

CMNDS IDENT H20 8/5/68

* COMMAND, SUBSYSTEM, AND USER INITIALIZATION TABLES

NCMEM EQU 57B
NTTB EQU 32
SEC90 DATA 90000 1 1/2 MIN
SEC130 DATA 180000 3 MIN

MSET MACRO D
LDA =D(1)
SBRM XTMSG
ENDM

MSG MACRO D
LDA =D(1)
SBRM XOMSG
ENDM

*INITIAL ENTRY POINT FROM MONITOR

SEXECI SBRM EXECRL
CLAB
STP PRGRL1 NO PROGRAM SPACE FOR USER
STP INFIL
STA UNO USER NOT LOGGED IN
STA ULFLG
STA ULINK
STA* TJOB CLEAR CHARGED TIME
STA LTJOB
STA CIN SET COMMAND I/O
STA GOFN1
LDA =1 TO TELETYPE
STA COUT
LDA REAL
STA UREAL RESET ELAPSED TIME
STA LREAL
LDX =-6
STX OPFE,2
BRX *-1
LDA =-1
STA SWTO
STA SYSTL NO SUBSYSTEM IN USE
STA MAGT NO MAGTAPE FILE IN USE
LDX UTTY
STA AUNN,2
STA WERIS,2
LDX =-1; BRS 29; MSET M42; BRS 47; LDA REAL; SBRM DAYTIM
BRU ABORT

```

$EXECP MSET MRST; BRU ABORT
MRST ASC '$SYS RESTART, PROCEED WITH CAUTIONS/'
* RE-ENTRY AFTER MEM AND ILLEG INST TRAPS AND ESCAPE

```

```

$EXECP CLA ; STA CIN; LDA =1; STA COUT; CXA
SKG =0 CHECK TYPE OF TRAP
BRU ABRT ESCAPE
SKG =1; BRU EXECP1; MSET M66; BRU EXECP2

```

```

EXECP1 MSET M65
EXECP2 LDA EXECL
SBRM PONO
BRU **2
ABRT MIN NESC
LDA UNO
SKG X0
BRU CLOSE2
SKN SWD
BRU ABORT
SBRM UPDATR
BRU ABORT

```

*GENERAL ERROR RE-ENTRY POINT

```

$ERRET MSET E1
ERRTM1 EXT ERRET-1
LDX --1
BRS 11
$ABORT SBRM BRS17 TERM MAG TAPE, CLOSE ALL FILES, ENABLE ESC
CLAB
STP INFIL
BRU CLOSE2

```

* CLOSING OUT SEQUENCE FOR COMMANDS

```

OUTEX LDA TERMWD WRITE 2 TERMWDS AT END OF FILE
WIO OUTFIL
WIO OUTFIL
CLOSE CLA 0 CLOSE MAGTAPE I/P FILE - INFIL
XMA INFIL
SBRM BRS20
CLA 0 CLOSE MAGTAPE O/P FILE - OUTFIL
XMA OUTFIL
SBRM BRS20

```

```

CLOSE2 LDX      =-1          SET ECHO TABLE 0
      CLA
      BRS      12
      BRS 78
      BRS      47          TURN ESCAPE ON
      LDA      UNO
      SKE      =0
      BRU      CRA          GET COMMAND
      SBRM     EXECRL
      LDX =TIMOUT; LDA SEC90; ADD UREAL; SUB REAL; BRS 108; SBRM TFORK
      BRU      ENTER          ATTEMPT LOG IN
$CRA  BRS 46; LDA LETFLG; SKG ULFLG; BRU **4; STA ULFLG; MSET MBUF
      BRS      47
      MSET     M23          TYPE '@'
      SBRM     EXECRL          UPDATE EXRL1, EXRL2
      SBRM     CKTIM
      SKA      LOGONT
      BRU      CRA1
      SKN      SWTD
      BRU      CRA1
      MSET     M43          'TIME RUNOUT, 3 MIN'
      MIN      SWTD
      LDX      =TIMOUT
      LDA      SEC180
      SBRM     TFORK          3 MINUTE TIMER

CRA1  LDX      UTTY
      LDA      LCW,2
      LRSH     16
      ETR      =177B
      SKG      X0
      BRU      CRA3
      BRS      46
      SUB      =1
      STA      ULINK
      CAX
      SBRM     OUTUSE
      MSET M5; LDA ULINK; SBRM PDNO
      MSET     M6
      BRS      28
      BRU      CRA

CRA3  LDA      CIN
      STA      GOFN1
      LDA      REAL
      SUB      LREAL
      SKA      =200000B
      SBRM     MFD
      SBRM GET3C; STA RLV2; LDX =-NCOMS
      SKE* ECMTBL,2; BRX *-1 FOUND THE COMMAND
      CXA; SKE X0; BRU **2; BRU CRA3A NOT THERE--A FILE? S.E.T.
      LDA ECMTBL,2; ADD =1; STX RLV2; SBRM XTMSG; LDX RLV2
      SKN EBRTBL,2; BRU* EBRTBL,2; SBRM OKTOGO; BRU* EBRTBL,2 DISPATCH

```

CRA3A LDA RLV2; MRG =200B; COPY B,X
SBRM BRS48; BRU ERRET; SBRM BRS16; BRU ERRET; STA INFIL
LDA =-1; STA GETPL; BRU GET10A

* GET 3 CHARS FROM TTY TO A, IN CORRECT ORDER

GET3C ZRO RLV2
CIO GOFN1; SKE QUOTCH; BRU *+2; BRU QT; CAX
CIO GOFN1; LRSB 8; CXA; LSH 8; CAX
CIO GOFN1; LRSB 8; CXA; LSH 8; BRR RLV2

\$OUTUSE ZRO RLV3
LDB WERIS,2
SBRM OUTU1
LDB WERISB,2
SBRM OUTU1
BRR RLV3

\$OUTU1 ZRO RLV1
STX RLV4
LDX --5
CLA
LSH 1
DIV T27+5,2
SKG =0
BRU *+3
ADD =40B
CIO COUT
BRX OUTU1+3
LDX RLV4
BRR RLV1

T27 DATA 531441,19683,729,27,1

* 'LOGIN SEQUENCE'

ENTER LDA REAL; SUB UREAL; SKG SEC90; BRU *+2; SBRM TIMEOUT; CLX
STX UTNO STANDARD TAPE UNIT =0
MSET M56
SBRM FNAC FINDS AND READS IN ACCT DIR
LDX =-1; BRS 11 TO PREVENT TYPE-AHEAD
MSET M71
SBRM GTS GET STR TERM BY SEMI COL
BRU ENTER6 NO SEMI
SBRM NMC CK FOR VALID PASSWORD
BRU ENTER6 NOT VALID
MSET M72
SBRM NMVF
MSET M21
SBRM NMV CHECK USER NAME
BRU ENTER6 NOT VALID
STX TLV1 USER AUD ENTRY ADDR
SBRM CKTIM PUT CURRENT HOUR BIT IN A
LDX =DTEMP
SKA 2,2 CK FOR VALID HR THIS ACCT
BRU *+2
BRU ENTER4 NO
LDA 2,2
STA LOGONT

LDX	TLV1	LOC OF NAME IN AUD
LDA	2,2	
ETR	=7777B	USER NO.
STA	UN01	
LDX	=-NTTB	CK IF USER IS ALREADY ON
LDB	=7777B	
SKM	AUNNE,2	CK IN AUNN
BRU	*+2	
BRU	ENTER6	ALREADY ON
BRX	*-3	
BRS	46	
BRS	108	USER VALID KILL TIMING FORK
RSH	12	
LDA	SAVELL	
RCY	12	
LDX	UTTY	
LDA	XWR3	
STA	WERIS,2	PUT USER NO. IN WERIS TABLE
LDA	XWR9	
STA	WERISB,2	
STB	AUNN,2	
LDX	TLV1	
LDA	2,2	
STA	PRWRD	
ETR	=37777777B	
STA	CPARW	
SBRM	RFD	READ FD TO TSBLOCK
EAX	UFDE	
LDA	-4,2	
SUB	-5,2	
LDX	JOB	
STA	DBA,2	
LDA	UN01	RECORD USER AS BEING LOGGED ON
STA	UN0	
MIN	NLOG	TOTAL LOG ONS
LDX	--1	
BRS	11	
BRU	CRA	
ENTER4 MSET	M34	
LDX	UTTY	
BRS	14	
BRU	LGOUT3	
ENTER6 MSET	M57	
BRU	ERRET+2	

* 'LOGOUT'

\$LOGOUT LDX UTTY
SBRM OUTUSE
SBRM OK TO GO
MSET M21
LDA REAL
SBRM DAYTIM
BRS 46
MSET M18
LDA UREAL
SBRM GETTIM
SBRM HOUR

LDX =UFDE; CLA; STA -8,2; XMA -7,2
SKG =0; BRU LGOUT4; STA GET2; LDX =9; BRS 1
BRU LGOUT4; BRS 66; BRS 2; LDA GET2; BRS 67

LGOUT4 LDX UTTY

BRS 14

\$LGOUT1 SBRM KILL6 RELEASE 'PROGRAM' MEMORY
SBRM BRS17 CLOSE ALL FILES
SBRM MFD

\$LGOUT3 BRS 108 KILL ALL FORKS

BRS 76

CLAB

LDX UTTY

STA AUNN,2 CLEAR AUD

STA WERIS,2

STA WERISB,2

LDX JOB

BRS 112 LET MONITOR CLEAN UP

* CALL FOR ANY SUBSYSTEM

```
$CSYS LDA EBRTBL,2
      ETR =37700000B
      SKA =12100000B
      SKA CPARW
      BRU *+2
      BRU ERRET
      COPY XA,B
      ADD =NCOMS
      LSH 2
      ADD =SUBTB1
      STA TLV1
      COPY AX
      LDB -1,2
      EAX PRGRL1
      SKB X2
      SBRM KILL6 NO, RELEASE USER'S MEMORY FIRST, THEN...
      SBRM RSYS GET SPECIFIED SUBSYSTEM
      LDA TLV1
      CLB
      SBRM GSYS
      CLA 0 BUILD PANIC TABLE FOR SUBSYS FORK
CSYS1 STA SSL
      LDX SYSTL EXTRICATE FORK STARTING ADDR FROM SUBSYS
      STX SYSRRL
      LDA -1,2
      ETR ADM SK
      ADM SSL START LOC

      LDA CIN
      LDB COUT
      STP SSA SET FORK INIT. AB
      LDX SYSRRL CONSTRUCT FORK MAP FROM STRUCTURE WORD I
      LDA 2,2
      SBRM DRSW
      STP SSR1
      LDX SYSRRL; LDA -1,2; ETR X4
      MRG SSL3
      BRS 9
      BRS 31

      SBRM EXECRL PASS ON TO EXEC, PROGR ANY CHANGES IN PA
      LDB =SSR1 ... SUBSYSTEM AND RELEASE MEMORY GRABBED
      LDX SYSRRL
      LDA 2,2
      SBRM RRR
      SBRM SEXR
      LDX SSM EXIT ACCORDING TO STATUS OF SUBSYS ON TE
      LDA SSL
      STA EXECL
      BRU *+1,2
      BRU ABORT
      BRU EXECP
      BRU EXECP
$SSL3 NOP SSL,3
```

* 'CONTINUE'
\$CONT SBRM PSCN
SBRM OKTOGO
LDA =1
BRU CSYS1

* 'KILL PROGRAM'
\$KILL EAX PRGRL1
SBRM RELST
BRU CRA

* 'CLEAR'
\$RESET SBRM KILL6
BRU CRA

* 'RENAME'
\$REN SBRM FDFNA
BRU ERRET
ADD =200000006B
STA XWR4
MSET M69

SBRM FDFNA
BRU *+2
BRU ERRET
SBRM OKTOGO
BRS 46
LDX =-3
LDA GBUFE,2
STA* XWR4
BRX *-2
BRU CRA

* 'GET'
\$STEAL SBRM RFINA; BRU ERRET; BRU ERRET; BRU ERRET
MSET M75; SBRM FNAC; CLA; CIO COUT; SBRM NMVF
LDX =-49
STEAL2 LDA DTEMPE,2
SKE XWR3
BRU STEAL1
EAX -1,2
LDA DTEMPE,2
ETR =7777B
SKG X0
BRU ERRET

CLB
LSH 2
ADD =400B
CAB
LDA =DTEMP
LDX =256
BRS 61
HLT
CAX ; LDB 247D,2
SBRM HCSDF
BRU ERRET
LDX UNPTR
SKN 2,2
BRU ERRET
SKN 0,2
BRU *+2
BRU ERRET
LDA 2,2
SKA X1
BRU *+6
LDX UTTY
LDA AUNN,2
LRSH 12
SKE SAVELL
BRU ERRET
LDX UNPTR
LDA 2,2
LRSH 17
ETR =7
STA COPYT
LDA 1,2
ETR XX
SKG X0
BRU ERRET
STA INFIL

STEAL 4 MSET M74

LDX --1
CLAB
SBRM BRS53
BRU STEAL 4
STA OUTFIL
CXA
SKE OKCH
BRU STEAL 4
LDA =100006B
STA UNPTR
LDA INFIL
LDX =8
BRS 1
BRU ERRET
STA INFIL
LDX COPYT
BRU COPY13

STEAL 1

CXA
ADD =3
CAX
BRX STEAL 2
BRU ERRET

* 'COPY'

```
$COPY CLX
      CLAB
      SBRM BRS58
      SBRM NEWTRY
      SKA X4
      BRU ERRET
      STA INFIL
      MSET M50
      LDX ==-1
      CLAB
      SBRM BRS58
      SBRM NEWTRY
      STA OUTFIL
      CXA
      SKE X0
      BRU COP2
      MSET M50
      LDX ==-1
      CLAB
      SBRM BRS58
      SBRM NEWTRY
COP1 STA UNPTR
      MSET M21
      LDA INFIL
      SBRM BRS16
      BRU ERRET
      STA INFIL
      STB COPYT
      STX UNPTR1
      XXB
COPY13 LDA OUTFIL
      SBRM BRS19
      BRU ERRET
      STA OUTFIL
      LDX COPYT
      LDP UNPTR
      SBRM BRS19
      BRU ERRET
      STA UNPTR
      LDX ==-1; LDA =2; BRS 12
      LDA COPYT
      SKE =3
      BRU COPY5
COPY3 CIO INFIL
      SKN INFIL
      BRU *+6
      CAB
      LDA INFIL
```

	SKA	=00400000B	
	BRU	COPY11	READ ERROR
	CBA		
	SKE	EO TCH	
	BRU	*+2	
	LDA	TERMCH	
	CIO	OUTFIL	
	SKN	OUTFIL	
	BRU	*+6	
	CAB		
	LDA	OUTFIL	
	SKA	=00400000B	
	BRU	COPY12	
	CBA		
	CIO	UNPTR	
	SKN	UNPTR	
	BRU	*+6	
	CAB		
	LDA	UNPTR	
	SKA	=00400000B	
	BRU	COPY12	
	CBA		
	SKE	TERMCH	
	BRU	COPY3	
	CIO	OUTFIL	
	CIO	OUTFIL	
	CIO	UNPTR	
	CIO	UNPTR	
	BRU	CLOSE	
COPY5	CLA		
COPY6	STA	SAVELL	
	WIO	INFIL	
	WIO	OUTFIL	
	WIO	UNPTR	
	SKE	TERMWD	
	BRU	COPY6	
	XMA	SAVELL	
	SKE	TERMWD	
	BRU	COPY6+1	
	MIN	NCOP	
	BRU	CLOSE	
COPY11	MSET	M9	'READ ERROR'
	BRU	COPY12+2	
COPY12	MSET	M10	'WRITE ERROR'
	LDA	OUTFIL	
	BRS	66	
	LDA	UNPTR	
	BRS	66	
	BRU	CLOSE	
COP2	LDA	=100006B	
	BRU	COP1	

* 'DELETE FILE'

\$DEL SBRM FDFNA
BRU ERRET
SBRM OKTOGO
SBRM DELFIL; BRU ERRET; BRU CRA
\$DELFIL ZRO RETURN; LDX UNPTR
SKN 0,2; BRR RETURN IF NOT PRIVATE ABORT
BRS 46
LDA 1,2
ETR =777774B
SKE X0
BRU *+2
BRU DELL1
LDX =9
BRS 1
BRR RETURN
BRS 66
BRS 2
LDX UNPTR
LDA 1,2
ETR =777774B
BRS 67
DELL1 LDA UNPTR
SBRM FDEL
MIN NDEL
SBRM MFD; MIN RETURN; BRR RETURN

* 'UPDATE FD'

\$UPFD SBRM MFD
BRU CRA

* 'LIBRARY'

\$PLD SBRM RLD; LDX =-128
PLD1 LDA LIBE,2; SKE =0; BRU *+2; BRU CRA; STX RLV5; LDX --3
CIO COUT; LRSH 8; BRX *-2; MIN RLV5; LDX RLV5; CLA
LDB LIBE,2; LSH 3; STA RLV4; LSH 3; CLA; RSH 6; STB RLV3
LDX RLV4; LDA T1,2; SBRM XTMSG; CLA; CIO COUT; LDA RLV3; SBRM P00
MSET M21; LDX RLV5; BRX PLD1; BRU CRA
RLD ZRO RLV1; LDA =LIB; LDB =256; LDX =128; BRS 61; HLT; BRR RLV1

* 'FD FOR'

\$LFDF SBRM FDFNA
BRU ERRET
LDA =0
CIO COUT
SBRM FPRINT
BRU CRA

* 'FILES'
\$SLFP LDB =-1
BRU LFP

* 'DI RECTORY'

\$LFDCOM CLB

LFP STB DEL1
LDA FDEND
SKG =0
BRU CRA
SKB X4
BRU *+3
MSET M15
LDA =UFD
STA UNPTR
ADD FDEND
STA UNPTR1
LFP1 LDA UNPTR1
SKG UNPTR
BRU CRA
SBRM PFINA
SKN DEL1
BRU LFP2
MSET M21
BRU *+2
LFP2 SBRM FPRINT
LDA =7
ADM UNPTR
BRU LFP1

SDRDERR MSET M9
BRU ERRET+2

SDWTERR MSET M10
BRU ERRET+2

```

*      'EXECUTE'
$GO   LDA      =-1
GET10 STA      GETPL
      SBRM     GIFN          READ INPUT FILE NAME
      SBRM     OKTOGO
GET10A BAC
      SKE      =1           CHECK FILE TYPE IS CORE-IMAGE
      BRU      GET9+1
      SKN GETPL; BRU *+3; LDA =PRGRL1; SBRM CKRLB
      SBRM     RSYS
      LDA      INFIL
      LDX      =PRGRL1
      SBRM     GET          READ FILE INTO PRGRM MEMORY
      BRU      DRDERR
      CLA
      STA      INFIL
      SKN     GETPL
      BRU     CRA          EXIT HERE IF 'PLACE'
GET8  SBRM     EXECRL
      LDA      GETSTL      SET PANIC TABLE FOR FORK
      MRG     =40100000B
      STA      SSL
      LDP     PRGRL1
      STP     SSR1
      LDA     GET9
      SKN     CPARW
      BRU     *+2
      MRG     X4          START AS EXECUTIVE
      BRS     9          START UP RECOVERED FILE
      BRS     31
      LDP     SSR1          UPDATE PRGRL AND RETURN FOR NEXT COMMAND
      STP     PRGRL1
      LDA     SSL
      STA     EXECL
      BRU     EXECPC
GET9  NOP     SSL,3
      MSET    XM6
      BRU     ERRET+2

```

* 'DEFINE'
* SETS UFD STATUSES

\$STFD SBRM FDFNA
BRU ERRET
MSET M69
SBRM DCHAR3
BRU ERRET
LDX UNPTR
SKE PRI
BRU ST1
LDA 1,2
ETR =37777777B
STA 1,2
CBA ; SKE COMCH; BRU ST5
SBRM DCHAR3
BRU ERRET
SKE REA
BRU ERRET
BRS 46
LDX UNPTR
LDA X4
ST4 MRG 1,2
STA 1,2
LDA X4
MRG 0,2
STA 0,2
BRU UPFD
ST1 XAB
SKE COMCH; BRU **2
BRU ERRET
COPY BA,B
SKE ACC
BRU ST3
ST2 BRS 46
XAB
XMA 2,2
ETR =67777777B
ADM 2,2
LDA 0,2
ETR =37777777B
STA 0,2
LDA X4
MRG 1,2
STA 1,2
BRU UPFD
ST3 SKE PUB
BRU ERRET
LDB X1
BRU ST2
ST5 BRS 46; CLA ; BRU ST4

* 'PLACE'
\$PLACE CLA
BRU GET10

```

*      'BRANCH'
SBRANCH SBRM  RONO
      XAB
      SKE      OKCH
      BRU      ERRET
$GOTO1  STB    GETSTL
      BRU      GET8

```

```

*      'SAVE'
$SAVE  SBRM  RONO      READ CORE BOUNDS
      STA  SAVEFL
      MSET M50
      SBRM RONO
      STA  SAVELL
      CBA
      MSET M16
      LDA  SAVELL      READ O/P FILE NAME AND OPEN FILE OF TYPE
      SUB  SAVEFL
      ADD  =1
      LDB  =1
      SBRM GOFN
      CBA ; SKE COMCH; BRU SAVE2
      MSET M3
      SBRM RONO      ... READ STARTING LOCATION
      BRU  *+2
SAVE2   LDA  =240B
      STA  SAVESL
      CBA ; SKE OKCH ; BRU ERRET
      LDA  SAVEFL      WRITE OUT CORE BOUNDS AND STARTING LOCAT
      STA  GET2
      WIO  OUTFIL
      LDA  SAVELL
      STA  GET3
      WIO  OUTFIL
      LDA  SAVESL
      WIO  OUTFIL
      LDA  =PRGRL1      SET UP DATA FOR GBIO, PERFORM INFORMN. TB
      A    GET5
      LDX  =GET2
      LDA  OUTFIL
      SBRM GBIO
      BRU  SAVE1
      BRU  OUTEX
SAVE1   MSET M10
      BRU  OUTEX

```

```

*      'DUMP'

```

```

SDUMP  SBRM      EXECRL      (UPDATE EXRL)
        CLA
        STA      TLV1
        LDA      =NCMEM
DUMP10 ADD      =1
        SKG      =77B
        BRU      DUMP11
        LDA      TLV1
        CLB
        LSH      11
        LDB      =4
        SBRM     GJFN
        LDA      OUTFIL
        SBRM     BRS95
        BRU      SAVE1
        BRU      OUTEX
DUMP11 BRS 39; BRU DUMP10; MIN TLV1
        BRU      DUMP10

```

```

*      'RECOVER'
SREC  SBRM     GJFN
        CBA
        SKE     =4
        BRU     ERRET
        SBRM    OKTGO
        SBRM    KILL6
        LDA     INFIL
        SBRM    BRS96
        BRU     DRDERR
        BRU     CLOSE

```

```

*      'RESTART'
*      RESTARTS A USER AFTER ACCIDENTAL DISCONNECT

```

```

$RESDMP LDA     =4000000B
        SKA     CPARW      LOOK FOR RESTART STATUS
        BRU     *+2
        BRU     ERRET
        EAX     UFDE
        LDA     -7.2
        SKG     =0
        BRU     RDMPM
        STA     GET2
        BRS     46
        SBRM    KILL6
        CLA
        XMA*    DBAJOB
        STA     TLV1
        LDA     GET2

```

```

LDX      =8
BRS      1
BRU      RESDP1
SBRM     BRS96          RECOVER RESTART FILE
BRU      RESDP1
CLA
XMA      INFIL
BRS      2
LDA      GET2
LDX      =9
BRS      1
BRU      RESDP1
BRS      66
BRS      2
LDA      GET2
BRS      67
CLAB
EAX      UFDE
STP      -8,2
LDA      TLV1
ADM*     DBAJOB
BRU      CLOSE
RDMPM    MSET           M63          'RESTART MISSING'
BRU      ERRET+2
RESDP1   LDA           TLV1
ADM*     DBAJOB
BRU      DRDERR

```

```

*      'TIMES'
*      PRINTS ELAPSED TIMES SINCE BEG OF ACCT PER.

```

```

$TIMES  SBRM           MFD
LDA     UREAL
SBRM    GETTIM
SBRM    HOUR
MSET    M4
LDA     UREAL
SBRM    DAYTIM
MSET    M40
SBRM    CHAR3
BRU     CRA
SKE     YES
BRU     CRA
MSET    M49
LDX     =UFDE
LDP     -3,2
COPY    AB,BA
SBRM    HOUR
MSET    M48
LDX     =UFDE
LDA     -6,2
MUL     =255
LSH     23
SBRM    PDNO

```

MSET M21
BRU CRA

\$SETDAT SBRM EXEUSE; SBRM SDAT; BRU CRA

SFIND SBRM NMVF
LDA XWR8
LDX =NTTB

FIND1 XXA
SKE =0
BRU FIND4
MSET M30
BRU CRA

FIND4 XXA
EAX -1,2
SKE WERIS,2
BRU FIND1
STX DEL2
STA DEL3
MSET M5

FIND2 LDA DEL2
SBRM PDNO

FIND3 LDA DEL2
SKE =0
BRU *+2
BRU CRA
SUB =1
STA DEL2
CAX
LDA WERIS,2
SKE DEL3
BRU FIND3
LDA COMCH
CIO COUT
BRU FIND2

* 'STATUS'

\$MA LDA PRGRL1; MRG PRGRL2; SKE =0; BRU *+2
BRU MA2; MSET M21; EAX PRGRL1; SBRM PRMA

MA2 SKN SYSTL; BRU *+2; BRU CRA; MSET M21
SBRM PSCN; CLA ; CIO COUT; EAX SSRL; SBRM PRMA; BRU CRA

* 'DISC ALLOCATION'

\$MA1 SBRM MFD
MSET M27
EAX UFDE
LDP -6,2
STP UNPTR
LDA -4,2
MUL =255
LSH 23
SBRM PDNO

MSET M36
LDA UNPTR
MUL =255
LSH 23
SBRM PDNO
MSET M54
LDA UNPTR1
MUL =255
LSH 23
SBRM PDNO
BRU CRA

PRMA ZRO PRMA1; STX DEL2; LDX =-8
PRMA5 STX DEL3; LDA DEL2; SBRM ERB; SKG =0
BRU PRMA4; SBRM PDNO
PRMA3 LDB SLSHCH; LDA DEL3; SKE =-5; CLB ; CBA ; CIO COUT
LDX DEL3; BRX PRMA5; BRR* PRMA
PRMA4 LDA MINCH; CIO COUT; BRU PRMA3

* DATE

\$DATE \ LDA REAL
SBRM DAYTIM
BRU CRA

\$WHO LDA =-1 ; STA RLV2
WHO1 MIN RLV2; LDA RLV2 ; SKG =NTTB-1
BRU *+2 ; BRU CRA
CAX; LDA AUNN,2; SKE =0; BRU *+2; BRU WHO1
SBRM OUTUSE USER NAME
MSET M5; LDA RLV2 ; SBRM PDNO LINE NO.
MSET M21; BRU WHO1

* 'ACCEPT LINKS'

\$AM LDX =-1
LDA =2
BRS 25
BRU CRA

* 'REFUSE LINKS'

\$RM LDX =-1
CLA
BRS 25
BRU CRA

* 'LINK'

\$ML SBRM RDNO; XAB; SKE OKCH; BRU ABORT
 ML1 BRS 46; CBA; BRS 23; BRU MLR
 ** MIN NLNK REMOVED 2 14 69. DD NEEDED COUNTER IN MONITOR
 MSET M21; LDX UTTY; SBRM OUTUSE; MSET M5
 LDA UTTY; SBRM PDNO; MSET M70; BRU CRA
 MLR SKA =77777700B
 BRU MLR1
 MSET M11 'BUSY'
 BRU CRA
 MLR1 SKA =60000000B
 BRU MLR2
 MSET M14
 BRU AM
 MLR2 MSET M13
 BRU CRA

* LINKS USER TO OPERATOR

\$CSHARE LDA OPR
 LDX REAL
 STX LGTM (TS-BLOCK)
 BRU ML1

* ...
 \$QT CIO CIN
 SKE EDTCH
 BRU *+2
 BRU CRA
 SKE CRCH
 BRU QT
 LDA LINF
 CIO COUT
 BRU QT

SETEXE SBRM RDNO; XAB; SKE OKCH; BRU ERRET; LDA PRWRD; SKB X4
 BRU STXE1; ETR =37777777B
 STXE1 STA CPARW; BRU CRA

* 'BREAK'
 \$BL BRS 24
 BRU CRA
 PRI ASC 'IRP'
 ACC ASC 'CCA'
 PUB ASC 'BUP'
 REA ASC 'DDR'
 YES ASC 'SEY'

* ASSORTED ROUTINES FOR SYSTEM DEBUGGING COMMANDS

* GET CORE BLOCK ADDRESS

SYSGBN ZRO DEL2; SBRM RDNO; SKG =7; BRU **2; BRU ERRET
CLB; LCY 11; BRR DEL2

* GET DRUM BAND ADDRESS

SYSGDA ZRO DEL2; SBRM RDNO; SKA =77777600B; BRU ERRET; LSH 13
MRG X4; STA GET1; CLA; LSH 11; SKE COMCH; BRU SYSGA1
MSET M67; SBRM RDNO; SKA =-4; BRU ERRET
LSH 11; ADM GET1; CLA; LSH 13
SYSGA1 SKE DOTCH; BRU ERRET; BRR DEL2

* GET CORE LIMITS AND DRUM ADDRESS

SYSGCD ZRO DEL1; SBRM SYSGBN; STA GET2; MSET M50 (TO)
SBRM SYSGBN; ADD =3777B; STA GET3; MSET M63 (BAND)
SBRM SYSGDA; BRR DEL1

* READ OR WRITE ON DRUM

SYSRWD ZRO DEL1; LDA =PRGRL; STA GET5; LDA GET1
SYSRWX LDX =GET2; SBRM GBIO; BRU ERRET; BRR DEL1

* GET DRUM ADDRESS AND SET UP TO READ OR WRITE DDT SYMTAB

SYSGBD ZRO DEL1; SBRM SYSGDA; CLA; LDB =17777B; STP GET2
LDA =SSRL; STA GET5; BRR DEL1

* READ OR WRITE FROM DDT RELABELING

SYSRWE ZRO DEL1; LDA GET1; BRU SYSRWX

EXEUSE ZRO RLV1

SKN CPARW; BRU ERRET; BRR RLV1

* SYSDUMP, SYSLOAD

\$SYSDP SBRM EXEUSE; SBRM SYSGCD; LDB =1; SBRM SYSRWD; BRU CRA
\$SYSLD SBRM EXEUSE; LDA =PRGRL1; SBRM CKRLB; SBRM SYSGCD
CLB; SBRM SYSRWD; BRU CRA

* SYSSAVE, SY SDEBUG

\$SYSSV SBRM EXEUSE; SBRM SYSGBD; LDB =1; SBRM SYSRWE; BRU CRA
\$SYSD SBRM EXEUSE; SBRM SYSGBD; LDA =DDTRLB; SBRM GSYS
CLB ; SBRM SYSRWE; LDA =1 ; BRU CSYS1

* SYSGO

\$SYSGO SBRM EXEUSE; SBRM SYSGDA; LDA GET1; ETR =7774000B; STA SYSB; BRU

* CREATE COMMAND TABLES ENTRY

CFLG1 EQU 10000000B
CFLG2 EQU 00100000B

CFLG3 EQU 02000000B
CFLG4 EQU 40000000B
NCOMS EQU 51

COM MACRO D,P,1
ZRO P(1)
C EQU *
BSS NCOMS-1
A EQU D(2)
N NARG
IF N-2
M EQU 3
RPT N-2
A EQU A+CFLG.D(M)
M EQU M+1
ENDR
ENDF
DATA A
BSS STRTBL-*
P(1) ASC 'D(1)/'
STRTBL EQU *
BSS C-STRTBL
ENDM

* TAGS:
* 1-OPERATOR STATUS, 2-OPEXEC, ARPAS, 3-LISP, 4-CONFIRMATION NEEDED
FRGT CFLG1; FRGT CFLG2; FRGT CFLG3; FRGT CFLG4

SCOMTBL EQU *
BRUTBL EQU *+NCOMS
ECMTBL EQU COMTBL+NCOMS
EBRTBL EQU BRUTBL+NCOMS
STRTBL EQU *+2*NCOMS

COM QED, CSYS, 4
COM CAL, CSYS, 4
COM BASIC, CSYS, 4
COM LISP, CSYS, 4, 3
COM SNOBOL, CSYS, 4
COM FTC, CSYS, 4
COM PRINT, CSYS, 4
COM FOS, CSYS, 4
COM DDT, CSYS, 4
COM ARPAS, CSYS, 4, 2
COM OPEXEC, CSYS, 4, 2
COM HBS, CSYS, 4
COM (COPY), COPY
COM (CONTINUE), CONT
COM (EXECUTE), GO
COM (RENAME), REN
COM (WHO IS ON? \$), WHO
COM UPDATE, UPFD, 4
COM (DELETE), DEL
COM (PLACE), PLACE
COM (BRANCH TO), BRANCH
COM (DISC ALLOCATION\$), MA1

```

COM SDATE, SETDAT, 4
COM (LIBRARY$$), PLD
COM (WHERE IS ), FIND
COM (SAVE CORE FROM ), SAVE
COM (FD FOR ), LFD
COM (DEFINE ), STFD
COM (LOGOUT ), LOGOUT
COM (FILESS), SLFP
COM DIRECTORY, LFD
COM (ACCEPT MESSAGES), AM, 4
COM AID, CSHARE, 4
COM (LINK TO TELETYPE ), ML
COM (BREAK LINK), BL, 4
COM (SET EXEC ), SETEXE
COM (REFUSE MESSAGES), RM, 4
COM (TIME USED), TIMES
COM CLEAR, RESET, 4
COM RESTART, RESDMP, 4
COM (KILL PROGRAM), KILL, 4
COM (DUMP ON ), DUMP
COM (RECOVER FROM ), REC
COM MAP, MA
COM (DATE ), DATE
COM (GET ), STEAL
COM (SSAVE ), SYSSV
COM (READ BLOCKS ), SYSLD
COM (WRITE BLOCKS ), SYSDP
COM (DEBUG ), SYSD
COM (SYSGO ), SYSGO
BSS STRTBL-*

```

```

$SUBTBL DATA 30030000B QED
DATA 0
DATA 2015B
DATA 11111111B
DATA 30004000B CAL
DATA 212223B
DATA 24120000B
DATA 11111111B
DATA 30020000B BASIC
DATA 0
DATA 31323311B
DATA 11111111B
DATA 30030000B LI SP
DATA 0
DATA 46475035B
DATA 11111111B
DATA 30024000B SNOBOL
DATA 0
DATA 25262730B
DATA 11111111B
DATA 30004000B FTC
DATA 424300B
DATA 0
DATA 11111111B

```



```
DATA 62000240B PRINT
DATA 100000B
DATA 0
DATA 11111111B
DATA 30004000B FOS
DATA 530000B
DATA 0
DATA 11111111B
DATA 50030000B DDT
DDTRLB DATA 0
DATA 1617B
DATA 11111511B
DATA 60500400B ARPAS
DATA 200000B
DATA 0
DATA 11111111B
DATA 70010000B OPEXEC
DATA 57015152B,00000000B OPEXEC DOES NOT USE PAGE 11
DATA 11111111B
DATA 70004000B,440045B,0,11111111B HBS
$SUBTB1 EQU SUBTBL+1
$NSYS4 EQU *-SUBTB1
END
```

GSUBR IDENT H18 3/7/68

* CONSTANTS

TCNT	EQU	199	
MAXACT	EQU	255	MAX NO OF ACCTS
NTTB	EQU	32	
NCMEM	EQU	57B	

MSET	MACRO	D
	LDA	=D(1)
	SBRM	XTMSG
	ENDM	

MSG	MACRO	D
	LDA	=D(1)
	SBRM	XOMSG
	ENDM	

* 0.3		UPDATE EXRL1, EXRL2
\$EXECRL	ZRO	RLV1
	BRS	43
	STP	EXRL1
	BRR	RLV1

* 0.9		UPDATE EXEC'S RL1, RL2
\$SEXR	ZRO	RLV1
	LDP	EXRL
	BRS	44
	BRR	RLV1

* 0.4			
\$PDNO	ZRO	RLV1	TYPE DECIMAL NO.
	LDB	=10	
PDN2	LDX	COUT	
	BRS	36	
	BRR	RLV1	

* 0.5			
\$PONO	ZRO	RLV1	TYPE OCTAL NO.
	LDB	=8	
	BRU	PDN2	

* 0.7

\$RONO	ZRO	RLV1	READ OCTAL NO.
	LDB	=8	
RON1	LDX	CIN	
	BRS	38	
	BRR	RLV1	

* 0.6

\$RDNO	ZRO	RLV1	READ DECIMAL NO.
	LDB	=10	
	BRU	RON1	

*
 \$OKTOGO ZRO RLV1
 CIO CIN; SKE OKCH; BRU ERRET; BRR RLV1

*ENTRY POINT FOR CARRIER OFF INTERRUPT

\$OFFTY	BRS	46	
	LDA	UNO	
	SKG	=0	IS USER LOGGED IN
	BRU	LGOUT3	
	LDA	=4000000B	
	SKA	CPARW	DOES USER HAVE RESTART BIT
	BRU	*+2	
	BRU	LGOUT1	
	SBRM	UPDATR	
	LDX	=UFDE	
	LDA	-7,2	
	LDX	=9	
	BRS	1	
	BRU	LGOUT1	
	CXB		
	LDX	=UFDE	
	STB	-7,2	SAVE INDEX
	SBRM	BRS95	
	NOP		
	LDA	TERMWD	
	WIO	OUTFIL	
	WIO	OUTFIL	
	LDA	OUTFIL	
	BRS	2	
	BRU	LGOUT1	

\$UPDATR	ZRO	RETURN
	BRS	108
	SBRM	EXECRL
	SKN	SYSTL
	BRU	*+2
	BRR	RETURN
	LDB	=SSR1
	LDX	SYSTL
	LDA	2,2
	SBRM	RRR
	SBRM	SEXR
	BRR	RETURN

```

* 4.3 HAND ALL OF USER'S MEMORY BACK TO MONITOR
SKILL6 ZRO RLV1
LDA =NCMEM
KILL7 ADD =1
BRS 121
SKE =77B
BRU KILL7
CLAB
STP PRGRL
STP SSRL
LDA =-1
STA SYSTL
BRS 43
CLB
STP EXRL
BRS 44
BRR RLV1

```

*READS USERS' FD FROM DISC AND SETS TIME PARAMS

```

$RFD ZRO RLV2
SBRM FFD DISC ADDR OF FD GOES TO B
LDA =UFD FD CORE ADDR
LDX =256
BRS 61 READ DISC
HLT
BRR RLV2

```

**THIS ROUTINE READ USERS FD FROM DISK AND UPDATES CPU, CONNECT
**AND DISC ALLOCATION CHARGES. IT THEN REWITES THE USER FD.

```

$MFD ZRO RLV3; BRS 46; BRS 57; SBRM FFD; STB TLV11 DISC ADR OF FD
LDA =DTEMPE; LDX =256; BRS 61; HLT

```

```

LDX =FDIRE; LDA -3,2; STA TLV21; LDA -2,2; STA TLV2

```

```

LDA LREAL; SBRM GETTIM; STA RLV5; STB TLV1

```

```

CBA; LDX =UFDE; ADM TLTIM; ADD TLV21; STA -3,2

```

```

LDA LTJOB; RSH 23; DIV =60; CNA; ADD RLV5
BRU MFD2 NOW UPDATE USER FD ON DISK

```

```

RSH 23; DIV TLV1; SKG =10; BRU *+2; BRU MFD1; LDA TLV21

```

```

MFD2 ADM TCTIM; ADD TLV2; STA -2,2

```

```

LDA REAL
STA LREAL
LDA* TJOB
STA LTJOB

```

```

LDA      -4,2
SUB*     DBAJOB
SKG      -6,2
BRU      *+2
STA      -6,2
STA      -5,2
LDB =FDIRE; XXB
LDA -15B,2; XXB; STA -15B,2
LDB      TLV11
LDA      =UFD
LDX      =256
BRS      62
HLT
BRS      47
BRR      RLV3
MFD1    LDA =5; MUL TLV1; CBA; BRU MFD2

```

```

*        DELETES FILE DIRECTORY ENTRY
*        (A) = ADDR OF UFD ENTRY

```

```

$FDEL   ZRO      RLV3
        ADD      =22000007B
        STA      COPYT
        LDA      =22000000B
        ADD      =UFD
        ADD      FDEND
        SKE      COPYT
        ADD      =42000000B
        STA      COPY1
        LDA      =-7
        ADM      FDEND
        COPY     AX
        CLA
        EXU      COPY1
        STA*     COPYT
        BRX      *-3
        BRR      RLV3

```

```

* 1.7    GENERAL ROUTINE FOR BLOCK I/O FROM RELABELING OTHER THAN

```

```

$GBIO   ZRO      RLV2
        STX      GBIOPT          CORE BOUNDS
        STP      GBIOFN          B GOES TO GBIO SW, A=RAD ADDR.

```

```

GBIO6   LDA      0,2
        SKG      1,2          TRANSFER COMPLETED ?
        BRU      GBIO3        NO
        MIN      RLV2
        BRR      RLV2

```

```

GBIO3   ETR      =3777B          MOVE BLOCK-FOR-TRANSFER INTO PAGE 4 OF T
        ADD      =200000B
        STA      EXSX
        LDA      0,2

```

ETR = 34000B
 ADD = 4000B-1
 STA GBIO2
 LDA 1,2
 SKG GBIO2
 STA GBIO2
 LDA GBIO2
 LRSH 11
 SUB =3
 LDB 3,2
 STP GBIOCR
 COPY AX,BA
 SBRM ERB
 STA EXSR1
 LDB EXRL2
 LSH 6
 LDA EXSR1
 RSH 6
 LDA EXRL1
 STP EXSR1

MAKE TRANSFER TO END OF BLOCK

IS THIS PAGE THE LAST ?
 YES, ADJUST FINAL TRANSFER ADDR
 DETERMINE RELABELLING FOR THIS PAGE

FORK MAP FOR P4=MAP FOR PAGE FOR XFER
 OTHER PAGES = EXEC MAP

LDX GBIOPT
 LDA GBIO2
 SUB 0,2
 ADD =1
 STA EXSA
 ADM 0,2
 LDA GBIOX-1
 SKN GBIOFN
 LDA GBIOX
 STA EXSL
 LDA EXSL7
 BRS 9
 BRS 31
 STX GBIO4
 LDX =-4
 LDA =EXSR1
 SBRM ERB
 LDX GBIOCR
 LDB GBIOCR2
 SBRM SRB
 SKR GBIO4
 BRR RLV2
 LDX GBIOPT
 BRU GBIO6
 \$EXSL7 NOP EXSL,6

NO. OF WORDS TO BE TRANSFERRED
 UPDATE TRANSFER ADDRESS FOR NEXT PAGE

TRANSMIT ANY MEMORY GRABBED BY TRANSFER
 ... OPERATION TO TRANSFER-AREA'S RELABEL

CHECK STATUS OF TRANSFER-FORK ON TERMINAL

IF NORMAL EXIT (BRS 10), RTN FOR NEXT PAGE

ZRO GBIODK
 GBIOX ZRO GBIOS

* FORK FOR GENERAL SEQUENTIAL I/O

GBIOS BIO GBIOFN
 HLT 0

* KLUDGE FORK FOR DRUM INPUT

GBIODK	LDB	20000B	
	LDB	=20000B	GRAB A BLOCK FOR PAGE 4
	LDA	GBIOFN	
	ETR	=3777777B	GET RAD ADDRESS
	XAB		
	LDX	GBIOSW	
	BRS	104,2	READ OR WRITE 2K ON RAD
	BRU	GBDKE,2	
	LDA	=4000B	
	ADM	GBIOFN	UPDATE RAD ADDRESS
	BRS	10	
GBDKE	BRU	++4	
	MSET	M1	
	HLT		
	MSET	M2	
	HLT		

* 2.5

THIS SUBROUTINE READS A SAVE FILE INTO THE RELABELING ADDRESS

* THE FILE NUMBER IS IN A. IT SKIPS IF IT IS SUCCESSFUL.

SGET	ZRO	RLV3	
	STA	GETFN	
	STX	GET5	
	SKG	X0	USER FILE
	BRU	++2	
	BRU	GET20	YES
	STB	GET2	CONVERT DRUM 'FILE NO.' TO ACTUAL DRUM B
	ETR	=777B	
	CLB		
	LSH	11	
	MRG	X4	ADD "DRUM-FILE" MARKER BIT
	STA	GETFN	
	LDA	GET2	EXTRACT INITIAL AND FINAL CORE ADDRESSES
	ETR	=37777B	
	XMA	GET2	
	ETR	=740000B	
	LRSH	3	
	SUB	=1	
	ADD	GET2	
	STA	GET3	
	CLA		
	XMA	GETFN	
GET20	BRU	GET21	READ INITIAL CORE ADDRESS
	WIO	GETFN	
	STA	GET2	
	WIO	GETFN	READ FINAL CORE ADDRESS

	Sta	GET3	
	WIO	GETFN	READ STARTING LOCATION
	STA	GETSTL	
GET21	LDA	GETFN	
	CLB		
	LDX	=GET2	
	SBRM	GBIO	TRANSFER INTO SPECIFIED CORE RANGE
	BRU	*+2	
	MIN	RLV3	
	LDA-	GETFN	CLOSE FILE
	SBRM	BRS20	
	BRR	LV3	

* 2.2 RELEASE CORE INDICATED BY MAP ADDRESSED BY X

\$RELST	ZRO	RLV3
	COPY	XB,A
	SBRM	RRR
	CLAB	
	STP*	DRSW9
	BRR	RLV3

* 2.3 GET NAME AND OPEN O/P FILE

\$GOFN	ZRO	SWOFF
	STP	GOFNS
GOFNR	LDX	== 1
	CLAB	
	SBRM	BRS58
	SBRM	NEWTRY
	STX	DEL3
	LDB	GOFNS
	LDX	GOFNS1
	SBRM	BRS19
	BRU	*- 5
	STA	OUTFIL
	LDB	DEL3
	BRR	SWOFF

* 2.4 GET NAME AND OPEN I/P FILE

\$GIFN	ZRO	SWOFF
	CLX	
	CLAB	
	SBRM	BRS58
	SBRM	NEWTRY
	SBRM	BRS16
	BRU	*- 2

STA INFIL
BRR SWOFF

* 1.1 NAME INCORRECTLY DELIVERED, TRY AGAIN
\$NEWTRY ZRO RETURN1
 MSET E1
 LDX RETURN1
 BRU -3,2

\$CKRLB ZRO RLV4 CALL WITH PTR TO RLBLG IN A
 STA DEL2
 LDX =-8
CKRLB1 LDA DEL2
 SBRM ERB
 SKG =0
 BRU *+3
 SKG =NCMEM
 BRU ERRET
 BRX CKRLB1
 BRR RLV4

* 0.11 EXTRACT BYTE FROM RELABELLING
\$ERB ZRO RLV1
 ETR =37777B
 STA ERB2
 LDP* ERB2
 EXU ERBSH,2
 ETR =77B
 BRR RLV1

* SHIFT TABLE TO GET BYTE
RCY 18
RCY 12
RCY 6
NOP
LCY 6
LCY 12
LCY 18
LCY 24
ERBSH EQU *

* SHIFT TABLE TO RESTORE BYTE
LCY 18
LCY 12
LCY 6
NOP
RCY 6
RCY 12
RCY 18
RCY 24
SRBSH EQU *

* 0.12 SET RELABELLING BYTE

\$SRB	ZRO	RLV1
	ETR	=77B
	STP	ERB2
	LDP*	ERB3
	EXU	ERBSH,2
	ETR	=7777700B
	MRG	ERB2
	EXU	SRBSH,2
	STP*	ERB3
	BRR	RLV1

* 1.8

* THIS ROUTINE DECODES A RELABELING STRUCTURE WORD IN A AND CONSTRUCTS
 * TWO RELABELING WORDS WHICH IT RETURNS IN AB.
 * THE RSW CODES ARE

*	0	SET TO 0. RELEASE ON RETURN
*	1	SET TO SUBSYSTEM BYTE. USE TO RESET BYTE ON RETURN.
*	2	SET TO EXEC BYTE. USE TO RESET BYTE ON RETURN
*	3	SET TO PROGRAM BYTE. USE TO RESET BYTE ON RETURN
*	4	SET TO TS. IGNORE ON RETURN
*	5	SET TO 0. RESET SUBSYSTEM BYTE TO 0 ON RETURN.

\$DRSW	ZRO	RLV2
	STA	DRSW2
	SBRM	EXECRL
	LDX	--3
DRSW5	LDB	DRSW2
	CLA	
	LSH	3
	STB	DRSW2
	COPY	XB,AX
	EXU	DRSWX,2
	CBX	
	SBRM	ERB
DRSW7	LDB	=DRSW3
	SBRM	SRB
	BRX	DRSW5
	LDP	DRSW3
	BRR	RLV2
DRSWX	BRU	DRSW6
	LDA	=SSRL
	LDA	=EXRL
	LDA	=PRGRL
	BRU	DRSW8
	BRU	DRSW8
	HLT	0
	HLT	0
DRSW6	COPY	BX,A
	BRU	DRSW7
DRSW8	LDA	EXRL1
	CBX	
	LRSH	18

BRU DRSW7

* 1.6

* CODE TO RESET RELABELLING REGS ACCORDING TO RSW IN A.
* RELABELING REGISTER ADDRESS IN B. THIS ROUTINE CAN ALSO BE USED T
* RELEASE SELECTED BLOCKS BY SETTING THE BW BYTES TO 0 FOR THE BLOCK
* TO BE RELEASED + TO 1 FOR ALL THE OTHERS. IT IS ONLY LESS
* INEFFICIENT THAN THE MONITOR RELEASE LOGIC

```
$RRR  ZRO      RLV2
      LDX      =-8
      STA      DRSW2
      CBA
      ETR      =37777B
      STA      DRSW9
RRR1  STX      DEL3
      LDB      DRSW2
      CLA
      LSH      3
      STB      DRSW2
      COPY     XB,AX
      EXU      RRRX,2
      CBX
      STA      DRSW3
      LDA      DRSW9
      SBRM     ERB
      LDB      DRSW3
      SBRM     SRB
RRR2  LDX      DEL3
      BRX      RRR1
      BRR      RLV2
RRRX  BRU      RRR3
      LDA      =SSRL
      LDA      =EXRL
      LDA      =PRGRL
      BRU      RRR2
      BRU      RRR5
      HLT      0
      HLT      0
RRR3  CBX
      LDA      DRSW9
      SBRM     ERB
      SKG      =NCMEM
      BRU      RRR2
      BRS      121
      BRU      RRR2
RRR5  CLA
      LDB      =SSRL
```

LDX DEL 3
BRU RRR2-1

* 1.3 PRINT CURRENT SUBSYSTEM NAME
\$P SCN ZRO RLV2
SKN SY STL
BRU *+2
BRU ERRET
LDA SY STL
SUB = SUBTB1
RSH 2
COPY AX
LDA COMTBL, 2
SBRM XTMSG
BRR RLV2

* 4.1 ACTIVATE A SUBSYSTEM WITH SUBSYS HASHT ADDRESS IN A, READ
\$G SYS ZRO RLV5
SKE SY STL SUBSYST ALREADY ALLOCATED TO THIS USER ?
BRU *+2
BRR RLV5 YES, NOTHING TO DO, RETURN
SBRM RSYS NO, RELEASE CURRENT SUBSYS, IF ANY
BRS 46
STA SY STL
CAX
STA SY SRRL
LDA -1, 2
SKA X1
BRU *+2
BRU GSYS5 NO, READ IT IN
LDP* SY SRRL YES, SET UP COMMON RELABELLING
STP SSRL
BRU GSYS9
GSYS5 LDX SY STL
CLAB
STA GET2
STP SSRL
LDB -1, 2
L SH 10
ETR =177B
MRG =40000000B
LDB* SY SRRL
LDX =SSRL
SBRM GET
BRU GSYS2 SUBSYSTEM INCORRECTLY READ IN
GSYS9 BRS 47
BRR RLV5

GSYS2	LDA	=ERRTM1	SET ERROR EXIT
	STA	RLV5	
	LDA	== -1	
	STA	SYSTL	RECORD NO SUBSYSTEM IN CORE
	EAX	SSRL	
	SBRM	RELST	RELEASE ANY MEMORY GRABBED BY READ-IN PR8
	BRU	GSYS9	

* 3.4 RELEASE THE CURRENT SUBSYSTEM

\$RSYS	ZRO	RLV4
	STA	TLV11
	LDA	== -1
	XMA	SYSTL
	EAX	SSRL
	SKE	== -1
	SBRM	RELST
	LDA	TLV11
	BRR	RLV4

\$FDFNA	ZRO	RLV3
	SBRM	RFINA
	BRU	ERRET
	BRU	ERRET
	BRU	ERRET
	SBRM	FDSCH
	BRR	RLV3
	MIN	RLV3
	BRR	RLV3

* PRINTS OUT FILE NAME

\$PFINA	ZRO	RLV1
	CLA	
	STA	TLV11
	STA	GOFNFL
	LDA	LSHCH
	CIO	COUT
PFINA1	LDA	TLV11
	MUL	=12525253B
	ADD	UNPTR
	ADD	=3
	CAX	
	LDA	0,2
	CAX	
	LSH	5
	ETR	=30B
	CNA	
	COPY	AX, XA, B
	LRSH	16,2

ETR =377B
 CIO COUT
 MIN TLV11
 SKG =77B
 MIN GOFNFL
 SKG =177B
 BRU PFINA1
 LDA SL SHCH
 CIO COUT
 BRR RLV1
 * PRINTS UFD ENTRY

\$FPRINT ZRO	RLV3	
LDA	GOFNFL	
SUB	=12	
COPY	AX,A	
CIO	COUT	
BRX	*-1	
LDA*	UNPTR	GET WORD CONTAINING FILE LENGTH
ETR	=37777777B	TAKE OFF SIGN BIT
SBRM	FSP	PRINT OUT THE DECIMAL LENGTH
LDX	UNPTR	
LDA	2,2	GET WORD CONTAINING FILE TYPE
RSH	17	
ETR	=7	
CAX		
LDA	T1,2	ADDRESS OF PROPER TYPE TO BE PRINTED
LDX	COUT	TELETYPE NUMBER
LDB	--1	
BRS	34	PRINT TYPE
LDX	UNPTR	
LDA	2,2	GET WORD TO TEST FOR SEQ. OR RDM.
SKA	X2	
BRU	FI5	
MSET	M19	BIT IS OFF-- ,SEQ
BRU	FI6	
FI5	MSET	BIT IS ON-- ,RDM
FI6	SKN*	CHECK FOR PRIVATE FILE
BRU	FI8	NOT PRIVATE
MSET	M45	IS PRIVATE-- PRIV
BRU	FRJ	GO IMMEDIATELY TO READ ONLY CHECK
FI8	LDX	
LDA	2,2	GET WORD FOR ACCT PUB OR PUB CHECK
SKA	X1	
BRU	FI10	SIGN BIT IS ON
MSET	M46	SIGN BIT IS OFF-- ACCT
BRU	FRJ	GO TO READ ONLY TEST
FI10	MSET	SIGN BIT IS ON-- PUB
FRJ	LDX	READ ONLY TEST
SKN	1,2	SKIP IF READ ONLY BIT IS ON
BRU	FI12	
MSET	M39	READ ONLY-- ,RO
FI12	LDX	
LDA	2,2	

	SKA	= 4000000B	
	BRU	*+2	
	BRU	FI 13	
	MSET	M33	'(DATA MISSING) / '
FI 13	MSET	M21	CARRIAGE RETURN
	BRR	RLV3	EXIT

\$T1	DATA	M51
	DATA	M60
	DATA	M59
	DATA	M58
	DATA	M52

\$FSP	ZRO	RLV2
	CLB	
	LDX	== 6
	SKG	TEN+6,2
	BRU	*+3
	SBRM	PDNO
	BRR	RLV2
	XAB	
	CIO	COU
	XAB	
	BRX	FSP+3
	BRU	*-6
TEN	DATA	999999,99999,9999,999,99,9

* PRINTS CPU AND LINE TIME

\$HOUR	ZRO	RLV4	
	STP	UNPTR	
	MSET	M31	'CPU: '
	LDA	UNPTR	
	RSH	23	
	DIV	= 3600	
	SBRM	PLZZ	
	RSH	23	
	DIV	= 60	
	SBRM	PLZZ	
	SBRM	PLZ	
	MSET	M26	
	LDA	UNPTR1	
	RSH	23	
	DIV	= 60	
	SBRM	PLZZ	
	SBRM	PLZ	

```

MSET M21
BRR RLV4

PLZ ZRO RLV2
SKG =9
BRU *+3

PLZ1 SBRM PDNO
BRR RLV2
CAX
LDA ZEROCH
CIO COUT
CXA
BRU PLZ1

PLZZ ZRO RLV3
STB UNPTR
SBRM PLZ
LDA COLNCH
CIO COUT
LDA UNPTR
BRR RLV3

```

* PRINTS TIME OF DAY

```

SDAYTIM ZRO RLV5
SUB X4
RSH 23
DIV =3600
ADD DMIN
RSH 23
DIV =44640
CAX
LDA MONTH,2
LDX =-3
CIO COUT
RCY 8
BRX *-2
RSH 23
CIO COUT
DIV =1440
STB TLV2
ADD =1
SBRM PDNO

```

```

MSET M73; LDA TLV2; LDB = ' P'; SKG =720; LDB = ' A'
STB RLV2; SKG =59; ADD =1440; SUB =720; SKG =59; ADD =720; RSH 23
DIV =60; STB TLV2; RSH 23; DIV =10; ADD =20B; SKE =20B; CIO COUT
CBA; ADD =20B; CIO COUT; LDA COLNCH; CIO COUT; LDA TLV2
RSH 23; DIV =10; ADD =20B; CIO COUT; CBA; ADD =20B; CIO COUT
CLA; CIO COUT; LDA RLV2; CIO COUT; MSET M76; BRR RLV5

```

* PRINTS TIME USED FOR LOG OUT

```

$GETTIM ZRO RLV1
        SUB REAL
        RSH 23
        DIV ==-3600
        COPY AX
        LDA* TJOB
        RSH 23
        DIV =60
        COPY XB
        BRR RLV1

```

* READS SPACES, THEN TWO CHARS AND A SPACE

```

$SEP ZRO RLV3
      SBRM DCHAR3
      BRR RLV3
      BRU ERRET

```

* SETS UP TIMING FORK
 * A= NO. OF MILLISEC

* X=ADDR OF INTERRUPT ROUTINE

```

$TFORK ZRO RLV1
        STA EXSA
        STX EXSX
        LDA =100000B
        BRS 78 ARM INTERRUPT 5
        LDA EXRL1
        CLB
        STP EXSR1
        LDA =TMFK
        STA EXSL
        LDA EXSL7
        BRS 9 START UP TIMING FORK
        BRR RLV1

```

* TIMING FORK

```

TMFK BRS 81 DISMISS FOR SPEC TIME
      CXA
      ETR ADMSK
      STA 205B
      LDA =5
      BRS 79 FIRE INT 5
      BRS 10

```

* SET UP TIME OUT

```

STIMOUT ZRO RLV1
          BRS 46
STM01 MSET M62
        LDX =-1
        BRS 14
        BRU LGOUT3

```

'PLEASE CALL, ETC'

WAIT FOR OUTPUTBUFFER TO EMPTY

* FINDS FILE DIRECTORY LOCATION ON DISC

```

$FFD ZRO RLV1
      LDA UN01
      CLB
      LSH 2
      ADD =400B
      CAB 0
      BRR RLV1

```

(A) = (B) = DISC ADDRESS

* CHECK FOR VALID PASSWORD

```

$NMC ZRO RLV1
      CLX
      LDA XWR4,2
      SKE DTEMP,2
      BRR RLV1
      EAX 1,2
      LDA XWR4,2
      SKE DTEMP,2
      BRR RLV1
      MIN RLV1
      BRR RLV1

```

* READ GROUP NO FROM TTY AND AUD FROM DISC

```

$FNAC ZRO RLV2; SBRM RNAC; BRU ERRET; STA SAVELL
      COPY AB; LDA =DTEMP; LDX =64; BRS 61; HLT; BRR RLV2

```

```

$RNAC ZRO RLV1; LDX =-3; LDB =2; COPY XA,B,BX; STA DTEMP,2
      EAX 12,2; STA DTEMP,2; COPY XB,AX,BA

```

```

RNAC1 MUL =5; STB SAVELL; CIO CIN; SKG =31B; SKG =17B
      BRR RLV1; SUB ZEROCH; ADD SAVELL; BRX RNAC1; SUB =100
      SKG =-1; BRR RLV1; SKG =MAXACT; MIN RLV1; BRR RLV1

```

* SETS DATE

```

$SSDAT ZRO RLV2; MSET M41; SBRM RDNO; SUB =1; MUL =22320
        STB TLV1; SBRM RDNO; SUB =1; MUL =720; CBA; ADM TLV1
        SBRM RDNO; RSH 23
        DIV =100
        COPY BA,AB
        ADM TLV1
        COPY BA,AB
        MUL =30
        COPY BA,B
        ADM TLV1
        LDA REAL
        SUB X4

```

RSH 23
DIV =3600
SUB TLV1; CNA
STA DMIN
BRR RLV2

* GET STRING FROM TTY. 6 CHARS MAX TERMINATE ON SEMICOLON

\$GTS ZRO RLV2
COPY A,B
STP XWR4
LDX --1; LDA =3; BRS 12
LDX =5

GTS1 STX TLV1
LDA TLV1
LSH 3
COPY AX,A
CIO CIN
SKE CRCH
BRU **2
BRU GTS4
SKE OKCH
BRU GTS3

GTS2 MIN RLV2
GTS4 LDX --1; CLA; BRS 12

BRR RLV2
GTS3 LCY 0,2
COPY AB,BA
ADM XWR4
COPY BA,B,X
EAX 1,2
ADM XWR4,2
SKR TLV1
BRU GTS1
BRU GTS2

\$CKTIM ZRO RLV2
LDA REAL
SUB X4
RSH 23
DIV =3600
ADD DMIN
RSH 23
DIV =1440
CLA
DIV =30
CAX
LDA X4
LRSH 0,2
BRR RLV2

CONVERT TO HOUR BIT

* FIND NAME AND VERIFY

\$NMV ZRO RLV2
LDX --52

```

NMV2  LDP      DTEMPE,2
      SKE      XWR2
      BRU      NMV1
      CBA
      SKE      XWR3
      BRU      NMV1
      MIN      RLV2
      EAX      DTEMPE,2
      BRR      RLV2
NMV1  EAX      3,2
      BRX      NMV2
      BRR      RLV2

```

* PUTS 1ST SIX 8-BIT CHARS FROM CIN IN XWR2,-3.
* PUTS MAX OF TEN MODULO-27 CHARS IN XWR8,-9 FOR WERIS,
* IGNORING CONTROL CHARACTERS, TERMINATING ON ANY NON-CONTROL
* NON-ALPHA.

```

$NMVF  ZRO      RLV2
      LDA      =400000006B
      STA      XWR1
      LDA      EQXWR8
      STA      XWR5
      LDA      =1
      STA      XWR6
      LDA      =4
      STA      XWR7
      CLA
      STA      XWR2
      STA      XWR3
      STA      XWR8
      STA      XWR9
NMVF1  LDA*     XWR5
      MUL      =27
      RSH      1
      CIO      CIN
      SKG      =72B
      SKG      =40B
      BRU      NMVF2
      STB*     XWR5
      SUB      =40B
      ADM*     XWR5
      ADD      =40B
      SKR      XWR7
      BRU      NMVF3
      MIN      XWR5
      LDB      =4
      STB      XWR7
      SKR      XWR6
      BRU      *+2
      BRR      RLV2
NMVF3  SKR      XWR1
      BRU      NMVF1
      COPY     AX,B
      LDA      XWR1

```

	LSH	3
	COPY	XB,AX,A
	LSH	0,2
	ADM	XWR2
	COPY	BA
	ADM	XWR3
	BRU	NMVF1
NMVF2	SKG	=100B
	BRR	RLV2
	BRU	NMVF3
EQXWR8	ZRO	XWR8

* UNIVERSAL CONSTANTS AND WORKING SPACE FOR EXEC.

*TABLE OF CHARACTERS

\$RPCH	ZRO	11B
\$LPCH	ZRO	10B
\$DOLCH	ZRO	4
\$LINF	ZRO	152B
\$CRCH	ZRO	155B
\$QMCH	ZRO	37B
\$SEMIC	ZRO	33B
\$LSHCH	ZRO	17B
\$GRCH	ZRO	36B
\$MINCH	ZRO	15B
\$BLKCH	ZRO	0
\$COLNCH	ZRO	32B
\$MBCH	ZRO	135B
\$DOTCH	ZRO	16B
\$STARCH	ZRO	12B
\$ZEROCH	ZRO	20B
\$EOTCH	ZRO	144B
\$QUOTCH	ZRO	2
\$COMCH	ZRO	14B
\$TERMCH	ZRO	137B

*STANDARD RESPONSES - FOR OUTPUT TO COMMAND O/P MEDIUM

\$E1	ASC	' ? / '
\$M1	ASC	'RAD READ ERRORS/ '
\$M2	ASC	'RAD WRITE ERRORS/ '
\$M3	ASC	' START AT / '
\$M4	ASC	'\$FROM / '
\$M5	ASC	' ON LINE / '

```

$M6   ASC      ' ATTEMPTED LINK$/ '
$M7   ASC      '(PE)/ '
$M8   ASC      '(PC)/ '
$M9   ASC      'DISC READ ERRORS/ '
$M10  ASC      'DISC WRITE ERRORS/ '
$M11  ASC      '$BUSY$/ '
$M12  ASC      '$DEVICE IS IN USE$/ '
$M13  ASC      '$IN 8-LEVEL MODE$/ '
$M14  ASC      '$NOT ACCEPTING LINK$/ '
$M15  ASC      '$$ FILE NAME      LENGTH      TYPE      STATUS$$/ '
$M16  ASC      ' ON / '
$M17  ASC      '$TAPE IS IN FILE PROTECTS/ '
$M18  ASC      '$TIME USED / '
$M19  ASC      '$SEQ / '
$M20  ASC      '$NO FILE CONTROL BLOCKS AVAILABLE$/ '
$M21  ASC      '$/ '
$M22  ASC      ' FILES $$/ '
$M23  DATA 33265040B (CR,LF,AT SIGN); ASC '/ '
$M24  ASC      '$DEVICE NOT ON-LINES/ '
$M25  ASC      '$YOU HAVE EXCEEDED YOUR DISC ALLOTTMENTS/ '
$M26  ASC      '$CONNECT: / '
$M27  ASC      'MAX = / '
$M29  ASC      '$TAPE IS IN USE$/ '
$M30  ASC      '$NOT ENTERED$/ '
$M31  ASC      '$CPU: / '
$M32  ASC      ',000$/ '
$M33  ASC      ' (DATA MISSING)/ '
$M34  ASC      '$ILLEGAL TIMES/ '
$M35  ASC      '$STATUS: / '
$M36  ASC      ', MAX USED = / '
$M38  ASC      ', EX/ '
$M39  ASC      ', RDO/ '
$M40  ASC      '$ALL? / '
$M41  ASC      '$ENTER [MO-DAY-TIME]: / '
$M42  ASC      '$HARVARD TIME SHARING SYSTEM (D00-H20): 8-6-68 $/ '
$M43  ASC      '$TIME RUN OUT, PLEASE LOGOUT (3 MIN)/ '
$M44  ASC      'RDM / '
$M45  ASC      'PRIV/ '
$M46  ASC      'ACCT/ '
$M47  ASC      'PUBL/ '
$M48  ASC      '$DISC: MAX WORDS USED= / '
$M49  ASC      '$TOTALS FOR THIS BILLING PERIOD$/ '
$M50  ASC      ' TO / '
$M51  ASC      ' UNDF,/ '
$M52  ASC      ' DUMP,/ '
$M54  ASC      ', IN USE = / '
$M56  ASC      '$GROUP NO:/ '
$M57  ASC      '$INVALID USER/ '
$M58  ASC      ' SYMB,/ '
$M59  ASC      ' BINA,/ '
$M60  ASC      ' SAVE,/ '
$M62  ASC      '$IF YOU NEED ASSISTANCE, CALL EXT. 3762 $$/ '
$M63  ASC      '$RESTART MISSING/ '
$M64  ASC      '$I-O ERROR. FILE HAS BEEN DELETED.$/ '

```

\$M65 ASC 'SILL. INSTR. AT /'
\$M66 ASC 'MEMORY TRAP AT /'

\$M67 ASC ' PAGE /'
\$M68 ASC ' BAND /'
\$M69 ASC ' AS /'
\$M70 ASC ' HAS LINKEDS /'
\$M71 ASC '\$PASSWORD: /'
\$M72 ASC '\$NAME: /'
\$M73 ASC ', 1969 /'
\$M74 ASC '\$NEW NAME: /'
\$M75 ASC ' FROM /'
\$M76 ASC '.M.\$ /'
\$MONTH ASC 'NAJBEFRAMRPAYAMNUJLUJGUAPESTCOVONCED'

END