

? EXECUTE ESPOL/DISK

PACKET 35  
INPUT 1882 CARDS FROM ZIP  
TIME 133A  
DATE 79219 TUESDAY, 08/07/79

\*\*\* BURROUGHS B5700 TSSMCP MARK XVI.0.172 AND INTRINSICS MARK XVI.0.132 \*\*\*

#NO MESSAGES TODAY

13:38:13 ? EXECUTE ESPOL/DISK  
13:38:14 ? CORE= 12000  
13:38:14 ? STACK= 1000  
13:38:14 ? FILE TAPF= SYMBOL/TSSMCP DISK SERIAL  
13:38:15 ? FILE DISK= TSS/MCPA  
13:38:15 ? FILE STUFF= TSSMCP/STUFF  
13:38:15 ? DATA CARD  
13:38:17 4:ESPOL/DISK/SITE= 4 BOJ 1338 05/12/77  
13:38:18 DKA OUT SFR CODE SITE:ESPOL/DISK= 4  
13:38:19 CDC IN CARD:ESPOL/DISK= 4  
13:38:20 DKA IN SER SYMBOL TSSMCP:ESPOL/DISK= 4  
13:38:25 DKA OUT SER TSSMCP STUFF:ESPOL/DISK= 4  
13:38:31 DKA OUT SFR CODISK SITE:ESPOL/DISK= 4  
13:38:37 DKA 1 RETRIES, MIX= 4, DA=0103170, SEGS=05, R=0140000542047165, IO=1  
13:38:37 DKA 3 RETRIES, MIX= 4, DA=0103180, SEGS=05, R=0140000562047165, IO=1  
13:38:50 DKA 1 RETRIES, MIX= 4, DA=0103370, SFGS=05, R=0140000542047165, IO=1  
13:38:51 DKA 2 RETRIES, MIX= 4, DA=0103390, SFGS=05, R=0140000562047165, IO=1  
13:40:01 DKA 1 RETRIES, MIX= 4, DA=0103795, SFGS=05, R=0140000542047422, IO=1  
13:40:36 DKA 1 RETRIES, MIX= 4, DA=0103955, SFGS=05, R=0140001142047422, IO=2  
13:40:38 DKA 2 RETRIES, MIX= 4, DA=0103965, SFGS=05, R=0140000562047422, IO=1  
13:40:48 4DS  
13:40:49 -OPRTR DS=FD:ESPOL/DISK= 4, S= 3, A= 34110  
13:40:49 TSSMCP/STUFF/SITE= 50 SEGS=-CREATED 08/07/79 AT 13:38:24:07  
13:40:50 DKA REL TSSMCP STUFF:ESPOL/DISK= 4  
13:40:50 CODISK/SITE/SITE= 600 SEGS=-CREATED 08/07/79 AT 13:38:30:00  
13:40:50 DKA REL CODISK SITE:ESPOL/DISK= 4  
13:40:50 CODE/SITE/SITE= 0 SEGS=-CREATED 08/07/79 AT 13:38:17:41  
13:40:50 DKA REL CODE SITE:ESPOL/DISK= 4  
13:41:14 CDC REL CARD:ESPOL/DISK= 4  
13:41:14 ? END.  
13:41:15 DKA REL SYMBOL TSSMCP:ESPOL/DISK= 4  
13:41:17 ESPOL/DISK/SITE= 4, PST= 1130 DS=ED  
13:41:18 FOR ESPOL/DISK= 4: PST= 89, TOT= 82, CORE=15360  
13:41:19 PKT#0035 REMOVED

?USER = SITE

PACKET 33  
INPUT 26 CARDS FROM CRA  
TIME 1332  
DATE 79219 TUESDAY, 08/07/79

\*\*\* BURROUGHS B5700 TSMCP MARK XVI.0.172 AND INTRINSICS MARK XVI.0.132 \*\*\*

#NO MESSAGES TODAY

13132150 ?USER= SITE  
13132150 ? EXECUTE PATCH/MERGE  
13132150 ?DATA CARD  
13132153 4:PATCH/MERGE/SITE= 3 ROJ 1332 05/12/77  
13132154 CDB IN CARD:PATCH/MERGE= 3  
13132155 PRD0034 OUT 011 LINE:PATCH/MERGE= 3  
13133100 DKA OUT SER MASTERF SITE:PATCH/MERGE= 3  
13133101 DKA IN SER PATCHES TSSMCP:PATCH/MERGE= 3  
13133101 DKA OUT SER PATCHDE SITE:PATCH/MERGE= 3  
13133102 DKA OUT SER SORTLIN SITE:PATCH/MERGE= 3  
13135123 ?END  
13135123 CDB REL CARD:PATCH/MERGE= 3  
13138112 PATCHDF/SITE/SITE= 0 SEGS==CREATED 08/07/79 AT 00:22:00:19  
13138113 DKA REL PATCHDE SITE:PATCH/MERGE= 3  
13141140 PRD0034 OUT 012:PATCH/MERGE= 3  
13141148 DKA REL PATCHES TSSMCP:PATCH/MERGE= 3  
13141148 SORTLIN/SITE/SITE= 1350 SEGS==CREATED 08/07/79 AT 13:33:01:00  
13141149 DKA REL SORTLIN SITE:PATCH/MERGE= 3  
13141150 PRD0034 REL 012 LINE 5483:PATCH/MERGE= 3  
13141150 MASTERF/SITE/SITE= 900 SEGS==CREATED 08/07/79 AT 13:32:59:19  
13141150 DKA REL MASTERF SITE:PATCH/MERGE= 3  
13141151 PATCH/MERGE/SITE= 3,PST= 3:06 EOJ  
13141151 FOR PATCH/MERGE= 3:PST= 185,IOT= 95,CORE= 9216  
13141151 PKT#0033 REMOVED

\*\*\*\*\* INPUT \*\*\*\*\*

\$@CARD MERGE ZIP LIST C 0000000/CARD
\$. 139 PATCHES FOR TSSMCP C

CARD INPUT IS CARD
PATCHES/TSSMCP WILL BE MERGED
PATCH /TSSMCP IS NOT ON DISK

\$\*EXECUTE ESPOL/DISK C
\$\*CORP = 12000 C
\$\*STACK = 1000 C
\$\*FILE TAPE = SYMBOL/TSSMCP DISK SERIAL C
\$\*FILE DISK = TSS/MCPA C
\$\*FILE STUFF = TSSMCP/STUFF C
\$\*DATA CARD C
\$= \*\*\*\*\* THIS LISTING SHOWS PATCHES IMPLEMENTED AT UCSC \*\*\*\*\* C
\$RESET LIST SET TAPE STUFF SINGLE C
\$ SET AUTODUMP PACKETS DUMP C
\$ RESET SEPTICTANK CHECKLINK SAVERRESULTS STATISTICS C
\$RESET DEBUGGING DFX DKRNOFX AUXMEM NEWLOGGING SHAREDISK C
\$ SET TWXONLY C

\$# PATCH NUMBER 101 FOR TSSMCP CONTAINS 001 CARDS. D 101 PATCHES/TSSMCP
BEGIN DS: 16 LIT"=SYSTEM HANG, F=") 00652400 D 101
\$: BY DDD D 101
\$: 7/16/74 D 101
\$: TR .... D 101
\$: THIS PATCH REFORMATS THE "SYSTEM HANG" MESSAGE. D 101
\$!\*\*\*\*\* D 101

\$#PATCH NUMBER 102 FOR TSSMCP CONTAINS 010 CARDS 00000000 D 102
SVOIDT 14218081 14218071 D 102
IF AIT[AIT[0]], [8:10] NEQ 1 THEN 14218072 D 102
BEGIN 14218074 D 102
I=1 14218076 D 102
WHILE AIT[I], [8:10] = 1 DO I=I+1; 14218078 D 102

```

FOR J:=AIT[0] STEP =1 UNTIL I DO
    AIT[J+1]:=AIT[J]
END ELSE I:=AIT[0]+1;
    AIT[0]:=*P(DUP) + 1;
    AIT[J]:=-(1&J[8:38:10]&M[C][F[F]]);
$! THIS CHANGE CORRECTS AN ERROR IN THE HANDLING OF COBOL68
$! OPEN STATEMENTS THAT OCCUR IN THE INPUT OR OUTPUT PROCEDURE
$! OF A SORT. THE FILES WILL NOW BE LEFT OPEN AS THEY SHOULD.
$! *****

```

```

14218080 D 102
14218082 D 102
14218084 D 102
14218086 D 102
14218088 D 102
90000000 D 102
90000005 D 102
90000010 D 102

```

```

$#PATCH NUMBER 103 FOR TSSMCP CONTAINS 1 CARDS.

```

```

IF P OR I THEN
$! THIS PATCH CORRECTS AN ERROR IN THE MCP PROCEDURE BACKCLOSE
$! THAT WOULD SOMETIMES CAUSE "PUD" FILES NOT TO BE PUNCHED EVEN WITH
$! THE OPTION AUTOPRNT SET.
$! *****

```

```

38608920 D 103
D 103
D 103
D 103
D 103

```

```

$#PATCH NUMBER 104 FOR TSSMCP CONTAINS 1 CARD

```

```

SCHEDLOOK(KTR,TYPE=7) % ES=6, XS=7
$! FIXES THE SCHEDULE LINE "XS" KEYIN
$! *****

```

```

16651000 D 104
D 104
D 104

```

```

$#PATCH NUMBER 105 FOR TSSMCP CONTAINS 2 CARDS

```

```

IF (T1 + PSFUDOMIX(P1MIX)) # 0 THEN
IF PACKETPAGE(T1 = 32) # 0 THEN
$! BY KEN KOVACS
$! DATA 4/22/74
$! THIS PATCH CORRECTS AN INVALID INDEX IN CONTROL STATE.
$! *****

```

```

39092165 D 105
39092167 D 105
D 105
D 105
D 105
D 105

```

```

$#PATCH NUMBER 106 FOR TSSMCP CONTAINS 8 CARDS.

```

```

$ VOID 07118401
IF PACKETCARD = 5 THEN
IF PTYPE = 3 AND NOT CONTINUE AND NOT ADECK THEN
IF PKTONLY THEN
$ SET OMIT = NOT(PACKETS)
IF PSFUD[UNITNO=32] = 0 THEN
PRINTTHECOVER(CARDLOC,UNITNO,PSOURCE);
$ POP OMIT
$! BY KFK
$! DATE 02/20/75

```

```

07118100 D 106
07118500 D 106
07139520 D 106
20602515 D 106
20609425 D 106
20609430 D 106
20609435 D 106
20609440 D 106
D 106
D 106
D 106

```

\$: MSA CENTRAL = DETROIT D 106  
 \$: THIS PATCH WILL ALLOW PACKETS AND DECKS TO BE MIXED. ALSO IF THE D 106  
 \$: SPD OPTION "PKTONLY" IS SET THEN ALL DECKS WILL BE USED AS IF THEY D 106  
 \$: WERE PACKETS. THAT IS THE USE OF PKTONLY IS NOW CHANGED SO THAT D 106  
 \$: IT NO LONGER CONTROLS THE LOADING OF DECKS BUT TO CONTROL OF THEIR D 106  
 \$: USE. D 106  
 \$: \*\*\*\*\* D 106

\$#PATCH NUMBER 107 FOR TSSMCP CONTAINS 1 CARDS. D 107  
  
 HEADR[4]=08(@1001)P2:38:10J&SYSNO;4:46:2J; 20289115 D 107  
 \$: BY KK D 107  
 \$: DATE 8/7/74 D 107  
 \$: THIS PATCH WILL CORRECT A CONDITION THAT COULD MARK THE PACKET D 107  
 \$: PAGE AS BEING IN USE BY THE WRONG SYSTEM AFTER A HALT/LOAD. D 107  
 \$: \*\*\*\*\* D 107

\$# PATCH NUMBER 108 FOR TSSMCP CONTAINS 001 CARD. D 108  
  
 IF P2>0 THEN MOVF(15,P2,T+4) ELSE M[T+4]+P2; 03120000 D 108  
 \$: BY DDD D 108  
 \$: DATE 8/29/74 D 108  
 \$: THIS PATCH ALLOWS ERROR MESSAGES LONGER THAN 60 CHARACTERS D 108  
 \$: TO BE PRINTED CORRECTLY ON A REMOTE TERMINAL THROUGH CANDE. D 108  
 \$: \*\*\*\*\* D 108

\$#PATCH NUMBER 109 FOR TSSMCP CONTAINS 1 CARD. D 109  
  
 IF LOGLINE = 0 THEN LOGLINE = UNITNO; 20602502 D 109  
 \$: BY DDD D 109  
 \$: DATE 9/19/74 D 109  
 \$: THIS PATCH REPLACES A LINE OF CODE THAT WAS INADVERTENTLY D 109  
 \$: LEFT OUT IN THE XV.3 RELEASE. WITHOUT THIS PATCH, ESPDISK, D 109  
 \$: IS NOT HANDLED CORRECTLY, INCORRECT LOG ENTRIES ARE MADE, D 109  
 \$: AND SYSTEM HALTS CAN OCCUR. D 109  
 \$: \*\*\*\*\* D 109

\$#PATCH NUMBER 110 FOR TSSMCP CONTAINS 1 CARDS. D 110  
  
 FIB[16].[CF1]=0;% 41204000 D 110  
 \$: BY KK D 110  
 \$: DATE 10/22/74 D 110  
 \$: THIS PATCH CORRECTS THE TSSMCP'S HANDLING OF TAPE FILES D 110  
 \$: CLOSED WITH THE "CLOSE(<FILEID>,\*)" SYNTAX. THE TSSMCP D 110  
 \$: WOULD "FORGET" THAT THE LAST I/O DONE ON THIS FILE WAS A D 110

\$: REVERSE I/O BY ZEROING FIB[16], THUS WHEN THE FILE IS D 110  
\$: OPFED AGAIN THE TSSMCP WOULD SPACE(FORWARD) OVER D 110  
\$: THE NEXT FILE, AND THE FILE THAT WAS JUST CLOSED WOULD BE D 110  
\$: ACCESSED AGAIN. D 110  
\$:\*\*\*\*\* D 110

\$# PATCH NUMBER 111 FOR TSSMCP CONTAINS 1 CARDS. D 111

\$ VOID 06098201 06097600 D 111  
\$: BY KK D 111  
\$: DATE 11/ 3/74 D 111  
\$: THIS PATCH WILL ALLOW A DISK SQUASH TO BE STOPPED ONLY AFTER IT D 111  
\$: HAS FINISHED MOVING A FILE. IF STOPPED SOONER THEN THE AVAILABLE D 111  
\$: DISK TABLE COULD BE DAMMAGED. D 111  
\$:\*\*\*\*\* D 111

\$# PATCH NUMBER 112 FOR TSSMCP CONTAINS 1 CARDS. D 112

RRRMECH + NOT TWO(U) AND RRRMECH; & LET STATUS FIND IT 04383900 D 112  
\$: BY KK D 112  
\$: DATE 11/ 4/74 D 112  
\$: THIS PATCH CORRECTS A PROBLEM WITH UNITS GOING NOT READY D 112  
\$: AND THEN NOT BEING PICKED UP AS BEING READY WHEN THEY GO READY D 112  
\$:\*\*\*\*\* D 112

\$#PATCH NUMBER 113 FOR TSSMCP CONTAINS 5 CARDS D 113

X: CLK[0] != 03067500 D 113  
\$ SFT OMIT = NOT(NEWLOGGING) 03067600 D 113  
IF LOGSTOPPED[P1MIX] THEN PROCTIME[P1MIX] ELSE 03067700 D 113  
\$ POP OMIT 03067800 D 113  
PROCTIME[P1MIX] + CLOCK + P(RTR) 03067900 D 113  
\$: BY KFK D 113  
\$: DATE 11/27/74 D 113  
\$: THIS PATCH CORRECTS A CONDITION THAT CAUSED THE LOG OFF MESSAGE TO D 113  
\$: CONTAIN ERRONEOUS DATE. AS A RESULT OF THE NEW MCP COMPILE TIME D 113  
\$: OPTION "NEWLOGGING" THE VALUE OF PROCTIME[P1MIX] IS CHANGED TO D 113  
\$: INDICATE IF THE PROGRAM SHOULD BE CHARGED FOR CPU TIME OR NOT. D 113  
\$: THIS WAS NOT TAKES INTO CONSIDERATION WHEN COMM2 PASSED THE VALUE D 113  
\$: TO BE USED AS "CLOCK" BY CANDE. D 113  
\$:\*\*\*\*\* D 113

\$#PATCH NUMBER 114 FOR TSSMCP CONTAINS 2 CARDS D 114

\$ VOID 37284150 D 114

```

      IF U < 16 THEN FPB[FN+3].[6:17]+PRNTABLE[U].[31:17] 37285410 D 114
$ BY KFK D 114
$ DATE 11/27/74 D 114
$ WHEN USING A MULTI-FILE PBT THE SECOND AND FOLLOWING FILFS WILL D 114
$ SOMETIMES BE PRINTED SEVERAL TIMES. THIS WAS CAUSED BY A CONFLICT D 114
$ IN THE USE OF A FIELD IN THE FPB OF A PROGRAM. THE SAME FIELD D 114
$ THAT IS USED TO HOLD THE NUMBER OF COPIES (FROM A FILE EQUATE) D 114
$ IS ALSO USED TO HOLD THE PRN OF THE TAPE. THIS PROBLEM IS D 114
$ FIXED BY NOT MOVING THE PRN TO THE FPB UNTIL THE FILE IS CLOSED. D 114
$ THEN WHEN THE FILE IS REOPENED A NEW SECTION OF THE FPB WILL BE D 114
$ USED AND THIS WILL CONTAIN THE CORRECT VALUE FOR COPIES. D 114
$ ***** D 114

```

```

$#PATCH NUMBER 115 FOR TSSMCP CONTAINS 1 CARDS D 115
      KIND+FIB[4].[8:4] IF FIB[13].[28:10]#0 THEN REEL+FIB[13].[28:10] 39092000 D 115
$ BY KFK D 115
$ DATE 11/27/74 D 115
$ IF THE REEL IS GIVEN ON A FILE(EQUATION) CONTROL CARD IT WILL D 115
$ NOT BE USED BY THE MCP WHEN IT LOOKS FOR A TAPE FILE AT FILE D 115
$ OPEN TIME. THIS IS BECAUSE THE REEL NUMBER IS MOVED FROM THE D 115
$ FIB. NOW THE REEL NUMBER IS ONLY MOVED FROM THE FIB IF IT IS D 115
$ NONZERO D 115
$ ***** D 115

```

```

$#PATCH NUMBER 116 FOR TSSMCP CONTAINS 1 CARDS D 116
$ VOIDT 38024081 38024010 D 116
$ BY KFK D 116
$ DATE 12/23/74 D 116
$ IF A PROGRAM READS A DISK FILE THAT HAS A ROW SIZE THAT IS LESS D 116
$ THAN THE PROGRAMS BUFFER SIZE, THE MCP WILL ONLY GET A CORE D 116
$ BUFFER THE SIZE OF THE DISK FILE'S ROW. BUT ANY PROGRAM THAT D 116
$ USES THE BUFFER WITHOUT GOING THUR THE INTRINSICS WILL NOT D 116
$ KNOW THAT THE BUFFER SIZE HAS BEEN CHANGED; THEREFORE, AN INVALID D 116
$ LINK MAY OCCURE IF ONE TRIFS TO USE THE PART OF THE BUFFER THAT D 116
$ IS NOT THERE. D 116
$ ***** D 116

```

```

$#PATCH NUMBER 117 FOR TSSMCP CONTAINS 2 CARDS D 117
      IF(NR[F]AND FOURMASK).[1:35]#0 THEN GO CLOSEIT; 06058630 D 117
      IF(NR[F+5]AND NINEMASK).[1:28]#0 THEN 06058640 D 117
$ BY DDD D 117
$ DATE 12/27/74 D 117
$ THIS PATCH CORRECTS A PROBLEM THAT WOULD CAUSE SOME FILFS TO D 117
$ BE MARKED IN USE AFTER A HALT/LOAD ON A SHARFDISK SYSTEM. D 117
$ ***** D 117

```

```

$#PATCH NUMBER 118 FOR TSSMCP CONTAINS 1 CARD                                D 118

    IF COUNT <0 THEN GO TO START;                                           37240050 D 118
$!BY DDD                                                                    D 118
$!2/5/75                                                                    D 118
$!SOFTWARE FLASH NUMBER 68                                                  D 118
$!IF TWO LIBMAIN/DISKS ARE BOTH WAITING FOR THE SAME TAPE TO BECOME        D 118
$!READY, ONE COPY WILL BE ASSIGNED THE CORRECT TAPE AND THE OTHER          D 118
$!COPY OF LIBMAIN WILL BE ASSIGNED TO MTA.                                  D 118
$!*****                                                                    D 118

```

```

$#PATCH NUMBER 119 FOR TSSMCP CONTAINS 34 CARDS:                            D 119

REAL BVD,UBLKSZ,SPB;                                                         27990410 D 119
    RCL1=[M[CT1=SPACE(41)]]&41[8:38:10];                                     27993900 D 119
    IF HDRTYPE NEQ 3                                                         27994200 D 119
        THEN FOR NT1:=15 STEP 1 UNTIL NT2 DO RCL[NT1-5]:=H[NT1]             27994210 D 119
        ELSE FOR NT1:=20 STEP 1 UNTIL NT2+5 DO                               27994220 D 119
            RCL[NT1-10]:=H[NT1];                                             27994230 D 119
    UBLKSZ:=NT2;                                                             27995310 D 119
    IF (NT4:=H[2].[5:3] = 2) THEN                                           % DECIMAL (4-BIT) 27995600 D 119
        BCL[0]:=((SPB:=NT2 DIV 30 + (NT2 MOD 30 NEQ 0))& % SEGMENTS/BLOCK 27996300 D 119
            NT1[[1:34:14]] % WORDS/RECORD 27996505 D 119
    IF HDRTYPE GEQ 2 THEN % FIGURE OUT EOF USING B6700 FORMULA             27996510 D 119
    BEGIN                                                                      27996515 D 119
        STREAM(B:=0 : W:=H[14]);                                             27996520 D 119
        BEGIN S1:=LOC W; D1:= LOC B; D1:=D1+4; SKIP 4 DB;                   27996525 D 119
            20(IF SB THEN DS:=SET ELSE DS:=RESFT) SKIP 1 SB);               27996530 D 119
        END;                                                                    27996535 D 119
        BVD:=P; % BITS OF VALID DATA IN LAST SEGMENT                       27996540 D 119
        IF H[2].[8:1]                                                         27996545 D 119
            THEN IF (NT2:=H[2].[5:3])=2                                       27996550 D 119
                THEN BEGIN NT1:=12; NT4:=3; END                               27996555 D 119
                ELSE IF NT2=4                                                 27996560 D 119
                    THEN BEGIN NT1:=6; NT4:=7; END                           27996565 D 119
                    ELSE BEGIN NT1:=8; NT4:=5; END                           27996570 D 119
                ELSE BEGIN NT1:=1; NT4:=47; END;                               27996575 D 119
        NT2:=((H[14].[20:28] MOD SPB)*NT1*30)                                  27996580 D 119
            +((BVD + NT4) DIV (48 DIV NT1));                                   27996585 D 119
        NT1:=IF NT2 LSS UBLKSZ THEN NT2 ELSE UBLKSZ;                          27996590 D 119
        NT2:=NT1+(H[14].[20:28] DIV SPB)*NT3*(NT4:=H[3].[32:16]);           27996595 D 119
        END;                                                                    27996600 D 119
        BCL[7]:=IF HDRTYPE GEQ 2 THEN (NT2+NT4-1) DIV NT4 - 1               27996605 D 119
            ELSE H[4];                                                         27996610 D 119
    SVOIDT                                                                    27996800 D 119
        H1=[M[SPACE(42)]]&30[8:38:10];                                       28070000 D 119
        H2=H&41[8:38:10];                                                    28274200 D 119
    $!TSSMCP, 4208                                                            D 119
    $!                                                                           D 119
    $! B6700 TAPES WITH NEW HEADERS                                         D 119
    $! THIS CHANGE ALLOWS THE B5700 LIBRARY MAINTENANCE TO LOAD FILES       D 119

```





```

BEGIN
SI:=LOC A; SKIP SB;
IF SB THEN
  BEGIN
  SKIP 39 SR;
  IF SB THEN DS:=8 LIT"OFAST "
  ELSE
    BEGIN
    SKIP SB;
    IF SB THEN DS:=8 LIT"OSLOW "
    ELSE
      BEGIN
      C/SI:=LOC B;
      DS:=6 LIT"OEU # "; DS:=2 DEC;
      JUMP OUT TO L;
      DS:=8 LIT"ODISK ";
      L:
      FND
      END
    END
  ELSE
    BEGIN
    SKIP 5 SB;
    DS:=LIT"O"; DS:=7 CHR;
    END;
  END;
SI BY KFK
SI DATE 3/18/75
SI MSA CENTRAL - DETROIT
SI THIS PATCH WILL PLACE THE OUTPUT UNIT NAME OF A LIBRARY COPY
SI IN WORD 27 OF THE SHEET SO THAT WHEN A TS IS DONE THIS NAME WILL
SI BE LISTED ON THE OUTPUT MESSAGE.
SI*****

```

```

20571710 D 123
20571720 D 123
20571730 D 123
20571740 D 123
20571750 D 123
20571760 D 123
20571770 D 123
20571780 D 123
20571782 D 123
20571790 D 123
20571800 D 123
20571810 D 123
20571820 D 123
20571830 D 123
20571840 D 123
20571842 D 123
20571850 D 123
20571860 D 123
20571870 D 123
20571880 D 123
20571890 D 123
20571900 D 123
20571910 D 123
20571920 D 123
20571930 D 123

```

```

S#PATCH NUMRER 124 FOR TSSMCP CONTAINS 2 CARDS.
IF KIND=7 THEN FPB[FNUM+3]+(*P(DUP))&T2[15:15:8]
&O[6:39:9];
SI BY KFK
SI 5/7/75
SI MSA CENTRAL - DETROIT
SI WHEN USING A MULTI-FILE PBT THE SECOND AND FOLLOWING FILES WILL
SI SOMETIMES BE PRINTED SEVERAL TIMES. THIS WAS CAUSED BY A CONFLICT
SI IN THE USE OF A FIELD IN THE FPB OF A PROGRAM. THE SAME FIELD
SI THAT IS USED TO HOLD THE NUMBER OF COPIES (FROM A FILE EQUATE)
SI IS ALSO USED TO HOLD THE PRN OF THE TAPE.
SI*****

```

```

38228510 D 124
38228520 D 124
D 124
D 124
D 124
D 124
D 124
D 124
D 124
D 124
D 124

```

```

S#PATCH NUMBER 126 FOR TSSMCP CONTAINS 1 CARD.
PROCVAl:=0;
SI BY KFK

```

```

20602860 D 126
D 126

```

```

$! DATE 04/05/75
$! MSA CENTRAL = DETROIT
$! THIS PATCH CORRECTS AN ERROR IN CONTROL CARD THAT COULD CAUSE
$! UNEXPLAINED CONTROL CARD ERRORS AND SYSTEM HALTS. THE ERROR
$! WAS CAUSED BY CONTROL CARD NOT RESETTING THE STACK VARIABLE
$! "PROCVAL" BACK TO ZERO AFTER A TYPED PROCEDURE WAS CALLED. THUS
$! THE NEXT TYPED PROCEDURE CALLED WOULD START WITH A NON ZERO VALUE.
$! *****

```

```

$#PATCH NUMBER 127 FOR TSSMCP CONTAINS 3 CARDS.

```

```

                IF (T:=SCAN) = ENDFI THEN          20608740 D 127
                GO TO PACK2;                        20608760 D 127
$ VOIDT                                             20608800 D 127
$! BY KFK
$! DATE 04/08/75
$! MSA CENTRAL = DETROIT
$! THIS PATCH WILL CORRECT THE WAY CONTROL CARD HANDLES PACKET
$! ERRORS. WITHOUT THIS PATCH THE PACKET WILL ONLY BE FLUSHED
$! TO THE NEXT FND OR WAIT CARD NOT TO JUST THE END CARD.
$! *****

```

```

$#PATCH NUMBER 128 FOR TSSMCP CONTAINS 3 CARDS.

```

```

                IF OU GEQ 0 THEN % WE HAVE AN OUTPUT UNIT      28304590 D 128
                IF IU GEQ 0 THEN % WE HAVE AN INPUT UNIT      28304700 D 128
                IF OU GEQ 0 THEN % WE HAVE AN OUTPUT UNIT      28306500 D 128
$! BY KFK
$! DATE 04/10/75
$! MSA CENTRAL = DETROIT
$! THIS PATCH WILL CORRECT AN ERROR IN THE NEW LIBMAIN/DISK PROGRAM.
$! IF THERE WAS A "NULL LIBRARY TRANSFER" THEN IF THE INPUT OR OUTPUT
$! UNIT IS DISK, WORD ONE OF MEMORY WOULD BE OVERWRITTEN BY A ZERO AS
$! THE MCP TRIED TO CLOSE THE UNIT. ONE OF THE MANY PROBLEMS CAUSED
$! BY THIS SITUATION WAS THAT NO USER CODES WOULD BE ACCEPTED BY
$! CONTROL CARD BECAUSE THE MCP'S USERCODE (STORED IN WORD ONE) IS ZERO.
$! NO ERROR MESSAGE WAS GIVEN TO INDICATE THIS CONDITION AND THE SYSTEM
$! CONTINUED TO RUN BUT ALL USERCODES WOULD BE ZERO.
$! *****

```

```

$#PATCH NUMBER 129 FOR TSSMCP CONTAINS 6 CARDS.

```

```

                SHEAT[24]:=MCP;                               07262100 D 129
                SHEAT[24]:=MCP;                               08276365 D 129
                HEADER[2]:=HEADER[5]:=MCP;                   20289113 D 129
                S[1]:=CARD; DS:=9 WDS;                        20289240 D 129
                M[BUF+87]:=MCP;                                20289252 D 129
                SHEAT[24]:=MCP;                                22069212 D 129

```

```

$! BY KFK D 129
$! DATE 04/12/75 D 129
$! MSA CENTRAL - DETROIT D 129
$! THIS PATCH WILL CAUSE PRNPBT/DISK AND AUTO=LDCTRL D 129
$! TO BE EXECUTED WITH THE MCP'S USERCODE. THIS INFORMATION D 129
$! IS PLACED INTO THE LOG FILE. ALSO WITH THIS PATCH D 129
$! PACKET PAGES WILL BE GIVEN THE MCP'S USER CODE NOT D 129
$! THE USER CODE "PACKET". D 129
$!***** D 129

```

```

$#PATCH NUMBER 131 FOR TSSMCP CONTAINS 2 CARDS. D 131

```

```

USFRS: IF(TI=SCAN)FQUAI THEN GO INCSC; 20605700 D 131
        IF(TI=SCAN)=PERIO THEN GO INCSC; 20605702 D 131
$! BY KFK D 131
$! DATE 04/10/75 D 131
$! MSA CENTRAL - DETROIT D 131
$! THIS PATCH WILL GIVE A CONTROL CARD ERROR IF D 131
$! EITHER THE EQUAL SIGN OR USER CODE IS MISSING ON A D 131
$! USER CONTROL CARD. D 131
$!***** D 131

```

```

$#PATCH NUMBER 132 FOR TSSMCP CONTAINS 1 CARD. D 132

```

```

        IF FASZ GTR 0 THEN LIBRARYHELP(5) FLSE IU:=OU:=(FPBPTR1=0)=1; 28246200 D 132
$! BY KFK D 132
$! DATE 04/10/75 D 132
$! MSA CENTRAL - DETROIT D 132
$! THIS PATCH WILL CORRECT A CONDITION IN LIBMAIN/DISK THAT D 132
$! WOULD CAUSE A LIBRARY TAPE TO BE CREATED WITH NO LIBRARY D 132
$! FILES ON IT. THIS TAPE RESULTED ON A NULL LIBRARY TRANSFER. D 132
$! ALSO CORE SPACE FOR THE BUFFERS IS NOT OBTAINED UNLESS THERE ARE D 132
$! SOME FILES TO BE COPIED. D 132
$!***** D 132

```

```

$#PATCH NUMBER 133 FOR TSSMCP CONTAINS 1 CARD. D 133

```

```

        DS:=8LIT"x0x4000"; DS:=8LIT"OPACKET"; 07418700 D 133
$! BY KFK D 133
$! DATE 04/12/75 D 133
$! MSA CENTRAL - DETROIT D 133
$! WITH SOME SYSTEMS THE FIRST I/O DONE TO A PRINTER FOR D 133
$! THE PACKET PAGE WILL CAUSE THE PRINTER TO HANG UP FOR D 133
$! EXTENDED PERIOD OF TIME. THIS WAS CAUSED BY DOING AN D 133
$! INVALID PRINTER OPERATION. THAT IS NO PRINTING AND NO D 133
$! PAPER MOVEMENT. D 133
$! THIS I/O RESULTED WHEN THE MCP CHANGED THE I/O D 133
$! DFSCRIPTOR FOR THE ABORTED LINE SO THAT THIS LINE WOULD D 133

```

```

$! NOT RE PRINTED. D 133
$! NOW A SINGLE SPACE IS GENERATED. D 133
$! THE MCP WOULD RECOVER FROM THIS CONDITION D 133
$! AFTER ABOUT 20 SECONDS. D 133
$!***** D 133

```

```

$#PATCH NUMBER 134 FOR TSSMCP CONTAINS 1 CARD. D 134

ELSE IF U=18 THEN P(DIRECTORYSEARCH("MFID,FID,13),DEL); 28446300 D 134
$! BY SR D 134
$! DATE 04/03/75 D 134
$! THIS PATCH CAUSES HEADER SPACE FOR DUPLICATE DISK FILES TO BE D 134
$! FORGOTTEN. PREVIOUSLY, IF THE LIBRARY MAINTENANCE TASK INVOLVED D 134
$! MANY DUPLICATIONS, THE HEADER SPACE OF EACH DUPLICATE FILE WAS D 134
$! LEFT IN CORE EVENTUALLY RESULTING IN A "NO-MEM" CONDITION ON D 134
$! MIX ZERO. D 134
$!***** D 134

```

```

$#PATCH NUMBER 135 FOR TSSMCP CONTAINS 1 CARD. D 135

MOVE(10,B,T1+2); 41323150 D 135
$! BY KFK D 135
$! DATE 04/22/75 D 135
$! THIS PATCH CORRECTS A CONDITION THAT WOULD CAUSE AN INVALID D 135
$! ADDRESS ON A HALT LOAD IF MEMORY MOD ONE IS OFF LINE. D 135
$!***** D 135

```

```

$#PATCH NUMBER 136 FOR TSSMCP CONTAINS 25 CARDS. D 136

LABEL PE,TF,PA; 18701010 D 136
SWITCH S1=PA,PE,TF,IT,US,D,TD,PR,IOT,TMR,AD,WD; 18701100 D 136
C1: IF (I41=I4) GEQ (-5) AND I4 LEQ 6 THEN 18710100 D 136
BFGIN GO TO S[I4+5]; 18710200 D 136
PA; 18710240 D 136
$ SET OMIT = NOT(PACKETS) 18710242 D 136
IF (I1=PSEUDOMIX[P1MIX]) GEQ 32 THEN 18710244 D 136
I41=PACKETACT[I-32]; 18710246 D 136
$ POP OMIT 18710248 D 136
GO TO INITIATE; 18710249 D 136
PF; 18710250 D 136
$ SET OMIT = NOT(PACKETS) 18710252 D 136
IF (I1=PSEUDOMIX[P1MIX]) GEQ 32 THEN 18710254 D 136
BFGIN 18710256 D 136
I41=PACKETERR[I-32]; 18710258 D 136
PACKETERR[I-32]=TRUE; 18710260 D 136
END; 18710262 D 136
$ POP OMIT 18710264 D 136
GO TO INITIATE; 18710266 D 136

```

```

TEI
$ SET OMIT = NOT(PACKETS)
IF (I:=PSEUDOMIX[P1MIX]) GEQ 32 THEN
  I4:=PACKETERR[I-32]
$ POP OMIT
GO TO INITIATE;
$: BY KFK
$: DATE 04/22/75
$: MSA CENTRAL = DETROIT
$: THIS PATCH ADDS THREE NEW TIME FUNCTIONS:
$: 1) TIME(-3), WILL RETURN THE CURRENT STATUS OF THE PACKETERR BIT
$: 2) TIME(-4), THIS FUNCTION WILL SET THE PACKETERR BIT
$: 3) TIME(-5), WILL RETURN THE CURRENT VALUE OF PACKETACT.
$: THESE THREE FUNCTIONS WERE ADDED TO GIVE A PROGRAM SOME CONTROL
$: OVER THE PACKET. NOW A PROGRAM CAN "KILL" A PACKET BY A TIME
$: FUNCTION RATHER THAN BY DS=ING ITSELF WITH A RUN TIME ERROR(DIV
$: BY ZERO, ETC). ALSO A PROGRAM CAN "SEE" IF A SISTER PROGRAM
$: HAS RUN INTO TROUBLE. AND A PROGRAM CAN TELL HOW MANY JOBS ARE
$: RUNNING FROM THE PACKET AT THIS TIME.
$: TIME(-4) WILL RETURN THE VALUE OF PACKETERR BEFORE IT SETS IT.
$:
$: EXAMPLES: (ALGOL)
$:
$:   IF A LSS 0 THEN          % WE HAVE A PROBLEM
$:     BEGIN                  % LETS KILL THIS RUN
$:       B:=TIME(-4);        % KILL PACKET
$:       GO TO EXIT;         % EXIT PROGRAM
$:     END;                  %
$:
$: AND IN A PROGRAM RUN FROM THE SAME PACKET AT THE SAME TIME
$:
$:     IF BOOLFAN(TIME(-3)) THEN GO TO EXIT; % ERROR IN OTHER PROGRAM.
$:
$: IF THE FIRST PROGRAM FINDS A PROBLEM AND WANTS TO STOP THIS RUN IT
$: SETS THE PACKETERR BIT THEN THE SECOND PROGRAM WILL "SEE" THIS
$: AND ALSO GO TO EOJ.
$: *****
$#PATCH NUMBER     138 FOR TSSMCP CONTAINS 3 CARDS.          D 138

      IF ZIPMIX NEQ 0 THEN           20608112  D 138
      IF (T:=PSEUDOMIX[ZIPMIX]) GEQ 32 THEN           20608114  D 138
        PACKETERR[T-32]:=TRUE;           20608116  D 138
$: BY KFK
$: DATE 04/23/75
$: MSA CENTRAL = DETROIT
$: THIS PATCH WILL SET THE PACKETERR BIT IF THERE
$: IS A ZIP ERROR IN THE PACKET.
$: *****
$#PATCH NUMBER     140 FOR TSSMCP CONTAINS 22 CARDS.        D 140

```

```

        BEGIN T+(A INX @540000000000000)&17[8:38:10];
        IF SEPARATE THEN T+T&(LUN#22)[32:47:1]
        ELSE T+T&(LUN#22)[28:47:1];
$ VOIDT 12727501
$ VOIDT 12739501
        IF KIND=1 THEN M[ALPHA=2]+P(DUP,LOD) &
        (IF SEPARATE THEN 1 ELSE @20)[27:42:6]
        ELSE
                IF NOT SEPARATE THEN
                        IF (TYPE<20) THEN
                                HEADER[73]+P(DUP,LOD)&(@20)[27:42:6];
M[ALPHA]+(*P(DUP))&(@60000)[CTF];
IF TYPE<20 THEN
        M[ALPHA]+(*P(DUP))&(IF SEPARATE THEN 1 ELSE @20)[27:42:6];
        IF NOT SEPARATE THEN M[ALPHA]+(*P(DUP))&1[27:42:6];
END;
IF NOT SEPARATE THEN M[ALPHA]+P(DUP,LOD)&1[27:42:6];
M[ALPHA],[20:1]+1;
IF FIB[14],[FF]#FIB[14],[CF] THEN
IF SEPARATE THEN P(WAITIO(@4000100000,0,U),DEL);
        ELSE P(WAITIO(@4002000000,0,U),DEL);
IF NOT SEPARATE THEN P(WAITIO(@4000100000,0,U),DEL);
$! BY KFK
$! DATE 5/7/75
$! MSA CENTRAL - DETROIT
$! THIS PATCH CHANGES THE USE OF THE SPO OPTION "SEPARATE". IT NOW
$! IS USED TO CONTROL THE PRINTER SPACING AROUND THE LABEL PAGE.
$! THAT IS IF SEPARATE IS RESET THEN ONLY A DOUBLE SPACE IS DONE
$! AFTER THE BEGINING LABEL AND BEFORE THE ENDING LABEL. IF IT IS
$! SET THE OPERATION OF THE PRINTER IS AS IT WAS BEFORE( A FULL
$! PAGE FOR THE LABEL RECORD).
$! THIS WAS DONE TO HELP WITH THE ENERGY CRUNCH AND PER CUBE REQUEST.

```

```

02287240 D 140
02287245 D 140
02287250 D 140
12726500 D 140
12728500 D 140
38280500 D 140
38280600 D 140
38280700 D 140
38281600 D 140
38281700 D 140
38281800 D 140
38590000 D 140
38590100 D 140
38590200 D 140
38592100 D 140
38594000 D 140
38594100 D 140
38594200 D 140
38594300 D 140
38741000 D 140
38741100 D 140
38742100 D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140
D 140

```

```

$#PATCH NUMBER 141 FOR TSSMCP CONTAINS 452 CARDS.
LS:
        IF SC # "" THEN IF SC # LEFTARROW THEN
                BEGIN SI := SI + 1; GO TO LS; END;
        IF SC = LEFTARROW THEN GO TO EXIT;
$ VOIDT 08135001
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
*****
*
*       P R I N T D I R E C T O R Y
*       - - - - -
*
PURPOSE: THIS PROCEDURE HANDLES THE PD, FX, LF, LC AND LS
MESSAGES. ALL MESSAGES ARE FORMS OF THE PD MESSAGE
EXCEPT FOR CERTAIN DIFFERENCES. THE PD MESSAGE WILL
LIST THE NAMES OF THE DESIRED FILES AS WELL AS
CERTAIN PIECES OF INFORMATION IF REQUESTED.
SYNTAX: THE FORMAT OF ALL THE KEYIN MESSAGES WITH THE
EXCEPTION OF THE LF MESSAGE IS AS FOLLOWS:

```

```

00000000 D 141
02615100 D 141
02615200 D 141
02615300 D 141
08095000 D 141
08095050 D 141
08095100 D 141
*08095150 D 141
*08095200 D 141
*08095250 D 141
*08095300 D 141
*08095350 D 141
*08095400 D 141
*08095450 D 141
*08095500 D 141
*08095550 D 141
*08095600 D 141
*08095650 D 141
*08095700 D 141

```





PROCEDURE PRINTDIRECTORY(BUFF);

VAL IF BUFF;  
REAL BUFF;

BEGIN

INTEGER

I, % NORMALLY CONTAINS OPTION NUMBER;  
J, % JUNK.  
CODE; % = 0 - PD,  
% = 1 - EX,  
% = 2 - LC,  
% = 3 - LF,  
% = 4 - LS.

REAL

MFID, % MFID OF DESIRED FILE OR -1 IF "=",  
FID, % FID OF DESIRED FILE OR -1 IF "=",  
C, % ADDRESS OF DISK HEADER,  
D, % MFID OF LAST FILE FOUND BY SEEKNAME,  
E, % FID OF LAST FILE FOUND BY SEEKNAME,  
N, % WORK VARIABLE USED BY SEEKNAME TO SAVE INFO,  
T, % NORMALLY USED TO SAVE DEST. INDEX,  
INFO, % BIT MASK FOR OPTIONS SELECTED,  
LABELREC, % ADDRESS OF LABEL RECORD FOR PRD ( IF LC ).  
USERID, % USERID IF LF.  
X; % JUNK.

ARRAY

HDR[\*], % DESCRIPTOR TO DISK HEADER,  
XLST[\*]; % SOME DAY WE MAY ALLOW EXCEPTION LIST.

BOOLEAN

PRDLOG, % TRUE IF SELECTED FILE IS A PRD.  
FOUNDALFILE; % TRUE IF WE LISTED OUT AT LEAST ONE FILE.

LABFL EXIT, DROPOUT, DUMMY, OPTIONS;

DEFINE NUMOPTS = 7#;

DEFINE PD = (CODE = 0)#,  
EX = (CODE = 1)#,  
LC = (CODE = 2)#,  
LF = (CODE = 3)#,  
LS = (CODE = 4)#;

DEFINE

PRIMARYUSER = HDR[2]#, % PRIV. USER CODE,  
SAVFACOR = HDR[3].[2:10]#, % SAVE FACTOR,  
LASTACCESSDATE = HDR[3].[12:18]#, % LAST ACCESS DATE,  
CREATIONDATE = HDR[3].[30:38]#, % CREATION DATE,  
GUARDFILEMFID = HDR[5]#, % MFID OF GUARD FILE,  
GUARDFILEFID = HDR[6]#, % FID OF GUARD FILE,  
EOFPOINTER = HDR[7]#, % EOF POINTER,  
SEGS PERROW = HDR[8]#, % SEG. PER ROW,  
NOOFROWS = HDR[9].[43:5]#, % NO. OF ROWS DECLARED,  
HEADERADDRESS = HDR.[CF]#, % CORE ADDRESS OF HEADER.

\*\*\*\*\* SUBROUTINES \*\*\*\*\*

SUBROUTINE GETREADY;

%

-----  
BEGIN

08098600 D 141  
08098650 D 141  
08098700 D 141  
08098750 D 141  
08098800 D 141  
08098850 D 141  
08098900 D 141  
08098950 D 141  
08099000 D 141  
08099050 D 141  
08099100 D 141  
08099150 D 141  
08099200 D 141  
08099250 D 141  
08099300 D 141  
08099350 D 141  
08099400 D 141  
08099450 D 141  
08099500 D 141  
08099550 D 141  
08099600 D 141  
08099650 D 141  
08099700 D 141  
08099750 D 141  
08099800 D 141  
08099850 D 141  
08099900 D 141  
08099950 D 141  
08100000 D 141  
08100050 D 141  
08100100 D 141  
08100150 D 141  
08100200 D 141  
08100250 D 141  
08100300 D 141  
08100350 D 141  
08100500 D 141  
08100550 D 141  
08100600 D 141  
08100650 D 141  
08100700 D 141  
08100750 D 141  
08100800 D 141  
08100850 D 141  
08100900 D 141  
08100950 D 141  
08101000 D 141  
08101050 D 141  
08101100 D 141  
08101150 D 141  
08101200 D 141  
08101250 D 141  
08101300 D 141  
08101350 D 141  
08101400 D 141  
08101450 D 141  
08101500 D 141

```

CODE := BUFF.[9:16]
BUFF := (T:=BUFF).[15:15] - 1
INFO := 0 & (LC OR LF OR LS)[42:47:1] & LS[43:47:1]
END OF GETREADY;
%
SUBROUTINE GETFILESPECIFIER;
% -----
BEGIN
NAMEID(MFID,T); % GET MFID (OR USERCODE IF "LF")
IF LF THEN USFRID := MFID; % FIRST THING IS USFRID FOR LF.
IF MFID = "+ " OR LF THEN
MFID := - 1;
NAMEID(FID,T);
IF FID = "/" THEN % GET FID
BEGIN
NAMEID(FID,T);
NAMEID(N,T); % GET NEXT ITEM.
END
ELSE % NO FID SPECIFIED IMPLIES FID OF "="
BEGIN
N := FID;
FID := -1;
END;
IF FID.[6:6] = LEFTARROW OR LF THEN % NO FID SPECIFIED
FID := - 1;
END OF GETFILESPECIFIER;
%
SUBROUTINE PROCSOPTIONLIST;
% -----
BEGIN
WHILE N # "+ " DO % ACCUMULATE OPTIONS
BEGIN
FOR I := -1 STEP 1 UNTIL (NUMOPTS = 2) DO
DUMMY := P(.OPTIONS,I,+ ,LOD) = N THEN % MATCHES AN OPTION WORD
BEGIN
INFO := INFO OR TWO(I+1); % SET BIT CORRESPONDING TO OPT
GO TO DROPOUT;
END
ELSE % CHECK FOR "ALL"
IF N = "ALL" THEN % SET ALL OPTION WORD BITS.
INFO := NOT 0;
DROPOUT:
NAMEID(N,T); % GET NEXT OPTION WORD.
IF N = ",", THEN NAMEID(N,T); % SKIP OVER COMMA.
END;
GO TO EXIT;
%
OPTIONS :=: "RFGS" " " % OPTION 0 (INFO.[47:1])
"LAST" " " % OPTION 1 (INFO.[46:1])
"DATE" " " % OPTION 2 (INFO.[45:1])
"SIZE" " " % OPTION 3 (INFO.[44:1])
"SECURIT" " " % OPTION 4 (INFO.[43:1])
"CREATOR" " " % OPTION 5 (INFO.[42:1])
"SAVE" " " % OPTION 6 (INFO.[41:1])
EXIT:
END OF PROCSOPTIONLIST;
%

```

```

08101550 D 141
08101600 D 141
08101650 D 141
08101700 D 141
08101750 D 141
08101800 D 141
08101850 D 141
08101900 D 141
08101950 D 141
08102000 D 141
08102050 D 141
08102100 D 141
08102150 D 141
08102200 D 141
08102250 D 141
08102300 D 141
08102350 D 141
08102400 D 141
08102450 D 141
08102500 D 141
08102550 D 141
08102600 D 141
08102650 D 141
08102700 D 141
08102750 D 141
08102800 D 141
08102850 D 141
08102900 D 141
08102950 D 141
08103000 D 141
08103050 D 141
08103100 D 141
08103150 D 141
08103200 D 141
08103250 D 141
08103300 D 141
08103350 D 141
08103400 D 141
08103450 D 141
08103500 D 141
08103550 D 141
08103600 D 141
08103650 D 141
08103700 D 141
08103750 D 141
08103800 D 141
08103850 D 141
08103900 D 141
08103950 D 141
08104000 D 141
08104050 D 141
08104100 D 141
08104150 D 141
08104200 D 141
08104250 D 141
08104300 D 141
08104350 D 141

```

```

SUBROUTINE GETSET;
% -----
  BFGIN
  IF FX OR INFO # 0 THEN % WE WILL NEED THE HDR;
  HDR := IOQUE & SPACE(30) [CTC];
  END OF GETSET;
%
BOOLEAN SUBROUTINE WEGOTAFILE;
% -----
  BFGIN
  SFFKNAM(MFID,FID,C,D,F,N,XLST); % FIND A FILE?
  WEGOTAFILE := C # 0; % C IS ADDRESS OF DISK HEADER.
  END OF WEGOTAFILE;
%
BOOLEAN SUBROUTINE WEWANTTHISFILE;
% -----
  BFGIN
  PBDTG := ((D EQV "PRD   ") = NOT 0 OR (D EQV "PUD   ")
            = NOT 0);
  IF HEADERADDRESS # 0 THEN % WE NEED THE HEADER.
  DISKWAIT(=HEADERADDRESS,30,C); % READ HEADER.
  IF LF THEN % CHECK TO SEE IF WE WANT THIS FILE.
  WEWANTTHISFILE := PRIMARYUSER = USERID
  ELSE
  IF FX THEN
  BFGIN
  STREAM(A:=CALCULATEPURGE(=SAVEFACTOR),X:=FX);
  BEGIN
  SI := LOC A; DS := 8 OCT;
  END;
  WEWANTTHISFILE := X > LASTACCESSDATE; % TRUE IF FILE EXPIRED
  END
  ELSE
  WEWANTTHISFILE := TRUE;
  END OF WEWANTTHISFILE;
%
SUBROUTINE PUTINFILENAME;
% -----
  BFGIN
  STREAM(A:=0 : D, F, RUFF); % SET UP FILE NAME;
  BEGIN
  SI := LOC D;
  DS := LIT " ";
  2 ( SI := SI + 1; DS := 7 CHR; DS := LIT "/" );
  DI := DI - 1;
  DS := 2 LIT " "; % NEED 2 SPACES TO ALLOW FOR ARROW.
  A := DI;
  END;
  T := P; % SAVE OFF DEST. INDEX.
  END OF PUTINFILENAME;
%
SUBROUTINE DORECS;
% -----
  BFGIN
  STREAM(A:=IF PBDTG THEN FOFPOINTER*5 ELSE EOFPOINTER+1 :T);
  BEGIN

```

```

08104400 D 141
08104450 D 141
08104500 D 141
08104550 D 141
08104600 D 141
08104650 D 141
08104700 D 141
08104750 D 141
08104800 D 141
08104850 D 141
08104900 D 141
08104950 D 141
08105000 D 141
08105050 D 141
08105100 D 141
08105150 D 141
08105200 D 141
08105250 D 141
08105300 D 141
08105350 D 141
08105400 D 141
08105450 D 141
08105500 D 141
08105550 D 141
08105600 D 141
08105650 D 141
08105700 D 141
08105750 D 141
08105800 D 141
08105850 D 141
08105900 D 141
08105950 D 141
08106000 D 141
08106050 D 141
08106100 D 141
08106150 D 141
08106200 D 141
08106250 D 141
08106300 D 141
08106350 D 141
08106400 D 141
08106450 D 141
08106500 D 141
08106550 D 141
08106600 D 141
08106650 D 141
08106700 D 141
08106750 D 141
08106800 D 141
08106850 D 141
08106900 D 141
08106950 D 141
08107000 D 141
08107050 D 141
08107100 D 141
08107150 D 141
08107200 D 141

```

```

        DS := 9 LIT "RECORDS: ";
        SI := LOC A;
        DS := 8 DEC; % CONVERT NUMBER OF RECORDS TO DEC.
        A := DI; % SAVE OFF DI BEFORE ZERO SUPPRESSING.
        DI := DI - 8;
        DS := 7 FILL;
    END;
    T := P; % SAVE DESTINATION ADDRESS.
END OF DORECS;
%
SUBROUTINE DODATEORLAST;
% -----
    BFGIN
        STREAM(X:=X);
        BFGIN
            SI := X;
            DS := 8 DEC; % CONVERT DATE TO DECIMAL.
        END;
        GIMEDATE(X),[CF],-X); % CONVERT JULIAN DATE TO 6 DIGITS.
        STREAM(A:=(I=1) ; X, T);
        BEGIN
            A ( DS := 10 LIT "ACCESSED: "; JUMP OUT TO L1);
            DS := 9 LIT "CREATED: ";
        L1:
            SI := LOC X;
            SI := SI + 2;
            3 ( DS := 2 CHR; DS := LIT "/" );
            DI := DI - 1; % FRASE THE EXTRA SLASH.
            A := DI; % SAVE DEST. INDEX.
        END;
        T := P; % SAVE DESTINATION ADDRESS.
    END OF DODATEORLAST;
%
SUBROUTINE DOSIZE;
% -----
    BFGIN
        NT2 := NOOFROWS; % NO. OF ROWS DECLARED.
        NT1 := 0; % NUMBER OF ROWS PROCESSED.
        FOR J := 1 STEP 1 UNTIL NT2 DO % CHECK TO SEE IF ROW EXISTS.
            IF HDR(J+9) # 0 THEN NT1 := NT1 + 1; % BUMP UP COUNT.
        STREAM(A:=NT1*SEGSPERROW ; T);
        BEGIN
            DS := 10 LIT "SEGMENTS: ";
            SI := LOC A;
            DS := 8 DEC;
            A := DI;
            DI := DI - 8;
            DS := 7 FILL;
        END;
        T := P; % SAVE DESTINATION ADDRESS.
    END OF DOSIZE;
%
SUBROUTINE DOSECURITY;
% -----
    BFGIN
        J := IF PRIMARYUSER = 0 THEN 0 % FREE FILE
            ELSE IF GUARDFILEMFD = MARK THEN % UNLOCK OR PUBLIC

```

```

08107250 D 141
08107300 D 141
08107350 D 141
08107400 D 141
08107450 D 141
08107500 D 141
08107550 D 141
08107600 D 141
08107650 D 141
08107700 D 141
08107750 D 141
08107800 D 141
08107850 D 141
08107950 D 141
08108000 D 141
08108050 D 141
08108100 D 141
08108150 D 141
08108200 D 141
08108250 D 141
08108300 D 141
08108350 D 141
08108400 D 141
08108450 D 141
08108500 D 141
08108550 D 141
08108600 D 141
08108650 D 141
08108700 D 141
08108750 D 141
08108800 D 141
08108850 D 141
08108900 D 141
08108950 D 141
08109000 D 141
08109050 D 141
08109100 D 141
08109150 D 141
08109200 D 141
08109250 D 141
08109300 D 141
08109350 D 141
08109400 D 141
08109450 D 141
08109500 D 141
08109550 D 141
08109600 D 141
08109650 D 141
08109700 D 141
08109750 D 141
08109800 D 141
08109850 D 141
08109900 D 141
08109950 D 141
08110000 D 141
08110050 D 141
08110100 D 141

```

```

                IF GUARDFILEFID = MARK THEN 1 % UNLOCKED
                ELSE 2 % PUBLIC
            ELSE IF GUARDFILEMFID < 0 THEN 3 % PRIVATE
                ELSE 4 % LOCKED
    STREAM(J ; A:=GUARDFILEMFID, R:=GUARDFILEFID, T);
    BEGIN
        DS := 10 LIT "SECURITY: ";
        CI := CI + J;
        GO TO FREE;
        GO TO UNLOCK;
        GO TO PUBLIC;
        GO TO PRIVATE;
    LOCK: DS := 6 LIT "LOCKED"; GO TO EXIT;
    PRIVATE: DS := 22 LIT "PRIVATE (SECURED WITH ";
        SI := LOC A;
        2 ( SI := SI + 1; DS := 7 CHR; DS := LIT "/" );
        DI := DI - 1;
        DS := LIT ")";
        GO TO EXIT;
    PUBLIC: DS := 6 LIT "PUBLIC"; GO TO EXIT;
    UNLOCK: DS := 8 LIT "UNLOCKED"; GO TO EXIT;
    FREE: DS := 4 LIT "FREE";
    EXIT: J := DI;
        END;
        T := P; % SAVE DESTINATION ADDRESS.
    END OF DOSECURITY;
%
% SUBROUTINE DOCREATOR;
% -----
    BEGIN
        $SFT OMIT = PACKETS
        IF J:=(PBDTOG AND (E.[30:18] = 1)) THEN % REEL 1 OF PBD
        $POP OMIT
        $SET OMIT = NOT(PACKETS)
        IF J:=(PBDTOG AND (E.[42:6] = 1)) THEN % REEL 1 OF PBD
        $POP OMIT
        BEGIN
            IF LABELREC = 0 THEN LABELREC := SPACE(30);
            DISKWAIT(-LABELREC,30,HDR[10]+2);
        END;
        STREAM(J ; B:=PRIMARYUSER=0, CI:=PRIMARYUSER,
            DI:=LABELREC INX 12, T);
        BEGIN
            DS := 9 LIT "CREATOR: ";
            R ( DS := 4 LIT "NONE"; JUMP OUT TO L2 );
            SI := LOC C;
            SI := SI + 1;
            DS := 7 CHR;
        L2:
            J ( DS := 2 LIT " (" );
            SI := DI;
            2 ( SI := SI + 1; DS := 7 CHR; DS := LIT "/" );
            DI := DI - 1; DS := 4 LIT " OF ";
            2 ( SI := SI + 1; DS := 7 CHR; DS := LIT "/" );
            DI := DI - 1; DS := LIT ")";
            J := DI;
        END;
    END;

```

```

08110150 D 141
08110200 D 141
08110250 D 141
08110300 D 141
08110350 D 141
08110400 D 141
08110450 D 141
08110500 D 141
08110550 D 141
08110600 D 141
08110650 D 141
08110700 D 141
08110750 D 141
08110800 D 141
08110850 D 141
08110900 D 141
08110950 D 141
08111000 D 141
08111050 D 141
08111100 D 141
08111150 D 141
08111200 D 141
08111250 D 141
08111300 D 141
08111350 D 141
08111400 D 141
08111450 D 141
08111500 D 141
08111550 D 141
08111600 D 141
08111650 D 141
08111700 D 141
08111750 D 141
08111800 D 141
08111850 D 141
08111900 D 141
08111950 D 141
08112000 D 141
08112050 D 141
08112100 D 141
08112150 D 141
08112200 D 141
08112250 D 141
08112300 D 141
08112350 D 141
08112400 D 141
08112450 D 141
08112500 D 141
08112550 D 141
08112600 D 141
08112650 D 141
08112700 D 141
08112750 D 141
08112800 D 141
08112850 D 141
08112900 D 141
08112950 D 141

```

T := P; % SAVE DESTINATION ADDRESS.	08113000	D	141
END OF DOCREATOR;	08113050	D	141
%	08113100	D	141
SUBROUTINE DOSAVEFACTOR;	08113150	D	141
% -----	08113200	D	141
BEGIN	08113250	D	141
STREAM(A := SAVEFACTOR ; T);	08113300	D	141
BEGIN	08113350	D	141
DS := 6 LIT "SAVE: ";	08113400	D	141
SI := LOC A;	08113450	D	141
DS := 3 DEC;	08113500	D	141
A := DI;	08113550	D	141
DI := DI - 3;	08113600	D	141
DS := 2 FILL;	08113650	D	141
END;	08113700	D	141
T := P; % SAVE DESTINATION ADDRESS.	08113750	D	141
END OF DOSAVEFACTOR;	08113800	D	141
%	08113850	D	141
SUBROUTINE DOOPTIONS;	08113900	D	141
% -----	08113950	D	141
BEGIN	08114000	D	141
FOR I := 0 STEP 1 UNTIL (NUMOPTS - 1) DO % SET IF OPTION BIT SET.	08114050	D	141
IF (TWO(I) AND INFO) # 0 THEN %OPTION SELECTED.	08114100	D	141
BEGIN	08114150	D	141
CASE I OF	08114200	D	141
BEGIN	08114210	D	141
DORECS; % CASE 0 = "RECS"	08114250	D	141
BEGIN % CASE 1 = "LAST"	08114290	D	141
X := LASTACCESSDATE;	08114300	D	141
DODATEFORLAST;	08114310	D	141
END OF CASE 1;	08114320	D	141
BEGIN % CASE 2 = "DATE"	08114340	D	141
X := CREATIONDATE;	08114350	D	141
DODATEFORLAST;	08114360	D	141
END OF CASE 2;	08114370	D	141
DOSIZE; % CASE 3 = "SIZE"	08114400	D	141
DOSECURITY; % CASE 4 = "SECURITY"	08114450	D	141
DOCREATOR; % CASE 5 = "CREATOR"	08114500	D	141
DOSAVEFACTOR; % CASE 6 = "SAVE"	08114550	D	141
END OF CASES;	08114600	D	141
STREAM(I ; T); % PUT COMMA AFTER LAST OPTION.	08114650	D	141
BEGIN	08114700	D	141
DS := 2 LIT ", ";	08114750	D	141
I := DI;	08114800	D	141
END;	08114850	D	141
T := P;	08114900	D	141
END OF LOOP TO PROCESS OPTIONS;	08114950	D	141
END OF DOOPTIONS;	08115000	D	141
%	08115050	D	141
SUBROUTINE MAKETHEMESSAGE;	08115100	D	141
% -----	08115150	D	141
BEGIN	08115200	D	141
IF FOUND A FILE THEN % WE NEED A BUFFER.	08115250	D	141
BUFF := SPACE(30);	08115300	D	141
PUTINFILENAME;	08115350	D	141
DOOPTIONS;	08115400	D	141
STREAM(T); % PUT IN THE LEFT ARROW.	08115450	D	141

```

      BFGIN
      DI := DI -2;
      DS := LIT LEFTARROW;
      END;
      END OF MAKETHEMESSAGE;
%
SUBROUTINE COMPLAIN;
% -----
      BFGIN
      STREAM(BUFF);
      DS := 8 LIT " NULL "; % ORIGINAL INPUT IS AT BUFF + 1.
      SPOUT(BUFF);
      END OF COMPLAIN;
%
SUBROUTINE FORGETFEVERYTHING;
% -----
      BFGIN
      IF HEADERADDRESS # 0 THEN FORGETSPACE(HEADERADDRESS);
      IF LABELREC # 0 THEN FORGETSPACE(LABELREC);
      END OF FORGETFEVERYTHING;
%***** S T A R T   O F   C O D E *****
      GFTREADY;
      GFTFILESPECIFER;
      PROCCSOPTIONLIST;
      GFTSET;
      WHILE WEGOTAFILE DO
      IF WEWANTTHISFILE THEN
      BEGIN
      MAKETHEMESSAGE;
      SPOUT(BUFF);
      FOUNDADFILE := TRUE;
      END;
      IF NOT FOUNDADFILE THEN COMPLAIN;
      FORGETFEVERYTHING;
      END OF PRINTDIRECTORY;
%: DATE 9/10/75
%: BY JTC - MSA CENTRAL
%: THIS PATCH REWRITES THE PRINTDIRECTORY PROCEDURE. MUCH NEEDLESS
%: CODE HAS BEEN ELIMINATED (THE PROCEDURE IS 30 WORDS SMALLER) AND
%: SOME NEW OPTIONS ARE AVAILABLE. THE SYNTAX FOR ALL MESSAGES
%: HANDLED BY THIS PROCEDURE (PD,EX,LF,LC, AND LS) WAS BEEN CHANGED
%: SLIGHTLY IN THAT A SLASH IS NOW REQUIRED BETWEEN THE FIRST AND
%: LAST NAMES OF A FILE (A SPACE WOULD SUFFICE PREVIOUSLY) AND
%: MULTIPLE OPTION WORDS MAY BE SPECIFIED AFTER ANY OF THE MESSAGES.
%: IN ADDITION, FOUR NEW OPTION WORDS HAVE BEEN ADDED. "SAVE"
%: LISTS OUT THE SAVE-FACTOR OF THE FILE, "CREATOR" LISTS OUT THE
%: PRIVILEGED USERCODE ASSOCIATED WITH THE FILE, "SECURITY"
%: LISTS OUT THE ACCESS RIGHTS OF THE FILE, I.E, LOCKED, UNLOCKED,
%: PUBLIC, PRIVATE, OR FREE, AND "ALL" LISTS OUT ALL OF THE INFOR-
%: MATION AVAILABLE FROM EACH OF THE INDIVIDUAL OPTIONS.
%:
%: SOME EXAMPLES OF THE MESSAGES THAT ARE POSSIBLE FOLLOW.
%:
%: PD A/B RECS,DATE,SIZE
%: LF SITE ALL
%: LCPBD RECS
%:

```

```

08115500 D 141
08115550 D 141
08115600 D 141
08115650 D 141
08115700 D 141
08115750 D 141
08115800 D 141
08115850 D 141
08115900 D 141
08115950 D 141
08116000 D 141
08116050 D 141
08116100 D 141
08116150 D 141
08116200 D 141
08116250 D 141
08116300 D 141
08116350 D 141
08116400 D 141
08116450 D 141
08116500 D 141
08116550 D 141
08116600 D 141
08116650 D 141
08116700 D 141
08116750 D 141
08116800 D 141
08116850 D 141
08116900 D 141
08116950 D 141
08117000 D 141
08117050 D 141
08117100 D 141
08117150 D 141
08117200 D 141
99990000 D 141
99990100 D 141
99990200 D 141
99990300 D 141
99990400 D 141
99990500 D 141
99990600 D 141
99990700 D 141
99990800 D 141
99990900 D 141
99991000 D 141
99991100 D 141
99991200 D 141
99991300 D 141
99991400 D 141
99991500 D 141
99991600 D 141
99991700 D 141
99991800 D 141
99991900 D 141
99992000 D 141
99992100 D 141

```

\$: CONSULT THE HEADING BLOCK OF THE PRINTDIRFACTORY PROCEDURE	99992200	D	141
\$: FOR ADDITIONAL INFORMATION.	99992300	D	141

\$#PATCH NUMBER 142 FOR TSSMCP CONTAINS 1 CARD:	00000000	D	142
---	----------	---	-----

REAL NEU; DEFINE U=UT#,UA=UT#,NFU1=NEU+J#,NFU2=NEU+NEU#;	06351055	D	142
\$: DATE 10/1/75	99990000	D	142
\$: BY JTC - MSA CENTRAL	99990100	D	142
\$: THIS PATCH CORRECTS A PROBLEM IN SHAREDISK MCPS	99990200	D	142
\$: WHICH COULD CAUSE INVALID LINK ERRORS. THE GLOBAL	99990300	D	142
\$: VARIABLE AVS WAS BEING USED INCORRECTLY IN THE	99990400	D	142
\$: PROCEDURE USERDISKSPFCIALCASE.	99990500	D	142

\$#PATCH NUMBER 143 FOR TSSMCP CONTAINS 5 CARDS.	00000000	D	143
--	----------	---	-----

LABEL PE,TE,PA,LL;	18701010	D	143
SWITCH S := LL,PA,PE,TE,IT,US,D,TD,PR,IOT,TMR,AD,WD;	18701100	D	143
C1: IF (I4:=I4) ≥ (-6) AND I4 ≤ 6 THEN	18710100	D	143
BEGIN GO TO S[I4+6];	18710200	D	143
LL: I4 := LOGLINE.[40:8]; GO TO INITIATE;	18710230	D	143
\$: DATE 9/25/75	99990000	D	143
\$: BY JTC - MSA CENTRAL;	99990100	D	143
\$: THIS PATCH IMPLEMENTS TIME(-6) WHICH RETURNS THE	99990200	D	143
\$: LOGICAL LINE NUMBER FROM WHICH A PARTICULAR PROGRAM	99990300	D	143
\$: IS RUNNING. BATCH PROGRAM WILL ALWAYS RETURN A ZERO.	99990400	D	143

\$#PATCH NUMBER 144 FOR TSSMCP CONTAINS 7 CARDS.	00000001	D	144
--	----------	---	-----

PROCEDURE FORMTIME(W,T); VALUE W,T; REAL W,T; FORWARD;	00480010	D	144
(S+2+(M[BUF INX Z],[1:5]#>""))[8:38:10];	02173296	D	144
FORMTIME([NT1],XCLOCK+P(RTR));	02173297	D	144
STREAM(NI=S-1,CL:=S*8-Y,AA:=BUF INX Z,BB := NT1,	02173300	D	144
BUF:=[RUF[R]]);	02173301	D	144
REGIN DS := 7 LIT " "; SI := LOC BR; DS := 8 CHR;	02173305	D	144
DS := 9 LIT " "; SI := AA;	02173306	D	144
\$: DATE 10/20/75	99990000	D	144
\$: BY JTC - MSA CENTRAL;	99990100	D	144
\$: THIS PATCH ADDS THE TIME OF DAY TO EACH MESSAGE AS IT IS	99990200	D	144
\$: ENTERED INTO THE PACKET PAGE.	99990300	D	144

\$#PATCH NUMBER 145 FOR TSSMCP CONTAINS 7 CARDS		D	145
---	--	---	-----

LABEL IPCERR;	19504100	D	145
INIT,SLOW,C34,C35,C36,C37,TW,TW,C40,C41,IPCERR,	19508100	D	145
IPCERR,IPCERR,IPCERR,IPCERR,IPCERR,TW,SLOW;	19508200	D	145



IPCERR: TERMINATE (P1MIX);	19575000	D	145
TERMINALMESSAGE (96);	19575100	D	145
"8INVALID", "8 IPC C",	41445500	D	145
"20M....",	41445600	D	145
\$: BY JTC - MSA CENTRAL, SOFTWARE FLASH 161		D	145
\$:*****		D	145
\$#PATCH NUMBER 146 FOR TSSMCP CONTAINS 41 CARDS.	00000000	D	146
LABFL FOF, CLEANUP;	38363000	D	146
DEFINE H = HEADER#;	38385700	D	146
\$ VOIDT 38477001	38446500	D	146
BEGIN % COMPILE TO LIBRARY	38447000	D	146
FOR T1 := 15 STEP 1 UNTIL 22 DO % COPY PROC, I/O,	38447500	D	146
SFGO[T1] := SKEL[T1]; % STACK, COMMON, PRIO., ETC.	38448000	D	146
IF (T2 := SKEL[13]) # 0 THEN % LAB. EQN. ENTRIES	38448500	D	146
BEGIN	38449000	D	146
SKEL[13] := 0; % SO TERMINATE WONT SEE ESPDISK STUFF	38449500	D	146
DISKWAIT(SKFL,[CF],30,T3); % REWRITE SKFL. SHEET	38450000	D	146
SFGO[15] := H[7] + 1; % ADDRESS OF 1ST ENTRY	38450500	D	146
DO	38451000	D	146
BEGIN % COPYING LABEL EQN. INFO INTO CODE FILE	38451500	D	146
IF DISKADDRESS(MID,FID,FPB[FNUM+3],	38452000	D	146
(T1:=H[7];H[7]+1),H,0) = 0 THEN % NO ROOM	38452500	D	146
BEGIN	38453000	D	146
DO	38453500	D	146
BEGIN	38453600	D	146
DISKWAIT(=SKFL,[CF],30,T2); % GET SEG.	38453700	D	146
FORGETESPDISK(T2);	38454000	D	146
END	38454100	D	146
UNTIL (T2 := SKEL[29]) = 0;	38454500	D	146
FILEMESS(="DISK ", "OVRFLOW",MID,FID,	38455000	D	146
R,D,C);	38455500	D	146
END	38456000	D	146
ELSE	38456500	D	146
BEGIN	38457000	D	146
DISKWAIT(=SKEL,[CF],30,T2); % GET SEG.	38457500	D	146
FORGETESPDISK(T2);	38458000	D	146
IF (T2 := SKEL[29]) # 0 THEN % MORE OF EM	38458500	D	146
SKEL[29] := T1 + 1; % ADDR. IN CODE FILE	38459000	D	146
DISKWAIT(SKEL,[CF],30,I:=(H[T1 DIV H[8]]	38459500	D	146
+10] + T1 MOD H[8]));	38460000	D	146
END	38460500	D	146
END	38461000	D	146
UNTIL T2 = 0;	38461500	D	146
END OF PROCESSING LABEL EQUATION INFO;	38462000	D	146
SEGO[6] := NABS(*P(DUP)); % NFW FORMAT SEG. ZERO	38462500	D	146
H[4],[10:1] := 1; % MARK AS PROG. FILE	38463000	D	146
DISKWAIT(SFGO,[CF],30,H[10]); % REWRITE SFG. ZERO	38463500	D	146
END OF COMPILE TO LIBRARY PROCESSING;	38464000	D	146
\$: DATE 1/19/76	99990000	D	146
\$: BY JTC - MSA CENTRAL,	99990100	D	146
\$: THIS PATCH CORRECTS AN UNLIKELY ERROR IN DISKCLOSE WHICH	99990200	D	146
\$: CAN CAUSE ESP DISK SEGMENTS TO BE LEFT IN USE IF A COMPILER	99990300	D	146

\$: GETS DS=ED FOR DISK OVERFLOW. IN ADDITION, THE PATCH USES	99990400	D	146
\$: THIS OPPORTUNITY TO REWRITE THE EXCEPTIONALLY BAD LOGIC	99990500	D	146
\$: CONTAINED IN THIS ROUTINE.	99990600	D	146

\$#PATCH NUMBER 147 FOR TSSMCP CONTAINS 2 CARDS D 147

UPTIMEOUT(MIX);	24035300	D	147
DOWNTIMEOUT(MIX);	24096800	D	147
\$: BY JTC - MSA CENTRAL, SOFTWARE FLASH 162		D	147
\$:*****		D	147

\$#PATCH NUMBER 148 FOR TSSMCP CONTAINS 1 CARD D 148

TFMP + F & K [CTF];	18273055	D	148
\$: BY JTC - MSA CENTRAL, SOFTWARE FLASH 163		D	148
\$: SEE ALSO DREXFL U. PATCHES 601,662,748		D	148
\$:*****		D	148

\$#PATCH NUMBER 149 FOR TSSMCP CONTAINS 7 CARD. D 149

DS := 8 DEC; DS := 2 LIT " +";	03733500	D	149
DI := DI - 10; DS := 7 FILL;	03734000	D	149
SI := F; DS := 10 CHR;	03735000	D	149
BEGIN SI:=LOC A; DS:=8 DEC; DS:=2 LIT " +";	19559120	D	149
DI:=DI-10; DS:=8 FILL; A:=DI;	19559140	D	149
DI:=T; SI:=A; DS:=10 CHR;	19559160	D	149
TWXOUT(T,10,0&1[CTF]&1[2:47:1],R6);	19559200	D	149
\$: DATE 2/11/76	99990000	D	149
\$: BY JTC - MSA CENTRAL	99990100	D	149
\$:	99990200	D	149
\$: THIS PATCH CAUSES AN EXTRA BLANK TO BE TYPED OUT AFTER THE SEQUENCE	99990300	D	149
\$: NUMBER IS SEQUENCE MODE. THIS IS CONVENIENT FOR LINING	99990400	D	149
\$: INPUT WITH PREVIOUSLY PRINTED OUTPUT SINCE THE LIST AND PRINT	99990500	D	149
\$: COMMANDS OF CANDE PRINT AN EXTRA SPACE AFTER THE SEQUENCE NUMBER.	99990600	D	149
\$:*****	99999999	D	149

\$#PATCH NUMBER 150 FOR TSSMCP CONTAINS 12 CARDS. D 150

STREAM(LCFLG:=LONGCARRIAGE[S1],V:=(V:=GETAREA(0))+1);	03826100	D	150
BEGIN	03826200	D	150
DS := 6 LIT "DEL<<#";	03826300	D	150
LCFLG (DS := 3 LIT "<");	03826400	D	150
DS := LIT LFFTARROW;	03826500	D	150
END;	03826600	D	150
STREAM(LCFLG:=LONGCARRIAGE[S1],V:=V+1);	03900000	D	150

```

BEGIN
  DS := 3 LIT "<<#";
  LCFLG (DS := 3 LIT "<");
  DS := LIT LEFTARROW;
END;

$! DATE 2/11/76
$! BY JTC - MSA CFNTRAL
$!
$! THIS PATCH ADDS AN EXTRA DELAY FOR TERMINALS WITH LONG CARRIAGES
$! AFTER THE CARRIAGE RETURN TYPED OUT WHEN THE USER TYPES A
$! LEFTARROW. THIS PREVENTS THE BEGINNING OF SEQUENCE NUMBERS
$! FROM BEING LOST IN SEQUENCE MODE.

```

```

03900100 D 150
03900200 D 150
03900300 D 150
03900400 D 150
03900500 D 150
99990000 D 150
99990100 D 150
99990200 D 150
99990300 D 150
99990400 D 150
99990500 D 150
99990600 D 150

```

```
$#PATCH NUMBER 151 FOR TSSMCP CONTAINS 38 CARDS.
```

```
00000000 D 151
```

```

*ITSALLBATCH = TOGGLE.[16:1]*
  END
  ELSE % ITS BELOW THE FENCE
  IF ITSALLBATCH THEN % WE WILL WAIT AWHILE
  BEGIN
    TANKS[LL] := NABS(T);
    CLICK := CLOCK + P(RTR) + 128; % 2 SEC.
    SLEEP(0,0); % WAIT FOR TIME OUT.
    GO TO START; % TRY AGAIN.
  END;
  IF P(0,RDS) < FENCE THEN % BELOW THE FENCE
  BEGIN CLICK := CLOCK + P(RTR) + 1023 + 1023; % 34 SEC.
  COMPLEXSLEEP(INPUTANK[T], INPUTN OR TERMSET(P1MIX));
  END ELSE
  IF [MEM[MIX,0]].[CF] ≥ FENCE OR ITSALLBATCH THEN
  IF T < @20000 OR T > @100000 THEN
  SUBROUTINE CLEARALLTANKS;
  BEGIN CLEARTANK(LOGLINE,0);
  TANKS[I:=LOGLINE,[40:8]] := (*P(DUP))&0[CTC];
  IF WORKING THEN % MAKE SURE HARRY ISNT
  IF HARRYSTA = I THEN % DIDDLING OUR LINE
  COMPLEXSLEEP(NOT WORKING OR HARRYSTA ≠ I);
  CLEARINPUTTANK;
  TANKOK[I] := 0;
  END OF CLEARALLTANKS;
  IF LOGLINE.[33:7] ≠ 0 AND P1MIX ≠ CANDYINX THEN
  CLEARALLTANKS;
  I := 0; % RE-INITIALIZE FOR CORF SEARCH
  CLEARALLTANKS;
  $VOIDT 14367035
  BFLOW I := (S[2],[8:10]=5) OR ITSALLBATCH;
  IF FENCE.[32:16] ≠ FENCE OR FENCE < 8192 THEN FENCE I= 16384;
  IF J < 7 THEN % NEED TO FIND HIGHEST ON LINE MOD BELOW FENCE
  +(CHUNKMAX=0) % IN CASE ITS ALL BATCH
  I:=CHUNKMAX+(CHUNKMAX=0);
  IF (T:=M[A+19],[32:16]) ≠ FENCE AND T ≠ 0 THEN
  ≥ @20000 AND T ≤ @100000 THEN FENCE I= T;
  ITSALLBATCH I= FENCE ≥ @60000;
  $! DATE 2/11/76

```

```

00081785 D 151
03147000 D 151
03147100 D 151
03147200 D 151
03147300 D 151
03147400 D 151
03147500 D 151
03147600 D 151
03147700 D 151
03147800 D 151
03299700 D 151
03299730 D 151
03299750 D 151
03299780 D 151
07500000 D 151
09916000 D 151
14351440 D 151
14351445 D 151
14351450 D 151
14351455 D 151
14351460 D 151
14351465 D 151
14351470 D 151
14351475 D 151
14351480 D 151
14360510 D 151
14360520 D 151
14360530 D 151
14367021 D 151
14367022 D 151
20163200 D 151
44104000 D 151
44109900 D 151
44151050 D 151
44163100 D 151
44191020 D 151
44191040 D 151
44420000 D 151
99990000 D 151

```

```

$# BY JTC = MSA CENTRAL (PATCH RECEIVED FROM STEVE KELLEY = LSSG U.K.) 99990100 D 151
$# 99990200 D 151
$# THIS PATCH ALLOWS THE FENCE TO BE MOVED TO THE TOP OF MEMORY, 99990300 D 151
$# SUPPRESSING ALL SWAPPING. ALL JOBS ENTERED ARE TREATED AS BATCH 99990400 D 151
$# JOB = REGARDLESS OF RUN, EXECUTE OR COMPILE. 99990500 D 151
$# 99990600 D 151
$# CANOE PROGRAMS USING TERMINALS MAY RUN WITH THE FENCE AT 32K, 99990700 D 151
$# HOWEVER, THEY WILL NOT BE SWAPPED AND THE JOBS MIGHT BE SCHEDULED 99990800 D 151
$# SINCE ALL JOBS ARE TREATED AS BATCH. 99990900 D 151
$# 99991000 D 151
$# IT IS INTENDED THAT ONLY LIMITED TERMINAL USAGE OCCURS IF THE FENCE 99991100 D 151
$# IS AT THE TOP OF MEMORY, SINCE REMOTE JOBS WILL TIE UP CORE WHEN 99991200 D 151
$# WAITING FOR TERMINAL INPUT OR WHEN STOPPED OR WAITING FOR EXTERNAL 99991300 D 151
$# CONDITION SATISFIED. 99991400 D 151
$# 99991500 D 151
$# THIS PATCH WILL CAUSE THE TSS MCP TO SIMULATE THE 99991600 D 151
$# BATCH, OR DC, MCP WHILE MAINTAINING THE TSS LOGGING 99991700 D 151
$# FACILITIES. THIS PATCH WILL NOT CAUSE REENTRANCY TO 99991800 D 151
$# OCCUR. 99991900 D 151
$#*****99999999 D 151

```

```

$#PATCH NUMBER 152 FOR TSSMCP CONTAINS 5 CARDS. 00000000 D 152

```

```

SPACESTACKSIZE = 85#; 00013870 D 152
SPACESTACKSIZE = 105#; 00013890 D 152
ARRAY SPACESTACK[*]; 00013910 D 152
$VOIDT 23399000 D 152
I := SPACESTACKSIZE; T := P(.SPACESTACK); FIX; % 44173100 D 152
$# DATE 3/1/76 99990000 D 152
$# BY JTC = MSA CENTRAL 99990100 D 152
$# 99990200 D 152
$# THIS PATCH INCREASES THE SPACE STACK SIZE FROM 80 TO 85 WORDS TO 99990300 D 152
$# PREVENT A POSSIBLE STACK OVERFLOW IN THE SPACE STACK. THE OVERFLOW 99990400 D 152
$# OCCURRED WHEN OLAY CALLED DISKSPACE AND DISKSPACE FORKED 99990500 D 152
$# GETMOREOLAYDISK FOR A PROGRAM RUNNING UNDER THE FENCE. OCCASIONALLY, 99990600 D 152
$# WHEN FORK WENT TO GET A FIVE WORD AREA TO BUILD THE INDEPENDENT 99990700 D 152
$# RUNNER ENTRY IN THE FORKQUE THERE WERE NO FIVE WORD AREAS AVAILABLE 99990800 D 152
$# AND IT WAS NECESSARY TO BREAK UP ONE OF THE LARGER AREAS. THIS 99990900 D 152
$# INVOLVED CALLING FORGETAREA AND IT WAS THIS CALL ON FORGETAREA 99991000 D 152
$# THAT OVERFLOWED THE STACK. 99991100 D 152
$#*****99999999 D 152

```

```

$#PATCH NUMBER 153 FOR TSSMCP CONTAINS 43 CARDS. 00000000 D 153

```

```

FIELD MAYRWORKEDON = [7:1]; 21015100 D 153
LABEL START; 21015200 D 153
X := @104; % THIS IS THE CODE WE WILL PASS TO 21015400 D 153
% GETSPACE THE FIRST TIME. IT REQUESTS 21015500 D 153
% OVERLAY MEMORY FOR THE MCP AND THAT 21015600 D 153
% WE WANT TO BE RETURNED TO ON A NO 21015700 D 153
% MEM. 21015800 D 153

```

```

START:
IF NOT M[S].MAYBEWORKEDON THEN % WAIT TILL OTHER GUY IS DOWN 21015900 D 153
SLEEP([M[S]],0 & TRUE [MAYBEWORKEDON]); 21016000 D 153
BEGIN 21017000 D 153
7 := Y.[8:10]; % SIZE OF PROCEDURE IN WORDS 21018100 D 153
M[S].MAYBEWORKEDON := FALSE; 21018200 D 153
% 21018300 D 153
% 21018400 D 153
% NOW WE WILL ATTEMPT TO GET SPACE FOR THIS MCP PROC. 21018500 D 153
% IF WE FAIL WE WILL WAIT FOR A SECOND AND THEN TRY 21018600 D 153
% AGAIN, THIS ENSURES THAT IF WE GET DS=ED WHILE 21018700 D 153
% SLEEPING WAITING FOR MEMORY WE WILL NOT LEAVE THE 21018800 D 153
% TOGGLE LOCKED UP FOR THIS PROCEDURE. 21018900 D 153
% 21019000 D 153
% IF (T:=GETSPACE(Z,65,X)+2) = 2 THEN % NO MEM 21019100 D 153
BEGIN 21019200 D 153
M[S].MAYBEWORKEDON := TRUE; 21019300 D 153
X.[46:1] := TRUE; % DONT PUT OUT MESSAGE 21019400 D 153
SLEEP([CLOCK],NOT CLOCK); % WAIT A SECOND 21019500 D 153
GO TO START; 21019600 D 153
END; 21020000 D 153
M[S] := (*P(DUP)) & T [CTC] & TRUE [MAYBEWORKEDON]; 21024000 D 153
% 24304100 D 153
% FIELDS OF SAVEF PARAMETER 24304110 D 153
% 24304120 D 153
% FIELD 24304130 D 153
% NEEDOLAY = 09:39 % THIS FIELD IS NON ZERO IF 24304140 D 153
% REQUESTOR WANTS OLAY SPACE 24304150 D 153
% IF SAVFF.[45:02] = 3 THEN 24304170 D 153
% JUSTRETURN = 45:01 % GETSPACE WILL TYPE OK MEM 24304180 D 153
% GETFROMFRONT = 47:01 % IF SUCCESSFULL, 24304190 D 153
% 24304200 D 153
% STREAM(X:=(MESS#0 OR SAVEF.[45:2]=3), MIX, SIZE, 24305700 D 153
% MESS:=MESS:=GETARFA(0)); 24305750 D 153
% IF SAVFF.JUSTRETURN THEN P(0,RTN); 24315000 D 153
% IF SAVFF.TYPENOMEMANDRETURN THEN P(0,RTN); 24317000 D 153
% IF MESS#0 OR SAVEF.[45:2] = 3 THEN TELLSPO; 24401000 D 153
% 99990000 D 153
% DATE 3/1/76 99990100 D 153
% BY JTC - MSA CENTRAL 99990200 D 153
% 99990300 D 153
% THIS PATCH WILL PREVENT A PROGRAM FROM BEING DS=ED WHILE 99990400 D 153
% TRYING TO BRING IN AN MCP PROCEDURE. THIS USUALLY LEFT THE TOGGLE 99990500 D 153
% FOR THE PROCEDURE LOCKED UP SO THAT THE PROCEDURE WAS PERMANENTLY 99990600 D 153
% PREVENTED FROM BEING BROUGHT INTO CORE. THIS CONDITION OCCURRED 99990700 D 153
% WHEN THE PROGRAM WAS SLEEPING WAITING FOR A NO MEM CONDITION TO 99990800 D 153
% CLEAR UP SO THAT THE PROCEDURE COULD BE BROUGHT IN AT THE SAME TIME 99990900 D 153
% THAT THE OPERATOR DS=ED THE PROGRAM. IF THE NO MEM CONDITION DID 99991000 D 153
% NOT CLEAR UP IN ABOUT 8 SECONDS, NSECOND WOULD FORCE 99991100 D 153
% THE STACK TO BE DS=ED. NOW, FSPBIT WILL HANDLE THE NO MEM 99991200 D 153
% CONDITION HIMSELF IN SUCH A WAY THAT NO TOGGLES WILL BE LOCKED 99991300 D 153
% SHOULD THE OPERATOR DS THE PROGRAM WHILE IT IS TRYING TO BRING IN 99991400 D 153
% THE PROCEDURE. 99991500 D 153
% 99991600 D 153
% IF A -NO MEM- MESSAGE IS DISPLAYED BY ESPBIT UNDER THE 99991700 D 153
% ABOVE CONDITIONS, THERE WILL NOT BE A CORRESPONDING 99991800 D 153
% "OK MEM" MESSAGE. 99999999 D 153
% *****

```

\$#PATCH NUMBER 154 FOR TSSMCP CONTAINS 30 CARDS. 00000000 D 154

```

%
% IF WE REACH THIS POINT THEN THIS JOB HAS USED 48038120 D 154
% UP ITS TIME SLICE (EITHER PROCESSOR OR I/O) OR 48038130 D 154
% 48038140 D 154
SVOIDT 48038401 48038149 D 154
% THIS JOB HAS BEEN DOING EXCESSIVE OVERLAY AND 48038150 D 154
% DESERVES ANOTHER CHUNK OF MEMORY. IF THE 48038160 D 154
% PROGRAM HAS USED UP ITS TIME SLICE, WE WILL 48038170 D 154
% SWAP IT OUT IF THERE IS SOME OTHER PROGRAM THAT 48038180 D 154
% IS READY TO RUN AND CAN USE THIS GUYS CHUNKS. IF 48038190 D 154
% THE PROGRAM HAS BEEN DOING EXCESSIVE OVERLAY AND 48038200 D 154
% THE PROGRAM HAS ROOM FOR EXPANSION AND DIDNT USE 48038210 D 154
% A MAX CORE CARD, WE WILL SWAP IT OUT 48038220 D 154
% AND SWAPPINGIO WILL ARRANGE TO OBTAIN ANOTHER 48038230 D 154
% CHUNK. 48038240 D 154
% 48038250 D 154
IF P(0,RDS) > FENCE THEN % SWAP JOB 48038260 D 154
BEGIN 48038270 D 154
FOR NT2 := SC[P1MIX] STEP 1 UNTIL LC[P1MIX] DO 48038280 D 154
IF ACTIVE[NT2] > 1 THEN % SOMEONE CAN USE THIS 48038290 D 154
BEGIN % CHUNK. 48038300 D 154
SWAP(TIMEND,1); % 48038310 D 154
GO TO RETURN; 48038320 D 154
END; 48038330 D 154
IF OLAYCTR[P1MIX] < 0 THEN % EXCESSIVE OVERLAY. 48038340 D 154
IF DONTEXPANDBITS[P1MIX] = 0 THEN % EXPANDABLE 48038350 D 154
BEGIN 48038360 D 154
SWAP(TIMEND,1); % 48038370 D 154
GO TO RETURN; 48038380 D 154
END; 48038390 D 154
END; 48038400 D 154
% 99990000 D 154
% 99990100 D 154
% 99990200 D 154
% 99990300 D 154
% 99990400 D 154
% *****99999999 D 154

```

```

% DATE 3/12/76
% BY JTC - MSA CENTRAL
%

```

\$#PATCH NUMBER 155 FOR TSSMCP CONTAINS 261 CARDS. 00000000 D 155

```

BEGIN STRFAM(A1=TYPE DSPACE(10, SPOUTMSGAREAV) ; ) ; % 00005230 D 155
DEFINE 00017005 D 155
TYPE DSPACE(TYPEDSPACE1, TYPEDSPACE2) = 00017010 D 155
(GETSPACE(TYPEDSPACE1, TYPEDSPACE2, 0) + 2) * % 00017015 D 155
ARRAYDESC(ARRAYDESC1, ARRAYDESC2) = 00017020 D 155
([MGETSPACE(ARRAYDESC1, ARRAYDESC2, 0) + 2]) & ARRAYDESC1 [SIZE] * % 00017025 D 155
SAVEARRAYDESC(SAVEARRAYDESC1, SAVEARRAYDESC2) = 00017030 D 155
([MGETSPACE(SAVEARRAYDESC1, SAVEARRAYDESC2, 1) + 2]) 00017035 D 155
& SAVEARRAYDESC1 [SIZE] * % 00017040 D 155
% 00017045 D 155
% *****00017100 D 155

```

```

%*****%00017110 n 155
%*%*00017120 D 155
%* MEMORY AREA TYPES STORED IN 316 FIELD OF FIRST MEMORY %*00017130 D 155
%* LINK OF ALL MEMORY AREAS %*00017140 D 155
%* %*00017150 D 155
%*****%00017160 n 155
%*****%00017170 D 155
%*****%00017180 D 155
%*****%00017190 n 155
%*****%00017200 D 155
%*****%00017210 D 155
%*****%00017220 D 155
%*****%00017230 D 155
%*****%00017240 n 155
%*****%00017250 D 155
%*****%00017260 D 155
%*****%00017270 D 155
%*****%00017280 n 155
%*****%00017290 D 155
%*****%00017300 D 155
%*****%00017310 D 155
%*****%00017320 D 155
%*****%00017330 D 155
%*****%00017340 D 155
%*****%00017350 D 155
%*****%00017360 D 155
%*****%00017370 D 155
%*****%00017380 D 155
%*****%00017390 D 155
%*****%00017400 n 155
%*****%00017410 n 155
%*****%00017420 D 155
%*****%00017430 D 155
%*****%00017440 D 155
%*****%00017450 D 155
%*****%00017460 D 155
%*****%00017470 D 155
%*****%00017480 D 155
%*****%00017490 D 155
%*****%00017500 D 155
%*****%00017510 n 155
%*****%00017520 n 155
%*****%00017530 D 155
%*****%00017540 D 155
%*****%00017550 D 155
%*****%00017560 D 155
%*****%00017562 D 155
%*****%00017564 D 155
%*****%00017566 D 155
%*****%00017568 D 155
%*****%00017570 n 155
%*****%00017572 n 155
%*****%00017574 D 155
%*****%00017576 D 155
%*****%00017578 D 155
%*****%00017580 D 155
%*****%00017582 D 155

```

DEFIN
UNKNOWNAREAV 0# %
.CODFARFAV 1# %
.DATAAREAV 2# %
.IOBUFFERAREAV 3# %
.ALGOLFIBARFAV 4# %
.INQUIRYBUFFARFAV 5# %
.COBOLFIBAREAV 6# %
.TYPE7INTAREAV 7# %
.DISKHEADERAREAV 8# %
.MAINTRUFFAREAV 9# %
.LBLEQNAREAV 10# %
.SFGZEROAREAV 11# %
.STACKAREAV 12# %
.TYPE13INTAREAV 13# %
.SCRATCHDIRAREAV 14# %
.OPSETAREAV 15# %
.DIRTOPAREAV 16# %
.SPOUTMSGAREAV 17# %
.UVROWAREAV 18# %
.JARROWAREAV 19# %
.CIDROWAREAV 20# %
.INQINPUTARFAV 21# %
.INTARRAYARFAV 22# %
.RJFINPUTARFAV 23# %
.DCQUEUEAREAV 24# %
.DALOCROWAREAV 25# %
.SHEETAREAV 26# %
.STAWORDAREAV 27# % DC MCP ONLY
.KEYINBUFFAREAV 28# %
.FSAREAV 29# %
.DC19QUEUEAREAV 30# % DC MCP ONLY
.AVTABLEAREAV 31# %
.TRACFTABLEAREAV 32# % DC MCP ONLY
.SFGDICTAREAV 33# %
.STACKPRTAREAV 34# %
.MCPTABLEAREAV 35# %
.IRSTACKAREAV 36# %
.FPBAREAV 37# %
.CONTROLCARDAREAV 38# %
.LABELAREAV 39# %
.MDUMPARAREAV 40# %
.ESPDISKAREAV 41# %
.LOGARFAV 42# %
.CANDEFINPUTARFAV 43# % TSS MCP ONLY
.ORJOBINPUTARFAV 44# % TSS MCP ONLY
.TYPE45 45# %
.TYPE46 46# %
.TYPE47 47# %
.TYPE48 48# %

```

;
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX00017600 D 155
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00017700 D 155
*****XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00017710 D 155
%*
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00017720 D 155
%*
MEMORY LINKS
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00017730 D 155
%*
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00017740 D 155
*****XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00017750 D 155
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX00017760 D 155
%
00017762 D 155
%
FIELDS OF MEMORY LINK 0 OF ALL AREAS
00017764 D 155
%
00017766 D 155
%
FIELD
00017770 D 155
ARFAAVAILF = 01:01 % = 0 FOR IN-USE AREA, = 1 FOR AVAIL. AREA 00017780 D 155
AREASAVEF = 02:01 % = 1 FOR IN-USE SAVE AREA, = 0 FOR OLAY AREA 00017790 D 155
ARFATYPEF = 03:06 % TYPE OF AREA (SEE ABOVE) 00017800 D 155
ARFAMIXF = 09:06 % MIX INDFX OF OWNER OF AREA 00017810 D 155
ARFABACKLINKF = 18:15 % ADDRESS OF PREVIOUS AREA 00017820 D 155
AREAFWDLINKF = 33:15 % ADDRESS OF NEXT AREA 00017830 D 155
;
00017840 D 155
$ VOID 00068001 00059000 D 155
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX00060000 D 155
*****XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00060010 D 155
%*
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00060020 D 155
%*
MISCELLANEOUS FIELD DEFINITIONS
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00060030 D 155
%*
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00060040 D 155
*****XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*00060050 D 155
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX00060060 D 155
%
00060070 D 155
%
00060080 D 155
%
FIELD
FF = 18:15 % 00060090 D 155
CF = 33:15 % 00060100 D 155
CTF = FF % 00060110 D 155
CTC = CF % 00060120 D 155
MSFF = 16:01 % 00060130 D 155
;
00060999 D 155
%
%
FIELDS OF AIT ENTRY
00061000 D 155
%
00061010 D 155
%
00061020 D 155
%
00061030 D 155
%
FIELD
FILEBIT = 01:01 % 00061040 D 155
OWNBIT = 02:01 % 00061045 D 155
DIMENSIONS = 03:05 % 00061050 D 155
BLKCNTN = 08:10 % 00061060 D 155
MOM = 18:15 % 00061070 D 155
;
00061999 D 155
%
%
FIELDS OF DATA DESCRIPTOR
00062000 D 155
%
%
00062010 D 155
%
00062020 D 155
%
FIELD
00062030 D 155
%
FLAGBITF = 00:00 % 00062040 D 155
%
DATABITF = 01:01 % ALWAYS OFF FOR A DATA DESCRIPTOR 00062050 D 155
%
PBIT = 02:01 % ON IF DESCRIPTOR POINTS TO AREA OF CORE 00062060 D 155
%
SIZE = 08:10 % SIZE OF ARRAY ROW IF ARRAY DESC. 00062070 D 155
%
% 0 FOR INDEXED DATA DESC. OR NAME DESC. 00062080 D 155
%
MOMADDRESSF = 18:15 % ADDRESS OF MOTHER DESCRIPTOR. 00062090 D 155
%
ADDRESSF = 33:15 % IF PBIT IS ON THEN THIS FIELD CONTAINS AN 00062100 D 155
%
% ACTUAL CORE ADDRESS. IF THE PBIT IS OFF THEN 00062110 D 155

```



```

% IF THE VALUE OF THIS FIELD IS GREATER THAN 00062120 D 155
% OR EQUAL TO 512 THEN THE FIELD CONTAINS A 00062130 D 155
% DALLOC ADDRESS WHICH CAN BE USED TO LOCATE THE 00062140 D 155
% DATA IN THE OVERLAY DISK AREA ASSIGNED TO THE 00062150 D 155
% THE PROGRAM. IF THE VALUE OF THE FIELD IS LESS 00062160 D 155
% THAN 512 THEN THIS FIELD CONTAINS A CODE 00062170 D 155
% INDICATING THE STATUS OF THE AREA. 00062180 D 155
% 0 NEVER ACCESSED OVERLAY AREA. 00062190 D 155
% 1 NEVER ACCESSED SAVE AREA. 00062200 D 155
% 2 NEVER ACCESSED OVERLAY AREA WHICH 00062210 D 155
% WHICH IS ASSIGNED TO AUXMEM. 00062220 D 155
% 5 INDICATES OVERLAY IS CURRENTLY IN 00062230 D 155
% PROCESS FOR THIS AREA 00062240 D 155
% 6 INDICATES OLAY HAD IRRECOVERABLE 00062250 D 155
% ERROR WHEN OVERLAYING THIS AREA. 00062260 D 155
% THE NEXT ACCESS TO THE AREA WILL 00062270 D 155
% CAUSE THE PROGRAM TO BE TERMI= 00062280 D 155
% NATFD. 00062290 D 155
% 00062999 D 155
% 00067000 D 155
% MISCELLANEOUS DEFINES 00067010 D 155
% 00067020 D 155
% 00067030 D 155
% 00067040 D 155
% 00067050 D 155
% 00067060 D 155
% 00067070 D 155
% 00067080 D 155
% 00067099 D 155
% 02134300 D 155
% 02134710 D 155
% 02205475 D 155
% S I= IF M(T),AREATYPFF = CODEAREAV THEN 02216000 D 155
% M(T+1),[CF] ELSE 0;% 02216010 D 155
% ELSE P(M(T=2),AREATYPFF,DUP) # TYPE7INTAREAV AND 02216600 D 155
% P(XCH) # TYPE13INTAREAV) OR L # 0; 02216610 D 155
% TP I= M(TA)=TYPEDSPACE(513,MDUMPAREAV))] 02434430 D 155
% & 513[SIZE] & 5(21:45:3));% 02434435 D 155
% LBL I= TYPEDSPACE(10,LABELAREAV));% 02639000 D 155
% INPUTANK[STA],[FF] I= R I= (IF T1 GTR 0 THEN T1 ELSE 03618800 D 155
% TYPEDSPACE(32,OBJOBNPUTAREAV))+2; 03618810 D 155
% LASTSFG[0],[FF] I= (S I= TYPEDSPACE 03638500 D 155
% (32,CANDFINPUTAREAV))+2; 03638600 D 155
% 05804200 D 155
% S VOIDT 05811460 D 155
% P(P1MIX)) P1MIX I= 0; 05811470 D 155
% LI=GETSPACE(A,SCRATCHDIRAREAV,SCRATCHSAVE); 07086000 D 155
% INBUFF I= GETSPACE(11,IOBUFFERAREAV,1)+2;% 07087000 D 155
% FIRSTCARD I= GETSPACE(10,CONTROLCARDAREAV,1)+2;% 07095000 D 155
% ELSE BEGIN OUTBUFFOLD I= OUTBUFF I= 07095100 D 155
% GETSPACE(60,IOBUFFFRAREAV,1)+2;% 07097000 D 155
% H I= SAVEARRAYDESC(30,DISKHFADERAREAV));% 07256400 D 155
% SHEAT I= M(F I= TYPEDSPACE(31,SHEETAREAV+64))] & 30[SIZE];% 07257400 D 155
% SEGO I= TYPEDSPACE(30,SEGZEROAREAV+64));% 07433000 D 155
% H I= CIDROW[R] I= [M(S I= 07433100 D 155
% GETSPACE(94,CIDROWAREAV+64,1)+2]] & 94 SIZE; 07446000 D 155
% TI=GETSPACE(13,CONTROLCARDAREAV+64,0)+4; 08274250 D 155
% SHEAT + [M(F I= TYPEDSPACE(31,SHEETAREAV+64))] & 30[SIZE];

```

```

SFG0 := TYPEDSPACE(30,SEGZEROAREAV+64);%
C := TYPEDSPACE(5,MAINTRUFFAREAV);%
REGIN R := TYPEDSPACE(5,MAINTRUFFAREAV);%
S := [M[TYPEDSPACE(31,SHFFETAREAV)]] & 30[SIZE];%
DLNK := [M[TYPEDSPACE(31,SHFFETAREAV)]] & 30[SIZE];%
INTRNSC := [M[INTLOC := GETSPACE(MAXINT+T17SIZE,INTARRAYAREAV,1)+2]] &
B := (GETSPACE(91,IOBUFFERAREAV,1)+2) & 90[SIZE];%
LOGINFO := SAVEARRAYDESC(20,LOGAREAV);%
NT1 := GETSPACE(5,MAINTBUFFAREAV+64,0)+2;
T := GETSPACE(12,CONTROLCARDAREAV+64,0)+4;%
SEG := IF M[R].AREATYPEF = CODEAREAV THEN M[R+1].[CF] ELSE 0;
ELSE P(M[R=2].AREATYPEF,DUP) # TYPE7INTAREAV AND P(XCH) #
TYPE13INTAREAV) OR LOCN = 0;
BUFF := TYPEDSPACE(60,KFYINBUFFAREAV) + 1;
FH := [M[F := TYPEDSPACE(TEMP,DISKHEADERAREAV,1) & TEMP[SIZE]]];%
FH := [M[F := TYPEDSPACE(30,DISKHEADERAREAV,1) & 30 SIZE];%
STRFAM(A,B,T := T := GETSPACE(10,CONTROLCARDAREAV+64,0)+4);
T := GETSPACE(1+5,CONTROLCARDAREAV+64,0)+4;%
S := [M[TYPEDSPACE(31,SHEETAREAV)]] & 30[8:38:10];%
HDR := GETSPACE(30,DISKHEADERAREAV,1)+2;
FPB := TYPEDSPACE(SFG0[5] INX 1,FPBAREAV);
FR := GETSPACE(SEG0[7].[CF]*ETRLNG,FPBAREAV,1)+2;
LRL := ARRAYDESC(30,LRLQNAREAV);%
LBL := ARRAYDESC(30,LRLQNAREAV);%
SEG0 := ARRAYDESC(30,SEGZEROAREAV);%
SINFO[MIX].[CF] := (STACKLOC :=
GETSPACE(SEG0[3] INX S[21] INX 64,STACKPRTAREAV,1) + 2;
TRP[4] := [M[T := GETSPACE(SEG0[1].[CF],SFG0[CT]AREAV,1)+2]];
STRFAM(D := DALOCROW[MIX]) := SAVEARRAYDESC(DALOCSIZE,DALOCROWAREAV));%
$ VOID
STREAM(D := UVSPACE := TYPEDSPACE(UVSIZE,UVROWAREAV));
M[HDR=2] := (*P(DUP)) & JARROWAREAV AREATYPEF);%
STRFAM(Q := FSROW[MIX]) := SAVEARRAYDESC(4,FSAREAV);%
TRP[3] := [M[GETSPACE(1,FPBAREAV,1)+2]] ELSE %
BEGIN CARD.[33:15] := TYPEDSPACE(13,CONTROLCARDAREAV)+2;
I := GETSPACE(30,JARROWAREAV&MIX[CTF],1)+2;
I := GETSPACE(UVSIZE,UVROWAREAV&MIX[CTF],1)+2;
SHEAT := [M[F := TYPEDSPACE(31,SHEETAREAV+64)]] & 30[8:38:10];%
SEG0 := TYPEDSPACE(30,SEGZEROAREAV+64);%
IF (MIX := LINK.AREAMIXF) = 0 THEN GO TO MCP;
IF P(LINK.AREATYPEF,DUP) = CODEAREAV OR P(XCH) =
TYPEDSPACE(13,CONTROLCARDAREAV) THEN GO TO CODE;
IF LINK.AREATYPEF = TYPE13INTAREAV THEN % TYPE 13 INTRINSIC
IF P(LINK.AREATYPEF,DUP) = TYPE7INTAREAV OR P(XCH) =
TYPEDSPACE(13,CONTROLCARDAREAV) THEN GO TO INTRINSIC;
IF LINK.AREATYPEF = DATAAREAV THEN % OVERLAYABLE MCP DATA
M[GETSPACE := T] := (*P(DUP))&TYPE[AREATYPEF]&MIX[AREAMIXF];
BEGIN TMP := GETSPACE(FPBPTR+ETRLNG,FPBAREAV,1)+2;%
H := [M[TYPEDSPACE(42,DISKHEADERAREAV)]] & 30[8:38:10];%
T := GETSPACE(30,DISKHEADERAREAV,1)+2;%
TMP := GETSPACE(12,CONTROLCARDAREAV+64,5)+4;%
LAB := ARRAYDESC(15,LABELAREAV);%
FA := TYPEDSPACE(TMP := TEMP+(TEMP DIV 2)+2,DATAAREAV)+1;%
CCA := SAVEARRAYDESC(30,ESPDISKAREAV);%
FA := TYPEDSPACE((BUMPFA := FASZ := IF MAX > 128 THEN 128 ELSE MAX)+
(FASZ DIV 2)+2,DATAAREAV)+1;%
S := ARRAYDESC(30,ESPDISKAREAV);%

```

```

08275500 D 155
08385100 D 155
08410000 D 155
08881000 D 155
08930000 D 155
09506000 D 155
12672500 D 155
12848500 D 155
14411820 D 155
14533000 D 155
16853000 D 155
16854200 D 155
16854300 D 155
16954200 D 155
18285200 D 155
18286000 D 155
18438120 D 155
18520300 D 155
20026600 D 155
20028700 D 155
20049000 D 155
20049800 D 155
20059100 D 155
20089800 D 155
20110400 D 155
20116100 D 155
20116200 D 155
20124900 D 155
20127300 D 155
20127400 D 155
20156800 D 155
20169700 D 155
20183610 D 155
20187400 D 155
20600950 D 155
21304000 D 155
21308000 D 155
22069050 D 155
22069100 D 155
22297000 D 155
22303000 D 155
22303100 D 155
22356100 D 155
22386000 D 155
22386050 D 155
22386100 D 155
24400000 D 155
28050800 D 155
28070000 D 155
28288000 D 155
28435000 D 155
28451800 D 155
28452600 D 155
28455400 D 155
28456800 D 155
28457000 D 155
28855000 D 155

```

```

BEGIN M[ALPHA+1]+(CURLOC+GETSPACE(BSIZE+4,IOBUFFERAREAV,1)+2)+2; 37406000 D 155
M[T1:=ALPHA-2]:=M OR (GETSPACE((T1:=M[T1],SIZE)+4,
      LABELAREAV,1)+4) & T1[SIZE] & CNTRLBITS[FTF];38107100 D 155
      BEGIN NT3 := TYPEDSPACE(13,LABELAREAV)+2;% 38107300 D 155
M[T1:=ALPHA-2]:=M OR (GETSPACE((T1:=M[T1],SIZE)+4,
      LABELAREAV,1)+4) & T1[SIZE] & CNTRLBITS[FTF];38205100 D 155
      SFG0:=[M[TYPEDSPACE(62,SEGZFR0AREAV)]]&30[SIZE];38434000 D 155
IF COBOL>0 OR FIB[4],[7:1] THEN 39304000 D 155
  M[FIB INX NOT 1],AREATYPEF := COBOLFIRAREAV 39304500 D 155
ELSE 39304600 D 155
  M[ALPHA=7],AREATYPEF := ALGOLFIRAREAV; 39305000 D 155
  KEY := GETSPACE(5,MAINTBUFFAREAV+64,0)+2; 41312700 D 155
  KEY := GETSPACE(5,MAINTBUFFAREAV+64,0)+2; 41314000 D 155
  T1 := TYPEDSPACE(15,MAINTBUFFAREAV);% 41323130 D 155
  MLA := [M[A:=GETSPACE(33,MAINTRUFFAREAV,0)+3]] & 32[SIZE];% 41328600 D 155
  I:=(T:=GETSPACE(J+2,INQUIRYBUFFAREAV,0)+1)+3 AND NOT 3; 44041000 D 155
  M[C:=J]:=TYPEDSPACE(1,MCPTABLFARFAV)+T1:=0; 44153500 D 155
  J:=GETSPACE(Z,AVTABLEAREAV,1)+2; 44240840 D 155
SCRATCHVEC:= [M[GETSPACE(10,SCRATCHDIRAREAV,SCRATCHSAVE)+2]] 44241180 D 155
  & 10 [TOSIZE]; 44241185 D 155
  BEGIN 48052000 D 155
    P(NT4,BLOB,NT1,GETSPACE(NT1,STACKAREAV,3));48052100 D 155
$ BY JTC - MSA CENTRAL 99990000 D 155
$ DATE 03/17/76, UPDATF 11/05/76 99990100 D 155
$ THIS PATCH CAUSES THE TSSMCP TO TYPE MANY OF THE MEMORY AREAS 99990200 D 155
$ THAT WERE LEFT UNTYPED BEFORE. THIS PRIMARILY BENEFITS THE DUMP 99990300 D 155
$ ANALYZER. 99990400 D 155
$ *****99999999 n 155

```

```

$#PATCH NUMBER 156 FOR TSSMCP MARK XVI CONTAINS 5 CARDS. D 156
IF HEADER.[CF] GEQ 64 THEN FORGETSPACE(HEADER); 12561600 D 156
$VOIDT 12572000 D 156
$VOIDT 12573000 D 156
      P(1) 12573500 D 156
      ELSE 12574000 D 156
$ BY JTC - MSA CENTRAL D 156
$ SOFTWARE FLASH #164 AND CORRECTION FLASH #190 D 156
$ ***** n 156

```

```

$#PATCH NUMBER 158 FOR TSSMCP CONTAINS 18 CARDS. 00000000 D 158
ARRAY WORKERSTACK[*]; % DESC. TO OLDWIERDHAROLD'S STACK, 00299500 D 158
ARRAY ESPTAB[*]; REAL ESPCOUNT; % 00399000 D 158
      GETESPDISK := ((P(DUP),[CF])=FSPTAR.[CF])x8 06031000 D 158
      S := (FSPTAR.[CF]+S.[30:15]) & S[30:13]; % 06038000 D 158
DEFINE HARRYSTA = WORKERSTACK[5];% STA 14344100 D 158
      I := W; T := P(.ESPTAB); FIX; % 44173200 D 158
      I := 100; T := P(.WORKERSTACK); FIX; % 44179100 D 158
      I := 0; 44179200 D 158
      FOR T := 1 STEP 1 UNTIL NUMSTACK DO 44179250 D 158

```

```

      BEGIN
      M[J] := I; % LINK TO PREVIOUS STACK IN Q.
      M[J+1] := WORDOFFEASE; % CLEAR STACK
      MOVE(STANDARDSTACK+2,J+1,J+2);
      I := J; % REMEMBER WHERE THIS STACK IS
      J := J + STANDARDSTACK; % COMPUTE ADDRESS OF NEXT STACK
      END;
      STACKQ := I; % POINT TO FIRST STACK IN Q.
      MOVE(99,I:=WORKERSTACK.[CF],I+1); %TO SPOT POSSIBLE OVER
$! DATE 3/19/76
$! BY JTC - MSA CENTRAL
$!
$! THIS PATCH CAUSES DESCRIPTORS TO BE BUILT FOR THE WORKERSTACK AND
$! ESPTAB TO HELP THE DUMP ANALYZER ANALYZE THE MCP SAVE AREAS.
$!*****99999999

```

```

$#PATCH NUMBER 159 FOR TSSMCP CONTAINS 25 CARDS

```

```

      REAL PROCEDURE INPUTSCAN(MODE,SOURCE,DEST,NUM,FLAGS,SEQMODE);
      VALUE NUM,FLAGS,SEQMODE;
      REAL MODE,SOURCE,DEST,NUM,FLAGS,SEQMODE;
      REAL PROCEDURE INPUTSCAN (MODE,SOURCE,DEST,NUM,FLAGS,SEQMODE);
      VALUE NUM,FLAGS,SEQMODE;
      REAL MODE,SOURCE,DEST,NUM,FLAGS,SEQMODE;
      U:=(DEST.[30:3]&DEST[30:33:15])-(INPTSC.[30:3]&INPTSC[30:33:15]);
      IF U=1 AND SEQMODE %U=1 WHEN ONLY "+" LEFT IN BUF., AND IF SEQMODE
      THEN %TRUE, THEN USER BACKSPACED TO FRONT OF BUFFER.
      BEGIN
      STREAM (A:=0:DEST);
      BEGIN DI:=DI-1; %MAKE INPUT LOOK LIKE A NULL LINE
      DS:=2LIT" +"; % (SPACE) TO PREVENT CANDE FROM
      A:=DI; %BELIEVING THIS IS END OF SEQ MODE
      END;
      P([DEST],STD);
      GO AUT;
      END;
      INPUTSCAN:=U;
      N1:=INPUTSCAN(MODE,SOURCE,A3,(M[A1].[FF]=X,[218]),FLAGS,0);
      N1:=INPUTSCAN(MODE,SOURCE,A3,M[A1].[FF],FLAGS,0);
      DEFINE SEQMODE = IF STATAB.STATIONTYPE#TWX THEN FALSE
      ELSE (LINEDISC[STA]=TWX AND SEQARRAY[STA]#0)#;
      NUM:=NUM+INPUTSCAN(MODE,SOURCE,J,BUFSZ=T1,FLAGS,SEQMODE);
      NUM:=NUM+INPUTSCAN(MODE,SOURCE,J,BUFSZ,FLAGS,SEQMODE);
$! BY DJZ=MSA CENTRAL
$! SOFTWARE FLASH #175 AND CORRECTION FLASH #185
$! SOFTWARE FLASH 200 CORRECTION
$!*****

```

```

$#PATCH NUMBER 160 FOR TSSMCP CONTAINS 12 CARDS.

```

```

      BEGIN

```

```

IF P(0,RDS) GTR FENCE                                06470000 D 160
SVOIDT                                                06471000 D 160
  FLSF COMPLXSLFFP((R OR RFLY[P1MIX] GTR 0 OR TERMSET(P1MIX))); 06472100 D 160
SVOIDT 06472501                                       06472200 D 160
  IF JAR[P1MIX,9].SYSJOB# NEQ LIBMAINCODE THEN GO INITIATE; 06473100 D 160
  EXIT; P(P&RCW[CTC],0,RDS,0,XCH,P&P[CTF],STF);          28101200 D 160
SVOIDT                                                37180675 D 160
  IF [MEM[P1MIX,MLINK1]].[CF] GEQ FFNCF THEN             37226100 D 160
SVOIDT                                                37226150 D 160
SVOIDT                                                37226500 D 160
SVOIDT                                                37227500 D 160
S: BY JTC, KFK = MSA CENTRAL, SOFTWARE FLASH 176      D 160
S:*****                                              D 160

```

```

S#PATCH NUMBER 161 FOR TSSMCP CONTAINS 1 CARD        D 161

ROMB: IF OU=18 THEN FORGETIT;%                        07231000 D 161
S: BY DJZ = MSA CENTRAL, SOFTWARE FLASH 177           D 161
S:*****                                              D 161

```

```

S#PATCH NUMBER 162 FOR TSSMCP CONTAINS 3 CARDS.     D 162

IF (NT2=P).[2:1] THEN %LEFT AT 12626500             12635500 D 162
SVOIDT                                                12638000 D 162
  BEGIN IF (IOD:=[T.[9:24] DIV 5]#0 THEN %SPACE FORWARD 12650500 D 162
S: BY DJZ = MSA CENTRAL, SOFTWARE FLASH 180           D 162
S:*****                                              D 162

```

```

S#PATCH NUMBER 163 FOR TSSMCP CONTAINS 3 CARDS.     D 163

S SET OMIT = PACKETS                                  20034060 D 163
  FLSF FORGETSPACE(HDR);                              20034070 D 163
S POP OMIT                                           20034080 D 163
S: BY DJZ = MSA CENTRAL, SOFTWARE FLASH 183           D 163
S:*****                                              D 163

```

```

S#PATCH NUMBER 164 FOR TSSMCP CONTAINS 16 CARDS.   00000000 D 164

REAL FR1,FR2,ER3; LABEL FRSYS;                       20511152 D 164
  ER1=")+ "%;                                         20511175 D 164
  OR SYSTEMFILE(CMM[0],CMM[1]) THEN%                 20511188 D 164
  BEGIN FRSYS: ER1="SYSTEM "%;ER2="FILE)+ "%; GO ERROR END; 20511190 D 164
  BEGIN ER1="SAME FI";ER2="LE)+ "%; GO ERROR END       20511303 D 164
  IF TYPE=USEV AND M[OPTN+2]<0 THEN%                 20511312 D 164
  BEGIN ER1="SECURIT";ER2="Y FILE)";ER3=")+ "%; GO ERR END;20511313 D 164

```

```

      BEGIN FR1:="INVALID";ER2:=" USER)+";%
FRR:   FORGETSPACE(OPTN);%
      END ELSE IF OPTN=2 THEN GO ERSYS% @ LINE 20511190
      ELSE IF OPTN=1 THEN BEGIN ER1:="IN USE");ER2:="+
      ELSE IF OPTN=0 THEN BEGIN ER1:="NOT ON ";ER2:="DISK)+ " END;
FRROR:
      STRFAM(AI=[CMM],FR1=[FR1],B1=(OPTN=SPACE(10)));
      DS1=25 LIT " SECURITY MAINT IGNORED ("; SI1=FR1%;
      3(SI1=SI1+1; DS1=7 CHR);%
$! BY DJZ - MSA CENTRAL (SUBMITTED BY S. KELLEY - NATAL UNIVERSITY)
$! DATE 05/12/76
$! THIS PATCH MAKES THE "SECURITY MAINT IGNORED" MESSAGE MORE
$! EXPLICIT BY ADDING ONE OF THE FOLLOWING REASONS TO THE MESSAGE:
$! 1) SYSTEM FILE
$! 2) SECURITY FILE
$! 3) INVALID USER
$! 4) IN USE
$! 5) NOT ON DISK
$! AND 6) SAME FILE.
$!*****99999999

```

```

$#PATCH NUMBER 165 FOR TSSMCP CONTAINS 1 CARD.

```

```

      AND (MFID EQV "BADISK ") NEQ NOT 0)
$! BY DJZ - MSA CENTRAL (SUBMITTED BY S. KELLEY - NATAL UNIVERSITY)
$! DATE 05/13/76
$! THIS PATCH WILL PREVENT "BADISK" FILES FROM BEING DUMPED TO A
$! LIBRARY TAPE. PREVIOUSLY, IF THESE FILES WERE RE-LOADED, THEY
$! WERE ALLOCATED AS NORMAL FILES, WITH A VERY SMALL CHANCE OF
$! BEING PLACED IN THE PROPER SPACE. FOR DUMPING PURPOSES ONLY,
$! ALL "BADISK" FILES WILL NOW BE TREATED AS SYSTEM FILES.
$! NOTE THAT SITES SHOULD NOT ALLOW USER FILES TO BE CREATED WITH
$! "BADISK" <MFID>S, AS THESE ALSO WILL NOT BE DUMPED.
$!*****99999999

```

```

$#PATCH NUMBER 166 FOR TSSMCP CONTAINS 18 CARDS.

```

```

% .[13:1] FILE TO BE ADDED (NOT ON DISK)
MUSTADD = DA.[13:1]#
%*****
% CHECK IF FILE IS TO BE ADDED
%*****
%
      IF ADDV THEN
      IF NOT MUSTADD THEN
      IF OU = 18 THEN% TAPE TO DISK
      IF DIRECTORYSEARCH(=MFID,FID,5) NEQ 0% ALREADY ON DISK
      THEN BEGIN
      LBMESS(ABS(MFID),FID,=67,68,TINU[IU],SPOUTUNIT,LIBMSG);
      IF DSED THEN ABORT;
      GO NEXT;

```

```

FND
ELSE BEGIN MIFAINFO+JJ.[13:1]#=1; MUSTADD=1 END;
IF MUSTADD THEN T=0 ELSE% ADD FILE NOT ON DISK
"ON DISK", " " , % 68
$! BY DJZ - MSA CENTRAL (SUBMITTED BY S. KELLEY - NATAL UNIVERSITY)
$! DATE 05/13/76
$! THIS PATCH WILL PREVENT THE MCP FROM SCANNING DOWN A LIBRARY
$! DUMP TAPE TO FIND A FILE WHEN THE FILE BEING ADDED IS ALREADY
$! ON DISK. A NEW MESSAGE IS ALSO INTRODUCED. THIS WILL BE SENT
$! TO THE PACKET PAGE AND ALSO TO THE SPO IF LIRMSG IS SET. THE
$! FORMAT OF THE NEW MESSAGE IS:
$!      .<MFID>/<FID> NOT COPIED (ON DISK), MT<UNIT>
$!*****99999999

```

```

$#PATCH NUMBER 167 FOR TSSMCP CONTAINS 1 CARD.
IF KIND=7 AND FIB[13].[28:10]#ABS(COBOL) THEN GO FIND ELSE
$! DATE 6/23/76
$! BY JTC - MSA CENTRAL
$!
$! THIS PATCH FIXES OTHERFILEOPENOUT SO THAT COROL68 PRINTER
$! BACK UP TAPES WILL HAVE CORRECTLY FORMATTED PRINT
$! FILE LABELS WRITTEN ON THEM.
$!*****99990600

```

```

$#PATCH NUMBER 168 FOR TSSMCP CONTAINS 11 CARDS. TAPE PARITY
LABEL ERRORS,DISKERR,DS,X,SW,LP,DK,DX,DX1,DC,OK,L1;
SWITCH TYPE := OK,LP,OK,OK,DK,OK,OK,OK,OK;
%
% CHECK FOR A PARTIAL WORD BINARY READ WITH NO PARITY ERRORS.
% THIS IS ILLEGAL AND IS MARKED AS BEING A PARITY ERROR.
%
IF (R.[18:12] AND @4462) = @0440 THEN % RIN READ=NO PAR
IF R.[15:31] # ((8-R.[22:1]) AND 7) THEN % PART WD XFER
R.[28:1] := MOD310S; % MARK AS PARITY ERROR IF MOD III
SVOIDT 04167801
IF U LFO 15 THEN % TAPE I/O
$! DATE 9/14/76
$! BY JTC - MSA CENTRAL
$!
$! THIS PATCH REWRITES THE CODE FOR HANDLING PARTIAL WORD TRANSFER
$! ON BINARY READS FROM TAPE. SINCE THIS IS ILLEGAL FOR THE B5700
$! SUCH I/O RESULTS ARE MARKED AS PARITY ERRORS. PREVIOUS TO THIS PATCH
$! SPACE REVERSE I/O S LOOKED LIKE PARTIAL WORD TRANSFERS AND THIS
$! CAUSED ONLINE/MAINT TAPE CONFIDENCE ROUTINES 1,2, AND 3 TO FAIL.
$! NOTE: SITES THAT READ NON-STANDARD TAPES THAT HAVE BINARY
$! PARITY AND PARTIAL WORD BLOCKS MAY WANT TO ELIMINATE THIS PATCH
$! SINCE IT WILL CAUSE RETRIES TO BE DONE ON SUCH TAPES.
$!*****99991100

```

```

$#PATCH NUMBER 169 FOR TSSMCP CONTAINS 4 CARDS.                                D 169
      PPCPROCESS:=0; 20602855 D 169
      IF UNITNO>32 THEN CIDROW[UNITNO-32].[13:5]:=0 ELSE% 20606900 D 169
      IF UNITNO=23 THEN READERR.[FF]:=0 ELSE% 20606910 D 169
      IF UNITNO=24 THEN READERR.[FF]:=0;% 20606920 D 169
$! BY DJZ = MSA CENTRAL D 169
$! SOFTWARE FLASH 201 D 169
$!***** D 169

```

```

$#PATCH NUMBER 170 FOR TSSMCP CONTAINS 1 CARDS.                                00000000 D 170
      BEGIN IF XLST=0 THEN XLST:=FM[SPACE(XLSTSZ:=30)1]&30[8:38:10]; 20567005 D 170
$! DATE 9/17/76 99990000 D 170
$! BY JTC = MSA CENTRAL 99990100 D 170
$! 99990200 D 170
$! THIS PATCH PREVENTS EXTRA EXCEPTION LISTS FROM BUILDING UP IN 99990300 D 170
$! CORE WHEN PROCESSING REMOVE OR COPY CARDS WITH MULTIPLE 99990400 D 170
$! EXCEPTION LISTS. 99990500 D 170
$!*****99990600 D 170

```

```

$#PATCH NUMBER 172 FOR TSSMCP CONTAINS 9 CARDS.                                00000000 D 172
      ANFCK = 0; FIRSTORSEC = 07108100 D 172
$ 07108999 D 172
$ 07109001 D 172
      BEGIN LABEL TRYAGAIN; 20289030 D 172
$ 20289330 D 172
      XI=6; MIBUF+17]:=0; 20289335 D 172
      129: GO TO EXTERNAL; % 35 = SPECIAL INTERRUPT 46011500 D 172
      130: GO TO EXTERNAL; % 36 = DKA READ CHECK 46012000 D 172
      131: GO TO EXTERNAL; % 37 = DKB READ CHECK 46012500 D 172
$! BY DJZ = MSC DETROIT 99990000 D 172
$! DATE 11/29/76 99990100 D 172
$! THIS PATCH ... 99990200 D 172
$! 1) REMOVES SOME REDUNDANT CODE FROM PROCEDURE "COM23"; 99990300 D 172
$! 2) REMOVES UNNECESSARY LABEL IN PROCEDURE "PRINTTHECOVER"; 99990400 D 172
$! 3) PREVENTS SYSTEM HANG IN "DO UNTIL FALSE" LOOP WHEN DISK 99990500 D 172
$! READ CHECK IS DETECTED. THIS WAS ORIGINALLY INTRODUCED IN 99990600 D 172
$! TEMPORARY PATCH #202, MARK XIV SYSTEM RELEASE. 99990700 D 172
$!*****99999999 D 172

```

```

$#PATCH NUMBER 173 FOR TSSMCP CONTAINS 2 CARDS.                                00000000 D 173
% R 5 7 0 0 T S = M C P M A R K X V I . 0 . 1 7 2 05/11/77 00001000 D 173
      "172" 00005040 D 173

```



```

S: BY DJZ = MSC DETROIT                                99990000 D 173
S: DATE 05/11/77                                       99990100 D 173
S: THIS PATCH UPDATES THE MARK LEVELS IN THE SYMBOL FILE. 99990200 D 173
S:*****99990300 D 173

```

```

$#PATCH NUMBER 201 FOR TSSMCP CONTAINS 005 CARDS      00000000 D 201
    IF U=30 THEN                                       04129100 D 201
        BEGIN M[153+C]=R;                               04129200 D 201
            DCIOFINISH(R);                               04129300 D 201
        END;                                           04129400 D 201
    IF (E=(M[153+C]=R).[26:7])+(V=T.[5:8]) NEQ 0 THEN 04130000 D 201
S: THIS CHANGE CAUSES THE RESULT DESCRIPTOR OF EACH I/O TO BE 00000002 D 201
S: STORED INTO M[153+I/O CHANNEL NUMBER].             00000002 D 201
S:                                                     00000002 D 201
S:BY JCF                                               00000002 D 201
S:TR .....                                           00000002 D 201
S:DATE 3/6/73                                          00000002 D 201
S:*****n 201

```

```

$#PATCH NUMBER 202 FOR TSSMCP,XV2 CONTAINS 25 CARDS  D 202
    MEMTOG = TOGGLE,[43:1];MEMTOGMASK =@20#           00080500 D 202
    IF (MEMWAITSTORE[1,0]).[CF]=0 THEN                00093110 D 202
        IF NOT MEMTOG THEN SLEEP((TOGGLE),MEMTOGMASK)ELSE ELSE00093120 D 202
    DEFINE STOREDY(STOREDY1,STOREDY2)= IF(MEM[STOREDY1,0]).[CF]=0 THEN 00093400 D 202
        MEMTOG:=STOREDY2 FLSF                          00093410 D 202
        MEM[STOREDY1,0].[17:1]=STOREDY2#;             00093420 D 202
        STOREDY(0,0);                                  02396130 D 202
        STOREDY(0,1);                                  02396500 D 202
            STOREDY(0,0);                               02434490 D 202
            STOREDY(0,1);                               02434530 D 202
        BEGIN WAITSTORE(0); STOREDY(0,0);             09617000 D 202
            STOREDY(0,1);                               09624000 D 202
        WAITSTORE(P1MIX); STOREDY(P1MIX,0);          18841400 D 202
        STOREDY(P1MIX,1);                              18843500 D 202
    WAITSTORE(P1MIX); STOREDY(P1MIX,0);             18843800 D 202
    STOREDY(P1MIX,1); % FREE MEMORY TO ALLOW "ARTN" TO BE BROUGHT IN 18845000 D 202
        WAITSTORE(MIX); STOREDY(MIX,0);              20119300 D 202
        STOREDY(MIX, 1);                               20120300 D 202
        STOREDY(0,0);                                  21032000 D 202
        STOREDY(0,1);                                  21093000 D 202
        AGAIN; WAITSTORE(MIX); STOREDY(MIX,0);        24309000 D 202
            STOREDY(MIX,1);                             24313000 D 202
        WAITSTORE(P1MIX); STOREDY(P1MIX,0);          29544000 D 202
        STOREDY(P1MIX,1);                              29548000 D 202
    MEMTOG:=1;                                         44114100 D 202
S:THIS PATCH ALLOWS CELL ZERO TO BE MONITORED FOR INVALID WRITES 90000004 D 202
S:(AFTER INITIALIZE). IN ORDER TO ENABLE THIS, A NEW TOGGLE IS 90000005 D 202
S: USED , MEMTOG, RATHER THAN M[0].[17:1], TO INTERLOCK MEMORY 90000010 D 202
S: BELOW THE FENCE. WITH THIS PATCH REFERENCE TO STOREDY 90000015 D 202

```

\$: MUST BE MADE AS "STORED Y(MIX, TOG);", WHERE MIX IS THE MIX INDEX	90000020	D	202
\$: OF THE PROCESS AND TOG IS EITHER 0 OR 1.	90000025	D	202
\$:	00000002	D	202
\$: BY AMS	00000002	D	202
\$: TR 1790	00000002	D	202
\$: DATE 3/6/73	00000002	D	202
\$: *****	00000002	D	202

\$: PATCH NUMBER 203 FOR TSSMCP, XV2 CONTAINS 29 CARDS		D	203
--	--	---	-----

BOOLEAN OKSEGZEROWRITE;	00422100	D	203
IF (P(IODESC.[1315])	04004110	D	203
\$ SET OMIT = DKBNODFX	04004119	D	203
, DUP) = @14 OR P(XCH	04004120	D	203
\$ POP OMIT	04004121	D	203
) = @6) AND	04004130	D	203
NOT IODESC.[2411] AND	04004140	D	203
(((P(MIODESC.[CF]), DUP) EQV 0) = NOT 0) OR	04004150	D	203
((P(XCH) EQV 32) = NOT 0)) AND	04004155	D	203
NOT OKSEGZEROWRITE THEN	04004160	D	203
BYBY("SEGMENT ZERO OVERWRITE", 23);	04004170	D	203
OKSEGZEROWRITE; = TRUE;	05665399	D	203
OKSEGZEROWRITE; = FALSE;	05665401	D	203
OKSEGZEROWRITE; = TRUE;	06059659	D	203
OKSEGZEROWRITE; = FALSE;	06059661	D	203
OKSEGZEROWRITE; = TRUE;	09645999	D	203
OKSEGZEROWRITE; = FALSE;	09656001	D	203
OKSEGZEROWRITE; = TRUE;	09682609	D	203
OKSEGZEROWRITE; = FALSE;	09682611	D	203
OKSEGZEROWRITE; = TRUE;	40249400	D	203
OKSEGZEROWRITE; = FALSE;	40262100	D	203
OKSEGZEROWRITE; = TRUE;	42600919	D	203
OKSEGZEROWRITE; = FALSE;	42600921	D	203
OKSEGZEROWRITE; = TRUE;	42713949	D	203
OKSEGZEROWRITE; = FALSE;	42713951	D	203
OKSEGZEROWRITE; = TRUE;	42730999	D	203
OKSEGZEROWRITE; = FALSE;	42731001	D	203
OKSEGZEROWRITE; = TRUE;	44411399	D	203
OKSEGZEROWRITE; = FALSE;	44411401	D	203
\$: THIS CHANGE IS PROVIDED FOR DEBUGGING PURPOSES AND CAUSES THE MCP	44419421	D	203
\$: TO HANG IN A "DO UNTIL FALSE" LOOP WHENEVER DISK SEGMENT ZERO IS	44419423	D	203
\$: ABOUT TO BE UNEXPECTEDLY OVERWRITTEN.	44419425	D	203
\$:	00000002	D	203
\$: BY JCF	00000002	D	203
\$: TR 1766	00000002	D	203
\$: DATE 3/6/73	00000002	D	203
\$: *****	00000002	D	203

\$: PATCH NUMBER 301 FOR TSSMCP, XV2 CONTAINS 3 CARDS		D	301
---	--	---	-----

IF MODE 1 = (MODE=0) AND BLFN = 20 THEN	38107610	D	301
---	----------	---	-----

```

          SAVWORD := SAVWORD OR TWO(U);
          CNTCTL := DIREC := 0;
$! THIS PATCH ALLOWS THE READING OF PURE BINARY CARD INPUT.
$! IF THE FILE IS DECLARED AS ALPHA WITH A BUFFER LENGTH
$! OF 20 WORDS, ALSO THE CARD READER WILL BE MARKED SAVED
$! WHEN THE PROGRAM CLOSES THE FILE. CARE SHOULD BE TAKEN
$! WHEN USING THIS FEATURE SINCE A "QEND" CARD WILL NOT
$! BE SEEN BY THE SYSTEM. IT IS THE PROGRAMS RESPONSIBILITY
$! TO DETECT WHEN THE END OF THE FILE HAS OCCURRED.
$!
$!BY JCF
$!DATE 3/6/73
$!*****

```

```

38107620 D 301
38107700 D 301
90000015 D 301
90000020 D 301
90000025 D 301
D 301
D 301
D 301
D 301
D 301
D 301
D 301
00000007 D 301
D 301

```

```

$#PATCH NUMBER 302 FOR TSSMCP CONTAINS 049 CARDS

```

```

00000000 D 302

```

```

REAL RESULT = -4;
BOOLEAN ZEROTIMEOUT;
ZFROTIMEOUT:=N.[32:1];
N.[32:1]:=0;
      IF ZFROTIMEOUT THEN
      BEGIN
          RESULT:=2;
          STREAM(D:=A.[CF]);
          DS:=8 LIT "TIME3 ";
          GO TO RETURN;
      END;
STREAM(T:=("END")&[12][24:42:6]):A2);
BEGIN SI:=A2; DI:=LOC T; DI:=DI+4;
      IF 4 SC NEQ DC THEN TALLY:=1;
      T:=TALLY;
END;
RESULT:=POLISH;
GO TO RETURN;
      IF PAPER TAPE THEN
      BEGIN STREAM(T:=("END"),SOURCE);
          BEGIN SI:=LOC T; DI:=SOURCE;
              SI:=SI+5; DI:=DI+1;
              IF 3 SC=DC THEN
              BEGIN DI:=DI-3; DS:=3 LIT " "; END;
          END;
          FLAGS:=4;
      END ELSE
      BEGIN STRFAM(T:=("END"),SOURCE);
          BEGIN SI:=SOURCE; DI:=LOC T;
              SI:=SI+1; DI:=DI+5;
              IF 3 SC NEQ DC THEN TALLY:=1;
              T:=TALLY;
          END;
          IF POLISH THEN GO TO CE;
      END;
      BEGIN STREAM(SOURCE,KIT:=PAPER TAP);
          IF SC=MARK THEN
          BEGIN DI:=LOC K; DS:=3 LIT "END";

```

```

03292300 D 302
03296250 D 302
03296300 D 302
03296350 D 302
03297410 D 302
03297415 D 302
03297420 D 302
03297425 D 302
03297430 D 302
03297435 D 302
03297440 D 302
03307100 D 302
03307110 D 302
03307120 D 302
03307130 D 302
03307140 D 302
03307150 D 302
03317500 D 302
03624400 D 302
03624401 D 302
03624402 D 302
03624403 D 302
03624404 D 302
03624405 D 302
03624406 D 302
03624407 D 302
03624408 D 302
03624410 D 302
03624420 D 302
03624430 D 302
03624440 D 302
03624450 D 302
03624460 D 302
03624470 D 302
03624480 D 302
03624600 D 302
03625300 D 302
03625310 D 302

```

```

          SI:=SI+1; DI:=DI-3;          03625320 D 302
          IF 3 SC=DC THEN              03625330 D 302
          BEGIN TALLY:=2; KI:=SI; DI:=K; 03625331 D 302
              T(DI:=DI-3;             03625332 D 302
                DS:=3 LIT " ");       03625333 D 302
          END ELSE TALLY:=1;          03625334 D 302
          END;                          03625340 D 302
          RI=POLISH;                    03625600 D 302
          FLAGS:=IF R THEN 4 ELSE      03625610 D 302
              IF ((R=2) AND PAPERTAPE) THEN 4 ELSE 03625620 D 302
              IF ((R=2) AND FLAGS) THEN 4 ELSE FLAGS; 03625630 D 302
          $: THIS PATCH IMPLEMENTS THE FOF BRANCH IF "QFN" IS TYPED 00000002 D 302
          $: TO A REMOTE JOB AS INPUT. 00000002 D 302
          $: ALSO THE PARITY LABEL WILL BE TAKEN IF DATA IS NOT PRESENT AND A ZERO 00000002 D 302
          $: TIME OUT WAS SPECIFIED. 00000002 D 302
          $:                                00000002 D 302
          $: BY JCF                          00000002 D 302
          $: TR .....                        00000002 D 302
          $: DATE 3/6/73                      00000002 D 302

```

```

$# PATCH NUMBER 401 FOR TSSMCP.XVI.00 CONTAINS 01 CARDS.FIVE MINUTE SLEEP LIMIT. D 401
          TIMELIMITMAX + 3600 * S + 18000 ; S=0; %401=GRK 22013130 D 401
          $: THIS PATCH INCREASES THE MAXIMUM SLEEP TIME ABOVE THE FENCE FROM D 401
          $: 15 SECONDS TO FIVE MINUTES PLUS ONE MINUTE FOR EACH ACTIVE JOB. D 401
          $: FAR TOO MANY JOBS GET KILLED AT 15 SECS THAN IS REASONABLE. D 401
          $: G.R.KENNEDY 25FEB76. BROCK U, PATCH 401 D 401
          $: ***** D 401

```

```

$# PATCH NUMBER 403 FOR TSSMCP.XVI.00 CONTAINS 01 CARDS.PREVENT FLAG BITS ON CM. D 403
          STREAM(BASE); BEGIN 3(DS=LIT"0"; DI=DI+7); END; %403=GRK 09680450 D 403
          $: THIS PATCH PREVENTS SYSTEM HANGS CAUSED BY GARBAGE IN SEGMENT ZERO D 403
          $: IN PROCEDURE CHANGEMCP. D 403
          $: THE CURRENT MCP NAME(S) AND BASE DISK ADDRESS MAY HAVE THE FLAG BIT D 403
          $: ON (IT HAPPENED AT LEAST ONCE). D 403
          $: G.R.KENNEDY 17FEB76. BROCK U, PATCH 403 D 403
          $: ***** D 403

```

```

$# PATCH NUMBER 407 FOR TSSMCP.XVI.00 CONTAINS 01 CARDS.CONTROLCARD SCREWUP. GRK D 407
          P(DIRECTORYSEARCH(NABS(CMM[0]),IF CMM[0]<0 THEN "DISK " ELSE 20606865 D 407
          $: THIS PATCH FIXES A PROBLEM IN CONTROL CARD WHICH CAUSED THE MCP SAVE D 407

```

```

$: CORF TO GRADUALLY INCREASE, THE PROBLEM WAS WITH CONTROL CARD          D 407
$: ERRORS ON COMPILE CARDS. AT "INCSC" THERE WAS A CALL ON                 D 407
$: DIRECTORYSEARCH WITH -CMMF01 SO AS TO NOT RETURN THE FILE HEADER.      D 407
$: BUT - COMPILERS HAVE CMMF01 NEGATIVE, SO ...                            D 407
$: RECEIVED FROM DREXEL UNIVERSITY BY PHONE ON 16JUL75.                   D 407
$:                                                                           D 407
$: G.R.KENNEDY 10OCT75. BROCK U. PATCH 407                                D 407
$: *****                                                                    n 407

```

```

$# PATCH NUMBER 410 FOR TSSMCP,XVI.00 CONTAINS 02 CARDS,LET "SD" OR "CL" WORK. D 410

      JAR[MIX,61.[1:1]+((TYPE=9) OR (TYPE=19)); %DS=8,SD=9,CL=19 %410 16212100 n 410
***% JAR[MIX,61.[1:1]+((TYPE=9) OR (TYPE=19)); %DS=8,SD=9,CL=19 %410 16216000 D 410
$:                                                                           D 410
$: THIS PATCH TO KEYING SHOULD ALLOW "SD" OR "CL" TO SAVE PSEUDO-DECKS.    D 410
$: WE WILL SETUP THE "SAVE-THE-DECK" BIT BEFORE CALLING TERMINATE, AS     D 410
$: IN THE DCMCP.                                                            D 410
$:                                                                           D 410
$: G.R.KENNEDY 13JUL76. BROCK U. PATCH 410                                D 410
$: *****                                                                    D 410

```

```

$# PATCH NUMBER 504 FOR TSSMCP,XVI.00 CONTAINS 22 CARDS,EOJ JOB STATISTICS,GRK. D 504

      REAL CPUIO=+14; %504=GRK 14351306 D 504
      P(0); %ZERO OUT CPUIO %504=GRK 14355060 D 504
      CPUIO+GETSPACE(10,0,1)+2; %504=GRK 14413020 D 504
      M[CPUIO+2]+VECTOR[0]; %JOB PREFIX %504=GRK 14413030 D 504
      M[CPUIO+3]+VECTOR[1]; %JOB SUFFIX %504=GRK 14413040 D 504
      M[CPUIO+4]+1; %JOB MIX INDFX %504=GRK 14413050 n 504
      M[CPUIO+5]+((S[1]+NEXTMOM)+30)DIV 60; %CPU TIME %504=GRK 14413060 D 504
      M[CPUIO+6]+((S[2]+VECTOR[4]+IOTIME[1])+30)DIV 60; %504=GRK 14413070 D 504
      M[CPUIO+7]+S[3]+(LC[1]-SC[1])xCHUNKZIZE+CHUNKZIZE; %504 14413080 D 504
      STREAM(1,CPUIO); %504=GRK 14430031 D 504
      BEGIN DS+5LIT" FOR "; SI+CPUIO; SI+SI+17; %504=GRK 14430032 D 504
      DS+7CHR; DS+LIT"/"; SI+SI+1; DS+7CHR; %504=GRK 14430033 D 504
      DS+LIT"="; DS+2DEC; I+DI; DI+DI-2; %504=GRK 14430034 D 504
      DS+5LIT"; PST="; DS+5DEC; %504=GRK 14430035 D 504
      I+DI; DI+DI-5; DS+4FILL; DI+1; %504=GRK 14430036 n 504
      DS+5LIT"; IOT="; DS+5DEC; I+DI; DI+DI-5; %504=GRK 14430037 D 504
      DS+4FILL; DI+1; DS+6LIT"; CORE="; %504=GRK 14430038 D 504
      DS+5DEC; I+DI; DI+DI-5; DS+4FILL; %504=GRK 14430039 D 504
      DI+1; DS+LIT"; %504=GRK 14430040 D 504
      END STREAM; %504=GRK 14430041 D 504
      SPOUTER(CPUIO,UNITNO,64); %504=GRK 14430042 D 504
      IF UNITNO#0 THEN %504=GRK 14430043 D 504
$:                                                                           D 504
$: THIS PATCH PROVIDES THE EOJ JOB STATISTICS MESSAGE SIMILAR TO          D 504
$: THE DCMCP MESSAGE. IN ADDITION, MAXIMUM CORE USAGE IS PROVIDED.      D 504
$:                                                                           n 504
$: SPOUTER DOCUMENTATION SPOUTER(MESSAGE,UNITNO,TYPE) n 504
$: ===== n 504

```

```

$!
$! DFFINES : SPOUT(MESSAGE) = SPOUTER(MESSAGE,0,1) D 504
$! ----- SPOUTIT(MESSAGE,TYPE) = SPOUTER(MESSAGE,0,TYPE) D 504
$! D 504
$! NOTES : --CONDITION-- MESSAGE DISPATCHED C O M M E N T S D 504
$! ----- UNITNO TYPE SPO PACKET LOG ----- D 504
$! N/A 0 N Y Y D 504
$! N/A 64 N Y N FOR PACKET PAGE ONLY D 504
$! N/A NFG. N Y N D 504
$! N/A TRUE Y Y Y D 504
$! D 504
$! G.R.KENNEDY 10JUN76. BROCK U. PATCH 504 D 504
$! THIS PATCH IS NEEDED BY PATCH 509 D 504
$!***** D 504

```

```

$# PATCH 508 FOR TSSMCP.XV.3 CONTAINS 4 CARDS. PREVENT FILEHOLD FOR "CHANGE" D 508
$! THIS REMOVES A POTENTIAL INVALID LINK PROBLEM IN PROCEDURE LIBCC WHICH D 508
$! WOULD OCCUR IF A FILE WERE REMOVED WHILE IN THE PROCESS OF BEING D 508
$! A) REMOVED OR B) CHANGED. D 508
$! THIS ALSO PREVENTS A FILE HOLD SITUATION FROM OCCURRING WHEN A CHANGE D 508
$! COMMAND IS ENTERED FROM A REMOTE TERMINAL. D 508
$! NATAL U. PATCH 508 D 508
$! IF T GEQ 64 THEN 20574917 D 508
$! T+DIRECTORYSEARCH(CMM[0],CMM[1]&(P(UNITNO,DUP)=25 OR 20577826 D 508
$! P(XCH)=30)[1;47;1],4); FND; 20577827 D 508
$! IF T GEQ 64 THEN 20577828 D 508
$!***** D 508

```

```

$# PATCH NUMBER 509 FOR TSSMCP.XVI.00 CONTAINS 17 CARDS,PRNPBT/LDCNTRL BOJ/EOJ D 509
$! IF P THEN MAKFLOG(MESSAGE,ABS(TYPE)); %509=GRK 02134000 D 509
$! END; IF TYPE ISS 0 THEN TYPE:=64; %509=GRK 02134005 D 509
$! SPOUTIT(LOGINF, %509=GRK 12704000 D 509
$! 64); %509=GRK 12704100 D 509
$! SPOUTIT(VECTOR INX 0, %509=GRK 14430000 D 509
$! IF NOT ( (M[CPU10+3] EQV "DISK ")=NOT 0 AND %509 14430002 D 509
$! ( (M[CPU10+2] EQV "PRNPBT ")=NOT 0 OR %50914430004 D 509
$! (M[CPU10+2] EQV "LDCNTRL")=NOT 0) ) %509 14430006 D 509
$! THEN ((LINK OR CANDYMESS) AND EOJMESS %509=GRK 14430008 D 509
$! AND (NOT JAR9).[2;1] OR EOJK) %509=GRK 14430010 D 509
$! ELSE "FOJK"); %509=GRK 14430012 D 509
$! SPOUTER(T,UNITNO, %509=GRK 20104200 D 509
$! IF NOT ( (JAR[MIX,1] EQV "DISK ")=NOT 0 AND %509=GRK 20104300 D 509
$! ( (JAR[MIX,0] EQV "PRNPBT ")=NOT 0 OR %509=GRK 20104310 D 509
$! (JAR[MIX,0] EQV "LDCNTRL")=NOT 0) ) %509=GRK 20104320 D 509
$! THEN ((NOT S[0]).[2;1] OR CANDYMESS) %509=GRK 20104330 D 509
$! AND BOJMESS AND I OR BOJK) ELSE "BOJK"); %509=GRK 20104340 D 509
$! D 509
$! THIS PATCH SUPPRESSES ALL "BOJ" AND "EOJ" SPO MESSAGES FOR D 509
$! "PRNPBT/DISK" AND "LDCNTRL/DISK". THIS SHOULD HELP ELIMINATE D 509

```

```

$! SPO "CLUTTER", D 509
$! NOTE THAT SPOUTER HAS BEEN MODIFIED TO LOG NEGATIVE MESSAGE TYPES D 509
$! BUT NOT SPOUT THEM (SIMILAR TO TYPE 64 - BUT TYPE IS RETAINED). D 509
$! D 509
$! G.R.KENNEDY 31MAY76. BROCK U. PATCH 509 D 509
$! THIS PATCH USES VARIABLE CPU10 WHICH IS DEFINED IN PATCH 504 D 509
$! CORRECTED TO INCLUDE BOJMESS BY DR. UCSC D 509
$! SEE ALSO DREXEL U. PATCH 716 D 509
$! ***** D 509

```

```

$# PATCH 510 FOR TSSMCP.XV.3 CONTAINS 2 CARDS. INITIALIZE TYPED PROCEDURES D 510
$! THIS PATCH CORRECTS AN ERROR INTRODUCED BY PATCH NUMBER 24 TO MCP.XV D 510
$! WHERE THE VARIABLE PROCVAL WAS NOT INITIALIZED IN THE PROCEDURES D 510
$! "ACESBIT" AND "CCCOMPILE". WHEN RETURNING FROM THE PROCEDURES THE D 510
$! VALUE OF THE TYPED PROCEDURES WAS TAKEN FROM THE LAST VALUE OF D 510
$! PROCVAL. THIS CAUSED ODD ERRORS. E.G. THE CARDS: D 510
$! Q LOAD FROM XYTAPE ANY/NAME D 510
$! Q COMPILE PROG/NAME FORTRAN D 510
$! WOULD RESULT IN "CONTROL CARD ERROR AT FORTRAN". THE TYPED D 510
$! PROCEDURES ARE NOW INITIALIZED. NOTE:- ACESBIT NOW CCSET D 510
$! NATAL U. PATCH 510 D 510
$!     CCCOMPILE+0; 20584340 D 510
$!     CCSET+0; 20711100 D 510
$! ***** D 510

```

```

$# PATCH 511 FOR TSSMCP.XV.3 CONTAINS 4 CARDS. "PC" WITH 65 JOBS ON DISK D 511
$! THIS IS TO FIX THE CASE WHERE THERE ARE 65 JOBS ON DISK AND A "PC" IS D 511
$! DONE. PRIOR TO THIS THE MESSAGE WOULD COME OUT AS:- 65 JOB, #NNNN D 511
$! WHERE "NNNN" WAS THE NUMBER OF THE LAST DECK ON DISK. D 511
$! NATAL U. PATCH 511 D 511
$!     SI+LOC T1; 07295100 D 511
$!     IF SC#"0" THEN GO TO AQ 07295102 D 511
$!     ELSE SI+SI+1; 07295104 D 511
$!     AQ; BEGIN DS+3 LIT "S, " 07295160 D 511
$! ***** D 511

```

```

$# PATCH 512 FOR TSSMCP.XV.3 CONTAINS 7 CARDS. ERROR PROGRAM PARAM CARDS D 512
$! THIS PATCH REARRANGES SOME CODING IN PROCEDURE PPC FOR CLARITY AND D 512
$! GENERATES A CONTROL CARD ERROR FOR IDENTIFIERS ON PROGRAM PARAMETER D 512
$! CARDS WHICH ARE UNDEFINED. D 512
$! NATAL U. PATCH 512 D 512
$!     ELSE IF TYPE=COREV THEN 20504500 D 512
$!     BEGIN X[20] + TPNO DIV 64; 20505000 D 512
$!     DO UNTIL (IOD + SCAN)=MAXV OR IOD=PERIOD; 20507000 D 512
$!     IF IOD=MAXV THEN P([X[20]],IOR) ELSE GO TO DOWN; 20507100 D 512

```

```

END 20507200 D 512
ELSE IF TYPE>=PRIOR AND TYPE<=SAVEV THEN 20507300 D 512
X[18+TYPE=PRIOR]+TPNO ELSE GO TO ERROR; 20507400 D 512
$!***** n 512

$# PATCH 517 FOR TSSMCP CONTAINS 3 CARDS. OVERLAY DISK SPACE USED UP D 517

$: WHEN A PROGRAM HAS EXCEEDED ITS OVERLAY DISK SPACE, THIS PATCH WILL D 517
$: CAUSE THE MCP TO ISSUE THE MESSAGE "ARRAY OLAY SPACE USED UP" WHEN D 517
$: DS=ING THE PROGRAM, INSTEAD OF "INVALID ARRAY SIZE IDN". THE LATTER D 517
$: MESSAGE IS MISLEADING. D 517
$: BY STEVE KELLEY, REQUESTED BY NATAL U. MAY, 1974 D 517
$: NATAL U. PATCH 517 D 517
$: MODIFIED FOR TSSMCP BY DR. UCSC JAN. 17, 1977 D 517
TERMINATE (MIX& 99[CTF]); 06411010 D 517
"8ARRAY ", "8OLAY S", % 99 41445700 D 517
"8PACE U", "6SFD UP", % 41445800 D 517
$!***** D 517

$# PATCH NUMBER 519 FOR TSSMCP.XVI.00 CONTAINS 01 CARDS.ENABLE "FILE IN USE" OK. D 519

BEGIN T+VWY&VOK[36:42:6]&(VIFXA.[3:1])[30:42:6]; %519=GRK 18025000 D 519
$: D 519
$: THIS PATCH ENABLES "#FIL IN USE" SITUATIONS TO BE "OK=FD". D 519
$: PREVIOUSLY (EXCEPT FOR LIBMAIN) THE ONLY ALTERNATIVE WAS "DS". D 519
$: D 519
$: G.R.KENNEDY 30JAN76. BROCK U. PATCH 519 D 519
$: SEE ALSO UCSC PATCH 815 D 519
$!***** n 519

$# PATCH 521 FOR TSSMCP.XV.3 CONTAINS 17 CARDS. FIX PROCEDURE "CCSET" D 521

$: THIS PATCH FIXES THE MESS THAT PROCEDURE CCSET IS IN. D 521
$: THE PROBLEMS FIXED ARE: D 521
$: A) FILE HEADERS OF ALL FILES REFERENCED BY THIS PROCEDURE ARE LEFT D 521
$: IN CORE. D 521
$: B) DIRECTORY WAS BEING UNLOCKED IN MANY CASES WHEN IT WAS NOT LOCKED D 521
$: BY THIS PROCEDURE. D 521
$: C) DISK HEADER WAS BFING WRITTEN BACK UNALTERED AT LINE 20739000 D 521
$: INSTEAD OF A FORGFTSPACE ON THE HEADER. D 521
$: NATAL U. PATCH 521 D 521
BEGIN T+2; GO TO L1; END; 20730000 D 521
BEGIN FORGFTSPACE(T); T+1; GO SKIP; ENn; 20739000 D 521
$ VOID 20739100 n 521
$ VOID 20739200 n 521
ELSE BEGIN 20769000 D 521
$ SET OMIT = SHAREDISK 20769100 D 521
UNLOCKDIRECTORY; 20769200 D 521

```



```

$ POP OMIT
    LRMESS(CMM[2],CMM[3],=(11+TOG),41,0, SPOUTUNIT,1);
    END;
    FORGETSPACE(T);
    ELSE BEGIN
$ SFT OMIT = SHARFDISK
    UNLOCKIRECTORY;
$ POP OMIT
    END;
$ VOIDT 20772301
$!*****
20769300 D 521
20769400 D 521
20769500 D 521
20769600 D 521
20771000 D 521
20771010 D 521
20771020 D 521
20771030 D 521
20772050 D 521
20772100 D 521
n 521

```

```

$# PATCH 523 FOR TSSMCP.XV.3 CONTAINS 1 CARD. "RESERVE DISK REMOVED" SPO ONLY D 523
$: THIS PATCH MAKES THE *** RESERVE DISK REMOVED MESSAGE GO TO THE SPO D 523
$: ONLY. PREVIOUSLY IT ALSO WENT TO THE MESSAGE PAGE FOR THE JOB. D 523
$: NATAL U. PATCH 523 D 523
    SPOUTFR(Z,25,(NOT LTRMSG) AND 1); 06353660 n 523
$!***** D 523

```

```

$# PATCH 527 FOR TSSMCP CONTAINS 3 CARDS. FIXES TO "SET" AND "RESET" D 527
$: THIS WILL ALLOW ONE TO ENTER A "SET" OR "RESET" CONTROL CARD FROM D 527
$: THE SPO AFTER A CONTROL CARD ERROR ON THE LAST INFORMATION ENTERED D 527
$: FROM THE SPO. PRIOR TO THIS, EVEN THOUGH THE CORRECT INFORMATION WAS D 527
$: ENTERED FOR A SET OR RESET CONTROL CARD, THE FIRST WORD WOULD NOT BE D 527
$: RECOGNIZED, RESULTING IN A FURTHER CONTROL CARD ERROR. D 527
$: THIS WILL ALSO FIX THE CASE WHERE A SET OR RESET CONTROL CARD D 527
$: FOLLOWED AN EXECUTE CARD. THE SET OR RESET WAS BEING TAKEN AS A D 527
$: PROGRAM PARAMETER CARD. D 527
$: NATAL U. PATCH 527 D 527
$: MODIFIED FOR TSSMCP BY DR, UCSC JAN. 17, 1977 n 527
CONTROL: IF (T < FILEV OR T > COBOL) AND ACCUMFO] # CMPLR THEN 20604300 D 527
    IF T GEQ UNLOCKV AND T LEQ RESETV THEN GO TO FINIS 20604350 D 527
    IF (T#PACKET) AND (T#RESETV) AND (T#RUNV) THEN 20608078 D 527
$!***** n 527

```

```

$# PATCH 528 FOR TSSMCP CONTAINS 1 CARD. TRY DISK SPACE AFTER RESERVES REMOVE D 528
$: THIS CAUSES ANOTHER ATTEMPT TO GET DISK SPACE AFTER RESERVE DISK IS D 528
$: REMOVED INSTEAD OF SPOUTING NO USER DISK AND FORCING THE OPERATOR TO D 528
$: DO AN "OK" FOR THE JOB. D 528
$: NATAL U. PATCH 528 D 528
$: MODIFIED FOR TSSMCP BY DR, UCSC JAN. 17, 1977 n 528
    FORGETSPACE(BUFF); P(XIT); 06353675 D 528
$!***** D 528

```

```

$# PATCH 534 FOR TSSMCP CONTAINS 3 CARDS.  FIX "XT" & "CT" FOR *N & "<EMPTY>  D 534
$! THIS PATCH CORRECTS PROCEDURE "TIMERELAXER" SO THAT THE INPUT          D 534
$! COMMANDS "XT" AND "CT" WORK CORRECTLY.  PREVIOUSLY IF THE CHANGES TO  D 534
$! THE TIME ESTIMATES WERE OF THE FORM *N THEN AN INV KBD RESULTED.      D 534
$! THIS WAS BECAUSE AFTER FINDING THE * , SI WAS NOT INCREMENTED TO SKIP  D 534
$! OVER IT.                                                                D 534
$! THIS ALSO FIXES THE CASE WHERE THE IOT IS OF THE FORM "<EMPTY>".      D 534
$! PREVIOUSLY, THIS WOULD ALSO HAVE GIVEN AN "INV KBD".                  D 534
$! NATAL U.  PATCH 534                                                    D 534
      IF SC=" " THEN GO L1; END;                                           08739000 D 534
      IF SC="*" THEN REGIN SI+SI+1; GO L5; END;                             08739500 D 534
      IF SC="<" THEN BEGIN TALLY+1; GO EXIT; END;                         08749500 D 534
$!*****                                                                    D 534

```

```

$# PATCH 535 FOR TSSMCP.XV.3 CONTAINS 23 CARDS.  FIX PROCEDURE CHANGEOPTION D 535
$! THIS PATCH FIXES SOME BAD CODING IN PROCEDURE CHANGEOPTION WHERE THE   D 535
$! SCAN MAY GO PAST THE END OF THE INPUT BUFFER.  IT IS A RE-WRITE OF     D 535
$! THE SCAN PORTION.  IT ALSO MAKES THE USE OF THE WORD "USE" OR "U"      D 535
$! OPTIONAL IN THE SO OR RO WITH THE OPTION MNEMONIC (AS WITH THE OPTION   D 535
$! NUMBER).                                                                D 535
$! NATAL U.  PATCH 535                                                    D 535
      LO: SI+BUFF;                                                         08634000 D 535
      L1: IF SC=" " THEN BEGIN SI+SI+1; GO TO L1; END;                     08635000 D 535
          IF SC<"0" THEN                                                  08636000 D 535
          IF SC#"<" THEN                                                  08637000 D 535
              BEGIN TALLY+0; T+TALLY; DI+LOC T;                          08638000 D 535
                  B(IF SC=" " THEN JUMP OUT ELSE                          08639000 D 535
                    IF SC="<" THEN JUMP OUT ELSE                          08640000 D 535
                    IF SC>"0" THEN JUMP OUT ELSE DS+CHR);                08641000 D 535
                  RUFF+SI; SI+OPTER;                                       08642000 D 535
                  63(DI+LOC T;                                             08643000 D 535
                    IF B SC=DC THEN JUMP OUT TO L2 ELSE                    08644000 D 535
                    IF SC="<" THEN JUMP OUT TO LO ELSE TALLY+TALLY+1); 08645000 D 535
                  GO TO L3;                                                08646000 D 535
          L2: IF SC="<" THEN GO TO LO;                                       08647000 D 535
              END ELSE                                                       08648000 D 535
          L3: TALLY+48 ELSE                                                  08649000 D 535
              REGIN DI+LOC T; SI+SI+1;                                     08650000 D 535
              IF SC<"0" THEN REGIN SI+SI-1; DS+1 OCT; FND                08651000 D 535
                  ELSE REGIN SI+SI-1; DS+2 OCT; FND;                     08652000 D 535
              TALLY+47; T(TALLY+TALLY+63);                                  08653000 D 535
              END;                                                         08654000 D 535
          T+TALLY; SI+LOC T; DI+R; DS+WDS;                                  08655000 D 535
      $ VOIDT                                                                08656000 D 535
$!*****                                                                    D 535

```

```

$# PATCH 538 FOR TSSMCP.XV.3 CONTAINS 2 CARDS.  FIX TIMING PROBLEM WITH PURGE D 538
$! THIS PATCH CORRECTS A TIMING PROBLEM WITH A PROGRAMMATIC PURGE OF A   D 538

```

```

$: TAPE FILE. THE PURGE IS HANDLED BY AN INDEPENDENT RUNNER WHICH HAS      D 538
$: TO WAIT UNTIL THE TAPE HAS REWOUND. IF A PROGRAM CLOSES FILE "A/B"      D 538
$: WITH PURGE AND THEN TRIES TO OPEN FILE "A/R", IF THE PURGE HAS NOT      D 538
$: OCCURRED, IT MAY TRY TO GRAB THAT UNIT (SEE FINDOUTPUT @ 37027000).      D 538
$: THE PRNTABLE STILL CONTAINS THE POINTER TO THE TOP I/O DESCRIPTOR,      D 538
$: THE SPACE FOR WHICH HAS ALREADY BEEN FORGOTTEN, SO THAT IT MAY PICK      D 538
$: UP GARBAGE. THIS IS FIXED BY SETTING THE MIX INDEX OF THE PROGRAM      D 538
$: ASSIGNED TO THE UNIT TO ZERO IN THE RDCTABLE BEFORE STARTING THE      D 538
$: INDEPENDENT RUNNER TO PURGE THE TAPE.                                    D 538
$: NATAL U. PATCH 538                                                       D 538
      BEGIN RDCTABLE[UNIT].[8:6]+0;                                         38788500 D 538
      FORK(PC.PURGEIT),U,-2,128,1) END FLSE SFTNOTINUSE(U,0);             38789000 D 538
$!*****                                                                    D 538

```

```

$# PATCH 539 FOR TSSMCP.XV.3 CONTAINS 2 CARDS. PURGE OF TAPE AFTER CLLP*   D 539
$: THIS PATCH CORRECTS THE CLLP* WHEN PRINTING FROM A BACKUP TAPE WITH A    D 539
$: W/RING. THE TAPE IS NOW REWOUND AND PURGED CORRECTLY. PREVIOUSLY,      D 539
$: THE PURGE DID NOT TAKE PLACE. PRIOR TO THIS PATCH, PRNPBT/DISK         D 539
$: REWOUND THE TAPE AND STARTED AN INDEPENDENT RUNNER (PURGEIT) TO DO     D 539
$: THE PURGE. TERMINALMESSAGEA VIA UNHOOQUE FOUND THE TAPE IN USE BY      D 539
$: PRNPBT/DISK FROM RDCTABLE[UNIT].[8:6] AND SET THE UNIT NOT READY IN    D 539
$: LABELTABLE (=0114). WHEN THE REWIND WAS COMPLETE, STATUS FINDS THE     D 539
$: TAPE NOT READY AND FORGETS THE PURGE, LEAVING PURGEIT UNLINKED IN      D 539
$: CORE WAITING ON AN I/O COMPLETE. (UNIT SAYS NO I/O=S BUT IOQUE SHOWS   D 539
$: I/O=S FOR WHICH THE SPACE HAS NOT BEEN RETURNED).                       D 539
$: NATAL U. PATCH 539                                                       D 539
      BEGIN RDCTABLE[UNIT].[8:6]+0;                                         12528400 D 539
      END                                                                    12528600 D 539
$!*****                                                                    D 539

```

```

$# PATCH 540 FOR TSSMCP CONTAINS 1 CARD. FIX "IL" FOR "#NO FIL ON DISK"    D 540
$: THIS PATCH CORRECTS THE RESULT OF AN "IL" INPUT MESSAGE IN REPLY TO    D 540
$: A "#NO FIL ON DISK" CONDITION. PREVIOUSLY, IF AN "ILCD*" WAS INPUT,    D 540
$: A SYSTEM HANG RESULTED. THIS WAS BECAUSE THE TEST WAS FOR AN ACTUAL    D 540
$: UNIT RATHER THAN ALL UNITS.                                             D 540
$: NATAL U. PATCH 540                                                       D 540
$: MODIFIED TO USE MAGIC NUMBER 35, BECAUSE TSSMCP DOES NOT HAVE         D 540
$: PSFUDOMAXT. OBSERVE ON LINE 00277000 OF TSSMCP THAT TINU[35] = DATA  D 540
$: FOR THE LAST PSFUDD READER. BY DR. UCSC JAN. 17, 1977                 D 540
      IF (SH+T2+RPLY[PIMIX],[FF]) > 35 THEN % IL 06463380 D 540
$!*****                                                                    D 540

```

```

$# PATCH 543 FOR TSSMCP.XVI.0 CONTAINS 13 CARDS. FIXES TO NEW LIBMAIN     D 543
$: THIS CORRECTS A NUMBER OF PROBLEMS IN THE NEW LIBMAIN/DISK.           D 543
$: 1. IF MORE THAN ONE EXCEPTION LIST APPEARS WITHIN ONE CONTROL CARD,    D 543

```

```

S: THEN THE SPACE FOR ALL EXCEPT THE LAST IS NOT FORGOTTEN. D 543
S: FIXED AT 20570070 AND 20573350. D 543
S: 2. IF THE EXCEPTION LIST CONTAINS MORE THAN 13 FILE NAMES AND THERE D 543
S: IS AN AS CLAUSE, THEN A CONTROL CARD ERROR MAY OCCUR EVEN THOUGH D 543
S: THERE REALLY IS NOT ONE. THIS IS BECAUSE THE ORIGINAL FILE NAME D 543
S: HAS BEEN OVERWRITTEN BY THE EXCEPTION LIST. FIXED AT 20569930 ETC D 543
S: 3. FIX TO USE DEFINES RATHER THAN CONSTANTS. D 543
S: NATAL U. PATCH 543 D 543
    IF T GEQ COPYN AND T LEQ LOAD THEN 20566807 D 543
    IF T GEQ COPYN AND T LEQ LOAD THEN 20566822 D 543
    T1.[46:1]+HOLD3+(CN=FEQUAL); 20569930 D 543
    IF HOLD3 THEN PROGICNT1;=-1 ELSE 20569935 D 543
    IF CN GEQ IDENT THEN PROGICNT1;=ACCUM[0] ELSE GO POWIE; 20569940 D 543
    T1.[47:1]+HOLD3+((CN+SCAN)=EQUAL); 20569960 D 543
    IF HOLD3 THEN PROGICNT+1;=-1 ELSE 20569965 D 543
    FORGETSPACE(XLST); XLST:=0; 20570070 D 543
    IF (CN+SCAN)=EQUAL THEN IF T1.[46:1] THEN 20570130 D 543
    ELSE IF CN GEQ IDENT THEN IF T1.[46:1] THEN GO POWIE 20570150 D 543
    IF (CN+SCAN)=EQUAL THEN IF T1 THEN 20570190 D 543
    ELSE IF CN GEQ IDENT THEN IF T1 THEN GO POWIE 20570210 D 543
    IF XLST NEQ 0 THEN BEGIN FORGETSPACE(XLST); XLST:=0; END; 20573350 D 543
S:***** D 543

```

```

S# PATCH 548 FOR TSSMCP.XVI.0 CONTAINS 1 CARD. WRONG FIB CHECK IN FINDOUTPUT D 548
S: THIS PATCH PREVENTS FINDOUTPUT FROM CHECKING ON A NON-EXISTENT FIB. D 548
S: THIS SHOWED UP BY USING: D 548
S: CC COPY =/= FROM A TO A; END D 548
S: WHERE THE SOURCE TAPE HAD A WRITE RING, LIBMAIN/DISK READ THE D 548
S: DIRECTORY CORRECTLY AND THEN TRIED TO FIND THE OUTPUT TAPE. D 548
S: FINDOUTPUT THEN FOUND THE TAPE OF CORRECT NAME, IN USE BY D 548
S: LIBMAIN/DISK, WITH A WRITE-RING AND TRIED THE "FIB". D 548
S: PRNTABLE[U],[15:15] WAS ZERO SO THAT M[M[3]+5],[41:1] WAS TESTED. D 548
S: (M[3] CONTAINS THE MCP VERSION NUMBER). THE CONTENTS OF THE MEMORY D 548
S: CELL TESTED COULD BE SUCH AS TO SATISFY THE TEST. THUS, LIBMAIN/DISK D 548
S: THEN WROTE AN OUTPUT LABEL ON THE SOURCE TAPE AND FINALLY WHEN TRYING D 548
S: TO DO THE NEXT READ, DECIDED THE TAPE WAS NOT A LIBRARY TAPE. D 548
S: THIS WOULD ONLY OCCUR IF THE SOURCE TAPE HAD A NON-ZERO PRN. D 548
S: BY RAY BARTHO, NATAL U. PATCH 548, MARCH 29, 1976 D 548
    IF T3.[15:15]#0 THEN % DONT USE NONEXISTENT FIB 37027500 D 548
S:***** D 548

```

```

S# PATCH 549 FOR TSSMCP.XVI.0 CONTAINS 5 CARDS. AUTO "OU" FOR "NO SORT MEM" D 549
    STRFAM(P1MIX,T+R5xR6,A+I+SPACE(7)); 18790600 D 549
    BEGIN DS=LIT ". "; S1=LOC P1MIX; %SPOUT ". ", NOT "#". %GRK=549 18790700 D 549
    REPLY[P1MIX],[CF] + 3; 18791200 D 549
    J + 1; 18791300 D 549
S VOINT 18791701 18791400 D 549
S: THIS PATCH PROVIDES AN AUTOMATICALLY ASSUMED RESPONSE OF "OU" TO THE D 549
S: "NO SORT MEM" MESSAGE. IN ADDITION, THE PREFIX OF THE MESSAGE D 549

```

```

$! HAS BEEN ALTERED TO "." FROM "#", TO IDENTIFY THIS MESSAGE AS          D 549
$! "INFORMATION ONLY" INSTEAD OF A "REQUEST FOR OPERATOR INTERVENTION".    D 549
$! G.R.KENNEDY 12FEB75. BROCK U. PATCH 549                                D 549
$! MODIFIED BY DR. UCSC, TO DELETE FROM THE SOURCE, THE CODE THAT NEVER   D 549
$! CAN BE EXECUTED AS A RESULT OF BROCK U. PATCH 540, JAN. 17, 1977       D 549
$! SEE ALSO UCSC DCMCP PATCH 713, NATAL U. PATCH 713, DREXEL U. PATCH 751  D 549
$!*****                                                                    n 549

```

```

$# PATCH 550 FOR TSSMCP.XVI.0 CONTAINS 3 CARDS. RESTART LOG AFTER INTERRUPT D 550

$! IF AN EXTERNAL INTERRUPT E.G. IOBUSY, IOCOMPLETE, PRINTERFINISH,        D 550
$! TIMER OCCURS SIMULTANEOUSLY WITH A PROCESSOR DEPENDENT INTERRUPT E.G.   D 550
$! COMMUNICATE, THEN THE EXTERNAL INTERRUPT WILL BE HANDLED FIRST AND     D 550
$! THE LOGGING STOPPED FOR THE JOB. IF THERE ARE NO INDEPENDENT           D 550
$! RUNNERS WAITING TO BE STARTED AND THERE IS NOTHING IN THE BED, THE     D 550
$! LOGGING WILL NOT BE RESTARTED FOR THE JOB WHEN ITS INTERRUPT IS        D 550
$! HANDLED. THIS WAS DISCOVERED WHEN USING THE TIME(2) FUNCTION.          D 550
$! BY RAY BARTHO, NATAL U. PATCH 550, APRIL 21, 1976                       D 550
$ SET OMIT = NOT(NEWLOGGING)                                               48100499 n 550
    STARTLOG(PIMIX)                                                         48100500 n 550
$ POP OMIT                                                                  48100501 n 550
$!*****                                                                    n 550

```

```

$# PATCH 560 FOR TSSMCP CONTAINS 4 CARDS, SMALLER PRD ROW SIZES           D 560

$! BROCK U. PATCH 560, NATAL U. PATCH 732                                  D 560
DEFINE PAGFSIZE = 450#; % PACKET PAGE PBD SIZE                            02113091 n 560
    150# % SEGMENTS PER ROW                                               08699150 D 560
    050# % PHYSICAL RECORDS PER ROW                                       08699350 D 560
    1000#% PHYSICAL RECORDS PER FILE                                       08699500 D 560
$!*****                                                                    n 560

```

```

$# PATCH 601 FOR TSSMCP CONTAINS 2 CARDS. FIX PRD NO DISK DS              00000100 D 601

$! CORRECT CLOSING ACTION FOR PRD/PUD FILES IN RACKCLOSE AFTER            00000200 n 601
$! A NO USER DISK DS; FILE WOULD GET STUCK IN USE.                       00000300 D 601
$! DREXEL U. PATCH 601 8/10/75                                            00000400 n 601
$! MODIFIED FOR TSSMCP BY DR. UCSC FEB. 4, 1977                          00000500 D 601
$! SEE ALSO OFFICIAL PATCH 155 TO DCMCP OR 148 TO TSSMCP, AND            00000600 n 601
$! DREXEL U. PATCH 662                                                    00000700 D 601
    END;                                                                    38601100 D 601
$ VOIDT                                                                    38621500 D 601
$!*****                                                                    n 601

```

```

$# PATCH 602 FOR TSSMCP CONTAINS 4 CARDS. BAD INCW                        D 602

```





```

$# PATCH 645 FOR TSSMCP CONTAINS 1 CARD.  FIX DISKIO          00000100  D  645
$! THIS PATCH INTEGRIZES THE ARGUMENTS TO DISKIO TO INSURE    00000200  D  645
$! THAT THE VALUES ARE INTERPRETED CORRECTLY.  BAD THINGS MIGHT 00000300  D  645
$! HAPPEN IF A REAL (NORMALIZED) NUMBER IS PASSED TO DISKIO.  00000400  D  645
$! DREXFL U. PATCH 645 12/2/73                                00000500  D  645
$!      CORF:=CORF; SIZE:=SIZE; DISK:=DISK; & INTEGRIZE        06004100  D  645
$!*****99999999  D  645

```

```

$# PATCH 647 FOR TSSMCP CONTAINS 2 CARDS.  + FOR SPOUT          D  647
$! THIS PATCH PREVENTS PROBLEMS WITH DATA PASSED TO SPOUT WITHOUT A      D  647
$! GROUP MARK (*).  SYMPTOMS USUALLY INCLUDE A BOMBED MEMORY LINK.        D  647
$! A "*" IS PLACED IN THE LAST CHARACTER OF THE MESSAGE.                D  647
$! DREXEL U. PATCH 647 2/2/74                                           D  647
$!      STREAM(X*((MESSAGE-1),[CF]-1) & 7 [CTF]));                      02142100  D  647
$!      DS + LIT "*";                                                    02142110  D  647
$!*****  D  647

```

```

$# PATCH 654 FOR TSSMCP CONTAINS 4 CARDS.  FIX LDCNTRL          00000100  D  654
$! THIS PATCH VOIDS SOME CODE FROM COM23 2/25/74                    00000200  D  654
$! IF LDCNTRL WAS STARTED FROM A PSEUDO DECK AND DS ON A NO FILE        00000300  D  654
$! THE SYSTEM WOULD LOSE WHICH PSEUDO=RDR THE JOB HAD STARTED FROM      00000400  D  654
$! WHEN LDCNTRL ENDED THE ACTIVITY COUNT FOR THE PSEUDO=RDR            00000500  D  654
$! WOULD NOT BE COUNTED DOWN THUS LEAVING THE DECK STUCK FOREVER        00000600  D  654
$! (OR UNTIL H/L WHICH EVER COMES FIRST)                                00000700  D  654
$! 7/3/74 CORRECTS A PROBLEM WITH LDCNTRL ENDING IF THE CARD READER     00000800  D  654
$! GOFS NOT READY AFTER READING A CONTINUE CARD                        00000900  D  654
$! DREXEL U. PATCH 654                                                00001000  D  654
$!      IF FIRST AND CDONLY AND NOT CONTINUE THEN                      07059110  D  654
$!      GO EXIT ELSE                                                    07059112  D  654
$ VOIDT 07071912                                                       07071899  D  654
$ VOIDT 07072902                                                       07072899  D  654
$!*****99999999  D  654

```

```

$# PATCH 675 FOR TSSMCP CONTAINS 5 CARDS.  PRNPRT ES=ED        D  675
$! THIS PATCH FIXES THE WAY TERMINATE HANDLES AN ES=FD PRNPRT/DISK.      D  675
$! THE CODE WAS ALL WRONG SINCE XV.2 REWROTE PRNPBT.                  D  675
$! NOW THE UNITS (PRINTER, PUNCH, TAPE) SHOULD BE CLEARED PROPERLY.     D  675
$! DREXEL U. PATCH 675 6/15/74                                         D  675
$! SEE DREXFL U. PATCHES 750, 826                                       D  675
$!      BEGIN                                                            02261250  D  675
$!      IF (LUN+L.[41:5])<16 THEN SLAPITOFF;                            02261300  D  675
$!      LUN+L.[46:2]+19; & LPA, LPB, OR CPA                            02261350  D  675

```



SLAPITOFF;	02261400	n	675
\$ VOINT 02261701	02261450	n	675
\$!*****		n	675
\$# PATCH 688 FOR TSSMCP CONTAINS 2 CARDS, FALSE CONTROLCARD ERRORS		D	688
CCOMPILE=0;   % INITIALIZE	20584390	n	688
CCUNIT=0;    % INITIALIZE	20590030	n	688
\$: THIS PATCH WILL CORRECT A PROBLEM WITH INCORRECT CONTROL CARD ERRORS		n	688
\$: DREXEL U. PATCH 688    4/27/75		D	688
\$: MODIFIED FOR TSSMCP BY DR, UCSC   FEB. 7, 1977		D	688
\$!*****		n	688
\$# PATCH 697 FOR TSSMCP CONTAINS 2 CARDS, IO FRR STATUS		D	697
\$: THIS PATCH CORRECTS A PROBLEM WHERE STATUS COULD GET AN UNEXP IO FRR		n	697
\$: WHILE PLAYING WITH A TAPE DRIVE. IT SHOULD RECOVER OK NOW.		D	697
\$: DREXEL U. PATCH 697    1/25/76		D	697
DO UNTIL (T1=WAITIO(AREA INX @340000012,@75,U))#0;	22112000	n	697
IF T1.[43:2]#0 THEN T1=WAITIO(@4200000000,5,U);	22112100	n	697
\$!*****		D	697
\$#PATCH NUMBER 708 FOR TSSMCP.XIII CONTAINS 3 CARDS PRINT PRN WITH PG		D	708
IF TFST THEN BEGIN STREAM(B+T,BUFF);	08051000	n	708
BEGIN DS=10 LIT"PG=ED,PRN="; SI=LOC B; DS=5 DEC; DS=2 LIT")+";	08051004	D	708
END;                                    END % PRINT PRN WITH PLAIN PGMT	08051005	D	708
\$: FROM DREXEL U.		D	708
\$: SEE ALSO BROCK U. PATCH 543 FOR TSSMCP; PATCH 708 REQUIRES LESS CODE		n	708
\$!*****		n	708
\$# PATCH 715 FOR TSSMCP.XV.3 CONTAINS 1 CARD. MT IO "ERRORS" TO "RETRIES"		D	715
\$: CHANGE "ERRORS" TO "RETRIES" IN MAG TAPE IO MESSAGE, BECAUSE USERS		D	715
\$: GET WORRIED BY IO ERRORS. (AS THEY RIGHTFULLY SHOULD. DR,UCSC)		D	715
\$: NATAL U. PATCH 715; SPELLING BY DR, UCSC		D	715
[12,30:18], "RETRIES",FPR[FN],FPR[FN+1],J,0,0);	37285320	n	715
\$!*****		n	715
\$# PATCH 717 FOR TSSMCP.XV.3 CONTAINS 13 CARDS, BACKUP TAPES NOT AUTOPRINT		D	717
\$: TO MAKE PRINTER BACKUP TAPES IGNORED BY THE AUTOPRINT MECHANISM.		D	717

```

$! MODIFIED FOR MARK XV.2 BY RAY BARTHO AUG. 1, 1974 D 717
$! MODIFIED FOR MARK XV.3 BY RAY BARTHO NOV. 1, 1974 D 717
$! MONASH PATCH 556, NATAL U. PATCH 717 D 717
    LARFL FOUND,FIREITUP,QUIT; 08255800 D 717
    FND ELSE % Q<=0, PR MT 08259800 D 717
    BEGIN RRRMECH+TWO(U+ABS(Q)) OR RRRMECH; 08259810 D 717
    LABELTABLE[U] + 08259820 D 717
    PRT&TINU[V][6:30:18]&@21[1:43:5]; 08259830 D 717
    MULTITABLE[V] + PRT; 08259840 D 717
    LABELTABLE[V] + PRT&TINU[U][6:30:18]& 08259850 D 717
    @21[1:43:5]; 08259860 D 717
    GO FIREITUP; 08259870 D 717
    FND 08259880 D 717
$ VOIDT 0826650; 08260899 D 717
$ VOIDT 3863910; 38639000 D 717
    IF J THEN SETNOTINUSE(U,0) ELSE LABELTABLE[U],P1:5)+1; 38640000 D 717
$!***** D 717

```

```

$#PATCH NUMBER 724 FOR TSSMCP CONTAINS 13 CARDS. FULL PAGE D 724

```

```

    IF RDCTABLE[U] OR MULTITABLE[U]="FULLPGE" 04391550 D 724
    THEN IF IOQUEF[$],[28:1] THEN IOQUEF$],[FF]+@40013 %DBL=CH 11 04391560 D 724
    ELSE IOQUEF$],[FF]+@40012 % DBL SINGLE = SKIP TO CH 10 04391570 D 724
    ELSE % SKIP TO CHAN 1 ON EOP IF NOT 66 LINES 04391580 D 724
RDCTABLE[V],[47:1]+INREC(O)="FULLPGE"; % LINES66 OPTION 12860100 D 724
    FORM = 76#, % SWITCH D(PCC)"FORM"="SPECIAL" 20240000 D 724
    LINES66 = 77#, % 66 LINES PER PAGE 20240020 D 724
    LINES66, % 20392880 D 724
    LINES66, % 20394080 D 724
LLINES66: % 20465050 D 724
    EQN[0] + "FULLPGE" ; %SET UP MFID FOR FULL PAGE 20465100 D 724
    "FORM " , 76 ; 41485600 D 724
    "LINES66" , 77 ; 41485620 D 724
$! D 724
$! REV 12/9/72 MARK XIV D 724
$! THIS PATCH WILL CAUSE THE CHANNEL 12 PUNCH ON THE VERTICAL D 724
$! FORMAT TAPE TO BE OVERLOOKED D 724
$! THE PRINTER IS CLEARED BY DOING A SKIP OF A SINGLE OR DOUBLE D 724
$! SPACE OR IS NOT CLEARED AT ALL DEPENDING ON THE ORIGINAL D 724
$! CONTROL INFORMATION D 724
$! THIS WAS TESTED ON MOD III IOS WITH B9242=4 PRINTERS D 724
$! THE DOCUMENTATION SAYS THIS SHOULD NOT WORK I.E. D 724
$! YOU ARE SUPPOSED TO CLEAR THE PRINTER WITH A SKIP TO CHANNEL D 724
$! THIS MIGHT BE THE CASE WITH DIFFERENT PRINTERS AND/OR IOS D 724
$! THIS VERSION WILL WRK ON ANY CARRIAGE TAPE ELIMINATING THE NEED D 724
$! FOR SPECIAL PUNCHES ON CHANNELS 10 & 11 D 724
$! D 724
$! MODIFIED JULY 1973 = MARK XV.1 D 724
$! MODIFIED 02/10/74 = MARK XV.2 D 724
$! MODIFIED 12/21/75 TO MAKE OTHER PATCHES WORK BETTER D 724
$! FROM DREXEL UNIVERSITY = JHH = 5/21/76 D 724
$! NONE OF THE ABOVE STUFF IS TRUE ANYMORE = IT WON'T WORK WITH B=321S D 724
$! THE PROBLEM IS THAT WHEN THE B=321 HITS CHANNEL 12 PUNCH IT KEEPS D 724
$! THE END OF PAGE STATUS BIT ON UNTIL A SKIP TO SOMETHING IS GIVEN D 724

```

```

$! THIS PUTS MCP INTO A TIGHT LOOP. SO, WE GO BACK TO USING A          D 724
$! SPFCIAL FORMAT TAPE WHICH HAS CHANNEL 11 PUNCHED A DOUBLE SPACE      D 724
$! AFTER CHANNEL 12, AND CHANNEL 10 PUNCHED A SINGLE SPACE AFTER      D 724
$! THIS PATCH AFFECTS PATCH 603                                          D 724
$!*****                                                                D 724

```

```

$# PATCH NUMBER 725 FOR TSSMCP,XIII CONTAINS 5 CARDS DONT ALLOW PRN=0   D 725

      DI+DI=8) DS=LIT "+";                                08029046 D 725
      IF NOT T.(1:1) THEN IF T=0 THEN                       08033980 D 725
      IF T.(1:1)=0 AND T=0 THEN BEGIN                       08033992 D 725
      STREAM(BUFF); DS=17 LIT "NOT PG=ED(PRN=0)+";          08033993 D 725
      LABELTABLE[U]+ @14) GO EXIT END) T+ABS(T);          08033994 D 725
$!                                                                    D 725
$! THIS PATCH WILL PROHIBIT THE OPERATOR FROM PURGING A TAPE THAT      D 725
$! HAS A PHYSICAL REEL NUMBER OF ZERO WITH THE PG MESSAGE UNLFSS      D 725
$! HE ENTERS A NUMBER, SUCH AS "PGMTB=122". HE MAY ENTER "PGMTC=0"    D 725
$! IF FOR SOME REASON A PRN IS NOT APPLICABLE                       D 725
$! FROM DREXEL U.                                                    D 725
$!*****                                                                D 725

```

```

$# PATCH 745 FOR TSSMCP CONTAINS 3 CARDS. PBT REFL NUMBERING          D 745

$! THIS CORRECTS THE PROBLEM WHERE A JOB USING PBT USED MORE THAN ONE  D 745
$! REFL BEFORE ENDING. A SUBSEQUENT JOB REQUIRING PBT CONTINUED ON THE D 745
$! FINAL REEL OF THE ABOVE JOB, BUT WHEN A REFL CHANGE OCCURRED, THE  D 745
$! VALUE OF REEL FOR THE CURRENT JOB WAS PUT IN THE LABEL OF THE NEW   D 745
$! REFL. THIS MEANT THAT WHEN PRINTING, THE REFL NUMBERS IN THE LABELS D 745
$! WERE NOT CONSECUTIVE THUS INTERFERING WITH AUTOMATIC REEL SWITCHING. D 745
$! THE REEL NUMBER OF THE FIRST TAPE IS NOW PLACED IN THE RDCTABLE, AND D 745
$! ASSIGNED TO NEW JOBS USING THE SAME PBT FROM THE RDCTABLE. THIS    D 745
$! CONFLICTS WITH THE DOCUMENTATION IN THAT THE DEFAULT REEL NUMBER FOR D 745
$! PBT'S NEED NOT NECESSARILY BE ZERO.                                D 745
$! NATAL U. PATCH 745                                                D 745
$! MODIFIED FOR TSSMCP BY DR, UCSC JAN. 17, 1977                  D 745
      RDCTABLE[U]+P(DUP,LOD)&REEL[14:3:10]                 37046177 D 745
      RCDATE[24:31:17]&CYCLE[41:41:7];                   37046178 D 745
      REFL+RDCTABLE[U].[14:10]; % GET ACTUAL REFL NUMBER 38247100 D 745
$!*****                                                                D 745

```

```

$# PATCH 749 FOR TSSMCP CONTAINS 1 CARD. MEMORY PARITY ERR CAUSES "SD" D 749

$! THIS PATCH WILL CAUSE THE MCP TO HOLD ON TO A PSFUDO=DECK OF A JOB  D 749
$! WHICH HAS BEEN DS=ED DUE TO A MEMORY PARITY. AFTER ALL, THAT WASN=T  D 749
$! HIS FAULT.                                                         D 749
$! MODIFIED 6/15/74 - SD BIT SET IN TERMINALMFSSAGA INSTEAD OF      D 749
$! IN OUTER BLOCK.                                                    D 749
$! DREXEL U. PATCH 749 (BROCK U. PATCH 749 IS THE UNMODIFIED VERSION) D 749

```

```

IF N=32 THEN JAR[P1MIX,6],[1:1]+1] % MEM PAR 02206100 D 749
$!*****
S# PATCH 753 FOR TSSMCP.XV.3 CONTAINS 2 CARDS. TIME(-6) TO GET MIX INDEX D 753
S: THIS ADDS A NEW TIME FUNCTION. TIME(-6) WILL RETURN THE MIX INDEX OF D 753
S: THE CALLING PROGRAM AS AN INTEGER. D 753
S: NATAL U. PATCH 753 D 753
    WDI 14+WEEKDAY; GO TO INITIATE; 18711500 D 753
    IF 14 = (-6) THEN 14+P1MIX; 18711690 D 753
$!*****
S# PATCH 757 FOR TSSMCP.XVI.0 CONTAINS 12 CARDS. REMOVE UNLOAD FILES AT END D 757
S: THIS PATCH CAUSES THE FILES SPECIFIED TO BE UNLOADED TO BE REMOVED D 757
S: ONLY AFTER THEY HAVE ALL BEEN DUMPED. ALSO IF A DISK PARITY OCCURS D 757
S: WHILE DUMPING THEN THAT FILE WILL NOT BE REMOVED. D 757
S: NATAL U. PATCH 757 D 757
    M[FINFO+J],[R11]=0] % DONT REMOVE IF UNLOAD 28093900 D 757
    P(M[FFA+1],M[FA]); % SAVE NAME FOR UNLOAD 28094500 D 757
    NT1=P; M[FA]=NT1; M[FA+1]=P(XCH); % REPLACE NAMES 28097100 D 757
S VOIDT 28265801 28265600 D 757
    TMP:=FASZ DIV 2 - 1; 28307700 D 757
    FOR J:=0 STEP 1 UNTIL TMP DO 28307705 D 757
        IF (DAI=M[FINFO+J]).[CF]=18 THEN % FROM DISK 28307710 D 757
            IF UNLOAD THEN 28307715 D 757
                BEGIN MFID:=M[FA+J*2]; FID:=M[FA+1+J*2]; 28307720 D 757
                    P(DIRECTORYSEARCH(=MFID&1[3:47:1],FID,7),DEL); 28307725 D 757
                    IF DSEF THEN GO TO INITIATE; 28307730 D 757
                FND; 28307735 D 757
$!*****
S# PATCH 764 FOR TSSMCP CONTAINS 3 CARDS. TAPES IN USE AFTER FM OR IL 00000100 D 764
S: THIS PATCH PREVENTS TAPE UNITS FROM BEING MARKED AS IN USE. THIS 00000200 D 764
S: OCCURS AFTER AN "IL" TO A TAPE UNIT IN RESPONSE TO A "#NO FIL ON 00000300 D 764
S: DISK" OR AN "FM" TO A TAPE UNIT CONTAINING A TAPE WITH NO WRITE RING 00000400 D 764
S: IN RESPONSE TO A "#FM ROD". THE TAPE WILL REMAIN IN USE UNTIL A H/L 00000500 D 764
S: OR THE UNIT IS CLEARED BY A "CL". 00000600 D 764
S: HERIOT-WATT U. PATCH 764 00000700 D 764
S: MODIFIED FOR TSSMCP BY DR. UCSC FEB. 3, 1977 D 764
    END ELSE LABELTABLE[T2]=>(*P(DUP)); 06463740 D 764
    IF PRNTABLE[U],[1:1] THEN ELSE 37053300 D 764
        BEGIN LABELTABLE[U]=>(*P(DUP));GO TO SOMEWHERE;FND; 37053400 D 764
$!*****

```

```

$# PATCH 791 FOR TSSMCP CONTAINS 4 CARDS, TYPE PBCOUNT AT HALT=LOAD D 791
$! PATCH TO TELL THE OPERATOR HOW MANY BACK-UP FILES THERE ARE ON DISK, 00000200 D 791
$! REGARDLESS OF THE SETTING OF "AUTOPRNT", 00000300 D 791
$! THE FIRST & THIRD CARDS REMOVE THE DECLARATION FOR THE UNUSED LOCAL "L" D 791
$! AND VOIDS SOME UNUSED CODE REFERENCING "L" TO SPOUT THE "PBCOUNT" MESSAGE, D 791
$! BY NILS OTTE, UNIVERSITY OF NATAL, SEPT., 1976 00000600 D 791
$! HERIOT-WATT U. PATCH 791 00000700 D 791
$! MODIFIED FOR TSSMCP BY DR. UCSC FEB, 8, 1977 D 791
REAL NEXTLINK, AD, X, K, SEVEN7, FORTY, EUSU; 40004500 D 791
      IF PBCOUNT > 0 THEN % TELL OPERATOR 40320100 D 791
      END; SPOUT(X); 40320600 D 791
$ VOIDT 40338000 D 791
$!***** D 791

```

```

$# PATCH 801 FOR TSSMCP CONTAINS 39 CARDS. MULTI KEYIN MESSAGES D 801
REAL BUFF, KTR, TYPE, MIX, A, I, J, K, T, 16951600 D 801
LABEL RESTART; 16951950 D 801
RESTART; 16954450 D 801
TYPE = KEYINSCAN(KTR, MIX); 16954502 D 801
IF (PROCFD=TYPE, [1:5])=1 AND ((I=TYPE, [CF])=8 OR I=VCC 16954504 D 801
OR I=33 OR I=34) OR (PROCFD=0 AND I=1) THEN ELSE % SS, CC, OC, FE, AX 16954506 D 801
BEGIN 16954508 D 801
  STREAM(KTR, I); 16954510 D 801
  BEGIN SI=KTR; 16954512 D 801
    B(60(IF SC="" THEN 16954514 D 801
      BEGIN LI SI+SI+1; 16954516 D 801
        63(SI+SI+1) IF SC="" THEN JUMP OUT; 16954518 D 801
          IF SC="" THEN JUMP OUT 3 TO YECH); 16954520 D 801
        SI=SI+1; 16954522 D 801
        IF SC="" THEN GO TO LI 16954524 D 801
      END; 16954526 D 801
      IF SC="" THEN JUMP OUT 2 TO YECH; 16954528 D 801
      IF SC=";" THEN 16954530 D 801
        BEGIN 16954532 D 801
          I+SI; TALLY+1; 16954534 D 801
          DI+I; DS=LIT "+"; 16954536 D 801
          X; SI+SI+1; IF SC=";" THEN GO TO X; I+SI; 16954538 D 801
          JUMP OUT 2 TO YECH; 16954540 D 801
        END; SI=SI+1)); 16954542 D 801
        YECH; KTR=TALLY; 16954544 D 801
      END STREAM; 16954546 D 801
      I+P; IF P THEN 16954548 D 801
      STREAM(I, T+T+GETSPACE(62, 0, 0)+3); 16954550 D 801
      BEGIN 16954552 D 801
        SI+I; 16954554 D 801
        B(60(IF SC="" THEN JUMP OUT 2 TO L FLSE DS+CHR)); LI DS=LIT "+"; 16954556 D 801
      END STREAM; 16954558 D 801
      END CHECK FOR KEYIN RECYCLE; 16954560 D 801
      IF PROCFD=7 THEN GO TO TBLERR; 16954600 D 801
      IF T>0 THEN % ANOTHER MESSAGE 16969710 D 801
      BEGIN 16969720 D 801
        BUFF+T; T=0; 16969730 D 801

```

```

GO RSTART;
END;
$! DREXEL U. PATCH 801          9/23/73
$! THIS PATCH WILL ALLOW MORE THAN ONE COMMAND IN EACH KEYBOARD MESSAGE
$! SEPARATED BY SEMICOLONS, AS SUCH:
$! MX;TS;BL;CC REMOVE A/B
$! NOTE: NO KEYIN COMMANDS MAY BE CONTINUED AFTER A CONTROL CARD IS
$! FOUND.
$! MODIFIED 11/10/73 - CHAINED MESSAGES IGNORED FOR FE, OC, AX.
$! MODIFIED 9/8/74
$! THIS PATCH INHIBITS THE RUNNING OF KEYIN PROCEDURE "KEYIN2" IF THE RED
$! IS GETTING FULL.
$! IF THE OPERATOR (OR A REMOTE USER) KEYS IN A MESSAGE CAUSING
$! INDEPENDENT RUNNER KEYIN2 TO RUN, AND THE RED IS WITHIN 4 ENTRIES OF
$! FULL, A "WAIT...(KEYIN MESSAGE)" WILL BE SPOUTED. (NOT NEEDED IN TSSMCP)
$! MODIFIED 10/5/75 - FIX PROBLEM WHERE IF INPUT MESSAGE IS LONGER THAN
$! 480 CHARACTERS, A * WOULD BE PUT IN THE FIRST CHARACTER OF THE
$! NEXT MEMORY LINK, LEADING TO GRAVE PROBLEMS.
$! ALSO, SUCCESSIVE SEMICOLONS ARE IGNORED.
$! MODIFIED FOR TSSMCP BY DR, UCSC FEB. 7, 1977
$!*****

```

```

16969740 D 801
16969750 D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801
D 801

```

```

$# PATCH 815 FOR TSSMCP CONTAINS 17 CARDS. CLEAR IN USE FILE D 815

```

```

$! DREXEL U. PATCH 815      2/10/74
$! THIS PATCH WILL ENABLE THE OPERATOR TO CLEAR AN IN=USE FILE BY
$! DOING A "Q RESET FILE A/B". IT SHOULD BE USED CAUTIOUSLY SINCE
$! DIRE CONSEQUENCES COULD RESULT FROM "CLOSING" A TRULY "IN=USE"
$! FILE. IT IS USEFUL FOR THOSE STUCK IN USE PRD FILES AND SUCH.
$! MODIFIED 10/27/75 - REPEAT MCP FILE IN USE MESSAGE ABOUT EVERY 2 MIN.
$! MODIFIED FOR MARK XVI BY DR, UCSC FEB. 11, 1977
$! SEE ALSO UCSC PATCH 519
    WHILE NOT TOG DO
        COMPLEXSLFEP((CLOCK AND @17777)=0 OR TOG);
        SLFEP([CLOCK],NOT CLOCK);
BOOLEAN FT=N; DEFINE FH(FH1)=M[1+FH1]; * RESET FILE A/B
SUBROUTINE CLEARTHEFILE; * CLEAR AN IN=USE FILE
BEGIN
    FH[4],[01106]+0; * EXCLUSIVE
    FH[4],[16120]+0; * OPEN COUNT 2
    FH[9],[01128]+0; * TOGS & OPEN COUNT 1
    DISKWAIT(T,[CF],30,T,[FF]); * FIX IT
    FILEHOLD(CMM[2],CMM[3],0,T,0); * WAKE UP WAITING PROCESSES
    LBMESS(CMM[2],CMM[3],11,26,0,SPROUTUNIT,1);
END CLEARTHEFILE;
    IF CN#ACCESSD THEN
        IF NOT (FT.FXTOG+(CN=FILEV)) THEN GO TO CCERR;
$ VOIDT
        END ELSE IF FT THEN CLEARTHEFILE ELSE BEGIN
$!*****

```

```

18031400 D 815
18032000 D 815
18032010 D 815
20704100 D 815
20704200 D 815
20704210 D 815
20704220 D 815
20704230 D 815
20704240 D 815
20704250 D 815
20704260 D 815
20704270 D 815
20704280 D 815
20714000 D 815
20714500 D 815
20768000 D 815
20769000 D 815
D 815

```

```

$#PATCH NUMBER 846 FOR TSSMCP CONTAINS 7 CARDS. DUMMY FILE D 846

```

```

IF 10D.(3:5)=30 THEN GO RETURN; % SPO
FORM          = 75#,% SWITCH D(PCC)"FORM"="SPECIAL"
LDUMMY,
LDUMMY,
LDUMMY;      TPNO+11; % " FORM SPO" = DUMMY FILE
"FORM " ,    75 ,    % SWITCH D(PCC)
"DUMMY " ,    76 ,
04105998 D 846
20240000 D 846
20392870 D 846
20394070 D 846
20451900 D 846
41485600 D 846
41485630 D 846
$!
$! THIS IS THE MCP PORTION OF THE DUMMY FILE PATCH, IT ALSO REQUIRES A
$! PATCH TO THE INTRINSICS. IT PROVIDES THE WORD "DUMMY" AS A UNIT
$! TYPE: OUTPUT TO "DUMMY" WILL CREATE A "SPECIAL FORM SPO FILE",
$! WHICH THE INTRINSICS WILL IGNORE.
$! FROM DREXEL U. PATCH 846 JHH AT UCSC
$!*****

```

```

$#PATCH 890 FOR TSSMCP CONTAINS 173 CARDS = DATA02q/EBCDIC D 890
$!AUG 1975 HERIOT-WATT UNIVERSITY * LOCAL ONLY* D 890
$!AND PERFORMS EBCDIC TO BCL CONVERSION D 890
$!IMPLEMENTS DATA029 CARD D 890
$!WAS PATCHES 900-902 IN MARK XV.1. *** NB: SEQUENCE NUMBERS CHANGED *** D 890
$! D 890
PROCEDURE FRTABLE; 07003270 D 890
BEGIN LABEL L; 07003280 D 890
P(O,XIT,L,DEL); 07003290 D 890
L:; 07003300 D 890
0," 32?","1???", "0TS?", "/???", "-LK?", "J???", "???", "???", "&CB?", "A???", 07003310 D 890
"#???", "???", "???", "???", "???", "???", 07003312 D 890
"9???", "???", "Z???", "???", "R???", "???", "???", "???", "I???", "???", 07003320 D 890
"???", "???", "???", "???", "???", "???", 07003322 D 890
"8#1?", "???", "Y,x?", "???", "Q$]?", "???", "???", "???", "H.1?", "???", 07003330 D 890
"???", "???", "???", "???", "???", "???", 07003332 D 890
@07770675,@05370413,@067146616,@65176473, 07003340 D 890
@47574656,@45554453,"???", "???", @27742620,@25352436, 07003342 D 890
$! %Q", "???", "???", "???", "???", "???", 07003344 D 890
%QMARKADDRESSES 07003350 D 890
"???", "???", "???", "???", "???", "???", "???", "???", "???", "???", 07003352 D 890
"???", "???", "???", "???", "???", "???", 07003354 D 890
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, % SPACE FOR ADDRESSES 07003360 D 890
0,0,0,0,0,0,0; 07003370 D 890
END; 07003380 D 890
REAL PROCEDURE SETUPERTABLE; 07003400 D 890
BEGIN REAL ADR; LABEL EXIT; 07003410 D 890
DEFINE STORFADDRESS= 07003412 D 890
XI=SI; SI=LOC XI; SI:=SI+5; 07003414 D 890
DS:=3 CHR; DI:=DI+5 #; 07003416 D 890
ERTABLE; 07003420 D 890
P([M[(ADR:=P(.FRTABLE) INX NOT 1)]]],IOR); 07003430 D 890
SETUPERTABLE:=ADR:=ADR.[CF]+3; 07003440 D 890
IF M[ADR+8] = 0 THEN 07003450 D 890
STREAM (XI=0, TABLE:=ADR, QMARKTABLEADDRESS:=ADR+32, LOCATIONS:=ADR+40); 07003462 D 890
BEGIN 07003464 D 890
DI:=TABLE; SI:=TABLE; SI:=SI+8; 07003466 D 890

```

```

2(40(SI:=SI+4; DS:=4 CHR)))
  %FIRST FILL ALL OF ADDRESSES WITH ADDRESS OF ?S
SI:=QMARKTABLEADDRESS;
X:=SI;
SI:=LOC X; SI:=SI+5; %POINT SI AT ADDRESS OF QMARKS
DI:=LOCATIONS;
DS:=3 CHR;
SI:=LOCATIONS;
DS:=63 WDS;
  %NOW SET ADDRESSES FOR VALID CHARS INTO EBCDIC TABLE
SI:=TABLE;
DI:=LOCATIONS;
3(STOREADDRESS; %STORE ADDRESSES FOR LP=0,1,2
  %CORRESPONDING TO BLANK,9 HOLE,8 HOLE
SI:=X;
SI:=SI+63; SI:=SI+1); %SKIP 8 WORDS DOWN TABLE
DI:=DI+8; %LP=3 IS INVALID
STOREADDRESS; %LP=4.....7 HOLE
DI:=DI+8; %LP=5 IS INVALID
SI:=X;
SI:=SI+1; %SKIP 1 CHAR DOWN TABLE
STOREADDRESS; %LP=6.....7&8 HOLES TOGETHER
DI:=DI+8;
SI:=X;
SI:=SI+1;
STOREADDRESS; %LP=8.....6 HOLE
DI:=DI+8;
SI:=X;
SI:=SI+1;
STOREADDRESS; %LP=10.....6&8 HOLES TOGETHER
DI:=DI+40; %MISS 5 WORDS...LP=11,12,13,14,15
SI:=X;
SI:=SI+1;
STOREADDRESS; %LP=16.....5 HOLE
DI:=DI+8;
SI:=X;
SI:=SI+1;
STOREADDRESS; %LP=18.....5&8 HOLES TOGETHER
2(DI:=DI+52); %MISS 13 WORDS...LP=19-31
SI:=X;
SI:=SI+1;
STOREADDRESS; %LP=32.....4 HOLE
DI:=DI+8;
SI:=X;
SI:=SI+1;
STOREADDRESS; %LP=34.....4&8 HOLES TOGETHER
END;
END;
*****
STREAM PROCEDURE EBCDICCONVERT(INTO, TABLE, POINTERS);
VALUE INTO, TABLE, POINTERS;
*****
BEGIN
LOCAL HP, LP, SRCE, DEST, HPPTR, LPPTR;
  %POINT HPPTR & LPPTR TO LAST CHAR OF HP, LP
SI:=LOC HP;
SI:=SI+7;

```

```

07003468 D 890
07003482 D 890
07003484 D 890
07003486 D 890
07003488 D 890
07003490 D 890
07003492 D 890
07003494 D 890
07003496 D 890
07003498 D 890
07003500 D 890
07003502 D 890
07003504 D 890
07003506 D 890
07003508 D 890
07003510 D 890
07003512 D 890
07003514 D 890
07003516 D 890
07003518 D 890
07003520 D 890
07003522 D 890
07003524 D 890
07003526 D 890
07003528 D 890
07003530 D 890
07003532 D 890
07003534 D 890
07003536 D 890
07003538 D 890
07003540 D 890
07003542 D 890
07003544 D 890
07003546 D 890
07003548 D 890
07003550 D 890
07003552 D 890
07003554 D 890
07003556 D 890
07003558 D 890
07003560 D 890
07003562 D 890
07003564 D 890
07003566 D 890
07003568 D 890
07003570 D 890
07003580 D 890
07003590 D 890
07003600 D 890
07003610 D 890
07003615 D 890
07003620 D 890
07003630 D 890
07003640 D 890
07003650 D 890
07003660 D 890
07003670 D 890

```



```

HPPTR:=SI;    %HIGH PART
SI:=LOC LP;
SI:=SI+7;
LPPTR:=SI;    %LOW PART
SI:=INTO;  SI:=SI+8;
DI:=INTO;
  %START CHARACTER TRANSLATE LOOP
2(40(
  DEST:=DI;
  %TRANSFER LOW & HIGH PARTS
DI:=HPPTR;
DS:=1 CHR;
DI:=LPPTR;
DS:=1 CHR;
SRCE:=SI;    %STORE SI FOR NEXT PASS THRU LOOP
  %NOW FIND THE POINTER INTO TABLE APPROPRIATE TO I.P
SI:=POINTERS;
LP(SI:=SI+8);
SI:=SC;    %SI NOW POINTS INTO TABLE @ POINT DEPENDANT ON I.P
DI:=DFST;
SI:=SI+HP;  %SKIP SI THROUGH TABLE HP CHARS
DS:=CHR;
SI:=SRCE);
DI:=TABLE; DI:=DI+3;    %POINT TO QMRK
END CONVERT;
BOOLEAN EBCDIC; REAL ERTABLEADR;
  IF EBCDIC THEN
  BEGIN
    EBCDIC:=FALSE;
    M(*P(,EBTABLE) INX NOT 1),[2:1]:=0;
  END;
  STREAM(X:="PACKETS";Y:="CONTINU";Z:="END.  ",
    EB:="DATA029",INBUFF);
  INBUFF:=SI;
  FLSE BEGIN SI:=INBUFF;DI:=LOC Y;
    SI:=INRUFF;
    IF 3 SC=DC THEN TALLY:=7
    ELSE BEGIN
      DI:=LOC EB;DI:=DI+1;
      SI:=INBUFF;
      IF 6 SC=DC THEN
        TALLY:=4;
    END;
  END;
  END;
  END;
  END;
  IF PTYPE = 4 THEN
  BEGIN PTYPE:=0; EBCDIC:=TRUE;
    EBTABLEADR:=SETUPEBTABLE;
  END;
  ELSE BEGIN WHILE(Q:=WAITIO(@40000000+INBUFF+
    FBCDIC@400000001,FIRST*4+
    IF EBCDIC THEN EBCDICCONVERT(INBUFF,
      EBTABLEADR,FBTABLEADR+40) ELSE
    STREAM(QMK:=12;BCL:=1-EBCDIC,INBUFF);
  BEGIN
    SI:=INBUFF;
    BCL (IF SC="" THEN JUMP OUT TO L1);

```

```

07003680 D 890
07003690 D 890
07003700 D 890
07003710 D 890
07003720 D 890
07003730 D 890
07003740 D 890
07003750 D 890
07003760 D 890
07003770 D 890
07003780 D 890
07003790 D 890
07003800 D 890
07003810 D 890
07003820 D 890
07003830 D 890
07003840 D 890
07003860 D 890
07003870 D 890
07003880 D 890
07003890 D 890
07003900 D 890
07003910 D 890
07003920 D 890
07003950 D 890
07009200 D 890
07031200 D 890
07031300 D 890
07031400 D 890
07031500 D 890
07031600 D 890
07032000 D 890
07032100 D 890
07034500 D 890
07036220 D 890
07036250 D 890
07036260 D 890
07036270 D 890
07036280 D 890
07036290 D 890
07036300 D 890
07036310 D 890
07036320 D 890
07036330 D 890
07036390 D 890
07039500 D 890
07039600 D 890
07039700 D 890
07039800 D 890
07059000 D 890
07059010 D 890
07060100 D 890
07060200 D 890
07066100 D 890
07066200 D 890
07066300 D 890
07066350 D 890

```

```

IF SC=#14 THEN                                07066400 D 890
L1: BEGIN SI:=LOC QMK;SI:=SI+7;                07066500 D 890
      DS:=1 CHR;                               07066550 D 890
      TALLY:=1;                                07066600 D 890
SI:=INBUFF;                                    07066610 D 890
2(36(IF SC=">" THEN                            07066620 D 890
  BEGIN INBUFF:=SI; DI:=INBUFF;              07066630 D 890
  DS:=LIT "="                                07066635 D 890
  END)
  IF SC=">" THEN                              07066640 D 890
  BEGIN INBUFF:=SI; DI:=INBUFF;              07066650 D 890
  DS:= LIT ""                                07066660 D 890
  END) SI:=SI+1)))                            07066665 D 890
END)                                           07066670 D 890
QMK      I=TALLY;                             07066670 D 890
END)                                           07066800 D 890
QI=P OR Q;                                    07066850 D 890
                                              07066900 D 890
$ VOIDT                                       07086000 D 890
      INBUFF := GETSPACE (21,0,1) + 2;        07101000 D 890
      STRFAM(CARDLOC);                         20374401 D 890
      BEGIN SI=CARDLOC;                       20374402 D 890
      2(36(IF SC=">" THEN                     20374403 D 890
        BEGIN CARDLOC+SI;DI=CARDI.OC;DS=LIT "=" END)
        IF SC=">" THEN                       20374404 D 890
        BEGIN CARDLOC+SI;DI=CARDI.OC;DS=LIT "" END)
        SI+SI+1)))                           20374405 D 890
      END)                                     20374406 D 890
      (IF KOUNT = "" THEN EQUAL FLSF % THIS IS AS IN SYMBOL
"DATA029", 41 ,                             20374407 D 890
                                              20374408 D 890
                                              20380000 D 890
                                              41483010 D 890

```

```

$# PATCH 891 FOR TSSMCP XV1.0 CONTAINS 8 CARDS. CORRECT 890 (DATA029) D 891
$# CORRECTS A MISPUNCH IN PATCH 890 (900) WHERE EBTABLE WAS SET UP WRONGLY D 891
$# DEREK BRUCE .. MAY 1976 .. H.W.U. D 891
$# D 891
07003250 D 891
#####07003251 D 891
07003252 D 891
"????","????","????","????","????","????", 07003344 D 891
#####07003390 D 891
#####07003391 D 891
07003392 D 891
      INBUFF + GETSPACE( 21,0,1 ) + 2; 07101000 D 891

```

```

$#PATCH 892 FOR TSSMCP CONTAINS 2 CARDS. CORRECT 890,891 C 892 0000000/CARD
@07770675,@05770413,@067146616,@65176473, 07003340 C 892
      INRUFF + GETSPACE(21,10RUFFRAREAV,1) + 2; 07101000 C 892
$# THIS PATCH MODIFIES THE DATA029 FEATURE (PATCH 890) SO THAT C 892
$# SINGLE QUOTE COMING IN IS CHANGED TO DOUBLE QUOTE. FOR THE FORTRAN CATS C 892
$# BY JHH, UCSC, 11 AUG 77 C 892

```

\$: MODIFIED 19 AUG 77 TO USE ARFA TYPING INTRODUCED IN PATCH 167	C	892	
\$#PATCH 892 FOR TSSMCP CONTAINS 2 CARDS. CORRECT 890,891	D	892	PATCHES/TSSMCP DISCARDED ###
\$# PATCH 901 FOR TSSMCP CONTAINS 1 CARD. TERMINAL LOG-IN MESSAGE #####	D	901	
\$: CHANGE THIS DATE EACH TIME ANY OTHER PATCHES ARE CHANGED. ++++++ LOOK <<<<<<	D	901	
\$: BY TJP, UCSC MODIFIED BY DR, UCSC	D	901	
DS+35LIT"=+UCSC B5700 TSSMCP OF 24 AUG 77, ";	D	901	03595560
\$!*****	D	901	
\$# PATCH NUMBER 902 FOR TSSMCP CONTAINS 3 CARDS. AUTORN OPTION	D	902	
AUTORN = OPTION.[17:1]#,		00416640	D 902
"AUTO", "RNOO", %17%		41434800	D 902
IF AUTORN THEN BEGIN RUNNUMBER:=4; STARTADECK(0); END;		44424100	D 902
\$:			D 902
\$: 3 CARDS DELETE THE RNALL OPTION USED WITH SHAREDISK ONLY AND			D 902
\$: REPLACE IT WITH THE NEW AUTORN OPTION WHICH WILL AUTOMATICALLY			D 902
\$: START PSEUDO READERS WHEN THE SYSTEM IS STARTED UP			D 902
\$: BY JHH MARK XVI.0			D 902
\$: SEE ALSO BROCK U. PATCH 571 FOR TSSMCP			D 902
\$!*****			D 902
\$#PATCH NUMBER 903 FOR TSSMCP CONTAINS 2 CARDS	D	903	
IF LO=23 THEN BEGIN LO:=27; HI:=28; FND ELSE		37195375	D 903
IF LO=27 THEN BEGIN LO:=0; HI:=15; END ELSE GO TO DUN;		37195400	D 903
\$: THIS PERMITS PROC. FINDINPUT TO FIND UNIT PRA. OBSERVE THAT PRA			D 903
\$: IS EXPECTED TO HAVE REEL NUMBER 1 MOUNTED, UNLESS SOME OTHER REEL			D 903
\$: NUMBER IS DESIGNATED EXPLICITLY. JHH&DR AUG. 24, 1976			D 903
\$!*****			D 903
\$# PATCH 905 FOR TSSMCP.XVI.0 CONTAINS 1 CARD. DELFTE UNUSED LABEL DECLARATION	D	905	
\$: BY DR, UCSC, OCT. 4, 1976			D 905
LABFL PX)		38660000	D 905
\$!*****			D 905
\$# PATCH NUMBER 917 FOR TSSMCP.XVI.00 CONTAINS 07 CARDS. "PBT ON MTX RW/L (PRN".	D	917	

```

STREAM(MTX+TINU[UNIT],PRN+PRNTARLF[UNIT],[30:18],%517=GRK 12530010 D 917
      D+T+SPACE(10)); %517=GRK 12530020 D 917
BEGIN SI+LOC MTX; SI+SI+5; DS+8LIT"#PBT ON ";%51712530030 D 917
      DS+3CHR; DS+7LIT" RW/L "; DS+5DEC; %517 12530040 D 917
      DS+2LIT")"; DI+DI-7; DS+4FILL; %517=GRK 12530050 D 917
      END STREAM; %517=GRK 12530060 D 917
      SPOUT(T); %517=GRK 12530070 D 917
$; D 917
$; THIS PATCH WILL SPOUT A RW/L MESSAGE WHEN A PBT IS LOCKED BY COM19 D 917
$; TO MAKE IT EASIER TO DETERMINE LATER WHICH TAPE WAS ACTUALLY D 917
$; PRINTED. D 917
$; D 917
$; G.R.KENNEDY 21MAR75. BROCK U. PATCH 517 D 917
$; SEE ALSO UCSC TSSMCP PATCH 940 = BROCK U. PATCH 540 D 917
$;***** D 917

$# PATCH NUMBER 950 FOR TSSMCP.XVI.00 CONTAINS 03 CARDS.CLEANUP CANDE STUFF.GRK. D 950

$ SET OMIT %SETUP SHORT/LONG CARRIAGE MESSAGE %550=GRK 05751400 D 950
$ POP OMIT %550=GRK 05754600 D 950
%*% SPOUTIT(T,CHRGK+1); %SHORT/LONG CARRIAGE MSG %550=GRK 05756500 D 950
$; D 950
$; THIS PATCH CLEANS UP SOME OF THE CANDE-RELATED MESSAGES A BIT. D 950
$; 1. DONT BOTHER BUILDING/SPOUTING THE "SHORT/LONG CARRIAGE" MESSAGE. D 950
$; D 950
$; G.R.KENNEDY 08MAR76. BROCK U. PATCH 550 D 950
$;***** D 950

$# PATCH 991 FOR TSSMCP CONTAINS 9 CARDS. SCHEDULER CHANGE D 991

$; THIS PATCH ATTEMPTS TO IMPROVE EFFICIENCY OF TSSMCP BY D 991
$; REDUCING UNNECESSARY SWAPPING WITHOUT HURTING TERMINAL JOBS. D 991
$; 1. THE TIME SLICE IS MADE MUCH LONGER FOR JOBS THAT USE A LOT D 991
$; OF CORE. D 991
$; 2. BATCH JOBS WHICH USE UP THEIR TIME SLICE ARE SWAPPED OUT D 991
$; WITH "TIMEND" AS BEFORE, BUT TERMINAL JOBS ARE SWAPPED OUT D 991
$; AT "FORCEND", GIVING THEM PRECEDENCE OVER THE "TIMEND" JOBS. D 991
$; 3. BATCH JOBS WHICH ARE FORCED OUT ARE SWAPPED OUT WITH "TIMEND" D 991
$; RATHER THAN AT "FORCEND". D 991
$; D 991
      NT1+(SLN[MIX]*90 + COUNT[MIX]*14 - NT1)*8 + 208; 21111540 D 991
      IF PRTR0W[MIX],[PSF]=3 THEN PRTR0W[MIX],[PSF]=0; %UCSC KLUDGE 21212010 D 991
      IF NT3=2 THEN STOPM ELSE SWAP( 48030000 D 991
      IF LOGLINE,[33:7]#0 THEN 48030010 D 991
      IF NOT SCHEDLINE[LOGLINE,[40:8]] THEN 48030015 D 991
      FORCESWAP ELSE TIMEND ELSE TIMEND,1); 48030020 D 991
      REGIN SWAP(IF LOGLINE,[33:7]#0 THEN 48038400 D 991
      IF NOT SCHEDLINE[LOGLINE,[40:8]] THEN FORCESWAP 48038405 D 991
      ELSE TIMEND ELSE TIMEND,1); GO TO RETURN END; 48038410 D 991
$; D 991

```

\$: BY J.H., UCSC, 7 APR 77; REVISED 16 APR 77 REVISED 28 APR 77 D 991  
\$!\*\*\*\*\* n 991

\$# PATCH 992 FOR TSSMCP CONTAINS 4 CARDS, LABEL THE HIGH-SPEED PAPER TAPE RDR D 992

\$: AUTOMATICALLY ASSIGN A LABEL TO THE HIGH-SPEED PAPER TAPE READER PORT D 992  
\$: WHENEVER IT GOES READY n 992  
\$: BY DR, UCSC FEB, 10, 1977 D 992  
\$: VOIDT D 992  
PAPER: LABELTABLE[U] ← MULTITABLE[U] ← "PRA "; 22199000 D 992  
RDC:TABLE[U] ← @100000000; % REEL NR: = 1 22201200 n 992  
GO TO COMMON; 22201400 n 992  
22201600 n 992  
\$!\*\*\*\*\* n 992

\$# PATCH 993 FOR TSSMCP CONTAINS 2 CARDS, AUTO-RY OF SCHEDULE LINES AT H/L D 993

\$: BY DR, UCSC FEB, 10, 1977 n 993  
FOR NT1←1 STEP 1 UNTIL LMAX DO 44430000 n 993  
IF SCHEDULE[NT1] THEN SCHEDULEBUSY[NT1] ← 0; 44430100 D 993  
\$!\*\*\*\*\* n 993

\$# PATCH 995 FOR TSSMCP CONTAINS 2 CARDS, DUPLICATE FORWARD REFERENCES 00000100 D 995

\$: WHERE THERE ARE 2 OR MORE FORWARD REFERENCES TO THE SAME 00000200 D 995  
\$: PROCEDURE, THIS PATCH DELETES ALL BUT THE 1ST FORWARD REFERENCE. 00000300 D 995  
\$: BY DR, UCSC FEB, 3, 1977 00000400 D 995  
\$: LINE 00364000 = LINE 06020500 00000500 D 995  
\$: LINE 00431500 = LINE 03099000 00000600 n 995  
\$: VOIDT 03099000 n 995  
\$: VOIDT 06020500 D 995  
\$!\*\*\*\*\* n 995

\$# PATCH NUMBER 996 FOR TSSMCP CONTAINS 1 CARD. AUTO-LOGOUT AT MIDNIGHT D 996

DEFINE MIDNIGHT = BEGIN LOGOUT; XCLOCK←XCLOCK-WITCHINGHOUR; 02393200 D 996  
\$: n 996  
\$: THIS AUTOMATICALLY CHANGES THE LOG FILE AT MIDNIGHT, SO THAT n 996  
\$: EACH LOG FILE CONTAINS ONLY ONE DAY-S WORTH OF LOGGING, TO D 996  
\$: FACILITATE ANALYSIS. D 996  
\$: BY JH, UCSC, 01/19/77 D 996  
\$!\*\*\*\*\* n 996

\$#PATCH NUMBER 998 FOR TSSMCP CONTAINS 2 CARDS, MIXMAX,MAXLMAX D 998

```
DEFINE MIXMAX=10#1 00004000 D 998
DEFINE MAXI,MAX=20 #1 * SHUD BE ABT 8 + ACTUAL MAX LINES 00004500 D 998
$!***** D 998
```

\*\*\*\*\* CONFLICTS \*\*\*\*\*

@07770675,@05770413,@067146616,@65176473,  
 @07770675,@05370413,@067146616,@65176473,

07003340 C 892 CONFLICTED WITH:  
 07003340 C 890 DISCARDED

INBUFF ← GETSPACE(21,IOBUFFRAREAV,1) + 2;  
 INBUFF ← GETSPACE( 21,0,1 ) + 2;

07101000 C 892 CONFLICTED WITH:  
 07101000 C 891 DISCARDED

\$ VOIDT

INBUFF := GETSPACE(11,IOBUFFRAREAV,1)+2;%

07086000 D 890 CONFLICTED WITH:  
 07086000 D 155 DISCARDED

INBUFF ← GETSPACE(21,IOBUFFRAREAV,1) + 2;  
 INBUFF := GETSPACE (21,0,1) + 2;

07101000 C 892 CONFLICTED WITH:  
 07101000 C 890 DISCARDED

LABEL PE,TF,PA,LL;  
 LABEL PE,TE,PA;

18701010 D 143 CONFLICTED WITH:  
 18701010 D 136 DISCARDED

SWITCH S := LL,PA,PF,TF,IT,US,D,TD,PR,IOT,TMR,AD,WDI;  
 SWITCH S := PA,PE,TF,IT,US,D,TD,PR,IOT,TMR,AD,WDI

18701100 D 143 CONFLICTED WITH:  
 18701100 D 136 DISCARDED

C1: IF (I4:=I4) ≥ (-6) AND I4 ≤ 6 THEN  
 C1: IF (I4:=I4) GEQ (-5) AND I4 LEQ 6 THEN

18710100 D 143 CONFLICTED WITH:  
 18710100 D 136 DISCARDED

BEGIN GO TO S[I4+6];  
 BEGIN GO TO S[I4+5];

18710200 D 143 CONFLICTED WITH:  
 18710200 D 136 DISCARDED

FORM = 75%, % SWITCH D(PCC)"FORM"="SPECIAL"  
 FORM = 76%, % SWITCH D(PCC)"FORM"="SPECIAL"  
 FORM = 77%, % SWITCH D(PCC)"FORM"="SPECIAL"

20240000 D 846 CONFLICTED WITH:  
 20240000 D 724 DISCARDED  
 20240000 D 603 DISCARDED

END ELSE IF FT THEN CLEARTHEFILE ELSE BEGIN  
 ELSE BEGIN

20769000 D 815 CONFLICTED WITH:  
 20769000 D 521 DISCARDED

H1:=FM[TYPEDSPACE(42,DISKHEADERAREAV))] & 30[8:38:10];  
 H1:=FM[SPACE(42)]&30[8:38:10];

28070000 D 155 CONFLICTED WITH:  
 28070000 D 119 DISCARDED

"FORM ", 75, % SWITCH D(PCC)  
 "FORM ", 76, % SWITCH D(PCC)  
 "FORM ", 77, % SWITCH D(PCC)

41485600 D 846 CONFLICTED WITH:  
 41485600 D 724 DISCARDED  
 41485600 D 603 DISCARDED

BEGIN SWAP(IF LOGLINE,[33:7]#0 THEN

48038400 D 991 CONFLICTED WITH:

FNDJ

48038400 D 154 DISCARDED









```

CURRLKCNTR = 16 # % %155=00067040 D 155
,AITNDX = 6 # % %155=00067050 D 155
,FTF = 18:18:15 # % %155=00067060 D 155
,FTC = 33:18:15 # % %155=00067070 D 155
,DFLTA = 11 # % %155=00067080 D 155
,
, MEMTOG = TOGLE,[43:1]#, MEMTOGMASK = @20# %155=00067999 D 155
, ITSALLBATCH = TOGLE,[16:1]# %202=00080500 D 202
IF [MEM[WAITSTOREF],0],[CF]=0 THEN %151=00081785 D 151
IF NOT MEMTOG THEN SLEEP([TOGLE],MEMTOGMASK) ELSE ELSE00093120 D 202
DEFINE STOREDY(STOREDY1,STOREDY2)= IF[MEM[STOREFDY1],0],[CF]=0 THEN %202=00093400 D 202
MEMTOG1=STOREFDY2 ELSE %202=00093410 D 202
MEM[STOREDY1,0],[17:1]=STOREDY2# %202=00093420 D 202
ARRAY WORKERSTACK[*]; % DESC. TO OLDWIERDHAROLD'S STACK. %158=00299500 D 158
ARRAY ESPTAB[*]; REAL ESPCOUNT; % %158=00399000 D 158
AUTORN = OPTION,[17:1]#, %902=00416640 D 902
BOOLEAN OKSPGZEROWRITE; %203=00422100 D 203
PROCEDURE FORMTIME(W,T); VALUE W,T; REAL W,T; FORWARD; %144=00480010 D 144
BEGIN DS:= 16 LIT "SYSTEM HANG, F="; %101=00652400 D 101
DEFINE PAGESIZE = 450#; % PACKET PAGE PBD SIZE %560=02113091 D 560
IF P THEN MAKELOG(MESSAGE,ABS(TYPEF)); %509=GRK 02134000 D 509
END; IF TYPE ISS 0 THEN TYPEI=64; %509=GRK 02134005 D 509
T := GETSPACE(S=1, SPOUTMSGAREAV+64, 0) + 1; %155=02134300 D 155
M[MESSAGE=1],[AREATYPEF] := SPOUTMSGAREAV; %155=02134710 D 155
STREAM(X+([M[MESSAGE=1],[CF]=1] & 7 [CTF])); %647=02142100 D 647
DS + LIT " "; %647=02142110 D 647
(S+2+(M[BUF INX Z],[1:5]#>"))[8:38:10]; 02173296 D 144
FORMTIME([T],XCLOCK+P(RTR)); %144=02173297 D 144
STREAM(N:=S-1,CL:=S*8-Y,AA:=RUF INX Z,BB := NT1,%144=02173300 D 144
RUF:=RUF[R]); %144=02173301 D 144
BEGIN DS := 7 LIT " "; SI := LOC BB; DS := 8 CHR; 02173305 D 144
DS := 9 LIT " "; SI := AA; %144=02173306 D 144
AI=TYPEDSPACE(15,SPOUTMSGAREAV); %155=02205475 D 155
IF N=32 THEN JAR[PIMIX,6],[1:1]+1; % MEM PAR %749=02206100 D 749
S := IF MIT,AREATYPEF = CODFAREAV THEN %155=02216000 D 155
MIT+1],[CF] ELSE 0; %155=02216010 D 155
ELSE P([MIT=2],[AREATYPEF,DUP] # TYPE7INTAREAV AND %155=02216600 D 155
P(XCH) # TYPE13INTAREAV) OR L = 0; %155=02216610 D 155
BEGIN %675=02261250 D 675
IF (LUN+L,[41:5])<16 THEN SLAPITOFF; %675=02261300 D 675
LUN+L,[46:2]+19; % LPA, LPB, OR CPA %675=02261350 D 675
SLAPITOFF; %675=02261400 D 675
SVOIDT 02261701 %675=02261450 D 675
BEGIN T+(A INX @5400000000000000)&17[8:38:10]; %140=02287240 D 140
IF SEPARATE THEN T+T&(LUN#22)[32:47:11] %140=02287245 D 140
ELSE T+T&(LUN#22)[28:47:11]; %140=02287250 D 140
DEFINE MIDNIGHT = BEGIN LOGOUT; XCLOCK:=XCLOCK-WITCHINGHOUR; %996=02393200 D 996
STOREDY(0,0); %202=02396130 D 202
STOREDY(0,1); %202=02396500 D 202
TP:=M[TA:=TYPEDSPACE(513,MDUMPARAV))] %155=02434430 D 155
& 513[SIZE] x 5[21:45:3]; %155=02434435 D 155
STOREDY(0,0); %202=02434490 D 202
STOREDY(0,1); %202=02434530 D 202
LS: IF SC # "" THEN IF SC # LEFTARROW THEN %141=02615100 D 141
BEGIN SI := SI + 1; GO TO LS; END; %141=02615200 D 141
IF SC = LEFTARROW THEN GO TO EXIT; %141=02615300 D 141
LRL := TYPEDSPACE(10,LABELAREAV); %155=02639000 D 155

```

```

X: CLK[0] :=
$ SET OMIT = NOT(NEWLOGGING)
IF LOGSTOPPED[P1MIX] THEN PROCTIME[P1MIX] ELSE
$ POP OMIT
PROCTIME[P1MIX] + CLOCK + P(RTR);
REAL PROCEDURE INPUTSCAN(MODE, SOURCE, DEST, NUM, FLAGS, SEQMODE);
VALUE NUM, FLAGS, SEQMODE;
REAL MODE, SOURCE, DEST, NUM, FLAGS, SEQMODE;
$ VOID
IF P2 > 0 THEN MOVF(15, P2, T+4) ELSE MET+4) * P2;
END
ELSE % ITS BELOW THE FENCE
IF ITSALLBATCH THEN % WE WILL WAIT AWHILE
BEGIN
TANKS[LL] := NARS(T);
CLICK := CLOCK + P(RTR) + 128; % 2 SEC.
SLEEP(0,0); % WAIT FOR TIME OUT.
GO TO START; % TRY AGAIN.
END;
REAL PROCEDURE INPUTSCAN (MODE, SOURCE, DEST, NUM, FLAGS, SEQMODE);
VALUE NUM, FLAGS, SEQMODE;
REAL MODE, SOURCE, DEST, NUM, FLAGS, SEQMODE;
U := (DEST.[30:3] & DEST.[30:33:15]) - (INPTSC.[30:3] & INPTSC.[30:33:15]);
IF U = 1 AND SEQMODE % U = 1 WHEN ONLY "+" LEFT IN BUF., AND IF SEQMODE
THEN % TRUE, THEN USER BACKSPACED TO FRONT OF BUFFER.
BEGIN
STREAM (A := 0 | DEST);
BEGIN DI := DI - 1; % MAKE INPUT LOOK LIKE A NULL LINE
DS := 2 LIT " +"; % (SPACE) TO PREVENT CANDE FROM
AI := DI; % RELIEVING THIS IS END OF SEQ MODE
END;
P([DEST], STD);
GO AUT;
END;
INPUTSCAN := U;
REAL RESULT = -4;
BOOLEAN ZEROTIMEOUT;
ZEROTIMEOUT := N.[32:1];
N.[32:1] := 0;
IF ZEROTIMEOUT THEN
BEGIN
RESULT := 2;
STREAM(DI = A, [CF]);
DS := 8 LIT "TIME3 ";
GO TO RETURN;
END;
IF P(0, RDS) < FENCE THEN % BELOW THE FENCE
BEGIN CLICK := CLOCK + P(RTR) + 1023 + 1023; % 34 SEC.
COMPLEXSIEFP(INPUTANK[T], INPUTN OR TFRMSET(P1MIX));
END ELSE
N1 := INPUTSCAN(MODE, SOURCE, A3, (MFA1).[FFF] - X, [2:8]), FLAGS, 0);
STREAM(TI = ((END) & [12])[24:42:16]) | A2);
BEGIN SI := A2; DI := LOC T; DI := DI + 4;
IF 4 SC NEQ Dc THEN TALLY := 1;
TI := TALLY;
END;
RESULT := POLISH;

```

```

%113-03067500 D 113
%113-03067600 D 113
%113-03067700 D 113
%113-03067800 D 113
%113-03067900 D 113
%159-03071000 D 159
%159-03072000 D 159
%159-03073000 D 159
%995-03099000 D 995
%108-03120000 D 108
%151-03147000 D 151
%151-03147100 D 151
%151-03147200 D 151
%151-03147300 D 151
%151-03147400 D 151
%151-03147500 D 151
%151-03147600 D 151
%151-03147700 D 151
%151-03147800 D 151
%159-03276800 D 159
%159-03277000 D 159
%159-03277200 D 159
%159-03290600 D 159
%159-03290700 D 159
%159-03290800 D 159
%159-03290810 D 159
%159-03290820 D 159
%159-03290830 D 159
%159-03290840 D 159
%159-03290850 D 159
%159-03290860 D 159
%159-03290870 D 159
%159-03290880 D 159
%159-03290890 D 159
%159-03290900 D 159
%302-03292300 D 302
%302-03296250 D 302
%302-03296300 D 302
%302-03296350 D 302
%302-03297410 D 302
%302-03297415 D 302
%302-03297420 D 302
%302-03297425 D 302
%302-03297430 D 302
%302-03297435 D 302
%302-03297440 D 302
%151-03299700 D 151
%151-03299730 D 151
%151-03299750 D 151
%151-03299780 D 151
%159-03307000 D 159
%302-03307100 D 302
%302-03307110 D 302
%302-03307120 D 302
%302-03307130 D 302
%302-03307140 D 302
%302-03307150 D 302

```

```

N1:=INPUTSCAN(MODE,SOURCE,A3,M[A1],[FF],FLAGS,0);
GO TO RETURN;
DS=35LIT"=**UCSC B5700 TSSMCP OF 24 AUG 77, ";
DEFINE SEQMODE = IF STATAB.STATIONTYPE#TWX THEN FALSE
ELSE (LINEDISC[STAJ]=TWX AND SEQARRAY[STAJ]#0);
NUM:=NUM+INPUTSCAN(MODE,SOURCE,J,BUFSZ=T1,FLAGS,SEQMODE);
INPUTANK[STAJ],[FF]=R1=(IF T1 GTR 0 THEN T1 ELSE
TYPEDSPACE(32,OBJOBINPUTAREAV))+2;
IF PAPER TAPE THEN
BEGIN STREAM(T1=("END"),SOURCE);
RFGIN SI:=LOC T; DI:=SOURCE;
SI:=SI+5; DI:=DI+1;
IF 3 SC=DC THEN
BEGIN DI:=DI-3; DS:=3 LIT " "; END;
END;
FLAGS:=4;
END ELSE
BEGIN STREAM(T1=("END"),SOURCE);
RFGIN SI:=SOURCE; DI:=LOC T;
SI:=SI+1; DI:=DI+5;
IF 3 SC NEQ DC THEN TALLY:=1;
T:=TALLY;
END;
IF POLISH THEN GO TO CE;
END;
BEGIN STREAM(SOURCE,KIT:=PAPER TAPE);
IF SC=MARK THEN
BEGIN DI:=LOC K; DS:=3 LIT "END";
SI:=SI+1; DI:=DI-3;
IF 3 SC=DC THEN
RFGIN TALLY:=2; K:=SI; DI:=K;
T(DI:=DI-3;
DS:=3 LIT " ");
END ELSE TALLY:=1;
END;
R:=POLISH;
FLAGS:=IF R THEN 4 ELSE
IF ((R=2) AND PAPER TAPE) THEN 4 ELSE
IF ((R=2) AND FLAGS) THEN 4 ELSE FLAGS;
NUM:=NUM+INPUTSCAN(MODE,SOURCE,J,BUFSZ,FLAGS,SEQMODE);
LASTSEG[0],[FF]= (SI#TYPEDSPACE
(32,CANNDINPUTAREAV))+2;
DS := 8 DEC; DS := 2 LIT " ";
DI := DI - 10; DS := 7 FILL;
SI := F; DS := 10 CHR;
STREAM(LCFLG:=LONGCARRIAGE[S1],V1=(V1#GETAREA(0))+1);
BEGIN
DS := 6 LIT "DELS<#";
LCFLG (DS := 3 LIT "<");
DS := LIT LEFTARROW;
END;
STREAM(LCFLG:=LONGCARRIAGE[S1],V1=V+1);
BEGIN
DS := 3 LIT "<<#";
LCFLG (DS := 3 LIT "<");
DS := LIT LEFTARROW;
END;

```

```

%159=03313400 D 159
%302=03317500 D 302
%901=03595560 D 901
%159=03614600 D 159
%159=03614700 D 159
%159=03618200 D 159
%155=03618800 D 155
%155=03618810 D 155
%302=03624400 D 302
%302=03624401 D 302
%302=03624402 D 302
%302=03624403 D 302
%302=03624404 D 302
%302=03624405 D 302
%302=03624406 D 302
%302=03624407 D 302
%302=03624408 D 302
%302=03624410 D 302
%302=03624420 D 302
%302=03624430 D 302
%302=03624440 D 302
%302=03624450 D 302
%302=03624460 D 302
%302=03624470 D 302
%302=03624480 D 302
%302=03624600 D 302
%302=03625300 D 302
%302=03625310 D 302
%302=03625320 D 302
%302=03625330 D 302
%302=03625331 D 302
%302=03625332 D 302
%302=03625333 D 302
%302=03625334 D 302
%302=03625340 D 302
%302=03625600 D 302
%302=03625610 D 302
%302=03625620 D 302
%302=03625630 D 302
%159=03634000 D 159
%155=03638500 D 155
%155=03638600 D 155
%149=03733500 D 149
%149=03734000 D 149
%149=03735000 D 149
%150=03826100 D 150
%150=03826200 D 150
%150=03826300 D 150
%150=03826400 D 150
%150=03826500 D 150
%150=03826600 D 150
%150=03900000 D 150
%150=03900100 D 150
%150=03900200 D 150
%150=03900300 D 150
%150=03900400 D 150
%150=03900500 D 150

```

```

IF (P(IODFSC.[3:5]
$ SET OMIT = DKBNODFX
      ,DUP) = @14 OR P(XCH
$ POP OMIT
      ) = @6) AND
      NOT IODFSC.[24:1] AND
      (((P(M)IODFSC.[CF]),DUP) FQV 0)=NOT 0) OR
      ((P(XCH) FQV 32)=NOT 0)) AND
      NOT OKSEGZEROWRITE THEN
      BYBY("SEGMENT ZERO OVERWRITE",23))
      IF IOD.[3:5]=30 THEN GO RETURN; % SP0
LAREL ERRORS,DISKERR,DS,X,SW,LP,DK,DX,DX1,DC,OK,L1;
SWITCH TYPE := OK,LP,OK,OK,DK,OK,OK,OK,OK,OK;
IF U=30 THEN
BEGIN Mf@153+C1:=R;
      DCIOFINISH(R);
END;

```

```

%
% CHECK FOR A PARTIAL WORD BINARY READ WITH NO PARITY ERRORS.
% THIS IS ILLEGAL AND IS MARKED AS BEING A PARITY ERROR.
%

```

```

IF (R.[18:12] AND @4462) = @0440 THEN % RIN READ=NO PAR
      IF R.[15:3] ≠ ((8-R.[22:1]) AND 7) THEN % PART WD XFER
      R.[28:1] := MOD310S; % MARK AS PARITY ERROR IF MOD III
IF (E:=(Mf@153+C1):=R).[26:7]+(V:=T.[5:8]) NEQ 0 THEN
SVOIDT 04167801
RRRMECH = NOT TWO(U) AND RRRMECH; % LET STATUS FIND IT
IF RDC(TARLEF(U) OR MULTITARLEF(U))="FULLPGF"
THEN IF IOQUEF(S).[28:1] THEN IOQUEF(S).[FF]@40013 %DBL=CH 11
ELSE IOQUEF(S).[FF]@40012 % DBL SINGLE = SKIP TO CH 10
ELSE % SKIP TO CHAN 1 ON EOP IF NOT 66 LINES
IF (NT1+JAR(P1MIX,10))=0 THEN GO SANDA; % NO LINE DICT
DISKWAIT(=T,30,NT1);
IF (AA+B[0],[FF])=0 OR S>B[1] OR A>1023 THEN
NT1+JAR(P1MIX,A); NT2+AA+S DIV 30;
IF (NT3+NT2 DIV NT1 + 10) > 29 THEN GO TO SANDA;
IF (NT3+JAR(P1MIX,NT3))=0 THEN GO TO SANDA;
I=NT3+NT2 MOD NT1; DISKWAIT(=T,30,I);
IF (J+R[S MOD 30]) < 0 THEN GO TO SANDA;
AA+J.[CF]; IF J.[FF] = 0 THEN J.[FF]=1;

```

```

$ VOIDT
      IF W*(W+S DIV 30) THEN DISKWAIT(=T,30,JAR(P1MIX,(NT1+AA+W)
      DIV (NT2+JAR(P1MIX,A))+10)+NT1 MOD NT2) DIV 1 );
OKSEGFROWRITE:=TRUE;
OKSEGFROWRITE:=FALSE;

```

```

$ SET OMIT      %SETUP SHORT/LONG CARRIAGE MESSAGE
$ POP OMIT
%**%
$ VOIDT
      SPOUTIT(T,CHRGK+1); %SHORT/LONG CARRIAGE MSG

```

```

P(P1MIX); P1MIX:=0;
L:=GETSPACE(A,SCRATCHD,RARFV,SCRATCHSAVE);
CORE:=CORF; SIZE:=SIZE; DISK:=DISK; % INTEGERIZE

```

```

$ VOIDT
      GETESPDISK := ((P(DUP).[CF]-ESPTAR.[CF])x8
      S := (ESPTAR.[CF]+S.[30:15]) & S[30:3]; %
      IF(NR[FF]AND FOURMASK).[1:35]≠0 THEN GO CLOSEIT;
      IF(NR[F+5]AND NINEMASK).[1:28]≠0 THEN

```

```

%203=04004110 D 203
%203=04004119 D 203
%203=04004120 D 203
%203=04004121 D 203
%203=04004130 D 203
%203=04004140 D 203
%203=04004150 D 203
%203=04004155 D 203
%203=04004160 D 203
%203=04004170 D 203
%846=04105998 D 846
%168=04125000 D 168
%168=04128000 D 168
%201=04129100 D 201
%201=04129200 D 201
%201=04129300 D 201
%201=04129400 D 201
%168=04129520 D 168
%168=04129530 D 168
%168=04129540 D 168
%168=04129550 D 168
%168=04129570 D 168
%168=04129580 D 168
%168=04129590 D 168
%201=04130000 D 201
%168=04167100 D 168
%112=04383900 D 112
%724=04391550 D 724
%112=04391560 D 724
%724=04391570 D 724
%724=04391580 D 724
%643=04709900 D 643
%643=04710000 D 643
%643=04711000 D 643
%643=04722000 D 643
%643=04723000 D 643
%643=04724000 D 643
%643=04725000 D 643
%643=04726000 D 643
%643=04727000 D 643
%643=04730000 D 643
%643=04731000 D 643
%203=05665399 D 203
%203=05665401 D 203
%550=GRK 05751400 D 950
%550=GRK 05754600 D 950
%550=GRK 05756500 D 950
%155=05804200 D 155
%155=05811460 D 155
%155=05811470 D 155
%645=06004100 D 645
%995=06020500 D 995
%158=06031000 D 158
%158=06038000 D 158
%117=06058630 D 117
%117=06058640 D 117

```

```

      OKSEGZEROWRITE:=TRUE;
      OKSGZEROWRITE:=FALSE;
SVOIDT 06098201
      REAL NEU; DEFINE U=UT#,UA=UT#,NEU1=NEU+J#,NFU2=NEU+NEU#;
                SPOUTER(Z,25*(NOT LIBMSG) AND 1);
                FORGETSPACE(BUFF); P(XIT);
                TERMINATE (MIX 8 99[CTF]);
                IF (SH+T2+REPLY[P1MIX],[FF]) > 35      THEN % IL
      END ELSE LABELTABLE[T?]=>(*P(DUP));
      BEGIN
        IF P(0,RDS) GTR FENCE
SVOIDT
      ELSE COMPLEXSLEEP((B OR REPLY[P1MIX] GTR 0 OR TERMSET(P1MIX)));
SVOIDT 06472501
      IF JAR[P1MIX,9],SYSJOB NEQ LIBMAINCODE THEN GO INITIATE;

```

```

#####
PROCEDURE EBTABLE;
BEGIN LABEL L;
  P(0,XIT,.L,DEL);
L:;
  0," 3??","1???", "0TS?", "/???", "-LK?", "J???", "???", "???", "RCB?", "A???", 07003310
  "f???", "???", "???", "???", "???", "???",
  "9???", "???", "Z???", "???", "R???", "???", "???", "???", "I???", "???", 07003320
  "???", "???", "???", "???", "???", "???",
  "8#1?", "???", "Y,x?", "???", "Q$]?", "???", "???", "???", "H.[?", "???", 07003330
  "???", "???", "???", "???", "???", "???",
  @07770675,@05770413,@067146616,@65176473,
  @47574656,@45554453,"???", "???", @27742620,@25352436,
  "???", "???", "???", "???", "???", "???",
  %QMARKADDRESSFS
  "???", "???", "???", "???", "???", "???", "???", "???", "???", "???", 07003352
  "???", "???", "???", "???", "???", "???",
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, % SPACE FOR ADDRESSSES
  0,0,0,0,0,0,0;
END;

```

```

#####
REAL PROCEDURE SETUPEBTABLE;
BEGIN REAL ADR; LABEL EXIT;
DEFINE STORADDRESS=
XI=SI; SI:=LOC X; SI:=SI+5;
DS:=3 CHR; DI:=DI+5 #;
EBTABLE;
P([M[(ADR:=*P(.EBTABLE)_INX NOT 1]])_IOR);
  SFTUPEBTABLE:=ADR:=ADR.[CF]+3;
  IF M[ADR+81] = 0 THEN
STREAM (XI=0, TABLE:=ADR, QMARKTABLEADDRESS:=ADR+32, LOCATIONS:=ADR+40);
  BEGIN
    DI:=TABLE; SI:=TABLE; SI:=SI+8;
    Z(40(SI:=SI+4; DS:=4 CHR));
    %FIRST FILL ALL OF ADDRESSFS WITH ADDRESS OF ?S
    SI:=QMARKTABLEADDRESS;
    XI:=SI;
    SI:=LOC X; SI:=SI+5; %POINT SI AT ADDRESS OF QMARKS

```

```

%203=06059659 D 203
%203=06059661 D 203
%111=06097600 D 111
%142=06351055 D 142
%523=06353660 D 523
%528=06353675 D 528
%517=06411010 D 517
%540=06463380 D 540
%764=06463740 D 764
%160=06468000 D 160
%160=06470000 D 160
%160=06471000 D 160
%160=06472100 D 160
%160=06472200 D 160
%160=06473100 D 160
%891=07003250 D 891
#####07003251 D 891
%891=07003252 D 891
%890=07003270 D 890
%890=07003280 D 890
%890=07003290 D 890
%890=07003300 D 890
%890=07003310 D 890
%890=07003312 D 890
%890=07003320 D 890
%890=07003322 D 890
%890=07003330 D 890
%890=07003332 D 890
%892=07003340 C 892
%890=07003342 D 890
%891=07003344 D 891
%890=07003350 D 890
%890=07003352 D 890
%890=07003354 D 890
%890=07003360 D 890
%890=07003370 D 890
%890=07003380 D 890
%891=07003390 D 891
#####07003391 D 891
%891=07003392 D 891
%890=07003400 D 890
%890=07003410 D 890
%890=07003412 D 890
%890=07003414 D 890
%890=07003416 D 890
%890=07003420 D 890
%890=07003430 D 890
%890=07003440 D 890
%890=07003450 D 890
07003462 D 890
%890=07003464 D 890
%890=07003466 D 890
%890=07003468 D 890
%890=07003482 D 890
%890=07003484 D 890
%890=07003486 D 890
%890=07003488 D 890

```



```

DI:=LOCATIONS;
DSI:=3 CHR;
SI:=LOCATIONS;
DSI:=63 WDS;
  XNOW SET ADDRESSES FOR VALID CHARS INTO EBCDIC TABLE
SI:=TABLE;
DI:=LOCATIONS;
3(STOREADDRESS;  XSTORE ADDRESSES FOR LP=0,1,2
  XCORRESPONDING TO BLANK,9 HOLE,8 HOLE
SI:=X;
SI:=SI+63; SI:=SI+1); XSKIP 8 WORDS DOWN TABLE
DI:=DI+8;  XLP=3 IS INVALID
STOREADDRESS;  XLP=4,.... 7 HOLE
DI:=DI+8;  XLP=5 IS INVALID
SI:=X;
SI:=SI+1;  XSKIP 1 CHAR DOWN TABLE
STOREADDRESS;  XLP=6,....7&8 HOLES TOGETHER
DI:=DI+8;
SI:=X;
SI:=SI+1;
STOREADDRESS;  XLP=8,....6 HOLE
DI:=DI+8;
SI:=X;
SI:=SI+1;
STOREADDRESS;  XLP=10,....6&8 HOLES TOGETHER
DI:=DI+40;  XMISS 5 WORDS,....LP=11,12,13,14,15
SI:=X;
SI:=SI+1;
STOREADDRESS;  XLP=16,.... 5 HOLE
DI:=DI+8;
SI:=X;
SI:=SI+1;
STOREADDRESS;  XLP=18,.... 5&8 HOLES TOGETHER
2(DI:=DI+52); XMISS 13 WORDS,....LP=19-31
SI:=X;
SI:=SI+1;
STOREADDRESS;  XLP=32,.... 4 HOLE
DI:=DI+8;
SI:=X;
SI:=SI+1;
STOREADDRESS;  XLP=34,.... 4&8 HOLES TOGETHER
END;
END;
*****
  STREAM PROCEDURE EBCDICCONVERT(,INTO,TABLE,POINTERS);
VALUE INTO,TABLE,POINTERS;
*****
BFCIN
LOCAL HP,LP,SRCE,DEST,HPPTR,LPPTR;
  XPOINT HPPTR & LPPTR TO LAST CHAR OF HP,LP
SI:=LOC HP;
SI:=SI+7;
HPPTR:=SI;  XHIGH PART
SI:=LOC LP;
SI:=SI+7;
LPPTR:=SI;  XLOW PART
SI:=INTO; SI:=SI+8;

```

```

%890-07003490 D 890
%890-07003492 D 890
%890-07003494 D 890
%890-07003496 D 890
%890-07003498 D 890
%890-07003500 D 890
%890-07003502 D 890
%890-07003504 D 890
%890-07003506 D 890
%890-07003508 D 890
%890-07003510 D 890
%890-07003512 D 890
%890-07003514 D 890
%890-07003516 D 890
%890-07003518 D 890
%890-07003520 D 890
%890-07003522 D 890
%890-07003524 D 890
%890-07003526 D 890
%890-07003528 D 890
%890-07003530 D 890
%890-07003532 D 890
%890-07003534 D 890
%890-07003536 D 890
%890-07003538 D 890
%890-07003540 D 890
%890-07003542 D 890
%890-07003544 D 890
%890-07003546 D 890
%890-07003548 D 890
%890-07003550 D 890
%890-07003552 D 890
%890-07003554 D 890
%890-07003556 D 890
%890-07003558 D 890
%890-07003560 D 890
%890-07003562 D 890
%890-07003564 D 890
%890-07003566 D 890
%890-07003568 D 890
%890-07003570 D 890
%890-07003580 D 890
%890-07003590 D 890
*****07003600 D 890
%890-07003610 D 890
%890-07003615 D 890
*****07003620 D 890
%890-07003630 D 890
%890-07003640 D 890
%890-07003650 D 890
%890-07003660 D 890
%890-07003670 D 890
%890-07003680 D 890
%890-07003690 D 890
%890-07003700 D 890
%890-07003710 D 890
%890-07003720 D 890

```

```

DI:=INTO;                                %890-07003730 D 890
  %START CHARACTER TRANSLATE LOOP          %890-07003740 D 890
2(40(                                     %890-07003750 D 890
DEST:=DI;                                 %890-07003760 D 890
  %TRANSFER LOW & HIGH PARTS              %890-07003770 D 890
DI:=HPPTR;                                %890-07003780 D 890
DS:=1 CHR;                                %890-07003790 D 890
DI:=LPPTR;                                %890-07003800 D 890
DS:=1 CHR;                                %890-07003810 D 890
SRCE:=SI;                                %890-07003820 D 890
  %STORE SI FOR NEXT PASS THRU LOOP
  %NOW FIND THE POINTER INTO TABLE APPROPRIATE TO LP %890-07003830 D 890
SI:=POINTERS;                              %890-07003840 D 890
LP(SI:=SI+8);                              %890-07003860 D 890
SI:=SC;                                    %890-07003870 D 890
  %SI NOW POINTS INTO TABLE @ POINT DEPENDANT ON LP
DI:=DFST;                                  %890-07003880 D 890
SI:=SI+HP;                                %890-07003890 D 890
  %SKIP SI THROUGH TABLE HP CHARS
DS:=CHR;                                  %890-07003900 D 890
SI:=SRCE);                                %890-07003910 D 890
DI:=TABLE; DI:=DI+3;                        %890-07003920 D 890
  %POINT TO QMRK
FND CONVERT;                               %890-07003950 D 890
BOOLEAN EBCDIC; REAL FBTABLEADR;          %890-07009200 D 890
  IF EBCDIC THEN                            %890-07031200 D 890
  BEGIN                                     %890-07031300 D 890
    EBCDIC:=FALSE;                          %890-07031400 D 890
    MFC(*P(.EBTABLE) (NX NOT 1)); (2:1)=0; %890-07031500 D 890
  END;                                       %890-07031600 D 890
  STREAM(XI="PACKETS";YI="CONTINU";ZI="END. ", %890-07032000 D 890
    EB="DATA029";INBUFF);                 %890-07032100 D 890
  INBUFF:=SI;                               %890-07034500 D 890
  ELSE BEGIN SI:=INBUFF;DI:=LOC Y;          %890-07036220 D 890
    SI:=INBUFF;                              %890-07036250 D 890
    IF 3 SC=DC THEN TALLY:=7                %890-07036260 D 890
    ELSE BEGIN                               %890-07036270 D 890
      DI:=LOC ER;DI:=DI+1;                  %890-07036280 D 890
      SI:=INBUFF;                            %890-07036290 D 890
      IF 6 SC=DC THEN                        %890-07036300 D 890
        TALLY:=4;                           %890-07036310 D 890
        %890-07036320 D 890
    END;                                     %890-07036330 D 890
  END;                                       %890-07036390 D 890
END END;                                     %890-07039500 D 890
IF PTYPE = 4 THEN                           %890-07039600 D 890
  BEGIN PTYPE:=0; EBCDIC:=TRUE;            %890-07039700 D 890
  EBTABLEADR:=SETUPERTABLE;                %890-07039800 D 890
END;                                         %890-07059000 D 890
  ELSE BEGIN WHILE(QI=WAITIO(@40000000+INBUFF+ %890-07059010 D 890
    EBCDIC*@40000000,FIRST*4;              %890-07059110 D 654
    IF FIRST AND CONLY AND NOT CONTINUE THEN %890-07059112 D 654
      GO EXIT ELSE                            %890-07060100 D 890
    IF EBCDIC THEN EBCDICCONVERT(INBUFF,    %890-07060200 D 890
      EBTABLEADR+FBTABLEADR+40) ELSE        %890-07066100 D 890
    STREAM(QMK:=12;BCL:=1-EBCDIC;INBUFF); %890-07066200 D 890
    BEGIN                                     %890-07066300 D 890
      SI:=INBUFF;                            %890-07066350 D 890
      BCL (IF SC="" THEN JUMP OUT TO L1); %890-07066400 D 890
      IF SC=@14 THEN                          %890-07066500 D 890
        L1 BEGIN SI:=LOC QMK;SI:=SI+7;      %890-07066550 D 890
          DS:=1 CHR;                          %890-07066550 D 890
        END;
      END;
    END;
  END;

```

```

        TALLY:=1;
        SI:=INBUFF;
        2(36( IF SC=">" THEN
        BEGIN INBUFF:=SI; DI:=INBUFF;
        DS:=LIT "="
        END;
        IF SC="≥" THEN
        BEGIN INBUFF:=SI; DI:=INBUFF;
        DS:=LIT ""
        END; SI:=SI+1;))
    END;
    QMK :=TALLY;
    END;
    Q:=P OR Q;

```

```

SVOIDT 07071912
SVOIDT 07072902
S VOIDT

```

```

FIRSTCARD := GETSPACE(10,CONTROLCARDAREAV,1)+2;
ELSE BEGIN OUTBUFFOLD := OUTBUFF :=
        GETSPACE(60,IOBUFFFRAREAV,1)+2;
        H := SAVEARRAYDESC(30,DISKHFADERAREAV);
        INBUFF := GETSPACE(21,IOBUFFFRAREAV,1) + 2;
        ADECK := 0; FIRSTORSEC :=

```

```

S
S
SVOIDT 07118401

```

```

        IF PACKETCARD = 5 THEN
        IF PTYPE = 3 AND NOT CONTINUE AND NOT ADECK THEN
        BOMB:  IF OU=18 THEN FORGETIT;
        SHEAT := [MFI:=TYPEDSPACE(31,SHEETAREAV+64)] & 30[SIZE];
        SFG0 := TYPEDSPACE(30,SEGZEROAREAV+64);
        SHEAT[24]:=MCP;

```

```

        SI:=LOC T1;
        IF SC#"0" THEN GO TO AQ
        ELSE SI:=SI+1;
        AQ:  BEGIN DS+3 LIT "S ";
        IF (AI=NI:=P(DUP)+1) > (H[7]+1) THEN
            IF I>0 THEN % NEXT BUFF EXISTS
            DS:=8LIT"ix0x4000"; DS:=8LIT"OPACKET";
        H := CIDROW[R] := [MFI:=
        GETSPACE(94,CIDROWAREAV+64,1)+2] & 94 SIZE;
        T:=GETSPACE(13,CONTROLCARDAREAV+64,0)+4;
        IF [MEM[MIX,0]].[CF] ≥ FENCE OR ITSAILBATCH THEN
            DI+DI:=8; DS=LIT "+";
        IF NOT T.[1:1] THEN IF T=0 THEN
            IF T.[1:1]=0 AND T=0 THEN BEGIN
                STREAM(BUFF); DS+17 LIT "NOT PG=ED(PRN=0)";
                LABELTABLE[U]+ @14; GO EXIT END; T=ARS(T);
            IF TEST THEN BEGIN STREAM(B+T,BUFF);
        BEGIN DS+10 LIT"PG=ED(PRN=); SI:=LOC B; DS+5 DEC; DS+2 LIT")";
    END;
    END % PRINT PRN WITH PLAIN PGMT

```

```

SVOIDT 08135001

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
*****
%*
%*
%*
%*
P R I N T D I R E C T O R Y
- - - - -

```

%890-07066600	D	890
%890-07066610	D	890
%890-07066620	D	890
%890-07066630	D	890
%890-07066635	D	890
%890-07066640	D	890
%890-07066650	D	890
%890-07066660	D	890
%890-07066665	D	890
%890-07066670	D	890
%890-07066700	D	890
%890-07066800	D	890
%890-07066850	D	890
%890-07066900	D	890
%654-07071899	D	654
%654-07072899	D	654
%890-07086000	D	890
%155-07087000	D	155
%155-07095000	D	155
%155-07095100	D	155
%155-07097000	D	155
%892-07101000	C	892
%172-07108100	D	172
%172-07108999	D	172
%172-07109001	D	172
%106-07118100	D	106
%106-07118500	D	106
%106-07139520	D	106
%161-07231000	D	161
%07256400	D	155
%155-07257400	D	155
%129-07262100	D	129
%511-07295100	D	511
%511-07295102	D	511
%511-07295104	D	511
%511-07295160	D	511
%639-07397000	D	639
%639-07401900	D	639
%133-07418700	D	133
%155-07433000	D	155
%155-07433100	D	155
%155-07446000	D	155
%151-07500000	D	151
%725-08029046	D	725
%725-08033980	D	725
%725-08033992	D	725
%725-08033993	D	725
%725-08033994	D	725
%708-08051000	D	708
%708-08051004	D	708
%708-08051005	D	708
%141-08095000	D	141
%08095050	D	141
%08095100	D	141
%08095150	D	141
%08095200	D	141
%08095250	D	141

PURPOSE: THIS PROCEDURE HANDLES THE PD, FX, LF, LC AND LS MESSAGES. ALL MESSAGES ARE FORMS OF THE PD MESSAGE EXCEPT FOR CERTAIN DIFFERENCES. THE PD MESSAGE WILL LIST THE NAMES OF THE DESIRED FILES AS WELL AS CERTAIN PIECES OF INFORMATION IF REQUESTED.

SYNTAX: THE FORMAT OF ALL THE KEYIN MESSAGES WITH THE EXCEPTION OF THE LF MESSAGE IS AS FOLLOWS:

<MESSAGE> <FILE SPECIFIER> <OPTION=LIST>  
 <MESSAGE> ::= PD / FX / LC / LS  
 <FILE SPECIFIER> ::= <FILENAME> / <FILENAME> "/" <FILENAME>  
 <OPTION=LIST> ::= <EMPTY> / <OPTION> / <OPTION> , <OPTION=LIST>  
 <OPTION> ::= RECS / LAST / DATE / SAVE / SIZE / CREATOR / SECURITY / ALL  
 <FILENAME> ::= [FROM 1 TO 7 CHARACTERS OPTIONALLY ENCLOSED IN QUOTES] / \*

THE SYNTAX FOR THE LF MESSAGE IS:

LF <USERID> <OPTION=LIST>

SEMANTICS: THE FX, LF, LC, AND LS MESSAGES ARE JUST MODIFIED FORMS OF THE PD MESSAGE. THE FX MESSAGE IS THE SAME AS A PD EXCEPT THAT ONLY THE EXPIRED FILES ARE LISTED. THE LF IS THE SAME AS A PD EXCEPT THAT ONLY THE FILES BELONGING TO THE SPECIFIED USER ARE LISTED. THE LC MESSAGE IS THE SAME AS A "PD <FILE SPECIFIER> CREATOR". THE LS MESSAGE IS THE SAME AS "PD <FILE SPECIFIER> SECURITY, CREATOR".

THE OPTIONS SPECIFY WHAT ADDITIONAL INFORMATION IS TO BE LISTED BESIDES THE DEFAULT INFORMATION. IF AN UNRECOGNIZABLE WORD IS ENCOUNTERED IN THE OPTION LIST, IT IS IGNORED. THE AVAILABLE OPTIONS ARE:

- RECS = NUMBER OF RECORDS IN THE FILE. (SPECIAL LOGIC IS INCORPORATED TO THE LIST THE CORRECT NUMBER FOR BACKUP FILES).
- LAST = LAST ACCESS DATE.
- DATE = CREATION DATE.
- SAVE = SAVE FACTOR.
- SIZE = SIZE OF THE FILE IN SEGMENTS.
- CREATOR = PRIVILEGED USERCODE ASSOCIATED WITH FILE.

*%08095300	D	141
*%08095350	D	141
*%08095400	D	141
*%08095450	D	141
*%08095500	D	141
*%08095550	D	141
*%08095600	D	141
*%08095650	D	141
*%08095700	D	141
*%08095750	D	141
*%08095800	D	141
*%08095850	D	141
*%08095900	D	141
*%08095950	D	141
*%08096000	D	141
*%08096050	D	141
*%08096100	D	141
*%08096150	D	141
*%08096200	D	141
*%08096250	D	141
*%08096300	D	141
*%08096350	D	141
*%08096400	D	141
*%08096450	D	141
*%08096500	D	141
*%08096550	D	141
*%08096600	D	141
*%08096650	D	141
*%08096700	D	141
*%08096750	D	141
*%08096800	D	141
*%08096850	D	141
*%08096900	D	141
*%08096950	D	141
*%08097000	D	141
*%08097050	D	141
*%08097100	D	141
*%08097150	D	141
*%08097200	D	141
*%08097250	D	141
*%08097300	D	141
*%08097350	D	141
*%08097400	D	141
*%08097450	D	141
*%08097500	D	141
*%08097550	D	141
*%08097600	D	141
*%08097650	D	141
*%08097700	D	141
*%08097750	D	141
*%08097800	D	141
*%08097850	D	141
*%08097900	D	141
*%08097950	D	141
*%08098000	D	141
*%08098050	D	141
*%08098100	D	141

```

%*          (FOR BACKUP FILES THE LABEL OF THE PRINT FILE          *%08098150 D 141
%*          AND THE NAME OF THE PROGRAM CREATING THE BACKUP        *%08098200 D 141
%*          IS ALSO LISTED),                                       *%08098250 D 141
%*          SECURITY = ACCESS PRIVILEGES OF THE FILE, I.F.,       *%08098300 D 141
%*          LOCKED, UNLOCKED, PUBLIC, PRIVATE, FREE.              *%08098350 D 141
%*          *%08098400 D 141
%*          *%08098450 D 141
%***** *%08098500 D 141
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX *%08098550 D 141
PROCEDURE PRINTDIRECTORY(BUFF);
  VALUF BUFF;
  REAL BUFF;
BEGIN
  INTFGR
  I,      % NORMALLY CONTAINS OPTION NUMBER.
  J,      % JUNK.
  CODE;   % = 0 = PD.
          % = 1 = EX.
          % = 2 = LC.
          % = 3 = LF.
          % = 4 = LS.
  REAL
  MFID,   % MFID OF DESIRED FILE OR -1 IF "=",
  FID,    % FID OF DESIRED FILE OR -1 IF "=",
  C,      % ADDRESS OF DISK HEADER.
  D,      % MFID OF LAST FILE FOUND BY SEEKNAME.
  E,      % FID OF LAST FILE FOUND BY SEEKNAME.
  N,      % WORK VARIABLE USED BY SEEKNAME TO SAVE INFO.
  T,      % NORMALLY USED TO SAVE DEST. INDEX.
  INFO,   % BIT MASK FOR OPTIONS SELECTED.
  LABELREC, % ADDRESS OF LABEL RECORD FOR PBD ( IF LC ).
  USERID, % USERID IF LF.
  X;      % JUNK.
  ARRAY
  HDR[*]; % DESCRIPTOR TO DISK HEADER.
  XLIST[*]; % SOME DAY WE MAY ALLOW EXCEPTION LIST.
  BOOLEAN
  PRDLOG, % TRUE IF SELECTED FILE IS A PBD.
  FOUNDATFILE; % TRUE IF WE LISTED OUT AT LEAST ONE FILE.
  LABEL EXIT, DROPOUT, DUMMY, OPTIONS;
  DEFINE NUMOPTS = 7;
  DEFINE PD = (CODE = 0);
  DEFINE EX = (CODE = 1);
  DEFINE LC = (CODE = 2);
  DEFINE LF = (CODE = 3);
  DEFINE LS = (CODE = 4);
  DEFINE
  PRIMARYUSER = HDR[2]; % PRIV. USER CODE,
  SAVFACTOR = HDR[3].[2:10]; % SAVE FACTOR.
  LASTACCESSDATE = HDR[3].[12:18]; % LAST ACCESS DATE.
  CREATIONDATE = HDR[3].[30:18]; % CREATION DATE.
  GUARDFILEMFID = HDR[5]; % MFID OF GUARD FILE.

```

GUARDFILEFID	= HDR[6]#,	% FID OF GUARD FILE.	%141-08101100	D	141
EOFPOINTER	= HDR[7]#,	% EOF POINTER.	%141-08101150	D	141
SFGSPERROW	= HDR[8]#,	% SEG. PER ROW.	%141-08101200	D	141
NOOFROWS	= HDR[9].[43:5]#,	% NO. OF ROWS DECLARED.	%141-08101250	D	141
HFADERADDRESS	= HDR.[CF]#,	% CORE ADDRESS OF HEADER.	%141-08101300	D	141
%***** S U B R O U T I N E S *****			%*****08101350	D	141
SUBROUTINE GETREADY;			%141-08101400	D	141
% -----			%141-08101450	D	141
BEGIN			%141-08101500	D	141
CODE I= BUFF.[9:16];			%141-08101550	D	141
BUFF I= (TI=BUFF).[15:15] = 1;			%141-08101600	D	141
INFO I= 0 & (LC OR LF OR LS)[42:47:1] & LS[43:47:1];			%141-08101650	D	141
END OF GETREADY;			%141-08101700	D	141
% -----			%141-08101750	D	141
SUBROUTINE GETFILESPECIFIER;			%141-08101800	D	141
% -----			%141-08101850	D	141
BEGIN			%141-08101900	D	141
NAMEID(MFID,T); % GET MFID (OR USERCODE IF "LF")			%141-08101950	D	141
IF LF THEN USERID I= MFID; % FIRST THING IS USERID FOR LF.			%141-08102000	D	141
IF MFID = "+ " OR LF THEN			%141-08102050	D	141
MFID I= - 1;			%141-08102100	D	141
NAMEID(FID,T)			%141-08102150	D	141
IF FID = "/" THEN % GET FID			%141-08102200	D	141
BEGIN			%141-08102250	D	141
NAMEID(FID,T);			%141-08102300	D	141
NAMEID(N,T); % GET NEXT ITEM.			%141-08102350	D	141
END			%141-08102400	D	141
ELSE % NO FID SPECIFIED IMPLIES FID OF "="			%141-08102450	D	141
BEGIN			%141-08102500	D	141
N I= FID;			%141-08102550	D	141
FID I= - 1;			%141-08102600	D	141
END;			%141-08102650	D	141
IF FID.[6:16] = LEFTARROW OR LF THEN % NO FID SPECIFIED			%141-08102700	D	141
FID I= - 1;			%141-08102750	D	141
END OF GETFILESPECIFIER;			%141-08102800	D	141
% -----			%141-08102850	D	141
SUBROUTINE PROCSOPT,ONLIST;			%141-08102900	D	141
% -----			%141-08102950	D	141
BEGIN			%141-08103000	D	141
WHILE N # "+ " DO % ACCUMULATE OPTIONS			%141-08103050	D	141
BEGIN			%141-08103100	D	141
FOR I I= - 1 STEP 1 UNTIL (NUMOPTS = 2) DO			%141-08103150	D	141
DUMMY: IF P(.OPTIONS,I,+,LOD) = N THEN % MATCHES AN OPTION WORD			08103200	D	141
BEGIN			%141-08103250	D	141
INFO I= INFO OR TWO(I+1); % SET BIT CORRESPONDING TO OPT			08103300	D	141
GO TO DROPOUT;			%141-08103350	D	141
END			%141-08103400	D	141
ELSE % CHECK FOR "ALL"			%141-08103450	D	141
IF N = "ALL" THEN % SET ALL OPTION WORD BITS.			%141-08103500	D	141
INFO I= NOT 0;			%141-08103550	D	141
DROPOUT:			%141-08103600	D	141
NAMEID(N,T); % GET NEXT OPTION WORD;			%141-08103650	D	141
IF N = "+ " THEN NAMEID(N,T); % SKIP OVER COMMA.			%141-08103700	D	141
END;			%141-08103750	D	141
GO TO EXIT;			%141-08103800	D	141
% -----			%141-08103850	D	141
OPTIONS :; "RECS " , % OPTION 0 (INFO.[47:1])			%141-08103900	D	141

```
"LAST " , % OPTION 1 (INFO.[46:1])
"DATE " , % OPTION 2 (INFO.[45:1])
"SIZE " , % OPTION 3 (INFO.[44:1])
"SECURITY" , % OPTION 4 (INFO.[43:1])
"CREATOR" , % OPTION 5 (INFO.[42:1])
"SAVE " ; % OPTION 6 (INFO.[41:1])
```

```
EXIT:
END OF PROCSSOPTIONLIST;
```

```
%
SUBROUTINE GETSET;
```

```
% -----
BEGIN
IF EX OR INFO # 0 THEN % WE WILL NEED THE HDR.
HDR I= %QUF & SPACE(30) [CTC]
END OF GETSET;
```

```
%
BOOLEAN SUBROUTINE WEGOTAFILE;
```

```
% -----
BEGIN
SFEKNAM(MFID,FID,C,D,E,N,XLIST); % FIND A FILE.
WEGOTAFILE I= C # 0; % C IS ADDRESS OF DISK HEADER.
END OF WEGOTAFILE;
```

```
%
BOOLEAN SUBROUTINE WEWANTTHISFILE;
```

```
% -----
BEGIN
PRDTOR I= ((D EQV "PRD " ) # NOT 0 OR (D EQV "PUD " )
# NOT 0);
IF HEADERADDRESS # 0 THEN % WE NEED THE HEADER.
DISKWAIT("HEADERADDRESS,30,C"); % READ HEADER.
IF LF THEN % CHECK TO SEE IF WE WANT THIS FILE.
WEWANTTHISFILE I= PRIMARYUSER = USERID
ELSE
IF EX THEN
BEGIN
STREAM(AI=CALCULATEPURGE(=SAVEFACTOR),XI={X});
BEGIN
SI I= LOC AI DS I= 8 OCT;
END;
WEWANTTHISFILE I= X > LASTACCESSDATE; % TRUE IF FILE EXPIRED
END
ELSE
WEWANTTHISFILE I= TRUE;
END OF WEWANTTHISFILE;
```

```
%
SUBROUTINE PUTINFILENAME;
```

```
% -----
BEGIN
STREAM(AI=0 ; D ; F ; RUFF); % SET UP FILE NAME.
BEGIN
SI I= LOC D;
DS I= LIT " ";
? ( SI I= SI + 1 ) DS I= 7 CHR; DS I= LIT "/" );
DI I= DI + 1;
DS I= 2 LIT " "; % NEED 2 SPACES TO ALLOW FOR ARROW.
A I= DI;
END;
```

```
%141-08103950 D 141
%141-08104000 D 141
%141-08104050 D 141
%141-08104100 D 141
%141-08104150 D 141
%141-08104200 D 141
%141-08104250 D 141
%141-08104300 D 141
%141-08104350 D 141
%141-08104400 D 141
%141-08104450 D 141
%141-08104500 D 141
%141-08104550 D 141
%141-08104600 D 141
%141-08104650 D 141
%141-08104700 D 141
%141-08104750 D 141
%141-08104800 D 141
%141-08104850 D 141
%141-08104900 D 141
%141-08104950 D 141
%141-08105000 D 141
%141-08105050 D 141
%141-08105100 D 141
%141-08105150 D 141
%141-08105200 D 141
%141-08105250 D 141
%141-08105300 D 141
%141-08105350 D 141
%141-08105400 D 141
%141-08105450 D 141
%141-08105500 D 141
%141-08105550 D 141
%141-08105600 D 141
%141-08105650 D 141
%141-08105700 D 141
%141-08105750 D 141
%141-08105800 D 141
%141-08105850 D 141
%141-08105900 D 141
%141-08105950 D 141
%141-08106000 D 141
%141-08106050 D 141
%141-08106100 D 141
%141-08106150 D 141
%141-08106200 D 141
%141-08106250 D 141
%141-08106300 D 141
%141-08106350 D 141
%141-08106400 D 141
%141-08106450 D 141
%141-08106500 D 141
%141-08106550 D 141
%141-08106600 D 141
%141-08106650 D 141
%141-08106700 D 141
%141-08106750 D 141
```

```

      T := P; % SAVE OFF DEST. INDEX,
      END OF PUTINFILENAME;
%
SUBROUTINE DORECS;
% -----
      BFGIN
      STREAM(A:=IF PBDT0G THEN EOFPOINTER*5 ELSE EOFPOINTER+1 ;T);

      BFGIN
      DS := 9 LIT "RECORDS: ";
      SI := LOC A;
      DS := 8 DEC; % CONVERT NUMBER OF RECORDS TO DEC.
      A := DI; % SAVE OFF DI BEFORE ZERO SUPPRESSING.
      DI := DI - 8;
      DS := 7 FILL;
      END;
      T := P; % SAVE DESTINATION ADDRESS.
      END OF DORECS;
%
SUBROUTINE DODATEORLAST;
% -----
      BFGIN
      STREAM(X:=X);
      BFGIN
      SI := X;
      DS := 8 DEC; % CONVERT DATE TO DECIMAL.
      END;
      GIMEDATE([X],[CF],-X); % CONVERT JULIAN DATE TO 6 DIGITS.
      STREAM(A:=(I=1) ; X, T);
      BFGIN
      A ( DS := 10 LIT "ACCESSFD: "; JUMP OUT TO L1);
      DS := 9 LIT "CREATED: ";
      L1:
      SI := LOC X;
      SI := SI + 2;
      3 ( DS := 2 CHR; DS := LIT "/*");
      DI := DI - 1; % ERASE THE EXTRA SLASH.
      A := DI; % SAVE DEST. INDEX.
      END;
      T := P; % SAVE DESTINATION ADDRESS.
      END OF DODATEORLAST;
%
SUBROUTINE DOSIZE;
% -----
      BFGIN
      NT2 := NOOFROWS; % NO. OF ROWS DECLARED.
      NT1 := 0; % NUMBER OF ROWS PROCESSED.
      FOR J := 1 STEP 1 UNTIL NT2 DO % CHECK TO SEE IF ROW EXISTS.
      IF HDR[J+9] # 0 THEN NT1 := NT1 + 1; % BUMP UP COUNT.
      STREAM(A:=NT1*SEGSPERROW ; T);
      BFGIN
      DS := 10 LIT "SEGMENTS: ";
      SI := LOC A;
      DS := 8 DEC;
      A := DI;
      DI := DI - 8;
      DS := 7 FILL;

```

```

%141-08106800 D 141
%141-08106850 D 141
%141-08106900 D 141
%141-08106950 D 141
%141-08107000 D 141
%141-08107050 D 141
%141-08107100 D 141
%141-08107150 D 141
%141-08107200 D 141
%141-08107250 D 141
%141-08107300 D 141
%141-08107350 D 141
%141-08107400 D 141
%141-08107450 D 141
%141-08107500 D 141
%141-08107550 D 141
%141-08107600 D 141
%141-08107650 D 141
%141-08107700 D 141
%141-08107750 D 141
%141-08107800 D 141
%141-08107850 D 141
%141-08107950 D 141
%141-08108000 D 141
%141-08108050 D 141
%141-08108100 D 141
%141-08108150 D 141
%141-08108200 D 141
%141-08108250 D 141
%141-08108300 D 141
%141-08108350 D 141
%141-08108400 D 141
%141-08108450 D 141
%141-08108500 D 141
%141-08108550 D 141
%141-08108600 D 141
%141-08108650 D 141
%141-08108700 D 141
%141-08108750 D 141
%141-08108800 D 141
%141-08108850 D 141
%141-08108900 D 141
%141-08108950 D 141
%141-08109000 D 141
%141-08109050 D 141
%141-08109100 D 141
%141-08109150 D 141
%141-08109200 D 141
%141-08109250 D 141
%141-08109300 D 141
%141-08109350 D 141
%141-08109400 D 141
%141-08109450 D 141
%141-08109500 D 141
%141-08109550 D 141
%141-08109600 D 141
%141-08109650 D 141

```



```

      END;
      T := P; % SAVE DESTINATION ADDRESS.
    END OF DOSIZE;
%
  SUBROUTINE DOSECURITY;
% -----
  BEGIN
    J := IF PRIMARYUSER = 0 THEN 0 % FREE FILE
        ELSE IF GUARDFILEMFD = MARK THEN % UNLOCK OR PUBLIC
            IF GUARDFILEFID = MARK THEN 1 % UNLOCKED
                ELSE 2 % PUBLIC
            ELSE IF GUARDFILEMFD < 0 THEN 3 % PRIVATE
                ELSE 4 % LOCKED
    STREAM(J ; A:=GUARDFILEMFD, B:=GUARDFILEFID, T);
    BEGIN
      DS := 10 LIT "SECURITY ";
      CI := CI + J;
      GO TO FREE;
      GO TO UNLOCK;
      GO TO PUBLIC;
      GO TO PRIVATE;
    LOCK: DS := 6 LIT "LOCKED"; GO TO EXIT;
    PRIVATE: DS := 22 LIT "PRIVATE (SECURED WITH ";
      SI := LOC A;
      2 ( SI := SI + 1; DS := 7 CHR; DS := LIT "/" );
      DI := DI + 1;
      DS := LIT " ");
      GO TO EXIT;
    PUBLIC: DS := 6 LIT "PUBLIC"; GO TO EXIT;
    UNLOCK: DS := 8 LIT "UNLOCKED"; GO TO EXIT;
    FREE: DS := 4 LIT "FREE";
    EXIT: J := DI;
  END;
  T := P; % SAVE DESTINATION ADDRESS.
  END OF DOSECURITY;
%
  SUBROUTINE DOREFATOR;
% -----
  BEGIN
    $SET OMIT = PACKETS
    IF J:=(PRD TOG AND (F:[30:18] = 1)) THEN % REEL 1 OF PRD
    $POP OMIT
    $SET OMIT = NOT(PACKETS)
    IF J:=(PBD TOG AND (E:[42:6] = 1)) THEN % REEL 1 OF PBD
    $POP OMIT
    BEGIN
      IF LABELREC = 0 THEN LABELREC := SPACE(30);
      DISKWAIT:=LABELREC,30,HDR[10]+2);
    END;
    STREAM(J ; B:=PRIMARYUSER=0, C:=PRIMARYUSER,
          DI:=LABELREC INX 12, T);
    BEGIN
      DS := 9 LIT "CREATOR ";
      B ( DS := 4 LIT "NONE"; JUMP OUT TO L2 );
      SI := LOC C;
      SI := SI + 1;
      DS := 7 CHR;

```

```

%141-08109700 D 141
%141-08109750 D 141
%141-08109800 D 141
%141-08109850 D 141
%141-08109900 D 141
%141-08109950 D 141
%141-08110000 D 141
%141-08110050 D 141
%141-08110100 D 141
%141-08110150 D 141
%141-08110200 D 141
%141-08110250 D 141
%141-08110300 D 141
%141-08110350 D 141
%141-08110400 D 141
%141-08110450 D 141
%141-08110500 D 141
%141-08110550 D 141
%141-08110600 D 141
%141-08110650 D 141
%141-08110700 D 141
%141-08110750 D 141
%141-08110800 D 141
%141-08110850 D 141
%141-08110900 D 141
%141-08110950 D 141
%141-08111000 D 141
%141-08111050 D 141
%141-08111100 D 141
%141-08111150 D 141
%141-08111200 D 141
%141-08111250 D 141
%141-08111300 D 141
%141-08111350 D 141
%141-08111400 D 141
%141-08111450 D 141
%141-08111500 D 141
%141-08111550 D 141
%141-08111600 D 141
%141-08111650 D 141
%141-08111700 D 141
%141-08111750 D 141
%141-08111800 D 141
%141-08111850 D 141
%141-08111900 D 141
%141-08111950 D 141
%141-08112000 D 141
%141-08112050 D 141
%141-08112100 D 141
%141-08112150 D 141
%141-08112200 D 141
%141-08112250 D 141
%141-08112300 D 141
%141-08112350 D 141
%141-08112400 D 141
%141-08112450 D 141
%141-08112500 D 141

```

```

L2:
  J ( DS := 2 LIT " (" )
  SI := DI
  2 ( SI := SI + 1 ) DS := 7 CHR; DS := LIT "/" )
  DI := DI - 1 ) DS := 4 LIT " OF " )
  2 ( SI := SI + 1 ) DS := 7 CHR; DS := LIT "/" )
  DI := DI - 1 ) DS := LIT ")" )
  J := DI;
END;
T := P; % SAVE DESTINATION ADDRESS.
END OF DOCRATOR;

%
SUBROUTINE DOSAVEFACTOR;
% -----
BEGIN
  STREAM(A := SAVEFACTOR ; T);
  BEGIN
    DS := 6 LIT "SAVE: ";
    SI := LOC A;
    DS := 3 DFC;
    A := DI;
    DI := DI - 3;
    DS := 2 FILL;
  END;
  T := P; % SAVE DESTINATION ADDRESS.
END OF DOSAVEFACTOR;

%
SUBROUTINE DOOPTIONS;
% -----
BEGIN
  FOR I := 0 STEP 1 UNTIL (NUMOPTS - 1) DO % SET IF OPTION BIT
  IF (TWO(I) AND INFO) # 0 THEN %OPTION SELECTED.
  BEGIN
    CASE I OF
      BEGIN
        DORECS; % CASE 0 = "RECS"
        BEGIN % CASE 1 = "LAST"
          X := LASTACCESSDATE;
          DODATEORLAST;
        FND OF CASE 1;
        BEGIN % CASE 2 = "DATE"
          X := CREATIONDATE;
          DODATEORLAST;
        FND OF CASE 2;
        DOSIZE; % CASE 3 = "SIZE"
        DOSECURITY; % CASE 4 = "SECURITY"
        DOCRATOR; % CASE 5 = "CREATOR"
        DOSAVEFACTOR; % CASE 6 = "SAVE"
      END OF CASES;
      STREAM(I ; T); % PUT COMMA AFTER LAST OPTION.
      BEGIN
        DS := 2 LIT ", ";
        I := DI;
      FND;
      T := P;
    END OF LOOP TO PROCESS OPTIONS;
  END OF DOOPTIONS;

```

%141-08112550	D	141
%141-08112600	D	141
%141-08112650	D	141
%141-08112700	D	141
%141-08112750	D	141
%141-08112800	D	141
%141-08112850	D	141
%141-08112900	D	141
%141-08112950	D	141
%141-08113000	D	141
%141-08113050	D	141
%141-08113100	D	141
%141-08113150	D	141
%141-08113200	D	141
%141-08113250	D	141
%141-08113300	D	141
%141-08113350	D	141
%141-08113400	D	141
%141-08113450	D	141
%141-08113500	D	141
%141-08113550	D	141
%141-08113600	D	141
%141-08113650	D	141
%141-08113700	D	141
%141-08113750	D	141
%141-08113800	D	141
%141-08113850	D	141
%141-08113900	D	141
%141-08113950	D	141
%141-08114000	D	141
SET. 08114050	D	141
%141-08114100	D	141
%141-08114150	D	141
%141-08114200	D	141
%141-08114210	D	141
%141-08114250	D	141
%141-08114290	D	141
%141-08114300	D	141
%141-08114310	D	141
%141-08114320	D	141
%141-08114340	D	141
%141-08114350	D	141
%141-08114360	D	141
%141-08114370	D	141
%141-08114400	D	141
%141-08114450	D	141
%141-08114500	D	141
%141-08114550	D	141
%141-08114600	D	141
%141-08114650	D	141
%141-08114700	D	141
%141-08114750	D	141
%141-08114800	D	141
%141-08114850	D	141
%141-08114900	D	141
%141-08114950	D	141
%141-08115000	D	141

```

%
% SUBROUTINE MAKETHEMESSAGE;
% -----
  RFGIN
  IF FOUNDAFILE THEN % WE NFEED A BUFFER.
  BUFF I= SPACE(30);
  PUTINFILENAME;
  DOOPTIONS;
  STREAM(T); % PUT IN THE LEFT ARROW.
  BEGIN
    DI I= DI -2;
    DS I= LIT LEFTARROW;
  END;
  END OF MAKETHEMESSAGE;
%
% SUBROUTINE COMPLAIN;
% -----
  RFGIN
  STREAM(BUFF);
  DS I= 8 LIT " NULL " % ORIGINAL INPUT IS AT BUFF + 1.
  SPOUT(BUFF);
  END OF COMPLAIN;
%
% SUBROUTINE FORGETEVERYTHING;
% -----
  BEGIN
  IF HEADERADDRESS # 0 THEN FORGETSPACE(HEADERADDRESS);
  IF LABFLREC # 0 THEN FORGETSPACE(LABFLREC);
  END OF FORGETEVERYTHING;
%***** S T A R T   O F   C O D E *****
GETREADY;
GETFILESPECIFIER;
PROCESOPTIONS;
GETSET;
WHILE WFGOTAFILE DO
  IF WEWANTTHISFILE THEN
    BEGIN
      MAKETHEMESSAGE;
      SPOUT(BUFF);
      FOUNDAFILE I= TRUE;
    END;
  IF NOT FOUNDAFILE THEN COMPLAIN;
FORGETEVERYTHING;
END OF PRINTDIRECTORY;
  LABEL FOUND,FIRFITUP,QUIT;
  END ELSE % Q<=0, PR MT
  BEGIN RRRMECH+TWO(U+ABS(Q)) OR RRRMECH;
  LABELTABLE[U] +
    PBT&TINU[V][6:30:18]&@21[1:43:5];
  MULTITABLE[V] + PBT;
  LABELTABLE[V] + PBT&TINU[U][6:30:18]&
    @21[1:43:5];
  GO FIRFITUP;
  END
%
%141-08115050 D 141
%141-08115100 D 141
%141-08115150 D 141
%141-08115200 D 141
%141-08115250 D 141
%141-08115300 D 141
%141-08115350 D 141
%141-08115400 D 141
%141-08115450 D 141
%141-08115500 D 141
%141-08115550 D 141
%141-08115600 D 141
%141-08115650 D 141
%141-08115700 D 141
%141-08115750 D 141
%141-08115800 D 141
%141-08115850 D 141
%141-08115900 D 141
%141-08115950 D 141
%141-08116000 D 141
%141-08116050 D 141
%141-08116100 D 141
%141-08116150 D 141
%141-08116200 D 141
%141-08116250 D 141
%141-08116300 D 141
%141-08116350 D 141
%141-08116400 D 141
%141-08116450 D 141
%141-08116500 D 141
%141-08116550 D 141
%141-08116600 D 141
%141-08116650 D 141
%141-08116700 D 141
%141-08116750 D 141
%141-08116800 D 141
%141-08116850 D 141
%141-08116900 D 141
%141-08116950 D 141
%141-08117000 D 141
%141-08117050 D 141
%141-08117100 D 141
%141-08117150 D 141
%141-08117200 D 141
%717-08255800 D 717
%717-08259800 D 717
%717-08259810 D 717
%717-08259820 D 717
%717-08259830 D 717
%717-08259840 D 717
%717-08259850 D 717
%717-08259860 D 717
%717-08259870 D 717
%717-08259880 D 717
%717-08260899 D 717
%155-08274250 D 155
%155-08275500 D 155

```

SVoidT 08266501

SHEAT[24]:=MCP;	%129=08276365	D	129
CNT: SI:=SI+1; B:=SI; IF SC=" " THEN GO TO CNT; TALLY:=0; 08291125	%155=08385100	D	155
CI=TYPEDSPACE(5,MAINTBUFFAREAV);%	%155=08410000	D	155
BEGIN RI=TYPEDSPACE(5,MAINTBUFFAREAV);%	%644=08533000	D	644
T=((CLOCK+P(RTR)) DIV 60).[31:17] =	%644=08533150	D	644
IF T<0 THEN T = T + @400000;	%535=08634000	D	535
LO: SI+BUFF;	%535=08635000	D	535
L1: IF SC=" " THEN BEGIN SI+SI+1; GO TO L1; END;	%535=08636000	D	535
IF SC<"0" THEN	%535=08637000	D	535
IF SC#"+" THEN	%535=08638000	D	535
BEGIN TALLY+0; T+TALLY; DI+LOC T;	%535=08639000	D	535
B(IF SC=" " THEN JUMP OUT FLSE	%535=08640000	D	535
IF SC#"+" THEN JUMP OUT FLSE	%535=08641000	D	535
IF SC>"0" THEN JUMP OUT FLSE DS+CHR);	%535=08642000	D	535
BUFF+SI; SI+OPTER;	%535=08643000	D	535
63(DI+LOC T)	%535=08644000	D	535
IF B SC=DC THEN JUMP OUT TO L2 ELSE	%535=08645000	D	535
IF SC#"+" THEN JUMP OUT TO LO ELSE TALLY+TALLY+1);	%535=08646000	D	535
GO TO L3;	%535=08647000	D	535
L2: IF SC#"+" THEN GO TO LO;	%535=08648000	D	535
END FLSE	%535=08649000	D	535
L3: TALLY+48 ELSE	%535=08650000	D	535
BEGIN DI+LOC T; SI+SI+1;	%535=08651000	D	535
IF SC<"0" THEN BEGIN SI+SI-1; DS+1 OCT; FND	%535=08652000	D	535
ELSE BEGIN SI+SI-1; DS+2 OCT; END;	%535=08653000	D	535
TALLY+47; T(TALLY+TALLY+63);	%535=08654000	D	535
END;	%535=08655000	D	535
T+TALLY; SI+LOC T; DI+R; DS+WDS;	%535=08656000	D	535
\$ VOIDT	%560=08699150	D	560
150# % SEGMENTS PER ROW	%560=08699350	D	560
050# % PHYSICAL RECORDS PER ROW	%560=08699500	D	560
1000#% PHYSICAL RECORDS PER FILE	%534=08739000	D	534
IF SC=" " THEN GO L1; END;	%534=08739500	D	534
IF SC#"+" THEN BEGIN SI+SI+1; GO L5; END;	%534=08749500	D	534
IF SC#"+" THEN BEGIN TALLY+1; GO EXIT; END;	%155=08881000	D	155
SI:=rM[TYPEDSPACE(31,SHEETAREAV))] & 30[SIZE];%	%155=08930000	D	155
DLNKI=[M[TYPEDSPACE(31,SHEETAREAV))] & 30[SIZE];%	09506000	D	155
INTRNSC := [M[INTLOC:=GETSPACE(MAXINT+T,SIZE,INTARRAYAREAV,1)+2]] &	%202=09617000	D	202
BEGIN WAITSTORF(0); STOREDY(0,0);	%202=09624000	D	202
STOREDY(0,1);	%203=09645999	D	203
OKSEGZEROWRITE:=TRUE;	%203=09656001	D	203
OKSEGZEROWRITE:=FALSE;	%403=GRK 09680450	D	403
STREAM(BASE); BEGIN 3(DS+LIT"0"; DI+DI+7); END;	%203=09682609	D	203
OKSEGZEROWRITE:=TRUE;	%203=09682611	D	203
OKSEGZFROWRITE:=FALSE;	%151=09916000	D	151
IF T < @20000 OR T > @100000 THEN	%539=12528400	D	539
BEGIN RDCTABLE[UNIT].r[816]+0;	%539=12528600	D	539
END	%517=GRK 12530010	D	917
STREAM(MTX+TINU[UNIT],PRN+PRNTABLE[UNIT].[30:18],	%517=GRK 12530020	D	917
D+T+SPACE(10));	%517=GRK 12530030	D	917
BEGIN SI+LOC MTX; SI+SI+5; DS+8LIT"#PBT ON ";	%517 12530040	D	917
DS+3CHR; DS+7LIT" RW/L ("; DS+5DFC;	%517=GRK 12530050	D	917
DS+2LIT")+"; DI+DI+7; DS+4FILL;	%517=GRK 12530060	D	917
END STREAM;	%517=GRK 12530070	D	917
SPOUT(T);	%156=12561600	D	156
IF HEADER.[ICE] GFQ 64 THEN FORGETSPACE(HEADER);	%156=12572000	D	156
SVOIDT			

```

$VOIDT
    P(1)
    ELSE
        IF (NT2:=P).[2:1] THEN %LEFT AT 12626500
$VOIDT
    BFGIN IF (10D:=T.[9:24] DIV 5)#0 THEN %SPACE FORWARD
    B:=(GETSPACE(9,IOBUFFFRAREAV,1)+2) & 90[SIZE];%
    SPOUTIT(LOGINFO,
        64))
$VOIDT 12727501
$VOIDT 12739501
    LOGINFO:=SAVEARRAYDFSC(20,LOGAREAV);%
    RDCTABLE[V],[47:1]+INDEC[0]="FULLPGE"; % I.LINES6A OPTION
    IF INCW.[CF]<@1777 THEN % SOMETHING VERY WRONG
    BEGIN JAR[P1MIX,6].[1:1]+1; % SD BIT
        FILEMESS("SEE TSS", "PATCH ",0,0,0,0,602)%KLUGE MSG&DS14015220
    END;
$VOIDT 14218081
    IF AIT[AIT[0]].[8:10] N#0 1 THEN
    BEGIN
        I:=1;
        WHILE AIT[I].[8:10] = 1 DO I:=I+1;
        FOR J:=AIT[0] STEP -1 UNTIL I DO
            AIT[J+1]:=AIT[J];
        END ELSE I:=AIT[0]+1;
        AIT[0]:=*P(DUP) + 1;
        AIT[I]:=*(1&1[8:38:10]&M[C][FTF]);
        DEFINE HARRYSTA = WORKERSTACK[5]#, % STA
        REAL CPUIO:=+14;
    SUBROUTINE CLEARALLTANKS;
    BEGIN CLEARANK(LOGLINF,0);
        TANKS[I:=LOGLINE.[40:8]]:=(+P(DUP))&0[CTC];
        IF WORKING THEN % MAKE SURE HARRY ISNT
        IF HARRYSTA = I THEN % DIDDLING OUR LINE
            COMPLEXSLEEP(NOT WORKING OR HARRYSTA # I);
        CLFARINPUTANK;
        TANKOK[I] := 0;
    END OF CLEARALLTANKS;
        P(0); %ZERO OUT CPUIO
        IF LOGLINE.[33:7] # 0 AND P1MIX # CANDYINX THEN
            CLEARALLTANKS;
            I := 0; % RE-INITIALIZE FOR CORE SEARCH
            CLEARALLTANKS;
$VOIDT 14367035
    NT1:=GETSPACE(5,MAINTBUFFAREAV+64,0)+2;
    CPUIO+GETSPACE(10,0,1)+2;
    M[CPUIO+2]+VECTOR[0]; %JOB PREFIX
    M[CPUIO+3]+VECTOR[1]; %JOB SUFFIX
    M[CPUIO+4]+1; %JOB MIX INDEX
    M[CPUIO+5]+((S[1]+NEXTMOM)+30)DIV 60; %CPU TIME
    M[CPUIO+6]+((S[2]+VECTOR[4]+10T1MF[1])+30)DIV 60;
    M[CPUIO+7]+S[3]+(LC[1]-SC[1])xCHUNKZIFE+CHUNKZIFE;
    SPOUTIT(VECTOR INX 0,
        IF NOT ( (M[CPUIO+3] EQV "DISK ")#NOT 0 AND
        ( (M[CPUIO+2] EQV "PRNPBT ")#NOT 0 OR
        (M[CPUIO+2] EQV "LDCNTRL")#NOT 0))
        THEN ((LINK OR CANDYMESS) AND EOJMFS

```

```

D 156 %156=12573000
D 156 %156=12573500
D 156 %156=12574000
D 162 %162=12635500
D 162 %162=12638000
D 162 %162=12650500
D 155 %155=12672500
D 509 %509=GRK 12704000
D 509 %509=GRK 12704100
D 140 %140=12726500
D 140 %140=12728500
D 155 %155=12848500
D 724 %724=12860100
D 602 %602=14015200
D 602 %602=14015210
D 602 %602=14015220
D 602 %602=14015240
D 102 %102=14218071
D 102 %102=14218072
D 102 %102=14218074
D 102 %102=14218076
D 102 %102=14218078
D 102 %102=14218080
D 102 %102=14218082
D 102 %102=14218084
D 102 %102=14218086
D 102 %102=14218088
D 158 %158=14344100
D 504 %504=GRK 14351306
D 151 %151=14351440
D 151 %151=14351445
D 151 %151=14351450
D 151 %151=14351455
D 151 %151=14351460
D 151 %151=14351465
D 151 %151=14351470
D 151 %151=14351475
D 151 %151=14351480
D 504 %504=GRK 14355060
D 151 %151=14360510
D 151 %151=14360520
D 151 %151=14360530
D 151 %151=14367021
D 151 %151=14367022
D 155 %155=14411820
D 504 %504=GRK 14413020
D 504 %504=GRK 14413030
D 504 %504=GRK 14413040
D 504 %504=GRK 14413050
D 504 %504=GRK 14413060
D 504 %504=GRK 14413070
D 504 %504=GRK 14413080
D 509 %509=GRK 14430000
D 509 %509=GRK 14430002
D 509 %509=GRK 14430004
D 509 %509=GRK 14430006
D 509 %509=GRK 14430008

```

```

AND (NOT JAR9).E2:1] OR EOJK) %509=GRK 14430010 D 509
FLSF =FOJK) %509=GRK 14430012 D 509
STREAM(I,CPU10) %504=GRK 14430031 D 504
BFGIN DS+5LIT" FOR "; SI+CPU10; SI+SI+17; %504=GRK 14430032 D 504
DS+7CHR; DS=LIT"/"; SI+SI+1; DS+7CHR; %504=GRK 14430033 D 504
DS=LIT"="; DS+2DEC; I+DI; DI+DI-2; %504=GRK 14430034 D 504
DS+5FILL; DI+I; DS+5LIT"IPST="; DS+5DEC; %504=GRK 14430035 D 504
I+DI; DI+DI-5; DS+4FILL; DI+I; %504=GRK 14430036 D 504
DS+5LIT",IOT="; DS+5DEC; I+DI; DI+DI-5; %504=GRK 14430037 D 504
DS+4FILL; DI+I; DS+6LIT",CORE="; %504=GRK 14430038 D 504
DS+5DEC; I+DI; DI+DI-5; DS+4FILL; %504=GRK 14430039 D 504
DI+I; DS=LIT"+"; %504=GRK 14430040 D 504
END STREAM; %504=GRK 14430041 D 504
SPOUTER(CPU10,UNITNO,64); %504=GRK 14430042 D 504
IF UNITNO#0 THEN %504=GRK 14430043 D 504
TI=GETSPACE(12,CONTROLCARDAREAV+64,0)+4; %155-14533000 D 155
JAR[MIX,6],[1:1]+((TYPE=9) OR (TYPE=19)); %DS=8,SD=9,CL=19 %410 16212100 D 410
***X JAR[MIX,6],[1:1]+((TYPE=9) OR (TYPE=19)); %DS=8,SD=9,CL=19 %410 16216000 D 410
SCHFDLOOK(KTR,TYPE=7) % FS=6, XS=7 %104-16651000 D 104
SEG I= IF MER],AREATYPEF = CODEARFAV THEN M[R+1].PCF] ELSE 0; 16853000 D 155
ELSE P(M[R-2],AREATYPEF,DUP) # TYPE7INTAREAV AND P(XCH) # %155-16854200 D 155
TYPE13INTAREAV) OR LOCN = 0; %155-16854300 D 155
REAL BUFF, KTR, TYPE, MIX, A, I, J, K, T, %801-16951600 D 801
LABEL RESTART; %801-16951950 D 801
BUFF I= TYPEDSPACE(60,KEYINBUFFAREAV) + 1; %155-16954200 D 155
RESTART; %801-16954450 D 801
TYPE + KEYINSCAN(KTR,MIX); %801-16954502 D 801
IF (PROCFD+TYPE,1151)=1 AND ((I+TYPE,[CF])=8 OR I=VCC %801-16954504 D 801
OR I=33 OR I=34) OR (PROCFD=0 AND I=1) THEN ELSE % SS,CC,OC,FE,AX 16954506 D 801
BEGIN %801-16954508 D 801
STREAM(KTR,I); %801-16954510 D 801
BFGIN SI+KTR; %801-16954512 D 801
8(60(IF SC="" THEN %801-16954514 D 801
BFGIN LI SI+SI+1; %801-16954516 D 801
63(SI+SI+1) IF SC="" THEN JUMP OUT; %801-16954518 D 801
IF SC="" THEN JUMP OUT 3 TO YECH); 16954520 D 801
SI+SI+1; %801-16954522 D 801
IF SC="" THEN GO TO I; %801-16954524 D 801
END; %801-16954526 D 801
IF SC="" THEN JUMP OUT 2 TO YECH; %801-16954528 D 801
IF SC="" THEN %801-16954530 D 801
BEGIN %801-16954532 D 801
I+SI; TALLY+1; %801-16954534 D 801
DI+I; DS=LIT "+"; %801-16954536 D 801
X: SI+SI+1; IF SC="" THEN GO TO X; I+SI; %801-16954538 D 801
JUMP OUT 2 TO YECH; %801-16954540 D 801
END; SI+SI+1); %801-16954542 D 801
YECH: KTR+TALLY; %801-16954544 D 801
END STREAM; %801-16954546 D 801
I+P; IF P THEN %801-16954548 D 801
STREAM(I, T+T+GETSPACE(62,0,0)+3); %801-16954550 D 801
BEGIN %801-16954552 D 801
SI+I; %801-16954554 D 801
8(60(IF SC="" THEN JUMP OUT 2 TO L ELSE DS+CHR)); LIDS=LIT "+"; 16954556 D 801
END STREAM; %801-16954558 D 801
END CHECK FOR KEYIN RECYCLE; %801-16954560 D 801
IF PROCED=7 THEN GO TO TBLERR; %801-16954600 D 801

```

```

IF T>0 THEN % ANOTHER MESSAGE
BEGIN
  BUFF←T; T←0;
  GO RESTART;
END;
BEGIN T←VWY&VOK[3614216]&(VIF×A,[311])[3014216];
  WHILE NOT TOG DO
    COMPLEXSLFFP((CLOCK AND @17777)=0 OR T0G);
    SLFFP((CLOCK),NOT CLOCK);
    TEMP ← F & K [CTF];
    FH1=[MFI=TYPE SPACE(TMP,DISKHEADFRARFAV)] & TEMP[SIZE];%
    FH 1=[MFI=TYPE SPACE(30,DISKHEADFRARFAV)] & 30 SIZE;
    STREAM(A,B,T1=T1=GFTSPACE(10,CONTROLCARDAREAV+64,0)+4);
    T1=GETSPACE(I+5,CONTROLCARDAREAV+64,0)+4;
    LABEL PE,TE,PA,LL;
    SWITCH S := LL,PA,PE,TE,IT,US,D,TD,PR,IOT,TMR,AD,WD;
C1: IF (I4=14) ≥ (-6) AND I4 ≤ 6 THEN
  BFGIN GO TO S[I4+6];
LL: I4 := LOGLINE.[40:8]; GO TO INITIATE;
  PA;
$ SET OMIT = NOT(PACKETS)
  IF (I1=PSEUDOMIX[P1MIX]) GEQ 32 THEN
    I4:=PACKETACT[I-32];
$ POP OMIT
  GO TO INITIATE;
  PE;
$ SET OMIT = NOT(PACKETS)
  IF (I1=PSEUDOMIX[P1MIX]) GEQ 32 THEN
    BFGIN
    I4:=PACKETERR[I-32];
    PACKETERR[I-32]:=TRUE;
  END;
$ POP OMIT
  GO TO INITIATE;
  TE;
$ SET OMIT = NOT(PACKETS)
  IF (I1=PSEUDOMIX[P1MIX]) GEQ 32 THEN
    I4:=PACKETERR[I-32];
$ POP OMIT
  GO TO INITIATE;
  WD: I4←WFEKDAY; GO TO INITIATE;
  IF I4 = (-6) THEN I4←P1MIX;
  STREAM(P1MIX,T←R5×R6,A←I+SPACE(7));
  BEGIN DS←I.IT ","; SI←LOC P1MIX; %SPOUT ".,", NOT "#", %GRK=549 18790700
  REPLY[P1MIX],[CF] ← 3;
  J ← 1;
SVOIDT 18791701
  WAITSTORE(P1MIX); STOREDY(P1MIX,0);
  STOREDY(P1MIX,1);
WAITSTORE(P1MIX); STOREDY(P1MIX,0);
STOREDY(P1MIX,1); % FREE MEMORY TO ALLOW "ARTN" TO BE BROUGHT IN
  LABEL IPCERR;
  INIT,SLOW,C34,C35,C36,C37,TW,TW,C40,C41,IPCERR, 19508100
  IPCERR,IPCERR,IPCERR,IPCERR,IPCERR,TW,SLOW;
BEGIN SI=LOC A; DS=8 DEC; DS=2 LIT " ";
  DI:=DI-10; DS:=8 FILL; AI=DI;
  DI:=T; SI:=A; DS:=10 CHR;

```

```

%801-16969710 D 801
%801-16969720 D 801
%801-16969730 D 801
%801-16969740 D 801
%801-16969750 D 801
%519-GRK 18025000 D 519
%815-18031400 D 815
%815-18032000 D 815
%815-18032010 D 815
%148-18273055 D 148
%18285200 D 155
%155-18286000 D 155
%18438120 D 155
%155-18520300 D 155
%143-18701010 D 143
%143-18701100 D 143
%143-18710100 D 143
%143-18710200 D 143
%143-18710230 D 143
%136-18710240 D 136
%136-18710242 D 136
%136-18710244 D 136
%136-18710246 D 136
%136-18710248 D 136
%136-18710249 D 136
%136-18710250 D 136
%136-18710252 D 136
%136-18710254 D 136
%136-18710256 D 136
%136-18710258 D 136
%136-18710260 D 136
%136-18710262 D 136
%136-18710264 D 136
%136-18710266 D 136
%136-18710268 D 136
%136-18710270 D 136
%136-18710272 D 136
%136-18710274 D 136
%136-18710276 D 136
%136-18710278 D 136
%753-18711500 D 753
%753-18711690 D 753
%549-18790600 D 549
%549-18790700 D 549
%549-18791200 D 549
%549-18791300 D 549
%549-18791400 D 549
%202-18841400 D 202
%202-18843500 D 202
%202-18843800 D 202
%202-18845000 D 202
%145-19504100 D 145
%19508100 D 145
%145-19508200 D 145
%149-19559120 D 149
%149-19559140 D 149
%149-19559160 D 149

```

```

      TWXOUT(T,10,0&1[CTF]&1[2:47:1],R6);
      IPCFRR: TERMINATE (P1MIX);
      TERMINALMESSAGE (96);
      S := [MITYPFDSPACE(31,SHEETAREAV)] & 30[8:38:10];%
      HDR := GETSPACE(30,DISKHEADFRAREAV,1)+2;
$ SET OMIT = PACKETS
      FLSE FORGETSPACE(HDR);
$ POP OMIT
      FPB := TYPEDSPACE(SFG0[5] INX 1, FPBAREAV);
      FR := GETSPACE(SEG0[7],[CF]xETRLNG,FPBAREAV,1)+2;
      LBL := ARRAYDESC(30,LBLFQNAREAV);%
      LBL := ARRAYDESC(30,LBLEQNAREAV);%
      SPOUTER(T,UNITNO,
      IF NOT ( (JAR[MIX,1] EQV "DISK ")=NOT 0 AND
      ( (JAR[MIX,0] EQV "PRNPRT ")=NOT 0 OR
      (JAR[MIX,0] EQV "LDCTRL")=NOT 0 ) )
      THEN ((NOT S[0]),[2:1] OR CANDYMESS)
      AND BOJMESS AND I OR BOJK) ELSE "BOJK");
      SEG0 := ARRAYDESC(30,SEGZEROAREAV);%
      SINFO[MIX],[CF] := (STACKLOC :=
      GETSPACE(SEG0[3] INX S[2] INX 64, STACKPRTAREAV, 1)) + 2;
      WAITSTORE(MIX); STOREDY(MIX,0);
      STOREDY(MIX, 1);
      TRP[4] := [MITI := GETSPACE(SEG0[1],[CF],SFGNICTAREAV,1)+2];%
      STRFAM[D] := DALOCROW[MIX] := SAVEARRAYDESC(DALOC SIZE,DALOCROWAREAV);%
$ VOID
      STREAM(D := UVSPACE := TYPEDSPACE(UVSIZE,UVROWAREAV));
      BELOW := (S[2],[8:10]=5) OR ITSALLBATCH;
      M[HDR=2] := (*P(DUP)) & JARROWAREAV AREATYPEF; %
      STREAM(Q := FSROW[MIX] := SAVEARRAYDESC(4,FSAREAV));%
      TRP[3] := [M[GETSPACE(1,FPBAREAV,1)+2]] ELSE %
      FORM = 75%;% SWITCH D(PCC)"FORM"="SPECIAL"
      LINES66 = 77%; % 66 LINES PER PAGE
      BEGIN LABEL TRYAGAIN;
      HEADER[2] := HEADER[5] := MCP;
      HEADER[4] := 0 & (@1001)[2:38:10] & SYSNO[4:46:2];
      SI := CARD; DS := 9 WDS;
      M[BUF+87] := MCP;
$
      X := 6; M[BUF+17] := 0;
      STREAM(CARDLOC);
      BEGIN SI := CARDLOC;
      2(36(IF SC=">" THEN
      BEGIN CARDLOC+SI;DI+CARDLOC;DS+ LIT "=" END;
      IF SC=">" THEN
      BEGIN CARDLOC+SI;DI+CARDLOC;DS+ LIT "" END;
      SI+SI+1));
      END;
      (IF KOUNT = "" THEN EQUAL FLSE & THIS IS AS IN SYMBOL
      LDUMMY,
      LLINES66, %
      LRANDOM,
      SWITCH D=LFORM,
      LDUMMY,
      LLINES66,
      LRANDOM,
      LNO,LDISK,LTAPE,LPUNCH,LPRINT,LPAPER,

```

```

%149-19559200 D 149
%145-19575000 D 145
%145-19575100 D 145
%155-20026600 D 155
%155-20028700 D 155
%163-20034060 D 163
%163-20034070 D 163
%163-20034080 D 163
%155-20049000 D 155
%155-20049800 D 155
%155-20059100 D 155
%155-20089800 D 155
%509-GRK 20104200 D 509
%509-GRK 20104300 D 509
%509-GRK 20104310 D 509
%509-GRK 20104320 D 509
%509-GRK 20104330 D 509
%509-GRK 20104340 D 509
%155-20110400 D 155
%155-20116100 D 155
20116200 D 155
%202-20119300 D 202
%202-20120300 D 202
%155-20124900 D 155
20127300 D 155
%155-20127400 D 155
%155-20156800 D 155
%151-20163200 D 151
%155-20169700 D 155
%155-20183610 D 155
%155-20187400 D 155
%846-20240000 D 846
%724-20240020 D 724
%172-20289030 D 172
%129-20289113 D 129
%107-20289115 D 107
%129-20289240 D 129
%129-20289252 D 129
%172-20289330 D 172
%172-20289335 D 172
%890-20374401 D 890
%890-20374402 D 890
%890-20374403 D 890
%890-20374404 D 890
%890-20374405 D 890
%890-20374406 D 890
%890-20374407 D 890
%890-20374408 D 890
20380000 D 890
%846-20392870 D 846
%724-20392880 D 724
%603-20392890 D 603
%603-20394000 D 603
%846-20394070 D 846
%724-20394080 D 724
%603-20394090 D 603
%603-20394900 D 603

```



```

LDUMMY:   TPNO+11; % " FORM SPO" = DUMMY FILE           %846=20451900  D  846
          TPNO+12; GO TO DSKCHECK; % "DISK" MEANS DISK SERIAL %603=20457000  D  603
          BFGIN TPNO+21; IF TYPE=COPYN THEN BEGIN        %603=20461300  D  603
            TPNO+22; GO LCOPY                             END ELSE %603=20461310  D  603
            IF SCAN#BACK THEN GO ERROR;                 %603=20461320  D  603
          END;                                           %603=20461330  D  603
          IF TYPE=COPYN THEN BFGIN TPNO+22; GO LCOPY END ELSE %603=20461810  D  603
LLINES66: %                                             %724=20465050  D  724
          EQN[0] + "FULLPGF" ; %SET UP MFID FOR FULL PAGE %724=20465100  D  724
          BFGIN TPNO+4; IF TYPE=COPYN THEN BEGIN        %603=20470000  D  603
            TPNO+15; GO TO LCOPY                         END ELSE %603=20470100  D  603
            IF SCAN#BACK THEN GO ERROR;                 %603=20470110  D  603
          END;                                           %603=20470120  D  603
          IF TYPE=COPYN THEN BEGIN TPNO+16; GO LCOPY END ELSE %603=20475100  D  603
LRANDOM: TPNO+10; GO TO DSKCHECK;                      %603=20481900  D  603
          ELSE IF TYPE=COREV THEN                       %512=20504500  D  512
            BFGIN X[20] + TPNO DIV 64;                  %512=20505000  D  512
            DO UNTIL (IOD + SCAN)=MAXV OR IOD=PERIOD;   %512=20507000  D  512
            IF IOD=MAXV THEN P([X[20]],IOD) ELSE GO TO DOWN; %512=20507100  D  512
            END                                           %512=20507200  D  512
          ELSE IF TYPE>=PRIOR AND TYPE<=SAVEV THEN     %512=20507300  D  512
            X[18+TYPE-PRIOR]+TPNO ELSE GO TO FRROR;    %512=20507400  D  512
REAL FR1,ER2,ER3; LABEL ERSYS;                          %164=20511152  D  164
          ER1=")"; %                                     %164=20511175  D  164
          OR SYSTEMFILE(CMM[0],CMM[1]) THEN%           %164=20511188  D  164
          BEGIN ERSYS; ER1="SYSTEM ";ER2="FILE)+ "; GO ERROR END; %164=20511190  D  164
          BEGIN FR1="SAME FI";ER2="LE)+ "; GO ERROR END %164=20511303  D  164
          IF TYPE=USEV AND M[OPTN+2]<0 THEN%           %164=20511312  D  164
          BEGIN FR1="SECURIT";ER2="Y FILE)";FR3=")"; GO ERR END; %20511313  D  164
          BEGIN FR1="INVALID";ER2=" USER)+"; %        %164=20511335  D  164
FRR:      FORGETSPACE(OPTN); %                          %164=20511340  D  164
          END ELSE IF OPTN=2 THEN GO ERSYS% @ LINE 20511190 %164=20511363  D  164
          ELSE IF OPTN=1 THEN BEGIN ER1="IN USE)";ER2=")"; % " END 20511364  D  164
          ELSE IF OPTN=0 THEN BEGIN ER1="NOT ON ";ER2="DISK)+ " END; %20511365  D  164
ERROR:     %164=20511366  D  164
          STREAM(AI=[CMM[0]],ER1=[ER1],BI=(OPTN=SPACE(10))); %164=20511370  D  164
          DSI=25 LIT " SECURITY MAINT IGNORED ("; SI=ER); %164=20511400  D  164
          3(SI=SI+1; DSI=7 CHR); %                      %164=20511405  D  164
SVOINT    %121=20566800  D  121
          IF T GEQ COPYN AND T LEQ LOAD THEN           %543=20566807  D  543
            OPTN=CN; PROG[29]=0;                       %121=20566817  D  121
          IF T GEQ COPYN AND T LEQ LOAD THEN           %543=20566822  D  543
          BFGIN IF XLST=0 THEN XLST1=FM[SPACE(XLSTSZ=30)1]&30[8138110]; %170=20567005  D  170
            T1.[4611]+HOLD3+(CN=EQUAL);                %543=20569930  D  543
            IF HOLD3 THEN PROG[CNT]=1 ELSE             %543=20569935  D  543
            IF CN GEQ IDENT THEN PROG[CNT]=ACCUM[0] ELSE GO POWIE; %543=20569940  D  543
            T1.[4711]+HOLD3+((CN+SCAN)=EQUAL);        %543=20569960  D  543
            IF HOLD3 THEN PROG[CNT+1]=1 ELSE           %543=20569965  D  543
            FORGETSPACE(XLST); XLST=0;                 %543=20570070  D  543
          IF (CN+SCAN)=EQUAL THEN IF T1.[4611] THEN   %543=20570130  D  543
          ELSE IF CN GEQ IDENT THEN IF T1.[4611] THEN GO POWIE %543=20570150  D  543
          IF (CN+SCAN)=EQUAL THEN IF T1 THEN          %543=20570190  D  543
          ELSE IF CN GEQ IDENT THEN IF T1 THEN GO POWIE %543=20570210  D  543
          STREAM(AI=TOHLD,BI=TOHLD.[4216]=1,C=TOHLD.[4216]#0, %123=20571700  D  123
            DI=[CMM[27]]); %                            %123=20571702  D  123
          BEGIN                                         %123=20571710  D  123
            SI=LOC A; SKIP SB;                          %123=20571720  D  123

```

IF SB THEN	%123-20571730	D	123
BEGIN	%123-20571740	D	123
SKIP 39 SR;	%123-20571750	D	123
IF SB THEN DSI=8 LIT"OFAST "	%123-20571760	D	123
ELSE	%123-20571770	D	123
BEGIN	%123-20571780	D	123
SKIP SR;	%123-20571782	D	123
IF SR THEN DSI=8 LIT"OSLOW "	%123-20571790	D	123
ELSE	%123-20571800	D	123
BEGIN	%123-20571810	D	123
C(SI)=LOC B;	%123-20571820	D	123
DSI=6 LIT"OEU # "; DSI=2 DEC;	%123-20571830	D	123
JUMP OUT TO L);	%123-20571840	D	123
DSI=8 LIT"ODISK "	%123-20571842	D	123
L1	%123-20571850	D	123
END	%123-20571860	D	123
END	%123-20571870	D	123
ELSE	%123-20571880	D	123
BEGIN	%123-20571890	D	123
SKIP 5 SR;	%123-20571900	D	123
DSI=LIT"O"; DSI=7 CHR;	%123-20571910	D	123
END;	%123-20571920	D	123
END;	%123-20571930	D	123
IF XLST NEQ 0 THEN BEGIN FORGETSPACE(XLST); XLST:=0; END;	%543-20573350	D	543
IF T GEQ 64 THEN	%508-20574917	D	508
T=DIRECTORYSEARCH(CMM[0],CMM[1])&(P(UNITNO,DUP)=25	OR 20577826	D	508
P(XC)=30)(1:47:1),4); END;	%508-20577827	D	508
IF T GEQ 64 THEN	%508-20577828	D	508
CCOMPILE=0;	%510-20584340	D	510
CCOMPILE=0; % INITIALIZE	%688-20584390	D	688
CCUNIT=0; % INITIALIZE	%688-20590030	D	688
BEGIN CARD.[33:15] := TYPEDSPACE(13,CONTROLCARDAREAV)+2;	20600950	D	155
IF LOGLINE = 0 THEN LOGLINE = UNITNO;	%109-20602502	D	109
IF PKTONLY THEN	%106-20602515	D	106
PPCPRCFSSI=0; %	%169-20602855	D	169
PROCVAl=0;	%126-20602860	D	126
CONTROLA: IF (T < FILEV OR T > COBOL) AND ACCUM[0] # CMPLR THEN	%527-20604300	D	527
IF T GEQ UNLOCKV AND T LEQ RESETV THEN GO TO FINIS	%527-20604350	D	527
USFRS: IF (TI=SCAN)#EQUAL THEN GO INCSC;	%131-20605700	D	131
IF (TI=SCAN)=PERIO THEN GO INCSC;	%131-20605702	D	131
P(DIRECTORYSEARCH(NABS(CMM[0]),IF CMM[0]<0 THEN "DISK " ELSE	20606865	D	407
IF UNITNO>32 THEN CIDROW[UNITNO-32].[3:5]:=0 ELSE%	%169-20606900	D	169
IF UNITNO=23 THEN READFR.[FF]:=0 ELSE%	%169-20606910	D	169
IF UNITNO=24 THEN READFR.[FF]:=0; %	%169-20606920	D	169
IF (T>PACKET) AND (T<RESETV) AND (T#RUNV) THEN	%527-20608078	D	527
IF ZIPMIX NEQ 0 THEN	%138-20608112	D	138
IF (TI=PSFUDOMIX[ZIPMIX]) GEQ 32 THEN	%138-20608114	D	138
PACKETERR(T=32)=TRUE;	%138-20608116	D	138
IF (TI=SCAN) = ENDFI THEN	%127-20608740	D	127
GO TO PACK2;	%127-20608760	D	127
\$ VOIDT	%127-20608800	D	127
\$ SET OMIT = NOT(PACKETS)	%106-20609425	D	106
IF PSEUDO(UNITNO=32) = 0 THEN	%106-20609430	D	106
PRINTTHECOVER(CARDLOC,UNITNO,PSOURCE);	%106-20609435	D	106
\$ POP OMIT	%106-20609440	D	106
BOOLEAN FT=N; DEFINE FH(FH1)=M[T+FH1]; % RESET FILE A/B	%815-20704100	D	815
SUBROUTINE CLEARTHEFILE; % CLEAR AN IN=USE FILE	%815-20704200	D	815

```

BEGIN
    FH[4],[01:06]+0;    % EXCLUSIVE
    FH[4],[16:20]+0;    % OPEN COUNT 2
    FH[9],[01:28]+0;    % TOGS & OPEN COUNT 1
    DISKWAIT(T,[CF],30,T,[FF]);    % FIX IT
    FILEHOLD(CMM[2],CMM[3],0,T,0);    % WAKE UP WAITING PROCESSES
    LBMESS(CMM[2],CMM[3],11,26,0,SPOUTUNIT,1);
END CLFARTHFILE;
    CCSFT+0;
    IF CN#ACCESSD THEN
        IF NOT (FT#FXTOG+(CN#FILEV)) THEN GO TO CCERR;
        BEGIN T+2; GO TO L1; END;
        BEGIN FORGETSPACE(T); T+1; GO SKIP; END;
$ VOIDT
$ VOIDT
$ VOIDT
    END ELSE IF FT THEN CLEARTHEFILE ELSE BEGIN
$ SET OMIT = SHAREDISK
        UNLOCKDIRECTORY;
$ POP OMIT
        LBMESS(CMM[2],CMM[3],(11+TOG),41,0,SPOUTUNIT,1);
        END;
        FORGETSPACE(T);
        ELSE BEGIN
$ SET OMIT = SHAREDISK
        UNLOCKDIRECTORY;
$ POP OMIT
        END;
$VOIDT 20772301
        FIELD MAYBEWORKEDON = [7:1];
        LABEL START;
        X := @104;    % THIS IS THE CODE WE WILL PASS TO
                    % GETSPACE THE FIRST TIME. IT REQUESTS
                    % OVERLAY MEMORY FOR THE MCP AND THAT
                    % WE WANT TO BE RETURNED TO ON A NO
                    % MEM.
START;
        IF NOT M[S],MAYBEWORKEDON THEN % WAIT TILL OTHER GUY IS D021016000
            SLEEP([M[S]],0 & TRUE [MAYBEWORKEDON]);
        BEGIN
            Z := Y.[8:10]; % SIZE OF PROCEDURE IN WORDS
            M[S],MAYBEWORKEDON := FALSE;
            NOW WE WILL ATTEMPT TO GET SPACE FOR THIS MCP PROC.
            IF WE FAIL WE WILL WAIT FOR A SECOND AND THEN TRY
            AGAIN, THIS ENSURES THAT IF WE GET DS=ED WHILE
            SLEEPING WAITING FOR MEMORY WE WILL NOT LEAVE THE
            TOGGLE LOCKED UP FOR THIS PROCEDURE.
            IF (T:=GETSPACE(Z,65,X)+2) = 2 THEN % NO MEM
                BEGIN
                    M[S],MAYBEWORKEDON := TRUE;
                    X,[46:1] := TRUE; % DONT PUT OUT MESSAGE
                    SLEEP([CLOCK],NOT CLOCK); % WAIT A SECOND
                    GO TO START;
                END;
            M[S] := (*P(DUP)) & T [CTC] & TRUE [MAYBEWORKEDON];

```

```

%815-20704210 D 815
%815-20704220 D 815
%815-20704230 D 815
%815-20704240 D 815
%815-20704250 D 815
%815-20704260 D 815
%815-20704270 D 815
%815-20704280 D 815
%510-20711100 D 510
%815-20714000 D 815
%815-20714500 D 815
%521-20730000 D 521
%521-20739000 D 521
%521-20739100 D 521
%521-20739200 D 521
%815-20768000 D 815
%815-20769000 D 815
%521-20769100 D 521
%521-20769200 D 521
%521-20769300 D 521
%521-20769400 D 521
%521-20769500 D 521
%521-20769600 D 521
%521-20771000 D 521
%521-20771010 D 521
%521-20771020 D 521
%521-20771030 D 521
%521-20772050 D 521
%521-20772100 D 521
%153-21015100 D 153
%153-21015200 D 153
%153-21015400 D 153
%153-21015500 D 153
%153-21015600 D 153
%153-21015700 D 153
%153-21015800 D 153
%153-21015900 D 153
%153-21016000 D 153
%153-21017000 D 153
%153-21018100 D 153
%153-21018200 D 153
%153-21018300 D 153
%153-21018400 D 153
%153-21018500 D 153
%153-21018600 D 153
%153-21018700 D 153
%153-21018800 D 153
%153-21018900 D 153
%153-21019000 D 153
%153-21019100 D 153
%153-21019200 D 153
%153-21019300 D 153
%153-21019400 D 153
%153-21019500 D 153
%153-21019600 D 153
%153-21020000 D 153
%153-21024000 D 153

```

```

STOREDY(0,0); %202=21032000 D 202
STOREDY(0,1); %202=21093000 D 202
NT1=(SLN[MIX]*30 + COUNT[MIX]*14 - NT1)*8 + 208; %991=21111540 D 991
IF PRTRW[MIX].[PSF]=3 THEN PRTRW[MIX].[PSF]=0; %UCSC KLUDGE 21212010 D 991
I1=GETSPACE(30,JARROWAREAV&MIX[CTF],1)+2; %155=21304000 D 155
I2=GETSPACE(UVSIZEF,UVROWAREAV&MIX[CTF],1)+2; %155=21308000 D 155
TIMELIMITMAX + 3600 * S + 18000 ; S=0; %401=GRK 22013130 D 401
SHEAT I=[MFFI=TYPEDSPACE(31,SHEETAREAV+64)] & 30[8:38:10];% 22069050 D 155
SEGO I= TYPEDSPACE(30,SEGZEROAREAV+64);% %155=22069100 D 155
SHEAT[24]=MCP; %129=22069212 D 129
DO UNTIL (T1+WAITIO(AREA INX @340000012,@75,U))=0; 22112000 D 697
IF T1.[43:2]#0 THEN T1+WAITIO(@4200000000,5,U); %697=22112100 D 697
%992=22199000 D 992
%992=22201200 D 992
%992=22201400 D 992
%992=22201600 D 992
%155=22297000 D 155
%155=22303000 D 155
%155=22303100 D 155
22356100 D 155
%155=22386000 D 155
%155=22386050 D 155
%155=22386100 D 155
%152=23399000 D 152
%147=24035300 D 147
%147=24096800 D 147
%153=24304100 D 153
%153=24304110 D 153
%153=24304120 D 153
%153=24304130 D 153
24304140 D 153
24304150 D 153
24304170 D 153
24304180 D 153
%153=24304190 D 153
%153=24304200 D 153
%153=24305700 D 153
%153=24305750 D 153
%202=24309000 D 202
%202=24313000 D 202
%153=24315000 D 153
%153=24317000 D 153
24400000 D 155
%153=24401000 D 153
%119=27990410 D 119
%119=27993900 D 119
%119=27994200 D 119
%119=27994210 D 119
%119=27994220 D 119
%119=27994230 D 119
%119=27995310 D 119
%119=27995600 D 119
%119=27996300 D 119
27996505 D 119
%119=27996510 D 119
%119=27996515 D 119
%119=27996520 D 119

```

S VOIDT

SVOIDT

%  
%  
%  
%

FIELDS OF SAVEF PARAMETER

```

FIELD
NEEDOLAY = 09:39 % THIS FIELD IS NON ZERO IF 24304140 D 153
% REQUESTOR WANTS OLAY SPACE 24304150 D 153
TYPENOMEMANDRETURN = 45:01 % IF SAVFF.[45:02] = 3 THEN 24304170 D 153
JUSTRETURN = 46:01 % GETSPACE WILL TYPE OK MEM 24304180 D 153
GETFROMFRONT = 47:01 % IF SUCESSFULL. %153=24304190 D 153

```

AGAIN:

REAL BVD,UBLKSZ,SPB;

IF HDRTYPE GEQ 2 THEN % FIGURE OUT EOF USING B6700 FORMULA

REGIN

```

      BEGIN S1:=LOC W; D1:=LOC B; D1:=D1+4; SKIP 4 DB;
      20;IF SB THEN DS:=SET ELSE DS:=RFSFT; SKIP 1 SB);
    END;
    BVD:=P; % BITS OF VALID DATA IN LAST SEGMENT
    IF H[2],[8:1]
    THEN IF (NT2:=H[2],[5:3])=2
      THEN BEGIN NT1:=12; NT4:=3; END
      ELSE IF NT2=4
        THEN BEGIN NT1:=6; NT4:=7; END
        ELSE BEGIN NT1:=8; NT4:=5; END
      ELSE BEGIN NT1:=1; NT4:=47; END;
    NT2:=((H[14],[20:28] MOD SPB)*NT1*30)
    +((BVD + NT4) DIV (48 DIV NT1));
    NT1:=IF NT2 LSS UBLK SZ THEN NT2 ELSE UBLK SZ;
    NT3:=NT1+(H[14],[20:28] DIV SPB)*NT1*(NT4:=H[3],[32:16]);
    END;
    RCL[7]:=IF HDRTYPE GEQ 2 THEN (NT2+NT4-1) DIV NT4 - 1
      ELSE H[4];
    D1:=LOC A; S1:=LOC H; SKIP 1 DB; DS:=SFT;
SVO1DT
  BEGIN TMP:=GETSPACE(FPBPTR+ETRLNG,FPBAREAV,1)+2;
  H:=M[MTYPEDSPACE(42,DISKHEADERAREAV)] & 30[8:38:10];
  M[FAINFO+J],[8:1]=0; % DONT REMOVE IF UNLOAD
  P(M[FA+1],M[FA]); % SAVE NAME FOR UNLOAD
  NT1:=P; M[FA]=NT1; M[FA+1]=P(XCH); % REPLACE NAMES
  EXIT: P(P&RCW[CTC],0,RD S,0,XCH,P&P[CTF],STF);
%   .[13:1] FILE TO BE ADDED (NOT ON DISK)
  MUSTADD = DA,[13:1]#
  IF FASZ GTR 0 THEN LIBRARYHELP(5) ELSE IU:=OU:=(FPBPTR:=0)-1;
SVO1DT 28265801
%*****
% CHECK IF FILE IS TO BE ADDED
%*****
%
  IF ADDV THEN
  IF NOT MUSTADD THEN
  IF OU = 18 THEN% TAPE TO DISK
  IF DIRECTORYSEARCH(=MFID,FID,5) NEQ 0% ALREADY ON DISK
  THEN BEGIN
    LRMFSS(ARS(MFID),FID,67,68,T,N[UIU],SPOUTUNIT,LIBMSG);
    IF DSED THEN ABORT;
    GO NEXT;
  END
  ELSE BEGIN M[FAINFO+J],[13:1]=1; MUSTADD=1 END;
  H:=H&41[8:38:10];
  IF MUSTADD THEN T:=0 ELSE% ADD FILE NOT ON DISK
    T:=GETSPACE(30,DISKHEADERAREAV,1)+2;
  IF OU GEQ 0 THEN % WE HAVE AN OUTPUT UNIT
  IF IU GEQ 0 THEN % WE HAVE AN INPUT UNIT
  IF OU GEQ 0 THEN % WE HAVE AN OUTPUT UNIT
  TMP:=FASZ DIV 2 - 1;
  FOR J:=0 STEP 1 UNTIL TMP DO
  IF (DA:=M[FAINFO+J]),[CF]=18 THEN % FROM DISK
  IF UNLOAD THEN
  BEGIN MFID:=M[FA+J*2]; FID:=M[FA+1+J*2];
  P(DIRECTORYSEARCH(=MFID&1[3:47:1],FID,7),DEL);
  IF DSED THEN GO TO INITIATE;

```

```

%119-27996525 D 119
%119-27996530 D 119
%119-27996535 D 119
%119-27996540 D 119
%119-27996545 D 119
%119-27996550 D 119
%119-27996555 D 119
%119-27996560 D 119
%119-27996565 D 119
%119-27996570 D 119
%119-27996575 D 119
%119-27996580 D 119
%119-27996585 D 119
%119-27996590 D 119
%119-27996595 D 119
%119-27996600 D 119
%119-27996605 D 119
%119-27996610 D 119
%121-27996675 D 121
%119-27996800 D 119
%155-28050800 D 155
%155-28070000 D 155
%757-28093900 D 757
%757-28094500 D 757
%757-28097100 D 757
%160-28101200 D 160
%166-28200061 D 166
%166-28208300 D 166
%132-28246200 D 132
%757-28265600 D 757
%166-28270806 D 166
%166-28270807 D 166
%166-28270808 D 166
%166-28270809 D 166
%166-28270810 D 166
%166-28270812 D 166
%166-28270814 D 166
%166-28270816 D 166
%166-28270818 D 166
%166-28270820 D 166
%166-28270822 D 166
%166-28270826 D 166
%166-28270828 D 166
%166-28270830 D 166
%119-28274200 D 119
%166-28281550 D 166
%155-28288000 D 155
%128-28304590 D 128
%128-28304700 D 128
%128-28306500 D 128
%757-28307700 D 757
%757-28307705 D 757
%757-28307710 D 757
%757-28307715 D 757
%757-28307720 D 757
%757-28307725 D 757
%757-28307730 D 757

```

```

END;
AND ((MFID EQV "RADISK ") NFO NOT 0)
    TMPI=GETSPACE(12,CONTROLCARDAREAV+64,5)+4;
ELSE IF U=18 THEN P(DIRECTORYSEARCH("MFID,FID,13),DEL);
LAB:=ARRAYDESC(15,LABELAREAV);%
FA:=TYPEDSPACE(TMP:=TMP+(TMP DIV 2)+2,DATAAREAV)+1;
CCA:=SAVEARRAYDESC(30,ESPDISKAREAV);%
FA:=TYPEDSPACE((BUMPFA:=FASZ:=IF MAX > 128 THEN 128 ELSE MAX)+
    (FASZ DIV 2)+2,DATAAREAV)+1;
$ SFT OMIT = NOT R6500LOAD
    IF R6500 AND (NOT DFSTIN.[1:1]) THEN
    BEGIN STREAM(TMP:=TMP+SPACE(10));
        DS:=32LIT" #B6700 TAPE TO TAPE NOT ALLOWED.";
        SPOUT(TMP);
        ABORT;
    END;
$ POP OMIT
S:=ARRAYDESC(30,ESPDISKAREAV);%
WAITSTORE(P1MIX); STOREDY(P1MIX,0);
STOREDY(P1MIX,1);
    IF T3.[15:15]#0 THEN % DONT USE NONEXISTENT FIB
        RDCTABLE[U]+P(DUP,L0D)&REEL.[14:13:10]
            &CDATE[24:31:17]&CYCLE[41:41:7];
    IF PRNTABLE[U].[1:1] THEN ELSE
        BEGIN LABELTABLE[U]:=(+P(DUP));GO TO SOMEWHERE;END;
SVOIDT
    IF L0=23 THEN BEGIN L0:=27; HI:=28; END ELSE
    IF L0=27 THEN BEGIN L0:=0; HI:=15; END ELSE GO TO DUN;
    IF [MEM[P1MIX,MLINK]].[CF] GEQ FENCE THEN
SVOIDT
SVOIDT
SVOIDT
    IF COUNT <= 0 THEN GO TO START;
$ VOID
    [12:30:18], "RETRIES",FPB[FN],FPR[FN+1],J,0,0;
    IF U < 16 THEN FPR[FN+3].[6:17]+PRNTABLE[U].[31:17];
BEGIN MIALPHA+1]+(CURLOC+GETSPACE(BSIZE+4,IOBUFFRAREAV,1)+2)+2;
SVOIDT 38024081
MIT:=ALPHA-2;:=M OR (GETSPACE((T1:=MIT),SIZE)+4,
    LABELAREAV,1)+4) & T1[SIZE] & CNTRLBITS[FTF];38107100
BEGIN NT3 := TYPEDSPACE(13,LABELAREAV)+2;
    IF MODF := (MODE=0) AND BLEN = 20 THEN
        SAVWORD := SAVWORD OR TWO(U);
        CNTCTL := DIREC := 0;
MIT:=ALPHA-2;:=M OR (GETSPACE((T1:=MIT),SIZE)+4,
    LABELAREAV,1)+4) & T1[SIZE] & CNTRLBITS[FTF];38205100
    IF KIND=7 THEN FPB[FNUM+3]+(+P(DUP))&T2[15:15:8]
        &0[6:139:9];
RFL+RDCTABLE[U].[14:10]; % GET ACTUAL REEL NUMBER
    IF KIND=7 AND FIB[13].[28:10]#ABS(COBL) THEN GO FIND ELSE
    IF KIND=1 THEN MIALPHA-2]+P(DUP,L0D) &
        (IF SEPARATE THEN 1 ELSE @20)[27:42:6]
    ELSE
        IF NOT SEPARATE THEN
            IF (TYPE<20) THEN
                HEADER[73]+P(DUP,L0D)&(@20)[27:42:6];
LABEL EOF, CLEANUP;

```

```

%757=28307735 D 757
%165=28415250 D 165
%155=28435000 D 155
%134=28446300 D 134
%155=28451800 D 155
%155=28452600 D 155
%155=28455400 D 155
%155=28456800 D 155
%155=28457000 D 155
%121=28475605 D 121
%121=28475610 D 121
%121=28475620 D 121
%121=28475630 D 121
%121=28475640 D 121
%121=28475650 D 121
%121=28475660 D 121
%121=28475665 D 121
%155=28855000 D 155
%202=29544000 D 202
%202=29548000 D 202
%548=37027500 D 548
%745=37046177 D 745
%745=37046178 D 745
%764=37053300 D 764
%764=37053400 D 764
%160=37180675 D 160
%903=37195375 D 903
%903=37195400 D 903
%160=37226100 D 160
%160=37226150 D 160
%160=37226500 D 160
%160=37227500 D 160
%118=37240050 D 118
%114=37284150 D 114
%715=37285320 D 715
%114=37285410 D 114
37406000 D 155
%116=38024010 D 116
%155=38107000 D 155
%155=38107100 D 155
%155=38107300 D 155
%301=38107610 D 301
%301=38107620 D 301
%301=38107700 D 301
%155=38205000 D 155
%155=38205100 D 155
%124=38228510 D 124
%124=38228520 D 124
%745=38247100 D 745
38272000 D 167
%140=38280500 D 140
%140=38280600 D 140
%140=38280700 D 140
%140=38281600 D 140
%140=38281700 D 140
%140=38281800 D 140
%146=38363000 D 146

```

```

DEFINE H = HEADER#;
SVOIDT 38477001
    SFG0:=([M[TYPE]FDSPACF(62,SFGZFR0AREAV)]&30[SIZE])38434000
    BEGIN % COMPILE TO LIBRARY
        FOR T1 := 15 STEP 1 UNTIL 22 DO % COPY PROC, I/O,
            SEGO[T1] := SKEL[T1]; % STACK, COMMON, PRIO., ETC.
            IF (T2 := SKEL[13]) # 0 THEN % LAB. EQN, ENTRIES
                BEGIN
                    SKEL[13] := 0; % SO TERMINATE WONT SEE FSPDISK STUFF
                    DISKWAIT(SKFL,[CF],30,T3); % REWRITE SKFL. SHEET
                    SEGO[15] := H[7] + 1; % ADDRESS OF 1ST ENTRY
                    DO
                        BEGIN % COPYING LABEL EQN. INFO INTO CODE FILE
                            IF DISKADDRSS(MID,FID,FPB[FNUM+3],
                                (T1:=H[7];H[7]+1),H,0) = 0 THEN % NO ROOM
                                BEGIN
                                    DO
                                        BEGIN
                                            DISKWAIT(=SKEL,[CF],30,T2); % GET SEG,
                                            FORGETFSPDISK(T2);
                                        END
                                        UNTIL (T2 := SKFL[29]) = 0;
                                        FILEMESS(="DISK ", "OVRFLOW",MID,FID,
                                            R,D,C);
                                    END
                                ELSE
                                    BEGIN
                                        DISKWAIT(=SKEL,[CF],30,T2); % GET SEG,
                                        FORGETFSPDISK(T2);
                                        IF (T2 := SKEL[29]) # 0 THEN % MORE OF EM
                                            SKEL[29] := T1 + 1; % ADDR. IN CODE FILE
                                            DISKWAIT(SKEL,[CF],30,1:=(H[T1 DIV H[8]]
                                                +10) + T1 MOD H[8]));
                                        END
                                    END
                                UNTIL T2 = 0;
                                END OF PROCESSING LABEL EQUATION INFO;
                                SEGO[6] := NABS(*P(DUP)); % NEW FORMAT SEG. ZERO
                                H[4],[10] := 1; % MARK AS PROG. FILE
                                DISKWAIT(SFG0,[CF],30,H[10]); % REWRITE SEG. ZFRO
                                END OF COMPILE TO LIBRARY PROCESSING;
                                M[ALPHA]+(*P(DUP))&(@60000)[CTF];
                                IF TYPE<20 THEN
                                    M[ALPHA]+(*P(DUP))&(IF SEPARATE THEN 1 ELSE @20)[27:42:6];
                                    IF NOT SEPARATE THEN M[ALPHA]+(*P(DUP))&1[27:42:6];
                                END;
                                IF NOT SEPARATE THEN M[ALPHA]+P(DUP,LOD)&1[27:42:6];
                                M[ALPHA],[20] := 1;
                                IF FIB[14],[FF]#FIB[14],[CF] THEN
                                    END;
                                IF P OR I THEN
SVOIDT 38639101
    IF J THEN SFTNOTINUSF(U,0) ELSE LABELTABLE(U,[15]+1)
    LABEL PX;
    IF SEPARATE THEN P(WAITIO(@4000,00000,0,U),DEI.)
    ELSE P(WAITIO(@4002000000,0,U),DEI.)

```

```

%146-38385700 D 146
%146-38434000 D 155
%146-38446500 D 146
%146-38447000 D 146
%146-38447500 D 146
38448000 D 146
%146-38448500 D 146
%146-38449000 D 146
38449500 D 146
38450000 D 146
%146-38450500 D 146
%146-38451000 D 146
38451500 D 146
%146-38452000 D 146
38452500 D 146
%146-38453000 D 146
%146-38453500 D 146
%146-38453600 D 146
38453700 D 146
%146-38454000 D 146
%146-38454100 D 146
%146-38454500 D 146
%146-38455000 D 146
%146-38455500 D 146
%146-38456000 D 146
%146-38456500 D 146
%146-38457000 D 146
%146-38457500 D 146
%146-38458000 D 146
38458500 D 146
%146-38459000 D 146
%146-38459500 D 146
%146-38460000 D 146
%146-38460500 D 146
%146-38461000 D 146
%146-38461500 D 146
%146-38462000 D 146
%146-38462500 D 146
%146-38463000 D 146
%146-38463500 D 146
%146-38464000 D 146
%140-38590000 D 140
%140-38590100 D 140
%140-38590200 D 140
%140-38592100 D 140
%140-38594000 D 140
%140-38594100 D 140
%140-38594200 D 140
%140-38594300 D 140
%601-38601100 D 601
%103-38608920 D 103
%601-38621500 D 601
%717-38639000 D 717
%717-38640000 D 717
%905-38660000 D 905
%140-38741000 D 140
%140-38741100 D 140

```

```

IF NOT SEPARATE THEN P(WAITIO(@4000100000,0,U),DEL);
BEGIN RDCTABLE[U],[8:6]+0;
FORK(P(.PURGEIT),U,-2,128,1) END ELSE SFTNOTINUSE(U,0);
KIND=FIB[4],[8:4]; IF FIB[13],[28:10]#0 THEN RFEL=FIB[13],[28:10];
IF (T1 + PSFUDOMIX[PIMIX]) # 0 THEN
IF PACKETPAGE[T1 = 32] # 0 THEN
IF COBOL>0 OR FIB[4],[7:1] THEN
MFFIB INX NOT 1],AREATYPEF I= COBOLFIRAREAV
ELSE
MEALPHA=7],AREATYPEF I= ALGOLFIBAREAV;
REAL NEXTLINK, AD, X, K, SEVEN7, FORTY, EUSU;
IF (RA=RL) GFO RADD THEN RLEN I= 0 ELSE
OKSEGFZROWRITE:=TRUE;
OKSEGFZROWRITE:=FALSE;
IF PBCOUNT > 0 THEN % TELL OPERATOR
END; SPOUT(X);
$ VOINT
FIB[16],[CF1]=0];
KEY I= GETSPACE(5,MAINTBUFFAREAV+64,0)+2;
KEY I= GETSPACE(5,MAINTBUFFAREAV+64,0)+2;
T1 I= TYPEDSPACE(15,MAINTRUFFAREAV);
MOVE(10,B,T1+2);
MLA I= [MFA]=GETSPACE(33,MAINTBUFFAREAV,0)+3]1 & 32[SIZEF]]];
" AUTO", "RNOO", %17%
" 8INVALID", " 8 IPC C",
" 20M....",
" 8ARRAY ", " 8OLAY S", % 99
" 8PACE U", " 6SFD UP", %
" DATA029", 4,
" FORM ", 75, % SWITCH D(PCC)
" RANDOM ", 78,
" LINES66", 77,
" DUMMY ", 76,
" ON DISK", " ", % 68
OKSEGZEROWRITE:=TRUE;
OKSEGFZROWRITE:=FALSE;
OKSEGFZROWRITE:=TRUE;
OKSEGFZROWRITE:=FALSE;
OKSEGZEROWRITE:=TRUE;
OKSEGZEROWRITE:=FALSE;
I I= (TI=GETSPACE(J+2,INQUIRYBUFFAREAV,0)+1)+3 AND NOT
IF FENCE,[32:16] # FENCE OR FENCE < 8192 THEN FENCE I= 16384;
IF J < 7 THEN % NEED TO FIND HIGHEST ON LINE MOD BELOW FENCE
MEMTOGI=1;
+(CHUNKMAX=0) % IN CASE ITS ALL BATCH
M[C]=J]=TYPEDSPACE(I,MCPTABLEFARFAV)+T]I=0;
I I= CHUNKMAX+(CHUNKMAX=0);
I I= SPACESTACKSIZE; T I= P(.SPACESTACK); FIX; %
I I= W; T I= P(.ESPTAB); FIX; %
I I= 100; T I= P(.WORKERSTACK); FIX; %
I I= 0;
FOR T I= 1 STEP 1 UNTIL NUMSTACK DO
BEGIN
M[J] := I; % LINK TO PREVIOUS STACK IN Q.
M[J+1] := WORDOFFASE; % CLEAR STACK
MOVE(STANDARDSTACK=2,J+1,J+2);
I I= J; % REMEMBER WHERE THIS STACK IS

```

```

%140=38742100 D 140
%538=38788500 D 538
%538=38789000 D 538
%39092000 D 115
%105=39092165 D 105
%105=39092167 D 105
%155=39304000 D 155
%155=39304500 D 155
%155=39304600 D 155
%155=39305000 D 155
%791=40004500 D 791
%604=40060060 D 604
%203=40249400 D 203
%203=40262100 D 203
%791=40320100 D 791
%791=40320600 D 791
%791=40338000 D 791
%110=41204000 D 110
%155=41312700 D 155
%155=41314000 D 155
%155=41323130 D 155
%135=41323150 D 135
%155=41328600 D 155
%902=41434800 D 902
%145=41445500 D 145
%145=41445600 D 145
%517=41445700 D 517
%517=41445800 D 517
%890=41483010 D 890
%846=41485600 D 846
%603=41485610 D 603
%724=41485620 D 724
%846=41485630 D 846
%166=41493085 D 166
%203=42600919 D 203
%203=42600921 D 203
%203=42713949 D 203
%203=42713951 D 203
%203=42730999 D 203
%203=42731001 D 203
%3] 44041000 D 155
%151=44104000 D 151
%151=44109900 D 151
%202=44114100 D 202
%151=44151050 D 151
%155=44153500 D 155
%151=44163100 D 151
%152=44173100 D 152
%158=44173200 D 158
%158=44179100 D 158
%158=44179200 D 158
%158=44179250 D 158
%158=44179300 D 158
%158=44179350 D 158
%158=44179400 D 158
%158=44179450 D 158
%158=44179500 D 158

```



```

      J := J + STANDARDSTACK; % COMPUTE ADDRESS OF NEXT STAC 44179550 D 158
      FND; %158-44179600 D 158
      STACKQ := 1; % POINT TO FIRST STACK IN Q. %158-44179650 D 158
      MOVE(99,1)=WORKERSTACK,[CF],1+1); %TO SPOT POSSIBLE OVER 44189600 D 158
      IF (T:=MFA+19],[32:16]) # FENCE AND T # 0 THEN %151-44191020 D 151
        # 20000 AND T # 100000 THEN FENCE := T; %151-44191040 D 151
      J:=GETSPACE(Z,AVTABLEAREAV,1)+2); %155-44240840 D 155
      SCRATCHVEC:=IM[GETSPACE(10,SCRATCHDIRAREAV,SCRATCHSAVE)+2]] %155-44241180 D 155
        & 10 [TOSIZE]); %155-44241185 D 155
      OKSEGZEROWRITE:=TRUE; %203-44411399 D 203
      OKSEGZFROWRITE:=FALSE; %203-44411401 D 203
      ITSALLRATCH := FENCE # 60000; %151-44420000 D 151
      IF AUTORN THEN BEGIN RUNNUMBER:=4; STARTADFC(0); END; %902-44424100 D 902
      FOR NT1+1 STEP 1 UNTIL LMAX DO %993-44430000 D 993
        IF SCHEDLINE[NT1] THEN SCHEDBUSY[NT1] # 0; %993-44430100 D 993
        129: GO TO EXTERNAL; % 35 = SPECIAL INTERRUPT 46011500 D 172
        130: GO TO EXTERNAL; % 36 = DKA READ CHECK 46012000 D 172
        131: GO TO EXTERNAL; % 37 = DKB READ CHECK 46012500 D 172
        IF NT3#2 THEN STOPM ELSE SWAP( %991-48030000 D 991
          IF LOGLINE,[33:7]#0 THEN %991-48030010 D 991
            IF NOT SCHEDLINE[LOGLINE,[40:8]] THEN %991-48030015 D 991
              FORCESWAP FLSF TIMEND ELSE TIMEND,1); 48030020 D 991
          %154-48038120 D 154
          IF WE RFACH THIS POINT THEN THIS JOB HAS USED %154-48038130 D 154
          UP ITS TIME SLICE (EITHER PROCESSOR OR I/O) OR %154-48038140 D 154
          %154-48038149 D 154
          THIS JOB HAS BEEN DOING EXCESSIVE OVERLAY AND %154-48038150 D 154
          DESERVES ANOTHER CHUNK OF MEMORY. IF THE %154-48038160 D 154
          PROGRAM HAS USED UP ITS TIME SLICE, WE WILL %154-48038170 D 154
          SWAP IT OUT IF THERE IS SOME OTHER PROGRAM THAT %154-48038180 D 154
          IS READY TO RUN AND CAN USE THIS GUYS CHUNKS, IF %154-48038190 D 154
          THE PROGRAM HAS BEEN DOING EXCESSIVE OVERLAY AND %154-48038200 D 154
          THE PROGRAM HAS ROOM FOR EXPANSION AND DIDNT USE %154-48038210 D 154
          A MAX CORE CARD, WE WILL SWAP IT OUT %154-48038220 D 154
          AND SWAPPINGIO WILL ARRANGE TO OBTAIN ANOTHER %154-48038230 D 154
          CHUNK. %154-48038240 D 154
          %154-48038250 D 154
          IF P(0,RDS) > FENCE THEN % SWAP JOB %154-48038260 D 154
          BEGIN %154-48038270 D 154
            FOR NT2 := SC[P1MIX] STEP 1 UNTIL LC[P1MIX] DO 48038280 D 154
              IF ACTIVE[NT2] > 1 THEN % SOMEONE CAN USE THIS 48038290 D 154
                BEGIN % CHUNK, %154-48038300 D 154
                  SWAP(TIMEND,1); % %154-48038310 D 154
                  GO TO RETURN; %154-48038320 D 154
                END; %154-48038330 D 154
              IF OLAYCTR[P1MIX] < 0 THEN % EXCESSIVE OVERLAY, 48038340 D 154
                IF DONTEXPANDBITS[P1MIX] = 0 THEN % EXPANDABLE 48038350 D 154
                  BEGIN %154-48038360 D 154
                    SWAP(TIMEND,1); % %154-48038370 D 154
                    GO TO RETURN; %154-48038380 D 154
                  END; %154-48038390 D 154
            BEGIN SWAP(IF LOGLINE,[33:7]#0 THEN %991-48038400 D 991
              IF NOT SCHEDLINE[LOGLINE,[40:8]] THEN FORCESWAP %991-48038405 D 991
                ELSE TIMEND ELSE TIMEND,1); GO TO RETURN END; %991-48038410 D 991
              BEGIN %155-48052000 D 155
                P(NT4,BLOB,NT1,GETSPACE(NT1,STACKAREAV,3));48052100 D 155
                $ SET OMIT = NOT(NEWLOGGING) %550-48100499 D 550

```

STARTLOG(P1MIX)  
\$ POP OMIT  
IF U LEQ 15 THEN \* TAPE I/O  
FND.  
? FND.

%550=48100500 D 550  
%550=48100501 D 550  
%168=94129560 D 168  
%168=99999999 D 168  
000000C1

NUMBER OF ERRORS DETECTED = 0.  
PROCESSOR TIME = 185 SECONDS.  
I/O TIME = 93 SECONDS.

LABEL 00000000LINE 00179219? EXECUTE PATCH/MERGE

PATCH /MERGE