DSKSPC),DE     ;  default spool file name
24D7 11332F    06550         LD      DE,SPLFCB       ;Fetch spool filespec
24DA           06560         @@FSPEC
24DA 3E4E      00021         LD      A,78
24DC EF        00022         RST     40
24DD E5        06570         PUSH    HL              ;Save separator char
24DE F5        06580 SPL2B   PUSH    AF              ;  & line pointer
24DF 21022B    06590         LD      HL,DSKSPC       ;Default to XX/SPL
24E2 2803      06600         JR      Z,SPL2C
24E4           06610         @@FSPEC                 ;  if user entered none
24E4 3E4E      00023         LD      A,78
24E6 EF        00024         RST     40
24E7 21052B    06620 SPL2C   LD      HL,SPLEXT       ;Default ext to SPL
24EA           06630         @@FEXT
24EA 3E4F      00025         LD      A,79
24EC EF        00026         RST     40
24ED F1        06640         POP     AF              ;Rcvr line pointer
24EE E1        06650         POP     HL
24EF FE3A      06660         CP      ':'             ;Drive entered?
24F1 201C      06670         JR      NZ,GETPRM
24F3 1A        06680 SPL3    LD      A,(DE)          ;Wait for ETX or EOL
24F4 FE0D      06690         CP      CR
24F6 2807      06700         JR      Z,SPL4
24F8 FE03      06710         CP      3
```

```
24FA 28Ø3      Ø672Ø          JR      Z,SPL4
24FC 13        Ø673Ø          INC     DE
24FD 18F4      Ø674Ø          JR      SPL3
24FF 3E3A      Ø675Ø SPL4     LD      A,':'           ;Stuff colon for drive
25Ø1 12        Ø676Ø          LD      (DE),A
25Ø2 13        Ø677Ø          INC     DE
25Ø3 7E        Ø678Ø          LD      A,(HL)          ;P/u possible drive #
25Ø4 FEØD      Ø679Ø          CP      CR
25Ø6 CA3524    Ø68ØØ          JP      Z,PRMERR        ;Parm error if EOL
25Ø9 23        Ø681Ø          INC     HL
25ØA 12        Ø682Ø          LD      (DE),A          ;Stuff drive #
25ØB 13        Ø683Ø          INC     DE
25ØC 3EØ3      Ø684Ø          LD      A,3
25ØE 12        Ø685Ø          LD      (DE),A          ;Stuff ETX
               Ø686Ø ;
               Ø687Ø ;        Get parameter entries
               Ø688Ø ;
25ØF 11ØD2B    Ø689Ø GETPRM   LD      DE,PRMTBL$      ;Get parms
2512           Ø69ØØ          @@PARAM
2512 3E11      ØØØ27          LD      A,17
2514 EF        ØØØ28          RST     4Ø
2515 C23524    Ø691Ø          JP      NZ,PRMERR       ;Jump on parm error
               Ø692Ø ;
               Ø693Ø ;        Check on OFF
               Ø694Ø ;
2518 11ØØØØ    Ø695Ø OPARM    LD      DE,Ø            ;P/u off parm
251B 1C        Ø696Ø          INC     E               ;On or off?
251C CA3128    Ø697Ø          JP      Z,SPLOFF        ;Jump if OFF
               Ø698Ø ;
               Ø699Ø ;        Check on RESUME despooling
               Ø7ØØØ ;
251F 11ØØØØ    Ø7Ø1Ø RPARM    LD      DE,Ø            ;P/u Resume parm
2522 1C        Ø7Ø2Ø          INC     E               ;Was it used?
2523 CA1C28    Ø7Ø3Ø          JP      Z,RESUME        ;Go if so
               Ø7Ø4Ø ;
               Ø7Ø5Ø ;        Check on PAUSE
               Ø7Ø6Ø ;
2526 11ØØØØ    Ø7Ø7Ø PPARM    LD      DE,Ø            ;P/u Pause parm
2529 1C        Ø7Ø8Ø          INC     E               ;Was it used?
252A CAØ728    Ø7Ø9Ø          JP      Z,PAUSE         ;Go if so
               Ø71ØØ ;
               Ø711Ø ;        Check on CLEAR
               Ø712Ø ;
252D 11ØØØØ    Ø713Ø CPARM    LD      DE,Ø            ;P/u Clear parm
253Ø 1C        Ø714Ø          INC     E               ;Was is used?
2531 CAD428    Ø715Ø          JP      Z,CLEAR         ;Go if so
               Ø716Ø ;
               Ø717Ø ;        Assume request to be to install
               Ø718Ø ;
2534 FDCBØ246  Ø719Ø          BIT     Ø,(IY+'C'-'A')  ;Can't insert spool if
2538 C24E24    Ø72ØØ          JP      NZ,CANT         ;  memory frozen
253B FDCBØ346  Ø721Ø          BIT     Ø,(IY+'D'-'A')  ;Is spooler already on?
253F C22124    Ø722Ø          JP      NZ,SPLONM       ;Quit if so
               Ø723Ø ;
               Ø724Ø ;        If module resident, ck parm sizes
               Ø725Ø ;
2542 11142D    Ø726Ø          LD      DE,SPOOL$
2545           Ø727Ø          @@GTMOD
2545 3E53      ØØØ29          LD      A,83
2547 EF        ØØØ3Ø          RST     4Ø
2548 2Ø2B      Ø728Ø          JR      NZ,FNDDCB       ;Go if no area in mem
```

```
254A DD6EØ4   Ø729Ø          LD      L,(IX+4)        ;Prev. MPARM
254D DD66Ø5   Ø73ØØ          LD      H,(IX+5)
255Ø ED5B8D26 Ø731Ø          LD      DE,(MPARM+1)
2554 B7       Ø732Ø          OR      A
2555 ED52     Ø733Ø          SBC     HL,DE           ;Old must = new
2557 C22D24   Ø734Ø          JP      NZ,PRMBGM       ;Abort if bad
255A DD6EØ6   Ø735Ø          LD      L,(IX+6)        ;D parm
255D DD66Ø7   Ø736Ø          LD      H,(IX+7)
256Ø ED5B1B26 Ø737Ø          LD      DE,(DPARM+1)
2564 ED52     Ø738Ø          SBC     HL,DE
2566 C22D24   Ø739Ø          JP      NZ,PRMBGM       ;Abort if not same as old
2569 DD7EØA   Ø74ØØ          LD      A,(IX+1Ø)       ;Bank #
256C E67F     Ø741Ø          AND     Ø7FH            ;Strip hi bit
256E 2AB625   Ø742Ø          LD      HL,(BPARM+1)
2571 BD       Ø743Ø          CP      L
2572 C22D24   Ø744Ø          JP      NZ,PRMBGM       ;Abort if not same as old
              Ø745Ø ;
              Ø746Ø ;        Find device to spool
              Ø747Ø ;
2575 ED5BE32A Ø748Ø FNDDCB   LD      DE,(SPLDCB+1)   ;P/u device name
2579          Ø749Ø          @@GTDCB                 ;Find in device tables
2579 3E52     ØØØ31          LD      A,82
257B EF       ØØØ32          RST     4Ø
257C C23724   Ø75ØØ          JP      NZ,IOERR        ;Jump on not found
257F CB66     Ø751Ø CKRTE    BIT     4,(HL)          ;Routed device?
2581 28Ø7     Ø752Ø          JR      Z,CKLNK         ;Bypass if not
2583 2C       Ø753Ø CKRTE1   INC     L               ;Pt to vector
2584 7E       Ø754Ø          LD      A,(HL)          ;P/u vector DCB address
2585 2C       Ø755Ø          INC     L               ;  & test it
2586 66       Ø756Ø          LD      H,(HL)
2587 6F       Ø757Ø          LD      L,A
2588 18F5     Ø758Ø          JR      CKRTE
258A CB6E     Ø759Ø CKLNK    BIT     5,(HL)          ;If linked, get DCB
258C 2ØF5     Ø76ØØ          JR      NZ,CKRTE1
258E CB76     Ø761Ø          BIT     6,(HL)          ;If filtered, trace to
2590 281B     Ø762Ø          JR      Z,SETDVR        ;  its DCB in header
2592 54       Ø763Ø          LD      D,H             ;Save to ck inactive
2593 5D       Ø764Ø          LD      E,L
2594 2C       Ø765Ø          INC     L
2595 7E       Ø766Ø          LD      A,(HL)          ;Get vector to module
2596 2C       Ø767Ø          INC     L
2597 66       Ø768Ø          LD      H,(HL)
2598 6F       Ø769Ø          LD      L,A
2599 Ø1Ø4ØØ   Ø77ØØ          LD      BC,4            ;HL now points to the
259C Ø9       Ø771Ø          ADD     HL,BC           ;  entry point. Get its
259D 4E       Ø772Ø          LD      C,(HL)          ;  DCB address by peeking
259E ØC       Ø773Ø          INC     C               ;  past the name field
259F Ø9       Ø774Ø          ADD     HL,BC
25AØ 7E       Ø775Ø          LD      A,(HL)          ;Get low-order
25A1 23       Ø776Ø          INC     HL
25A2 66       Ø777Ø          LD      H,(HL)          ;Get hi-order
25A3 6F       Ø778Ø          LD      L,A
25A4 E5       Ø779Ø          PUSH    HL              ;If DCB is itself, then
25A5 ED52     Ø78ØØ          SBC     HL,DE           ;  it's some error
25A7 E1       Ø781Ø          POP     HL              ;  else continue to
25A8 2ØD5     Ø782Ø          JR      NZ,CKRTE        ;  search the chain
25AA C34224   Ø783Ø          JP      BADDCB
              Ø784Ø ;
              Ø785Ø ;        Found the device
              Ø786Ø ;
25AD CB4E     Ø787Ø SETDVR   BIT     1,(HL)          ;Device handle @PUT?
```

```
25AF CA4224    Ø788Ø          JP      Z,BADDCB        ;Can't do if not
25B2 22252D    Ø789Ø          LD      (MODDVR),HL     ;Stuff DCB table address
               Ø79ØØ ;
               Ø791Ø ;        Check BANK spec for availability
               Ø792Ø ;
25B5 Ø1ØØØØ    Ø793Ø BPARM    LD      BC,Ø            ;Pick up the bank #
25B8 79        Ø794Ø          LD      A,C             ;If bank=Ø, no need
25B9 BØ        Ø795Ø          OR      B               ;  to test for
25BA 281B      Ø796Ø          JR      Z,LOCDCB        ;  availability
25BC FEØ8      Ø797Ø          CP      8               ;This version supports
25BE D23524    Ø798Ø          JP      NC,PRMERR       ;  banks Ø-7
25C1 CBFF      Ø799Ø          SET     7,A             ;Set for transfer
25C3 32612D    Ø8ØØØ          LD      (BANKX1),A      ;Stuff for module A
25C6 32832D    Ø8Ø1Ø          LD      (BANKX2),A
25C9 329Ø2D    Ø8Ø2Ø          LD      (BANKX3),A
25CC 32272D    Ø8Ø3Ø          LD      (SPLBNK),A
25CF Ø6Ø2      Ø8Ø4Ø          LD      B,2             ;Test if bank 1-x
25D1           Ø8Ø5Ø          @@BANK                  ;  is available
25D1 3E66      ØØØ33          LD      A,1Ø2
25D3 EF        ØØØ34          RST     4Ø
25D4 C24A24    Ø8Ø6Ø          JP      NZ,INUSE        ;Quit if not available
               Ø8Ø7Ø ;
               Ø8Ø8Ø ;        Locate a spare DCB for the spool module
               Ø8Ø9Ø ;
25D7 11ØØØØ    Ø81ØØ LOCDCB   LD      DE,Ø
25DA           Ø811Ø          @@GTDCB
25DA 3E52      ØØØ35          LD      A,82
25DC EF        ØØØ36          RST     4Ø
25DD 3E21      Ø812Ø          LD      A,33            ;Init "No device space...
25DF C23724    Ø813Ø          JP      NZ,IOERR        ;Go if no spare DCBs
25E2 22192D    Ø814Ø          LD      (SØDCB),HL      ;Save pointer
25E5 22B62D    Ø815Ø          LD      (SØDCB1),HL
25E8 E5        Ø816Ø          PUSH    HL              ;Save DCB field pointer
25E9 11532F    Ø817Ø          LD      DE,'/S'         ;Let's find a link name
25EC 14        Ø818Ø NAMLP    INC     D               ;Bump "2nd" character
25ED           Ø819Ø          @@GTDCB                 ;If we find this name
25ED 3E52      ØØØ37          LD      A,82
25EF EF        ØØØ38          RST     4Ø
25FØ 28FA      Ø82ØØ          JR      Z,NAMLP         ;  look for another
25F2 E1        Ø821Ø          POP     HL              ;Get DCB pointer
25F3 CBDE      Ø822Ø          SET     3,(HL)          ;NIL in case of error
25F5 Ø1Ø6ØØ    Ø823Ø          LD      BC,6
25F8 Ø9        Ø824Ø          ADD     HL,BC           ;Point to name field
25F9 73        Ø825Ø          LD      (HL),E          ;  & stuff in the
25FA 2C        Ø826Ø          INC     L               ;  selected spool name
25FB 72        Ø827Ø          LD      (HL),D
               Ø828Ø ;
               Ø829Ø ;        Create a disk file for spooling
               Ø83ØØ ;
25FC ED5B1B26  Ø831Ø          LD      DE,(DPARM+1)    ;P/u disk size
26ØØ DD73Ø6    Ø832Ø          LD      (IX+6),E        ;Save DPARM for next time
26Ø3 DD72Ø7    Ø833Ø          LD      (IX+7),D        ;  SPOOL is entered
26Ø6 7A        Ø834Ø          LD      A,D
26Ø7 B3        Ø835Ø          OR      E
26Ø8 284E      Ø836Ø          JR      Z,PUHIGH        ;No file if DISK=Ø
26ØA 11332F    Ø837Ø          LD      DE,SPLFCB       ;Init the spool file
26ØD 21ØØ2C    Ø838Ø          LD      HL,BUFFER
261Ø FDCB12C6  Ø839Ø          SET     Ø,(IY+'S'-'A')  ;Init file open bit
2614           Ø84ØØ          @@INIT
2614 3E3A      ØØØ39          LD      A,58
2616 EF        ØØØ4Ø          RST     4Ø
```

```
2617 C23724    Ø841Ø         JP      NZ,IOERR          ;Jump on init error
261A Ø1Ø4ØØ    Ø842Ø DPARM   LD      BC,4              ;D parm (default=4K)
261D 78        Ø843Ø         LD      A,B               ;Parm error if > 4Ø95K
261E E6FØ      Ø844Ø         AND     ØFØH
2620 C23524    Ø845Ø PRMERRA JP      NZ,PRMERR         ;Go if too big a file
2623 CB21      Ø846Ø         SLA     C                 ;Multiply K by 4
2625 CB1Ø      Ø847Ø         RL      B                 ;  to calculate sectors
2627 CB21      Ø848Ø         SLA     C
2629 CB1Ø      Ø849Ø         RL      B
262B ED43FF26  Ø85ØØ         LD      (IPLDSK1+1),BC    ;Stuff disk K
262F ØB        Ø851Ø         DEC     BC                ;Adjust for Ø offset
2630           Ø852Ø         @@POSN                    ;Position to end
2630 3E42      ØØØ41         LD      A,66
2632 EF        ØØØ42         RST     4Ø
2633           Ø853Ø         @@WRITE                   ;Write a junk sector
2633 3E4B      ØØØ43         LD      A,75
2635 EF        ØØØ44         RST     4Ø
2636 C23724    Ø854Ø         JP      NZ,IOERR          ;Jump on write error
2639           Ø855Ø         @@REW                     ;Restore file to Ø
2639 3E44      ØØØ45         LD      A,68
263B EF        ØØØ46         RST     4Ø
263C           Ø856Ø         @@WEOF                    ;End of file mark
263C 3E4A      ØØØ47         LD      A,74
263E EF        ØØØ48         RST     4Ø
263F ED4B392F  Ø857Ø         LD      BC,(SPLFCB+6)     ;P/u DEC & drive
2643           Ø858Ø         @@DIRRD                   ;Read its dir record
2643 3E57      ØØØ49         LD      A,87
2645 EF        ØØØ5Ø         RST     4Ø
2646 C23724    Ø859Ø         JP      NZ,IOERR          ;Quit on read error
2649 23        Ø86ØØ         INC     HL                ;Point to DIR+1
264A CBFE      Ø861Ø         SET     7,(HL)            ;Turn on CREATE bit
264C           Ø862Ø         @@DIRWR                   ;Write dir back
264C 3E58      ØØØ51         LD      A,88
264E EF        ØØØ52         RST     4Ø
264F C23724    Ø863Ø         JP      NZ,IOERR          ;Jump on error
2652 21ØØ23    Ø864Ø         LD      HL,LOWBUF$        ;Revise I/O buffer
2655 22362F    Ø865Ø         LD      (SPLFCB+3),HL     ;  in file's FCB
               Ø866Ø ;
               Ø867Ø ;       Get current HIGH$
               Ø868Ø ;
2658 21ØØØØ   Ø869Ø PUHIGH   LD      HL,Ø              ;Set HLB to zero to
265B 45        Ø87ØØ         LD      B,L               ;  fetch HIGH$
265C           Ø871Ø         @@HIGH$
265C 3E64      ØØØ53         LD      A,1ØØ
265E EF        ØØØ54         RST     4Ø
265F 22BC26    Ø872Ø         LD      (GBUF1+1),HL      ;Save for later
2662 22BD27    Ø873Ø         LD      (OLDHI+1),HL
               Ø874Ø ;
               Ø875Ø ;       If bank RAM, recalculate MPARM
               Ø876Ø ;
2665 3AB625    Ø877Ø         LD      A,(BPARM+1)       ;Alternate banks requested?
2668 B7        Ø878Ø         OR      A
2669 2821      Ø879Ø         JR      Z,MPARM           ;Go if not
               Ø88ØØ ;
               Ø881Ø ;       Recalculate MEM parameter to use the
               Ø882Ø ;       Maximum space in bank RAM. Formula is
               Ø883Ø ;       [((32768 - 512 for module) - ((DPARM+1)*16)]
               Ø884Ø ;       Divided by 26Ø. The 26Ø is derived from
               Ø885Ø ;       256-byte page plus 4-bytes for pointers.
               Ø886Ø ;       Then revise to K from pages.
               Ø887Ø ;
```

```
266B 2A1B26   Ø888Ø          LD      HL,(DPARM+1)
266E 23       Ø889Ø          INC     HL              ;Bump by 1
266F 29       Ø89ØØ          ADD     HL,HL           ;Times 16
267Ø 29       Ø891Ø          ADD     HL,HL
2671 29       Ø892Ø          ADD     HL,HL
2672 29       Ø893Ø          ADD     HL,HL
2673 4D       Ø894Ø          LD      C,L             ;Xfer to BC for subtract
2674 44       Ø895Ø          LD      B,H
2675 21ØØ7E   Ø896Ø          LD      HL,32768-512    ;RAM space - module space
2678 AF       Ø897Ø          XOR     A               ;The remainder is for
2679 ED42     Ø898Ø          SBC     HL,BC           ;  MEM pages and ptrs
267B CB3C     Ø899Ø          SRL     H               ;A divide by 26Ø is
267D CB1D     Ø9ØØØ          RR      L               ;  accomplished by
267F ØE82     Ø9Ø1Ø          LD      C,26Ø/2         ;  div 2 and div 13Ø
2681          Ø9Ø2Ø          @@DIV16
2681 3E5E     ØØØ55          LD      A,94
2683 EF       ØØØ56          RST     4Ø
2684 4D       Ø9Ø3Ø          LD      C,L
2685 44       Ø9Ø4Ø          LD      B,H
2686 CB39     Ø9Ø5Ø          SRL     C               ;Calculate
2688 CB39     Ø9Ø6Ø          SRL     C               ;  K = pages / 4
268A 1815     Ø9Ø7Ø          JR      CALCPTR
              Ø9Ø8Ø ;
              Ø9Ø9Ø ;        Calculate memory buffer space requested
              Ø91ØØ ;
268C Ø1Ø1ØØ   Ø911Ø MPARM    LD      BC,1            ;MEM parm - default 1K
268F 78       Ø912Ø          LD      A,B             ;Test entry - must be
269Ø B7       Ø913Ø          OR      A               ;< 33K
2691 2Ø8D     Ø914Ø          JR      NZ,PRMERRA      ;Hi-order must = Ø
2693 B9       Ø915Ø          CP      C               ;MEM = Ø ?
2694 CA3524   Ø916Ø          JP      Z,PRMERR        ;Yes - Parameter Error
2697 7C       Ø917Ø          LD      A,H             ;Get hi-order HIGH$
2698 D681     Ø918Ø          SUB     81H             ;Can't go below 8ØØØH
269A 1F       Ø919Ø          RRA                     ;Divide by 2
269B CB3F     Ø92ØØ          SRL     A               ;  & again to calc K
269D 91       Ø921Ø          SUB     C               ;Reduce by K of req
269E DA3524   Ø922Ø          JP      C,PRMERR        ;Error if goes < 8ØØØH
26A1 DD71Ø4   Ø923Ø CALCPTR  LD      (IX+4),C        ;Stuff MEM parm into
26A4 DD7ØØ5   Ø924Ø          LD      (IX+5),B        ;  save area
              Ø925Ø ;
              Ø926Ø ;        Add memory buffer space to disk space
              Ø927Ø ;
26A7 2A1B26   Ø928Ø          LD      HL,(DPARM+1)    ;P/u disk K
26AA Ø9       Ø929Ø          ADD     HL,BC           ;Calculate pointer
26AB 29       Ø93ØØ          ADD     HL,HL           ;  buffer space required
26AC 29       Ø931Ø          ADD     HL,HL           ;  (D+M)*16 + 4 extra
26AD 23       Ø932Ø          INC     HL              ;  used to detect the end
26AE 29       Ø933Ø          ADD     HL,HL
26AF 29       Ø934Ø          ADD     HL,HL
              Ø935Ø ;
              Ø936Ø ;        Retrieve high memory space for buffers
              Ø937Ø ;
26BØ E5       Ø938Ø          PUSH    HL
26B1 DD6EØØ   Ø939Ø          LD      L,(IX+Ø)        ;P/u saved HIGH$ entry
26B4 DD66Ø1   Ø94ØØ          LD      H,(IX+1)
26B7 7D       Ø941Ø          LD      A,L             ;If <> Ø, then SPOOL
26B8 B4       Ø942Ø          OR      H               ;  is resident
26B9 2ØØ9     Ø943Ø          JR      NZ,GBUF2
26BB 21ØØØØ   Ø944Ø GBUF1    LD      HL,$-$          ;P/u current high
26BE DD75ØØ   Ø945Ø          LD      (IX+Ø),L        ;Save the value
26C1 DD74Ø1   Ø946Ø          LD      (IX+1),H
```

```
26C4 223629   09470 GBUF2  LD      (SVEND+1),HL    ;Stuff highest byte used
              09480 ;
              09490 ;       If bank RAM used, reset to X'FFFF'
              09500 ;
26C7 23       09510         INC     HL              ;Pt to lower 256
26C8 2E00     09520         LD      L,0             ;  byte block boundary
26CA 3AB625   09530         LD      A,(BPARM+1)     ;Bank specified?
26CD B7       09540         OR      A
26CE 2801     09550         JR      Z,$+3           ;Jump if not
26D0 65       09560         LD      H,L             ;Set to FFFF+1 = 0
26D1 4F       09570         LD      C,A             ;Reside bank for
26D2 0600     09580         LD      B,0             ;  module B
26D4          09590         @@BANK
26D4 3E66     00057         LD      A,102
26D6 EF       00058         RST     40
26D7 C23724   09600         JP      NZ,IOERR        ;Quit on bank error
26DA DD7E04   09610         LD      A,(IX+4)        ;P/u MEM parm
26DD 07       09620         RLCA                    ;MEM * 4 = # of
26DE 07       09630         RLCA                    ;  256-byte blocks
26DF 47       09640         LD      B,A             ;Go back that number
26E0 ED44     09650         NEG                     ;  of blocks needed
26E2 84       09660         ADD     A,H             ;HL now reserves buffer
26E3 67       09670         LD      H,A             ;  blocks for MEM
26E4 22532F   09680         LD      (PUTNXT$),HL    ;Stuff pointers to begin
26E7 22552F   09690         LD      (GETNXT$),HL    ;  of memory blocks
26EA 4C       09700         LD      C,H
26EB D1       09710         POP     DE              ;Get (D+M)*16+4
26EC AF       09720         XOR     A               ;Backup the # of D+M
26ED ED52     09730         SBC     HL,DE           ;  sectors
              09740 ;
              09750 ;       Initialize the memory pointers
              09760 ;
26EF E5       09770         PUSH    HL              ;Save pointer
26F0 3600     09780 IPLMEM  LD      (HL),0          ;Init the 4-byte fields
26F2 23       09790         INC     HL              ;  for each of the disk &
26F3 3600     09800         LD      (HL),0          ;  memory block pointers
26F5 23       09810         INC     HL
26F6 71       09820         LD      (HL),C          ;Stuff memory hi-order
26F7 23       09830         INC     HL
26F8 3600     09840         LD      (HL),0
26FA 23       09850         INC     HL
26FB 0C       09860         INC     C               ;Bump memory hi-order
26FC 10F2     09870         DJNZ    IPLMEM
              09880 ;
              09890 ;       Initialize the disk pointers
              09900 ;
26FE 110000   09910 IPLDSK1 LD      DE,0            ;P/u # of disk sectors
2701 7A       09920         LD      A,D
2702 B3       09930         OR      E
2703 2815     09940         JR      Z,IPLDSK3       ;Bypass if none
2705 010000   09950         LD      BC,0            ;Set up a disk pointer
2708 3600     09960 IPLDSK2 LD      (HL),0          ;  for each disk block
270A 23       09970         INC     HL
270B 3600     09980         LD      (HL),0
270D 23       09990         INC     HL
270E 71       10000         LD      (HL),C          ;Stuff block number
270F 23       10010         INC     HL
2710 70       10020         LD      (HL),B
2711 CBFE     10030         SET     7,(HL)          ;Set bit 7 to indicate
2713 23       10040         INC     HL              ;  this pntr is for disk
2714 03       10050         INC     BC              ;Inc block number
```

```
2715 1B       1ØØ6Ø          DEC     DE
2716 7A       1ØØ7Ø          LD      A,D             ;Test if finished
2717 B3       1ØØ8Ø          OR      E
2718 2ØEE     1ØØ9Ø          JR      NZ,IPLDSK2      ;Loop if not
271A 36ØØ     1Ø1ØØ IPLDSK3  LD      (HL),Ø          ;One more used to
271C 23       1Ø11Ø          INC     HL              ;  indicate the end
271D 36ØØ     1Ø12Ø          LD      (HL),Ø          ;  of the pointer fields
271F 23       1Ø13Ø          INC     HL
2720 36FF     1Ø14Ø          LD      (HL),ØFFH
2722 23       1Ø15Ø          INC     HL
2723 36FF     1Ø16Ø          LD      (HL),ØFFH
2725 E1       1Ø17Ø          POP     HL              ;Rcvr start of pointers
2726 E5       1Ø18Ø          PUSH    HL
2727 22572F   1Ø19Ø          LD      (MAPPTR$),HL    ;  & stuff for later use
272A 22592F   1Ø2ØØ          LD      (PUTBLK$),HL
272D 225B2F   1Ø21Ø          LD      (GETBLK$),HL
2730 7C       1Ø22Ø          LD      A,H             ;Stuff 1st MEM block ptr
2731 75       1Ø23Ø          LD      (HL),L          ;  with table addr
2732 23       1Ø24Ø          INC     HL
2733 77       1Ø25Ø          LD      (HL),A
              1Ø26Ø ;
              1Ø27Ø ;        Modify driver for spooling device vector address
              1Ø28Ø ;
2734 FD7E1F   1Ø29Ø          LD      A,(IY+KITSK@)   ;Transfer device hook
2737 FD6E2Ø   1Ø3ØØ          LD      L,(IY+KITSK@+1)
273A FD6621   1Ø31Ø          LD      H,(IY+KITSK@+2)
273D 32982D   1Ø32Ø          LD      (GETBLKØ),A
2740 22992D   1Ø33Ø          LD      (GETBLKØ+1),HL
2743 FDE5     1Ø34Ø          PUSH    IY              ;Set up address
2745 D1       1Ø35Ø          POP     DE              ;  in despooler
2746 21ØFØØ   1Ø36Ø          LD      HL,'P'-'A'      ;PFLAG use to
2749 19       1Ø37Ø          ADD     HL,DE           ;  determine if spooler
274A 226D2D   1Ø38Ø          LD      (DFLAG),HL      ;  is Paused
274D 226F2E   1Ø39Ø          LD      (DFLAG1),HL
2750 C1       1Ø4ØØ          POP     BC              ;Start of MEM block tbl
2751 ØB       1Ø41Ø          DEC     BC              ;Get last byte to use
2752 C5       1Ø42Ø          PUSH    BC
2753 115E2F   1Ø43Ø          LD      DE,DVRBEND      ;Point to module B
2756 215F2F   1Ø44Ø          LD      HL,RELTABB      ;Point to table
2759 CD3Ø29   1Ø45Ø          CALL    RELO            ;Relocate absolute refs
              1Ø46Ø ;
              1Ø47Ø ;        If bank Ø, then wipe out module A
              1Ø48Ø ;
275C 3AB625   1Ø49Ø          LD      A,(BPARM+1)     ;Alternate bank used?
275F B7       1Ø5ØØ          OR      A
2760 2Ø19     1Ø51Ø          JR      NZ,MOVMODB      ;Skip next if so
2762 21ØØ2D   1Ø52Ø          LD      HL,FIX1         ;Shift in JPs for the
2765 115C2D   1Ø53Ø          LD      DE,PUTCHAR      ;  bank transfer
2768 Ø1Ø5ØØ   1Ø54Ø          LD      BC,5            ;  linkage
276B EDBØ     1Ø55Ø          LDIR
276D 117F2D   1Ø56Ø          LD      DE,FIX2UP
2770 ØE Ø5    1Ø57Ø          LD      C,5
2772 EDBØ     1Ø58Ø          LDIR
2774 118C2D   1Ø59Ø          LD      DE,GETBLK
2777 ØE Ø5    1Ø6ØØ          LD      C,5
2779 EDBØ     1Ø61Ø          LDIR
277B D1       1Ø62Ø MOVMODB  POP     DE              ;Get last byte to use
277C 215E2F   1Ø63Ø          LD      HL,DVRBEND      ;Last byte of module
277F Ø1A9Ø1   1Ø64Ø          LD      BC,DVRBEND-DVRBBGN+1
2782 EDB8     1Ø65Ø          LDDR
2784 D5       1Ø66Ø          PUSH    DE              ;New last byte
```

```
                10670 ;
                10680 ;       Switch to bank Ø always
                10690 ;
2785 Ø10000     10700         LD      BC,Ø            ;Set function Ø, bank Ø
2788            10710         @@BANK
2788 3E66       00059         LD      A,102
278A EF         00060         RST     40
278B C23724     10720         JP      NZ,IOERR        ;Quit on bank error
278E C1         10730         POP     BC
                10740 ;
                10750 ;       If module B is in bank x, switch HIGH for A
                10760 ;
278F 3AB625     10770         LD      A,(BPARM+1)     ;P/u alternate bank resp.
2792 B7         10780         OR      A
2793 280D       10790         JR      Z,STORHI        ;Go if none used
2795 4F         10800         LD      C,A             ;Reserve the bank
2796 0603       10810         LD      B,3             ;Set function # 3
2798            10820         @@BANK
2798 3E66       00061         LD      A,102
279A EF         00062         RST     40
279B C23724     10830         JP      NZ,IOERR        ;Quit on bank error
279E ED4B3629   10840         LD      BC,(SVEND+1)    ;P/u bank-A HIGH$
27A2 C5         10850 STORHI  PUSH    BC
                10860 ;
                10870 ;       Relocate absolute addresses in module A
                10880 ;
27A3 119A2D     10890         LD      DE,DVRAEND      ;Point to last used in A
27A6 219B2D     10900         LD      HL,RELTABA      ;Point to relo table
27A9 CD3029     10910         CALL    RELO            ;Relocate absolute refs
27AC D1         10920         POP     DE              ;Get last byte to use
27AD 219A2D     10930         LD      HL,DVRAEND      ;Last byte of module
27B0 018C00     10940         LD      BC,DVRAEND-DVRABGN+1
27B3 EDB8       10950         LDDR
27B5 211100     10960         LD      HL,PRMSAV+2-DVRABGN+1
27B8 19         10970         ADD     HL,DE           ;Data area in himem
27B9 73         10980         LD      (HL),E
27BA 23         10990         INC     HL
27BB 72         11000         LD      (HL),D          ;Save high$ there
                11010 ;
                11020 ;       Stuff new HIGH$
                11030 ;
27BC 210000     11040 OLDHI   LD      HL,$-$          ;P/u old HIGH$
27BF ED52       11050         SBC     HL,DE           ;Check if old HIGH$ is
27C1 3805       11060         JR      C,NOSET         ;  already lower
27C3 EB         11070         EX      DE,HL           ;New HIGH$ to HL
27C4            11080         @@HIGH$                 ;Stuff new HIGH$
27C4 3E64       00063         LD      A,100
27C6 EF         00064         RST     40
27C7 EB         11090         EX      DE,HL
27C8 13         11100 NOSET   INC     DE              ;Point to module start
27C9 2A192D     11110         LD      HL,(SØDCB)      ;P/u module DCB pointer
27CC 3646       11120         LD      (HL),46H        ;  & set to filter
27CE 23         11130         INC     HL              ;Pt to DCB vector pos
27CF 73         11140         LD      (HL),E          ;Put entry point of
27D0 23         11150         INC     HL              ;  module into DCB
27D1 72         11160         LD      (HL),D
27D2 116A2D     11170         LD      DE,DSPLTCB      ;Calculate entry point
27D3            11180 RX05    EQU     $-2
27D5 0E09       11190         LD      C,9             ;This will be task 9
27D7            11200         @@ADTSK
27D7 3E1D       00065         LD      A,29
```

```
27D9 EF       ØØØ66          RST     4Ø
27DA 218C2D   11210          LD      HL,GETBLK       ;Now update device hook
27DB          11220 RXØ6     EQU     $-2
              11230 ;
              11240 ;        Interface to KITSK - No need to DI/EI
              11250 ;        As KITSK not used by interrupts
              11260 ;
27DD FD7421   11270          LD      (IY+KITSK@+2),H ;Move in device
27EØ FD7520   11280          LD      (IY+KITSK@+1),L ;  vector and a
27E3 3EC3     11290          LD      A,ØC3H          ;  JP opcode
27E5 FD771F   11300          LD      (IY+KITSK@),A
27E8 FDCBØ3C6 11310          SET     Ø,(IY+'D'-'A')  ;Turn on device flag bit
27EC FDCBØFFE 11320          SET     7,(IY+'P'-'A')  ;Show not paused
              11330 ;
              11340 ;        Filter the specified device driver
              11350 ;
27FØ 2A252D   11360          LD      HL,(MODDVR)     ;Ptr to device DCB
27F3 ED5B192D 11370          LD      DE,(SØDCB)      ;Ptr to spooler DCB
27F7 Ø6Ø3     11380          LD      B,3
27F9 F3       11390          DI                      ;Interrupts off during swap
27FA 4E       11400 SWAP     LD      C,(HL)          ;Swap first 3 bytes
27FB 1A       11410          LD      A,(DE)
27FC 77       11420          LD      (HL),A          ;  of the DCBs
27FD 79       11430          LD      A,C
27FE 12       11440          LD      (DE),A
27FF 2C       11450          INC     L               ;Bump DCB pointers
2800 1C       11460          INC     E
2801 10F7     11470          DJNZ    SWAP            ;Loop 3 times
2803 FB       11480          EI
2804 C31D24   11490          JP      SPLACT          ;Set complete
              11500 ;
              11510 ;        Routine to PAUSE despooling
              11520 ;
2807 FDCBØ346 11530 PAUSE    BIT     Ø,(IY+'D'-'A')  ;Spool resident?
280B CA2524   11540          JP      Z,SPLOFM        ;Quit if not
280E FDCBØF7E 11550          BIT     7,(IY+'P'-'A')  ;Ck first if active
2812 FDCBØFBE 11560          RES     7,(IY+'P'-'A')  ;  then turn off
2816 CA2924   11570          JP      Z,SPLOPM        ;Quit if already paused
2819 C31924   11580          JP      PAUSED
              11590 ;
              11600 ;        Routine to RESUME despooling
              11610 ;
281C FDCBØ346 11620 RESUME   BIT     Ø,(IY+'D'-'A')  ;Ck first if resident
2820 CA2524   11630          JP      Z,SPLOFM        ;Quit if not
2823 FDCBØF7E 11640          BIT     7,(IY+'P'-'A')  ;See if despooling now
2827 FDCBØFFE 11650          SET     7,(IY+'P'-'A')
282B C22124   11660          JP      NZ,SPLONM       ;Can't resume if so
282E C31524   11670          JP      RESUMD
              11680 ;
              11690 ;        Routine to turn off the spooler
              11700 ;
2831 FDCBØ346 11710 SPLOFF   BIT     Ø,(IY+'D'-'A')  ;Test if off already
2835 CA2524   11720          JP      Z,SPLOFM        ;Go if off now
2838 11142D   11730          LD      DE,SPOOL$       ;Find out where spooler
283B          11740          @@GTMOD
283B 3E53     ØØØ67          LD      A,83
283D EF       ØØØ68          RST     4Ø
283E C24624   11750          JP      NZ,NOFIND       ;Quit if not there
2841 FDCBØ386 11760          RES     Ø,(IY+'D'-'A')  ;  else turn off
2845 FDCBØFBE 11770          RES     7,(IY+'P'-'A')  ;Show paused also
2849 Ø189ØØ   11780          LD      BC,GETBLKØ-DVRABGN
```

```
284C Ø9       11179Ø          ADD     HL,BC           ;Point to KITSK@ swap
284D 7E       11180Ø          LD      A,(HL)
284E FD771F   11181Ø          LD      (IY+KITSK@),A   ;Restuff KIHOOK
2851 23       11182Ø          INC     HL
2852 7E       11183Ø          LD      A,(HL)
2853 FD7520   11184Ø          LD      (IY+KITSK@+1),L
2856 23       11185Ø          INC     HL
2857 7E       11186Ø          LD      A,(HL)
2858 FD7721   11187Ø          LD      (IY+KITSK@+2),A
285B ØE Ø9    11188Ø          LD      C,9             ;Remove task 9 from
285D          11189Ø          @@RMTSK                 ;  the task processor
285D 3E1E     ØØØ69           LD      A,3Ø
285F EF       ØØØ7Ø           RST     4Ø
              11190Ø ;
              11191Ø ;        Attempt to reclaim memory space
              11192Ø ;
2860 210000   11193Ø          LD      HL,Ø            ;Get current HIGH$
2863 45       11194Ø          LD      B,L
2864          11195Ø          @@HIGH$
2864 3E64     ØØØ71           LD      A,1ØØ
2866 EF       ØØØ72           RST     4Ø
2867 DD5E02   11196Ø          LD      E,(IX+2)        ;Get HIGH$ after spool
286A DD5603   11197Ø          LD      D,(IX+3)        ;  was installed
286D AF       11198Ø          XOR     A
286E ED52     11199Ø          SBC     HL,DE           ;Is it the same now?
2870 DD5E00   12000           LD      E,(IX+Ø)        ;Get HIGH before res
2873 DD7700   12010           LD      (IX+Ø),A        ;  & clear data
2876 DD5601   12020           LD      D,(IX+1)
2879 DD7701   12030           LD      (IX+1),A
287C 21552A   12040           LD      HL,NOMEM$       ;Init "Can't reclaim...
287F 2007     12050           JR      NZ,LOGMEM       ;Go if HIGH$ different
2881 EB       12060           EX      DE,HL           ;  else release memory
2882          12070           @@HIGH$                 ;  by resetting HIGH$
2882 3E64     ØØØ73           LD      A,1ØØ
2884 EF       ØØØ74           RST     4Ø
2885 21702A   12080           LD      HL,GOTMEM$      ;"Memory released
2888          12090 LOGMEM    @@LOGOT
              ØØØ75           IFEQ    ØØH,1
              ØØØ76           LD      HL,
              ØØØ77           ENDIF
2888 3E0C     ØØØ78           LD      A,12
288A EF       ØØØ79           RST     4Ø
              12100 ;
              12110 ;         De-use any RAM bank
              12120 ;
288B DD7E0A   12130           LD      A,(IX+1Ø)       ;P/u bank used
288E E607     12140           AND     7               ;Strip 3-7
2890 2811     12150           JR      Z,RESDCB        ;Jump if only bank Ø
2892 4F       12160           LD      C,A             ;Save bank in C
2893 F630     12170           OR      'Ø'             ;Convert to ASCII for dsply
2895 328C2A   12180           LD      (FREBNK$+5),A   ;Stuff in message
2898 0601     12190           LD      B,1             ;Function to reset BUR$
289A          12200           @@BANK                  ;Free the bank
289A 3E66     ØØØ8Ø           LD      A,1Ø2
289C EF       ØØØ81           RST     4Ø
289D 21872A   12210           LD      HL,FREBNK$      ;Init"Bank released
28A0          12220           @@LOGOT
              ØØØ82           IFEQ    ØØH,1
              ØØØ83           LD      HL,
              ØØØ84           ENDIF
28A0 3E0C     ØØØ85           LD      A,12
```

```
28A2 EF        00086          RST     40
               12230 ;
               12240 ;        RESET the device
               12250 ;
28A3 210000    12260 RESDCB   LD      HL,$-$
28A4           12270 SVDCB    EQU     $-2             ;Point to SPOOL DCB ptr
28A6 5E        12280          LD      E,(HL)          ;  & get the DCB ptr
28A7 23        12290          INC     HL              ;  into reg DE
28A8 56        12300          LD      D,(HL)
               12310 ;
               12320 ;        Check if DCB is inactive [user RESET *PR]
               12330 ;
28A9 D5        12340          PUSH    DE              ;Save DCB pointer
28AA 1C        12350          INC     E
28AB 1A        12360          LD      A,(DE)          ;Get DCB vector contents
28AC 4F        12370          LD      C,A             ;  into reg BC
28AD 1C        12380          INC     E
28AE 1A        12390          LD      A,(DE)
28AF 47        12400          LD      B,A
28B0 D1        12410          POP     DE              ;Recover DCB pointer
28B1 210000    12420          LD      HL,$-$          ;P/u start of module
28B2           12430 SVBGN    EQU     $-2
28B4 AF        12440          XOR     A               ;If vector points to
28B5 ED42      12450          SBC     HL,BC           ;  start of module,
28B7 C8        12460          RET     Z               ;  we are finished
28B8 DD6E08    12470          LD      L,(IX+8)        ;P/u DCB being spooled
28BB DD6609    12480          LD      H,(IX+9)
28BE EB        12490          EX      DE,HL           ;SPOOL to HL,
28BF 0603      12500          LD      B,3             ;  device to DE
28C1 F3        12510          DI
28C2 7E        12520 SWAP1    LD      A,(HL)          ;Undo the SPOOLer
28C3 3600      12530          LD      (HL),0          ;  by swapping back the
28C5 12        12540          LD      (DE),A          ;  original vector and
28C6 2C        12550          INC     L               ;  zeroing the spool DCB
28C7 1C        12560          INC     E
28C8 10F8      12570          DJNZ    SWAP1
28CA 2C        12580          INC     L               ;Point to name field
28CB 2C        12590          INC     L               ;  & wipe it out
28CC 2C        12600          INC     L
28CD 3600      12610          LD      (HL),0
28CF 2C        12620          INC     L
28D0 3600      12630          LD      (HL),0
28D2 FB        12640          EI
28D3 C9        12650          RET                     ;Done - return
               12660 ;
               12670 ;        Routine to CLEAR the SPOOL buffer
               12680 ;
28D4 FDCB0346  12690 CLEAR    BIT     0,(IY+'D'-'A')  ;Spool active?
28D8 CA2524    12700          JP      Z,SPLOFM        ;Go if not active
28DB DD210000  12710          LD      IX,$-$          ;Parms in himem
28DD           12720 HIPARM   EQU     $-2
28DF DD6E0B    12730          LD      L,(IX+11)       ;P/u SPLFCB pointer
28E2 DD660C    12740          LD      H,(IX+12)
28E5 DD4E0A    12750          LD      C,(IX+10)       ;P/u RAM bank used
28E8 CBB9      12760          RES     7,C             ;Set on xfer to bank
28EA 0600      12770          LD      B,0             ;Function to load bank
28EC F3        12780          DI                      ;Interrupts off now
28ED           12790          @@BANK
28ED 3E66      00087          LD      A,102
28EF EF        00088          RST     40
28F0 C5        12800          PUSH    BC              ;Save old bank
```

```
28F1 E5          12810          PUSH    HL              ;Xfer FCB pointer
28F2 DDE1        12820          POP     IX              ;  to IX
28F4 DD6E24      12830          LD      L,(IX+24H)      ;Get LINK map pointer
28F7 DD6625      12840          LD      H,(IX+25H)
28FA E5          12850          PUSH    HL
28FB 3600        12860 CLEAR1   LD      (HL),Ø          ;Reset every link
28FD 23          12870          INC     HL
28FE 3600        12880          LD      (HL),Ø
2900 23          12890          INC     HL              ;Bypass page pointer
2901 23          12900          INC     HL
2902 7E          12910          LD      A,(HL)          ;Ck for TABEND
2903 23          12920          INC     HL
2904 3C          12930          INC     A               ;End if it was X'FF'
2905 20F4        12940          JR      NZ,CLEAR1
2907 E1          12950          POP     HL              ;Point to LINK start
2908 DD7526      12960          LD      (IX+26H),L      ;Reset PUTBLK$
290B DD7427      12970          LD      (IX+27H),H
290E DD7528      12980          LD      (IX+28H),L      ;Reset GETBLK$
2911 DD7429      12990          LD      (IX+29H),H
2914 7C          13000          LD      A,H             ;Show 1st LINK position
2915 75          13010          LD      (HL),L          ;  is the "last" one by
2916 23          13020          INC     HL              ;  loading it to point
2917 77          13030          LD      (HL),A          ;  to itself
2918 23          13040          INC     HL              ;P/u the page pointer
2919 46          13050          LD      B,(HL)
291A DD7021      13060          LD      (IX+21H),B      ;Stuff PUTNXT$
291D DD362000    13070          LD      (IX+20H),Ø
2921 DD7023      13080          LD      (IX+23H),B      ;Stuff GETNXT$
2924 DD3622ØØ    13090          LD      (IX+22H),Ø
2928 C1          13100          POP     BC              ;Reload previous bank
2929             13110          @@BANK
2929 3E66        ØØØ89          LD      A,102
292B EF          ØØØ90          RST     4Ø
292C FB          13120          EI
292D C31124      13130          JP      CLEARD
                 13140 ;
                 13150 ;        Relocate internal references in driver
                 13160 ;        HL = relocation table
                 13170 ;        DE = pointer to last byte of module
                 13180 ;        BC = pointer to last RAM byte to use
                 13190 ;
2930 DDE5        13200 RELO     PUSH    IX              ;Save index reg
2932 E5          13210          PUSH    HL              ;Xfer rel tab pointer
2933 DDE1        13220          POP     IX
2935 210000      13230 SVEND    LD      HL,$-$          ;P/u last byte used
2938 22112D      13240          LD      (DVRABGN+2),HL  ;  & stuff into module
293B 60          13250          LD      H,B             ;Xfer last byte to
293C 69          13260          LD      L,C             ;  use to HL
293D AF          13270          XOR     A               ;Clear carry flag
293E ED52        13280          SBC     HL,DE
2940 44          13290          LD      B,H             ;Move to BC
2941 4D          13300          LD      C,L
2942 DD7E00      13310          LD      A,(IX)          ;Get table length
2945 DD23        13320          INC     IX
2947 DD6E00      13330 RLOOP    LD      L,(IX)          ;Get address to change
294A DD6601      13340          LD      H,(IX+1)
294D 5E          13350          LD      E,(HL)          ;P/U address
294E 23          13360          INC     HL
294F 56          13370          LD      D,(HL)
2950 EB          13380          EX      DE,HL           ;Offset it
2951 09          13390          ADD     HL,BC
```

```
2952 EB        13400          EX      DE,HL
2953 72        13410          LD      (HL),D          ;And put back
2954 2B        13420          DEC     HL
2955 73        13430          LD      (HL),E
2956 DD23      13440          INC     IX
2958 DD23      13450          INC     IX
295A 3D        13460          DEC     A
295B 20EA      13470          JR      NZ,RLOOP        ;Loop till done
295D DDE1      13480          POP     IX
295F C9        13490          RET
               13500 ;
2960 43        13510 NOFIND$ DB       'Can''t locate SPOOL in memory',CR
     61 6E 27 74 20 6C 6F 63
     61 74 65 20 53 50 4F 4F
     4C 20 69 6E 20 6D 65 6D
     6F 72 79 0D
297D 53        13520 CLEARD$ DB       'SPOOL buffer cleared',CR
     50 4F 4F 4C 20 62 75 66
     66 65 72 20 63 6C 65 61
     72 65 64 0D
2992 44        13530 PAUSED$ DB       'Despooling now paused',CR
     65 73 70 6F 6F 6C 69 6E
     67 20 6E 6F 77 20 70 61
     75 73 65 64 0D
29A8 44        13540 RESUMD$ DB       'Despooling now resumed',CR
     65 73 70 6F 6F 6C 69 6E
     67 20 6E 6F 77 20 72 65
     73 75 6D 65 64 0D
29BF 53        13550 SPLOFM$ DB       'Spooler is not active',CR
     70 6F 6F 6C 65 72 20 69
     73 20 6E 6F 74 20 61 63
     74 69 76 65 0D
29D5 53        13560 SPLONM$ DB       'Spooler is already active',CR
     70 6F 6F 6C 65 72 20 69
     73 20 61 6C 72 65 61 64
     79 20 61 63 74 69 76 65
     0D
29EF 53        13570 SPLOPM$ DB       'Spooler already active but paused',CR
     70 6F 6F 6C 65 72 20 61
     6C 72 65 61 64 79 20 61
     63 74 69 76 65 20 62 75
     74 20 70 61 75 73 65 64
     0D
2A11 43        13580 PRMBGM$ DB       'Cannot reinstall with altered parameters',CR
     61 6E 6E 6F 74 20 72 65
     69 6E 73 74 61 6C 6C 20
     77 69 74 68 20 61 6C 74
     65 72 65 64 20 70 61 72
     61 6D 65 74 65 72 73 0D
2A3A 53        13590 SPLACT$ DB       'Spooler is now operational',CR
     70 6F 6F 6C 65 72 20 69
     73 20 6E 6F 77 20 6F 70
     65 72 61 74 69 6F 6E 61
     6C 0D
2A55 43        13600 NOMEM$  DB       'Can''t reclaim memory space',CR
     61 6E 27 74 20 72 65 63
     6C 61 69 6D 20 6D 65 6D
     6F 72 79 20 73 70 61 63
     65 0D
2A70 4D        13610 GOTMEM$ DB       'Memory space reclaimed',CR
     65 6D 6F 72 79 20 73 70
```

```
     61 63 65 2Ø 72 65 63 6C
     61 69 6D 65 64 ØD
2A87 42        13620 FREBNK$ DB      'Bank x released',CR
     61 6E 6B 2Ø 78 2Ø 72 65
     6C 65 61 73 65 64 ØD
2A97 52        13630 INUSE$  DB      'Requested bank in use',CR
     65 71 75 65 73 74 65 64
     2Ø 62 61 6E 6B 2Ø 69 6E
     2Ø 75 73 65 ØD
2AAD 4E        13640 CANT$   DB      'No memory space available',CR
     6F 2Ø 6D 65 6D 6F 72 79
     2Ø 73 7Ø 61 63 65 2Ø 61
     76 61 69 6C 61 62 6C 65
     ØD
2AC7 44        13650 BADDCB$ DB      'Device driver incompatible',CR
     65 76 69 63 65 2Ø 64 72
     69 76 65 72 2Ø 69 6E 63
     6F 6D 7Ø 61 74 69 62 6C
     65 ØD
ØØ2Ø           13660 SPLDCB  DS      32
2BØ2 58        13670 DSKSPC  DB      'XX/'
     58 2F
2BØ5 53        13680 SPLEXT  DB      'SPL',3
     5Ø 4C Ø3
2BØ9 2A        13690 PR$     DB      '*PR',ETX
     5Ø 52 Ø3
               13700 ;
2BØD D2        13710 PRMTBL$ DB      'R'!8ØH         ;6.x table format
ØØ8Ø           13720 VAL     EQU     8ØH
ØØ4Ø           13730 SW      EQU     4ØH
ØØ2Ø           13740 STR     EQU     2ØH
ØØ1Ø           13750 SGL     EQU     1ØH
2BØE 93        13760         DB      VAL!SGL!3,'MEM',Ø
     4D 45 4D ØØ
2B13 8D26      13770         DW      MPARM+1
2B15 94        13780         DB      VAL!SGL!4,'DISK',Ø
     44 49 53 4B ØØ
2B1B 1B26      13790         DW      DPARM+1
2B1D 43        13800         DB      SW!3,'OFF',Ø
     4F 46 46 ØØ
2B22 1925      13810         DW      OPARM+1
2B24 41        13820         DB      SW!1,'N',Ø
     4E ØØ
2B27 1925      13830         DW      OPARM+1
2B29 94        13840         DB      VAL!SGL!4,'BANK',Ø
     42 41 4E 4B ØØ
2B2F B625      13850         DW      BPARM+1
2B31 55        13860         DB      SW!SGL!5,'PAUSE',Ø
     5Ø 41 55 53 45 ØØ
2B38 2725      13870         DW      PPARM+1
2B3A 56        13880         DB      SW!SGL!6,'RESUME',Ø
     52 45 53 55 4D 45 ØØ
2B42 2Ø25      13890         DW      RPARM+1
2B44 55        13900         DB      SW!SGL!5,'CLEAR',Ø
     43 4C 45 41 52 ØØ
2B4B 2E25      13910         DW      CPARM+1
2B4D ØØ        13920         NOP
               13930 ;
2CØØ           13940         ORG     $<-8+1<8
Ø1ØØ           13950 BUFFER  DS      256
```

```
                    13970 ;
2DØØ                13980 *GET    LBSPOOLB:3
                    13990 ;LBSPOOLB/ASM - Actual spool driver
                    14000 ;
                    14010 ;
                    14020 ;       This is the spool/despool driver - bank Ø
                    14030 ;
2DØØ CDC32D         14040 FIX1    CALL    SPLR2           ;Code mods to module-A
2DØ1                14050 RX16    EQU     $-2             ;  if strictly bank-Ø
2DØ3 AF             14060         XOR     A
2DØ4 C9             14070         RET
2DØ5 CD732E         14080         CALL    DSPLR1
2DØ6                14090 RX17    EQU     $-2
2DØ8 AF             14100         XOR     A
2DØ9 C9             14110         RET
2DØA CDBA2E         14120         CALL    GETBLK2
2DØB                14130 RX18    EQU     $-2
2DØD AF             14140         XOR     A
2DØE C9             14150         RET
                    14160 ;
2DØF                14170 DVRABGN EQU     $
2DØF 1819           14180         JR      DVREPT          ;Branch around linkage
2D11 ØØØØ           14190         DW      $-$             ;Last byte used
2D13 Ø5             14200         DB      5
2D14 53             14210 SPOOL$  DB      'SPOOL'
     5Ø 4F 4F 4C
2D19 ØØØØ           14220 SØDCB   DW      Ø,Ø             ;Space for DCB pointers
     ØØØØ
2D1D ØØØØ           14230 PRMSAV  DW      Ø               ;HIGH$ before space alloc
2D1F ØE2D           14240         DW      DVRABGN-1       ;HIGH$ after space alloc
2D21 ØØØØ           14250         DW      Ø               ;Previous MPARM
2D23 ØØØØ           14260         DW      Ø               ;Previous DPARM
2D25 ØØØØ           14270 MODDVR  DW      Ø               ;DCB address of device
2D27 ØØ             14280 SPLBNK  DB      Ø               ;Bank of RAM
2D28 332F           14290         DW      SPLFCB          ;FCB pointer
2D28                14300 RX8     EQU     $-2
                    14310 ;
2D2A 283Ø           14320 DVREPT  JR      Z,PUTCHAR       ;Go on @PUT
2D2C 3ØØ3           14330         JR      NC,CKCTL        ;Go on @CTL
2D2E 3EØØ           14340         LD      A,Ø             ;Show nothing on @GET
2D3Ø C9             14350         RET
2D31 AF             14360 CKCTL   XOR     A
2D32 C9             14370         RET                     ;Show available on CTL-Ø
                    14380 ;
                    14390 ;       Move stack low if needed
                    14400 ;
2D33 E5             14410 STKCTL  PUSH    HL
2D34 21Ø58Ø         14420         LD      HL,8ØØ5H
2D37 39             14430         ADD     HL,SP           ;Stack in what bank?
2D38 E1             14440         POP     HL
2D39 DØ             14450         RET     NC              ;Ok if stack low
2D3A F3             14460         DI
2D3B 225A2D         14470         LD      (SVHL),HL
2D3C                14480 RX19    EQU     $-2
2D3E E1             14490         POP     HL              ;Return address
2D3F ED73582D       14500         LD      (SAVSTK),SP     ;Save the current stack
2D41                14510 RX2Ø    EQU     $-2
2D43 316ØØ3         14520         LD      SP,STACK$-2ØH   ;Force it low
2D46 E5             14530         PUSH    HL              ;User ret back on
2D47 21512D         14540         LD      HL,SPOLRET
2D48                14550 RX21    EQU     $-2
2D4A E3             14560         EX      (SP),HL         ;Now, ultimate ret 1st,
```

```
2D4B E5       14570          PUSH    HL              ;  and local ret 2nd
2D4C 2A5A2D   14580          LD      HL,(SVHL)       ;Restore HL
2D4D          14590 RX23     EQU     $-2
2D4F FB       14600          EI
2D50 C9       14610          RET
              14620 ;
              14630 SPOLRET
2D51 F3       14640          DI                      ;Just in case
2D52 ED7B582D 14650          LD      SP,(SAVSTK)     ;Get the old stack
2D54          14660 RX22     EQU     $-2
2D56 FB       14670          EI
2D57 C9       14680          RET
2D58 0000     14690 SAVSTK   DW      0
2D5A 0000     14700 SVHL     DW      0
              14710 ;
              14720 ;        Character to PUT - Get bank activated
              14730 ;
2D5C 21B82D   14740 PUTCHAR  LD      HL,SPLR1A       ;Point to transfer addr
2D5D          14750 RX01     EQU     $-2
2D5F 59       14760          LD      E,C             ;Xfer char to PUT
2D60 018000   14770          LD      BC,0<8!0!80H    ;P/u function & bank req
2D61          14780 BANKX1   EQU     $-2
2D63 CD332D   14790          CALL    STKCTL          ;Move stack low?
2D64          14800 RX24     EQU     $-2
2D66          14810          @@BANK                  ;  & invoke bank transfer
2D66 3E66     00091          LD      A,102
2D68 EF       00092          RST     40
2D69 C9       14820          RET
              14830 ;
              14840 ;        Task processor despooling routine
              14850 ;
2D6A 6C2D     14860 DSPLTCB  DW      DSPLR           ;Despool task control blk
2D6A          14870 RX02     EQU     $-2
2D6C 3A0000   14880 DSPLR    LD      A,($-$)         ;P/u PFLAG$ & ck if
2D6D          14890 DFLAG    EQU     $-2             ;  despooling is active
2D6F 07       14900          RLCA
2D70 D0       14910          RET     NC              ;Back if not
2D71 2A192D   14920          LD      HL,(S0DCB)      ;If spooler is inactive
2D72          14930 RX08     EQU     $-2             ;  then don't try to
2D74 2C       14940          INC     L               ;  do any despooling
2D75 4E       14950          LD      C,(HL)          ;Get DCB vector
2D76 2C       14960          INC     L
2D77 46       14970          LD      B,(HL)          ;  into BC
2D78 210F2D   14980          LD      HL,DVRABGN      ;P/u spooler start
2D79          14990 RX09     EQU     $-2
2D7B AF       15000          XOR     A
2D7C ED42     15010          SBC     HL,BC           ;If the same, then
2D7E C8       15020          RET     Z               ;  module is not active
2D7F 21642E   15030 FIX2UP   LD      HL,DSPLR0       ;Point to transfer addr
2D80          15040 RX03     EQU     $-2
2D82 018000   15050          LD      BC,0<8!0!80H    ;P/u function & bank req
2D83          15060 BANKX2   EQU     $-2
2D85 CD332D   15070          CALL    STKCTL          ;Move stack low?
2D86          15080 RX25     EQU     $-2
2D88          15090          @@BANK                  ;  & invoke bank transfer
2D88 3E66     00093          LD      A,102
2D8A EF       00094          RST     40
2D8B C9       15100          RET
              15110 ;
              15120 ;        KI task routine to get a disk block
              15130 ;
```

```
2D8C 21BØ2E   15140 GETBLK  LD      HL,GETBLK1      ;Point to transfer addr
2D8D          15150 RXØ4    EQU     $-2
2D8F Ø18ØØØ   15160         LD      BC,Ø<8!Ø!8ØH    ;P/u function & bank req
2D9Ø          15170 BANKX3  EQU     $-2
2D92 CD332D   15180         CALL    STKCTL          ;Move stack low?
2D93          15190 RX26    EQU     $-2
2D95          15200         @@BANK                  ;  & invoke bank transfer
2D95 3E66     ØØØ95         LD      A,1Ø2
2D97 EF       ØØØ96         RST     4Ø
2D98 ØØ       15210 GETBLKØ DB      Ø,Ø,Ø           ;Orig @KITSK
     ØØ ØØ
2D9A          15220 DVRAEND EQU     $-1
2D9B ØD       15230 RELTABA DB      TABALEN
2D9C 6A2D     15240         DW      RXØ2,RXØ5,RXØ6,RXØ8,RXØ9
     D327 DB27 722D 792D
2DA6 3C2D     15250         DW      RX19,RX2Ø,RX21,RX22,RX23
     412D 482D 542D 4D2D
2DBØ 642D     15260         DW      RX24,RX25,RX26
     862D 932D
ØØØD          15270 TABALEN EQU     $-RELTABA/2
              15280 ;
              15290 ;
2DB6          15300 DVRBBGN EQU     $
              15310 ;
              15320 ;       Spool/despool module in bank x
              15330 ;
2DB6 ØØØØ     15340 SØDCB1  DW      Ø
              15350 ;
              15360 ;       Spool a character
              15370 ;
2DB8 E5       15380 SPLR1A  PUSH    HL              ;Save new transfer addr
2DB9 C5       15390         PUSH    BC              ;Save invocation
2DBA 4B       15400         LD      C,E             ;Get char to PUT
2DBB CDC32D   15410         CALL    SPLR2           ;Spool a char
2DBC          15420 RX1     EQU     $-2
2DBE C1       15430         POP     BC
2DBF E1       15440         POP     HL
2DCØ          15450         @@BANK                  ;Transfer back
2DCØ 3E66     ØØØ97         LD      A,1Ø2
2DC2 EF       ØØØ98         RST     4Ø
2DC3 DD21332F 15460 SPLR2   LD      IX,SPLFCB
2DC5          15470 RX2     EQU     $-2
2DC7 DD6E2Ø   15480         LD      L,(IX+2ØH)      ;P/u next buffer position
2DCA DD6621   15490         LD      H,(IX+21H)
2DCD 71       15500         LD      (HL),C          ;Stuff char received
2DCE F3       15510         DI
2DCF DD342Ø   15520 SPLR3   INC     (IX+2ØH)        ;Inc lo order
2DD2 28Ø7     15530         JR      Z,SPLR5         ;Jump if block full
2DD4 CD6E2E   15540         CALL    DSPLR1X         ;Output to *PR if avail
2DD5          15550 RX3     EQU     $-2
2DD7 FB       15560         EI
2DD8 C3BA2E   15570         JP      GETBLK2
2DD9          15580 RX4     EQU     $-2
2DDB DD6E24   15590 SPLR5   LD      L,(IX+24H)      ;P/u pointer to table
2DDE DD6625   15600         LD      H,(IX+25H)
2DE1 7E       15610 SPLR6   LD      A,(HL)          ;Find spare pointer field
2DE2 23       15620         INC     HL
2DE3 B6       15630         OR      (HL)
2DE4 28Ø5     15640         JR      Z,SPLR7         ;Jump if spare
2DE6 23       15650         INC     HL              ;  else index to next
2DE7 23       15660         INC     HL
```

```
2DE8 23        15670          INC     HL
2DE9 18F6      15680          JR      SPLR6
2DEB 2B        15690 SPLR7    DEC     HL
2DEC E5        15700          PUSH    HL              ;Save pointer
2DED 23        15710          INC     HL
2DEE 23        15720          INC     HL
2DEF 46        15730          LD      B,(HL)          ;P/u memory segment byte
2DFØ 23        15740          INC     HL              ;  or lo-order recno
2DF1 7E        15750          LD      A,(HL)          ;End of table?
2DF2 FEFF      15760          CP      ØFFH
2DF4 2ØØB      15770          JR      NZ,SPLR1Ø       ;Jump if still space
2DF6 E1        15780          POP     HL
2DF7 CD6E2E    15790 SPLR8    CALL    DSPLR1X         ;Output to *PR if avail
2DF8           15800 RX5      EQU     $-2
2DFA FB        15810          EI
2DFB CDBA2E    15820 SPLR9    CALL    GETBLK2         ;Disk rec to bring back?
2DFC           15830 RX6      EQU     $-2
2DFE F3        15840          DI
2DFF 18DA      15850          JR      SPLR5
2EØ1 CB7F      15860 SPLR1Ø   BIT     7,A             ;Test mem or disk block
2EØ3 2Ø21      15870          JR      NZ,SPLR12       ;Jump if disk pointer
               15880 ;
               15890 ;        Found a spare memory pointer
               15900 ;
2EØ5 E1        15910          POP     HL              ;Rcvr ptr to start
2EØ6 E5        15920          PUSH    HL              ;  of spare table field
2EØ7 75        15930          LD      (HL),L          ;Place address of table
2EØ8 7C        15940          LD      A,H             ;  position into table
2EØ9 23        15950          INC     HL
2EØA 77        15960          LD      (HL),A
2EØB DD7Ø21    15970          LD      (IX+21H),B      ;Stuff hi-order free mem
2EØE DD6E26    15980          LD      L,(IX+26H)      ;P/u current table ptr
2E11 DD6627    15990          LD      H,(IX+27H)
2E14 DD752A    16ØØØ          LD      (IX+2AH),L      ;Stuff last block pointer
2E17 DD742B    16Ø1Ø          LD      (IX+2BH),H
2E1A C1        16Ø2Ø          POP     BC              ;Rcvr ptr to field start
2E1B 71        16Ø3Ø          LD      (HL),C          ;Set up a link to this
2E1C 23        16Ø4Ø          INC     HL              ;  block in last block
2E1D 7Ø        16Ø5Ø          LD      (HL),B
2E1E DD7126    16Ø6Ø          LD      (IX+26H),C      ;Reset current block ptr
2E21 DD7Ø27    16Ø7Ø          LD      (IX+27H),B
2E24 FB        16Ø8Ø          EI
2E25 C9        16Ø9Ø          RET
               161ØØ ;
2E26           16110 SPLR12   EQU     $
2E26 FB        16120          EI
2E27 48        16130          LD      C,B             ;Lo order recno
2E28 E6ØF      16140          AND     ØFH             ;Hi-order recno
2E2A 47        16150          LD      B,A
2E2B DDE5      16160          PUSH    IX              ;Position file to recno
2E2D D1        16170          POP     DE              ;FCB to DE
2E2E           16180          @@POSN                  ;Posn to desired record
2E2E 3E42      ØØØ99          LD      A,66
2E3Ø EF        ØØ1ØØ          RST     4Ø
2E31 ØØ        16190          DB      Ø,Ø,Ø           ;Reserved
     ØØ ØØ
2E34 DD4E26    162ØØ          LD      C,(IX+26H)      ;P/u current table ptr
2E37 DD4627    16210          LD      B,(IX+27H)
2E3A E1        16220          POP     HL              ;P/u ptr to this disk
2E3B E5        16230          PUSH    HL              ;Block table field
2E3C 71        16240          LD      (HL),C          ;Tell this disk block
```

```
2E3D 23          16250          INC     HL              ;  field which memory
2E3E 70          16260          LD      (HL),B          ;  block it is storing
2E3F C1          16270          POP     BC              ;Exchange link in last
2E40 DD6E2A      16280          LD      L,(IX+2AH)      ;  block pointer with new
2E43 DD662B      16290          LD      H,(IX+2BH)      ;  link since this memory
2E46 71          16300          LD      (HL),C          ;  block is now on disk
2E47 23          16310          INC     HL
2E48 70          16320          LD      (HL),B
2E49 DD712A      16330          LD      (IX+2AH),C
2E4C DD702B      16340          LD      (IX+2BH),B
2E4F 010001      16350          LD      BC,256          ;Block move the page
2E52 69          16360          LD      L,C             ;  into the low memory
2E53 DD6621      16370          LD      H,(IX+21H)      ;  buuffer fo disk I/O
2E56 D5          16380          PUSH    DE              ;Save fcb pointer
2E57 110023      16390          LD      DE,LOWBUF$
2E5A EDB0        16400          LDIR
2E5C D1          16410          POP     DE
2E5D             16420          @@WRITE                 ;Write out the block
2E5D 3E4B        00101          LD      A,75
2E5F EF          00102          RST     40
2E60 00          16430          DB      0,0,0           ;Reserved
     00 00
2E63 C9          16440          RET
                 16450 ;
2E64 E5          16460 DSPLR0   PUSH    HL              ;Save new transfer addr
2E65 C5          16470          PUSH    BC              ;Save invocation
2E66 CD732E      16480          CALL    DSPLR1          ;Send device a char
2E67             16490 RX7      EQU     $-2
2E69 C1          16500          POP     BC
2E6A E1          16510          POP     HL
2E6B             16520          @@BANK                  ;Transfer back
2E6B 3E66        00103          LD      A,102
2E6D EF          00104          RST     40
                 16530 ;
                 16540 ;        Routine will output a char to PR if available
                 16550 ;
2E6E 3A0000      16560 DSPLR1X  LD      A,($-$)         ;P/u PFLAG$ & ck if
2E6F             16570 DFLAG1   EQU     $-2             ;  despooling is active
2E71 07          16580          RLCA
2E72 D0          16590          RET     NC              ;Back if not active
2E73 010004      16600 DSPLR1   LD      BC,4<8!0        ;Init to CTL-0
2E76 DDE5        16610          PUSH    IX              ;Save current DCB vector
2E78 DD2AB62D    16620 PUDCB    LD      IX,(S0DCB1)     ;Point to printer
2E7A             16630 RX9      EQU     $-2
2E7C             16640          @@CHNIO                 ;Ck *PR status
2E7C 3E14        00105          LD      A,20
2E7E EF          00106          RST     40
2E7F DDE1        16650          POP     IX
2E81 C0          16660          RET     NZ              ;Ret if not avail
2E82 DD21332F    16670 PUFCB    LD      IX,SPLFCB
2E84             16680 RX10     EQU     $-2
2E86 DD7E23      16690          LD      A,(IX+23H)      ;See if char in buffer
2E89 B7          16700          OR      A               ;  to send to the printer
2E8A C8          16710          RET     Z               ;Ret if none
2E8B 67          16720          LD      H,A             ;None also if GET pointer
2E8C DDBE21      16730          CP      (IX+21H)        ;  is same as PUT pointer
2E8F DD7E22      16740          LD      A,(IX+22H)
2E92 2004        16750          JR      NZ,DSPLR2
2E94 DDBE20      16760          CP      (IX+20H)
2E97 C8          16770          RET     Z               ;In & out the same
2E98 6F          16780 DSPLR2   LD      L,A             ;P/u the character
```

```
2E99 4E        16790          LD      C,(HL)          ;Xfer to output reg
2E9A DDE5      16800          PUSH    IX              ;Save this dcb vector
2E9C Ø6Ø2      16810          LD      B,2             ;Set chain for PUT
2E9E DD2AB62D  16820 PUDCB1   LD      IX,(SØDCB1)     ;Pt to spooled DCB
2EAØ           16830 RX11     EQU     $-2
2EA2           16840          @@CHNIO                 ;Output the char
2EA2 3E14      ØØ1Ø7          LD      A,2Ø
2EA4 EF        ØØ1Ø8          RST     4Ø
2EA5 DDE1      16850          POP     IX              ;Rcvr dcb
2EA7 DD3422    16860          INC     (IX+22H)        ;Bump PUT (out) ptr
2EAA CØ        16870          RET     NZ              ;Ret if still more in buf
2EAB DD3623ØØ  16880          LD      (IX+23H),Ø      ;Block empty, turn off
2EAF C9        16890          RET                     ;  hi-order & exit
               16900 ;
               16910 ;        Routine to see if we can grab a block back from
               16920 ;        the disk file & insert into a spare memory block
               16930 ;
2EBØ E5        16940 GETBLK1  PUSH    HL              ;Save new transfer addr
2EB1 C5        16950          PUSH    BC              ;Save invocation
2EB2 CDBA2E    16960          CALL    GETBLK2         ;Spool a char
2EB3           16970 RX12     EQU     $-2
2EB5 C1        16980          POP     BC
2EB6 E1        16990          POP     HL
2EB7           17000          @@BANK                  ;Transfer back
2EB7 3E66      ØØ1Ø9          LD      A,1Ø2
2EB9 EF        ØØ11Ø          RST     4Ø
2EBA F5        17010 GETBLK2  PUSH    AF
2EBB DDE5      17020          PUSH    IX              ;Save in case DSPLR1
2EBD 3A562F    17030 GETBLK3  LD      A,(SPLFCB+23H)  ;Jump if block being
2EBE           17040 RX13     EQU     $-2
2ECØ B7        17050          OR      A               ;  printed is not empty
2EC1 2ØD       17060          JR      NZ,GETBLK6
               17070 ;
               17080 ;        The block's been despooled. Get 1 from D,M?
               17090 ;
2EC3 E5        17100          PUSH    HL
2EC4 D5        17110          PUSH    DE
2EC5 C5        17120          PUSH    BC
2EC6 DD21332F  17130 GETBLK4  LD      IX,SPLFCB
2EC8           17140 RX14     EQU     $-2
2ECA DD6E28    17150          LD      L,(IX+28H)      ;P/u pointer of block
2ECD DD6629    17160          LD      H,(IX+29H)      ;  being printed
2EDØ E5        17170          PUSH    HL
2ED1 4E        17180          LD      C,(HL)          ;Point to POSN or hi-mem
2ED2 23        17190          INC     HL              ;  byte of field this
2ED3 46        17200          LD      B,(HL)          ;  field points to
2ED4 21Ø2ØØ    17210          LD      HL,2            ;P/u the POSN or hi-order
2ED7 Ø9        17220          ADD     HL,BC
2ED8 7E        17230          LD      A,(HL)
2ED9 23        17240          INC     HL
2EDA CB7E      17250          BIT     7,(HL)          ;Jump if it is pointing
2EDC 2Ø1B      17260          JR      NZ,GETBLK7      ;  to a disk block
2EDE DD7723    17270          LD      (IX+23H),A      ;Reset hi order of where
2EE1 E1        17280          POP     HL              ;  next block coming from
2EE2 36ØØ      17290          LD      (HL),Ø          ;Show that this block
2EE4 23        17300          INC     HL              ;  is spare
2EE5 36ØØ      17310          LD      (HL),Ø
2EE7 DD7128    17320          LD      (IX+28H),C      ;Set the next block
2EEA DD7Ø29    17330          LD      (IX+29H),B      ;  to be printed
2EED C1        17340 GETBLK5  POP     BC
2EEE D1        17350          POP     DE
```

```
2EEF E1        17360          POP     HL
2EF0           17370 GETBLK6  EQU     $
2EF0 F3        17380          DI
2EF1 CD6E2E    17390          CALL    DSPLR1X         ;Check if we can despool
2EF2           17400 RX15     EQU     $-2             ;  a character while
2EF4 FB        17410          EI                      ;  we are in here
2EF5 DDE1      17420          POP     IX
2EF7 F1        17430          POP     AF
2EF8 C9        17440          RET
               17450 ;
               17460 ;        Next block to print is on disk, get it
               17470 ;
2EF9 4F        17480 GETBLK7  LD      C,A             ;Set up position
2EFA 7E        17490          LD      A,(HL)
2EFB E60F      17500          AND     0FH
2EFD 47        17510          LD      B,A
2EFE DDE5      17520          PUSH    IX              ;Set FCB & position
2F00 D1        17530          POP     DE              ;  the spool file
2F01           17540          @@POSN
2F01 3E42      00111          LD      A,66
2F03 EF        00112          RST     40
2F04 00        17550          DB      0,0,0           ;Reserved
     00 00
2F07 E1        17560          POP     HL              ;Rcvr table ptr of
2F08 E5        17570          PUSH    HL              ;  block just finished
2F09 23        17580          INC     HL
2F0A 23        17590          INC     HL              ;P/u its buffer location
2F0B           17600          @@READ                  ;Read block from file
2F0B 3E43      00113          LD      A,67
2F0D EF        00114          RST     40
2F0E 00        17610          DB      0,0,0           ;Reserved
     00 00
2F11 010001    17620          LD      BC,256          ;Block move from the
2F14 56        17630          LD      D,(HL)          ;  low memory buffer
2F15 59        17640          LD      E,C             ;  to RAM page
2F16 210023    17650          LD      HL,LOWBUF$
2F19 EDB0      17660          LDIR
2F1B E1        17670          POP     HL              ;Tbl ptr of block fin'd
2F1C E5        17680          PUSH    HL
2F1D 7E        17690          LD      A,(HL)          ;P/u field addrress of
2F1E 23        17700          INC     HL              ;  the next table field
2F1F 66        17710          LD      H,(HL)          ;  that gets printed
2F20 6F        17720          LD      L,A
2F21 4E        17730          LD      C,(HL)          ;P/u field address of the
2F22 3600      17740          LD      (HL),0          ;  block after that to
2F24 23        17750          INC     HL              ;  print & show this one
2F25 46        17760          LD      B,(HL)          ;  as empty (this was the
2F26 3600      17770          LD      (HL),0          ;  disk one just vacated)
2F28 E1        17780          POP     HL
2F29 71        17790          LD      (HL),C          ;Reset current printing
2F2A 23        17800          INC     HL              ;  block & point to
2F2B 70        17810          LD      (HL),B          ;  new next block
2F2C 23        17820          INC     HL
2F2D 46        17830          LD      B,(HL)
2F2E DD7023    17840          LD      (IX+23H),B      ;Show where new printing
2F31 18BA      17850          JR      GETBLK5         ;  block is & exit
               17860 ;
               17870 ;        File control block area for spool file
               17880 ;
0020           17890 SPLFCB   DS      32
               17900 ;
```

```
                17910 ;       Pointer to the memory location where the next
                17920 ;       character received from *XX can be put
                17930 ;
ØØØ2            17940 PUTNXT$ DS      2
                17950 ;
                17960 ;       Pointer to where the next character to print
                17970 ;       is going to come from
                17980 ;
ØØØ2            17990 GETNXT$ DS      2
                18000 ;
                18010 ;       Pointer to start of block map table
                18020 ;       table configuration is:
                18030 ;        TAB+Ø/1 -> store the TAB address if the field
                18040 ;        is currently in use for receiving. If a
                18050 ;        block is not current, it stores the
                18060 ;        address of the next TAB to print. If
                18070 ;        a disk field, then stores the TAB address
                18080 ;        of the previous block (link back)
                18090 ;
                18100 ;        TAB+2 -> for a memory block, stores the hi-order
                18110 ;        byte of where the memory block starts. If
                18120 ;        a disk byte, stores the lo-order POSN #.
                18130 ;
                18140 ;        TAB+3 -> s/b Ø for a memory block or if a disk
                18150 ;        block, it contains the hi-order POSN
                18160 ;        vector with bit 7 set.
                18170 ;
ØØØ2            18180 MAPPTR$ DS      2
                18190 ;
                18200 ;       Pointer to the beginning of the memory table
                18210 ;       block pointer for the block that is receiving
                18220 ;       characters from *PR calls (the current block)
                18230 ;
ØØØ2            18240 PUTBLK$ DS      2
                18250 ;
                18260 ;       Pointer to the memory table field for the block
                18270 ;       that is printing characters
                18280 ;
ØØØ2            18290 GETBLK$ DS      2
                18300 ;
                18310 ;       Points to the table position of the previous
                18320 ;       block. This is used to reset the previous link
                18330 ;
2F5D ØØØØ       18340         DW      Ø
2F5E            18350 DVRBEND EQU     $-1
2F5F 15         18360 RELTABB DB      TABBLEN
2F6Ø 5D2D       18370         DW      RXØ1,RXØ3,RXØ4
     8Ø2D 8D2D
2F66 BC2D       18380         DW      RX1,RX2,RX3,RX4,RX5,RX6,RX7,RX8
     C52D D52D D92D F82D FC2D 672E 282D
2F76 7A2E       18390         DW      RX9,RX1Ø,RX11,RX12,RX13,RX14,RX15
     842E AØ2E B32E BE2E C82E F22E
2F84 Ø12D       18400         DW      RX16,RX17,RX18
     Ø62D ØB2D
ØØ15            18410 TABBLEN EQU     $-RELTABB/2
                18420 ;
                ØØ14Ø ;
2F8A            ØØ15Ø         SUBTTL  <>
24ØØ            ØØ16Ø         END     SPOOL
```

@$SYS	Ø8FØ	@@1	ØØØØ	@@2	ØØØØ		
@@3	ØØØØ	@@4	ØØØØ	@BANK	Ø877		
@BYTEIO	13ØØ	@CHNIO	Ø689	@CKBRKC	Ø553		
@CLS	Ø545	@CTL	Ø623	@DATE	Ø7A8		
@DIV16	Ø6E3	@DSP	Ø642	@DSPLY	Ø52D		
@FRENCH	ØØØØ	@GERMAN	ØØØØ	@GET	Ø638		
@HEX16	Ø7BD	@HEX8	Ø7C2	@HEXDEC	Ø6F6		
@HZ5Ø	ØØØØ	@INTL	ØØØØ	@JCL	Ø63Ø		
@KBD	Ø635	@KEY	Ø628	@KEYIN	Ø585		
@KITSK	ØØ89	@LOGER	Ø5Ø3	@LOGOT	Ø5ØØ		
@MOD2	ØØØØ	@MOD4	FFFF	@MSG	Ø53Ø		
@MUL16	Ø6C9	@OPREG	ØØ84	@PRINT	Ø528		
@PRT	Ø63D	@PUT	Ø645	@RSTNMI	ØFE9		
@RSTREG	Ø68Ø	@TIME	Ø78D	@USA	FFFF		
@VDCTL	ØB99	@VDCTL3	ØD38	@_VDCTL	ØD42		
ADDR_2_ROWCOL	ØDF1	BADDCB	2442	BADDCB$	2AC7		
BANKX1	2D61	BANKX2	2D83	BANKX3	2D9Ø		
BAR$	Ø2Ø1	BOOTST$	439D	BPARM	25B5		
BUFFER	2CØØ	BUR$	Ø2ØØ	CALCPTR	26A1		
CANT	244E	CANT$	2AAD	CASHK$	ØA7B		
CFLAG$	ØØ6C	CKCTL	2D31	CKLNK	258A		
CKRTE	257F	CKRTE1	2583	CLEAR	28D4		
CLEAR1	28FB	CLEARD	2411	CLEARD$	297D		
CORE$	Ø3ØØ	CPARM	252D	CR	ØØØD		
CRTBGN$	F8ØØ	DATE$	ØØ33	DAYTBL$	Ø4C7		
DCBKL$	ØØ31	DCT$	Ø47Ø	DFLAG	2D6D		
DFLAG$	ØØ6D	DFLAG1	2E6F	DIS_DO_RAM	Ø846		
DODATA$	ØB94	DODCB$	Ø21Ø	DO_CONTROL	ØC44		
DO_DSPCHAR	ØCB8	DO_INVERT_DIS	ØC8C	DO_INVERT_ENA	ØC89		
DO_INVERT_OFF	ØC9B	DO_MASK	ØØØØ	DO_RET	ØBCB		
DO_RET1	ØBCC	DO_SCROLL	ØCCE	DO_TABS	ØBEA		
DPARM	261A	DSKSPC	2BØ2	DSKTYP$	Ø4CØ		
DSPLR	2D6C	DSPLRØ	2E64	DSPLR1	2E73		
DSPLR1X	2E6E	DSPLR2	2E98	DSPLTCB	2D6A		
DTPMT$	Ø4C2	DVRABGN	2DØF	DVRAEND	2D9A		
DVRBBGN	2DB6	DVRBEND	2F5E	DVREND$	ØFF4		
DVREPT	2D2A	DVRHI$	Ø2Ø6	ENADIS_DO_RAM	Ø817		
ERREXIT	2454	ETX	ØØØ3	EXIT	24Ø7		
FDDINT$	ØØØE	FIX1	2DØØ	FIX2UP	2D7F		
FLGTAB$	ØØ6A	FNDDCB	2575	FREBNK$	2A87		
GBUF1	26BB	GBUF2	26C4	GETBLK	2D8C		
GETBLK$	2F5B	GETBLKØ	2D98	GETBLK1	2EBØ		
GETBLK2	2EBA	GETBLK3	2EBD	GETBLK4	2EC6		
GETBLK5	2EED	GETBLK6	2EFØ	GETBLK7	2EF9		
GETNXT$	2F55	GETPRM	25ØF	GET_@_ROWCOL	ØDAE		
GOTMEM$	2A7Ø	HERTZ$	Ø75Ø	HIGH$	Ø4ØE		
HIPARM	28DD	IFLAG$	ØØ72	INBUF$	Ø42Ø		
INTVC$	ØØ3E	INUSE	244A	INUSE$	2A97		
IOERR	2437	IPLDSK1	26FE	IPLDSK2	27Ø8		
IPLDSK3	271A	IPLMEM	26FØ	JCLCB$	Ø2Ø3		
JLDCB$	Ø23Ø	KCK@	Ø7D6	KFLAG$	ØØ74		
KIDATA$	Ø8FC	KIDCB$	Ø2Ø8	KITSK@	ØØ1F		
LBANK$	Ø2Ø2	LOCDCB	25D7	LOGMEM	2888		
LOWBUF$	23ØØ	MAPPTR$	2F57	MAXDAY$	Ø4Ø1		
MODDVR	2D25	MODOUT$	ØØ76	MONTBL$	Ø4DC		
MOVMODB	277B	MPARM	268C	NAMLP	25EC		
NFLAG$	ØØ77	NOFIND	2446	NOFIND$	296Ø		
NOMEM$	2A55	NOSET	27C8	OLDHI	27BC		

OPARM	2518	OPREG$	ØØ78	OPREG_SV_AREA	Ø86E
OPREG_SV_PTR	Ø835	PAKNAM$	Ø41Ø	PAR_ERR	ØØ2C
PAUSE	28Ø7	PAUSE@	Ø382	PAUSED	2419
PAUSED$	2992	PCSAVE$	Ø7AF	PDRV$	ØØ1B
PPARM	2526	PR$	2BØ9	PRDCB$	Ø218
PRMBGM	242D	PRMBGM$	2A11	PRMERR	2435
PRMERRA	262Ø	PRMSAV	2D1D	PRMTBL$	2BØD
PUDCB	2E78	PUDCB1	2E9E	PUFCB	2E82
PUHIGH	2658	PUTA@DE	ØDCD	PUTBLK$	2F59
PUTCHAR	2D5C	PUTNXT$	2F53	PUT_@	ØDCA
PUT_@_ROWCOL	ØDC6	RELO	293Ø	RELTABA	2D9B
RELTABB	2F5F	RESDCB	28A3	RESUMD	2415
RESUMD$	29A8	RESUME	281C	RFLAG$	ØØ7B
RLOOP	2947	ROWCOL_2_ADDR	ØDDØ	RPARM	251F
RST28	ØØ28	RSTOR$	Ø4C4	RXØ1	2D5D
RXØ2	2D6A	RXØ3	2D8Ø	RXØ4	2D8D
RXØ5	27D3	RXØ6	27DB	RXØ8	2D72
RXØ9	2D79	RX1	2DBC	RX1Ø	2E84
RX11	2EAØ	RX12	2EB3	RX13	2EBE
RX14	2EC8	RX15	2EF2	RX16	2DØ1
RX17	2DØ6	RX18	2DØB	RX19	2D3C
RX2	2DC5	RX2Ø	2D41	RX21	2D48
RX22	2D54	RX23	2D4D	RX24	2D64
RX25	2D86	RX26	2D93	RX3	2DD5
RX4	2DD9	RX5	2DF8	RX6	2DFC
RX7	2E67	RX8	2D28	RX9	2E7A
SØDCB	2D19	SØDCB1	2DB6	S1DCB$	Ø238
SAVESP	24ØA	SAVSTK	2D58	SETDVR	25AD
SET_SCROLL	ØCF3	SFLAG$	ØØ7C	SGL	ØØ1Ø
SIDCB$	Ø22Ø	SODCB$	Ø228	SPL1	249B
SPL2	249C	SPL2A	24CE	SPL2B	24DE
SPL2C	24E7	SPL3	24F3	SPL4	24FF
SPLACT	241D	SPLACT$	2A3A	SPLBNK	2D27
SPLDCB	2AE2	SPLEXT	2BØ5	SPLFCB	2F33
SPLOFF	2831	SPLOFM	2425	SPLOFM$	29BF
SPLONM	2421	SPLONM$	29D5	SPLOPM	2429
SPLOPM$	29EF	SPLR1Ø	2EØ1	SPLR12	2E26
SPLR1A	2DB8	SPLR2	2DC3	SPLR3	2DCF
SPLR5	2DDB	SPLR6	2DE1	SPLR7	2DEB
SPLR8	2DF7	SPLR9	2DFB	SPOLRET	2D51
SPOOL	24ØØ	SPOOL$	2D14	SPOOL1	2459
STACK$	Ø38Ø	START$	ØØØØ	STKCTL	2D33
STORHI	27A2	STR	ØØ2Ø	SVBGN	28B2
SVDCB	28A4	SVEND	2935	SVHL	2D5A
SW	ØØ4Ø	SWAP	27FA	SWAP1	28C2
TABALEN	ØØØD	TABBLEN	ØØ15	TIME$	ØØ2D
TIMER$	ØØ2C	TIMSL$	ØØ2B	TIMTSK$	Ø713
TMPMT$	Ø4C3	TRACE_INT	Ø7B1	TYPHK$	ØA8F
TYPTSK$	ØB26	USEPR	24C3	VAL	ØØ8Ø
VFLAG$	ØØ7F	ZERO$	Ø4Ø1	@@ABORT	6C5Ø
@@ADTSK	6CE3	@@BANK	71FB	@@BKSP	6EDB
@@BREAK	7211	@@CHNIO	6C3B	@@CKBRKC	725F
@@CKDRV	6D37	@@CKEOF	6EFØ	@@CKTSK	6CCE
@@CLOSE	6EC6	@@CLS	7249	@@CMNDI	6C7A
@@CMNDR	6C8F	@@CTL	6A9F	@@DATE	6C11
@@DCSTAT	6D76	@@DEBUG	6CB9	@@DECHEX	717B
@@DIRRD	7ØE8	@@DIRWR	7ØFD	@@DIV16	7166
@@DIV8	7151	@@DODIR	6D4C	@@DSP	6A63
@@DSPLY	6BØ3	@@ERROR	6CA4	@@EXIT	6C65

```
@@FEXT          7Ø55 @@FLAGS        71E5 @@FNAME        7Ø6A
@@FSPEC         7Ø4Ø @@GATRD        7ØD3 @@GATWR        7112
@@GET           6A77 @@GTDCB        7Ø94 @@GTDCT        7Ø7F
@@GTMOD         7ØA9 @@HDFMT        6E1E @@HEX16        71BA
@@HEX8          71A5 @@HEXDEC       719Ø @@HIGH$        71CF
@@INIT          6E9C @@KBD          6ADB @@KEY          6A4F
@@KEYIN         6AEF @@KLTSK        6D22 @@LOAD         7Ø16
@@LOC           6FØ5 @@LOF          6F1A @@LOGER        6B3A
@@LOGOT         6B4F @@MSG          6B86 @@MUL16        713C
@@MUL8          7127 @@OPEN         6EB1 @@PARAM        6BFC
@@PAUSE         6BE7 @@PEOF         6F2F @@POSN         6F44
@@PRINT         6B9B @@PRT          6AB3 @@PUT          6A8B
@@RAMDIR        6D61 @@RDSEC        6DF4 @@RDSSC        7ØBE
@@READ          6F59 @@REMOV        6E87 @@RENAM        6E72
@@REW           6F6E @@RMTSK        6CF8 @@RPTSK        6DØD
@@RREAD         6F83 @@RSLCT        6DDF @@RSTOR        6DAØ
@@RUN           7Ø2B @@RWRIT        6F98 @@SEEK         6DCA
@@SEEKSC        6FAD @@SKIP         6FC2 @@SLCT         6D8B
@@STEPI         6DB5 @@TIME         6C26 @@VDCTL        6BD2
@@VER           6FD7 @@VRSEC        6EØ9 @@WEOF         6FEC
@@WHERE         6AC7 @@WRITE        7ØØ1 @@WRSEC        6E33
@@WRSSC         6E48 @@WRTRK        6E5D
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

Command: SYSGEN

Library: SYS8/SYS

ISAM # : 1CH

```
                ØØ1ØØ ;LBSYSGEN/ASM - SYSGEN Command
ØØØØ            ØØ11Ø          TITLE   <SYSGEN - LS-DOS 6.2>
                ØØ12Ø ;
ØØ2C            ØØ13Ø PAR_ERR  EQU     44              ;Parameter Error
ØØØD            ØØ14Ø CR       EQU     13
43ØØ            ØØ15Ø MOD3BUF  EQU     43ØØH
ØØ28            ØØ16Ø RST28    EQU     28H
                ØØ17Ø ;
ØØØØ            ØØ18Ø *GET     SVCMAC:3                ;SVC Macro equivalents
                ØØ1Ø ;SVCMAC/ASM - LS-DOS Version VI
                ØØ2Ø *LIST     OFF
                Ø39ØØ *LIST    ON
                ØØ19Ø *LIST    OFF                     ;Get SYSØ/EQU
                ØØ21Ø *LIST    ON
                ØØ22Ø ;
24ØØ            ØØ23Ø          ORG     24ØØH
                ØØ24Ø ;
24ØØ            ØØ25Ø SYSGEN   @@CKBRKC                ;Break key down?
24ØØ 3E6A       ØØØØ1          LD      A,1Ø6
24Ø2 EF         ØØØØ2          RST     4Ø
24Ø3 28Ø4       ØØ26Ø          JR      Z,BEGINA        ;Ok if not
24Ø5 21FFFF     ØØ27Ø          LD      HL,-1           ;  else abort
24Ø8 C9         ØØ28Ø          RET
                ØØ29Ø ;
                ØØ3ØØ ;        Save stack & call the sysgen routine
                ØØ31Ø ;
24Ø9 ED731424   ØØ32Ø BEGINA   LD      (SAVESP+1),SP   ;Save the stack pointer
24ØD CD4624     ØØ33Ø          CALL    SYSGEN1         ;Do the SYSGEN
241Ø 21ØØØØ     ØØ34Ø EXIT     LD      HL,Ø            ;Set for no error
2413 31ØØØØ     ØØ35Ø SAVESP   LD      SP,$-$          ;P/u original SP
2416            ØØ36Ø          @@CKBRKC                ;Clear pending <BREAK>
2416 3E6A       ØØØØ3          LD      A,1Ø6
2418 EF         ØØØØ4          RST     4Ø
2419 C9         ØØ37Ø          RET                     ;Back to DOS
                ØØ38Ø ;
                ØØ39Ø ;        I/O Error Handler
                ØØ4ØØ ;
241A 3E2C       ØØ41Ø PRMERR   LD      A,PAR_ERR       ;Parameter Error
241C 6F         ØØ42Ø IOERR    LD      L,A             ;Error # to HL
241D 26ØØ       ØØ43Ø          LD      H,Ø
241F F6CØ       ØØ44Ø          OR      ØCØH            ;Abbrev, return
2421 4F         ØØ45Ø          LD      C,A
2422            ØØ46Ø          @@ERROR                 ;Display error
2422 3E1A       ØØØØ5          LD      A,26
2424 EF         ØØØØ6          RST     4Ø
2425 18EC       ØØ47Ø          JR      SAVESP          ;Exit
                ØØ48Ø ;
                ØØ49Ø ;        Internal Error Message Handler
                ØØ5ØØ ;
2427 213D27     ØØ51Ø NOMDSK   LD      HL,NOMDSK$      ;"Memdisk active
242A DD         ØØ52Ø          DB      ØDDH
242B 215B27     ØØ53Ø NORTES   LD      HL,NORTES$      ;"File routes active
242E DD         ØØ54Ø          DB      ØDDH
242F 219126     ØØ55Ø NOTFND   LD      HL,NOTFND$      ;"No config found
2432 DD         ØØ56Ø          DB      ØDDH
2433 212127     ØØ57Ø NOSPL    LD      HL,NOSPL$       ;"Spooler active
2436 DD         ØØ58Ø          DB      ØDDH
2437 21D626     ØØ59Ø NOINDO   LD      HL,NOINDO$      ;"JCL active
243A DD         ØØ6ØØ          DB      ØDDH
243B 21FD26     ØØ61Ø NOCANDO  LD      HL,NOCANDO$     ;"Sysgen inhibit set
243E CDØØØ5     ØØ62Ø          CALL    @LOGOT
```

```
2441 21FFFF   ØØ63Ø          LD      HL,-1           ;Set abort code
2444 18CD     ØØ64Ø          JR      SAVESP          ;Exit
              ØØ65Ø ;
              ØØ66Ø ;        SYSGEN1 - Save the state of the system
              ØØ67Ø ;
2446 117F27   ØØ68Ø SYSGEN1  LD      DE,PRMTBL$      ;Check for user parameters
2449          ØØ69Ø          @@PARAM
2449 3E11     ØØØØ7          LD      A,17
244B EF       ØØØØ8          RST     4Ø
244C C21A24   ØØ7ØØ          JP      NZ,PRMERR       ;Quit on parm error
244F          ØØ71Ø          @@FLAGS
244F 3E65     ØØØØ9          LD      A,1Ø1
2451 EF       ØØØ1Ø          RST     4Ø
              ØØ72Ø ;
              ØØ73Ø ;        Routine to process DRIVE parameter
              ØØ74Ø ;
2452 Ø1ØØØØ   ØØ75Ø DRIVE    LD      BC,Ø            ;Drive parm response
2455 79       ØØ76Ø          LD      A,C
2456 FEØ8     ØØ77Ø          CP      8               ;Drive in range?
2458 D21A24   ØØ78Ø          JP      NC,PRMERR       ;Go if > 7
245B F63Ø     ØØ79Ø          OR      'Ø'             ;Make it ASCII
245D 322A26   ØØ8ØØ          LD      (CFGSPEC),A     ;  & stuff drive spec
246Ø 327Ø26   ØØ81Ø          LD      (ATRSPEC),A
2463 CDEB25   ØØ82Ø          CALL    NPARM           ;Ck on yes or no entry
2466 CAB825   ØØ83Ø          JP      Z,CFGOFF        ;Jump if SYSGEN=OFF
2469 FDCB126E ØØ84Ø          BIT     5,(IY+'S'-'A')  ;Can't sysgen during DO
246D C23724   ØØ85Ø          JP      NZ,NOINDO
247Ø FDCBØ346 ØØ86Ø          BIT     Ø,(IY+'D'-'A')  ;Can't sysgen if spooler
2474 C23324   ØØ87Ø          JP      NZ,NOSPL        ;  is active
2477 FDCBØ366 ØØ88Ø          BIT     4,(IY+'D'-'A')  ;Can't sysgen if memdisk
247B C22724   ØØ89Ø          JP      NZ,NOMDSK       ;  is active
247E FDCBØ26E ØØ9ØØ          BIT     5,(IY+'C'-'A')  ;CK user sysgen inhibit
2482 C23B24   ØØ91Ø          JP      NZ,NOCANDO      ;Quit if set
2485 CDF525   ØØ92Ø          CALL    CKRTES          ;Ck if any route to file
2488 C22B24   ØØ93Ø          JP      NZ,NORTES       ;Can't if so
              ØØ94Ø ;
              ØØ95Ø ;        Initialize a CONFIG file on disk Ø, or user spec'ed drive
              ØØ96Ø ;
248B 111B26   ØØ97Ø          LD      DE,GENDCB       ;"CONFIG/SYS.CCC
248E 21ØØ29   ØØ98Ø          LD      HL,GENBUF       ;Config I/O buffer
2491 Ø6ØØ     ØØ99Ø          LD      B,Ø             ;Set LRL=256
2493          Ø1ØØØ          @@INIT                  ;Create config file
2493 3E3A     ØØØ11          LD      A,58
2495 EF       ØØØ12          RST     4Ø
2496 C21C24   Ø1Ø1Ø          JP      NZ,IOERR        ;Quit on init error
              Ø1Ø2Ø ;
              Ø1Ø3Ø ;        Generate the file header block
              Ø1Ø4Ø ;
2499 3EØ5     Ø1Ø5Ø          LD      A,5             ;Put file header
249B CD1326   Ø1Ø6Ø          CALL    PUTOUT
249E 3EØ6     Ø1Ø7Ø          LD      A,6             ;Put header length
24AØ CD1326   Ø1Ø8Ø          CALL    PUTOUT
24A3 Ø6Ø6     Ø1Ø9Ø          LD      B,6             ;Put header name
24A5 213B26   Ø11ØØ          LD      HL,CFGNAM$      ;"CONFIG"
24A8 7E       Ø111Ø WRNAM    LD      A,(HL)          ;Write name to
24A9 23       Ø112Ø          INC     HL              ;  config file header
24AA CD1326   Ø113Ø          CALL    PUTOUT
24AD 1ØF9     Ø114Ø          DJNZ    WRNAM
              Ø115Ø ;
              Ø116Ø ;        Generate disabling of all interrupts
              Ø117Ø ;
```

```
24AF 0601      01180         LD      B,1             ;Set block length
24B1 113800    01190         LD      DE,@RST38       ;Set load address
24B4 219425    01200         LD      HL,C9FLD        ;Point to block (X'C9')
24B7 CD9625    01210         CALL    WRBLCK          ;Write the block
               01220 ;
               01230 ;       Dump memory from (DVRHI$) to DVREND$
               01240 ;
24BA 2A0602    01250         LD      HL,(DVRHI$)     ;Point to max memory
24BD 11F40F    01260         LD      DE,DVREND$      ;Point to start
24C0 AF        01270         XOR     A
24C1 ED52      01280         SBC     HL,DE           ;How much to save
24C3 44        01290         LD      B,H             ;  into BC
24C4 4D        01300         LD      C,L
24C5 C45C25    01310         CALL    NZ,DUMP         ;File DVREND$ to top
               01320 ;
               01330 ;       Dump memory from HIGH$ to top-of-memory FIRST
               01340 ;
24C8 ED5B1C00  01350         LD      DE,(PHIGH$)     ;Point to max memory
24CC 210000    01360         LD      HL,0            ;Set up to get current
24CF 45        01370         LD      B,L             ;  HIGH$
24D0           01380         @@HIGH$
24D0 3E64      00013         LD      A,100
24D2 EF        00014         RST     40
24D3 EB        01390         EX      DE,HL           ;HIGH$ to DE,
24D4 AF        01400         XOR     A               ;  PHIGH to HL
24D5 ED52      01410         SBC     HL,DE           ;How much to save?
24D7 44        01420         LD      B,H
24D8 4D        01430         LD      C,L
24D9 13        01440         INC     DE              ;First byte in use
24DA C45C25    01450         CALL    NZ,DUMP         ;File HIGH$+1 to top
               01460 ;
               01470 ;       Generate all data from address table
               01480 ;
24DD 21A427    01490         LD      HL,ADRTBL$      ;Point to grab table
24E0 4E        01500 WRTBL   LD      C,(HL)          ;P/u lo order length
24E1 23        01510         INC     HL
24E2 46        01520         LD      B,(HL)          ;P/u hi order length
24E3 23        01530         INC     HL
24E4 78        01540         LD      A,B
24E5 B1        01550         OR      C
24E6 280B      01560         JR      Z,GENDCT        ;Exit on zero length
24E8 5E        01570         LD      E,(HL)          ;P/u lo order address
24E9 23        01580         INC     HL
24EA 56        01590         LD      D,(HL)          ;P/u hi order address
24EB 23        01600         INC     HL
24EC E5        01610         PUSH    HL              ;Save table pointer
24ED CD5C25    01620         CALL    DUMP            ;P/u the code & file it
24F0 E1        01630         POP     HL              ;Restore table pointer
24F1 18ED      01640         JR      WRTBL
               01650 ;
               01660 ;       Generate the DCT$ (offset)
               01670 ;
24F3 0650      01680 GENDCT  LD      B,80            ;Table is 80 bytes
24F5 110043    01690         LD      DE,MOD3BUF      ;Place to stuff DCT$
24F8 0E00      01700         LD      C,0
24FA           01710         @@GTDCT                 ;Write out the DCTs
24FA 3E51      00015         LD      A,81
24FC EF        00016         RST     40
24FD FDE5      01720         PUSH    IY              ;  handled by SYS0
24FF E1        01730         POP     HL
2500 CD9625    01740         CALL    WRBLCK
```

```
                    Ø175Ø ;
                    Ø176Ø ;       Generate the original interrupt vector JP
                    Ø177Ø ;
25Ø3 Ø6Ø1           Ø178Ø INTVEC  LD      B,1             ;File the interrupt
25Ø5 113ØØ          Ø179Ø         LD      DE,@RST38       ;  vector address
25Ø8 219525         Ø18ØØ         LD      HL,JPINST       ;  with a JP
25ØB CD9625         Ø181Ø         CALL    WRBLCK          ;Write the block
                    Ø182Ø ;
                    Ø183Ø ;       Generate the transfer address
                    Ø184Ø ;
25ØE 3EØ2           Ø185Ø         LD      A,2             ;Transfer address
251Ø CD1326         Ø186Ø         CALL    PUTOUT
2513 3EØ2           Ø187Ø         LD      A,2
2515 CD1326         Ø188Ø         CALL    PUTOUT
2518 AF             Ø189Ø         XOR     A               ;Xfer address is ØØØØ
2519 CD1326         Ø19ØØ         CALL    PUTOUT
251C AF             Ø191Ø         XOR     A
251D CD1326         Ø192Ø         CALL    PUTOUT
252Ø                Ø193Ø         @@CLOSE                 ;Close config file
252Ø 3E3C           ØØØ17         LD      A,6Ø
2522 EF             ØØØ18         RST     4Ø
2523 C21C24         Ø194Ø         JP      NZ,IOERR        ;Quit on close error
                    Ø195Ø ;
                    Ø196Ø ;       Let the config sector know there's a CONFIG/SYS
                    Ø197Ø ;
                    Ø198Ø         IF      @MOD2
                    Ø199Ø ;
                    Ø2ØØØ         LD      A,(DRIVE+1)     ;Drive #
                    Ø2Ø1Ø         LD      C,A             ;Pass drive #
                    Ø2Ø2Ø         LD      HL,SBUFF$       ;I/O buffer
2526                Ø2Ø3Ø         @@GTDCT                 ;Fetch DCT
                    ØØØ19         LD      A,81
                    ØØØ2Ø         RST     4Ø
                    Ø2Ø4Ø ;
                    Ø2Ø5Ø         LD      A,(IY+3)        ;Get DCT data
                    Ø2Ø6Ø         AND     28H             ;Bit 5/3
                    Ø2Ø7Ø         CP      2ØH             ;8" floppy?
                    Ø2Ø8Ø         JR      NZ,SETSYS1      ;Go if not
                    Ø2Ø9Ø         LD      A,(IY+4)        ;Get DCT data
                    Ø21ØØ         AND     5ØH             ;Bit 6/4
                    Ø211Ø         CP      4ØH             ;DD and not alien?
                    Ø212Ø         JR      NZ,SETSYS1      ;Go if not
                    Ø213Ø ;
                    Ø214Ø         LD      D,(IY+9)        ;Get GAT address
                    Ø215Ø         LD      E,Ø             ;DE=>gat
2526                Ø216Ø         @@RDSEC                 ;Read sector
                    ØØØ21         LD      A,49
                    ØØØ22         RST     4Ø
                    Ø217Ø         CP      6               ;Directory?
                    Ø218Ø         JP      NZ,IOERR        ;Go on disk error
                    Ø219Ø         LD      A,(SBUFF$+ØCDH) ;Get gat data byte
                    Ø22ØØ         BIT     7,A             ;Data disk?
                    Ø221Ø SETSYS1 LD      D,Ø             ;Init sys info sector
                    Ø222Ø         JR      NZ,$+3          ;Go if not 8" floppy sys
                    Ø223Ø         INC     D               ;  else sysinfo on cyl 1
                    Ø224Ø         LD      E,2             ;Sysinfo sector
                    Ø225Ø ;
                    Ø226Ø         ENDIF
                    Ø227Ø ;
2526 21ØØ1D         Ø228Ø         LD      HL,SBUFF$       ;Use this as disk buffer
                    Ø229Ø ;
```

```
                    02300           IF      @MOD4
2529 110200         02310           LD      DE,0<8+2        ;Trk 0, sect 2 (SIS)
                    02320           ENDIF
                    02330 ;
252C 3A5324         02340           LD      A,(DRIVE+1)     ;P/u logical drive #
252F 4F             02350           LD      C,A             ;  of drive (DRIVE)
2530                02360           @@RDSEC                 ;Read System Info Sector
2530 3E31           00023           LD      A,49
2532 EF             00024           RST     40
2533 C21C24         02370           JP      NZ,IOERR        ;Quit on read error
2536 AF             02380           XOR     A               ;Turn on standard config
2537 32011D         02390           LD      (SBUFF$+1),A
253A                02400           @@WRSEC                 ;Write it back
253A 3E35           00025           LD      A,53
253C EF             00026           RST     40
253D C21C24         02410           JP      NZ,IOERR        ;Quit on write error
2540 2EC0           02420           LD      L,0C0H          ;Pt to SYS/DATA byte
2542 B6             02430           OR      (HL)            ;Check if System disk
2543 21AD26         02440           LD      HL,WARN$        ;  and inform in not
2546 2003           02450           JR      NZ,$+5
2548                02460           @@LOGOT
                    00027           IFEQ    00H,1
                    00028           LD      HL,
                    00029           ENDIF
2548 3E0C           00030           LD      A,12
254A EF             00031           RST     40
                    02470 ;
                    02480 ;         Let user know it's done
                    02490 ;
254B                02500           @@LOGOT CFGBLT          ;Log completion
                    00032           IFEQ    01H,1
254B 214126         00033           LD      HL,CFGBLT
                    00034           ENDIF
254E 3E0C           00035           LD      A,12
2550 EF             00036           RST     40
2551 215A26         02510           LD      HL,ATRBIT       ;Make CONFIG invisible
2554 3E18           02520           LD      A,24            ;Init for CMNDI
2556 C32800         02530           JP      RST28           ;  & execute
2559                02540           @@CMNDI
2559 3E18           00037           LD      A,24
255B EF             00038           RST     40
                    02550 ;
                    02560 ;         Routine to DUMP core
                    02570 ;
255C EB             02580 DUMP      EX      DE,HL           ;Load address to HL
255D 111B26         02590           LD      DE,GENDCB       ;Config/sys FCB
2560 E5             02600 DUMP1     PUSH    HL
2561 C5             02610           PUSH    BC              ;Save block length
2562 60             02620           LD      H,B             ;Length of this block
2563 69             02630           LD      L,C             ;  to HL
2564 01FE00         02640           LD      BC,254          ;Write block in groups
2567 AF             02650           XOR     A               ;Of 254 bytes max
2568 ED42           02660           SBC     HL,BC           ;Sub blk len fm dump len
256A 3006           02670           JR      NC,DUMP2        ;Go if write len > 254
256C C1             02680           POP     BC              ;Recover block len
256D 210000         02690           LD      HL,0            ;Show block written
2570 1801           02700           JR      DUMP3
2572 F1             02710 DUMP2     POP     AF              ;Remove old len from stack
2573 E3             02720 DUMP3     EX      (SP),HL         ;Xchg len w/address
2574 41             02730           LD      B,C
2575 3E01           02740           LD      A,1             ;Start of block
```

```
2577 CD1326   Ø2750          CALL    PUTOUT
257A 78       Ø2760          LD      A,B
257B C6Ø2     Ø2770          ADD     A,2             ;Add for address
257D CD1326   Ø2780          CALL    PUTOUT          ;Write block length
258Ø 7D       Ø2790          LD      A,L
2581 CD1326   Ø2800          CALL    PUTOUT          ;Write lo order address
2584 7C       Ø2810          LD      A,H
2585 CD1326   Ø2820          CALL    PUTOUT          ;Write hi order address
2588 7E       Ø2830 DUMP4    LD      A,(HL)
2589 23       Ø2840          INC     HL
258A CD1326   Ø2850          CALL    PUTOUT          ;Write the block
258D 1ØF9     Ø2860          DJNZ    DUMP4
258F C1       Ø2870          POP     BC              ;Get remaining length
2590 78       Ø2880          LD      A,B
2591 B1       Ø2890          OR      C
2592 2ØCC     Ø2900          JR      NZ,DUMP1        ;Loop if block not done
2594 C9       Ø2910 C9FLD    RET                     ;  else go back
              Ø2920 ;
              Ø2930 ;        Routine to write an offset block
              Ø2940 ;
2595 C3       Ø2950 JPINST   DB      ØC3H            ;JP instruction
2596 E5       Ø2960 WRBLCK   PUSH    HL              ;Save real load address
2597 62       Ø2970          LD      H,D             ;HL = offset address
2598 6B       Ø2980          LD      L,E
2599 111B26   Ø2990          LD      DE,GENDCB       ;Config/sys FCB
259C 3EØ1     Ø3000          LD      A,1             ;Start of block
259E CD1326   Ø3010          CALL    PUTOUT
25A1 78       Ø3020          LD      A,B
25A2 C6Ø2     Ø3030          ADD     A,2             ;Adj for address
25A4 CD1326   Ø3040          CALL    PUTOUT          ;Write block length
25A7 7D       Ø3050          LD      A,L
25A8 CD1326   Ø3060          CALL    PUTOUT          ;Write lo order load
25AB 7C       Ø3070          LD      A,H
25AC CD1326   Ø3080          CALL    PUTOUT          ;Write hi order load
25AF E1       Ø3090          POP     HL              ;Recover real address
25BØ 7E       Ø3100 WRBLK1   LD      A,(HL)          ;  and p/u data from there
25B1 23       Ø3110          INC     HL
25B2 CD1326   Ø3120          CALL    PUTOUT          ;Write the block
25B5 1ØF9     Ø3130          DJNZ    WRBLK1
25B7 C9       Ø3140          RET
              Ø3150 ;
              Ø3160 ;        Perform SYSGEN OFF
              Ø3170 ;
25B8 21ØØ1D   Ø3180 CFGOFF   LD      HL,SBUFF$       ;Set disk I/O buffer
              Ø3190 ;
              Ø3200          IF      @MOD2
              Ø3210 ;
              Ø3220          LD      A,(DRIVE+1)     ;Get drive
              Ø3230          LD      C,A             ;Pass to C
25BB          Ø3240          @@GTDCT                 ;Fetch DCT
              ØØØ39          LD      A,81
              ØØØ4Ø          RST     4Ø
              Ø3250          LD      D,(IY+9)        ;Dir cylinder
              Ø3260 ;
              Ø3270          LD      A,(IY+3)        ;Get dct data
              Ø3280          AND     28H             ;Bit 5/3
              Ø3290          CP      2ØH             ;8" floppy?
              Ø3300          JR      NZ,SETSYS2      ;Go if not
              Ø3310          LD      A,(IY+4)        ;Get data
              Ø3320          AND     5ØH             ;Bit 6/4
              Ø3330          CP      4ØH             ;DD not alien?
```

```
                    03340           JR      NZ,SETSYS2      ;Go if not
                    03350 ;
                    03360           LD      E,0             ;GAT table
25BB                03370           @@RDSEC                 ;Read sector
                    00041           LD      A,49
                    00042           RST     40
                    03380           CP      6               ;Directory read?
                    03390           JP      NZ,IOERR        ;Go if not
                    03400           LD      A,(SBUFF$+0CDH) ;Get gat data byte
                    03410           BIT     7,A             ;System disk?
                    03420 SETSYS2   LD      D,0             ;Cyl 0 if not
                    03430           JR      NZ,$+3          ;Go if not
                    03440           INC     D               ;Else on cyl 1
                    03450           LD      E,2             ;Sysinfo sector
                    03460 ;
                    03470           ENDIF
                    03480 ;
                    03490           IF      @MOD4
25BB 110200         03500           LD      DE,0<8+2        ;Trk 0, sect 2 (SIS)
                    03510           ENDIF
                    03520 ;
25BE 3A5324         03530           LD      A,(DRIVE+1)     ;P/u drive number
25C1 4F             03540           LD      C,A             ;  & note
25C2                03550           @@RDSEC                 ;  that no config is
25C2 3E31           00043           LD      A,49
25C4 EF             00044           RST     40
25C5 C21C24         03560           JP      NZ,IOERR        ;  on the disk
25C8 3EC9           03570           LD      A,0C9H
25CA 32011D         03580           LD      (SBUFF$+1),A    ;A X'C9' indicates
25CD                03590           @@WRSEC                 ;  no config
25CD 3E35           00045           LD      A,53
25CF EF             00046           RST     40
25D0 C21C24         03600           JP      NZ,IOERR        ;Quit if write error
                    03610 ;
                    03620 ;         Now that none is shown, kill the CONFIG/SYS
                    03630 ;
25D3 111B26         03640           LD      DE,GENDCB       ;Pt to FCB
25D6 0600           03650           LD      B,0
25D8                03660           @@OPEN                  ;Try to open the config
25D8 3E3B           00047           LD      A,59
25DA EF             00048           RST     40
25DB C22F24         03670           JP      NZ,NOTFND       ;Jump if not there
25DE                03680           @@REMOV                 ;Kill the config/sys
25DE 3E39           00049           LD      A,57
25E0 EF             00050           RST     40
25E1 C21C24         03690           JP      NZ,IOERR        ;Quit if can't be killed
25E4                03700           @@LOGOT CFGDEL$         ;"config deleted..
                    00051           IFEQ    01H,1
25E4 217626         00052           LD      HL,CFGDEL$
                    00053           ENDIF
25E7 3E0C           00054           LD      A,12
25E9 EF             00055           RST     40
25EA C9             03710           RET                     ;Done - return
                    03720 ;
                    03730 ;         Parameter parsing of yes/no
                    03740 ;
25EB 010000         03750 NPARM     LD      BC,0            ;P/u "no" parm entry
25EE 78             03760           LD      A,B
25EF B1             03770           OR      C               ;If NO, make Z
25F0 EEFF           03780           XOR     0FFH
25F2 C9             03790           RET
```

```
25F3 ØØØØ      Ø38ØØ YPARM   DW      Ø
               Ø381Ø ;
               Ø382Ø ;       Any devices routed to files?
               Ø383Ø ;
25F5 114B49    Ø384Ø CKRTES  LD      DE,'IK'          ;Point to begin of area
25F8           Ø385Ø         @@GTDCB
25F8 3E52      ØØØ56         LD      A,82
25FA EF        ØØØ57         RST     4Ø
25FB E5        Ø386Ø CKRTØ   PUSH    HL               ;Save pointer
25FC CB66      Ø387Ø CKRT1   BIT     4,(HL)           ;Routed device?
25FE 28ØB      Ø388Ø         JR      Z,CKRT2          ;Jump if not
26ØØ 2C        Ø389Ø         INC     L                ;Bypass TYPE code
26Ø1 7E        Ø39ØØ         LD      A,(HL)           ;P/u route vector
26Ø2 2C        Ø391Ø         INC     L                ;  lo-order
26Ø3 66        Ø392Ø         LD      H,(HL)           ;P/u vector hi-order
26Ø4 6F        Ø393Ø         LD      L,A
26Ø5 CB7E      Ø394Ø         BIT     7,(HL)           ;Routed to a file?
26Ø7 28F3      Ø395Ø         JR      Z,CKRT1          ;Ck further route if not
26Ø9 E1        Ø396Ø         POP     HL               ;  else exit with NZ
26ØA C9        Ø397Ø         RET
               Ø398Ø ;
               Ø399Ø ;       Point to next device
               Ø4ØØØ ;
26ØB E1        Ø4Ø1Ø CKRT2   POP     HL               ;Recover DCB pointer
26ØC 7D        Ø4Ø2Ø         LD      A,L              ;Advance to next DCB area
26ØD C6Ø8      Ø4Ø3Ø         ADD     A,8              ;Loop through all
26ØF 6F        Ø4Ø4Ø         LD      L,A              ;  devices while checking
261Ø 2ØE9      Ø4Ø5Ø         JR      NZ,CKRTØ         ;Loop until table end
2612 C9        Ø4Ø6Ø         RET
               Ø4Ø7Ø ;
2613 4F        Ø4Ø8Ø PUTOUT  LD      C,A              ;Byte into C
2614           Ø4Ø9Ø         @@PUT                    ;Do the put
2614 3EØ4      ØØØ58         LD      A,4
2616 EF        ØØØ59         RST     4Ø
2617 C8        Ø41ØØ         RET     Z                ;Back if good
2618 C31C24    Ø411Ø         JP      IOERR            ;  else abort
               Ø412Ø ;
261B 43        Ø413Ø GENDCB  DB      'CONFIG/SYS.CCC:Ø',Ø
     4F 4E 46 49 47 2F 53 59
     53 2E 43 43 43 3A 3Ø ØØ
262A           Ø414Ø CFGSPEC EQU     $-2
ØØØF           Ø415Ø         DS      32-$+GENDCB
263B 43        Ø416Ø CFGNAM$ DB      'CONFIG'
     4F 4E 46 49 47
2641 55        Ø417Ø CFGBLT  DB      'User configuration built',CR
     73 65 72 2Ø 63 6F 6E 66
     69 67 75 72 61 74 69 6F
     6E 2Ø 62 75 69 6C 74 ØD
265A 41        Ø418Ø ATRBIT  DB      'Attrib CONFIG/SYS.CCC:d (I)',CR
     74 74 72 69 62 2Ø 43 4F
     4E 46 49 47 2F 53 59 53
     2E 43 43 43 3A 64 2Ø 28
     49 29 ØD
267Ø           Ø419Ø ATRSPEC EQU     $-6
2676 55        Ø42ØØ CFGDEL$ DB      'User configuration deleted',CR
     73 65 72 2Ø 63 6F 6E 66
     69 67 75 72 61 74 69 6F
     6E 2Ø 64 65 6C 65 74 65
     64 ØD
2691 4E        Ø421Ø NOTFND$ DB      'No user configuration found',CR
     6F 2Ø 75 73 65 72 2Ø 63
```

```
     6F 6E 66 69 67 75 72 61
     74 69 6F 6E 2Ø 66 6F 75
     6E 64 ØD
26AD 57       Ø422Ø WARN$   DB      'Warning: Target drive contains no system',CR
     61 72 6E 69 6E 67 3A 2Ø
     54 61 72 67 65 74 2Ø 64
     72 69 76 65 2Ø 63 6F 6E
     74 61 69 6E 73 2Ø 6E 6F
     2Ø 73 79 73 74 65 6D ØD
26D6 43       Ø423Ø NOINDO$ DB      'Command invalid during <DO> processing',CR
     6F 6D 6D 61 6E 64 2Ø 69
     6E 76 61 6C 69 64 2Ø 64
     75 72 69 6E 67 2Ø 3C 44
     4F 3E 2Ø 7Ø 72 6F 63 65
     73 73 69 6E 67 ØD
26FD 2A       Ø424Ø NOCANDO$        DB      '** SYSGEN inhibited at this time **',CR
     2A 2Ø 53 59 53 47 45 4E
     2Ø 69 6E 68 69 62 69 74
     65 64 2Ø 61 74 2Ø 74 68
     69 73 2Ø 74 69 6D 65 2Ø
     2A 2A ØD
2721 43       Ø425Ø NOSPL$  DB      'Can''t while SPOOL is active',CR
     61 6E 27 74 2Ø 77 68 69
     6C 65 2Ø 53 5Ø 4F 4F 4C
     2Ø 69 73 2Ø 61 63 74 69
     76 65 ØD
273D 43       Ø426Ø NOMDSK$ DB      'Can''t while MEMDISK is active',CR
     61 6E 27 74 2Ø 77 68 69
     6C 65 2Ø 4D 45 4D 44 49
     53 4B 2Ø 69 73 2Ø 61 63
     74 69 76 65 ØD
275B 43       Ø427Ø NORTES$ DB      'Can''t while route-to-file is active',CR
     61 6E 27 74 2Ø 77 68 69
     6C 65 2Ø 72 6F 75 74 65
     2D 74 6F 2D 66 69 6C 65
     2Ø 69 73 2Ø 61 63 74 69
     76 65 ØD
             Ø428Ø ;
277F 8Ø       Ø429Ø PRMTBL$ DB      8ØH,42H,'ON',Ø
     42 4F 4E ØØ
2784 F325     Ø43ØØ         DW      YPARM
2786 43       Ø431Ø         DB      43H,'OFF',Ø
     4F 46 46 ØØ
278B EC25     Ø432Ø         DW      NPARM+1
278D 53       Ø433Ø         DB      53H,'YES',Ø
     59 45 53 ØØ
2792 F325     Ø434Ø         DW      YPARM
2794 52       Ø435Ø         DB      52H,'NO',Ø
     4E 4F ØØ
2798 EC25     Ø436Ø         DW      NPARM+1
279A 95       Ø437Ø         DB      95H,'DRIVE',Ø
     44 52 49 56 45 ØØ
27A1 5324     Ø438Ø         DW      DRIVE+1
27A3 ØØ       Ø439Ø         NOP
             Ø44ØØ ;
             Ø441Ø ;       Table of regions to sysgen
             Ø442Ø ;
27A4 Ø2ØØ     Ø443Ø ADRTBL$ DW      2,HIGH$         ;Save HIGH$
     ØE04
27A8 Ø2ØØ     Ø444Ø         DW      2,LOW$          ;Lowest user address
     1EØØ
```

```
27AC 0100 04450          DW      1,TIMSL$        ;Time slice
     2B00
27B0 FA00 04460          DW      250,DVRHI$      ;Save primary DCBs
     0602
27B4 0300 04470          DW      3,EXTDBG$       ;Ext DEBUG vector & DBGHK
     A419
27B8 3200 04480          DW      50,INTIM$       ;Table & TCBs
     3C00
27BC 2200 04490          DW      34,FLGTAB$      ;Flag table & assorted
     6A00
27C0 0200 04500          DW      2,75*2+SVCTAB$  ;Save WRITE vector
     9601
27C4 0300 04510          DW      3,HKRES$        ;Sys overlay hook
     6C1A
27C8 0200 04520          DW      2,KIDATA$+2
     FE08
27CC 0100 04530          DW      1,DODATA$
     940B
27D0 0200 04540          DW      2,DODATA$+3
     970B
27D4 1000 04550          DW      16,240+SVCTAB$  ;SVCs 120 - 127
     F001
27D8 1000 04560          DW      16,@RST08       ;RST zones 8 and 10
     0800
27DC 0100 04570          DW      1,HERTZ$        ;Hertz rate for timer
     5007
               04580 ;
               04590          IF      @MOD2
               04600          DW      3,$CRSCHAR      ;Cursor char + column siz
               04610          ENDIF
               04620          IF      @MOD4
27E0 0000 04630          DW      0,0
     0000
               04640          ENDIF
               04650 ;
27E4 0000 04660          DW      0,0
     0000
27E8 0000 04670          DW      0,0
     0000
27EC 0000 04680          DW      0,0
     0000
27F0 0000 04690          DW      0,0
     0000
27F4 0000 04700          DW      0,0
     0000
27F8 0000 04710          DW      0,0
     0000
27FC 0000 04720          DW      0,0
     0000
2800 0000 04730          DW      0               ;End of table
2802 00   04740          DC      20,0            ;Patch space
     00 00 00 00 00 00 00 00
     00 00 00 00 00 00 00 00
     00 00 00
               04750 ;
2900           04760          ORG     $<-8+1<+8
2900           04770 GENBUF   EQU     $
               04780 ;
2400           04790          END     SYSGEN
```

$A1	Ø3B7	$A2	Ø3B8
$CKEOF	147Ø	@$SYS	Ø8FØ
@@2	ØØØØ	@@3	ØØØØ
@ABORT	1BØ8	@ADTSK	1CDA
@BKSP	1486	@BREAK	196F
@CHNIO	Ø689	@CKBRKC	Ø553
@CKEOF	158F	@CKTSK	1CF5
@CLS	Ø545	@CMNDI	197E
@CTL	Ø623	@DATE	Ø7A8
@DCINIT	19CØ	@DCRES	19C4
@DCTBYT	1A2B	@DEBUG	19AØ
@DIRCYL	18F7	@DIRRD	18BB
@DIV16	Ø6E3	@DIV8	1927
@DOKEY	19A9	@DSP	Ø642
@ERROR	1BØF	@EXIT	1BØB
@FLAGS	196A	@FNAME	199C
@FSPEC	1981	@GATRD	1874
@GERMAN	ØØØØ	@GET	Ø638
@GTDCT	1A1E	@GTMOD	19B2
@HEX16	Ø7BD	@HEX8	Ø7C2
@HIGH$	1948	@HITRD	1897
@HZ5Ø	ØØØØ	@ICNFG	ØØ86
@INTL	ØØØØ	@IPL	1BF2
@KBD	Ø635	@KEY	Ø628
@KITSK	ØØ89	@KLTSK	1CDØ
@LOC	14B3	@LOF	14DE
@LOGOT	Ø5ØØ	@MOD2	ØØØØ
@MSG	Ø53Ø	@MUL16	Ø6C9
@NMI	ØØ66	@OPEN	198A
@PARAM	1987	@PAUSE	Ø382
@POSN	1434	@PRINT	Ø528
@PUT	Ø645	@RAMDIR	19AC
@RDSEC	19F4	@RDSSC	18D8
@READ	1513	@REMOVE	19A6
@REW	149B	@RMTSK	1CD7
@RREAD	1473	@RSLCT	19D4
@RSTØ8	ØØØ8	@RST1Ø	ØØ1Ø
@RST2Ø	ØØ2Ø	@RST28	ØØ28
@RST38	ØØ38	@RSTNMI	ØFE9
@RSTREG	Ø68Ø	@RUN	1B1D
@SEEK	19DØ	@SEEKSC	1421
@SLCT	19BC	@SOUND	Ø392
@TIME	Ø78D	@USA	FFFF
@VDCTL3	ØD38	@VER	156Ø
@WEOF	14EC	@WHERE	1979
@WRSEC	19E8	@WRSSC	19EC
@_VDCTL	ØD42	ADDR_2_ROWCOL	ØDF1
AFLAG$	ØØ6A	ATRBIT	265A
AUTO?	1FF1	BAR$	Ø2Ø1
BOOTST$	439D	BREAK?	1C6Ø
BUR$	Ø2ØØ	C9FLD	2594
CFCB$	ØØEØ	CFGBLT	2641
CFGFCB$	ØØEØ	CFGNAM$	263B
CFGSPEC	262A	CFLAG$	ØØ6C
CKOPEN@	1568	CKRTØ	25FB
CKRT2	26ØB	CKRTES	25F5
CORE$	Ø3ØØ	CR	ØØØD
CYL_GRN	16AE	D@FBYT8	1A26
DAYTBL$	Ø4C7	DBGSV$	ØØAØ
DCT$	Ø47Ø	DCTBYT8@	1A29

1CDA	@BANK	Ø877	
196F	@BYTEIO	13ØØ	
Ø553	@CKDRV	1993	
1CF5	@CLOSE	1999	
197E	@CMNDR	197B	
Ø7A8	@DBGHK	199F	
19C4	@DCSTAT	19B5	
19AØ	@DECHEX	Ø3E1	
18BB	@DIRWR	18Ø3	
1927	@DODIR	19AF	
Ø642	@DSPLY	Ø52D	
1BØB	@FEXT	1984	
199C	@FRENCH	ØØØØ	
1874	@GATWR	1875	
Ø638	@GTDCB	199Ø	
19B2	@HDFMT	19E4	
Ø7C2	@HEXDEC	Ø6F6	
1897	@HITWR	1898	
ØØ86	@INIT	198D	
1BF2	@JCL	Ø63Ø	
Ø628	@KEYIN	Ø585	
1CDØ	@LOAD	1B38	
14DE	@LOGER	Ø5Ø3	
ØØØØ	@MOD4	FFFF	
Ø6C9	@MUL8	19ØA	
198A	@OPREG	ØØ84	
Ø382	@PEOF	14A2	
Ø528	@PRT	Ø63D	
19AC	@RDHDR	19D8	
18D8	@RDTRK	19EØ	
19A6	@RENAME	1996	
1CD7	@RPTSK	1CEB	
19D4	@RSTØØ	ØØØØ	
ØØ1Ø	@RST18	ØØ18	
ØØ28	@RST3Ø	ØØ3Ø	
ØFE9	@RSTOR	19C8	
1B1D	@RWRIT	13AD	
1421	@SKIP	143Ø	
Ø392	@STEPI	19CC	
FFFF	@VDCTL	ØB99	
156Ø	@VRSEC	19DC	
1979	@WRITE	1531	
19EC	@WRTRK	19FØ	
ØDF1	ADRTBL$	27A4	
265A	ATRSPEC	267Ø	
Ø2Ø1	BEGINA	24Ø9	
1C6Ø	BRKVEC$	1C88	
2594	CASHK$	ØA7B	
2641	CFGDEL$	2676	
263B	CFGOFF	25B8	
ØØ6C	CKMOD@	1A7F	
25FB	CKRT1	25FC	
25F5	CONFIG$	2Ø3F	
ØØØD	CRTBGN$	F8ØØ	
1A26	DATE$	ØØ33	
ØØAØ	DCBKL$	ØØ31	
1A29	DCTFLD@	1A34	

DFLAG$	ØØ6D	DIRBUF$	23ØØ	DIS_DO_RAM	Ø846
DODATA$	ØB94	DODCB$	Ø21Ø	DO_CONTROL	ØC44
DO_DSPCHAR	ØCB8	DO_INVERT_DIS	ØC8C	DO_INVERT_ENA	ØC89
DO_INVERT_OFF	ØC9B	DO_MASK	ØØØØ	DO_RET	ØBCB
DO_RET1	ØBCC	DO_SCROLL	ØCCE	DO_TABS	ØBEA
DRIVE	2452	DSKTYP$	Ø4CØ	DTPMT$	Ø4C2
DUMP	255C	DUMP1	256Ø	DUMP2	2572
DUMP3	2573	DUMP4	2588	DVREND$	ØFF4
DVRHI$	Ø2Ø6	EFLAG$	ØØ6E	ENADIS_DO_RAM	Ø817
EXIT	241Ø	EXTDBG$	19A4	FDDINT$	ØØØE
FEMSK$	ØØ6F	FLGTAB$	ØØ6A	GENBUF	29ØØ
GENDCB	261B	GENDCT	24F3	GET_@_ROWCOL	ØDAE
HERTZ$	Ø75Ø	HIGH$	Ø4ØE	HKRES$	1A6C
IFLAG$	ØØ72	INBUF$	Ø42Ø	INTIM$	ØØ3C
INTMSK$	ØØ3D	INTVC$	ØØ3E	INTVEC	25Ø3
IOERR	241C	JCLCB$	Ø2Ø3	JDCB$	ØØ24
JFCB$	ØØCØ	JLDCB$	Ø23Ø	JPINST	2595
JRET$	ØØ26	KCK@	Ø7D6	KFLAG$	ØØ74
KIDATA$	Ø8FC	KIDCB$	Ø2Ø8	LBANK$	Ø2Ø2
LDRV$	ØØ23	LFLAG$	ØØ75	LNKFCB@	1566
LOW$	ØØ1E	LSVC$	ØØØD	MAXCOR$	24ØØ
MAXDAY$	Ø4Ø1	MINCOR$	3ØØØ	MOD3BUF	43ØØ
MODOUT$	ØØ76	MONTBL$	Ø4DC	NFLAG$	ØØ77
NOCANDO	243B	NOCANDO$	26FD	NOINDO	2437
NOINDO$	26D6	NOMDSK	2427	NOMDSK$	273D
NORTES	242B	NORTES$	275B	NOSPL	2433
NOSPL$	2721	NOTFND	242F	NOTFND$	2691
NPARM	25EB	OPREG$	ØØ78	OPREG_SV_AREA	Ø86E
OPREG_SV_PTR	Ø835	ORARET@	14DC	OSRLS$	ØØ3B
OSVER$	ØØ85	OVRLY$	ØØ69	PAKNAM$	Ø41Ø
PAR_ERR	ØØ2C	PAUSE@	Ø382	PCSAVE$	Ø7AF
PDRV$	ØØ1B	PHIGH$	ØØ1C	PRDCB$	Ø218
PRMERR	241A	PRMTBL$	277F	PUTA@DE	ØDCD
PUTOUT	2613	PUT_@	ØDCA	PUT_@_ROWCOL	ØDC6
RFLAG$	ØØ7B	ROWCOL_2_ADDR	ØDDØ	RST28	ØØ28
RST38@	1BFF	RSTOR$	Ø4C4	RWRIT@	13A2
S1DCB$	Ø238	SAVESP	2413	SBUFF$	1DØØ
SET@EXEC	1A79	SET_SCROLL	ØCF3	SFCB$	ØØ8C
SFLAG$	ØØ7C	SIDCB$	Ø22Ø	SODCB$	Ø228
SPACE4$	2142	STACK$	Ø38Ø	START$	ØØØØ
SVCRET$	ØØØB	SVCTAB$	Ø1ØØ	SYSERR$	1B13
SYSGEN	24ØØ	SYSGEN1	2446	TCB$	ØØ4E
TFLAG$	ØØ7D	TIME$	ØØ2D	TIMER$	ØØ2C
TIMSL$	ØØ2B	TIMTSK$	Ø713	TMPMT$	Ø4C3
TRACE_INT	Ø7B1	TYPHK$	ØA8F	TYPTSK$	ØB26
USTOR$	ØØ13	VFLAG$	ØØ7F	WARN$	26AD
WRBLCK	2596	WRBLK1	25BØ	WRINT$	ØØ8Ø
WRNAM	24A8	WRTBL	24EØ	YPARM	25F3
ZERO$	Ø4Ø1	ZEROA@	13AØ	@@ABORT	9749
@@ADTSK	97DC	@@BANK	9CF4	@@BKSP	99D4
@@BREAK	9DØA	@@CHNIO	9734	@@CKBRKC	9D58
@@CKDRV	983Ø	@@CKEOF	99E9	@@CKTSK	97C7
@@CLOSE	99BF	@@CLS	9D42	@@CMNDI	9773
@@CMNDR	9788	@@CTL	9598	@@DATE	97ØA
@@DCSTAT	986F	@@DEBUG	97B2	@@DECHEX	9C74
@@DIRRD	9BE1	@@DIRWR	9BF6	@@DIV16	9C5F
@@DIV8	9C4A	@@DODIR	9845	@@DSP	955C
@@DSPLY	95FC	@@ERROR	979D	@@EXIT	975E
@@FEXT	9B4E	@@FLAGS	9CDE	@@FNAME	9B63
@@FSPEC	9B39	@@GATRD	9BCC	@@GATWR	9CØB
@@GET	957Ø	@@GTDCB	9B8D	@@GTDCT	9B78

@@GTMOD	9BA2	@@HDFMT	9917	@@HEX16	9CB3		
@@HEX8	9C9E	@@HEXDEC	9C89	@@HIGH$	9CC8		
@@INIT	9995	@@KBD	95D4	@@KEY	9548		
@@KEYIN	95E8	@@KLTSK	981B	@@LOAD	9BØF		
@@LOC	99FE	@@LOF	9A13	@@LOGER	9633		
@@LOGOT	9648	@@MSG	967F	@@MUL16	9C35		
@@MUL8	9C2Ø	@@OPEN	99AA	@@PARAM	96F5		
@@PAUSE	96EØ	@@PEOF	9A28	@@POSN	9A3D		
@@PRINT	9694	@@PRT	95AC	@@PUT	9584		
@@RAMDIR	985A	@@RDSEC	98ED	@@RDSSC	9BB7		
@@READ	9A52	@@REMOV	998Ø	@@RENAM	996B		
@@REW	9A67	@@RMTSK	97F1	@@RPTSK	98Ø6		
@@RREAD	9A7C	@@RSLCT	98D8	@@RSTOR	9899		
@@RUN	9B24	@@RWRIT	9A91	@@SEEK	98C3		
@@SEEKSC	9AA6	@@SKIP	9ABB	@@SLCT	9884		
@@STEPI	98AE	@@TIME	971F	@@VDCTL	96CB		
@@VER	9ADØ	@@VRSEC	99Ø2	@@WEOF	9AE5		
@@WHERE	95CØ	@@WRITE	9AFA	@@WRSEC	992C		
@@WRSSC	9941	@@WRTRK	9956				

2ØØØ is the transfer address
ØØØØØ Total errors

NOTES:

NOTES:

Command: SYSTEM

Library: SYS8/SYS

ISAM # : A1H

```
ØØØØ            ØØ1ØØ *GET    LBSYSTEM:3
                ØØØ1Ø ;LBSYSTEM/ASM - SYSTEM Command
ØØØØ            ØØØ2Ø         TITLE   <SYSTEM - LS-DOS 6.2>
                ØØØ3Ø ;
ØØEC            ØØØ4Ø MODPORT EQU     ØECH
ØØØD            ØØØ5Ø CR      EQU     13
                ØØØ6Ø ;
ØØØØ            ØØØ7Ø *GET    SVCMAC:3                ;SVC Macro equivalents
                ØØØ8Ø ;SVCMAC/ASM - LS-DOS Version VI
                ØØØ9Ø *LIST   OFF
                Ø397Ø *LIST   ON
                Ø399Ø *LIST   OFF                     ;Get SYSØ/EQU
                Ø7Ø1Ø *LIST   ON
                Ø7Ø2Ø ;
2400            Ø7Ø3Ø         ORG     2400H
                Ø7Ø4Ø ;
                Ø7Ø5Ø SYSTEM
2400            Ø7Ø6Ø         @@CKBRKC                ;Check for <BREAK>
2400 3E6A       ØØØØ1         LD      A,1Ø6
2402 EF         ØØØØ2         RST     4Ø
2403 2825       Ø7Ø7Ø         JR      Z,SYSTEM1       ;Go if not
2405 21FFFF     Ø7Ø8Ø         LD      HL,-1           ;  else abort
2408 C9         Ø7Ø9Ø         RET
                Ø71ØØ ;
2409 31ØØØØ    Ø711Ø SAVESP  LD      SP,$-$          ;P/u original stack
240C            Ø712Ø         @@CKBRKC                ;Clear any <BREAK>
240C 3E6A       ØØØØ3         LD      A,1Ø6
240E EF         ØØØØ4         RST     4Ø
240F C9         Ø713Ø         RET                     ;Ret to DOS
                Ø714Ø ;
                Ø715Ø ;       Internal Error Message Handling
                Ø716Ø ;
2410 217C29     Ø717Ø NOMEM   LD      HL,NOMEM$       ;"No memroy available
2413 DD         Ø718Ø         DB      ØDDH
2414 219629     Ø719Ø PRMERR  LD      HL,PRMERR$      ;"Parm error
2417            Ø72ØØ SETERR  @@LOGOT
                ØØØØ5         IFEQ    ØØH,1
                ØØØØ6         LD      HL,
                ØØØØ7         ENDIF
2417 3EØC       ØØØØ8         LD      A,12
2419 EF         ØØØØ9         RST     4Ø
241A 21FFFF     Ø721Ø FRCERR  LD      HL,-1           ;Set abort code
241D 18EA       Ø722Ø         JR      SAVESP          ;Exit
                Ø723Ø ;
                Ø724Ø ;       I/O Error Handler
                Ø725Ø ;
241F 6F         Ø726Ø IOERR   LD      L,A             ;Error # to HL
2420 2600       Ø727Ø         LD      H,Ø
2422 F6CØ       Ø728Ø         OR      ØCØH            ;Set brief to return
2424 4F         Ø729Ø         LD      C,A
2425            Ø73ØØ         @@ERROR                 ;Display the error
2425 3E1A       ØØ1Ø          LD      A,26
2427 EF         ØØ11          RST     4Ø
2428 18DF       Ø731Ø         JR      SAVESP          ;Exit
                Ø732Ø ;
                Ø733Ø ;       SYSTEM1 - Set the O/S's parameters
                Ø734Ø ;
                Ø735Ø SYSTEM1
242A ED73ØA24   Ø736Ø         LD      (SAVESP+1),SP   ;Save the stack
                Ø737Ø ;
242E 11A32A     Ø738Ø         LD      DE,PRMTBL$      ;Get parms
```

```
2431            07390           @@PARAM
2431 3E11       00012           LD      A,17
2433 EF         00013           RST     40
2434 C21424     07400           JP      NZ,PRMERR       ;Jump on parm error
2437            07410           @@FLAGS                 ;Get IY = flagtab
2437 3E65       00014           LD      A,101
2439 EF         00015           RST     40
243A 010000     07420 SLOW      LD      BC,0            ;Test SLOW parm
243D 78         07430           LD      A,B
243E B1         07440           OR      C
243F 2013       07450           JR      NZ,SETSLOW      ;Jump if SLOW entered
2441 010000     07460 FPARM     LD      BC,0            ;P/u FAST parm
2444 78         07470           LD      A,B
2445 B1         07480           OR      C
2446 281D       07490           JR      Z,FPEND         ;Neither SLOW nor FAST?
2448 FDCB12DE   07500           SET     3,(IY+'S'-'A')  ;Set fast in flag
                07510           IF      @MOD4
244C FDCB00CF6  07520           SET     6,(IY+'M'-'A')  ;Set bit 6 in port mask
                07530           ENDIF
2450 0655       07540           LD      B,55H           ;Init time slice byte
2452 1808       07550           JR      SETCLK          ;Go set it
                07560 ;
                07570 ;         Turn off FAST
                07580 ;
2454 FDCB129E   07590 SETSLOW   RES     3,(IY+'S'-'A')  ;Reset the FAST bit
                07600           IF      @MOD4
2458 FDCB00CB6  07610           RES     6,(IY+'M'-'A')  ;Reset bit in port mask
                07620           ENDIF
245C 78         07630 SETCLK    LD      A,B
245D 322B00     07640           LD      (TIMSL$),A      ;Set 55H=fast, FFH=slow
                07650           IF      @MOD4
2460 FD7E0C     07660           LD      A,(IY+'M'-'A')  ;Update the port
2463 D3EC       07670           OUT     (MODPORT),A
                07680           ENDIF
2465            07690 FPEND     EQU     $
                07700 ;
                07710 ;         Trace entry
                07720 ;
2465 010100     07730 TRACE     LD      BC,1            ;Init to pass by
2468 78         07740           LD      A,B
2469 B1         07750           OR      C
246A 280D       07760           JR      Z,TROFF         ;Go if Trace off
246C 3C         07770           INC     A
246D 200F       07780           JR      NZ,TREND        ;Go if not entered
                07790           IF      @MOD4
246F 11B107     07800           LD      DE,TRACE_INT    ;Pt to trace SYSRES rtn
                07810           ENDIF
                07820           IF      @MOD2
                07830           LD      DE,DO_TRACE     ;Address
                07840           ENDIF
2472 0E07       07850           LD      C,7             ;Init to task 7
2474            07860           @@ADTSK                 ;Turn TRACE (ON)
2474 3E1D       00016           LD      A,29
2476 EF         00017           RST     40
2477 1805       07870           JR      TREND
2479 0E07       07880 TROFF     LD      C,7             ;Init to task 7
247B            07890           @@RMTSK                 ;Remove TRACE task
247B 3E1E       00018           LD      A,30
247D EF         00019           RST     40
247E            07900 TREND     EQU     $
                07910 ;
```

```
                    07920 ;       Routine to handle BREAK
                    07930 ;
247E Ø1Ø1ØØ         07940 BREAK   LD      BC,1            ;Init BREAK=ON
2481 78             07950         LD      A,B
2482 B1             07960         OR      C
2483 28Ø9           07970         JR      Z,BRKOFF        ;Jump if BREAK=OFF
2485 3C             07980         INC     A               ;Check Break=ON
2486 2ØØA           07990         JR      NZ,BRKEND       ;Go if no parm used
2488 FDCB12A6       08000         RES     4,(IY+'S'-'A')  ;Correct system flag
248C 18Ø4           08010         JR      BRKEND
                    08020 ;
                    08030 ;       Disable <BREAK> key
                    08040 ;
248E FDCB12E6       08050 BRKOFF  SET     4,(IY+'S'-'A')
2492                08060 BRKEND  EQU     $
                    08070 ;
                    08080 ;       Date prompt suppression during BOOT
                    08090 ;
2492 Ø1Ø1ØØ         08100 DATE    LD      BC,1            ;Init to pass by
2495 78             08110         LD      A,B
2496 B1             08120         OR      C
2497 Ø6FF           08130         LD      B,ØFFH          ;Init to DATE Suppress
2499 28Ø4           08140         JR      Z,DOFF          ;Go if suppress
249B 3C             08150         INC     A
249C 2ØØB           08160         JR      NZ,DATEND       ;Go if not entered
249E 47             08170         LD      B,A             ;B=Ø on no suppress
249F CD5F29         08180 DOFF    CALL    GETCFG          ;Get track Ø, sector 2
24A2 78             08190         LD      A,B             ;Get back the code byte
24A3 32C21D         08200         LD      (DTPMT$&ØFFH+SBUFF$),A
24A6 CD5C29         08210         CALL    PUTCFG          ;Write it back
24A9                08220 DATEND  EQU     $
                    08230 ;
                    08240 ;       Time prompt suppression during BOOT
                    08250 ;
24A9 Ø1Ø1ØØ         08260 TIME    LD      BC,1            ;Init to pass by
24AC 78             08270         LD      A,B
24AD B1             08280         OR      C
24AE Ø6FF           08290         LD      B,ØFFH          ;Init to TIME Suppress
24BØ 28Ø4           08300         JR      Z,TOFF          ;Go if suppress
24B2 3C             08310         INC     A
24B3 2ØØB           08320         JR      NZ,TIMEND       ;Go if not entered
24B5 47             08330         LD      B,A             ;B=Ø on no suppress
24B6 CD5F29         08340 TOFF    CALL    GETCFG          ;Get track Ø, sector 2
24B9 78             08350         LD      A,B             ;Get back the code byte
24BA 32C31D         08360         LD      (TMPMT$&ØFFH+SBUFF$),A
24BD CD5C29         08370         CALL    PUTCFG          ;Write it back
24CØ                08380 TIMEND  EQU     $
                    08390 ;
                    08400 ;       BOOT Step rate adjust
                    08410 ;
24CØ Ø1FFFF         08420 BSPARM  LD      BC,-1           ;P/u parm
24C3 Ø4             08430         INC     B               ;User entry?
24C4 281B           08440         JR      Z,BSPEND        ;Go if not entered
24C6 79             08450         LD      A,C             ;Transfer parm
24C7 FEØ4           08460         CP      4
24C9 38Ø8           08470         JR      C,GUDBS         ;Ok if < 4
24CB 21832A         08480         LD      HL,BADBS$       ;  else bad step rate
24CE CD1724         08490         CALL    SETERR
24D1 18ØE           08500         JR      BSPEND
24D3 47             08510 GUDBS   LD      B,A             ;Save step
24D4 CD5F29         08520         CALL    GETCFG          ;Get System Info Sector
```

```
24D7 2E73       08530           LD      L,70H+3         ;Offset to data
24D9 7E         08540           LD      A,(HL)          ;P/u DCT+3 parms
24DA E6FC       08550           AND     0FCH            ;Strip off boot step
24DC B0         08560           OR      B               ;Merge new
24DD 77         08570           LD      (HL),A          ;Stuff back
24DE CD5C29     08580           CALL    PUTCFG          ;Write it back
24E1            08590 BSPEND    EQU     $
                08600 ;
                08610 ;         BLINK parm
                08620 ;
24E1 010100     08630 BLINK     LD      BC,1            ;P/U parm
24E4 78         08640           LD      A,B
24E5 B1         08650           OR      C
24E6 3D         08660           DEC     A               ;Specified?
24E7 2829       08670           JR      Z,BLIEND        ;Go if not
24E9 3C         08680           INC     A
24EA 2006       08690           JR      NZ,BLNKON       ;Go if BLINK=ON
24EC FDCB15F6   08700           SET     6,(IY+'V'-'A')  ;Turn off blinking
                08710           IF      @MOD2
                08720           LD      A,($CRSCHAR)    ;Get current char
                08730           AND     1FH             ;Keep size only
                08740           JR      SETBLK          ;Setup
                08750           ENDIF
                08760           IF      @MOD4
24F0 1820       08770           JR      BLIEND          ;Continue
                08780           ENDIF
24F2 FDCB15B6   08790 BLNKON    RES     6,(IY+'V'-'A')  ;Turn on blinking
24F6 FEFF       08800           CP      0FFH            ;If not just BLINK,
                08810           IF      @MOD4
24F8 2012       08820           JR      NZ,SETBLK       ;  set user cursor
                08830           ENDIF
                08840           IF      @MOD2
                08850           JR      Z,LARGE         ;Go if not
                08860 BLIADJ    SUB     7               ;Force in range 0-7
                08870           JR      NC,BLIADJ       ;Go till in range
                08880           ADD     A,7             ;Add back last sub
                08890           OR      01000000B       ;Set blink enable 1/16
                08900           JR      SETBLK          ;Continue
                08910           ENDIF
24FA 010000     08920 LARGE     LD      BC,0            ;  else test if large,
24FD 0C         08930           INC     C               ;  small, or default
                08940           IF      @MOD2
                08950           LD      A,01000111B     ;Standard cursor
                08960           ENDIF
                08970           IF      @MOD4
24FE 3E5F       08980           LD      A,'_'           ;Standard cursor
                08990           ENDIF
2500 2002       09000           JR      NZ,SMALPRM
                09010           IF      @MOD2
                09020           LD      A,01000000B     ;To large
                09030           ENDIF
                09040           IF      @MOD4
2502 3E8F       09050           LD      A,8FH           ;To large
                09060           ENDIF
2504 010000     09070 SMALPRM   LD      BC,0
2507 0C         09080           INC     C
2508 2002       09090           JR      NZ,SETBLK
                09100           IF      @MOD2
                09110           LD      A,01000101B     ;Small
                09120           ENDIF
                09130           IF      @MOD4
```

```
250A 3E88      Ø9140          LD      A,88H           ;Small
               Ø9150          ENDIF
250C 4F        Ø9160 SETBLK   LD      C,A             ;Xfer cursor char to C
250D Ø6Ø8      Ø9170          LD      B,8             ;Cursor update function
250F           Ø9180          @@VDCTL
250F 3EØF      ØØØ2Ø          LD      A,15
2511 EF        ØØØ21          RST     4Ø
2512           Ø919Ø BLIEND   EQU     $
               Ø92ØØ ;
               Ø921Ø ;        RESTORE parameter
               Ø922Ø ;
2512 Ø1Ø1ØØ    Ø923Ø RPARM    LD      BC,1            ;P/U parm
2515 78        Ø924Ø          LD      A,B
2516 B1        Ø925Ø          OR      C
2517 3D        Ø926Ø          DEC     A               ;Specified?
2518 28ØF      Ø927Ø          JR      Z,REND          ;Go if not
251A 3C        Ø928Ø          INC     A
251B 2ØØ1      Ø929Ø          JR      NZ,RSTRON       ;Go if RESTORE=on
251D 3E        Ø93ØØ          DB      3EH             ;Make LD A,n
251E AF        Ø931Ø RSTRON   XOR     A
251F 47        Ø932Ø          LD      B,A             ;Save step
252Ø CD5F29    Ø933Ø          CALL    GETCFG          ;Get config sector
2523 2EC4      Ø934Ø          LD      L,RSTOR$&ØFFH   ;Pt to RESTORE flag
2525 7Ø        Ø935Ø          LD      (HL),B          ;Stuff back
2526 CD5C29    Ø936Ø          CALL    PUTCFG          ;Put it back
2529           Ø937Ø REND     EQU     $
               Ø938Ø ;
               Ø939Ø ;        Type ahead processing
               Ø94ØØ ;
2529 Ø1Ø1ØØ    Ø941Ø TYPE     LD      BC,1            ;P/u Type parm value
252C 78        Ø942Ø          LD      A,B
252D B1        Ø943Ø          OR      C
252E 3D        Ø944Ø          DEC     A               ;Was Type used?
252F 28ØD      Ø945Ø          JR      Z,TYPEND        ;Jump if TYPE not entered
2531 3C        Ø946Ø          INC     A
2532 2ØØ6      Ø947Ø          JR      NZ,TYPEON       ;Jump if TYPE=ON
2534 FDCBØ38E  Ø948Ø          RES     1,(IY+'D'-'A')  ;Turn TYPE off
2538 18Ø4      Ø949Ø          JR      TYPEND
253A FDCBØ3CE  Ø95ØØ TYPEON   SET     1,(IY+'D'-'A')  ;Turn TYPE on
253E           Ø951Ø TYPEND   EQU     $
               Ø952Ø ;
               Ø953Ø ;        Process SMOOTH
               Ø954Ø ;
253E Ø1Ø1ØØ    Ø955Ø SMOOTH   LD      BC,1            ;P/u Smooth parm value
2541 78        Ø956Ø          LD      A,B
2542 B1        Ø957Ø          OR      C
2543 3D        Ø958Ø          DEC     A               ;Was Smooth used?
2544 2817      Ø959Ø          JR      Z,SMEND         ;Skip if not entered
2546 3C        Ø96ØØ          INC     A
2547 2ØØB      Ø961Ø          JR      NZ,SMON         ;Go if Smooth=ON
               Ø962Ø          IF      @MOD4
2549 FDCBØ39E  Ø963Ø          RES     3,(IY+'D'-'A')  ;Show OFF in DFLAG$
254D 3EØØ      Ø964Ø          LD      A,ØØ            ;Set to store NOP
254F 32ØEØØ    Ø965Ø          LD      (FDDINT$),A     ;  put it there
               Ø966Ø          ENDIF
2552 18Ø9      Ø967Ø          JR      SMEND
2554           Ø968Ø SMON     EQU     $
               Ø969Ø          IF      @MOD4
2554 FDCBØ3DE  Ø97ØØ          SET     3,(IY+'D'-'A')  ;Show ON in DFLAG$
2558 3EF3      Ø971Ø          LD      A,ØF3H          ;DI opcode
255A 32ØEØØ    Ø972Ø          LD      (FDDINT$),A     ;  put it there
```

```
                    Ø973Ø           ENDIF
255D                Ø974Ø SMEND     EQU     $
                    Ø975Ø ;
                    Ø976Ø ;         Routine to process HERTZ selection
                    Ø977Ø ;
255D Ø1Ø1ØØ         Ø978Ø HERTZ5Ø   LD      BC,1            ;P/u H5Ø parm value
256Ø 78             Ø979Ø           LD      A,B
2561 B1             Ø98ØØ           OR      C
2562 3D             Ø981Ø           DEC     A               ;Check if parm entered
2563 28Ø5           Ø982Ø           JR      Z,HZ5ØEND       ;Go if not
2565 3E19           Ø983Ø           LD      A,25            ;Init 25 ints/sec
2567 325ØØ7         Ø984Ø           LD      (HERTZ$),A      ;Put value in timer code
256A                Ø985Ø HZ5ØEND   EQU     $
256A Ø1Ø1ØØ         Ø986Ø HERTZ6Ø   LD      BC,1            ;P/u H6Ø parm value
256D 78             Ø987Ø           LD      A,B
256E B1             Ø988Ø           OR      C
256F 3D             Ø989Ø           DEC     A               ;Check if parm entered
257Ø 28Ø5           Ø99ØØ           JR      Z,HZ6ØEND       ;Go if not
2572 3E1E           Ø991Ø           LD      A,3Ø            ;Init 3Ø ints/sec
2574 325ØØ7         Ø992Ø           LD      (HERTZ$),A      ;  & load it to timer
2577                Ø993Ø HZ6ØEND   EQU     $
                    Ø994Ø ;
                    Ø995Ø ;         Routine to process GRAPHIC
                    Ø996Ø ;
2577 Ø1Ø1ØØ         Ø997Ø GRAPHI    LD      BC,1            ;Init for bypass
257A 78             Ø998Ø           LD      A,B
257B B1             Ø999Ø           OR      C
257C 3D             1ØØØØ           DEC     A               ;Check if parm entered
257D 28ØD           1ØØ1Ø           JR      Z,GRAEND        ;Go if not entered
257F 3C             1ØØ2Ø           INC     A               ;Test of GRAPHIC=off
258Ø 28Ø6           1ØØ3Ø           JR      Z,GROFF         ;Go if off
2582 FDCBØ3FE       1ØØ4Ø           SET     7,(IY+'D'-'A')  ;  else turn on
2586 18Ø4           1ØØ5Ø           JR      GRAEND
2588 FDCBØ3BE       1ØØ6Ø GROFF     RES     7,(IY+'D'-'A')  ;Turn graphic off
258C                1ØØ7Ø GRAEND    EQU     $
                    1ØØ8Ø ;
                    1ØØ9Ø ;         Routine to process ALIVE
                    1Ø1ØØ ;
258C Ø1Ø1ØØ         1Ø11Ø ALIVE     LD      BC,1            ;P/u Alive parm value
258F 78             1Ø12Ø           LD      A,B
259Ø B1             1Ø13Ø           OR      C
2591 3D             1Ø14Ø           DEC     A               ;Check if entered
2592 2866           1Ø15Ø           JR      Z,ALVEND        ;Bypass if parm omitted
2594 3C             1Ø16Ø           INC     A
2595 2ØØ7           1Ø17Ø           JR      NZ,ALIVEON      ;Jump if ALIVE=ON
                    1Ø18Ø           IF      @MOD2
                    1Ø19Ø           LD      C,6             ;Slot alive
                    1Ø2ØØ           ENDIF
                    1Ø21Ø           IF      @MOD4
2597 ØE Ø3          1Ø22Ø           LD      C,3             ;Slot alive
                    1Ø23Ø           ENDIF
2599                1Ø24Ø           @@RMTSK                 ;Remove it if OFF
2599 3E1E           ØØØ22           LD      A,3Ø
259B EF             ØØØ23           RST     4Ø
259C 185C           1Ø25Ø           JR      ALVEND
                    1Ø26Ø ;
259E                1Ø27Ø ALIVEON   EQU     $
                    1Ø28Ø           IF      @MOD2
                    1Ø29Ø           LD      C,6             ;Slot alive
                    1Ø3ØØ           LD      DE,DO_ALIVE     ;Task address
259E                1Ø31Ø           @@ADTSK                 ;Add task
```

```
                00024           LD      A,29
                00025           RST     40
                1032Ø           ENDIF
                1033Ø           IF      @MOD4
259E CD7729     1034Ø           CALL    GOTMEM?         ;Is HIGH$ frozen?
25A1 DA1Ø24     1035Ø           JP      C,NOMEM         ;Quit if so
25A4 CD6F29     1036Ø           CALL    GETHI$          ;Get address for Alive
25A7 22D325     1037Ø           LD      (ALVBGN+2),HL   ;Stuff last byte used
25AA Ø12ØØØ     1038Ø           LD      BC,ALVEND-ALVBGN        ;Get len of code
25AD AF         1039Ø           XOR     A
25AE ED42       1Ø4ØØ           SBC     HL,BC           ;Calc new HIGH$
25BØ Ø6ØØ       1Ø41Ø           LD      B,Ø
25B2            1Ø42Ø           @@HIGH$                 ;Stuff new HIGH$
25B2 3E64       00026           LD      A,1ØØ
25B4 EF         00027           RST     40
25B5 23         1Ø43Ø           INC     HL              ;Point to module start
25B6 E5         1Ø44Ø           PUSH    HL              ;Save start
25B7 ØE1Ø       1Ø45Ø           LD      C,ALVTCB-ALVBGN+2
25B9 Ø9         1Ø46Ø           ADD     HL,BC           ;Point to alive task
25BA 22DF25     1Ø47Ø           LD      (ALVTCB),HL     ;  & stuff ALIVE TCB
25BD E3         1Ø48Ø           EX      (SP),HL         ;Rcvr start of module
25BE EB         1Ø49Ø           EX      DE,HL           ;  into DE
25BF 21D125     1Ø5ØØ           LD      HL,ALVBGN
25C2 Ø12ØØØ     1Ø51Ø           LD      BC,ALVEND-ALVBGN
25C5 EDBØ       1Ø52Ø           LDIR                    ;Move to high memory
25C7 D1         1Ø53Ø           POP     DE              ;Rcvr start of task
25C8 1B         1Ø54Ø           DEC     DE              ;Back up to TCB
25C9 1B         1Ø55Ø           DEC     DE
25CA ØE03       1Ø56Ø           LD      C,3             ;Add as task 3
25CC            1Ø57Ø           @@ADTSK
25CC 3E1D       00028           LD      A,29
25CE EF         00029           RST     40
25CF 1829       1Ø58Ø           JR      ALVEND
                1Ø59Ø           ENDIF
                1Ø6ØØ ;
                1Ø61Ø ;         ALIVE high memory module
                1Ø62Ø ;
                1Ø63Ø           IF      @MOD4
25D1 18FE       1Ø64Ø ALVBGN    JR      $
25D3 ØØØØ       1Ø65Ø           DW      $-$
25D5 Ø5         1Ø66Ø           DB      5,'ALIVE'
    41 4C 49 56 45
25DB ØØØØ       1Ø67Ø           DW      Ø,Ø
    ØØØØ
25DF ØØØØ       1Ø68Ø ALVTCB    DW      Ø
25E1 214FØØ     1Ø69Ø           LD      HL,79           ;Get character at Ø,79
25E4 E5         1Ø7ØØ           PUSH    HL
25E5 Ø6Ø1       1Ø71Ø           LD      B,1             ;Set function 1
25E7 CD99ØB     1Ø72Ø           CALL    @VDCTL          ;P/u character currently
25EA FEA3       1Ø73Ø           CP      ØA3H            ;  on the screen and
25EC 3E93       1Ø74Ø           LD      A,93H           ;  exchange it
25EE 28Ø2       1Ø75Ø           JR      Z,ALIVE1
25FØ 3EA3       1Ø76Ø           LD      A,ØA3H
25F2 E1         1Ø77Ø ALIVE1    POP     HL
25F3 Ø6Ø2       1Ø78Ø           LD      B,2             ;Function to put char
25F5 4F         1Ø79Ø           LD      C,A
25F6 CD99ØB     1Ø8ØØ           CALL    @VDCTL          ;  at row,col
25F9 C9         1Ø81Ø           RET
                1Ø82Ø           ENDIF
25FA            1Ø83Ø ALVEND    EQU     $
                1Ø84Ø ;
```

```
                1Ø85Ø ;       Routine to process SYSRES
                1Ø86Ø ;
25FA Ø1FFFF     1Ø87Ø SYSRES  LD      BC,-1           ;P/u Sysres parm value
25FD Ø4         1Ø88Ø         INC     B               ;Test if parm entered
25FE CA3B27     1Ø89Ø         JP      Z,RESEND        ;Go if not
26Ø1 CD7729     1Ø9ØØ         CALL    GOTMEM?         ;Is high memory available
26Ø4 DA1Ø24     1Ø91Ø         JP      C,NOMEM         ;Quit if not
26Ø7 79         1Ø92Ø         LD      A,C             ;Get module to reside
26Ø8 B7         1Ø93Ø         OR      A
26Ø9 2837       1Ø94Ø         JR      Z,BADSYS        ;Cannot sysres Ø
26ØB FEØ2       1Ø95Ø         CP      2               ;Check if it is SYS2
26ØD 2ØØ5       1Ø96Ø         JR      NZ,TEST6        ;Go if not
26ØF 2174Ø4     1Ø97Ø         LD      HL,DCT$+4       ;Point to SYSTEM DCT
2612 CBBE       1Ø98Ø         RES     7,(HL)          ;  & allow CKDRV now
2614 FEØ6       1Ø99Ø TEST6   CP      6
2616 38Ø8       11ØØØ         JR      C,GUDSYS        ;Can sysres 1-5
2618 FEØ9       11Ø1Ø         CP      9
261A 3826       11Ø2Ø         JR      C,BADSYS        ;Cannot sysres 6-8
261C FEØD       11Ø3Ø         CP      13
261E 3Ø22       11Ø4Ø         JR      NC,BADSYS       ;Nothing > 12 yet
262Ø F5         11Ø5Ø GUDSYS  PUSH    AF              ;Save sysres req
2621 216D1A     11Ø6Ø         LD      HL,HKRES$+1     ;Check if the driver
2624 5E         11Ø7Ø         LD      E,(HL)          ;  is already resident
2625 23         11Ø8Ø         INC     HL
2626 56         11Ø9Ø         LD      D,(HL)
2627 217F1A     111ØØ         LD      HL,CKMOD@       ;Standard address
262A B7         1111Ø         OR      A               ;Reset carry
262B ED52       1112Ø         SBC     HL,DE           ;No driver if =
262D 2871       1113Ø         JR      Z,PUTDVR        ;Go and install driver
262F 21EØFF     1114Ø         LD      HL,-32          ;Find restab$ pos
2632 19         1115Ø         ADD     HL,DE           ;Pt to vector table
2633 Ø7         1116Ø         RLCA                    ;Request x 2
2634 85         1117Ø         ADD     A,L
2635 6F         1118Ø         LD      L,A
2636 3ØØ1       1119Ø         JR      NC,$+3
2638 24         112ØØ         INC     H               ;RESTAB$ + entry index
2639 23         1121Ø         INC     HL              ;Pt to hi order
263A 7E         1122Ø         LD      A,(HL)          ;P/u hi order
263B B7         1123Ø         OR      A
263C 281B       1124Ø         JR      Z,MOVITIN       ;Go if not there
263E 21Ø22A     1125Ø         LD      HL,MODRES$      ;  else show already res'ed
2641 DD         1126Ø         DB      ØDDH
2642 21E429     1127Ø BADSYS  LD      HL,BADSYS$
2645 CD1724     1128Ø         CALL    SETERR          ;Log & Set error
2648 F1         1129Ø         POP     AF
2649 C33B27     113ØØ         JP      GETSEC2
264C 18FE       1131Ø RES1    JR      $               ;No real entry
264E ØØØØ       1132Ø         DW      $-$             ;Last used
2650 Ø4         1133Ø         DB      4,'SYS '
     53 59 53 2Ø
2655 ØØØØ       1134Ø         DW      Ø,Ø
     ØØØØ
2658            1135Ø RES1E   EQU     $-1
2659 326ØØØ     1136Ø MOVITIN LD      (OVRLY$),A      ;Show no overlay res
265C F1         1137Ø         POP     AF              ;Rcvr requested SYS
265D F5         1138Ø         PUSH    AF
265E C6Ø2       1139Ø         ADD     A,2             ;Adjust for dec
2660 F68Ø       114ØØ         OR      8ØH             ;Set Sys request
2662 E5         1141Ø         PUSH    HL              ;Save restab$ addr
2663 CD7F1A     1142Ø         CALL    CKMOD@          ;Load SYSn, no exec
2666 CD6F29     1143Ø         CALL    GETHI$          ;P/u HIGH$
```

```
2669 224E26   11440         LD      (RES1+2),HL     ;Save in header
266C EB       11450         EX      DE,HL
266D ED4BFE23 11460         LD      BC,(MAXCOR$-2)  ;P/u length of SYSx
2671 21001E   11470         LD      HL,1E00H        ;Pt to start of overlay
2674 09       11480         ADD     HL,BC
2675 2B       11490         DEC     HL              ;Point to last byte
2676 C5       11500         PUSH    BC              ;Save length
2677 EDB8     11510         LDDR
2679 C1       11520         POP     BC              ;Rcvr length of SYSx
267A 78       11530         LD      A,B             ;  & stuff in front
267B 12       11540         LD      (DE),A          ;  of SYSRESed module
267C 1B       11550         DEC     DE
267D 79       11560         LD      A,C
267E 12       11570         LD      (DE),A
267F E1       11580         POP     HL              ;Rcvr RESTAB ptr
2680 72       11590         LD      (HL),D          ;Save hi order addr
2681 2B       11600         DEC     HL
2682 73       11610         LD      (HL),E          ;Save low order addr
2683 F1       11620         POP     AF              ;Rcvr module requested
2684 FE0A     11630         CP      10              ;Convert to ASCII
2686 3802     11640         JR      C,$+4
2688 C607     11650         ADD     A,7
268A C630     11660         ADD     A,'0'
268C 215426   11670         LD      HL,RES1+8
268F 77       11680         LD      (HL),A          ;Stuff module name
2690 1B       11690         DEC     DE              ;Pt to 1st free byte
              11700                                 ;  & move in the
2691 010D00   11710         LD      BC,13           ;  linkage protocol
2694 215826   11720         LD      HL,RES1E
2697 EDB8     11730         LDDR
2699 EB       11740         EX      DE,HL
269A          11750         @@HIGH$                 ;Stuff new HIGH$ (B=0)
269A 3E64     00030         LD      A,100
269C EF       00031         RST     40
269D C33B27   11760         JP      RESEND
              11770 ;
26A0 CD6F29   11780 PUTDVR  CALL    GETHI$          ;P/u high mem
26A3 22CF26   11790         LD      (RESBGN+2),HL   ;Stuff last used
26A6 016E00   11800         LD      BC,RESEND-RESBGN
26A9 ED42     11810         SBC     HL,BC           ;Make space for driver
26AB 0600     11820         LD      B,0
26AD          11830         @@HIGH$                 ;Stuff new value
26AD 3E64     00032         LD      A,100
26AF EF       00033         RST     40
26B0 E5       11840         PUSH    HL              ;Save new HIGH$
26B1 110D00   11850         LD      DE,13           ;Pt to RESTAB$-3
26B4 19       11860         ADD     HL,DE
26B5 220827   11870         LD      (RESD1+1),HL    ;Stuff ptr to RESTAB$-3
26B8 E1       11880         POP     HL
26B9 23       11890         INC     HL              ;Resbgn
26BA E5       11900         PUSH    HL
26BB EB       11910         EX      DE,HL           ;Where it goes to DE
26BC 21CD26   11920         LD      HL,RESBGN       ;Where it's now
26BF EDB0     11930         LDIR                    ;Move up to high
26C1 E1       11940         POP     HL              ;Rcvr where it got to
26C2 112F00   11950         LD      DE,32+RESTAB$-RESBGN    ;Index start of dvr
26C5 19       11960         ADD     HL,DE           ;Pt to driver entry
26C6 226D1A   11970         LD      (HKRES$+1),HL   ;Hook into SYS0
26C9 F1       11980         POP     AF              ;Rcvr code & loop
26CA C32026   11990         JP      GUDSYS          ;  to get the request
              12000 ;
```

```
26CD 18FE      12010 RESBGN  JR      $               ;No real entry
26CF 0000      12020         DW      $-$
26D1 06        12030         DB      6,'SYSRES'
     53 59 53 52 45 53
26D8 0000      12040         DW      0,0
     0000
26DC 0100      12050 RESTAB$ DW      1,0,0,0,0,0,1,1
     0000 0000 0000 0000 0000 0100 0100
26EC 0000      12060         DW      0,0,0,0,0,0,0,0
     0000 0000 0000 0000 0000 0000 0000
               12070 ;
26FC E5        12080 RESDVR  PUSH    HL
26FD F5        12090         PUSH    AF              ;Save SYS needed
26FE 216900    12100         LD      HL,OVRLY$       ;Check if already in
2701 AE        12110         XOR     (HL)            ;  overlay area
2702 E60F      12120         AND     0FH             ;Strip garbage
2704 2830      12130         JR      Z,NOTRES
2706 F1        12140         POP     AF
2707 21DD26    12150 RESD1   LD      HL,RESTAB$+1    ;P/u table ptr
270A F5        12160         PUSH    AF
270B E60F      12170         AND     0FH             ;Check if in high mem
270D 07        12180         RLCA                    ;X 2
270E 85        12190         ADD     A,L
270F 6F        12200         LD      L,A
2710 3001      12210         JR      NC,$+3
2712 24        12220         INC     H
2713 7E        12230         LD      A,(HL)          ;P/u hi order
2714 B7        12240         OR      A
2715 281F      12250         JR      Z,NOTRES        ;Go if not in high
2717 2B        12260         DEC     HL              ;Pt to low
2718 6E        12270         LD      L,(HL)          ;P/u lo order vector
2719 67        12280         LD      H,A             ;Xfer hi order
271A D5        12290         PUSH    DE              ;Save these regs
271B C5        12300         PUSH    BC
271C 4E        12310         LD      C,(HL)          ;P/u module length
271D 23        12320         INC     HL              ;  into BC
271E 46        12330         LD      B,(HL)
271F 23        12340         INC     HL              ;Point to module start
2720 11001E    12350         LD      DE,1E00H        ;Point to overlay region
2723 ED53751A  12360         LD      (HKRES$+9),DE   ;Upd TRAADR
2727 C5        12370         PUSH    BC              ;Save the length
2728 EDB0      12380         LDIR
272A E1        12390         POP     HL              ;Rcvr the length
272B 22FE23    12400         LD      (MAXCOR$-2),HL  ;Stuff pointer
272E C1        12410         POP     BC              ;Restore regs
272F D1        12420         POP     DE
2730 F1        12430         POP     AF
2731 32701A    12440         LD      (HKRES$+4),A    ;Show its resident
2734 E1        12450         POP     HL              ;  in tempy loc'n
2735 C9        12460         RET
2736 F1        12470 NOTRES  POP     AF
2737 E1        12480         POP     HL
2738 C37F1A    12490         JP      CKMOD@
273B           12500 RESEND  EQU     $
               12510 ;
               12520 ;       Routine to process DRIVE parameter
               12530 ;
273B 3A7B27    12540 GETSEC2 LD      A,(CYLPRM+1)    ;If CYL=c entered,
273E B7        12550         OR      A               ;  get config sector
273F C45F29    12560         CALL    NZ,GETCFG
2742 01FFFF    12570 DRIVE   LD      BC,-1
```

```
2745 DD217Ø1D 12580         LD      IX,SBUFF$+7ØH   ;Pt to DCT sector fields
2749 FD217ØØ4 12590         LD      IY,DCT$         ;Init to do all DCT$'s *
274D Ø4       12600         INC     B
274E 28ØB     12610         JR      Z,CKEI          ;Go if Drive not entered
2750 79       12620         LD      A,C
2751 FEØ8     12630         CP      8               ;Drive in range?
2753 D21424   12640         JP      NC,PRMERR       ;Go if > 7
2756          12650         @@GTDCT                 ;Get its DCT
2756 3E51     ØØØ34         LD      A,81
2758 EF       ØØØ35         RST     4Ø
2759 FDE5     12660         PUSH    IY              ;Save the DCT address
275B          12670 CKEI    EQU     $
              12680 ;
275B FDE5     12690 ADJEI   PUSH    IY              ;Calc IY offset into
275D E1       12700         POP     HL              ;  DCTs to adjust IX
275E 1190FB   12710         LD      DE,-DCT$        ;  without affecting IY
2761 19       12720         ADD     HL,DE
2762 EB       12730         EX      DE,HL
2763 DD19     12740         ADD     IX,DE
2765          12750 DRVLOOP EQU     $
              12760 ;
              12770 ;       Routine to process WP parameter
              12780 ;
2765 Ø1Ø1ØØ   12790 WPARM   LD      BC,1            ;P/u WP parm value
2768 78       12800         LD      A,B
2769 B1       12810         OR      C
276A 3D       12820         DEC     A               ;Check if entered
276B 28ØD     12830         JR      Z,WPEND         ;  and go if not
276D 3C       12840         INC     A
276E 2806     12850         JR      Z,WPOFF         ;Go if OFF
2770 FDCBØ3FE 12860 WPON    SET     7,(IY+3)        ;Turn on write protect
2774 18Ø4     12870         JR      WPEND
2776 FDCBØ3BE 12880 WPOFF   RES     7,(IY+3)        ;Turn off write protect
277A          12890 WPEND   EQU     $
              12900 ;
              12910 ;       Routine to process CYL parameter
              12920 ;
277A Ø1ØØØØ   12930 CYLPRM  LD      BC,Ø            ;P/u Cyl parm value
277D 78       12940         LD      A,B
277E B1       12950         OR      C
277F 282B     12960         JR      Z,CYLEND        ;Go if parm not entered
2781 DDCBØ36E 12970         BIT     5,(IX+3)        ;If drive is 8",
2785 28ØC     12980         JR      Z,CYL1          ;  can't do it
2787 3A4327   12990         LD      A,(DRIVE+1)     ;If drive entered,
278A B7       13000         OR      A               ;  give error msg
278B 21A629   13010         LD      HL,NOCYL8$      ;'Can't CYL on 8" drive '
278E C41724   13020         CALL    NZ,SETERR       ;Msg if DRIVE=d
2791 1819     13030         JR      DISABL
2793 79       13040 CYL1    LD      A,C             ;P/u user entry
2794 3D       13050         DEC     A
2795 FE22     13060         CP      34              ;Not < 35 allowed
2797 3804     13070         JR      C,BADCYL        ;Go if bad
2799 FE60     13080         CP      96              ;Max for 5-1/4" floppy
279B 380C     13090         JR      C,GUDCYL        ;Good if < 96
279D 21C029   13100 BADCYL  LD      HL,BADCYL$      ;"Cyl out of range
27AØ CD1724   13110         CALL    SETERR          ;Dsply and quit
27A3 AF       13120         XOR     A
27A4 327B27   13130         LD      (CYLPRM+1),A
27A7 18Ø3     13140         JR      DISABL
27A9 DD77Ø6   13150 GUDCYL  LD      (IX+6),A        ;Stuff default DCT
27AC          13160 CYLEND  EQU     $
```

```
                13170 ;
                13180 ;       Routine to process DISABL parameter
                13190 ;
27AC Ø1ØØØØ     13200 DISABL  LD      BC,Ø            ;P/u parm value
27AF 78         13210         LD      A,B
27BØ B1         13220         OR      C
27B1 28ØA       13230         JR      Z,ENABLE        ;Go if not DISABLE
27B3 3A4327     13240         LD      A,(DRIVE+1)     ;Don't permit disable
27B6 3C         13250         INC     A               ;  if DRIVE parm not
27B7 281Ø       13260         JR      Z,STEP          ;  entered
27B9 3EC9       13270         LD      A,ØC9H          ;Init disable byte
27BB 18Ø9       13280         JR      ENADIS
27BD Ø1ØØØØ     13290 ENABLE  LD      BC,Ø            ;P/u parm value
27CØ 78         13300         LD      A,B
27C1 B1         13310         OR      C
27C2 28Ø5       13320         JR      Z,STEP          ;Go if not ENABLE
27C4 3EC3       13330         LD      A,ØC3H          ;Init enable byte
27C6 FD77ØØ     13340 ENADIS  LD      (IY),A          ;Stuff Ena or Dis byte
                13350 ;
                13360 ;       Routine to process STEP parameter
                13370 ;
27C9 Ø1FFFF     13380 STEP    LD      BC,-1           ;P/u Step parm value
27CC Ø4         13390         INC     B
27CD 2815       13400         JR      Z,STEPEND       ;Go if STEP not entered
27CF 79         13410         LD      A,C
27DØ FEØ4       13420         CP      4               ;Step must be < 4
27D2 D21424     13430         JP      NC,PRMERR       ;Quit if out of range
27D5 FDCBØ466   13440         BIT     4,(IY+4)        ;Alien controller?
27D9 2Ø39       13450         JR      NZ,WDESD        ;Bypass if alien
27DB FD7EØ3     13460         LD      A,(IY+3)        ;P/u DCT byte and
27DE E6FC       13470         AND     ØFCH            ;  mask out old step
27EØ B1         13480         OR      C               ;Or in the new value
27E1 FD77Ø3     13490         LD      (IY+3),A        ;  and put back in DCT
27E4            13500 STEPEND EQU     $
                13510 ;
                13520 ;       Routine to process DELAY parameter
                13530 ;
27E4 Ø1Ø1ØØ     13540 DELAY   LD      BC,1            ;P/u the parm value
27E7 78         13550         LD      A,B
27E8 B1         13560         OR      C
27E9 3D         13570         DEC     A               ;Check if entered
27EA 281Ø       13580         JR      Z,CKDRV         ;Go if no delay entered
27EC 3C         13590         INC     A               ;Check if OFF
27ED Ø6ØØ       13600         LD      B,Ø             ;Init delay=on
27EF 2ØØ2       13610         JR      NZ,DELAY1       ;Go if ON
27F1 Ø6Ø4       13620         LD      B,4             ;Init delay=off
27F3 FD7EØ3     13630 DELAY1  LD      A,(IY+3)        ;Update delay
27F6 E6FB       13640         AND     ØFBH            ;  in DCT
27F8 BØ         13650         OR      B
27F9 FD77Ø3     13660         LD      (IY+3),A
                13670 ;
                13680 ;       Routine to process CKDRV parameter
                13690 ;
27FC Ø1Ø1ØØ     13700 CKDRV   LD      BC,1            ;P/u Ckdrv parm value
27FF 78         13710         LD      A,B
2800 B1         13720         OR      C
2801 3D         13730         DEC     A               ;Check if entered
2802 281Ø       13740         JR      Z,WDESD         ;Go if not entered
2804 3C         13750         INC     A               ;See if OFF or ON
2805 Ø6ØØ       13760         LD      B,Ø             ;Init ckdrv=on
2807 2ØØ2       13770         JR      NZ,CKDRV1       ;Go if ON
```

```
28Ø9 Ø68Ø     13778Ø          LD      B,8ØH           ;Set bit 7, ckdrv off
28ØB FD7EØ4   13779Ø CKDRV1   LD      A,(IY+4)        ;Update ckdrv bit
28ØE E67F     13788Ø          AND     7FH             ;  by removing old state,
2811Ø BØ      13789Ø          OR      B               ;  merging Ø or 8ØH
2811 FD77Ø4   13792Ø          LD      (IY+4),A        ;  and putting back in DCT
2814 3A4327   13793Ø WDESD    LD      A,(DRIVE+1)     ;Drive parm used prev ?
2817 3C       13794Ø          INC     A
2818 2Ø14     13795Ø          JR      NZ,PUTSEC2      ;Go if DRIVE entered
281A 11ØAØØ   13796Ø          LD      DE,1Ø           ;  else loop thru all DCTs
281D DD19     13797Ø          ADD     IX,DE           ;Advance to next DCT
281F FD19     13798Ø          ADD     IY,DE
2821 FDE5     13799Ø          PUSH    IY
2823 E1       13888Ø          POP     HL              ;Xfer DCT start to HL
2824 7D       13891Ø          LD      A,L             ;Ck on end of DCTs
2825 FECØ     13892Ø          CP      8Ø+7ØH          ;Offset is 7ØH
2827 DA6527   13893Ø          JP      C,DRVLOOP       ;Loop until done
282A 21ØØØØ   13894Ø          LD      HL,Ø            ;Show no DCT given
282D E5       13895Ø          PUSH    HL
282E 3A7B27   13896Ø PUTSEC2  LD      A,(CYLPRM+1)    ;If CYL=c entered
2831 B7       13897Ø          OR      A               ;  put config sector
2832 C45C29   13898Ø          CALL    NZ,PUTCFG
              13899Ø ;
              14ØØØ ;         Routine to process DRIVER parameter
              14Ø1Ø ;
2835 11ØØØØ   14Ø2Ø DVRPARM   LD      DE,Ø            ;P/u parm value
2838 7A       14Ø3Ø           LD      A,D
2839 B3       14Ø4Ø           OR      E
283A CAD928   14Ø5Ø           JP      Z,DVR4          ;Go if DRIVER not entered
283D 3C       14Ø6Ø           INC     A               ;Full name given?
283E 2823     14Ø7Ø           JR      Z,GETN3         ;Go if not & prompt
2841Ø 219E2B  14Ø8Ø           LD      HL,FCB1+1       ;  else pick it up
2843 E5       14Ø9Ø           PUSH    HL              ;Save ptr to start
2844 1A       14188Ø GETN1    LD      A,(DE)          ;P/u name char
2845 FEØD     14189Ø          CP      CR
2847 2815     14198Ø          JR      Z,GETN2         ;Exit on eol
2849 FE2C     14199Ø          CP      ','
284B 2811     14288Ø          JR      Z,GETN2         ;Exit on end of parm
284D FE22     14289Ø          CP      '"'
284F 28ØD     14298Ø          JR      Z,GETN2         ;Exit on closing quote
2851 FE29     14299Ø          CP      ')'
2853 28Ø9     14388Ø          JR      Z,GETN2         ;Exit on end of parms
2855 FEØ3     14389Ø          CP      3
2857 28Ø5     14398Ø          JR      Z,GETN2         ;Exit on JCL line end
2859 77       14399Ø          LD      (HL),A          ;  else xfer the char
285A 13       14488Ø          INC     DE
285B 23       14489Ø          INC     HL
285C 18E6     14498Ø          JR      GETN1           ;Loop
285E 36ØD     14499Ø GETN2    LD      (HL),CR         ;End with CR
2861Ø E1      14588Ø          POP     HL              ;Recover name ptr
2861 1844     14589Ø          JR      DVR2
2863          14598Ø GETN3    @@DSPLY EIPMPT          ;"Enter dct dvr name
              ØØØ36           IFEQ    Ø1H,1
2863 21212A   ØØØ37           LD      HL,EIPMPT
              ØØØ38           ENDIF
2866 3EØA     ØØØ39           LD      A,1Ø
2868 EF       ØØØ4Ø           RST     4Ø
2869 219E2B   14688Ø          LD      HL,FCB1+1       ;Pt to buffer
286C Ø1ØØ18   14689Ø          LD      BC,24<8         ;24 char max
286F          14698Ø          @@KEYIN                 ;Get filespec
286F 3EØ9     ØØØ41           LD      A,9
2871 EF       ØØØ42           RST     4Ø
```

```
2872 3Ø33      14320          JR      NC,DVR2          ;Go if not BREAK
               14330 ;
               14340 ;        BREAK entered on DRIVER request
               14350 ;        Reset all drive code table positions
               14360 ;        if DRIVE not entered
               14370 ;
2874 CD5F29    14380          CALL    GETCFG           ;Get SIS
2877 21ØØ19    14390          LD      HL,SBUFF$+7ØH-DCT$      ;Pt to tables "offset"
287A 3A4327    14400 DDCT1    LD      A,(DRIVE+1)      ;Was Drive entered?
287D 3C        14410          INC     A
287E D1        14420          POP     DE               ;Rcvr drive #
287F 2ØØ3      14430          JR      NZ,DDCT2
2881 117ØØ4    14440          LD      DE,DCT$          ;Else use all 8
2884 19        14450 DDCT2    ADD     HL,DE            ;Index the specific DCT
2885 7B        14460 DVRA     LD      A,E
2886 FE98      14470          CP      4Ø+7ØH
2888 3EC9      14480          LD      A,ØC9H           ;Disable drives 4-7
288A 3ØØ2      14490          JR      NC,DVRB
288C 3EC3      14500          LD      A,ØC3H
288E 77        14510 DVRB     LD      (HL),A           ;Stuff vector (JP/RET)
288F 2C        14520          INC     L                ;Pt to default DCT+3
2890 2C        14530          INC     L
2891 2C        14540          INC     L
2892 1C        14550          INC     E                ;Pt to resident DCT+3
2893 1C        14560          INC     E
2894 1C        14570          INC     E
2895 Ø1Ø7ØØ    14580          LD      BC,7             ;Shift defaults to
2898 EDBØ      14590          LDIR                     ;  resident DCT
289A 3A4327    14600 DVRC     LD      A,(DRIVE+1)      ;Test if drive entered
289D 3C        14610          INC     A
289E 2Ø3A      14620          JR      NZ,SYSPRM        ;Go if yes
28AØ 7B        14630          LD      A,E              ;Ck on end of DCTs
28A1 FECØ      14640          CP      8Ø+7ØH
28A3 3ØEØ      14650          JR      NC,DVRA          ;Loop until all 8
28A5 1833      14660          JR      SYSPRM
28A7 7E        14670 DVR2     LD      A,(HL)           ;Is first char <ENTER>?
28A8 FEØD      14680          CP      CR
28AA 282D      14690          JR      Z,DVR4           ;Bypass if default
28AC CD7729    14700          CALL    GOTMEM?
28AF D1        14710          POP     DE               ;Pop to keep stack
28BØ DA1Ø24    14720          JP      C,NOMEM          ;  straight if error
28B3 D5        14730          PUSH    DE
28B4 119D2B    14740          LD      DE,FCB1          ;Else fetch spec
28B7           14750          @@FSPEC
28B7 3E4E      ØØØ43          LD      A,78
28B9 EF        ØØØ44          RST     4Ø
28BA 21AØ2A    14760          LD      HL,DCTEXT        ;Use /DCT as the
28BD           14770          @@FEXT                   ;  default extension
28BD 3E4F      ØØØ45          LD      A,79
28BF EF        ØØØ46          RST     4Ø
28CØ           14780          @@FLAGS
28CØ 3E65      ØØØ47          LD      A,1Ø1
28C2 EF        ØØØ48          RST     4Ø
28C3 FDCB12D6  14790          SET     2,(IY+'S'-'A')   ;Set RUN bit
28C7           14800          @@LOAD                   ;Load the prog
28C7 3E4C      ØØØ49          LD      A,76
28C9 EF        ØØØ5Ø          RST     4Ø
28CA D1        14810          POP     DE               ;Recover drive table
28CB C21F24    14820          JP      NZ,IOERR         ;Quit on load error
28CE D5        14830          PUSH    DE
28CF E5        14840          PUSH    HL               ;Save prog's traadr
```

```
28DØ 21D928   14850          LD      HL,DVR4          ;Pt to our return
28D3 E3       14860          EX      (SP),HL          ;Exch with his
28D4 FDCBØ2DE 14870          SET     3,(IY+'C'-'A')   ;Set system request
28D8 E9       14880          JP      (HL)             ;Go to prog
28D9 D1       14890 DVR4     POP     DE               ;Rcvr drive table
              14900 ;
              14910 ;        System = new system drive
              14920 ;
28DA Ø1FFFF   14930 SYSPRM   LD      BC,-1            ;P/u parm value
28DD 78       14940          LD      A,B
28DE C3842B   14950          JP      PATCH1           ;Set HL=Ø, no abort if exit
              14960 ;
28E1 3D       14970 CONT1    DEC     A
28E2 FEØ8     14980          CP      8                ;Drive in range?
28E4 D21424   14990          JP      NC,PRMERR        ;Go if > 7
              15000 ;
              15010 ;        Check on diskette in place
              15020 ;
28E7          15030          @@CKDRV
28E7 3E21     ØØØ51          LD      A,33
28E9 EF       ØØØ52          RST     4Ø
28EA 2Ø25     15040          JR      NZ,SYSP2         ;Prompt if no diskette
28EC          15050          @@GTDCT                  ;Get its DCT
28EC 3E51     ØØØ53          LD      A,81
28EE EF       ØØØ54          RST     4Ø
              15060 ;
              15070 ;        Perform minimal check for SYSTEM disk
              15080 ;
28EF FD56Ø9   15090 SYSP1    LD      D,(IY+9)         ;P/u the dir cyl
28F2 21ØØ1D   15100          LD      HL,SBUFF$
28F5 5D       15110          LD      E,L              ;Init for GAT read
28F6          15120          @@RDSSC                  ;Read the sector
28F6 3E55     ØØØ55          LD      A,85
28F8 EF       ØØØ56          RST     4Ø
28F9 C21F24   15130          JP      NZ,IOERR         ;Quit on read error
28FC 2ECD     15140          LD      L,ØCDH           ;Check the SYSTEM bit
28FE CB7E     15150          BIT     7,(HL)
29ØØ 2ØØF     15160          JR      NZ,SYSP2         ;Not system? Go prompt
29Ø2 21ØØ1D   15170          LD      HL,SBUFF$        ;Point to buffer
29Ø5 1EØ5     15180          LD      E,5              ;Point to dir sec #5
29Ø7          15190          @@RDSSC                  ;Read the sector
29Ø7 3E55     ØØØ57          LD      A,85
29Ø9 EF       ØØØ58          RST     4Ø
29ØA 7E       15200          LD      A,(HL)           ;P/u the byte
29ØB E65Ø     15210          AND     Ø1Ø1ØØØØB        ;Check for a system file
29ØD FE5Ø     15220          CP      Ø1Ø1ØØØØB        ;  and not killed
              15230                                   ;  in this slot, (SYS1)
29ØF 2818     15240          JR      Z,SYSP3          ;Alive and a SYS file
              15250 ;
              15260 ;        Diskette is not a SYSTEM diskette
              15270 ;
2911 79       15280 SYSP2    LD      A,C              ;Stuff drive # in msg
2912 C63Ø     15290          ADD     A,'Ø'            ;Cvrt drive to ASCII
2914 328Ø2A   15300          LD      (NDSYS$+32),A
2917          15310          @@DSPLY NDSYS$           ;"Need system diskette..
              ØØØ59          IFEQ    Ø1H,1
2917 216Ø2A   ØØØ6Ø          LD      HL,NDSYS$
              ØØØ61          ENDIF
291A 3EØA     ØØØ62          LD      A,1Ø
291C EF       ØØØ63          RST     4Ø
291D 219D2B   15320          LD      HL,FCB1          ;Create response buffer
```

```
292Ø Ø1ØØØØ  15330          LD      BC,Ø<8          ;Only  BREAK or ENTER
2923          15340          @@KEYIN                 ;Wait for entry
2923 3EØ9     ØØØ64          LD      A,9
2925 EF       ØØØ65          RST     4Ø
2926 D8       15350          RET     C               ;Exit on <BREAK>
2927 18B1     15360          JR      SYSPRM          ;Continue to check
              15370 ;
2929 2174Ø4   15380 SYSP3    LD      HL,DCT$+4
292C CB7E     15390          BIT     7,(HL)          ;Check Ckdrv bit in DCT
292E CBBE     15400          RES     7,(HL)          ;Enable it for now
2930 2170Ø4   15410          LD      HL,DCT$
2933 119D2B   15420          LD      DE,FCB1         ;Use FCB space
2936 D5       15430          PUSH    DE              ;Save where temp
2937 E5       15440          PUSH    HL              ;Save where new SYSTEM
2938 Ø1ØA ØØ  15450          LD      BC,1Ø           ;Save current SYS's DCT
293B EDBØ     15460          LDIR
293D FDCBØ4BE 15470          RES     7,(IY+4)        ;Make sure OFF
2941 28Ø4     15480          JR      Z,NOSET         ;Only set if it was set
              15490                                  ;  on the SYSTEM drive
2943 FDCBØ4FE 15500          SET     7,(IY+4)        ;CKDRV inhibit drive Ø
2947 FDE5     15510 NOSET    PUSH    IY              ;New DCT to HL
2949 E1       15520          POP     HL
294A D1       15530          POP     DE              ;Rcvr SYS DCT now
294B E5       15540          PUSH    HL              ;Save where it goes
294C ØEØA     15550          LD      C,1Ø
294E EDBØ     15560          LDIR                    ;New DCT into SYSTEM
295Ø D1       15570          POP     DE              ;P/u where old SYS goes
2951 E1       15580          POP     HL              ;P/u temp address
2952 ØEØA     15590          LD      C,1Ø            ;Move old SYS to new DCT
2954 EDBØ     15600          LDIR
2956 21452A   15610          LD      HL,NWSYS$       ;"New sys drive installed
2959 C31724   15620          JP      SETERR          ;Display and exit w/abort
              15630                                  ;  in case JCL active
              15640 ;
              15650 ;        Routines to read/write the config sector
              15660 ;
295C 3E35     15670 PUTCFG   LD      A,53            ;@WRSEC
              15680 ;
              15690          IF      @MOD4
295E 21       15700          DB      21H             ;Ignore next with LD HL,
              15710          ENDIF
              15720          IF      @MOD2
              15730          JR      PUTGETC         ;Jump over if mod2
              15740          ENDIF
              15750 ;
295F          15760 GETCFG   EQU     $
              15770          IF      @MOD4
295F 3E31     15780          LD      A,49            ;@RDSEC
2961 11Ø2ØØ   15790          LD      DE,Ø<8+2        ;Config sector
              15800          ENDIF
              15810 ;
              15820          IF      @MOD2
              15830          LD      A,(DCT$+3)      ;Drive Ø dct data
              15840          AND     28H             ;Bit 5/3
              15850          CP      2ØH             ;8" floppy?
              15860          JR      NZ,SETSYS1      ;Go if not
              15870          LD      A,(DCT$+4)      ;Get data
              15880          AND     5ØH             ;Bit 6/4
              15890          CP      4ØH             ;DD not alien?
              15900          JR      NZ,SETSYS1      ;Go if not
              15910 ;
```

```
                15920           LD      HL,SBUFF$       ;System buffer
                15930           LD      A,(DCT$+9)      ;Get dir cylinder
                15940           LD      D,A             ;Pass to D
                15950           LD      E,Ø             ;Init sector Ø
                15960           LD      C,E             ;Init drive Ø
2964            15970           @@RDSSC                 ;Read directory
                ØØØ66           LD      A,85
                ØØØ67           RST     4Ø
                15980           JR      NZ,PUTGETE      ;Go on disk error
                15990           LD      A,(SBUFF$+ØCDH) ;Get gat data byte
                16ØØØ           BIT     7,A             ;Is data disk?
                16Ø1Ø SETSYS1   LD      DE,Ø<8+2        ;Init sysinfo sector
                16Ø2Ø           JR      NZ,$+3          ;Go if data disk
                16Ø3Ø           INC     D               ;Else new sysinfo
                16Ø4Ø           LD      A,49            ;Init RDSEC
                16Ø5Ø           ENDIF
                16Ø6Ø ;
2964 21ØØ1D     16Ø7Ø PUTGETC   LD      HL,SBUFF$       ;System buffer
2967 ØEØØ       16Ø8Ø           LD      C,Ø             ;  in system drive
2969 EF         16Ø9Ø           RST     4Ø              ;Read or Write
296A C8         16ØØØ           RET     Z               ;Back if good i/o
296B E1         16111 PUTGETE   POP     HL
296C C31F24     16121           JP      IOERR           ;  else abort
                16131 ;
                16141 ;         Routine to get HIGH$
                16151 ;
296F Ø6ØØ       16161 GETHI$    LD      B,Ø             ;Init to get HIGH$
2971 6Ø         16171           LD      H,B
2972 68         16181           LD      L,B
2973            16191           @@HIGH$                 ;Get current HIGH$
2973 3E64       ØØØ68           LD      A,1ØØ
2975 EF         ØØØ69           RST     4Ø
2976 C9         16211           RET
                16221 ;
                16231 ;         Routine to test if HIGH$ is frozen
                16241 ;
2977 3A6CØØ     16251 GOTMEM?   LD      A,(CFLAG$)      ;Check if memory can
297A ØF         16261           RRCA                    ;  be altered
297B C9         16271           RET
                16281 ;
297C 4E         16291 NOMEM$    DB      'No memory space available',CR
     6F 2Ø 6D 65 6D 6F 72 79
     2Ø 73 7Ø 61 63 65 2Ø 61
     76 61 69 6C 61 62 6C 65
     ØD
2996 5Ø         16301 PRMERR$   DB      'Parameter error',CR
     61 72 61 6D 65 74 65 72
     2Ø 65 72 72 6F 72 ØD
29A6 43         16311 NOCYL8$   DB      'CYL=c invalid on 8" drive',CR
     59 4C 3D 63 2Ø 69 6E 76
     61 6C 69 64 2Ø 6F 6E 2Ø
     38 22 2Ø 64 72 69 76 65
     ØD
29CØ 43         16321 BADCYL$   DB      'Cylinder count out of range <35-96>',CR
     79 6C 69 6E 64 65 72 2Ø
     63 6F 75 6E 74 2Ø 6F 75
     74 2Ø 6F 66 2Ø 72 61 6E
     67 65 2Ø 3C 33 35 2D 39
     36 3E ØD
29E4 43         16331 BADSYS$   DB      'Can''t SYSRES requested module',CR
     61 6E 27 74 2Ø 53 59 53
```

```
     52 45 53 2Ø 72 65 71 75
     65 73 74 65 64 2Ø 6D 6F
     64 75 6C 65 ØD
2AØ2 53       16330 MODRES$ DB      'System module already resident',CR
     79 73 74 65 6D 2Ø 6D 6F
     64 75 6C 65 2Ø 61 6C 72
     65 61 64 79 2Ø 72 65 73
     69 64 65 6E 74 ØD
2A21 45       16340 EIPMPT  DM      'Enter DCT driver <BREAK=default> : ',3'
     6E 74 65 72 2Ø 44 43 54
     2Ø 64 72 69 76 65 72 2Ø
     3C 42 52 45 41 4B 3D 64
     65 66 61 75 6C 74 3E 2Ø
     3A 2Ø Ø3
2A45 4E       16350 NWSYS$  DB      'New SYSTEM drive installed',CR
     65 77 2Ø 53 59 53 54 45
     4D 2Ø 64 72 69 76 65 2Ø
     69 6E 73 74 61 6C 6C 65
     64 ØD
2A6Ø 49       16360 NDSYS$  DB      'Insert SYSTEM diskette in drive X ',CR
     6E 73 65 72 74 2Ø 53 59
     53 54 45 4D 2Ø 64 69 73
     6B 65 74 74 65 2Ø 69 6E
     2Ø 64 72 69 76 65 2Ø 58
     2Ø ØD
2A83 42       16370 BADBS$  DB      'Boot step out of range <Ø-3>',CR
     6F 6F 74 2Ø 73 74 65 7Ø
     2Ø 6F 75 74 2Ø 6F 66 2Ø
     72 61 6E 67 65 2Ø 3C 3Ø
     2D 33 3E ØD
2AAØ 44       16380 DCTEXT  DM      'DCT'
     43 54
              16390 ;
2AA3          16400 PRMTBL$ EQU     $
2AA3 41       16410         DB      'ALIVE '
     4C 49 56 45 2Ø
2AA9 8D25     16420         DW      ALIVE+1
2AAB 42       16430         DB      'BSTEP '
     53 54 45 5Ø 2Ø
2AB1 C124     16440         DW      BSPARM+1
2AB3 44       16450         DM      'DRIVER'
     52 49 56 45 52
2AB9 3628     16460         DW      DVRPARM+1
2ABB 46       16470         DB      'FAST  '
     41 53 54 2Ø 2Ø
2AC1 4224     16480         DW      FPARM+1
2AC3 53       16490         DB      'SLOW  '
     4C 4F 57 2Ø 2Ø
2AC9 3B24     16500         DW      SLOW+1
2ACB 53       16510         DB      'SYSRES'
     59 53 52 45 53
2AD1 FB25     16520         DW      SYSRES+1
2AD3 53       16530         DB      'SYSTEM'
     59 53 54 45 4D
2AD9 DB28     16540         DW      SYSPRM+1
2ADB 54       16550         DB      'TYPE  '
     59 5Ø 45 2Ø 2Ø
2AE1 2A25     16560         DW      TYPE+1
2AE3 57       16570         DB      'WP    '
     5Ø 2Ø 2Ø 2Ø 2Ø
2AE9 6627     16580         DW      WPARM+1
```

```
2AEB 42          16590          DB      'BLINK '
     4C 49 4E 4B 2Ø
2AF1 E224        16600          DW      BLINK+1
2AF3 42          16610          DB      'BREAK '
     52 45 41 4B 2Ø
2AF9 7F24        16620          DW      BREAK+1
2AFB 43          16630          DB      'CYL   '
     59 4C 2Ø 2Ø 2Ø
2BØ1 7B27        16640          DW      CYLPRM+1
2BØ3 44          16650          DB      'DATE  '
     41 54 45 2Ø 2Ø
2BØ9 9324        16660          DW      DATE+1
2BØB 44          16670          DB      'DELAY '
     45 4C 41 59 2Ø
2B11 E527        16680          DW      DELAY+1
2B13 43          16690          DB      'CKDRV '
     4B 44 52 56 2Ø
2B19 FD27        16700          DW      CKDRV+1
2B1B 44          16710          DB      'DISABL'
     49 53 41 42 4C
2B21 AD27        16720          DW      DISABL+1
2B23 44          16730          DB      'DRIVE '
     52 49 56 45 2Ø
2B29 4327        16740          DW      DRIVE+1
2B2B 45          16750          DB      'ENABLE'
     4E 41 42 4C 45
2B31 BE27        16760          DW      ENABLE+1
2B33 47          16770          DB      'GRAPHI'
     52 41 5Ø 48 49
2B39 7825        16780          DW      GRAPHI+1
2B3B 4C          16790          DB      'LARGE '
     41 52 47 45 2Ø
2B41 FB24        16800          DW      LARGE+1
2B43 52          16810          DB      'RESTOR'
     45 53 54 4F 52
2B49 1325        16820          DW      RPARM+1
2B4B 53          16830          DB      'SMALL '
     4D 41 4C 4C 2Ø
2B51 Ø525        16840          DW      SMALPRM+1
2B53 53          16850          DB      'STEP  '
     54 45 5Ø 2Ø 2Ø
2B59 CA27        16860          DW      STEP+1
2B5B 54          16870          DB      'TIME  '
     49 4D 45 2Ø 2Ø
2B61 AA24        16880          DW      TIME+1
2B63 54          16890          DB      'TRACE '
     52 41 43 45 2Ø
2B69 6624        16900          DW      TRACE+1
2B6B 53          16910          DB      'SMOOTH'
     4D 4F 4F 54 48
2B71 3F25        16920          DW      SMOOTH+1
2B73 48          16930          DB      'HERTZ5'
     45 52 54 5A 35
2B79 5E25        16940          DW      HERTZ5Ø+1
2B7B 48          16950          DB      'HERTZ6'
     45 52 54 5A 36
2B81 6B25        16960          DW      HERTZ6Ø+1
2B83 ØØ          16970          NOP
                 16980 ;
2B84 21ØØØØ      16990 PATCH1   LD      HL,Ø            ;Set HL=Ø "NO ERRORS"
2B87 B1          17000          OR      C
```

```
2B88 3C          17Ø1Ø          INC     A               ;Parm entered?
2B89 C8          17Ø2Ø          RET     Z               ;Exit if not
2B8A C3E128      17Ø3Ø          JP      CONT1           ;Continue on now
                 17Ø4Ø ;
2B8D ØØ          17Ø5Ø          DC      16,Ø            ;Patch space
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ
2B9D ØØ          17Ø6Ø FCB1     DC      32,Ø
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
     ØØ ØØ ØØ ØØ ØØ ØØ ØØ
                 17Ø7Ø ;
24ØØ             ØØ11Ø          END     SYSTEM
```

$A1	Ø3B7	$A2	Ø3B8
$CKEOF	147Ø	@$SYS	Ø8FØ
@@2	ØØØØ	@@3	ØØØØ
@ABORT	1BØ8	@ADTSK	1CDA
@BKSP	1486	@BREAK	196F
@CHNIO	Ø689	@CKBRKC	Ø553
@CKEOF	158F	@CKTSK	1CF5
@CLS	Ø545	@CMNDI	197E
@CTL	Ø623	@DATE	Ø7A8
@DCINIT	19CØ	@DCRES	19C4
@DCTBYT	1A2B	@DEBUG	19AØ
@DIRCYL	18F7	@DIRRD	18BB
@DIV16	Ø6E3	@DIV8	1927
@DOKEY	19A9	@DSP	Ø642
@ERROR	1BØF	@EXIT	1BØB
@FLAGS	196A	@FNAME	199C
@FSPEC	1981	@GATRD	1874
@GERMAN	ØØØØ	@GET	Ø638
@GTDCT	1A1E	@GTMOD	19B2
@HEX16	Ø7BD	@HEX8	Ø7C2
@HIGH$	1948	@HITRD	1897
@HZ5Ø	ØØØØ	@ICNFG	ØØ86
@INTL	ØØØØ	@IPL	1BF2
@KBD	Ø635	@KEY	Ø628
@KITSK	ØØ89	@KLTSK	1CDØ
@LOC	14B3	@LOF	14DE
@LOGOT	Ø5ØØ	@MOD2	ØØØØ
@MSG	Ø53Ø	@MUL16	Ø6C9
@NMI	ØØ66	@OPEN	198A
@PARAM	1987	@PAUSE	Ø382
@POSN	1434	@PRINT	Ø528
@PUT	Ø645	@RAMDIR	19AC
@RDSEC	19F4	@RDSSC	18D8
@READ	1513	@REMOVE	19A6
@REW	149B	@RMTSK	1CD7
@RREAD	1473	@RSLCT	19D4
@RST Ø8	ØØØ8	@RST1Ø	ØØ1Ø
@RST2Ø	ØØ2Ø	@RST28	ØØ28
@RST38	ØØ38	@RSTNMI	ØFE9
@RSTREG	Ø68Ø	@RUN	1B1D
@SEEK	19DØ	@SEEKSC	1421
@SLCT	19BC	@SOUND	Ø392
@TIME	Ø78D	@USA	FFFF
@VDCTL3	ØD38	@VER	156Ø
@WEOF	14EC	@WHERE	1979
@WRSEC	19E8	@WRSSC	19EC
@_VDCTL	ØD42	ADDR_2_ROWCOL	ØDF1
AFLAG$	ØØ6A	ALIVE	258C
ALIVEON	259E	ALVBGN	25D1
ALVTCB	25DF	AUTO?	1FF1
BADCYL	279D	BADCYL$	29CØ
BADSYS$	29E4	BAR$	Ø2Ø1
BLINK	24E1	BLNKON	24F2
BREAK	247E	BREAK?	1C6Ø
BRKOFF	248E	BRKVEC$	1C88
BSPEND	24E1	BUR$	Ø2ØØ
CFCB$	ØØEØ	CFGFCB$	ØØEØ
CKDRV	27FC	CKDRV1	28ØB
CKMOD@	1A7F	CKOPEN@	1568
CONT1	28E1	CORE$	Ø3ØØ

$A2	Ø3B9
@$SYS	ØØØØ
@@1	ØØØØ
@@3	ØØØØ
@@4	Ø877
@ADTSK	13ØØ
@BANK	1993
@BREAK	1999
@BYTEIO	197B
@CKBRKC	199F
@CKDRV	19B5
@CKTSK	Ø3E1
@CLOSE	18Ø3
@CMNDI	19AF
@CMNDR	Ø52D
@DATE	1984
@DBGHK	ØØØØ
@DCRES	1875
@DCSTAT	199Ø
@DEBUG	19E4
@DECHEX	Ø6F6
@DIRCYL	1898
@DIRRD	198D
@DIRWR	Ø63Ø
@DIV8	Ø585
@DODIR	1B38
@DSP	Ø5Ø3
@DSPLY	FFFF
@EXIT	19ØA
@FEXT	ØØ84
@FNAME	14A2
@FRENCH	Ø63D
@GATRD	19D8
@GATWR	19EØ
@GET	1996
@GTDCB	1CEB
@GTMOD	ØØØØ
@HDFMT	ØØ18
@HEX8	ØØ3Ø
@HEXDEC	19C8
@HITRD	13AD
@HITWR	143Ø
@INIT	19CC
@IPL	ØB99
@JCL	19DC
@KEY	1531
@KEYIN	19FØ
@LOAD	275B
@LOGER	25F2
@MOD2	25FA
@MOD4	2A83
@MUL16	2642
@MUL8	2512
@OPEN	439D
@OPREG	2492
@PAUSE	24CØ
@PEOF	ØA7B
@PRINT	ØØ6C
@PRT	275B
@RAMDIR	2Ø3F
@RDHDR	ØØØD

CRTBGN$	F8ØØ	CYL1	2793	CYLEND	27AC
CYLPRM	277A	CYL_GRN	16AE	D@FBYT8	1A26
DATE	2492	DATE$	ØØ33	DATEND	24A9
DAYTBL$	Ø4C7	DBGSV$	ØØAØ	DCBKL$	ØØ31
DCT$	Ø47Ø	DCTBYT8@	1A29	DCTEXT	2AAØ
DCTFLD@	1A34	DDCT1	287A	DDCT2	2884
DELAY	27E4	DELAY1	27F3	DFLAG$	ØØ6D
DIRBUF$	23ØØ	DISABL	27AC	DIS_DO_RAM	Ø846
DODATA$	ØB94	DODCB$	Ø21Ø	DOFF	249F
DO_CONTROL	ØC44	DO_DSPCHAR	ØCB8	DO_INVERT_DIS	ØC8C
DO_INVERT_ENA	ØC89	DO_INVERT_OFF	ØC9B	DO_MASK	ØØØØ
DO_RET	ØBCB	DO_RET1	ØBCC	DO_SCROLL	ØCCE
DO_TABS	ØBEA	DRIVE	2742	DRVLOOP	2765
DSKTYP$	Ø4CØ	DTPMT$	Ø4C2	DVR2	28A7
DVR4	28D9	DVRA	2885	DVRB	288E
DVRC	289A	DVREND$	ØFF4	DVRHI$	Ø2Ø6
DVRPARM	2835	EFLAG$	ØØ6E	EIPMPT	2A21
ENABLE	27BD	ENADIS	27C6	ENADIS_DO_RAM	Ø817
EXTDBG$	19A4	FCB1	2B9D	FDDINT$	ØØØE
FEMSK$	ØØ6F	FLGTAB$	ØØ6A	FPARM	2441
FPEND	2465	FRCERR	241A	GETCFG	295F
GETHI$	296F	GETN1	2844	GETN2	285E
GETN3	2863	GETSEC2	273B	GET_@_ROWCOL	ØDAE
GOTMEM?	2977	GRAEND	258C	GRAPHI	2577
GROFF	2588	GUDBS	24D3	GUDCYL	27A9
GUDSYS	262Ø	HERTZ$	Ø75Ø	HERTZ5Ø	255D
HERTZ6Ø	256A	HIGH$	Ø4ØE	HKRES$	1A6C
HZ5ØEND	256A	HZ6ØEND	2577	IFLAG$	ØØ72
INBUF$	Ø42Ø	INTIM$	ØØ3C	INTMSK$	ØØ3D
INTVC$	ØØ3E	IOERR	241F	JCLCB$	Ø2Ø3
JDCB$	ØØ24	JFCB$	ØØCØ	JLDCB$	Ø23Ø
JRET$	ØØ26	KCK@	Ø7D6	KFLAG$	ØØ74
KIDATA$	Ø8FC	KIDCB$	Ø2Ø8	LARGE	24FA
LBANK$	Ø2Ø2	LDRV$	ØØ23	LFLAG$	ØØ75
LNKFCB@	1566	LOW$	ØØ1E	LSVC$	ØØØD
MAXCOR$	24ØØ	MAXDAY$	Ø4Ø1	MINCOR$	3ØØØ
MODOUT$	ØØ76	MODPORT	ØØEC	MODRES$	2AØ2
MONTBL$	Ø4DC	MOVITIN	2659	NDSYS$	2A6Ø
NFLAG$	ØØ77	NOCYL8$	29A6	NOMEM	241Ø
NOMEM$	297C	NOSET	2947	NOTRES	2736
NWSYS$	2A45	OPREG$	ØØ78	OPREG_SV_AREA	Ø86E
OPREG_SV_PTR	Ø835	ORARET@	14DC	OSRLS$	ØØ3B
OSVER$	ØØ85	OVRLY$	ØØ69	PAKNAM$	Ø41Ø
PATCH1	2B84	PAUSE@	Ø382	PCSAVE$	Ø7AF
PDRV$	ØØ1B	PHIGH$	ØØ1C	PRDCB$	Ø218
PRMERR	2414	PRMERR$	2996	PRMTBL$	2AA3
PUTA@DE	ØDCD	PUTCFG	295C	PUTDVR	26AØ
PUTGETC	2964	PUTGETE	296B	PUTSEC2	282E
PUT_@	ØDCA	PUT_@_ROWCOL	ØDC6	REND	2529
RESI	264C	RESIE	2658	RESBGN	26CD
RESD1	27Ø7	RESDVR	26FC	RESEND	273B
RESTAB$	26DC	RFLAG$	ØØ7B	ROWCOL_2_ADDR	ØDDØ
RPARM	2512	RST38@	1BFF	RSTOR$	Ø4C4
RSTRON	251E	RWRIT@	13A2	S1DCB$	Ø238
SAVESP	24Ø9	SBUFF$	1DØØ	SET@EXEC	1A79
SETBLK	25ØC	SETCLK	245C	SETERR	2417
SETSLOW	2454	SET_SCROLL	ØCF3	SFCB$	ØØ8C
SFLAG$	ØØ7C	SIDCB$	Ø22Ø	SLOW	243A
SMALPRM	25Ø4	SMEND	255D	SMON	2554
SMOOTH	253E	SODCB$	Ø228	SPACE4$	2142
STACK$	Ø38Ø	START$	ØØØØ	STEP	27C9

```
STEPEND         27E4 SVCRET$         ØØØB SVCTAB$         Ø1ØØ
SYSERR$         1B13 SYSP1           28EF SYSP2           2911
SYSP3           2929 SYSPRM          28DA SYSRES          25FA
SYSTEM          24ØØ SYSTEM1         242A TCB$            ØØ4E
TEST6           2614 TFLAG$          ØØ7D TIME            24A9
TIME$           ØØ2D TIMEND          24CØ TIMER$          ØØ2C
TIMSL$          ØØ2B TIMTSK$         Ø713 TMPMT$          Ø4C3
TOFF            24B6 TRACE           2465 TRACE_INT       Ø7B1
TREND           247E TROFF           2479 TYPE            2529
TYPEND          253E TYPEON          253A TYPHK$          ØA8F
TYPTSK$         ØB26 USTOR$          ØØ13 VFLAG$          ØØ7F
WDESD           2814 WPARM           2765 WPEND           277A
WPOFF           2776 WPON            277Ø WRINT$          ØØ8Ø
ZERO$           Ø4Ø1 ZEROA@          13AØ @@ABORT         6BFF
@@ADTSK         6C92 @@BANK          71AA @@BKSP          6E8A
@@BREAK         71CØ @@CHNIO         6BEA @@CKBRKC        72ØE
@@CKDRV         6CE6 @@CKEOF         6E9F @@CKTSK         6C7D
@@CLOSE         6E75 @@CLS           71F8 @@CMNDI         6C29
@@CMNDR         6C3E @@CTL           6A4E @@DATE          6BCØ
@@DCSTAT        6D25 @@DEBUG         6C68 @@DECHEX        712A
@@DIRRD         7Ø97 @@DIRWR         7ØAC @@DIV16         7115
@@DIV8          71ØØ @@DODIR         6CFB @@DSP           6A12
@@DSPLY         6AB2 @@ERROR         6C53 @@EXIT          6C14
@@FEXT          7ØØ4 @@FLAGS         7194 @@FNAME         7Ø19
@@FSPEC         6FEF @@GATRD         7Ø82 @@GATWR         7ØC1
@@GET           6A26 @@GTDCB         7Ø43 @@GTDCT         7Ø2E
@@GTMOD         7Ø58 @@HDFMT         6DCD @@HEX16         7169
@@HEX8          7154 @@HEXDEC        713F @@HIGH$         717E
@@INIT          6E4B @@KBD           6A8A @@KEY           69FE
@@KEYIN         6A9E @@KLTSK         6CD1 @@LOAD          6FC5
@@LOC           6EB4 @@LOF           6EC9 @@LOGER         6AE9
@@LOGOT         6AFE @@MSG           6B35 @@MUL16         7ØEB
@@MUL8          7ØD6 @@OPEN          6E6Ø @@PARAM         6BAB
@@PAUSE         6B96 @@PEOF          6EDE @@POSN          6EF3
@@PRINT         6B4A @@PRT           6A62 @@PUT           6A3A
@@RAMDIR        6D1Ø @@RDSEC         6DA3 @@RDSSC         7Ø6D
@@READ          6FØ8 @@REMOV         6E36 @@RENAM         6E21
@@REW           6F1D @@RMTSK         6CA7 @@RPTSK         6CBC
@@RREAD         6F32 @@RSLCT         6D8E @@RSTOR         6D4F
@@RUN           6FDA @@RWRIT         6F47 @@SEEK          6D79
@@SEEKSC        6F5C @@SKIP          6F71 @@SLCT          6D3A
@@STEPI         6D64 @@TIME          6BD5 @@VDCTL         6B81
@@VER           6F86 @@VRSEC         6DB8 @@WEOF          6F9B
@@WHERE         6A76 @@WRITE         6FBØ @@WRSEC         6DE2
@@WRSSC         6DF7 @@WRTRK         6EØC
24ØØ is the transfer address
ØØØØØ Total errors
```

NOTES:

NOTES:

NOTES:

NOTES:

NOTES:

```
                ØØ1ØØ ;LDOS6Ø/EQU - Equates from cross reference of Lowcore
ØØØØ            ØØ11Ø          TITLE   <LDOS6Ø/EQU>
                ØØ12Ø ;
Ø8FØ            ØØ13Ø @$SYS    EQU     Ø8FØH
ØØØØ            ØØ14Ø @@1      DEFL    ØØØØH
ØØØØ            ØØ15Ø @@2      DEFL    ØØØØH
ØØØØ            ØØ16Ø @@3      DEFL    ØØØØH
ØØØØ            ØØ17Ø @@4      DEFL    ØØØØH
Ø877            ØØ18Ø @BANK    EQU     Ø877H
13ØØ            ØØ19Ø @BYTEIO  EQU     13ØØH
Ø689            ØØ2ØØ @CHNIO   EQU     Ø689H
Ø553            ØØ21Ø @CKBRKC  EQU     Ø553H
Ø545            ØØ22Ø @CLS     EQU     Ø545H
Ø623            ØØ23Ø @CTL     EQU     Ø623H
Ø7A8            ØØ24Ø @DATE    EQU     Ø7A8H
Ø6E3            ØØ25Ø @DIV16   EQU     Ø6E3H
Ø642            ØØ26Ø @DSP     EQU     Ø642H
Ø52D            ØØ27Ø @DSPLY   EQU     Ø52DH
ØØØØ            ØØ28Ø @FRENCH  EQU     ØØØØH
ØØØØ            ØØ29Ø @GERMAN  EQU     ØØØØH
Ø638            ØØ3ØØ @GET     EQU     Ø638H
Ø7BD            ØØ31Ø @HEX16   EQU     Ø7BDH
Ø7C2            ØØ32Ø @HEX8    EQU     Ø7C2H
Ø6F6            ØØ33Ø @HEXDEC  EQU     Ø6F6H
ØØØØ            ØØ34Ø @HZ5Ø    EQU     ØØØØH
ØØØØ            ØØ35Ø @INTL    EQU     ØØØØH
Ø63Ø            ØØ36Ø @JCL     EQU     Ø63ØH
Ø635            ØØ37Ø @KBD     EQU     Ø635H
Ø628            ØØ38Ø @KEY     EQU     Ø628H
Ø585            ØØ39Ø @KEYIN   EQU     Ø585H
ØØ89            ØØ4ØØ @KITSK   EQU     ØØ89H
Ø5Ø3            ØØ41Ø @LOGER   EQU     Ø5Ø3H
Ø5ØØ            ØØ42Ø @LOGOT   EQU     Ø5ØØH
ØØØØ            ØØ43Ø @MOD2    EQU     ØØØØH
FFFF            ØØ44Ø @MOD4    EQU     ØFFFFH
Ø53Ø            ØØ45Ø @MSG     EQU     Ø53ØH
Ø6C9            ØØ46Ø @MUL16   EQU     Ø6C9H
ØØ84            ØØ47Ø @OPREG   EQU     ØØ84H
Ø528            ØØ48Ø @PRINT   EQU     Ø528H
Ø63D            ØØ49Ø @PRT     EQU     Ø63DH
Ø645            ØØ5ØØ @PUT     EQU     Ø645H
ØFE9            ØØ51Ø @RSTNMI  EQU     ØFE9H
Ø68Ø            ØØ52Ø @RSTREG  EQU     Ø68ØH
Ø78D            ØØ53Ø @TIME    EQU     Ø78DH
FFFF            ØØ54Ø @USA     EQU     ØFFFFH
ØB99            ØØ55Ø @VDCTL   EQU     ØB99H
ØD38            ØØ56Ø @VDCTL3  EQU     ØD38H
ØD42            ØØ57Ø @_VDCTL  EQU     ØD42H
ØDF1            ØØ58Ø ADDR_2_ROWCOL    EQU     ØDF1H
Ø2Ø1            ØØ59Ø BAR$     EQU     Ø2Ø1H
439D            ØØ6ØØ BOOTST$  EQU     439DH
Ø2ØØ            ØØ61Ø BUR$     EQU     Ø2ØØH
ØA7B            ØØ62Ø CASHK$   EQU     ØA7BH
ØØ6C            ØØ63Ø CFLAG$   EQU     ØØ6CH
Ø3ØØ            ØØ64Ø CORE$    DEFL    Ø3ØØH
F8ØØ            ØØ65Ø CRTBGN$  EQU     ØF8ØØH
ØØ33            ØØ66Ø DATE$    EQU     ØØ33H
Ø4C7            ØØ67Ø DAYTBL$  EQU     Ø4C7H
ØØ31            ØØ68Ø DCBKL$   EQU     ØØ31H
Ø47Ø            ØØ69Ø DCT$     EQU     Ø47ØH
ØØ6D            ØØ7ØØ DFLAG$   EQU     ØØ6DH
```

```
0846          00710 DIS_DO_RAM      EQU     0846H
0B94          00720 DODATA$ EQU     0B94H
0210          00730 DODCB$  EQU     0210H
0C44          00740 DO_CONTROL      EQU     0C44H
0CB8          00750 DO_DSPCHAR      EQU     0CB8H
0C8C          00760 DO_INVERT_DIS   EQU     0C8CH
0C89          00770 DO_INVERT_ENA   EQU     0C89H
0C9B          00780 DO_INVERT_OFF   EQU     0C9BH
0000          00790 DO_MASK EQU     0000H
0BCB          00800 DO_RET  EQU     0BCBH
0BCC          00810 DO_RET1 EQU     0BCCH
0CCE          00820 DO_SCROLL       EQU     0CCEH
0BEA          00830 DO_TABS EQU     0BEAH
04C0          00840 DSKTYP$ EQU     04C0H
04C2          00850 DTPMT$  EQU     04C2H
0FF4          00860 DVREND$ EQU     0FF4H
0206          00870 DVRHI$  EQU     0206H
0817          00880 ENADIS_DO_RAM   EQU     0817H
000E          00890 FDDINT$ EQU     000EH
006A          00900 FLGTAB$ EQU     006AH
0DAE          00910 GET_@_ROWCOL    EQU     0DAEH
0750          00920 HERTZ$  EQU     0750H
040E          00930 HIGH$   EQU     040EH
0072          00940 IFLAG$  EQU     0072H
0420          00950 INBUF$  EQU     0420H
003E          00960 INTVC$  EQU     003EH
0203          00970 JCLCB$  EQU     0203H
0230          00980 JLDCB$  EQU     0230H
07D6          00990 KCK@    EQU     07D6H
0074          01000 KFLAG$  EQU     0074H
08FC          01010 KIDATA$ EQU     08FCH
0208          01020 KIDCB$  EQU     0208H
0202          01030 LBANK$  EQU     0202H
0401          01040 MAXDAY$ EQU     0401H
0076          01050 MODOUT$ EQU     0076H
04DC          01060 MONTBL$ EQU     04DCH
0077          01070 NFLAG$  EQU     0077H
0078          01080 OPREG$  EQU     0078H
086E          01090 OPREG_SV_AREA   EQU     086EH
0835          01100 OPREG_SV_PTR    EQU     0835H
0410          01110 PAKNAM$ EQU     0410H
0382          01120 PAUSE@  EQU     0382H
07AF          01130 PCSAVE$ EQU     07AFH
001B          01140 PDRV$   EQU     001BH
0218          01150 PRDCB$  EQU     0218H
0DCD          01160 PUTA@DE EQU     0DCDH
0DCA          01170 PUT_@   EQU     0DCAH
0DC6          01180 PUT_@_ROWCOL    EQU     0DC6H
007B          01190 RFLAG$  EQU     007BH
0DD0          01200 ROWCOL_2_ADDR   EQU     0DD0H
04C4          01210 RSTOR$  EQU     04C4H
0238          01220 S1DCB$  EQU     0238H
0CF3          01230 SET_SCROLL      EQU     0CF3H
007C          01240 SFLAG$  EQU     007CH
0220          01250 SIDCB$  EQU     0220H
0228          01260 SODCB$  EQU     0228H
0380          01270 STACK$  EQU     0380H
0000          01280 START$  EQU     0000H
002D          01290 TIME$   EQU     002DH
002C          01300 TIMER$  EQU     002CH
002B          01310 TIMSL$  EQU     002BH
```

```
0713          01320 TIMTSK$ EQU     0713H
04C3          01330 TMPMT$  EQU     04C3H
07B1          01340 TRACE_INT       EQU     07B1H
0A8F          01350 TYPHK$  EQU     0A8FH
0B26          01360 TYPTSK$ EQU     0B26H
007F          01370 VFLAG$  EQU     007FH
0401          01380 ZERO$   EQU     0401H
No end statement
00000 Total errors
```

```
                ØØ1ØØ ;SYSØ/EQU - Equates from cross reference of Sysres
ØØØØ            ØØ11Ø          TITLE   <SYSØ/EQU>
                ØØ12Ø ;
Ø3B7            ØØ13Ø $A1      EQU     Ø3B7H
Ø3B8            ØØ14Ø $A2      EQU     Ø3B8H
Ø3B9            ØØ15Ø $A3      EQU     Ø3B9H
147Ø            ØØ16Ø $CKEOF   EQU     147ØH
Ø8FØ            ØØ17Ø @$SYS    EQU     Ø8FØH
ØØØØ            ØØ18Ø @@1      DEFL    ØØØØH
ØØØØ            ØØ19Ø @@1      DEFL    ØØØØH
ØØØØ            ØØ2ØØ @@2      DEFL    ØØØØH
ØØØØ            ØØ21Ø @@2      DEFL    ØØØØH
ØØØØ            ØØ22Ø @@3      DEFL    ØØØØH
ØØØØ            ØØ23Ø @@3      DEFL    ØØØØH
ØØØØ            ØØ24Ø @@4      DEFL    ØØØØH
ØØØØ            ØØ25Ø @@4      DEFL    ØØØØH
1BØ8            ØØ26Ø @ABORT   EQU     1BØ8H
1CDA            ØØ27Ø @ADTSK   EQU     1CDAH
Ø877            ØØ28Ø @BANK    EQU     Ø877H
1486            ØØ29Ø @BKSP    EQU     1486H
196F            ØØ3ØØ @BREAK   EQU     196FH
13ØØ            ØØ31Ø @BYTEIO  EQU     13ØØH
Ø689            ØØ32Ø @CHNIO   EQU     Ø689H
Ø553            ØØ33Ø @CKBRKC  EQU     Ø553H
1993            ØØ34Ø @CKDRV   EQU     1993H
158F            ØØ35Ø @CKEOF   EQU     158FH
1CF5            ØØ36Ø @CKTSK   EQU     1CF5H
1999            ØØ37Ø @CLOSE   EQU     1999H
Ø545            ØØ38Ø @CLS     EQU     Ø545H
197E            ØØ39Ø @CMNDI   EQU     197EH
197B            ØØ4ØØ @CMNDR   EQU     197BH
Ø623            ØØ41Ø @CTL     EQU     Ø623H
Ø7A8            ØØ42Ø @DATE    EQU     Ø7A8H
199F            ØØ43Ø @DBGHK   EQU     199FH
19CØ            ØØ44Ø @DCINIT  EQU     19CØH
19C4            ØØ45Ø @DCRES   EQU     19C4H
19B5            ØØ46Ø @DCSTAT  EQU     19B5H
1A2B            ØØ47Ø @DCTBYT  EQU     1A2BH
19AØ            ØØ48Ø @DEBUG   EQU     19AØH
Ø3E1            ØØ49Ø @DECHEX  EQU     Ø3E1H
18F7            ØØ5ØØ @DIRCYL  EQU     18F7H
18BB            ØØ51Ø @DIRRD   EQU     18BBH
18Ø3            ØØ52Ø @DIRWR   EQU     18Ø3H
Ø6E3            ØØ53Ø @DIV16   EQU     Ø6E3H
1927            ØØ54Ø @DIV8    EQU     1927H
19AF            ØØ55Ø @DODIR   EQU     19AFH
19A9            ØØ56Ø @DOKEY   EQU     19A9H
Ø642            ØØ57Ø @DSP     EQU     Ø642H
Ø52D            ØØ58Ø @DSPLY   EQU     Ø52DH
1BØF            ØØ59Ø @ERROR   EQU     1BØFH
1BØB            ØØ6ØØ @EXIT    EQU     1BØBH
1984            ØØ61Ø @FEXT    EQU     1984H
196A            ØØ62Ø @FLAGS   EQU     196AH
199C            ØØ63Ø @FNAME   EQU     199CH
ØØØØ            ØØ64Ø @FRENCH  EQU     ØØØØH
1981            ØØ65Ø @FSPEC   EQU     1981H
1874            ØØ66Ø @GATRD   EQU     1874H
1875            ØØ67Ø @GATWR   EQU     1875H
ØØØØ            ØØ68Ø @GERMAN  EQU     ØØØØH
Ø638            ØØ69Ø @GET     EQU     Ø638H
199Ø            ØØ7ØØ @GTDCB   EQU     199ØH
```

```
1A1E          ØØ71Ø @GTDCT  EQU     1A1EH
19B2          ØØ72Ø @GTMOD  EQU     19B2H
19E4          ØØ73Ø @HDFMT  EQU     19E4H
Ø7BD          ØØ74Ø @HEX16  EQU     Ø7BDH
Ø7C2          ØØ75Ø @HEX8   EQU     Ø7C2H
Ø6F6          ØØ76Ø @HEXDEC EQU     Ø6F6H
1948          ØØ77Ø @HIGH$  EQU     1948H
1897          ØØ78Ø @HITRD  EQU     1897H
1898          ØØ79Ø @HITWR  EQU     1898H
ØØØØ          ØØ8ØØ @HZ5Ø   EQU     ØØØØH
ØØ86          ØØ81Ø @ICNFG  EQU     ØØ86H
198D          ØØ82Ø @INIT   EQU     198DH
ØØØØ          ØØ83Ø @INTL   EQU     ØØØØH
1BF2          ØØ84Ø @IPL    EQU     1BF2H
Ø63Ø          ØØ85Ø @JCL    EQU     Ø63ØH
Ø635          ØØ86Ø @KBD    EQU     Ø635H
Ø628          ØØ87Ø @KEY    EQU     Ø628H
Ø585          ØØ88Ø @KEYIN  EQU     Ø585H
ØØ89          ØØ89Ø @KITSK  EQU     ØØ89H
ØØ89          ØØ9ØØ @KITSK  EQU     ØØ89H
1CDØ          ØØ91Ø @KLTSK  EQU     1CDØH
1B38          ØØ92Ø @LOAD   EQU     1B38H
14B3          ØØ93Ø @LOC    EQU     14B3H
14DE          ØØ94Ø @LOF    EQU     14DEH
Ø5Ø3          ØØ95Ø @LOGER  EQU     Ø5Ø3H
Ø5ØØ          ØØ96Ø @LOGOT  EQU     Ø5ØØH
ØØØØ          ØØ97Ø @MOD2   EQU     ØØØØH
FFFF          ØØ98Ø @MOD4   EQU     ØFFFFH
Ø53Ø          ØØ99Ø @MSG    EQU     Ø53ØH
Ø6C9          Ø1ØØØ @MUL16  EQU     Ø6C9H
19ØA          Ø1Ø1Ø @MUL8   EQU     19ØAH
ØØ66          Ø1Ø2Ø @NMI    EQU     ØØ66H
198A          Ø1Ø3Ø @OPEN   EQU     198AH
ØØ84          Ø1Ø4Ø @OPREG  EQU     ØØ84H
1987          Ø1Ø5Ø @PARAM  EQU     1987H
Ø382          Ø1Ø6Ø @PAUSE  EQU     Ø382H
14A2          Ø1Ø7Ø @PEOF   EQU     14A2H
1434          Ø1Ø8Ø @POSN   EQU     1434H
Ø528          Ø1Ø9Ø @PRINT  EQU     Ø528H
Ø63D          Ø11ØØ @PRT    EQU     Ø63DH
Ø645          Ø111Ø @PUT    EQU     Ø645H
19AC          Ø112Ø @RAMDIR EQU     19ACH
19D8          Ø113Ø @RDHDR  EQU     19D8H
19F4          Ø114Ø @RDSEC  EQU     19F4H
18D8          Ø115Ø @RDSSC  EQU     18D8H
19EØ          Ø116Ø @RDTRK  EQU     19EØH
1513          Ø117Ø @READ   EQU     1513H
19A6          Ø118Ø @REMOVE EQU     19A6H
1996          Ø119Ø @RENAME EQU     1996H
149B          Ø12ØØ @REW    EQU     149BH
1CD7          Ø121Ø @RMTSK  EQU     1CD7H
1CEB          Ø122Ø @RPTSK  EQU     1CEBH
1473          Ø123Ø @RREAD  EQU     1473H
19D4          Ø124Ø @RSLCT  EQU     19D4H
ØØØØ          Ø125Ø @RSTØØ  EQU     ØØØØH
ØØØ8          Ø126Ø @RSTØ8  EQU     ØØØ8H
ØØ1Ø          Ø127Ø @RST1Ø  EQU     ØØ1ØH
ØØ18          Ø128Ø @RST18  EQU     ØØ18H
ØØ2Ø          Ø129Ø @RST2Ø  EQU     ØØ2ØH
ØØ28          Ø13ØØ @RST28  EQU     ØØ28H
ØØ3Ø          Ø131Ø @RST3Ø  EQU     ØØ3ØH
```

```
ØØ38          Ø132Ø @RST38  EQU     ØØ38H
ØFE9          Ø133Ø @RSTNMI EQU     ØFE9H
19C8          Ø134Ø @RSTOR  EQU     19C8H
Ø68Ø          Ø135Ø @RSTREG EQU     Ø68ØH
1B1D          Ø136Ø @RUN    EQU     1B1DH
13AD          Ø137Ø @RWRIT  EQU     13ADH
19DØ          Ø138Ø @SEEK   EQU     19DØH
1421          Ø139Ø @SEEKSC EQU     1421H
143Ø          Ø14ØØ @SKIP   EQU     143ØH
19BC          Ø141Ø @SLCT   EQU     19BCH
Ø392          Ø142Ø @SOUND  EQU     Ø392H
19CC          Ø143Ø @STEPI  EQU     19CCH
Ø78D          Ø144Ø @TIME   EQU     Ø78DH
FFFF          Ø145Ø @USA    EQU     ØFFFFH
ØB99          Ø146Ø @VDCTL  EQU     ØB99H
ØD38          Ø147Ø @VDCTL3 EQU     ØD38H
156Ø          Ø148Ø @VER    EQU     156ØH
19DC          Ø149Ø @VRSEC  EQU     19DCH
14EC          Ø15ØØ @WEOF   EQU     14ECH
1979          Ø151Ø @WHERE  EQU     1979H
1531          Ø152Ø @WRITE  EQU     1531H
19E8          Ø153Ø @WRSEC  EQU     19E8H
19EC          Ø154Ø @WRSSC  EQU     19ECH
19FØ          Ø155Ø @WRTRK  EQU     19FØH
ØD42          Ø156Ø @_VDCTL EQU     ØD42H
ØDF1          Ø157Ø ADDR_2_ROWCOL   EQU     ØDF1H
ØØ6A          Ø158Ø AFLAG$  EQU     ØØ6AH
1FF1          Ø159Ø AUTO?   EQU     1FF1H
Ø2Ø1          Ø16ØØ BAR$    EQU     Ø2Ø1H
439D          Ø161Ø BOOTST$ EQU     439DH
1C6Ø          Ø162Ø BREAK?  EQU     1C6ØH
1C88          Ø163Ø BRKVEC$ EQU     1C88H
Ø2ØØ          Ø164Ø BUR$    EQU     Ø2ØØH
ØA7B          Ø165Ø CASHK$  EQU     ØA7BH
ØØEØ          Ø166Ø CFCB$   EQU     ØØEØH
ØØEØ          Ø167Ø CFGFCB$ EQU     ØØEØH
ØØ6C          Ø168Ø CFLAG$  EQU     ØØ6CH
ØØ6C          Ø169Ø CFLAG$  EQU     ØØ6CH
1A7F          Ø17ØØ CKMOD@  EQU     1A7FH
1568          Ø171Ø CKOPEN@ EQU     1568H
2Ø3F          Ø172Ø CONFIG$ EQU     2Ø3FH
1CFF          Ø173Ø CORE$   DEFL    1CFFH
1BFF          Ø174Ø CORE$   DEFL    1BFFH
1948          Ø175Ø CORE$   DEFL    1948H
1948          Ø176Ø CORE$   DEFL    1948H
Ø3ØØ          Ø177Ø CORE$   DEFL    Ø3ØØH
F8ØØ          Ø178Ø CRTBGN$ EQU     ØF8ØØH
16AE          Ø179Ø CYL_GRN EQU     16AEH
1A26          Ø18ØØ D@FBYT8 EQU     1A26H
ØØ33          Ø181Ø DATE$   EQU     ØØ33H
ØØ33          Ø182Ø DATE$   EQU     ØØ33H
Ø4C7          Ø183Ø DAYTBL$ EQU     Ø4C7H
ØØAØ          Ø184Ø DBGSV$  EQU     ØØAØH
ØØ31          Ø185Ø DCBKL$  EQU     ØØ31H
Ø47Ø          Ø186Ø DCT$    EQU     Ø47ØH
1A29          Ø187Ø DCTBYT8@        EQU     1A29H
1A34          Ø188Ø DCTFLD@ EQU     1A34H
ØØ6D          Ø189Ø DFLAG$  EQU     ØØ6DH
ØØ6D          Ø19ØØ DFLAG$  EQU     ØØ6DH
23ØØ          Ø191Ø DIRBUF$ EQU     23ØØH
Ø846          Ø192Ø DIS_DO_RAM      EQU     Ø846H
```

```
ØB94          Ø193Ø DODATA$ EQU     ØB94H
Ø21Ø          Ø194Ø DODCB$  EQU     Ø21ØH
ØC44          Ø195Ø DO_CONTROL      EQU     ØC44H
ØCB8          Ø196Ø DO_DSPCHAR      EQU     ØCB8H
ØC8C          Ø197Ø DO_INVERT_DIS   EQU     ØC8CH
ØC89          Ø198Ø DO_INVERT_ENA   EQU     ØC89H
ØC9B          Ø199Ø DO_INVERT_OFF   EQU     ØC9BH
ØØØØ          Ø2ØØØ DO_MASK EQU     ØØØØH
ØBCB          Ø2Ø1Ø DO_RET  EQU     ØBCBH
ØBCC          Ø2Ø2Ø DO_RET1 EQU     ØBCCH
ØCCE          Ø2Ø3Ø DO_SCROLL       EQU     ØCCEH
ØBEA          Ø2Ø4Ø DO_TABS EQU     ØBEAH
Ø4CØ          Ø2Ø5Ø DSKTYP$ EQU     Ø4CØH
Ø4C2          Ø2Ø6Ø DTPMT$  EQU     Ø4C2H
ØFF4          Ø2Ø7Ø DVREND$ EQU     ØFF4H
Ø2Ø6          Ø2Ø8Ø DVRHI$  EQU     Ø2Ø6H
ØØ6E          Ø2Ø9Ø EFLAG$  EQU     ØØ6EH
Ø817          Ø21ØØ ENADIS_DO_RAM   EQU     Ø817H
19A4          Ø211Ø EXTDBG$ EQU     19A4H
ØØØE          Ø212Ø FDDINT$ EQU     ØØØEH
ØØØE          Ø213Ø FDDINT$ EQU     ØØØEH
ØØ6F          Ø214Ø FEMSK$  EQU     ØØ6FH
ØØ6A          Ø215Ø FLGTAB$ EQU     ØØ6AH
ØØ6A          Ø216Ø FLGTAB$ EQU     ØØ6AH
ØDAE          Ø217Ø GET_@_ROWCOL    EQU     ØDAEH
Ø75Ø          Ø218Ø HERTZ$  EQU     Ø75ØH
Ø4ØE          Ø219Ø HIGH$   EQU     Ø4ØEH
1A6C          Ø22ØØ HKRES$  EQU     1A6CH
ØØ72          Ø221Ø IFLAG$  EQU     ØØ72H
ØØ72          Ø222Ø IFLAG$  EQU     ØØ72H
Ø42Ø          Ø223Ø INBUF$  EQU     Ø42ØH
ØØ3C          Ø224Ø INTIM$  EQU     ØØ3CH
ØØ3D          Ø225Ø INTMSK$ EQU     ØØ3DH
ØØ3E          Ø226Ø INTVC$  EQU     ØØ3EH
ØØ3E          Ø227Ø INTVC$  EQU     ØØ3EH
Ø2Ø3          Ø228Ø JCLCB$  EQU     Ø2Ø3H
ØØ24          Ø229Ø JDCB$   EQU     ØØ24H
ØØCØ          Ø23ØØ JFCB$   EQU     ØØCØH
Ø23Ø          Ø231Ø JLDCB$  EQU     Ø23ØH
ØØ26          Ø232Ø JRET$   EQU     ØØ26H
Ø7D6          Ø233Ø KCK@    EQU     Ø7D6H
ØØ74          Ø234Ø KFLAG$  EQU     ØØ74H
ØØ74          Ø235Ø KFLAG$  EQU     ØØ74H
Ø8FC          Ø236Ø KIDATA$ EQU     Ø8FCH
Ø2Ø8          Ø237Ø KIDCB$  EQU     Ø2Ø8H
Ø2Ø2          Ø238Ø LBANK$  EQU     Ø2Ø2H
ØØ23          Ø239Ø LDRV$   EQU     ØØ23H
ØØ75          Ø24ØØ LFLAG$  EQU     ØØ75H
1566          Ø241Ø LNKFCB@ EQU     1566H
ØØ1E          Ø242Ø LOW$    EQU     ØØ1EH
ØØØD          Ø243Ø LSVC$   EQU     ØØØDH
24ØØ          Ø244Ø MAXCOR$ EQU     24ØØH
Ø4Ø1          Ø245Ø MAXDAY$ EQU     Ø4Ø1H
3ØØØ          Ø246Ø MINCOR$ EQU     3ØØØH
ØØ76          Ø247Ø MODOUT$ EQU     ØØ76H
ØØ76          Ø248Ø MODOUT$ EQU     ØØ76H
Ø4DC          Ø249Ø MONTBL$ EQU     Ø4DCH
ØØ77          Ø25ØØ NFLAG$  EQU     ØØ77H
ØØ78          Ø251Ø OPREG$  EQU     ØØ78H
ØØ78          Ø252Ø OPREG$  EQU     ØØ78H
Ø86E          Ø253Ø OPREG_SV_AREA   EQU     Ø86EH
```

```
Ø835          Ø254Ø OPREG_SV_PTR    EQU     Ø835H
14DC          Ø255Ø ORARET@ EQU     14DCH
ØØ3B          Ø256Ø OSRLS$  EQU     ØØ3BH
ØØ85          Ø257Ø OSVER$  EQU     ØØ85H
ØØ69          Ø258Ø OVRLY$  EQU     ØØ69H
Ø41Ø          Ø259Ø PAKNAM$ EQU     Ø41ØH
Ø382          Ø26ØØ PAUSE@  EQU     Ø382H
Ø7AF          Ø261Ø PCSAVE$ EQU     Ø7AFH
ØØ1B          Ø262Ø PDRV$   EQU     ØØ1BH
ØØ1B          Ø263Ø PDRV$   EQU     ØØ1BH
ØØ1C          Ø264Ø PHIGH$  EQU     ØØ1CH
Ø218          Ø265Ø PRDCB$  EQU     Ø218H
ØDCD          Ø266Ø PUTA@DE EQU     ØDCDH
ØDCA          Ø267Ø PUT_@   EQU     ØDCAH
ØDC6          Ø268Ø PUT_@_ROWCOL    EQU     ØDC6H
ØØ7B          Ø269Ø RFLAG$  EQU     ØØ7BH
ØØ7B          Ø27ØØ RFLAG$  EQU     ØØ7BH
ØDDØ          Ø271Ø ROWCOL_2_ADDR   EQU     ØDDØH
1BFF          Ø272Ø RST38@  EQU     1BFFH
Ø4C4          Ø273Ø RSTOR$  EQU     Ø4C4H
13A2          Ø274Ø RWRIT@  EQU     13A2H
Ø238          Ø275Ø S1DCB$  EQU     Ø238H
1DØØ          Ø276Ø SBUFF$  EQU     1DØØH
1A79          Ø277Ø SET@EXEC        EQU     1A79H
ØCF3          Ø278Ø SET_SCROLL      EQU     ØCF3H
ØØ8C          Ø279Ø SFCB$   EQU     ØØ8CH
ØØ7C          Ø28ØØ SFLAG$  EQU     ØØ7CH
ØØ7C          Ø281Ø SFLAG$  EQU     ØØ7CH
Ø22Ø          Ø282Ø SIDCB$  EQU     Ø22ØH
Ø228          Ø283Ø SODCB$  EQU     Ø228H
2142          Ø284Ø SPACE4$ EQU     2142H
Ø38Ø          Ø285Ø STACK$  EQU     Ø38ØH
ØØØØ          Ø286Ø START$  EQU     ØØØØH
ØØØØ          Ø287Ø START$  EQU     ØØØØH
ØØØB          Ø288Ø SVCRET$ EQU     ØØØBH
Ø1ØØ          Ø289Ø SVCTAB$ EQU     Ø1ØØH
1B13          Ø29ØØ SYSERR$ EQU     1B13H
ØØ4E          Ø291Ø TCB$    EQU     ØØ4EH
ØØ7D          Ø292Ø TFLAG$  EQU     ØØ7DH
ØØ2D          Ø293Ø TIME$   EQU     ØØ2DH
ØØ2D          Ø294Ø TIME$   EQU     ØØ2DH
ØØ2C          Ø295Ø TIMER$  EQU     ØØ2CH
ØØ2C          Ø296Ø TIMER$  EQU     ØØ2CH
ØØ2B          Ø297Ø TIMSL$  EQU     ØØ2BH
ØØ2B          Ø298Ø TIMSL$  EQU     ØØ2BH
Ø713          Ø299Ø TIMTSK$ EQU     Ø713H
Ø4C3          Ø3ØØØ TMPMT$  EQU     Ø4C3H
Ø7B1          Ø3Ø1Ø TRACE_INT       EQU     Ø7B1H
ØA8F          Ø3Ø2Ø TYPHK$  EQU     ØA8FH
ØB26          Ø3Ø3Ø TYPTSK$ EQU     ØB26H
ØØ13          Ø3Ø4Ø USTOR$  EQU     ØØ13H
ØØ7F          Ø3Ø5Ø VFLAG$  EQU     ØØ7FH
ØØ7F          Ø3Ø6Ø VFLAG$  EQU     ØØ7FH
ØØ8Ø          Ø3Ø7Ø WRINT$  EQU     ØØ8ØH
Ø4Ø1          Ø3Ø8Ø ZERO$   EQU     Ø4Ø1H
13AØ          Ø3Ø9Ø ZEROA@  EQU     13AØH
No end statement
ØØØØØ Total errors
```

```
                00100 ;SVCMAC/ASM - LS-DOS Version VI
0000            00110         TITLE   <SVCMAC - MACRO EQUIVALENTS>
                00120 ;*LIST  OFF
                00130 ;
0000            00140 @MOD2   EQU     0
FFFF            00150 @MOD4   EQU     -1
0000            00160 @@KEY   MACRO
0000            00170         LD      A,1
0000            00180         RST     40
0000            00190         ENDM
0000            00200 @@DSP   MACRO
0000            00210         LD      A,2
0000            00220         RST     40
0000            00230         ENDM
0000            00240 @@GET   MACRO
0000            00250         LD      A,3
0000            00260         RST     40
0000            00270         ENDM
0000            00280 @@PUT   MACRO
0000            00290         LD      A,4
0000            00300         RST     40
0000            00310         ENDM
0000            00320 @@CTL   MACRO
0000            00330         LD      A,5
0000            00340         RST     40
0000            00350         ENDM
0000            00360 @@PRT   MACRO
0000            00370         LD      A,6
0000            00380         RST     40
0000            00390         ENDM
0000            00400 @@WHERE MACRO
0000            00410         LD      A,7
0000            00420         RST     40
0000            00430         ENDM
0000            00440 @@KBD   MACRO
0000            00450         LD      A,8
0000            00460         RST     40
0000            00470         ENDM
0000            00480 @@KEYIN MACRO
0000            00490         LD      A,9
0000            00500         RST     40
0000            00510         ENDM
0000            00520 @@DSPLY MACRO   #MSG
0000            00530         IFEQ    %%,1
0000            00540         LD      HL,#MSG
0000            00550         ENDIF
0000            00560         LD      A,10
0000            00570         RST     40
0000            00580         ENDM
0000            00590 @@LOGER MACRO
0000            00600         LD      A,11
0000            00610         RST     40
0000            00620         ENDM
0000            00630 @@LOGOT MACRO   #MSG
0000            00640         IFEQ    %%,1
0000            00650         LD      HL,#MSG
0000            00660         ENDIF
0000            00670         LD      A,12
0000            00680         RST     40
0000            00690         ENDM
0000            00700 @@MSG   MACRO
```

```
0000  00710          LD      A,13
0000  00720          RST     40
0000  00730          ENDM
0000  00740 @@PRINT  MACRO   #MSG
0000  00750          IFEQ    %%,1
0000  00760          LD      HL,#MSG
0000  00770          ENDIF
0000  00780          LD      A,14
0000  00790          RST     40
0000  00800          ENDM
0000  00810 @@VDCTL  MACRO
0000  00820          LD      A,15
0000  00830          RST     40
0000  00840          ENDM
0000  00850 @@PAUSE  MACRO
0000  00860          LD      A,16
0000  00870          RST     40
0000  00880          ENDM
0000  00890 @@PARAM  MACRO
0000  00900          LD      A,17
0000  00910          RST     40
0000  00920          ENDM
0000  00930 @@DATE   MACRO
0000  00940          LD      A,18
0000  00950          RST     40
0000  00960          ENDM
0000  00970 @@TIME   MACRO
0000  00980          LD      A,19
0000  00990          RST     40
0000  01000          ENDM
0000  01010 @@CHNIO  MACRO
0000  01020          LD      A,20
0000  01030          RST     40
0000  01040          ENDM
0000  01050 @@ABORT  MACRO
0000  01060          LD      A,21
0000  01070          RST     40
0000  01080          ENDM
0000  01090 @@EXIT   MACRO
0000  01100          LD      A,22
0000  01110          RST     40
0000  01120          ENDM
0000  01130 @@CMNDI  MACRO
0000  01140          LD      A,24
0000  01150          RST     40
0000  01160          ENDM
0000  01170 @@CMNDR  MACRO
0000  01180          LD      A,25
0000  01190          RST     40
0000  01200          ENDM
0000  01210 @@ERROR  MACRO
0000  01220          LD      A,26
0000  01230          RST     40
0000  01240          ENDM
0000  01250 @@DEBUG  MACRO
0000  01260          LD      A,27
0000  01270          RST     40
0000  01280          ENDM
0000  01290 @@CKTSK  MACRO
0000  01300          LD      A,28
0000  01310          RST     40
```

```
0000  01320          ENDM
0000  01330 @@ADTSK  MACRO
0000  01340          LD      A,29
0000  01350          RST     40
0000  01360          ENDM
0000  01370 @@RMTSK  MACRO
0000  01380          LD      A,30
0000  01390          RST     40
0000  01400          ENDM
0000  01410 @@RPTSK  MACRO
0000  01420          LD      A,31
0000  01430          RST     40
0000  01440          ENDM
0000  01450 @@KLTSK  MACRO
0000  01460          LD      A,32
0000  01470          RST     40
0000  01480          ENDM
0000  01490 @@CKDRV  MACRO
0000  01500          LD      A,33
0000  01510          RST     40
0000  01520          ENDM
0000  01530 @@DODIR  MACRO
0000  01540          LD      A,34
0000  01550          RST     40
0000  01560          ENDM
0000  01570 @@RAMDIR          MACRO
0000  01580          LD      A,35
0000  01590          RST     40
0000  01600          ENDM
0000  01610 @@DCSTAT          MACRO
0000  01620          LD      A,40
0000  01630          RST     40
0000  01640          ENDM
0000  01650 @@SLCT   MACRO
0000  01660          LD      A,41
0000  01670          RST     40
0000  01680          ENDM
0000  01690 @@RSTOR  MACRO
0000  01700          LD      A,44
0000  01710          RST     40
0000  01720          ENDM
0000  01730 @@STEPI  MACRO
0000  01740          LD      A,45
0000  01750          RST     40
0000  01760          ENDM
0000  01770 @@SEEK   MACRO
0000  01780          LD      A,46
0000  01790          RST     40
0000  01800          ENDM
0000  01810 @@RSLCT  MACRO
0000  01820          LD      A,47
0000  01830          RST     40
0000  01840          ENDM
0000  01850 @@RDSEC  MACRO
0000  01860          LD      A,49
0000  01870          RST     40
0000  01880          ENDM
0000  01890 @@VRSEC  MACRO
0000  01900          LD      A,50
0000  01910          RST     40
0000  01920          ENDM
```

```
0000  01930 @@HDFMT MACRO
0000  01940         LD      A,52
0000  01950         RST     40
0000  01960         ENDM
0000  01970 @@WRSEC MACRO
0000  01980         LD      A,53
0000  01990         RST     40
0000  02000         ENDM
0000  02010 @@WRSSC MACRO
0000  02020         LD      A,54
0000  02030         RST     40
0000  02040         ENDM
0000  02050 @@WRTRK MACRO
0000  02060         LD      A,55
0000  02070         RST     40
0000  02080         ENDM
0000  02090 @@RENAM MACRO
0000  02100         LD      A,56
0000  02110         RST     40
0000  02120         ENDM
0000  02130 @@REMOV MACRO
0000  02140         LD      A,57
0000  02150         RST     40
0000  02160         ENDM
0000  02170 @@INIT  MACRO
0000  02180         LD      A,58
0000  02190         RST     40
0000  02200         ENDM
0000  02210 @@OPEN  MACRO
0000  02220         LD      A,59
0000  02230         RST     40
0000  02240         ENDM
0000  02250 @@CLOSE MACRO
0000  02260         LD      A,60
0000  02270         RST     40
0000  02280         ENDM
0000  02290 @@BKSP  MACRO
0000  02300         LD      A,61
0000  02310         RST     40
0000  02320         ENDM
0000  02330 @@CKEOF MACRO
0000  02340         LD      A,62
0000  02350         RST     40
0000  02360         ENDM
0000  02370 @@LOC   MACRO
0000  02380         LD      A,63
0000  02390         RST     40
0000  02400         ENDM
0000  02410 @@LOF   MACRO
0000  02420         LD      A,64
0000  02430         RST     40
0000  02440         ENDM
0000  02450 @@PEOF  MACRO
0000  02460         LD      A,65
0000  02470         RST     40
0000  02480         ENDM
0000  02490 @@POSN  MACRO
0000  02500         LD      A,66
0000  02510         RST     40
0000  02520         ENDM
0000  02530 @@READ  MACRO
```

```
0000  02540          LD      A,67
0000  02550          RST     40
0000  02560          ENDM
0000  02570 @@REW     MACRO
0000  02580          LD      A,68
0000  02590          RST     40
0000  02600          ENDM
0000  02610 @@RREAD   MACRO
0000  02620          LD      A,69
0000  02630          RST     40
0000  02640          ENDM
0000  02650 @@RWRIT   MACRO
0000  02660          LD      A,70
0000  02670          RST     40
0000  02680          ENDM
0000  02690 @@SEEKSC           MACRO
0000  02700          LD      A,71
0000  02710          RST     40
0000  02720          ENDM
0000  02730 @@SKIP    MACRO
0000  02740          LD      A,72
0000  02750          RST     40
0000  02760          ENDM
0000  02770 @@VER     MACRO
0000  02780          LD      A,73
0000  02790          RST     40
0000  02800          ENDM
0000  02810 @@WEOF    MACRO
0000  02820          LD      A,74
0000  02830          RST     40
0000  02840          ENDM
0000  02850 @@WRITE   MACRO
0000  02860          LD      A,75
0000  02870          RST     40
0000  02880          ENDM
0000  02890 @@LOAD    MACRO
0000  02900          LD      A,76
0000  02910          RST     40
0000  02920          ENDM
0000  02930 @@RUN     MACRO
0000  02940          LD      A,77
0000  02950          RST     40
0000  02960          ENDM
0000  02970 @@FSPEC   MACRO
0000  02980          LD      A,78
0000  02990          RST     40
0000  03000          ENDM
0000  03010 @@FEXT    MACRO
0000  03020          LD      A,79
0000  03030          RST     40
0000  03040          ENDM
0000  03050 @@FNAME   MACRO
0000  03060          LD      A,80
0000  03070          RST     40
0000  03080          ENDM
0000  03090 @@GTDCT   MACRO
0000  03100          LD      A,81
0000  03110          RST     40
0000  03120          ENDM
0000  03130 @@GTDCB   MACRO
0000  03140          LD      A,82
```

```
0000  03150          RST     40
0000  03160          ENDM
0000  03170 @@GTMOD  MACRO
0000  03180          LD      A,83
0000  03190          RST     40
0000  03200          ENDM
0000  03210 @@RDSSC  MACRO
0000  03220          LD      A,85
0000  03230          RST     40
0000  03240          ENDM
0000  03250 @@GATRD  MACRO
0000  03260          LD      A,86
0000  03270          RST     40
0000  03280          ENDM
0000  03290 @@DIRRD  MACRO
0000  03300          LD      A,87
0000  03310          RST     40
0000  03320          ENDM
0000  03330 @@DIRWR  MACRO
0000  03340          LD      A,88
0000  03350          RST     40
0000  03360          ENDM
0000  03370 @@GATWR  MACRO
0000  03380          LD      A,89
0000  03390          RST     40
0000  03400          ENDM
0000  03410 @@MUL8   MACRO
0000  03420          LD      A,90
0000  03430          RST     40
0000  03440          ENDM
0000  03450 @@MUL16  MACRO
0000  03460          LD      A,91
0000  03470          RST     40
0000  03480          ENDM
0000  03490 @@DIV8   MACRO
0000  03500          LD      A,93
0000  03510          RST     40
0000  03520          ENDM
0000  03530 @@DIV16  MACRO
0000  03540          LD      A,94
0000  03550          RST     40
0000  03560          ENDM
0000  03570 @@DECHEX         MACRO
0000  03580          LD      A,96
0000  03590          RST     40
0000  03600          ENDM
0000  03610 @@HEXDEC         MACRO
0000  03620          LD      A,97
0000  03630          RST     40
0000  03640          ENDM
0000  03650 @@HEX8   MACRO
0000  03660          LD      A,98
0000  03670          RST     40
0000  03680          ENDM
0000  03690 @@HEX16  MACRO
0000  03700          LD      A,99
0000  03710          RST     40
0000  03720          ENDM
0000  03730 @@HIGH$  MACRO
0000  03740          LD      A,100
0000  03750          RST     40
```

```
0000          03760          ENDM
0000          03770 @@FLAGS  MACRO
0000          03780          LD      A,101
0000          03790          RST     40
0000          03800          ENDM
0000          03810 @@BANK   MACRO
0000          03820          LD      A,102
0000          03830          RST     40
0000          03840          ENDM
0000          03850 @@BREAK  MACRO   #ADR
0000          03860          IFEQ    %%,1
0000          03870          LD      HL,#ADR
0000          03880          ENDIF
0000          03890          LD      A,103
0000          03900          RST     40
0000          03910          ENDM
0000          03920 @@CLS    MACRO
0000          03930          LD      A,105
0000          03940          RST     40
0000          03950          ENDM
0000          03960 @@CKBRKC         MACRO
0000          03970          LD      A,106
0000          03980          RST     40
0000          03990          ENDM
              04000 *LIST    ON
0000          04010          END
00000 Total errors
```