Table of contents

```
                                        .TITLE  TSKMN1 -- TSX-Plus Keyboard Monitor
                                        .ENABL  LC
                                        .DSABL  GBL
                                        .ASECT
000042                          .       =       42
000042  003716'                        .WORD   KMSTK           ;DEFINE INITIAL STACK POINTER
000300                          .       =       300
000300  000000G                        .WORD   MDT             ;# MINUTES OF SYSTEM UPTIME IF DEMO VERSION
                                        .CSECT  TSKMON
                                TSKMON:
                                TSKMN1:
                                ;
                                ;  TSKMON is the terminal command processing module of the TSX
                                ;  operating system.
                                ;
                                ;  TSKMON is broken into three parts:
                                ;
                                ;  TSKMN1 is the portion of TSKMON that contains the code to acquire
                                ;  a command and branch off to the appropriate command processing routine.
                                ;
                                ;  TSKMN2 contains the actual command processing routines.
                                ;
                                ;  TSKMN3 contains subroutines.
                                ;
                                ;  TSKMON runs in user mode and has memory mapping set up as follows:
                                ;   000000-040000 --> TSGEN
                                ;   140000-157777 --> Job context block
                                ;   160000-177777 --> Monitor vector (MONVEC)
                                ;
                                ;  Copyright 1978,1979,1980,1981,1982,1983,1984,1985,1986,1987,1988.
                                ;  S&H Computer Systems,  Inc.
                                ;  Nashville,  Tennessee
                                ;
                                ;
                                        .MCALL  .CSISPC,.TTOUTR,.SRESET
                                        .MCALL  .READW,.TTYIN,.TTYOUT,.PURGE
                                        .MCALL  .CSIGEN,.SAVEST,.REOPEN
                                        .MCALL  .GTLIN,.GTIM,.DATE
                                        .MCALL  .PRINT,.CLOSE,.LOOKUP
                                        .MCALL  .WRITW,.ENTER,.EXIT
                                        .MCALL  .SERR,.HERR,.FPROT,.GVAL,.PVAL
                                ;
                                ;  Global definitions
                                ;
                                        .GLOBL  INGEMT,INGADR,INPEMT,INPADR,IIBUF
                                        .GLOBL  TSKMON,GREET,PRTGRT,GRTFIN,KMNOT1
                                        .GLOBL  LGOVER,BOTEMT,KEYBUF
                                        .GLOBL  START,PBUFND
                                        .GLOBL  REMNDR,KEYEND,KMSTK
                                        .GLOBL  ALDEMT,DLCEMT,TALEMT,ALCDEV,CDBUF
                                        .GLOBL  TSKMON,TSKMN1
                                        .GLOBL  EM$NUK,ESC,UCIDEF
                                        .GLOBL  GENMON,CDGEMT,CDGADR,CDPEMT,CDPADR
                                ;
                                ;  Global references
                                ;
                                        .GLOBL  KMONCE,K    T,EXCJOB
```

```
                .GLOBL  $NOVLN, P2$VIR, PRIVA2, PRIVS2, SC$ERR, SUCF2
                .GLOBL  INSSRC, II$PRV, II$NPV, AFCF, II$FLG, EM$NUC
                .GLOBL  ISPF5, ISPF6, ISPF7, ISPF9, PRIVFO, SKPSPC, ISPF11
                .GLOBL  LBSPRI, GETSYP, JPWDEV, JPWTYP, SBPSUF, CFSTRT
                .GLOBL  ABRTAD, ABRTCD, CINFLG, INIUKD, R50KMN, ABRTOV
                .GLOBL  CORUSR, LSW, SERFLG, KMPRMT
                .GLOBL  UTRPAD, JSWLOC, MAXMEM, MAXPRI, JPWFLG
                .GLOBL  $KINIT, CFSTK, DFJMEM, $SUCF
                .GLOBL  SPUBUF, SXBPNT, LWINDO, CMDSPN, CMDRSM
                .GLOBL  RUNCHN, RUNNAM
                .GLOBL  CMDHD, $SYSPS
                .GLOBL  VEDIT, WILDFL, $NOIN, $NOWTT, $CHACT
                .GLOBL  PRIVCO, PRIVAO, PRIVSO, PVNPW, RUNARG
                .GLOBL  JS$KMN, JS$ON, LMONHD, EM$OVL, II$$SZ, RSTPRV, EM$SFP
                .GLOBL  MSTALC, ALCERR, ABM24, ABM23, ABM22, ABM21, ABM20, SMONHD
                .GLOBL  ABM17, ABM16, ABM15, ABM14, ABM13, ABM12, ABM11, JS$LOG
                .GLOBL  ABM10, ABM7, ABM6, ABM5, ABM4, ABM3, ABM2, ABM1, AB1
                .GLOBL  AB2, AB3, AB4, AB5, AB6, AB10, AB11, AB12, AB13, AB14, AB15, AB16
                .GLOBL  REGEN, STDNAM, IMVT10, IMVT52, IMADM3, IMHAZL, MONTXT, FILERM
                .GLOBL  LOCMSG, TOOLNG, NOCF, NUMPRM, STRLEN, ABCMD, MISUCL
                .GLOBL  URCMD, UNSUP, LINNTX, CMDINS, TM$LN1
                .GLOBL  TECO, EDIT, $1STLG, $DIBOL, HANCHN
                .GLOBL  CMDYEL
                .GLOBL  CMDRSY, CMDRUN, CMDRST, CMDSHO, CMDSET, CMDSND
                .GLOBL  CMDWHO, CMDFMT, CMDDAT, CMDTIM, CMDMNT, CMDMON
                .GLOBL  CMDDMT, CMDDSP, CMDASN, CMDALC, CMDDLC, CMDREM
                .GLOBL  CMDMEM, OPRCMD, CMDSPO, CMDKIL, CMDPAU, CMDDET, CMDRCL
                .GLOBL  CMDACC, CMDUSE, CMDINI, CMDSQZ, UCLCMD, CMDBOT, CMDNAC
                .GLOBL  CMDSHT, PROFLG, CMDDEF
                .GLOBL  LSW9, $DEBUG, LSW11
                .GLOBL  HAZEL, HAZLFL, HAZLNO, MDT
                .GLOBL  LRDTIM
                .GLOBL  $VTESC
                .GLOBL  UCISPC
                .GLOBL  LAFSIZ, LFWLIM, ILSW2
                .GLOBL  $SLLET
                .GLOBL  LSUCF, $CCLRN, $UKMON, $UKMRN
                .GLOBL  KL3CLR, $PRGLK, LSW5
                .GLOBL  LSTDL, $DETCH
                .GLOBL  LPROJ, LPROG, LUNAME, UPPN
                .GLOBL  LCONTM
                .GLOBL  USPLCH, SPLCHN
                .GLOBL  CFBUF, CFEND, CCLSAV
                .GLOBL  MINTIM, LSECPT
                .GLOBL  OKFILE, OKFEND, $CLTST, $NOINT
                .GLOBL  ERRSPC
                .GLOBL  GENTOP, CMDOFF
                .GLOBL  $CTRLC, LSW2, UFORM, $INDAB
                .GLOBL  $CFABT, INDSTA, INDERR
                .GLOBL  UHIMEM
                .GLOBL  LSW3, LSW2S, $DUPRN
                .GLOBL  LSCCA, $RNIOP, $SETRN
                .GLOBL  $ECHO, $LC
                .GLOBL  UCHAN, $CFDCC, $CFCCL
                .GLOBL  LNPRIM, $NTGCC
                .GLOBL  $DOOFF, LRBFIL, CFIND, LPARNT
                .GLOBL  CD$$SZ
```

```
                        .GLOBL  LTSCMD,LNSPAC,VU$CL,UCLNAM
                        .GLOBL  $CFOPN,PBFEND,CFSP,VUCLMC
                        .GLOBL  UFPTRP,$UCLCF,VUCLOR
                        .GLOBL  $UCLRN
                        .GLOBL  NSPLDV,$UCLCM,$UCLCL
                        .GLOBL  $DEFER,CFCHAN,SCHAIN
                        .GLOBL  CFPNT,$QUIET,DIABFL,KDOCIN,CFSQEZ
                        .GLOBL  DIABNO,VT52NO,LA36NO,LA36FL,POPCF
                        .GLOBL  LSW4,KL4CLR
                        .GLOBL  $KED
                        .GLOBL  PRMPNT
                        .GLOBL  LSTPRM,PRMBUF,PRMEND,CFSPND
                        .GLOBL  LCOL,$TECO
                        .GLOBL  $WILD,ERRSEV,UERSEV,PASLIN
                        .GLOBL  LSTPL
                        .GLOBL  KMPRMT,MXPRMT
                        .GLOBL  SJEMT,RJEMT
                        .GLOBL  CINDAT
                        .GLOBL  TSXLN,TSXSIT,GRT1,TRGRET,LICTXT,SUPCOD,NAMTOP,SUMS,SUCS
                        .GLOBL  LPRG1,LPRG2
                        .GLOBL  LCBIT,LA36,LA120,VT52,VT100,DIABLO,QUME
                        .GLOBL  ADM3A,LTRMTP,LA12FL,LA12NO,VT52FL
                        .GLOBL  VT10FL,VT10NO,ADM3FL
                        .GLOBL  ADM3NO
                        .GLOBL  RESDEV,OKFAND,UKFNND
                        .GLOBL  VT2007,VT2008,VT20NO,VT20FL
                        .GLOBL  LSW6,$SNWTT
                        .GLOBL  $INDDF,$INDRN,IN$ACT,IN$CNT,IN$CMD,INDSAV
                        .GLOBL  LSW7,$CFKIL
                        .GLOBL  MXJPRI
                        .GLOBL  LOGCHN
                        .GLOBL  UCLBLK,UCLDAT
                        .GLOBL  UC$NDC,UC$MDC
                        .GLOBL  PLOAD,DORUN,CMDFRM,CMDDSN,DATTIM,PRGALL
                        .GLOBL  LDCLEN,R50TSX,R50UCL,INDABT
                        .GLOBL  RDCMD,DKSAV,SYSAV,SEARCH
                        .GLOBL  FKILL,ABRTCF,ACRFN,XAREA,FPRINT
                        .GLOBL  PUSHCF,FILNAM,R50DIR,R50SY,R50IND,R50SAV
                        .GLOBL  INDACT,R50DUP,R50PIP,R50KED,R50K52,R50KEX,R50COM
                        .GLOBL  BLKO,RDERM,R50VIR,RUNEMT
                        .GLOBL  LDNAM,NOCIN,SIZVAL,ASKLNM,BADCMD,KCSIBF
                        .GLOBL  ASDEX,KCSIMS,R50BUF,R50LDO,MNTDEV,DMTARG
                        .GLOBL  NOFLAG,ILLCMD
                        .GLOBL  R50LD,R50DSK,R50LD7
                        .GLOBL  CSIMS2,R50NO,AMBOPT
                        .GLOBL  PRTDEC,DEVUNT,HANIDX,HNBUF
                        .GLOBL  CSIMS1,NOIND
                        .GLOBL  CRLF,R50LOG
                        .GLOBL  SPLHLA
                        .GLOBL  CPUAH,CPUAL,CMDBUF,CALUCL
                        .GLOBL  PRTBUF
                        .GLOBL  EDTFIL,MNFLGS,MNBPC
                        .GLOBL  MNBASE,MNTOP,MONAR1,MONAR2
                        .GLOBL  ALDEX,ALDBLK
                        .GLOBL  SPGEMT
                        .GLOBL  MSGBUF,MSGEND
                        .GLOBL  YELEMT
```

```
                                        .GLOBL  AUTHFN,OFFEMT,KILEMT
                                        .GLOBL  DIVSOR
                                        .GLOBL  SIZEMT,CSIMS4,MNTARG,HUPARG,R50TT
                                        .GLOBL  KMNNAM,CCLNAM,CMDCCL,CHKTTD
                                        .GLOBL  OCTPRT,CSIMS3,OVLYEQ
                                        .GLOBL  PRTR50,ALFN,R50DK
                                        .GLOBL  DETARG,R50MON
```

```
                                ;
                                ;  Assembly parameters
                                ;
                                ;  Greeting message control option
                                ;  If GREET=2, full greeting message is printed.
                                ;  If GREET=1, truncated greeting message is printed.
                                ;  If GREET=0, greeting message is not printed.  (SHOULD NEVER BE USED)
                                ;
000002                          GREET   =       2
                                ;
                                ;  Assembly constants
                                ;
000012                          LF      =       12              ;LINE FEED
000015                          CR      =       15              ;CARRIAGE RETURN
000040                          BLANK   =       40              ;ASCII SPACE
000007                          BELL    =       07              ;ASCII BELL
000011                          TAB     =       11              ;HORIZONTAL TAB
000014                          FF      =       14              ;FORM FEED
000033                          ESC     =       33              ;Escape character
000400                          BLKWDS  =       256.            ;# OF WORDS IN DISK BLOCK
000016                          HANCHN  =       16              ;Channel used to access handler files
                                ;
                                ;  Flags which are passed to CCL in chain location 510.
                                ;
000001                          C$TECO  =       1               ;USE TECO AS DEFAULT EDITOR
000002                          C$WILD  =       2               ;USE IMPLICIT WILDCARD NAMES
000004                          C$QUIT  =       4               ;SET TT QUIET IS IN EFFECT
000010                          C$TEST  =       10              ;DISPLAY CCL GENERATED COMMANDS
000020                          C$KED   =       20              ;DEFAULT EDITOR IS KED
000040                          C$K52   =       40              ;DEFAULT EDITOR IS K52
000100                          C$DIBL  =       100             ;DEFAULT TO DIBOL RATHER THAN DBL
```

```
                                        ;
                                        ;  General data area
                                        ;
000000  075250  102405  057760  UCIDEF: .RAD50  /SY UKMON SAV/  ;Default program for UCI
000006  073376
000010  000000  000000          DIVSOR: .WORD   0,0
000014  000000  000000          REMNDR: .WORD   0,0
000020  037266  023112  050572  R50MON: .RAD50  /JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC/
000026  004322  050601  040726
000034  040724  004617  073630
000042  057114  054756  014713
000050  000000G 000000G 000000G KCSIMS: .WORD   CSIMS1,CSIMS2,CSIMS3,CSIMS4
000056  000000G
000060                          IIBUF:  .BLKW   14.
```

```
                                ;
                                ;  Abort error message index table.
                                ;
000114  000000G                        .WORD   EM$SFP          ;-31
000116  000000G                        .WORD   EM$OVL          ;-30
000120  000000G                        .WORD   MSTALC          ;-27
000122  000000G                        .WORD   ALCERR          ;-26
000124  000000G                        .WORD   LGOVER          ;-25
000126  000000G                        .WORD   ABM24           ;-24
000130  000000G                        .WORD   ABM23           ;-23
000132  000000G                        .WORD   ABM22           ;-22
000134  000000G                        .WORD   ABM21           ;-21
000136  000000G                        .WORD   ABM20           ;-20
000140  000000G                        .WORD   ABM17           ;-17
000142  000000G                        .WORD   ABM16           ;-16
000144  000000G                        .WORD   ABM15           ;-15
000146  000000G                        .WORD   ABM14           ;-14
000150  000000G                        .WORD   ABM13           ;-13
000152  000000G                        .WORD   ABM12           ;-12
000154  000000G                        .WORD   ABM11           ;-11
000156  000000G                        .WORD   ABM10           ;-10
000160  000000G                        .WORD   ABM7            ;-7
000162  000000G                        .WORD   ABM6            ;-6
000164  000000G                        .WORD   ABM5            ;-5
000166  000000G                        .WORD   ABM4            ;-4
000170  000000G                        .WORD   ABM3            ;-3
000172  000000G                        .WORD   ABM2            ;-2
000174  000000G                        .WORD   ABM1            ;-1
000176  000000          ABTMSG:       .WORD   0               ;0
000200  000000G                       .WORD   AB1             ;+1
000202  000000G                       .WORD   AB2             ;+2
000204  000000G                       .WORD   AB3             ;+3
000206  000000G                       .WORD   AB4             ;+4
000210  000000G                       .WORD   AB5             ;+5
000212  000000G                       .WORD   AB6             ;+6
000214  000000G                       .WORD   RDERM           ;+7
000216  000000G                       .WORD   AB10            ;+10
000220  000000G                       .WORD   AB11            ;+11
000222  000000G                       .WORD   AB12            ;+12
000224  000000G                       .WORD   AB13            ;+13
000226  000000G                       .WORD   AB14            ;+14
000230  000000G                       .WORD   AB15            ;+15
000232  000000G                       .WORD   AB16            ;+16
```

```
                        ;
                        ;-------------------------------------------------
                        ;  MACRO TO CAUSE A FATAL ERROR MESSAGE TO BE PRINTED.
                        ;
                              .MACRO  FERR    MSG
                              MOV     R5,-(SP)
                              MOV     MSG,R5
                              CALL    FPRINT
                              MOV     (SP)+,R5
                              .ENDM   FERR
                        ;
                        ;-------------------------------------------------
                        ;  MACRO TO PRINT A FATAL ERROR MESSAGE, CLEAN UP
                        ;  AND THEN JUMP TO RDCMD.
                        ;
                              .MACRO  FABORT  MSG
                              MOV     MSG,R5
                              JMP     FKILL
                              .ENDM   FABORT
                        ;
                        ;-------------------------------------------------------------
                        ;  MACRO TO START A STANDARD OPTION TABLE.
                        ;  NAME = 1 TO 4 CHARACTER TABLE NAME.
                        ;  NA = NUMBER OF ARGUMENTS PER TABLE ENTRY.
                        ;
                              .MACRO  TBLDEF  NAME,NA
                        NARGS =       NA
                              .CSECT  CMDVK1
                        NAME'HD: .WORD  2*NA
                              .ENDM   TBLDEF
                        ;
                        ;-------------------------------------------------------------
                        ;  MACRO TO ENTER AN OPTION TEXT NAME AND A SET OF PARAMETERS
                        ;  INTO THE CURRENTLY OPEN TABLE.
                        ;  STRNG = ASCII NAME
                        ;  A,B,C = SET OF OPTION PARAMETERS TO STORE IN TABLE WITH NAME.
                        ;
                              .MACRO  CMDDEF  STRNG,A,B,C
                              .CSECT  NAMEK1
                        L     =       .
                              .ASCIZ  /STRNG/
                              .CSECT  CMDVK1
                              .WORD   L               ;POINTER TO NAME STRING
                              .WORD   A
                        .IIF  GE,<NARGS-2>    .WORD   B
                        .IIF  GE,<NARGS-3>    .WORD   C
                              .ENDM   CMDDEF
                        ;
                        ;-------------------------------------------------------------
                        ;  MACRO TO END A SET OF TABLE ENTRIES.
                        ;
                              .MACRO  TBLEND
                              .CSECT  CMDVK1
                              .WORD   0
                              .CSECT  TSKMON
                              .ENDM   TBLEND
```

```
                        .SBTTL  Table of command keywords
                ;-----------------------------------------------------------
                ;  Define commands
                ;
                ;  Arg 1 = command string name
                ;  Arg 2 = processing routine
                ;
000234                  TBLDEF  CMD,1
000002                  CMDDEF  R,CMDRSY
000006                  CMDDEF  RU*N,CMDRUN
000012                  CMDDEF  REN*AME,CMDCCL
000016                  CMDDEF  REM*OVE,CMDREM
000022                  CMDDEF  RES*UME,CMDRSM
000026                  CMDDEF  RESE*T,CMDRST
000032                  CMDDEF  COP*Y,CMDCCL
000036                  CMDDEF  DIR*ECTORY,CMDCCL
000042                  CMDDEF  ED*IT,CMDCCL
000046                  CMDDEF  TY*PE,CMDCCL
000052                  CMDDEF  PRI*NT,CMDCCL
000056                  CMDDEF  SH*OW,CMDSHO
000062                  CMDDEF  SE*T,CMDSET
000066                  CMDDEF  SEN*D,CMDSND
000072                  CMDDEF  YE*LL,CMDYEL
000076                  CMDDEF  W*HO,CMDWHO
000102                  CMDDEF  SY*STAT,CMDWHO
000106                  CMDDEF  DEL*ETE,CMDCCL
000112                  CMDDEF  COM*PILE,CMDCCL
000116                  CMDDEF  LIN*K,CMDCCL
000122                  CMDDEF  EX*ECUTE,CMDCCL
000126                  CMDDEF  COB*OL,CMDCCL
000132                  CMDDEF  FOR*TRAN,CMDCCL
000136                  CMDDEF  FORM,CMDFRM
000142                  CMDDEF  FORMA*T,CMDFMT
000146                  CMDDEF  DIB*OL,CMDCCL
000152                  CMDDEF  MA*CRO,CMDCCL
000156                  CMDDEF  MAK*E,CMDCCL
000162                  CMDDEF  DA*TE,CMDDAT
000166                  CMDDEF  TI*ME,CMDTIM
000172                  CMDDEF  MO*UNT,CMDMNT
000176                  CMDDEF  MON*ITOR,CMDMON
000202                  CMDDEF  DIS*MOUNT,CMDDMT
000206                  CMDDEF  DISP*LAY,CMDDSP
000212                  CMDDEF  DEF*INE,CMDDEF
000216                  CMDDEF  OF*F,CMDOFF
000222                  CMDDEF  LOG*OFF,CMDOFF
000226                  CMDDEF  KJ*OB,CMDOFF
000232                  CMDDEF  BY*E,CMDOFF
000236                  CMDDEF  AS*SIGN,CMDASN
000242                  CMDDEF  DEA,CMDDSN
000246                  CMDDEF  DEAS*SIGN,CMDDSN
000252                  CMDDEF  ALLO*CATE,CMDALC
000256                  CMDDEF  DEAL*LOCATE,CMDDLC
000262                  CMDDEF  IND,CMDIND
000266                  CMDDEF  DIF*FERENCES,CMDCCL
000272                  CMDDEF  DU*MP,CMDCCL
000276                  CMDDEF  H*ELP,CMDCCL
000302                  CMDDEF  ME*MORY,CMDMEM
```

```
     58 000306                          CMDDEF   BA*CKUP, CMDCCL
     59 000312                          CMDDEF   PRO*TECT, CMDCCL
     60 000316                          CMDDEF   UNP*ROTECT, CMDCCL
     61 000322                          CMDDEF   OP*ERATOR, OPRCMD
     62 000326                          CMDDEF   SP*OOL, CMDSPO
     63 000332                          CMDDEF   CR*EATE, CMDCCL
     64 000336                          CMDDEF   KI*LL, CMDKIL
     65 000342                          CMDDEF   PA*USE, CMDPAU
     66 000346                          CMDDEF   REC*ALL, CMDRCL
     67 000352                          CMDDEF   DET*ACH, CMDDET
     68 000356                          CMDDEF   AC*CESS, CMDACC
     69 000362                          CMDDEF   NOAC*CESS, CMDNAC
     70 000366                          CMDDEF   US*E, CMDUSE
     71 000372                          CMDDEF   INI*TIALIZE, CMDINI
     72 000376                          CMDDEF   SQ*UEEZE, CMDSQZ
     73 000402                          CMDDEF   LIB*RARY, CMDCCL
     74 000406                          CMDDEF   TE*CO, CMDCCL
     75 000412                          CMDDEF   UC*L, UCLCMD
     76 000416                          CMDDEF   MU*NG, CMDCCL
     77 000422                          CMDDEF   CL*OSE, RDCMD
     78 000426                          CMDDEF   GT, RDCMD
     79 000432                          CMDDEF   INS*TALL, CMDINS
     80 000436                          CMDDEF   LO*AD, RDCMD
     81 000442                          CMDDEF   SU*SPEND, CMDSPN
     82 000446                          CMDDEF   UNL*OAD, RDCMD
     83 000452                          CMDDEF   BO*OT, CMDBOT
     84 000456                          CMDDEF   $ST*OP, CMDBOT
     85 000462                          CMDDEF   $SH*UTDOWN, CMDSHT
     86 000466                          TBLEND
```

Table of command keywords

```
      1                                     ;-------------------------------------------------------------
      2 000234  000000  000000  000000  ASDEX:   .WORD   0,0,0,0
        000242  000000
      3 000244  015270  012445          DKCOM:   .RAD50  /DK COM/
      4 000250  075250  073376          SYSAV:   .RAD50  /SY SAV/
      5 000254  015270  073376          DKSAV:   .RAD50  /DK SAV/
      6 000260  100040                  R50TT:   .RAD50  /TT /
      7 000262  015270                  R50DK:   .RAD50  /DK /
      8 000264  075250                  R50SY:   .RAD50  /SY /
      9 000266  046537                  R50LOG:  .RAD50  /LOG/
     10 000270  016003                  R50DSK:  .RAD50  /DSK/
     11 000272  042614                  R50KED:  .RAD50  /KED/
     12 000274  045130                  R50K52:  .RAD50  /K52/
     13 000276  042640                  R50KEX:  .RAD50  /KEX/
     14 000300  062570                  R50PIP:  .RAD50  /PIP/
     15 000302  015172                  R50DIR:  .RAD50  /DIR/
     16 000304  016130                  R50DUP:  .RAD50  /DUP/
     17 000306  035164                  R50IND:  .RAD50  /IND/
     18 000310  100020                  R50TSX:  .RAD50  /TSX/
     19 000312  101704                  R50UCL:  .RAD50  /UCL/
     20 000314  105372                  R50VIR:  .RAD50  /VIR/
     21 000316  045640                  R50LD:   .RAD50  /LD /
     22 000320  045676                  R50LDO:  .RAD50  /LDO/
     23 000322  045705                  R50LD7:  .RAD50  /LD7/
     24 000324  012445                  R50COM:  .RAD50  /COM/
     25 000326  073376                  R50SAV:  .RAD50  /SAV/
     26 000330  004056                  ALDEX:   .RAD50  /ALN/
     27 000332  015270  004051  026760  ALDBLK:  .RAD50  /DK ALIGN ALN/
        000340  004056
     28 000342  075250  003273  021113  AUTHFN:  .RAD50  /SY ACCESSTSX/
        000350  100020
     29 000352  075250  100003  051646  KMNNAM:  .RAD50  /SY TSKMONSAV/
        000360  073376
     30 000362  075250  011504  000000  CCLNAM:  .RAD50  /SY CCL    SAV/
        000370  073376
     31 000372  075250  035164  000000  INDNAM:  .RAD50  /SY IND    SAV/
        000400  073376
     32 000402  075250  045640  000000  LDNAM:   .RAD50  /SY LD     SYS/
        000410  075273
     33 000412  043327  053600          R50KMN:  .RAD50  /KMON  /
     34 000416  075250  114730  000000  HNBUF:   .RAD50  /SY XXX    TSX/
        000424  100020
     35 000426  054730                  R50NO:   .RAD50  /NO /
     36                                 ;
     37 000430  000000                  DEVUNT:  .WORD   0
     38 000432  000000                  ASKLNM:  .WORD   0
     39 000434  000000                  INDRFL:  .WORD   0
     40 000436  000000                  QUOTFL:  .WORD   0
     41 000440                          FILNAM:  .BLKW   5
     42 000452                          XAREA:   .BLKW   8.
     43 000472                          R50BUF:  .BLKW   2
     44 000476  000000                  CPUAH:   .WORD   0
     45 000500  000000                  CPUAL:   .WORD   0
     46 000502  000000                  HANIDX:  .WORD   0
     47 000504  000000  000000  000000  MNTDEV:  .WORD   0,0,0,0
        000512  000000
     48 000514  000000                  CMDEND:  .WORD   0       ;Pointer past end of current command string
```

```
     49 000516  000000  000000  000000  ALCDEV: .WORD   0,0,0,0 ;Name of dev being allocated or deallocated
        000524  000000
```

```
                                        ;
                                        ;  Emt arg block to run a program.
                                        ;
000526      000     126         RUNEMT: .BYTE   0,126
000530   000000                         .WORD   0
                                        ;
                                        ;  Emt arg block to start a spooler.
                                        ;
000532      000     126         SPGEMT: .BYTE   0,126
000534   000001                         .WORD   1
000536   000000                         .WORD   0
                                        ;
                                        ;  Emt arg block to control detached jobs.
                                        ;
000540      000     132         DETARG: .BYTE   0,132
000542   000000                         .WORD   0
                                        ;
                                        ;  Emt argument block to set the size of the job
                                        ;
000544      000     141         SIZEMT: .BYTE   0,141
000546   000000                 SIZVAL: .WORD   0
                                        ;
                                        ;  Emt arg block to mount a file structure.
                                        ;
000550      000     134         MNTARG: .BYTE   0,134
000552   000504'                        .WORD   MNTDEV
                                        ;
                                        ;  Emt arg block to dismount a file structure.
                                        ;
000554      000     135         DMTARG: .BYTE   0,135
000556   000504'                        .WORD   MNTDEV
                                        ;
                                        ;  Emt arg block to copy the file status and privileges from a parent job
                                        ;
000560      001     160         CPYCXT: .BYTE   1,160
000562   000000                         .WORD   0
                                        ;
                                        ;  Emt arg block to send a message to a line.
                                        ;
000564      000     127         YELEMT: .BYTE   0,127
000566   000000                         .WORD   0
000570   000572'                        .WORD   MSGBUF
000572                          MSGBUF: .BLKB   84.
000716                          MSGEND:
                                        ;
                                        ;  Emt to allocate a device
                                        ;
000716      000     156         ALDEMT: .BYTE   0,156
000720   000516'                        .WORD   ALCDEV          ;Pointer to device spec
                                        ;
                                        ;  Emt to deallocate a device
                                        ;
000722      001     156         DLCEMT: .BYTE   1,156
000724   000516'                        .WORD   ALCDEV
                                        ;
                                        ;  Emt to determine if a device is allocated to another user
                                        ;
```

```
                                TALEMT: .BYTE   2,156
                                        .WORD   ALCDEV
                                ;
                                ;  Emt argument block to read a cached device descriptor block into CDBUF
                                ;
                                CDGEMT: .BYTE   0,126
                                        .WORD   10              ;Subfunction code (peek)
                                CDGADR: .WORD   0               ;Address of block within kernel
                                        .WORD   CD$$SZ          ;Amt of data to get
                                        .WORD   CDBUF           ;Destination buffer
                                ;
                                ;  Emt argument block to copy a cached device descriptor block from CDBUF
                                ;  back into the kernel data area.
                                ;
                                CDPEMT: .BYTE   0,126
                                        .WORD   11              ;Subfunction code (poke)
                                CDPADR: .WORD   0               ;Address of block within kernel
                                        .WORD   CD$$SZ
                                        .WORD   CDBUF           ;Destination buffer
                                ;
                                ;  Emt argument block to move INSTALL entry from kernel into IIBUF
                                ;
                                INGEMT: .BYTE   0,126
                                        .WORD   10              ;Subfunction code (peek)
                                INGADR: .WORD   0               ;Address of block within kernel
                                        .WORD   II$$SZ          ;Amt of data to get
                                        .WORD   IIBUF           ;Destination buffer
                                ;
                                ;  Emt argument block to store INSTALL entry into kernel data
                                ;
                                INPEMT: .BYTE   0,126
                                        .WORD   11              ;Subfunction code (poke)
                                INPADR: .WORD   0               ;Address of block within kernel
                                        .WORD   II$$SZ          ;Amt of data to move
                                        .WORD   IIBUF           ;Source buffer
                                ;
                                ;  Emt arg block to log off the current job.
                                ;
                                OFFEMT: .BYTE   0,126
                                        .WORD   2
                                        .WORD   0               ;0==>Default time to drop DTR
                                ;
                                ;  Emt to update running copy of a handler.
                                ;
                                HUPARG: .BYTE   0,126
                                        .WORD   3
                                        .BLKW   2
                                ;
                                ;  Emt to reboot the system.
                                ;
                                BOTEMT: .BYTE   0,126
                                        .WORD   4
                                        .WORD   START
                                ;
                                ;  Emt to force logoff of a job.
                                ;
                                KILEMT: .BYTE   0,126
```

```
                                        .WORD   5
                                        .WORD   0
                                ;
                                ;  Emt to suspend execution of a job
                                ;
001034      000     126         SJEMT:  .BYTE   0,126
001036  000012                          .WORD   12
001040  000000                          .WORD   0
                                ;
                                ;  Emt to resume execution of a job
                                ;
001042      001     126         RJEMT:  .BYTE   1,126
001044  000012                          .WORD   12
001046  000000                          .WORD   0
                                ;
                                ;  EMT arg block to set "hold" mode for a spooled log file
                                ;
001050      0000    151         SPLHLA: .BYTE   LOGCHN,151
001052  000000                          .WORD   0
001054  000001                          .WORD   1
                                ;
                                ;  EMT arg block to copy data from the context block of another job
                                ;  into our context block.
                                ;
001056      000     126         EMCXCP: .BYTE   0,126
001060  000014                          .WORD   14
001062  000000                          .WORD   0               ;# of job we are copying from
001064  000000                          .WORD   0               ;Address of item being copied
001066  000000                          .WORD   0               ;Number of bytes to copy
                                ;
                                ;  EMT arg block to set the job execution priority
                                ;
001070      000     150         PRIEMT: .BYTE   0,150
001072  000000                          .WORD   0
                                ;
                                ;  EMT arg block to create a window for this job
                                ;
001074      000     161         MAKWIN: .BYTE   0,161
001076      001     001                 .BYTE   1,1
001100      120     020                 .BYTE   80.,16.
001102      001     000                 .BYTE   1,0
001104  000000                          .WORD   0
                                ;
                                ;  EMT arg block to select window 1 as the current window
                                ;
001106      001     161         MAPWIN: .BYTE   1,161
001110      001     000                 .BYTE   1,0
                                ;
                                ;  EMT arg block to delete all temporary display windows for the job
                                ;
001112      002     161         DELWIN: .BYTE   2,161
001114      000     000                 .BYTE   0,0
                                ;
                                ;  Emt arg block to initiate performance monitoring.
                                ;
001116      000     136         MONAR1: .BYTE   0,136
001120  000000                  MNBASE: .WORD   0
```

```
    172 001122  000000                MNTOP:  .WORD   0
    173 001124  000000                MNBPC:  .WORD   0
    174 001126  000000                MNFLGS: .WORD   0
    175                               ;
    176                               ;  Emt arg block to start performance monitoring.
    177                               ;
    178 001130      001     136       MONAR2: .BYTE   1,136
    179                               ;
    180                               ;  Emt arg block to broadcast job status change info to monitoring jobs
    181                               ;
    182 001132      002     157       GENMON: .BYTE   2,157
    183 001134  000000                        .WORD   0               ;Store value to broadcase here
```

Table of command keywords

```
      1 001136                          CMDBUF:  .BLKW   66.
      2 001342                          CBFEND:                          ;END OF CMDBUF
      3 001342                          KEYBUF:  .BLKW   8.              ;Used by SEARCH to hold keywords
      4 001362                          KEYEND:                          ;End of KEYBUF
      5 001362                          BLK0:    .BLKW   512.
      6          001362'                KCSIBF   =        BLK0           ;OVERLAY WITH BLK0
      7          001362'                INBUF    =        BLK0           ;OVERLAY WITH BLK0
      8 003362                          CDBUF:   .BLKW   10.             ;Should be at least as large as CD$$SZ
      9                                 ;  TSKMON STACK
     10 003406                                   .BLKW   100.
     11          003716'                KMSTK    =        .
     12                                 ;
     13                                 ;  BYTE DATA
     14                                 ;
     15                                 ;  PRINT BUFFER
     16 003716                          PRTBUF:  .BLKB   256.
     17 004316      200                 PBUFND:  .BYTE   200
     18 004317      000                 NOFLAG:  .BYTE   0
     19 004320      000                 DOLRAT:  .BYTE   0               ;INDICATES $@ SCANNED
     20 004321      000                 DOTAT:   .BYTE   0               ;INDICATES #@ SCANNED
     21 004322      000                 COLEQL:  .BYTE   0               ;INDICATES := SCANNED
     22 004323      000                 NOUCL:   .BYTE   0               ;Non-zero==>Don't call UCL for this command
     23 004324      123      131   072  SYTXT:   .ASCIZ  /SY:/
        004327      000
     24 004330      123      131   072  SYINTX:  .ASCIZ  /SY:IND /
        004333      111      116   104
        004336      040      000
     25                                 ;
     26                                          .EVEN
     27                                 ;
```

```
                                        .SBTTL  Job Initialization
                                ;-----------------------------------------------
                                ;  ENTER TSKMON AT KMNSTR.
                                ;
004340  000240                  START:  NOP                     ;DEBUGGING BPT INSERTION POINT
004342  116701  000000G                 MOVB    CORUSR,R1       ;GET USER INDEX #
                                ;
                                ;  IF THE SYSTEM-WIDE ONCE-ONLY INITIALIZATION HASN'T BEEN DONE YET,
                                ;  THEN DISABLE THE JOB SCHEDULER TO ENSURE THIS JOB GETS A CHANCE TO DO IT
                                ;  BEFORE ANYBODY ELSE CAN RUN.
                                ;
004346  105767  000000G                 TSTB    KMONCE          ;Has once only initialization been done yet?
004352  001402                          BEQ     40$             ;Br if it has already been done
004354  110167  000000G                 MOVB    R1,EXCJOB       ;If not, disable scheduling of any other jobs
                                ;
                                ;  INITIALIZATION PERFORMED FOR LINE THE FIRST TIME
                                ;  TSKMON IS ENTERED.
                                ;
004360  032761  000000G 000000G 40$:    BIT     #$KINIT,LSW(R1) ;IS THIS 1ST ENTRY FOR THIS LINE?
004366  001402                          BEQ     10$             ;BR IF YES
004370  000167  001060                  JMP     KMNOT1
                                ;
                                ;  Make sure TSGEN hasn't been modified without relinking TSKMON.
                                ;
004374  026727  000000G 123456  10$:    CMP     GENTOP,#123456  ;DOES THIS LOOK LIKE THE TOP OF TSGEN?
004402  001406                          BEQ     22$             ;BR IF OK
004404                                  FERR    #REGEN          ;NEED TO RELINK TSKMON
004420  016761  000000G 000000G 22$:    MOV     MINTIM,LCONTM(R1);SET JOB LOG-ON TIME
004426  042761  000000G 000000G         BIC     #$CFKIL,LSW6(R1) ;IN CASE OF ^C^C DURING GETSYP
004434  042761  000000G 000000G         BIC     #$CTRLC,LSW(R1) ;DO NOT DO CONTROL-C ABORT
                                ;
                                ;  Initially set the job name to all blanks
                                ;
004442  010105                          MOV     R1,R5           ;GET JOB INDEX #
004444  070527  000006                  MUL     #6,R5           ;* 12. BYTES PER ENTRY
004450  062705  000000G                 ADD     #LUNAME,R5      ;POINT TO USER NAME ENTRY FOR THIS LINE
004454  012704  000014                  MOV     #12.,R4
004460  112725  000040          24$:    MOVB    #' ,(R5)+       ;SET USER NAME TO BLANKS
004464  077403                          SOB     R4,24$
                                ;
                                ;  Initialize cells in job context block
                                ;
004466  012767  000000G 000000G         MOV     #CFSTK,CFSP     ;COMMAND FILE STACK
004474  016700  000000G                 MOV     DFJMEM,R0       ;GET MAX # KB JOB IS ALLOWED TO USE
004500  072027  000012                  ASH     #10.,R0         ;CONVERT TO ADDRESS
004504  001002                          BNE     25$             ;BR IF DIDN'T OVERFLOW 64KB
004506  012700  177774                  MOV     #177774,R0      ;SET MAX MEMORY AS 64KB
004512  010067  000000G          25$:    MOV     R0,MAXMEM       ;SET AS TOP OF MEMORY FOR JOB
004516  012767  000000G 000000G         MOV     #NAMTOP,UHIMEM  ;HIGHEST LEGAL ADDRESS FOR JOB
004524  012767  000004G 000000G         MOV     #SPUBUF+4,SXBPNT;SPOOL FILE BUFFER
004532  112767  000000G 000000G         MOVB    #MAXPRI,MXJPRI  ;SET MAXIMUM JOB PRIORITY
004540  112767  000056  000000G         MOVB    #'.,KMPRMT      ;SET UP DEFAULT KMON PROMPT (".")
004546  112767  000200  000001G         MOVB    #200,KMPRMT+1   ;TERMINATE PROMPT STRING
004554  012703  000000G                 MOV     #STDNAM,R3      ;SET UP CURRENT (DEFAULT) SPOOL FORM NAME
004560  012704  000000G                 MOV     #UFORM,R4
004564  012700  000006                  MOV     #6,R0
004570  112324                  3$:     MOVB    (R3)+,(R4)+
```

```
     58 004572  077002                          SOB     R0,3$
     59 004574  112767  000000G 000000G         MOVB    #SC$ERR,ERRSEV  ;ERROR ABORT SEVERITY LEVEL
     60                                ;
     61                                ;  Start job with all privileges granted and no access restrictions
     62                                ;  Privileges may be changed by running LOGON or by use of
     63                                ;  the SET PROCESS/AUTH/PRIV=(list)  command.
     64                                ;
     65 004602  005067  000000G               CLR     RESDEV          ;Say no access restrictions
     66 004606  012767  000000G 000000G       MOV     #OKFILE,OKFAND  ;Init last access entry ptr
     67 004614  012767  000000G 000000G       MOV     #OKFEND,OKFNND  ;Init first noaccess entry ptr
     68 004622  012700  000000G               MOV     #PVNPW,R0       ;Get # words in each privilege vector
     69 004626  005002                        CLR     R2              ;Set index for 1st word
     70 004630  012703  177777               MOV     #177777,R3      ;Get all privilege flags turned on
     71 004634  010362  000000G         77$:  MOV     R3,PRIVA0(R2)   ;Authorized privileges
     72 004640  010362  000000G               MOV     R3,PRIVS0(R2)   ;Set privileges
     73 004644  010362  000000G               MOV     R3,PRIVF0(R2)   ;Command file privileges
     74 004650  010362  000000G               MOV     R3,PRIVC0(R2)   ;Current (program) privileges
     75 004654  062702  000002                ADD     #2,R2           ;Increment vector index
     76 004660  077013                        SOB     R0,77$          ;Set all privileges
     77 004662  020127  000000G               CMP     R1,#LSTPL       ;Is this a primary line?
     78 004666  003016                        BGT     32$             ;Br if not
     79 004670  032761  000000G 000000G       BIT     #$NOVLN,ILSW2(R1);Disallow subprocess usage?
     80 004676  001412                        BEQ     32$             ;Br if don't need to disallow subprocesses
     81 004700  012700  000000G               MOV     #P2$VIR,R0      ;Get subprocess privilege flag
     82 004704  040067  000000G               BIC     R0,PRIVA2       ;Remove subprocess privilege
     83 004710  040067  000000G               BIC     R0,PRIVS2
     84 004714  040067  000000G               BIC     R0,PRIVF0
     85 004720  040067  000000G               BIC     R0,PRIVC0
     86 004724  004767  000000G         32$:  CALL    RSTPRV          ;Setup some other privilege flags
     87                                ;
     88                                ;  Open spool file channel
     89                                ;
     90 004730  105767  000000G               TSTB    NSPLDV          ;ARE THERE ANY SPOOLED DEVICES?
     91 004734  001410                        BEQ     28$             ;BR IF NOT
     92 004736                                .REOPEN #XAREA,#USPLCH,#SPLCHN ;OPEN CHANNEL FOR WRITES TO SPOOL FILE
     93                                ;
     94                                ;  Initialize TSXUCL data base for this job
     95                                ;  (Say that there are no user-defined commands yet)
     96                                ;
     97 004756  016705  000000G         28$:  MOV     UCLBLK,R5       ;Get # blocks in file per job
     98 004762  001446                        BEQ     2$              ;Br if TSXUCL data file not needed
     99 004764                                .LOOKUP #XAREA,#1,#UCLDAT ;Lookup TSXUCL data file
    100 005004  103435                        BCS     2$              ;Br if not there
    101 005006  010100                        MOV     R1,R0           ;Get job index number
    102 005010  006200                        ASR     R0              ;Convert to job number
    103 005012  005300                        DEC     R0              ;Make first job # 0
    104 005014  070500                        MUL     R0,R5           ;Compute block number of data for this job
    105 005016  012702  001362'               MOV     #BLK0,R2        ;Point to data buffer
    106 005022  012762  177777  000000G       MOV     #-1,UC$NDC(R2)  ;Set flag saying TSXUCL should init data
    107 005030  016762  000000G 000000G       MOV     VUCLMC,UC$MDC(R2) ;Set maximum allowed number of commands
    108 005036                                .WRITW  #XAREA,#1,R2,#256.,R5 ;Write out data for job
    109 005072                                .CLOSE  #1              ;Close TSXUCL file
    110                               ;
    111                               ;  Set default editor
    112                               ;
    113 005100  126727  000000G 000000G 2$:   CMPB    VEDIT,#EDIT     ;IS DEFAULT EDITOR EDIT?
    114 005106  001413                        BEQ     7$              ;BR IF YES
```

```
     115 005110  126727  000000G 000000G          CMPB    VEDIT,#TECO      ;IS DEFAULT EDITOR TECO?
     116 005116  001004                           BNE     21$              ;BR IF NOT
     117 005120  052761  000000G 000000G          BIS     #$TECO,LSW5(R1)  ;SET DEFAULT EDITOR FOR JOB
     118 005126  000403                           BR      7$
     119 005130  052761  000000G 000000G 21$:     BIS     #$KED,LSW5(R1)   ;DEFAULT EDITOR MUST BE KED
     120                                 ;
     121                                 ;  Set default UCL FIRST/MIDDLE/LAST
     122                                 ;
     123 005136  126727  000000G 000001  7$:      CMPB    VUCLOR,#1        ;SET UCL FIRST?
     124 005144  001003                           BNE     30$              ;BR IF NOT
     125 005146  052761  000000G 000000G          BIS     #$UCLCF,LSW7(R1)
     126 005154  126727  000000G 000002  30$:     CMPB    VUCLOR,#2        ;SET UCL MIDDLE?
     127 005162  001003                           BNE     29$              ;BR IF NOT
     128 005164  052761  000000G 000000G          BIS     #$UCLCM,LSW7(R1)
     129 005172  126727  000000G 000003  29$:     CMPB    VUCLOR,#3        ;SET UCL LAST?
     130 005200  001003                           BNE     27$              ;BR IF NOT
     131 005202  052761  000000G 000000G          BIS     #$UCLCL,LSW7(R1)
     132                                 ;
     133                                 ;  Set default SL mode
     134                                 ;
     135 005210  052761  000000G 000000G 27$:     BIS     #$SLLET,LSW7(R1);Default SL substitution on
     136                                 ;
     137                                 ;  Set default wildcard flag
     138                                 ;
     139 005216  005727  000000G          31$:     TST     #WILDFL          ;EXPLICIT OR IMPLICIT WILDCARDS?
     140 005222  001403                           BEQ     8$               ;BR IF EXPLICIT
     141 005224  052761  000000G 000000G          BIS     #$WILD,LSW5(R1)  ;SET IMPLICIT WILDCARD FLAG
     142 005232  020127  000000G          8$:     CMP     R1,#LSTDL        ;REAL OR VIRTUAL LINE?
     143 005236  003022                           BGT     12$              ;BR IF VIRTUAL
     144 005240  020127  000000G                  CMP     R1,#LSTPL        ;REAL OR DETACHED?
     145 005244  003021                           BGT     1$               ;BR DETACHED
     146                                 ;
     147                                 ;  See if we need to get a system password (primary line only).
     148                                 ;
     149 005246  032761  000000G 000000G          BIT     #$SYSPS,LSW2(R1);Do we need to accept a system password?
     150 005254  001402                           BEQ     13$              ;Br if not
     151 005256  004767  000000G                  CALL    GETSYP           ;Accept system password
     152                                 ;
     153                                 ;  Mount SY device for this job (primary line only).
     154                                 ;
     155 005262  016767  172776  173214 13$:     MOV     R50SY,MNTDEV     ;SET SY AS DEVICE TO MOUNT
     156 005270  012700  000550'                  MOV     #MNTARG,R0       ;MOUNT SY
     157 005274  104375                           EMT     375
     158                                 ;
     159                                 ;  Terminal dependent initialization for primary lines.
     160                                 ;
     161 005276  004767  004714                   CALL    TRMINI           ;Do terminal-dependent initialization
     162 005302  000402                           BR      1$
     163                                 ;
     164                                 ;  Initialization for subprocesses only.
     165                                 ;  Print n> to show which subprocess we are switching to.
     166                                 ;
     167 005304  004767  005170           12$:     CALL    VIRINI           ;Do virtual line initialization
     168                                 ;
     169                                 ;  Do some initialization for detached jobs
     170                                 ;
     171 005310  032761  000000G 000000G 1$:      BIT     #$DETCH,LSW(R1)  ;Is this a detached job?
```

```
    172 005316  001403                              BEQ     GRTINI          ;Br if not
    173 005320  012761  000000C 000000G              MOV     #$ECHO!$LC!$DEFER,LSW2(R1) ;Init control flags
    174                                 ;
    175                                 ;  Perform code which is executed only once each time TSX is started
    176                                 ;  and Print TSX greeting message
    177                                 ;
    178 005326  105767  000000G         GRTINI: TSTB    KMONCE          ;Has once only code been done yet?
    179 005332  001402                          BEQ     3$              ;Br if it has already been done
    180 005334  004767  000000G                 CALL    KMINIT          ;If not, call once only code
    181 005340  004767  005774          3$:     CALL    PRTGRT          ;Print the TSX-Plus logon greeting
    182 005344  016161  000000G 000000G        MOV     LSW2(R1),LSW2S(R1) ;SAVE IN CASE OF CTRL-C REENTRY
    183                                 ;
    184                                 ;  If this job has a parent job, copy file context and privilege from parent
    185                                 ;
    186 005352  004767  005254                  CALL    CPYPRN          ;Copy info from parent job
    187                                 ;
    188                                 ;  Set flag saying job initialization has been done
    189                                 ;
    190 005356  052761  000000G 000000G        BIS     #$KINIT,LSW(R1) ;SAY LINE HAS BEEN INITIALIZED
    191                                 ;
    192                                 ;  See if this line has an associated start-up command file.
    193                                 ;
    194 005364  020127  000000G                 CMP     R1,#LSTDL       ;Is this a virtual line?
    195 005370  101004                          BHI     1$              ;Br if yes
    196 005372  016102  000000G                 MOV     LSUCF(R1),R2    ;IS THERE A START-UP COMMAND FILE?
    197 005376  001407                          BEQ     61$             ;BR IF NOT
    198 005400  000402                          BR      2$
    199 005402  012702  000000G         1$:     MOV     #SBPSUF,R2      ;Point to job context cell with file name
    200 005406  105712                  2$:     TSTB    (R2)            ;IS FILE NAME NULL?
    201 005410  001402                          BEQ     61$             ;BR IF YES
    202 005412  004767  006324                  CALL    SETSUF          ;Set up start-up command file for execution
    203                                 ;
    204                                 ;  Broadcast status message to monitoring jobs telling them that
    205                                 ;  this job logged on.
    206                                 ;
    207 005416  012767  000000G 173510  61$:     MOV     #JS$ON,GENMON+2 ;Set logged-on status code
    208 005424  012700  001132'                 MOV     #GENMON,R0      ;Point to EMT argument block
    209 005430  104375                          EMT     375             ;Tell monitoring jobs that we are initiated
    210 005432  005761  000000G                 TST     LPARNT(R1)      ;Do we have a parent job?
    211 005436  001406                          BEQ     KMNOT1          ;Br if not
    212 005440  012767  000000G 173466          MOV     #JS$LOG,GENMON+2;Set status code saying we have logged on
    213 005446  012700  001132'                 MOV     #GENMON,R0      ;Point to EMT argument block
    214 005452  104375                          EMT     375             ;Broadcast status code that job logged on
```

```
                                        .SBTTL  Entry to KMON
                                ;-------------------------------------------------------------------
                                ;  End of initialization code that is performed during job login.
                                ;  Begin processing that is performed each time KMON is entered.
                                ;
005454  016761  172732  000000G KMNOT1:  MOV     R50KMN,LPRG1(R1);SET "KMON" AS RUNNING PROGRAM NAME
005462  016761  172726  000000G          MOV     R50KMN+2,LPRG2(R1)
005470  005067  000000G                  CLR     UTRPAD          ;CLEAR USER TRAP CONTROL
005474  005067  000000G                  CLR     UFPTRP          ;RESET FLOATING POINT TRAP CONTROL
005500  005061  000000G                  CLR     LSCCA(R1)       ;RESET .SCCA TRAP CONTROL
005504  105067  000000G                  CLRB    SERFLG          ;DO .HERR
005510  105067  000000G                  CLRB    RUNARG          ;No argument string from RUN command
005514  005767  000000G                  TST     UCHAN           ;DID USER DO A .CDFN?
005520  001404                          BEQ     4$              ;BR IF NOT
005522  005067  000000G                  CLR     UCHAN           ;UNDO THE .CDFN
005526  004767  000000G                  CALL    PRGALL          ;PURGE ALL CHANNELS
005532  005061  000000G          4$:     CLR     LTSCMD(R1)      ;NO PENDING SPECIAL COMMAND
005536  042761  000000G 000000G          BIC     #$SETRN,LSW9(R1);Say SETUP is no longer running
005544                                  .CLOSE  #RUNCHN         ;CLOSE PROGRAM SAV FILE
005552  004767  000000G                  CALL    RSTPRV          ;Reset job privileges
005556  032761  000000G 000000G          BIT     #$DUPRN,LSW6(R1);WAS DUP THE PROGRAM THAT EXITED?
005564  001407                          BEQ     7$              ;BR IF NOT
005566  042761  000000G 000000G          BIC     #$DUPRN,LSW6(R1);SAY DUP IS NO LONGER RUNNING
005574  004767  000000G                  CALL    PRGALL          ;PURGE ALL CHANNELS
005600  004767  000000G                  CALL    LDCLEN          ;DO "LD CLEAN" OPERATION
005604  032761  000000G 000000G 7$:     BIT     #$DOOFF,LSW(R1) ;SHOULD WE LOG USER OFF?
005612  001402                          BEQ     6$              ;BRANCH IF NOT
005614  000167  000000G                  JMP     CMDOFF          ;FORCE LOGOFF
                                ;  CHECK FOR SYSTEM ABORT
005620  116702  000000G          6$:     MOVB    ABRTCD,R2       ;WAS USER ABORTED?
005624  001467                          BEQ     NOABRT          ;BRANCH IF NOT
                                ;
                                ;  USER WAS ABORTED -- PUT OUT ABORT MESSAGE.
                                ;
005626  112767  000010  000000G         MOVB    #10,UERSEV      ;SAY ERROR SEVERITY LEVEL = SEVERE
005634  006302                          ASL     R2              ;CVT ERROR CODE TO WORD INDEX
005636  016203  000176'                 MOV     ABTMSG(R2),R3   ;GET ADDR OF ERROR MESSAGE
005642                                  .PRINT  #MONTXT
005650                                  .PRINT  R3
                                ;  Print name of file that caused the error
005654  005767  000000G                  TST     ERRSPC          ;Do we have a file spec to print?
005660  001414                          BEQ     1$              ;Br if not
005662                                  .PRINT  #FILERM         ;Print heading message
005670  012703  001362'                 MOV     #BLKO,R3        ;Point to buffer where result is to be stored
005674  012704  000000G                  MOV     #ERRSPC,R4      ;Point to RAD50 file spec
005700  004767  000000G                  CALL    EDTFIL          ;Convert file spec to ascii
005704                                  .PRINT  #BLKO           ;Print the file spec
                                ;  PUT OUT LOCATION OF ABORTED INSTRUCTION.
005712                          1$:     .PRINT  #LOCMSG
005720  016702  000000G                  MOV     ABRTAD,R2       ;GET ADDR OF ABORT INST
005724  004767  000000G                  CALL    OCTPRT          ;PRINT THE OCTAL VALUE
005730  020227  120000                  CMP     R2,#120000      ;IN OVERLAY REGION?
005734  000407                          BR      10$             ;*** skip overlay this version ***
                                ;       BLOS    10$             ;BR IF NOT
005736                                  .PRINT  #OVLYEQ         ;PRINT "  OVERLAY = "
005744  016700  000000G                  MOV     ABRTOV,R0       ;GET OVERLAY NAME
005750  004767  000000G                  CALL    PRTR50          ;AND DISPLAY IT
```

Entry to KMON

```
     58 005754                                10$:     .PRINT  #CRLF
     59 005762  005067  000000G                       CLR     ABRTAD          ;CLEAN ABORT ADDR
     60 005766  105067  000000G                       CLRB    ABRTCD          ;AND CODE
     61 005772  042737  004440  000000G               BIC     #004440,@#JSWLOC;RELEASE CHAIN AREA, PASLIN, AND SCHAIN
     62 006000  105067  000000G                       CLRB    CINFLG          ;KILL ANY CHAIN REQUEST
```

```
                                        ;
                                        ;  User did not abort.  See if this is a .CHAIN request
                                        ;
006004  005067  000000G         NOABRT: CLR     ERRSPC          ;CLEAR ANY ERROR FILE SPEC
006010  032761  000000C 000000G         BIT     #$CFKIL!$CFABT,LSW6(R1) ;REQUEST TO ABORT ALL COM FILES?
006016  001405                         BEQ     1$              ;BR IF NOT
006020  105067  000000G                CLRB    CINFLG          ;CLEAR ANY CHAIN REQUEST
006024  042737  000000C 000000G         BIC     #PASLIN!SCHAIN,@#JSWLOC ;KILL ANY PASSED COMMAND
006032  105767  000000G         1$:     TSTB    CINFLG          ;IS THIS .CHAIN REQUEST?
006036  001402                         BEQ     NOCIN           ;BRANCH IF NOT
006040  000167  000000G                JMP     KDOCIN          ;GO DO THE CHAIN
                                        ;
                                        ;  Not .CHAIN request so enter normal KMON.
                                        ;
                                        ;  Delete any temporary display windows for this job
                                        ;
006044  012700  001112'         NOCIN:  MOV     #DELWIN,R0      ;Point to EMT argument block
006050  104375                         EMT     375             ;Delete all temporary windows for job
                                        ;
                                        ;  Clean up various cells
                                        ;
006052  012761  000040  000000G         MOV     #BLANK,LRBFIL(R1) ;USE SPACE FOR RUBOUT FOR KMON
006060  016161  000000G 000000G         MOV     LSW2S(R1),LSW2(R1);RESET STATUS OF LSW2
006066  042761  000000G 000000G         BIC     #KL4CLR,LSW4(R1)
006074  042761  000000G 000000G         BIC     #$VTESC,LSW5(R1);TURN OFF VT52 ESC-LETTER ACTIVATION
006102  042761  000000C 000000G         BIC     #$DEBUG!$RNIOP,LSW9(R1);Not debugger, I/O page
006110  005061  000000G                CLR     LNSPAC(R1)      ;NO SPECIAL ACTIVATION CHARS
006114  005061  000000G                CLR     LRDTIM(R1)      ;CLEAR TT READ TIME-OUT VALUE
006120  005061  000000G                CLR     LAFSIZ(R1)      ;CLEAR FIELD WIDTH ACTIVATION
006124  005061  000000G                CLR     LFWLIM(R1)      ;CLEAR FIELD WIDTH LIMIT
006130  042761  000000G 000000G         BIC     #KL3CLR,LSW3(R1);CLEAR MISC BITS IN LSW3
006136  042761  000000G 000000G         BIC     #$CFDCC,LSW4(R1);CLEAR DEFERRED-CTRL-C FLAG
006144  042761  000000C 000000G         BIC     #<$NOWTT!$CHACT>,LSW5(R1);TURN OFF NO WAIT AND SINGLE CHAR ACT
006152  042761  000000G 000000G         BIC     #$NOINT,LSW7(R1);Reset non-interactive run switch
006160  042761  000000G 000000G         BIC     #$NTGCC,LSW9(R1);Clear non-terminating .GTLIN ctrl-C flag
006166  032761  000000G 000000G         BIT     #$SNWTT,LSW6(R1);DID USER DO "SET TT NOWAIT"?
006174  001403                         BEQ     3$              ;BR IF NOT
006176  052761  000000G 000000G         BIS     #$NOWTT,LSW5(R1);SET NO-WAIT FLAG
006204  016700  000000G         3$:     MOV     CFSPND,R0       ;WAS A COMMAND FILE SUSPENDED?
006210  001404                         BEQ     4$              ;BR IF NOT
006212  004767  000000G                CALL    CFSTRT          ;Restart it
006216  005067  000000G                CLR     CFSPND
                                        ;
                                        ;  Check if program specified an error severity level.
                                        ;
006222  156767  000000G 000000G 4$:     BISB    UERSEV,INDERR   ;SAVE ERROR STATUS FOR IND
006230  126767  000000G 000000G         CMPB    UERSEV,ERRSEV   ;DID PROGRAM SPECIFY ERROR SEVERITY?
006236  103413                         BLO     1$              ;BR IF OK
006240  004767  000000G                CALL    ABRTCF          ;SEVER ERROR -- ABORT COMMAND FILE
006244  042761  000000G 000000G         BIC     #$CCLRN,LSW5(R1);SAY CCL NOT RUNNING
006252  032761  000000G 000000G         BIT     #$INDAB,LSW7(R1);DOES HE WANT TO ABORT IND COMMAND FILES?
006260  001402                         BEQ     1$              ;BR IF NOT
006262  004767  000000G                CALL    INDABT          ;Abort IND execution
006266  105067  000000G         1$:     CLRB    UERSEV          ;CLEAR ERROR STATUS
                                        ;
                                        ;  Purge all of user's channels
                                        ;
```

Entry to KMON

```
      58 006272  004767  000000G                    CALL    PRGALL              ;PURGE ALL CHANNELS
      59 006276  032761  000000G 000000G             BIT     #$CFKIL,LSW6(R1);Should we abort IND and command files?
      60 006304  001403                             BEQ     2$                  ;Br if not
      61 006306  004767  000000G                    CALL    INDABT              ;Abort IND and nested command files
      62 006312  000406                             BR      5$
      63 006314  032761  000000G 000000G 2$:         BIT     #$CFABT,LSW6(R1);SHOULD WE ABORT ALL ACTIVE COMMAND FILES?
      64 006322  001405                             BEQ     6$                  ;BR IF NOT
      65 006324  004767  000000G                    CALL    ABRTCF              ;ABORT ALL ACTIVE COMMAND FILES
      66 006330  042761  000000G 000000G 5$:         BIC     #$CCLRN,LSW5(R1);STOP EXECUTION OF CCL
      67                                        ;
      68                                        ;  Tell any jobs that are monitoring us that we just entered TSKMON
      69                                        ;
      70 006336  005761  000000G                 6$:    TST     LMONHD(R1)          ;Are we being monitored?
      71 006342  001003                             BNE     7$                  ;Br if yes
      72 006344  005767  000000G                    TST     SMONHD              ;Anyone monitoring all jobs?
      73 006350  001406                             BEQ     CKPASL              ;Br if not
      74 006352  012700  001132'                 7$:    MOV     #GENMON,R0          ;Point to EMT argument block
      75 006356  012760  000000G 000002            MOV     #JS$KMN,2(R0)       ;Set status code
      76 006364  104375                             EMT     375                 ;Broadcast status message
```

```
      1                                 ;
      2                                 ;  See if program passed a set of command lines to Kmon
      3                                 ;  when it exited.
      4                                 ;
      5 006366  032737  000000C 000000G CKPASL: BIT     #PASLIN!SCHAIN,@#JSWLOC;DID PROGRAM PASS US A COMMAND?
      6 006374  001546                          BEQ     4$              ;BR IF NOT
      7                                 ;
      8                                 ;  Program did pass a set of commands to Kmon.
      9                                 ;  Set it up to look like a fake command file.
     10                                 ;  That is, the commands are stored in the command file buffer so that
     11                                 ;  they are read as if they came from a command file, but no actual
     12                                 ;  file is open.
     13                                 ;
     14                                 ;  Determine if we should abort the currently open command file.
     15                                 ;
     16 006376  032761  000000C 000000G          BIT     #$CCLRN!$INDRN,LSW5(R1);IS CCL OR IND RUNNING?
     17 006404  001006                           BNE     5$              ;BR IF YES
     18 006406  032737  000000G 000000G          BIT     #SCHAIN,@#JSWLOC;SHOULD BE ABORT CURRENT COMMAND FILE?
     19 006414  001002                           BNE     5$              ;BR IF NOT
     20 006416  004767  000000G                  CALL    ABRTCF          ;ABORT ALL CURRENTLY OPEN COMMAND FILES
     21 006422  005767  000010G          5$:     TST     CINDAT+10       ;DID HE PASS US A NULL COMMAND FILE?
     22 006426  001531                           BEQ     4$              ;IF YES THEN WE ARE FINISHED
     23                                 ;
     24                                 ;  If command line is being passed to us by the TSXUCL program,
     25                                 ;  check to see if this is a case where TSXUCL could not recognize
     26                                 ;  the command and is passing it back to us for processing.
     27                                 ;  If so, just move the command to the command buffer and then proceed
     28                                 ;  with normal KMON command checking.
     29                                 ;
     30 006430  032761  000000G 000000G          BIT     #$UCLRN,LSW7(R1);Is TSXUCL program running?
     31 006436  001432                           BEQ     15$             ;Br if not
     32 006440  126727  000012G 000077          CMPB    CINDAT+12,#'?   ;Is TSXUCL throwing command back to us?
     33 006446  001026                           BNE     15$             ;Br if not
     34 006450  012703  000013G                  MOV     #CINDAT+13,R3   ;Point to chain data area (past "?" char)
     35 006454  012702  001136'                  MOV     #CMDBUF,R2      ;Point to command buffer
     36 006460  112322                  16$:     MOVB    (R3)+,(R2)+     ;Move command to command buffer
     37 006462  001376                           BNE     16$             ;Loop till asciz null moved
     38 006464  005302                           DEC     R2              ;Make R2 point to null at end of command
     39 006466  010267  172022                   MOV     R2,CMDEND       ;Save pointer to end of command
     40 006472  042761  000000G 000000G          BIC     #$UCLRN,LSW7(R1);Say TSXUCL program no longer running
     41 006500  042737  000000C 000000G          BIC     #PASLIN!SCHAIN,@#JSWLOC ;Clear command-passed flag
     42 006506  052737  000000G 000000G          BIS     #LCBIT,@#JSWLOC ;Enable lower-case input
     43 006514  105267  175603                   INCB    NOUCL           ;Set flag saying not to call TSXUCL again
     44 006520  000167  001762                   JMP     IDNCMD          ;Go process the command
     45                                 ;
     46                                 ;  If we have pending commands in the command file buffer, compress
     47                                 ;  them to make room for new commands.
     48                                 ;  If input is coming from a real command file we do not need to
     49                                 ;  compress since we will reread buffer when we hit end of new commands.
     50                                 ;
     51 006524  032761  000000G 000000G 15$:     BIT     #$CFOPN,LSW4(R1);IS A COMMAND FILE OPEN?
     52 006532  001403                           BEQ     11$             ;BR IF NOT
     53 006534  012705  001000G                  MOV     #CFBUF+512.,R5  ;SAY ENTIRE COMMAND FILE BUFFER IS FREE
     54 006540  000403                           BR      12$
     55 006542  004767  000000G          11$:    CALL    CFSQEZ          ;COMPRESS INFO IN CURRENT COMMAND FILE BUFFER
     56 006546  010005                           MOV     R0,R5           ;SAVE ADDRESS OF END OF FREE SPACE IN BUFFER
     57 006550  004767  000000G          12$:    CALL    PUSHCF          ;OPEN A NEW COMMAND FILE (PUSH CURRENT ONE)
```

```
                                        ;
                                        ;  Move command line from chain area to command file buffer
                                        ;
006554  012703  000010G                 MOV     #CINDAT+10,R3   ;POINT TO CELL WITH CHAR COUNT
006560  012302                          MOV     (R3)+,R2        ;GET COUNT OF # CHARS IN COMMAND
006562  020227  000270                  CMP     R2,#<1000-510>  ;CHECK LENGTH OF COMMAND STRING
006566  003407                          BLE     6$              ;BR IF COMMAND IS SMALL ENOUGH
006570  042737  000000C 000000G 10$:    BIC     #PASLIN!SCHAIN,@#JSWLOC ;CLEAR FLAGS BEFORE ABORT
006576                                  FABORT  #TOOLNG         ;LINE TOO LONG ERROR MESSAGE
006606  012704  000000G           6$:    MOV     #CFBUF,R4       ;POINT TO START OF COMMAND FILE BUFFER
006612  112324                    2$:    MOVB    (R3)+,(R4)+     ;MOVE COMMAND TO BUFFER
006614  001005                          BNE     1$              ;BR IF NORMAL CHARACTER
                                        ;  Put cr-lf in place of asciz nulls
006616  112764  000015  177777          MOVB    #CR,-1(R4)
006624  112724  000012                  MOVB    #LF,(R4)+
006630  077210                    1$:    SOB     R2,2$           ;MOVE ALL CHARACTERS
006632  020405                          CMP     R4,R5           ;DID WE OVERFLOW BUFFER SPACE?
006634  101355                          BHI     10$             ;BR IF YES -- COMMAND TOO LONG
                                        ;
                                        ;  Null fill remainder of buffer
                                        ;
006636  020405                    14$:  CMP     R4,R5           ;HAVE WE REACHED THE END OF THE BUFFER?
006640  103002                          BHIS    13$             ;BR IF YES
006642  105024                          CLRB    (R4)+           ;NULL FILL REST OF BUFFER
006644  000774                          BR      14$
                                        ;
                                        ;  If special chain exit is being used, or if command is coming
                                        ;  from CCL or IND, then don't list the commands.
                                        ;
006646  032761  000000C 000000G 13$:    BIT     #$CCLRN!$INDRN,LSW5(R1) ;IS CCL OR IND RUNNING?
006654  001004                          BNE     8$              ;BR IF YES
006656  032737  000000G 000000G         BIT     #SCHAIN,@#JSWLOC;IS THIS A SPECIAL CHAIN EXIT?
006664  001403                          BEQ     9$              ;BR IF NOT
006666  052761  000000G 000000G 8$:     BIS     #$QUIET,LSW4(R1);SET FLAG TO SUPPRESS LISTING COMMAND LINES
006674  032761  000000G 000000G 9$:     BIT     #$CCLRN,LSW5(R1);IS THIS AN EXPANDED CCL COMMAND?
006702  001403                          BEQ     4$              ;BR IF NOT
006704  052761  000000G 000000G         BIS     #$CFCCL,LSW4(R1);REMEMBER THAT THIS IS A CCL COMMAND
                                        ;  Clear flags which say CCL or IND is running.
006712  042761  000000C 000000G 4$:     BIC     #$CCLRN!$INDRN,LSW5(R1);CCL AND IND ARE NO LONGER RUNNING
006720  042737  000000C 000000G         BIC     #PASLIN!SCHAIN,@#JSWLOC ;CLEAR COMMAND-PASSED FLAGS
006726  042761  000000C 000000G         BIC     #$UCLRN!$UKMRN,LSW7(R1);SAY TSXUCL IS NO LONGER RUNNING
                                        ;
                                        ;  See if we are exiting from a locked program
                                        ;
006734  032761  000000G 000000G CKLK:    BIT     #$PRGLK,LSW5(R1);IS A LOCKED PROGRAM EXITING?
006742  001402                          BEQ     CKSF2           ;BR IF NOT
006744  000167  000000G                 JMP     CMDOFF          ;EXIT FROM LOCKED PROGRAM==>LOGOFF
                                        ;
                                        ;  See if we need to start execution of a secondary start-up command file
                                        ;  (The secondary start-up command file runs without privilege after the
                                        ;  initial start-up command file finishes).
                                        ;
006750  105767  000000G           CKSF2: TSTB    SUCF2           ;Is there a pending secondary command file?
006754  001417                          BEQ     NEWCMD          ;Br if not
006756  005767  000000G                 TST     CFPNT           ;Are we currently in another command file?
006762  001014                          BNE     NEWCMD          ;Br if yes -- Wait for it to finish
006764  042761  000000G 000000G         BIC     #$SUCF,LSW9(R1) ;Say we are finished with 1st startup file
```

Entry to KMON

```
    115 006772  042761  000000G 000000G         BIC     #$NOIN,LSW3(R1) ;Allow input to be accepted for line
    116 007000  012702  000000G                 MOV     #SUCF2,R2       ;Point to name of secondary command file
    117 007004  004767  004732                  CALL    SETSUF          ;Set up command file
    118 007010  105067  000000G                 CLRB    SUCF2           ;Say secondary file no longer pending
```

```
                                        .SBTTL  Get keyboard command
                                ;
                                ;  Print CR-LF to get to left margin if we are not already there.
                                ;
007014  052737  000000G 000000G NEWCMD: BIS     #LCBIT,@#JSWLOC ;ENABLE LOWER-CASE INPUT
007022  116701  000000G                MOVB    CORUSR,R1       ;GET JOB INDEX NUMBER
007026  105761  000000G                TSTB    LCOL(R1)        ;ARE WE ALREADY AT LEFT MARGIN?
007032  001403                         BEQ     RDCMD           ;BR IF YES -- NO CR-LF NEEDED
007034                                 .PRINT  #CRLF           ;PRINT CR-LF
                                ;
                                ;  See if IND is in control and we need to call it to get the next command
                                ;
007042  116701  000000G        RDCMD:  MOVB    CORUSR,R1       ;GET JOB INDEX NUMBER
007046  012702  000000G                MOV     #INDSTA,R2      ;GET POINTER TO IND STATUS BYTE
007052  132712  000000G                BITB    #IN$ACT,@R2     ;IS IND ACTIVE?
007056  001406                         BEQ     2$              ;BR IF NOT
007060  142712  000000C                BICB    #IN$ACT!IN$CMD,@R2 ;CLEAR STATUS FLAGS FOR IND
007064  152712  000000G                BISB    #IN$CNT,@R2     ;SAY WE ARE CONTINUING EXECUTION OF IND
007070  000167  002476                 JMP     INDRUN          ;GO RUN IND PROGRAM
                                ;
                                ;  If user-written command interface program is active, call it to
                                ;  get the next command.
                                ;
007074  032761  000000G 000000G 2$:     BIT     #$UKMON,LSW7(R1);Are we to use user-written command program?
007102  001405                         BEQ     3$              ;Br if not
007104  005767  000000G                TST     CFPNT           ;Are we getting commands from a command file?
007110  001002                         BNE     3$              ;Br if yes -- Don't call user program till end
007112  000167  002162                 JMP     CALUKM          ;Enter user command processor program
                                ;
                                ;  Read next command line
                                ;
007116                          3$:     .GTLIN  #INBUF,#KMPRMT  ;PROMPT FOR AND ACCEPT COMMAND LINE
```

```
                                        ;
                                        ;  Input line is now in INBUF in asciz form.
                                        ;  Move line to CMDBUF while looking for start of comments or
                                        ;  indirect file reference.
                                        ;
007136  012702  001136'         PRSCMD: MOV     #CMDBUF,R2      ;MOVE FINISHED COMMAND HERE
007142  005067  171266                  CLR     INDRFL          ;SAY NO INDIRECT FILE YET
007146  105067  171264                  CLRB    QUOTFL          ;Say we are not in quoted string
007152  105067  175142                  CLRB    DOLRAT          ;SAY HAVE NOT SEEN "$@"
007156  105067  175137                  CLRB    DOTAT           ;SAY HAVE NOT SEEN "#@"
007162  105067  175134                  CLRB    COLEQL          ;SAY HAVE NOT SEEN ":="
007166  105067  175131                  CLRB    NOUCL           ;SAY WE MAY CALL UCL FOR THIS COMMAND
007172  012704  001362'         SCNCMD: MOV     #INBUF,R4       ;SCAN FROM HERE
007176  112400                  8$:     MOVB    (R4)+,R0        ;GET NEXT CHAR FROM INPUT LINE
007200  001546                          BEQ     4$              ;BR IF END OF LINE HIT
007202  120027  000040                  CMPB    R0,#'           ;SKIP OVER LEADING SPACES
007206  001773                          BEQ     8$
007210  120027  000011                  CMPB    R0,#TAB         ;SKIP LEADING TABS
007214  001770                          BEQ     8$
007216  120027  000014                  CMPB    R0,#FF          ;SKIP LEADING FORM FEEDS
007222  001765                          BEQ     8$
007224  000402                          BR      9$              ;BEGIN SCANNING REAL COMMAND
                                        ;
                                        ;  Get next character from input line
                                        ;
007226  112400                  6$:     MOVB    (R4)+,R0        ;GET NEXT CHAR FROM INPUT LINE
007230  001530                          BEQ     11$             ;BR IF END OF LINE HIT
                                        ;
                                        ;  See if we are in a quoted string
                                        ;
007232  120067  171200          9$:     CMPB    R0,QUOTFL       ;Is this the terminating quote mark?
007236  001003                          BNE     17$             ;Br if not
007240  105067  171172                  CLRB    QUOTFL          ;Say not within quoted field now
007244  000511                          BR      5$              ;Go store terminating quote character
007246  105767  171164          17$:    TSTB    QUOTFL          ;Are we inside a quoted string now?
007252  001106                          BNE     5$              ;Br if yes -- Go store char without checking
007254  120027  000047                  CMPB    R0,#47          ;Apostrophe character?
007260  001403                          BEQ     19$             ;Br if yes
007262  120027  000042                  CMPB    R0,#42          ;Quote character?
007266  001003                          BNE     18$             ;Br if not
007270  110067  171142          19$:    MOVB    R0,QUOTFL       ;Remember we are inside a quoted string
007274  000475                          BR      5$              ;Go store character without further checking
                                        ;
                                        ;  Check for start of comments
                                        ;
007276  120027  000041          18$:    CMPB    R0,#'!          ;START OF COMMENT FIELD?
007302  001503                          BEQ     11$             ;BR IF YES
                                        ;
                                        ;  Check for :=
                                        ;
007304  105767  175012                  TSTB    COLEQL          ;Have we already seen :=?
007310  001067                          BNE     5$              ;Br if yes -- Ignore @'s after :=
007312  120027  000072                  CMPB    R0,#':          ;Start of := sequence?
007316  001010                          BNE     12$             ;Br if not
007320  110022                          MOVB    R0,(R2)+        ;Store into result string
007322  112400                          MOVB    (R4)+,R0        ;Get character following colon
007324  001472                          BEQ     11$             ;Br if end of line hit
```

Get keyboard command

```
     58 007326  120027  000075                CMPB    R0,#'=          ;Is this := ?
     59 007332  001002                        BNE     12$             ;Br if not
     60 007334  105267  174762                INCB    COLEQL          ;Remember := seen within command string
     61                                ;
     62                                ;  Check for $@
     63                                ;
     64 007340  120027  000044        12$:    CMPB    R0,#'$          ;COULD THIS BE START OF "$@"?
     65 007344  001014                        BNE     10$             ;BR IF NOT
     66 007346  005767  171062                TST     INDRFL          ;HAVE WE ALREADY SEEN @?
     67 007352  001046                        BNE     5$              ;BR IF YES
     68 007354  121427  000100                CMPB    (R4),#'@        ;IS THIS "$@"?
     69 007360  001043                        BNE     5$              ;BR IF NOT
     70 007362  010267  171046                MOV     R2,INDRFL       ;REMEMBER INDIRECT FILE NAME LOCATION
     71 007366  105267  174726                INCB    DOLRAT          ;REMEMBER PREFIX WAS "$@"
     72 007372  005204                        INC     R4              ;SKIP PAST "$"
     73 007374  000714                        BR      6$              ;START GETTING FILE NAME
     74                                ;
     75                                ;  Check for #@
     76                                ;
     77 007376  120027  000043        10$:    CMPB    R0,#'#          ;COULD THIS BE START OF "#@"?
     78 007402  001014                        BNE     3$              ;BR IF NOT
     79 007404  005767  171024                TST     INDRFL          ;HAVE WE ALREADY SEEN @?
     80 007410  001027                        BNE     5$              ;BR IF YES
     81 007412  121427  000100                CMPB    (R4),#'@        ;IS THIS "#@"?
     82 007416  001024                        BNE     5$              ;BR IF NOT
     83 007420  010267  171010                MOV     R2,INDRFL       ;REMEMBER INDIRECT FILE NAME LOCATION
     84 007424  105267  174671                INCB    DOTAT           ;REMEMBER PREFIX WAS "#@"
     85 007430  005204                        INC     R4              ;POINT PAST "."
     86 007432  000675                        BR      6$              ;START GETTING FILE NAME
     87                                ;
     88                                ;  Check for @ and @@
     89                                ;
     90 007434  120027  000100        3$:     CMPB    R0,#'@          ;START OF INDIRECT FILE REFERENCE?
     91 007440  001013                        BNE     5$              ;BR IF NOT
     92 007442  121427  000100                CMPB    (R4),#'@        ;Is this "@@"?
     93 007446  001002                        BNE     20$             ;Br if not
     94 007450  005204                        INC     R4              ;Skip past second at-sign
     95 007452  000406                        BR      5$              ;Translate "@@" to "@" and pass with command
     96 007454  005767  170754        20$:    TST     INDRFL          ;ALREADY SEEN @ BEFORE?
     97 007460  001003                        BNE     5$              ;IF YES THEN IGNORE THIS ONE FOR NOW
     98 007462  010267  170746                MOV     R2,INDRFL       ;SAVE POINTER TO START OF FILE NAME
     99 007466  000657                        BR      6$              ;DON'T BOTHER STORING @ IN CMDBUF
    100                                ;
    101                                ;  Move character to buffer
    102                                ;
    103 007470  020227  001342'      5$:     CMP     R2,#CBFEND      ;MAKE SURE WE DON'T OVERFLOW BUFFER
    104 007474  103404                       BLO     1$              ;BR IF OK
    105 007476                               FABORT  #ILLCMD         ;COMMAND TOO LONG
    106 007506  110022               1$:     MOVB    R0,(R2)+        ;MOVE CHAR TO CMDBUF
    107 007510  000646                       BR      6$              ;GO GET REST OF LINE
    108                                ;
    109                                ;  Reached end of line -- strip off any trailing spaces and tabs
    110                                ;
    111 007512  010267  170776       11$:    MOV     R2,CMDEND       ;SAVE POINTER PAST END OF COMMAND STRING
    112 007516  020227  001136'      4$:     CMP     R2,#CMDBUF      ;HAVE WE GONE PAST START OF BUFFER?
    113 007522  001002                       BNE     16$             ;BR IF NOT
    114 007524  000167  177312               JMP     RDCMD           ;YES -- THIS IS A NULL COMMAND
```

```
    115 007530  124227  000040          16$:     CMPB    -(R2),#'        ;IS NEXT CHARACTER A SPACE?
    116 007534  001770                           BEQ     4$              ;LOOP BACKWARD OVER SPACES
    117 007536  121227  000011                   CMPB    (R2),#TAB       ;IS THIS A TAB?
    118 007542  001765                           BEQ     4$              ;LOOP IF YES
    119 007544  005202                           INC     R2              ;POINT BEYOND LAST NON-BLANK CHARACTER
    120                                  ;
    121                                  ;  See if command line is continued
    122                                  ;
    123 007546  126227  177777  000055           CMPB    -1(R2),#'-      ;IS LINE CONTINUED?
    124 007554  001013                           BNE     GOTCML          ;BR IF NOT
    125                                  ;  Line is continued -- get more
    126 007556  005302                           DEC     R2              ;POINT BACK TO -
    127 007560                                   .GTLIN  #INBUF,#KMPRMT  ;READ COMMAND CONTINUATION LINE
    128 007600  000167  177366                   JMP     SCNCMD          ;CONTINUE SCANNING LINE
```

```
                                        ;
                                        ;  End of command found.
                                        ;  See if this command contains := which indicates this is a user
                                        ;  request to define a new command keyword.
                                        ;
007604  105012                  GOTCML: CLRB    (R2)              ;store null at end of command
007606  026727  170622  001136'         CMP     INDRFL,#CMDBUF    ;Was "@" first character of command?
007614  001443                          BEQ     INDCMD            ;Br if yes -- Don't call UCL for this
007616  105767  174500                  TSTB    COLEQL            ;Was := seen within command line?
007622  001411                          BEQ     13$               ;Br if not
007624  005767  000000G                 TST     UCLBLK            ;Are we supporting user-defined commands?
007630  001004                          BNE     1$                ;Br if yes
007632                                  FABORT  #EM$NUC           ;No user-defined commands allowed
007642  000167  001306          1$:     JMP     CALUCL            ;Go call UCL to process it
                                        ;
                                        ;  See if command line contains an indirect command file reference
                                        ;
007646  016703  170562          13$:    MOV     INDRFL,R3         ;Did we have indirect file reference?
007652  001024                          BNE     INDCMD            ;Br if yes
                                        ;
                                        ;  If we are to call TSXUCL before normal processing, determine if we
                                        ;  should do it for this command.
                                        ;
007654  126727  171256  000137          CMPB    CMDBUF,#'_        ;Was "_" specified as first char of command?
007662  001411                          BEQ     15$               ;Br if yes -- Don't call TSXUCL for _command
007664  032761  0000000G 0000000G       BIT     #$UCLCF,LSW7(R1)  ;Should we call TSXUCL before normal commands?
007672  001412                          BEQ     16$               ;Br if not
007674  005767  0000000G                TST     UCLBLK            ;Are we allowing user-defined commands?
007700  001407                          BEQ     16$               ;Br if not
007702  000167  001246                  JMP     CALUCL            ;Call TSXUCL to try to process this command
007706  112767  000040  171222  15$:    MOVB    #' ,CMDBUF        ;Replace leading underscore with space
007714  105267  174403                  INCB    NOUCL             ;Remember not to call TSXUCL for this command
007720  000167  000562          16$:    JMP     IDNCMD            ;Go try to identify the command
```

```
 1                                      ;
 2                                      ;  Command line contains indirect file reference.
 3                                      ;
 4                                      ;  Accrue indirect file name and try to open the file.
 5                                      ;
 6 007724  116701  000000G          INDCMD: MOVB    CORUSR,R1       ;GET JOB INDEX #
 7 007730  016703  170500                  MOV     INDRFL,R3       ;GET POINTER TO COMMAND FILE NAME
 8                                      ;
 9                                      ;  Accrue the command file spec
10                                      ;
11 007734  010304                          MOV     R3,R4           ;Save pointer to start of command file name
12 007736  012705  000244'                 MOV     #DKCOM,R5       ;SET DEFAULT DEV AND EXT
13 007742  004767  000000G                 CALL    ACRFN           ;ACCRUE THE FILE NAME
14 007746  103002                          BCC     4$              ;BR IF GOT NAME OK
15 007750  000167  177066                  JMP     RDCMD           ;ERROR WHILE GETTING FILE NAME
16                                      ;
17                                      ;  See if we should let IND process the indirect command file
18                                      ;
19 007754  026727  170454  001136' 4$:     CMP     INDRFL,#CMDBUF  ;Was "@" first character of command?
20 007762  001031                          BNE     1$              ;Br if not
21 007764  105767  174331                  TSTB    DOTAT           ;WAS PREFIX "#@"?
22 007770  001407                          BEQ     2$              ;BR IF NOT
23 007772  005767  000000G                 TST     INDSAV          ;IS IND AVAILABLE ON SYSTEM?
24 007776  001013                          BNE     3$              ;BR IF YES
25 010000                                  FABORT  #NOIND          ;IND IS NOT AVAILABLE
26 010010  032761  000000G 000000G 2$:     BIT     #$INDDF,LSW5(R1);IS IND TO BE CALLED BY DEFAULT?
27 010016  001413                          BEQ     1$              ;BR IF NOT
28 010020  105767  174274                  TSTB    DOLRAT          ;WAS COMMAND PREFIX "$@"?
29 010024  001010                          BNE     1$              ;BR IF YES -- DON'T USE IND
30 010026  132767  000000G 000000G 3$:     BITB    #IN$ACT,INDSTA  ;IS IND ACTIVE NOW?
31 010034  001004                          BNE     1$              ;BR IF YES
32 010036  010403                          MOV     R4,R3           ;Get back pointer to start of file spec
33 010040  005005                          CLR     R5              ;NO DEFAULT DEVICE
34 010042  000167  001372                  JMP     INDINI          ;GO ENTER IND
35                                      ;
36                                      ;  Process this indirect command file directly rather than calling IND
37                                      ;
38 010046  016700  170366          1$:     MOV     FILNAM,R0       ;Get the device name
39 010052  004767  000000G                 CALL    CHKTTD          ;Is the device TT?
40 010056  103441                          BCS     NOICMD          ;Br if yes -- Error
41 010060  004767  000000G                 CALL    PUSHCF          ;PUSH CURRENT @FILE ON STACK
42 010064  112767  000001  000000G         MOVB    #1,SERFLG       ;DO .SERR
43 010072                                  .LOOKUP #XAREA,#CFCHAN,#FILNAM
44 010112  112767  000000  000000G         MOVB    #0,SERFLG       ;DO .HERR, DON'T CLEAR CARRY FLAG
45 010120  103024                          BCC     CFOPEN          ;BR IF OPEN OK
46 010122  116701  000000G                 MOVB    CORUSR,R1       ;GET JOB INDEX NUMBER
47 010126  032761  000000G 000000G         BIT     #$SUCF,LSW9(R1) ;ARE WE WITHIN A START-UP COMMAND FILE?
48 010134  001410                          BEQ     5$              ;BR IF NOT
49 010136                                  FERR    #NOCF           ;PRINT ERROR MESSAGE
50 010152  000167  000000G                 JMP     CMDOFF          ;LOG THE JOB OFF
51 010156  004767  000000G          5$:     CALL    POPCF           ;POP PUSHED COMMAND FILE STATUS
52 010162                           NOICMD: FABORT  #NOCF           ;COULDN'T OPEN @FILE
53                                      ;
54                                      ;  We have successfully opened the indirect command file.
55                                      ;
56 010172  116701  000000G          CFOPEN: MOVB    CORUSR,R1
57 010176  052761  000000G 000000G         BIS     #$CFOPN,LSW4(R1);SAY CFCHAN IS OPEN
```

```
                                ;
                                ;  See if command file was installed with any privileges.
                                ;
010204  004767  001726                CALL    INSCF           ;See if command file was installed
                                ;
                                ;  READ IN 1ST BLOCK FROM INDIRECT FILE
                                ;
010210                                .READW  #XAREA,#CFCHAN,#CFBUF,#256.,#0
010246  103003                        BCC     CFPRM           ;BR IF READ OK
                                ;  ERROR OCCURED ON COMMAND FILE READ.
                                ;  THIS MUST MEAN THAT WE HAVE AN EMPTY COMMAND FILE.
                                ;  SET BUFFER POINTER TO CAUSE US TO IGNORE THIS BUFFER FULL.
010250  012767  000000G 000000G        MOV     #CFEND,CFPNT    ;SAY BUFFER IS EMPTY
                                ;
                                ;  SEE IF INDIRECT FILE HAS PARAMETERS.
                                ;
                                ;  R3 NOW POINTS PAST END OF COMMAND FILE NAME.
                                ;  SCAN ACROSS COMMAND LOOKING FOR 1ST PARAMETER
010256  105713                CFPRM:  TSTB    (R3)            ;HIT END OF COMMAND?
010260  001437                        BEQ     RDREST          ;BR IF END HIT
010262  122327  000040                CMPB    (R3)+,#' '      ;SKIP OVER LEADING SPACES
010266  001773                        BEQ     CFPRM
                                ;  SEE IF WE SHOULD USE SPACE OR \ AS PARAMETER DELIMITER
010270  012701  000040                MOV     #' ,R1          ;ASSUME SPACE IS DELIMITER
010274  124327  000134                CMPB    -(R3),#'\       ;DOES HE WANT TO USE \?
010300  001001                        BNE     8$              ;BR IF NOT -- USE SPACE
010302  112301                        MOVB    (R3)+,R1        ;USE \ AS PARAM SEPARATOR
                                ;  THERE ARE SOME PARAMETERS -- ACCRUE THEM
010304  012704  000000G         8$:    MOV     #PRMPNT,R4      ;POINT TO PARAM POINTER CELLS
010310  012705  000000G                MOV     #PRMBUF,R5      ;POINT TO PARAMETER STRING BUFFER
010314  020427  000000G         7$:    CMP     R4,#LSTPRM      ;TOO MANY PARAMETERS?
010320  103062                        BHIS    TOMPRM          ;BR IF TOO MANY
010322  010524                        MOV     R5,(R4)+        ;SET PARAMETER STRING POINTER
010324  121301                6$:     CMPB    (R3),R1         ;REACHED PARAM DELIMITER YET?
010326  001407                        BEQ     4$              ;BR IF YES
010330  105713                        TSTB    (R3)            ;HIT END OF COMMAND?
010332  001405                        BEQ     4$              ;BR IF YES
010334  020527  000000G               CMP     R5,#PRMEND      ;HIT END OF PARAM STRING BUFFER?
010340  103056                        BHIS    PTL             ;BR IF PARAM STRING TOO LONG
010342  112325                        MOVB    (R3)+,(R5)+     ;MOVE PARAMETER TO BUFFER
010344  000767                        BR      6$
                                ;  HIT END OF PARAMETER -- STORE NULL IN STRING TO MARK END.
010346  105025                4$:     CLRB    (R5)+           ;FLAG END OF THIS PARAMETER STRING
010350  105723                        TSTB    (R3)+           ;MORE TO ACCRUE?
010352  001360                        BNE     7$              ;BR IF YES
010354  010567  000000G               MOV     R5,PBFEND       ;SAVE POINTER TO END OF PARAM STRING
                                ;
                                ;  THE COMMAND FILE HAS BEEN OPENED AND ITS PARAMETERS HAVE
                                ;  BEEN ACCRUED AND STORED AWAY.
                                ;  IF THIS COMMAND FILE IS PART OF ANOTHER COMMAND (PART OF SAME LINE)
                                ;  THEN GO AND READ THE REST OF THE COMMAND.
                                ;  ELSE GO READ NEXT COMMAND LINE WHICH WILL COME FROM THE INDIRECT
                                ;  COMMAND FILE WE JUST OPENED.
                                ;
010360  016702  170050        RDREST: MOV     INDRFL,R2       ;GET POINTER TO START OF @SPECIFICATION
010364  001436                        BEQ     CFJMP           ;BR IF IMPLICIT @FILE EXECUTION
010366  020227  001136'               CMP     R2,#CMDBUF      ;IS @FILE 1ST THING IN COMMAND?
```

```
    115 010372  001433                        BEQ     CFJMP           ;BR IF YES
    116                                 ;
    117                                 ;  THIS INDIRECT FILE REFERENCE IS PART OF ANOTHER COMMAND SO
    118                                 ;  SUPPRESS THE LISTING OF THE 1ST LINE OF @FILE AS WE READ IT.
    119                                 ;
    120 010374  116701  000000G             MOVB    CORUSR,R1       ;GET USER INDEX #
    121 010400  016146  000000G             MOV     LSW4(R1),-(SP)  ;SAVE COMMAND FILE STATUS FLAGS
    122 010404  012705  000000G             MOV     #$QUIET,R5      ;GET QUIET FLAG INTO REGISTER
    123 010410  050561  000000G             BIS     R5,LSW4(R1)     ;TURN QUIET ON
    124 010414                              .GTLIN  #INBUF          ;READ LINE FROM @FILE
    125 010432  030526                      BIT     R5,(SP)+        ;NEED TO SET OR RESET QUIET MODE?
    126 010434  001002                      BNE     10$             ;BR IF WANT TO LEAVE IT SET
    127 010436  040561  000000G             BIC     R5,LSW4(R1)     ;RESET QUIET MODE
    128 010442  005067  167766          10$:    CLR     INDRFL          ;SAY @FILE REFERENCE HAS BEEN RESOLVED
    129 010446  105067  173646              CLRB    DOLRAT          ;SAY HAVEN'T SEEN "$@"
    130 010452  105067  173643              CLRB    DOTAT           ;SAY HAVEN'T SEEN "#@"
    131 010456  000167  176510              JMP     SCNCMD          ;GO GET REST OF COMMAND
    132 010462  000167  176354          CFJMP:  JMP     RDCMD           ;GO READ NEXT COMMAND
    133                                 ;  ERROR -- TOO MANY PARAMETERS
    134 010466                          TOMPRM: FABORT  #NUMPRM
    135                                 ;  ERROR -- PARAMETER STRING TOO LONG
    136 010476                          PTL:    FABORT  #STRLEN
```

```
                                        .SBTTL  Identify command
                                ;-------------------------------------------------------------
                                ;  AT THIS POINT A COMMAND LINE HAS BEEN ACCEPTED, CONTINUATION
                                ;  LINES READ, COMMENTS STRIPPED AND INDIRECT FILE REFERENCES
                                ;  RESOLVED.  THE RESULTING COMMAND IS STORED IN CMDBUF IN
                                ;  ASCIZ FORM WITH R2 POINTING TO THE END OF THE COMMAND.
                                ;
                                ;  TRY TO IDENTIFY IT AS A SYSTEM COMMAND.
                                ;
010506  012704  000000'        IDNCMD: MOV     #CMDHD,R4       ;POINT TO TABLE OF SYSTEM COMMANDS
010512  012703  001136'              MOV     #CMDBUF,R3      ;POINT TO OUR COMMAND
010516  004767  000000G              CALL    SEARCH          ;LOOK UP THE COMMAND KEYWORD
010522  103006                       BCC     FNDCMD          ;BR IF IDENTIFIED THE COMMAND
                                ;  ERROR DURING SEARCH.  SEE IF UNRECOGNIZED OR AMBIGUOUS.
010524  005704                       TST     R4              ;AMBIGUOUS OR UNRECOGNIZED?
010526  001407                       BEQ     NOCMD           ;BR IF UNRECOGNIZED
010530                               FABORT  #ABCMD          ;AMBIGUOUS COMMAND
                                ;
                                ;  WE HAVE A VALID SYSTEM COMMAND.
                                ;
                                ;  BRANCH OFF TO COMMAND PROCESSING ROUTINE.
                                ;  AT THIS POINT THE FOLLOWING REGISTERS ARE SET UP:
                                ;    R1 = USER INDEX NUMBER
                                ;    R2 = ADDRESS OF END OF COMMAND STRING
                                ;    R3 = ADDRESS OF START OF COMMAND ARGUMENT FIELD.
                                ;
010540  116701  000000G         FNDCMD: MOVB    CORUSR,R1       ;GET USER INDEX #
010544  000134                  JMP     @(R4)+          ;ENTER COMMAND PROCESSING ROUTINE
```

```
                                        ;------------------------------------------------------------
                                        ;  We could not identify the command as a standard system command
                                        ;  so we try to identify it as an implicit command in the following way:
                                        ;
                                        ;  1. See if it is a user-defined command (if SET UCL MIDDLE).
                                        ;  2. See if there is a command file on "DK:" with command name.
                                        ;  3. See if there is a command file on "SY:" with command name.
                                        ;  4. See if there is a SAV file on "SY:" with command name.
                                        ;  5. See if there is a SY:UCL program to process the command (if UCL LAST).
                                        ;
                                        ;  See if we should call TSXUCL to process this command.
                                        ;
010546  116701  000000G          NOCMD:  MOVB    CORUSR,R1       ;GET CURRENT JOB INDEX NUMBER
010552  032761  000000G 000000G          BIT     #$UCLCM,LSW7(R1);SET UCL MIDDLE?
010560  001410                          BEQ     7$              ;BR IF NOT
010562  105767  173535                  TSTB    NOUCL           ;DID UCL ALREADY REJECT THIS COMMAND?
010566  001005                          BNE     7$              ;BR IF YES
010570  005767  000000G                 TST     UCLBLK          ;ARE WE ALLOWING USER-DEFINED COMMANDS?
010574  001402                          BEQ     7$              ;BR IF NOT
010576  000167  000352                  JMP     CALUCL          ;SEND COMMAND TO TSXUCL
                                        ;
                                        ;  See if there is a command file on DK device with command name.
                                        ;
010602  012703  001136'          7$:     MOV     #CMDBUF,R3      ;POINT TO KEYWORD NAME
010606  012705  000244'                  MOV     #DKCOM,R5       ;GET DEV AND EXT DEFAULTS
010612  004767  000000G                 CALL    ACRFN           ;ACCRUE THE FILE NAME
010616  103513                          BCS     6$              ;BR IF ERROR IN GETTING FILE NAME
010620  016700  167614                  MOV     FILNAM,R0       ;Get the device name
010624  004767  000000G                 CALL    CHKTTD          ;Is it TT?
010630  103506                          BCS     6$              ;Error if yes
010632  004767  000000G                 CALL    PUSHCF          ;PUSH CURRENT @FILE STATUS
010636  112767  000001  000000G         MOVB    #1,SERFLG       ;DON'T ABORT ON ERRORS
010644                                  .LOOKUP #XAREA,#CFCHAN,#FILNAM
010664  112767  000000  000000G         MOVB    #0,SERFLG       ;DO .HERR -- DON'T CLEAR CARRY FLAG
010672  103402                          BCS     3$              ;BR IF NOT FOUND
010674  005005                          CLR     R5              ;SAY NO DEFAULT DEVICE NAME
010676  000424                          BR      4$              ;GO SEE IF WE SHOULD RUN IND
                                        ;
                                        ;  See if there is a command file on SY device with command name.
                                        ;
010700  016767  167360  167532  3$:     MOV     R50SY,FILNAM    ;CHANGE DEVICE NAME TO BE "SY"
010706  112767  000001  000000G         MOVB    #1,SERFLG       ;DO .SERR
010714                                  .LOOKUP #XAREA,#CFCHAN,#FILNAM
010734  112767  000000  000000G         MOVB    #0,SERFLG       ;DO .HERR, DON'T CLEAR CARRY FLAG
010742  103430                          BCS     1$              ;BR IF COMMAND IS NOT A COMMAND FILE
                                        ;
                                        ;  We located a command file on DK or SY.
                                        ;  See if we should call IND to execute it.
                                        ;
010744  012705  004324'                  MOV     #SYTXT,R5       ;GET DEFAULT DEVICE FOR IND FILE
010750  132767  000000G 000000G 4$:     BITB    #IN$ACT,CFIND   ;IS IND ACTIVE NOW?
010756  001015                          BNE     2$              ;BR IF ALREADY ACTIVE
010760  032761  000000G 000000G         BIT     #$INDDF,LSW5(R1);IS IND WANTED?
010766  001411                          BEQ     2$              ;BR IF NOT
010770                                  .PURGE  #CFCHAN         ;PURGE CHANNEL WE OPENED TO FILE
010776  004767  000000G                 CALL    POPCF           ;POP UP TO OLD FILE
011002  012703  001136'                  MOV     #CMDBUF,R3      ;POINT TO START OF COMMAND LINE
```

```
     58 011006  000167  000426                    JMP     INDINI          ;GO START UP IND
     59                                ;
     60                                ;  This is an implicit command file execution.
     61                                ;
     62 011012  052761  000000G 000000G 2$:       BIS     #$QUIET,LSW4(R1);ALWAYS SET QUIET IF IMPLICIT RUN
     63 011020  000167  177146                    JMP     CFOPEN          ;CONTINUE PROCESSING STARTUP OF @FILE
     64                                ;
     65                                ;  This is not an implicit @file call.
     66                                ;
     67 011024  004767  000000G          1$:     CALL    POPCF           ;REOPEN PREVIOUS @FILE
     68                                ;
     69                                ;  See if there is a program on "SY" with command name.
     70                                ;
     71 011030  012703  001136'                   MOV     #CMDBUF,R3      ;POINT TO COMMAND KEYWORD
     72 011034  012705  000250'                   MOV     #SYSAV,R5       ;SET DEFAULT DEV AND EXT
     73 011040  004767  000000G                   CALL    ACRFN           ;ACCRUE FILE NAME
     74 011044  103002                            BCC     5$              ;BR IF GOT FILE NAME OK
     75 011046  000167  175770          6$:       JMP     RDCMD           ;ERROR IN GETTING FILE NAME
     76 011052  016700  167362          5$:       MOV     FILNAM,R0       ;Get the device name
     77 011056  004767  000000G                   CALL    CHKTTD          ;Is the device TT?
     78 011062  103421                            BCS     BADCMD          ;Error if yes
     79 011064  112767  000001  000000G           MOVB    #1,SERFLG       ;DO .SERR
     80 011072                                    .LOOKUP #XAREA,#RUNCHN,#FILNAM
     81 011112  112767  000000  000000G           MOVB    #0,SERFLG       ;DO .HERR, DON'T CLEAR CARRY FLAG
     82 011120  103406                            BCS     TRYUCL          ;COULDN'T FIND PROGRAM
     83 011122  000167  000000G                   JMP     DORUN           ;START RUNNING THE PROGRAM
     84                                ;
     85 011126                         BADCMD:    FABORT  #ILLCMD         ;INVALID COMMAND
```

```
                                ;-----------------------------------------------------------------
                                ;  See if there is a SY:TSXUCL program to process the command
                                ;
011136  032761  000000G 000000G TRYUCL:  BIT     #$UCLCL,LSW7(R1);SHOULD WE CALL TSXUCL LAST?
011144  001451                          BEQ     URERR           ;BR IF NOT
011146  105767  173151                  TSTB    NOUCL           ;DID UCL ALREADY REJECT THIS COMMAND?
011152  001046                          BNE     URERR           ;BR IF YES
                                ;
                                ;  Call the TSXUCL program to process this command
                                ;
011154  105767  000000G         CALUCL: TSTB    VU$CL           ;IS THERE A UCL PROGRAM?
011160  001443                          BEQ     URERR           ;BR IF NOT
                                ;
                                ;  UCL option is genned in.  See if we can find UCL program.
                                ;
011162                                  .LOOKUP #XAREA,#RUNCHN,#UCLNAM ;TRY TO FIND SY:UCL.SAV
011202  103426                          BCS     9$              ;BR IF CAN'T FIND UCL PROGRAM
                                ;
                                ;  We found the UCL program.
                                ;  Pass command line to it in chain area.
                                ;
011204  012703  000012G                 MOV     #CINDAT+12,R3   ;POINT TO CHAIN AREA
011210  012704  001136'                 MOV     #CMDBUF,R4      ;POINT TO COMMAND LINE BUFFER
011214  112423                  1$:     MOVB    (R4)+,(R3)+     ;MOVE COMMAND TO CHAIN DATA AREA
011216  001376                          BNE     1$              ;LOOP TILL ASCIZ NULL MOVED
011220  162703  000013G                 SUB     #CINDAT+13,R3   ;COMPUTE LENGTH OF COMMAND LINE
011224  010367  000010G                 MOV     R3,CINDAT+10    ;AND STORE LENGTH INTO 510
011230  012704  000000G                 MOV     #UCLNAM,R4      ;POINT TO NAME OF UCL PROGRAM
011234  105267  000000G                 INCB    CINFLG          ;SIMULATE .CHAIN
011240  116701  000000G                 MOVB    CORUSR,R1       ;GET CURRENT JOB INDEX NUMBER
011244  052761  000000G 000000G        BIS     #$UCLRN,LSW7(R1);SAY THAT UCL PROGRAM IS RUNNING
011252  005000                          CLR     R0              ;NO RUN OPTION FLAGS FOR PLOAD
011254  000167  000000G                 JMP     PLOAD           ;LOAD AND RUN UCL PROGRAM
                                ;
                                ;  Cannot find TSXUCL program
                                ;
011260                          9$:     FABORT  #MISUCL         ;SAY UCL IS MISSING
                                ;
                                ;  Could not identify command
                                ;
011270                          URERR:  FABORT  #URCMD          ;UNRECOGNIZABLE COMMAND
```

```
                                        .SBTTL  CALUKM -- Start user-written command processor
                                ;-----------------------------------------------------------------------
                                ;  Call user-written command interface program.
                                ;
011300  116701  000000G         CALUKM: MOVB    CORUSR,R1       ;Get job index number
011304  012704  000000G                 MOV     #UCISPC,R4      ;Get pointer to program name for PLOAD
011310                                   .LOOKUP #XAREA,#RUNCHN,R4;Try to lookup program
011326  103406                         BCS     9$              ;Br if cannot find program
                                ;
                                ;  We found the program.  Enter it.
                                ;
011330  052761  000000G 000000G        BIS     #$UKMRN,LSW7(R1);Say user command processor is running
011336  005000                         CLR     R0              ;No run option flags for PLOAD
011340  000167  000000G                JMP     PLOAD           ;Enter the program
                                ;
                                ;  Cannot find the program.
                                ;
011344  042761  000000G 000000G 9$:     BIC     #$UKMON,LSW7(R1);Don't try to use the program again
011352                                   FABORT  #EM$NUK         ;Program not there
```

```
                                        .SBTTL  CMDIND -- IND command
                                ;-------------------------------------------------------------------------
                                ;  A command of the form "IND file" has been entered.
                                ;  Call the IND program to process the indirect command file.
                                ;
011362                          CMDIND:
                                ;
                                ;  See if IND is available
                                ;
011362  005767  000000G               TST     INDSAV          ;Is IND available on system?
011366  001004                        BNE     1$              ;Br if yes
011370                                FABORT  #NOIND          ;IND is not available
                                ;
                                ;  Accrue the file name so that we can check later to see if the
                                ;  command file was installed with any special privileges.
                                ;
011400  004767  000000G         1$:   CALL    SKPSPC          ;Skip over any spaces
011404  105713                        TSTB    (R3)            ;Any file specified?
011406  001411                        BEQ     4$              ;Br if not
011410  010304                        MOV     R3,R4           ;Save pointer to start of command file
011412  012705  000244'               MOV     #DKCOM,R5       ;Set default device and extension
011416  004767  000000G               CALL    ACRFN           ;Accrue the file name
011422  103002                        BCC     2$              ;Br if got file name ok
011424  000167  175412                JMP     RDCMD           ;Error accruing file name
011430  010403                  2$:   MOV     R4,R3           ;Restore pointer to start of file spec
                                ;
                                ;  Start execution of IND
                                ;
011432  005005                  4$:   CLR     R5              ;No default device string
011434  000167  000000                JMP     INDINI          ;Startup IND
```

```
                                        .SBTTL  INDINI -- Start IND program
                                ;-------------------------------------------------------------------
                                ;  Call IND to process an indirect command file.
                                ;
                                ;  Inputs:
                                ;   R3 = Pointer to asciz command line.
                                ;   R5 = Pointer to default device string (asciz)
                                ;   FILNAM = File spec for command file being started
                                ;
011440                          INDINI:
                                ;
                                ;  Error if IND is already running
                                ;
011440  132767  000000G 000000G        BITB    #IN$ACT,INDSTA  ;Is IND running now?
011446  001404                         BEQ     6$              ;Br if not
011450                                 FABORT  #INDACT         ;IND is already active
                                ;
                                ;  Set flag that says IND is being started
                                ;
011460  116701  000000G         6$:    MOVB    CORUSR,R1       ;Get job index number
011464  052761  000000G 000000G       BIS     #$INDRN,LSW5(R1);Say IND is running
                                ;
                                ;  See if command file was installed with any privileges
                                ;
011472  004767  000440                 CALL    INSCF           ;See if command file was installed
                                ;
                                ;  Build command line of the form "SY:IND file-name"
                                ;
011476  012702  001362'                MOV     #BLKO,R2        ;Point to area where we will build command
                                ;
                                ;  Insert "SY:IND "
                                ;
011502  012704  004330'                MOV     #SYINTX,R4      ;Point to "SY:IND " text string
011506  112422                 1$:    MOVB    (R4)+,(R2)+     ;Copy text string
011510  001376                         BNE     1$              ;Loop till null hit
011512  005302                         DEC     R2              ;Point back to null
                                ;
                                ;  See if we need to insert default device name
                                ;
011514  005705                         TST     R5              ;Do we have a default device name?
011516  001403                         BEQ     2$              ;Br if not
011520  112522                 3$:    MOVB    (R5)+,(R2)+     ;Move in default device name
011522  001376                         BNE     3$
011524  005302                         DEC     R2              ;Point back over terminating null
                                ;
                                ;  Copy command text string
                                ;
011526  010304                 2$:    MOV     R3,R4           ;Get pointer to command buffer
011530  112422                 4$:    MOVB    (R4)+,(R2)+     ;Copy command text string
011532  001376                         BNE     4$
                                ;
                                ;  Now move string back to command buffer
                                ;
011534  012702  001362'                MOV     #BLKO,R2        ;Point to start of new string
011540  010304                         MOV     R3,R4           ;Point to destination area
011542  112224                 5$:    MOVB    (R2)+,(R4)+     ;Copy string
011544  001376                         BNE     5$
```

```
                                        ;
                                        ;  Now process like a R command
                                        ;
011546  156767  000000G 000000G         BISB    UERSEV,INDERR   ;Pass error sev level to IND
011554  142767  000000G 000000G         BICB    #IN$CMD,INDSTA  ;Say no IND command pending
011562  012705  000250'                 MOV     #SYSAV,R5       ;Point to default device (SY)
011566  000167  000000G                 JMP     RUNNAM          ;Enter RUN code
                                        ;
                                        ;  Reopen channel to IND.SAV file
                                        ;
011572                                  INDRUN: .REOPEN #XAREA,#RUNCHN,#INDSAV ;REOPEN CHANNEL 16 TO IND
011612  012704  000372'                 MOV     #INDNAM,R4      ;POINT TO NAME OF IND
011616  105267  000000G                 INCB    CINFLG          ;SAY A .CHAIN IS IN PROGRESS
011622  156767  000000G 000000G         BISB    UERSEV,INDERR   ;PASS ERROR SEV LEVEL TO IND
011630  116701  000000G                 MOVB    CORUSR,R1       ;GET JOB INDEX NUMBER
011634  052761  000000G 000000G         BIS     #$INDRN,LSW5(R1);SAY IND IS RUNNING
011642  142767  000000G 000000G         BICB    #IN$CMD,INDSTA  ;SAY NO IND COMMAND PENDING
011650  005000                          CLR     R0              ;No run option flags for PLOAD
011652  000167  000000G                 JMP     PLOAD           ;GO START RUNNING IND
```

```
                                        .SBTTL  Process CCL commands
                                ;-------------------------------------------------------------
                                ;  THIS ROUTINE HANDLES CCL COMMANDS SUCH AS COMPILE, COPY, ETC.
                                ;  IT MOVES THE CCL COMMAND TO THE CHAIN DATA AREA THEN CALLS
                                ;  THE PROGRAM "SY:CCL.SAV" WHICH TRANSLATES THE CCL COMMAND
                                ;  INTO A SERIES OF SIMPLE COMMANDS WHICH ARE RETURN TO KMON
                                ;  IN THE CHAIN AREA WHEN CCL.SAV EXITS.
                                ;
                                ;  MOVE CCL COMMAND TO CHAIN AREA.
                                ;
011656  012703  0000126     CMDCCL: MOV     #CINDAT+12,R3   ;POINT TO CHAIN DATA AREA
011662  012704  001136'             MOV     #CMDBUF,R4      ;POINT TO COMMAND BUFFER
011666  112400              1$:     MOVB    (R4)+,R0        ;GET NEXT CHAR FROM COMMAND LINE
011670  120027  000011              CMPB    R0,#TAB         ;IS CHARACTER TAB?
011674  001403                      BEQ     10$             ;BR IF YES
011676  120027  000014              CMPB    R0,#FF          ;FORM-FEED?
011702  001002                      BNE     11$
011704  112700  000040      10$:    MOVB    #' ,R0          ;TRANSLATE TAB AND FF TO SPACE
011710  120027  000141      11$:    CMPB    R0,#141         ;SEE IF IT IS A LOWER CASE LETTER
011714  103405                      BLO     6$              ;BR IF NOT
011716  120027  000172              CMPB    R0,#172         ;LOWER CASE Z
011722  101002                      BHI     6$              ;BR IF DELIMITER
011724  162700  000040              SUB     #40,R0          ;CONVERT LOWER-CASE UP UPPER-CASE
011730  110023              6$:     MOVB    R0,(R3)+        ;MOVE CHAR TO CHAIN AREA
011732  001355                      BNE     1$              ;LOOP TILL ASCIZ NULL MOVED
                                ;  NOW SET SOME FLAGS IN LOCATION 510
011734  005003                      CLR     R3              ;FORM FLAG WORD IN R3
011736  016104  0000006             MOV     LSW5(R1),R4     ;PICK UP STATUS WORD
011742  032704  000000C             BIT     #$TECO!$KED,R4  ;IS DEFAULT EDITOR TECO OR KED?
011746  001417                      BEQ     2$              ;BR IF NOT (MUST BE EDIT THEN)
011750  032704  0000006             BIT     #$TECO,R4       ;IS IT TECO?
011754  001403                      BEQ     7$              ;BR IF NOT
011756  052703  000001              BIS     #C$TECO,R3      ;TELL CCL TO USE TECO
011762  000411                      BR      2$
011764  026127  0000006 0000006 7$:     CMP     LTRMTP(R1),#VT52;IS TERMINAL TYPE VT52?
011772  001403                      BEQ     8$              ;BR IF YES (EDITOR = K52)
011774  052703  000020              BIS     #C$KED,R3       ;DEFAULT EDITOR = KED
012000  000402                      BR      2$
012002  052703  000040          8$:     BIS     #C$K52,R3       ;DEFAULT EDITOR = K52
012006  032761  0000006 0000006 2$:     BIT     #$DIBOL,LSW6(R1);IS DIBOL OR DBL DEFAULT COMPILER?
012014  001402                      BEQ     9$              ;BR IF DBL IS DEFAULT
012016  052703  000100              BIS     #C$DIBL,R3      ;REMEMBER DIBOL IS DEFAULT
012022  032704  0000006         9$:     BIT     #$WILD,R4       ;WANT IMPLICIT/EXPLICIT WILDCARDS?
012026  001402                      BEQ     3$              ;BR IF WANTS EXPLICIT
012030  052703  000002              BIS     #C$WILD,R3      ;SET IMPLICIT-WILDCARD FLAG
012034  032704  0000006         3$:     BIT     #$CLTST,R4      ;DO CCL IN TEST MODE?
012040  001402                      BEQ     5$              ;BR IF NOT
012042  052703  000010              BIS     #C$TEST,R3      ;SET TEST MODE FLAG
012046  032761  0000006 0000006 5$:     BIT     #$QUIET,LSW4(R1);ARE COMMAND FILES IN QUIET MODE?
012054  001402                      BEQ     4$              ;BR IF NOT
012056  052703  000004              BIS     #C$QUIT,R3      ;SET QUIET FLAG
012062  110337  0000106         4$:     MOVB    R3,@#CINDAT+10  ;STORE FLAG WORD INTO CHAIN DATA AREA
012066  110137  0000116             MOVB    R1,@#CINDAT+11  ;STORE USER INDEX # IN 511 FOR CCL
                                ;  Reopen channel 16 with CCL file status saved during initialization.
012072                                  .REOPEN #XAREA,#RUNCHN,#CCLSAV ;OPEN CHANNEL TO CCL.SAV FILE
012112  012704  000362'              MOV     #CCLNAM,R4      ;POINT TO CELL WITH CCL NAME
012116  105267  0000006             INCB    CINFLG          ;SIMULATE CHAIN
```

Process CCL commands

```
     58 012122  052761  000000G 000000G          BIS     #$CCLRN,LSW5(R1);REMEMBER CCL.SAV IS RUNNING
     59 012130  005000                          CLR     R0                ;No run option flags for PLOAD
     60 012132  000167  000000G                 JMP     PLOAD             ;LOAD AND START CCL.SAV
```

```
                                        .SBTTL  INSCF  -- See if a command file is installed with priv
                                ;-----------------------------------------------------------------------
                                ;  This routine is called when we are starting a command file to see
                                ;  if the command file has been installed with any privileges.
                                ;  If so, the privileges are applied to the command file and current
                                ;  privileges for the job.
                                ;  PUSHCF should be called before this routine.
                                ;
                                ;  Inputs:
                                ;   FILNAM = File spec for command file being started
                                ;
012136  010246                  INSCF:  MOV     R2,-(SP)
                                ;
                                ;  See if this command file is installed
                                ;
012140  012700  000440'                 MOV     #FILNAM,R0      ;Point to file spec
012144  004767  0000000G                CALL    INSSRC          ;See if file is in install table
012150  103420                          BCS     9$              ;Br if not
                                ;
                                ;  Command file is installed.
                                ;  Apply any privilege changes.
                                ;
012152  012702  000000C                 MOV     #2*<PVNPW-1>,R2 ;Get index to last privilege word
012156  056262  000000C 0000000G 1$:     BIS     II$PRV+IIBUF(R2),PRIVFO(R2) ;Set some flags
012164  046262  000000C 0000000G        BIC     II$NPV+IIBUF(R2),PRIVFO(R2) ;Clear some flags
012172  162702  000002                  SUB     #2,R2           ;More to do?
012176  002367                          BGE     1$              ;Loop if yes
012200  056767  000001C 0000000G        BIS     II$FLG+IIBUF,AFCF ;Set command file attribute flags
012206  004767  0000000G                CALL    RSTPRV          ;Set current attributes for command file
                                ;
                                ;  Finished
                                ;
012212  012602                  9$:     MOV     (SP)+,R2
012214  000207                          RETURN
```

```
                                        .SBTTL  TRMINI -- Perform terminal-dependent initialization
                                ;-----------------------------------------------------------------------
                                ;  TRMINI is called during job start-up initialization to perform
                                ;  terminal dependent initialization.
                                ;
                                ;  Inputs:
                                ;   R1 = Job index number.
                                ;
012216  010246                  TRMINI: MOV     R2,-(SP)
                                ;
                                ;  Get initial LSW2 flags
                                ;
012220  016100  000000G                 MOV     LTRMTP(R1),R0   ;GET TERMINAL TYPE FLAGS
012224  016102  000000G                 MOV     LSW2(R1),R2     ;GET LSW2 FLAGS
                                ;  VT100
012230  032700  000000G                 BIT     #VT100,R0       ;IS THIS A VT100 TERMINAL?
012234  001410                          BEQ     14$             ;BR IF NOT
012236  042702  000000G                 BIC     #VT10NO,R2      ;SET FLAGS FOR LSW2
012242  052702  000000G                 BIS     #VT10FL,R2
012246                                  .PRINT  #IMVT10         ;CLEAR SCREEN
012254  000503                          BR      15$
                                ;  VT200
012256  032700  000000C         14$:    BIT     #VT2007!VT2008,R0 ;VT200 terminal?
012262  001410                          BEQ     32$             ;Br if not
012264  042702  000000G                 BIC     #VT20NO,R2      ;Set flags for VT200
012270  052702  000000G                 BIS     #VT20FL,R2
012274                                  .PRINT  #IMVT10         ;Clear screen
012302  000470                          BR      15$
                                ;  VT52
012304  032700  000000G         32$:    BIT     #VT52,R0        ;IS THIS A VT52 TERMINAL?
012310  001410                          BEQ     16$             ;BR IF NOT
012312                                  .PRINT  #IMVT52         ;CLEAR SCREEN
012320  042702  000000G                 BIC     #VT52NO,R2      ;SET FLAGS IN LSW2
012324  052702  000000G                 BIS     #VT52FL,R2
012330  000455                          BR      15$
                                ;  ADM3A
012332  032700  000000G         16$:    BIT     #ADM3A,R0       ;IS THIS AN ADM3A TERMINAL?
012336  001410                          BEQ     17$             ;BR IF NOT
012340                                  .PRINT  #IMADM3         ;CLEAR SCREEN
012346  042702  000000G                 BIC     #ADM3NO,R2      ;SET LSW2 FLAGS
012352  052702  000000G                 BIS     #ADM3FL,R2
012356  000442                          BR      15$
                                ;  LA36
012360  032700  000000G         17$:    BIT     #LA36,R0        ;IS THIS AN LA36?
012364  001405                          BEQ     18$             ;BR IF NOT
012366  042702  000000G                 BIC     #LA36NO,R2      ;SET FLAGS
012372  052702  000000G                 BIS     #LA36FL,R2
012376  000432                          BR      15$
                                ;  LA120
012400  032700  000000G         18$:    BIT     #LA120,R0       ;IS THIS AN LA120?
012404  001405                          BEQ     19$             ;BR IF NOT
012406  042702  000000G                 BIC     #LA12NO,R2      ;SET FLAGS
012412  052702  000000G                 BIS     #LA12FL,R2
012416  000422                          BR      15$
                                ;  Hazeltine
012420  032700  000000G         19$:    BIT     #HAZEL,R0       ;HAZELTINE TERMINAL?
012424  001410                          BEQ     20$             ;BR IF NOT
```

```
     58 012426                                          .PRINT  #IMHAZL          ;CLEAR SCREEN
     59 012434  042702  0000006                        BIC     #HAZLNO,R2       ;SET TERMINAL CONTROL FLAGS
     60 012440  052702  0000006                        BIS     #HAZLFL,R2
     61 012444  000407                                 BR      15$
     62                                    ;  DIABLE & QUME
     63 012446  032700  000000C                  20$:  BIT     #DIABLO!QUME,R0  ;DIABLO OR QUME TERMINAL?
     64 012452  001410                                 BEQ     1$               ;BR IF NOT
     65 012454  042702  0000006                        BIC     #DIABNO,R2       ;SET FLAGS
     66 012460  052702  0000006                        BIS     #DIABFL,R2
     67                                    ;  Store updated LSW2 flags.
     68 012464  010261  0000006                  15$:  MOV     R2,LSW2(R1)
     69 012470  010261  0000006                        MOV     R2,LSW2S(R1)
     70                                    ;
     71                                    ;  Finished
     72                                    ;
     73 012474  012602                           1$:   MOV     (SP)+,R2
     74 012476  000207                                 RETURN
```

```
                                        .SBTTL  VIRINI -- Virtual line initialization
                                ;-------------------------------------------------------------------------
                                ;  Perform initialization for virtual lines during job startup.
                                ;  If the primary line is using display windows then create a display
                                ;  window for the virtual line.  Otherwise print n>.
                                ;
                                ;  Inputs:
                                ;   R1 = Job index number
                                ;
012500  010246                  VIRINI: MOV     R2,-(SP)
012502  010346                          MOV     R3,-(SP)
012504  010446                          MOV     R4,-(SP)
                                ;
                                ;  Get job index number of our primary job
                                ;
012506  016102  000000G                 MOV     LNPRIM(R1),R2   ;GET PRIMARY LINE INDEX NUMBER
                                ;
                                ;  If primary line is using a display window, try to create one for
                                ;  this job.
                                ;
012512  005762  000000G                 TST     LWINDO(R2)      ;Is primary line using a display window?
012516  001413                          BEQ     1$              ;Br if not
012520  012700  001074'                 MOV     #MAKWIN,R0      ;Point to make-window argument block
012524  110260  000007                  MOVB    R2,7(R0)        ;Set index # of primary job
012530  106260  000007                  ASRB    7(R0)           ;Convert index to job number
012534  104375                          EMT     375             ;Try to create a window
012536  103403                          BCS     1$              ;Br if unable to create a window
012540  012700  001106'                 MOV     #MAPWIN,R0      ;Point to map-window argument block
012544  104375                          EMT     375             ;Select this window
                                ;
                                ;  Print n>
                                ;
012546  016204  000000G         1$:     MOV     LSECPT(R2),R4   ;GET ADDRESS OF TABLE WITH VIRTUAL LINES
012552  005003                          CLR     R3              ;COUNT VIRTUAL LINE #'S
012554  005203                  13$:    INC     R3
012556  120124                          CMPB    R1,(R4)+        ;LOOK UP OUR VIRTUAL LINE #
012560  001375                          BNE     13$
012562  062703  000060                  ADD     #'0,R3          ;CONVERT NUMBER TO ASCII
012566                                  .PRINT  #CRLF           ;PRINT CR-LF
012574                                  .TTYOUT R3              ;PRINT LINE #
012602                                  .TTYOUT #76             ;PRINT '>'
                                ;
                                ;  Try to copy any key definitions from primary line
                                ;
012612  016102  000000G                 MOV     LNPRIM(R1),R2   ;Get our primary line number
012616  004767  000000G                 CALL    INIUKD          ;Copy any user-defined keys
                                ;
                                ;  Finished
                                ;
012622  012604                  9$:     MOV     (SP)+,R4
012624  012603                          MOV     (SP)+,R3
012626  012602                          MOV     (SP)+,R2
012630  000207                          RETURN
```

```
                                        .SBTTL  CPYPRN -- Copy context info from parent job
                                ;-----------------------------------------------------------------------
                                ;  Copy context information from a parent job.
                                ;
012632  010246                  CPYPRN: MOV     R2,-(SP)
012634  010346                          MOV     R3,-(SP)
012636  010546                          MOV     R5,-(SP)
                                ;
                                ;  Return immediately if there is no parent job
                                ;
012640  016102  000000G                 MOV     LPARNT(R1),R2   ;Get # of parent job
012644  001002                          BNE     2$              ;Br if there is a parent job
012646  000167  000456                  JMP     9$              ;No parent job -- Nothing to copy
                                ;
                                ;  Copy file context and privilege information
                                ;
012652  010267  165704          2$:     MOV     R2,CPYCXT+2     ;Get # of parent job
012656  006267  165700                  ASR     CPYCXT+2        ;Convert to job number
012662  012700  000560'                 MOV     #CPYCXT,R0      ;Point to EMT arg block
012666  104375                          EMT     375             ;Copy context from parent job
012670  004767  000000G                 CALL    LDCLEN          ;Reinit LD status
                                ;
                                ;  Copy execution priority
                                ;
012674  012700  001070'                 MOV     #PRIEMT,R0      ;Point to emt argument block
012700  116260  000000G 000002          MOVB    LBSPRI(R2),2(R0);Get base priority from parent job
012706  104375                          EMT     375             ;Set the job priority
                                ;
                                ;  Copy user name and PPN
                                ;
012710  010203                          MOV     R2,R3           ;Get # of parent job
012712  010105                          MOV     R1,R5           ;Get # of our job
012714  070327  000006                  MUL     #6.,R3          ;Get offset to name of parent job
012720  070527  000006                  MUL     #6.,R5          ;Get offset to name of our job
012724  062703  000000G                 ADD     #LUNAME,R3      ;Point to tables
012730  062705  000000G                 ADD     #LUNAME,R5
012734  012700  000014                  MOV     #12.,R0         ;Get # chars to move
012740  112325                  1$:     MOVB    (R3)+,(R5)+     ;Copy user names
012742  077002                          SOB     R0,1$
012744  016261  000000G 000000G         MOV     LPROJ(R2),LPROJ(R1);Copy project number
012752  016261  000000G 000000G         MOV     LPROG(R2),LPROG(R1);Copy programmer number
012760  016167  000000G 000000G         MOV     LPROJ(R1),UPPN  ;Set project number
012766  016167  000000G 000002G         MOV     LPROG(R1),UPPN+2;Set programmer number
                                ;
                                ;  Copy flags from LSW2
                                ;
012774  016261  000000G 000000G         MOV     LSW2(R2),LSW2(R1) ;Copy LSW2 flags
013002  016261  000000G 000000G         MOV     LSW2S(R2),LSW2S(R1)
                                ;
                                ;  Copy some flags from LSW5
                                ;
013010  016200  000000G                 MOV     LSW5(R2),R0     ;Get flags from parent's LSW5
013014  042700  000000C                 BIC     #^C<ISPF5>,R0   ;Clear all but selected flags
013020  042761  000000G 000000G         BIC     #ISPF5,LSW5(R1) ;Clear those flags in our LSW5
013026  050061  000000G                 BIS     R0,LSW5(R1)     ;Transfer flags from parent job
                                ;
                                ;  Copy some flags from LSW6
```

```
                                        ;
013032  016200  000000G                 MOV     LSW6(R2),R0     ;Get flags from parent's LSW6
013036  042700  000000C                 BIC     #^C<ISPF6>,R0   ;Clear all but selected flags
013042  042761  000000G 000000G         BIC     #ISPF6,LSW6(R1) ;Clear those flags in our LSW6
013050  050061  000000G                 BIS     R0,LSW6(R1)     ;Transfer flags from parent job
                                        ;
                                        ;  Copy some flags from LSW7
                                        ;
013054  016200  000000G                 MOV     LSW7(R2),R0     ;Get flags from parent's LSW7
013060  042700  000000C                 BIC     #^C<ISPF7>,R0   ;Clear all but selected flags
013064  042761  000000G 000000G         BIC     #ISPF7,LSW7(R1) ;Clear those flags in our LSW7
013072  050061  000000G                 BIS     R0,LSW7(R1)     ;Transfer flags from parent job
                                        ;
                                        ;  Copy some flags from LSW9
                                        ;
013076  016200  000000G                 MOV     LSW9(R2),R0     ;Get flags from parent's LSW9
013102  042700  000000C                 BIC     #^C<ISPF9>,R0   ;Clear all but selected flags
013106  042761  000000G 000000G         BIC     #ISPF9,LSW9(R1) ;Clear those flags in our LSW9
013114  050061  000000G                 BIS     R0,LSW9(R1)     ;Transfer flags from parent job
                                        ;
                                        ;  Copy some flags from LSW11
                                        ;
013120  016200  000000G                 MOV     LSW11(R2),R0    ;Get flags from parent's LSW11
013124  042700  000000C                 BIC     #^C<ISPF11>,R0  ;Clear all but selexted flags
013130  042761  000000G 000000G         BIC     #ISPF11,LSW11(R1);Clear those flags in our LSW11
013136  050061  000000G                 BIS     R0,LSW11(R1)    ;Transfer flags from parent job
                                        ;
                                        ;  Copy Kmon prompt string
                                        ;
013142  010267  165714                  MOV     R2,EMCXCP+4     ;Set # of job we are copying from
013146  012767  000000G 165710          MOV     #KMPRMT,EMCXCP+6;Set address of item
013154  012767  000000C 165704          MOV     #<<MXPRMT+2>&76>,EMCXCP+8. ;Set # bytes to copy
013162  012700  001056'                 MOV     #EMCXCP,R0      ;Point to EMT arg block
013166  104375                          EMT     375             ;Copy Kmon prompt string
                                        ;
                                        ;  Copy default printer form name
                                        ;
013170  012767  000000G 165666          MOV     #UFORM,EMCXCP+6 ;Set address of item
013176  012767  000006  165662          MOV     #6,EMCXCP+8.    ;Set # of bytes to copy
013204  012700  001056'                 MOV     #EMCXCP,R0      ;Point to EMT arg block
013210  104375                          EMT     375             ;Copy default form name
                                        ;
                                        ;  Copy User Command Interface (UCI) file spec
                                        ;
013212  012767  000000G 165644          MOV     #UCISPC,EMCXCP+6;Set address of item to copy
013220  012767  000010  165640          MOV     #8.,EMCXCP+8.   ;Set # bytes to copy
013226  012700  001056'                 MOV     #EMCXCP,R0      ;Point to EMT arg block
013232  104375                          EMT     375             ;Copy UCI file spec
                                        ;
                                        ;  Copy subprocess start-up command file spec
                                        ;
013234  012767  000000G 165622          MOV     #SBPSUF,EMCXCP+6;Set address of item to copy
013242  012767  000020  165616          MOV     #16.,EMCXCP+8.  ;Set # bytes to copy
013250  012700  001056'                 MOV     #EMCXCP,R0      ;Point to EMT arg block
013254  104375                          EMT     375             ;Copy subprocess start-up file spec
                                        ;
                                        ;  Copy print-window device information
```

```
    115                                         ;
    116 013256  012767  000000G 165600          MOV     #JPWDEV,EMCXCP+6;Set address of print-window device
    117 013264  012767  000002  165574          MOV     #2,EMCXCP+8.    ;Set # bytes to copy
    118 013272  012700  001056'                 MOV     #EMCXCP,R0      ;Point to EMT arg block
    119 013276  104375                          EMT     375             ;Copy JPWDEV
    120 013300  012767  000000G 165556          MOV     #JPWTYP,EMCXCP+6;Set address of print-window dev type
    121 013306  012700  001056'                 MOV     #EMCXCP,R0      ;Point to EMT arg block
    122 013312  104375                          EMT     375             ;Copy JPWTYP
    123 013314  012767  000000G 165542          MOV     #JPWFLG,EMCXCP+6;Set address of print-window flag word
    124 013322  012700  001056'                 MOV     #EMCXCP,R0      ;Point to EMT arg block
    125 013326  104375                          EMT     375             ;Copy JPWFLG
    126                                 ;
    127                                 ;  Finihsed
    128                                 ;
    129 013330  012605                  9$:     MOV     (SP)+,R5
    130 013332  012603                          MOV     (SP)+,R3
    131 013334  012602                          MOV     (SP)+,R2
    132 013336  000207                          RETURN
```

```
                                        .SBTTL  PRTGRT -- Print the logon greeting message
                                ;-------------------------------------------------------------------------
                                ;  During logon processing, print the TSX-Plus  logon greeting message.
                                ;  The greeting message is different depending on the value of GREET and
                                ;  the PROFLG (flag indicating a professional machine).
                                ;
013340  020127  000000G       PRTGRT: CMP     R1,#LSTPL       ;IS THIS A REAL OR VIRTUAL LINE?
013344  003157                        BGT     GRTVIR          ;BR IF VIRTUAL -- ABBREVIATE GREETING
013346  005727  000002                TST     #GREET          ;IS PRINTED GREETING WANTED?
013352  001403                        BEQ     23$             ;BR IF NOT
013354                                .PRINT  #GRT1           ;TSX NAME AND VERSION NUMBER
013362  032761  000000G 000000G 23$:  BIT     #$1STLG,LSW6(R1);IS THIS THE 1ST TIME THIS LINE LOGGED ON?
013370  001064                        BNE     GRTFIN          ;BR IF NOT
013372  052761  000000G 000000G       BIS     #$1STLG,LSW6(R1);REMEMBER 1ST LOGON HAS OCCURED
013400  020127  000000G               CMP     R1,#LSTPL       ;IS THIS A REAL OR VIRTUAL LINE?
013404  101056                        BHI     GRTFIN          ;BR IF VIRTUAL -- SKIP LONG MESSAGE
013406  005727  000002                TST     #GREET          ;SHOULD WE PRINT GREETING MESSAGE?
013412  001453                        BEQ     GRTFIN          ;BR IF NOT
                                ;
                                ;  Calculate the message checksum.
                                ;
013414  012702  176532G               MOV     #SUMS-1246,R2   ;GET ADDRESS OF START OF MESSAGE
013420  005003                        CLR     R3              ;FORM CHECKSUM IN R3
013422  012704  001362'               MOV     #BLK0,R4        ;TEMPORARILY STORE GREETING HERE
013426  112200                2$:     MOVB    (R2)+,R0        ;GET CHARACTER FROM MESSAGE
013430  060003                        ADD     R0,R3           ;FORM CHECKSUM
013432  005400                        NEG     R0              ;DECRIPT CHARACTER
013434  001424                        BEQ     3$              ;BR IF END OF MESSAGE TEXT
013436  020227  000000G               CMP     R2,#LICTXT      ;CHECK FOR LICENSE TEXT
013442  103371                        BHIS    2$              ;BR IF INTO THE LICENSE TEXT
                                ;
                                ;  Check for various flags that indicated message variations.
                                ;
013444  105767  000000G               TSTB    PROFLG          ;ARE WE RUNNING ON A PROFESSIONAL?
013450  001003                        BNE     4$              ;BR IF YES -- PRINT TRUNCATED GREETING
013452  005727  000001                TST     #GREET-1        ;CHECK FOR TRUNCATED GREETING MESSAGE
013456  001013                        BNE     3$              ;BR IF FULL MESSAGE IS PRINTED
013460  020227  000000G        4$:    CMP     R2,#TRGRET      ;CHECK FOR TRUNCATED MESSAGE END
013464  103410                        BLO     3$              ;CONTINUE MOVING THE GREETING MESSAGE
013466  101357                        BHI     2$              ;GET THE NEXT CHARACTER
013470  112724  000056               MOVB    #'.,(R4)+       ;PLACE PERIOD AT THE END OF TRUNCATED MESSAGE
013474  112724  000015               MOVB    #CR,(R4)+       ;PLACE CARRIAGE RETURN AFTER PERIOD
013500  112724  000012               MOVB    #LF,(R4)+       ;PLACE LINE-FEED AFTER PERIOD
013504  000750                        BR      2$              ;GET THE NEXT CHARACTER
013506  110024                3$:     MOVB    R0,(R4)+        ;PUT IN TEMP BUFFER
013510  001346                        BNE     2$              ;BR IF MORE OF MESSAGE TEXT
013512                          1$:     .PRINT  #BLK0           ;PRINT GREETING MESSAGE
013520  005403                        NEG     R3              ;DECRIPT CHECKSUM
013522  005203                        INC     R3
                                ;*************************************************************
                                ;  If the following range of instructions are altered --
                                ;  change the program that assigns micro TSX-Plus license numbers.
013524  020327  000000G               CMP     R3,#SUCS        ;CHECK CHECKSUM
013530  001404                        BEQ     GRTFIN          ;BR IF OK
                                ;  Checksum failure -- kill TSX
013532  005037  000100                CLR     @#100           ;DIE HORRIBLY...
013536  005037  000060                CLR     @#60
```

```
                                ;
                                ;  Checksum ok, set up site number.
                                ;
013542  105767  000000G         GRTFIN: TSTB    PROFLG          ;ARE WE RUNNING ON A PROFESSIONAL?
013546  001415                          BEQ     1$              ;BR IF NOT
                                ;
                                ;  Print license number on a Professional
                                ;
013550                                  .PRINT  #TM$LN1         ;"License # = mmmm-TPS-"
013556  016705  000000G                 MOV     TSXSIT,R5       ;Get the license number
013562  004767  000000G                 CALL    PRTDEC          ;Print it
013566                                  .PRINT  #CRLF           ;Terminate the print line
013574  004767  000000G                 CALL    DATTIM          ;Print date and time
013600  000441                          BR      GRTVIR          ;Go print line number
                                ;
                                ;  Print license number if not on a professional
                                ;
013602  012767  000000G 000000G 1$:     MOV     #TSXLN,TSXSIT   ;STORE SITE NUMBER
                                ;  If the preceeding range of instructions are altered --
                                ;  change the program that assigns micro TSX-Plus license numbers.
                                ;*****************************************************************
013610  005727  000001                  TST     #GREET-1        ;SHOULD WE PRINT DATE AND TIME?
013614  002451                          BLT     GRTEND          ;BR IF NOT
013616  012704  001362'                 MOV     #BLK0,R4        ;TEMPORARILY STORE GREETING HERE
013622  012702  000000G                 MOV     #LICTXT,R2      ;OBTAIN LICENSE TEXT ADDRESS
013626  112200                  21$:    MOVB    (R2)+,R0        ;GET CHARACTER FROM MESSAGE
013630  005400                          NEG     R0              ;DECRIPT CHARACTER
013632  110024                  31$:    MOVB    R0,(R4)+        ;PUT IN TEMP BUFFER
013634  001374                          BNE     21$             ;BR IF MORE OF MESSAGE TEXT
013636  112724  000200                 MOVB    #200,(R4)+      ;NO CR/LF FOLLOWING TEXT
013642                                  .PRINT  #BLK0           ;PRINT GREETING MESSAGE
                                ;
                                ;  If not supported ("S" or "s"), then print license is unsupported.
                                ;
013650  116700  000000G         PRTSUP: MOVB    SUPCOD,R0       ;OBTAIN THE SUPPORT CODE (SIGN EXTEND)
013654  005400                          NEG     R0              ;DECRIPT THE CHARACTER
013656  120027  000123                  CMPB    R0,#'S          ;CHECK FOR SUPPORTED LICENSE
013662  001406                          BEQ     61$             ;BR IF LICENSE SUPPORTED
013664  120027  000163                  CMPB    R0,#'s          ;CHECK FOR SUPPORTED LICENSE
013670  001403                          BEQ     61$             ;BR IF LICENSE SUPPORTED
013672                                  .PRINT  #UNSUP          ;PRINT UNSUPPORTED LICENSE TEXT
013700  004767  000000G         61$:    CALL    DATTIM          ;PRINT CURRENT DATE & TIME
                                ;
                                ;  Print line number.
                                ;
013704                          GRTVIR: .PRINT  #LINNTX         ;DISPLAY LINE #
013712  010105                          MOV     R1,R5
013714  006205                          ASR     R5
013716  004767  000000G                 CALL    PRTDEC
013722                                  .PRINT  #CRLF
013730                                  .TTYOUT #LF
013740  000207                  GRTEND: RETURN
```

```
                                        .SBTTL  SETSUF -- Set up a start-up command file
                                ;-------------------------------------------------------------------
                                ;  Set up a start-up command file for execution by the job.
                                ;
                                ;  Inputs:
                                ;   R1 = Job index number
                                ;   R2 = Pointer to asciz command file name string.
                                ;
013742  010246                  SETSUF: MOV     R2,-(SP)
013744  010346                          MOV     R3,-(SP)
                                ;
                                ;  Start execution of a new command file
                                ;
013746  004767  000000G                 CALL    PUSHCF          ;SET UP FOR NEW COMMAND FILE
                                ;
                                ;  Move command file name to buffer
                                ;
013752  012703  000000G                 MOV     #CFBUF,R3       ;POINT TO COMMAND FILE BUFFER
013756  032761  000000G 000000G         BIT     #$DETCH,LSW(R1) ;Is this a detached job?
013764  001004                          BNE     63$             ;Br if yes
013766  112723  000136                  MOVB    #'^,(R3)+       ;PUT '^(@ AT FRONT OF COMMAND
013772  112723  000050                  MOVB    #'(,(R3)+
013776  112723  000100          63$:    MOVB    #'@,(R3)+
014002  112223                  5$:     MOVB    (R2)+,(R3)+     ;MOVE FILE NAME TO COMMAND AREA
014004  001376                          BNE     5$              ;LOOP TILL ALL MOVED
014006  112763  000015  177777        MOVB    #CR,-1(R3)      ;END LINE WITH CR-LF
014014  112723  000012                  MOVB    #LF,(R3)+
014020  105023                  6$:     CLRB    (R3)+           ;FILL REST OF BUFFER WITH NULLS
014022  020327  000000G                 CMP     R3,#CFEND
014026  103774                          BLO     6$
                                ;
                                ;  Finished
                                ;
014030  012603                          MOV     (SP)+,R3
014032  012602                          MOV     (SP)+,R2
014034  000207                          RETURN
                                ;
                004340'                 .END    START
Errors detected:  0

*** Assembler statistics


Work  file  reads:  0
Work  file writes:  0
Size of work file:  11288 Words  ( 45 Pages)
Size of core pool:  18176 Words  ( 71 Pages)
Operating  system:  RT-11

Elapsed time:  00:01:07.19
,LP:TSKMN1=DK:TSKMN1/C/N:SYM
```

Cross reference table (CREF V05.05)

```
$1STLG    1-79     29-12    29-14
$CCLRN    1-94     12-50    12-66    13-16    13-87    13-92    13-96    24-58
$CFABT    1-106    12-5     12-63
$CFCCL    1-111    13-94
$CFDCC    1-111    12-32
$CFKIL    1-143    10-29    12-5     12-59
$CFOPN    1-116    13-51    17-57
$CHACT    1-69     12-33
$CLTST    1-102    24-46
$CTRLC    1-105    10-30
$DEBUG    1-87     12-26
$DEFER    1-120    10-173
$DETCH    1-96     10-171   30-19
$DIBOL    1-79     24-40
$DOOFF    1-113    11-26
$DUPRN    1-108    11-21    11-23
$ECHO     1-110    10-173
$INDAB    1-105    12-51
$INDDF    1-142    17-26    19-53
$INDRN    1-142    13-16    13-87    13-96    23-21    23-73
$KED      1-124    10-119   24-29
$KINIT    1-65     10-19    10-190
$LC       1-110    10-173
$NOIN     1-69     13-115
$NOINT    1-102    12-34
$NOVLN    1-58     10-79
$NOWTT    1-69     12-33    12-38
$NTGCC    1-112    12-35
$PRGLK    1-95     13-102
$QUIET    1-121    13-91    17-122   19-62    24-49
$RNIOP    1-109    12-26
$SETRN    1-109    11-18
$SLLET    1-93     10-135
$SNWTT    1-141    12-36
$SUCF     1-65     13-114   17-47
$SYSPS    1-68     10-149
$TECO     1-127    10-117   24-29    24-31
$UCLCF    1-117    10-125   16-26
$UCLCL    1-119    10-131   20-4
$UCLCM    1-119    10-128   19-14
$UCLRN    1-118    13-30    13-40    13-98    20-31
$UKMON    1-94     14-24    21-18
$UKMRN    1-94     13-98    21-12
$VTESC    1-90     12-25
$WILD     1-128    10-141   24-43
...V1     10-92    10-92    10-99    10-99    10-99    10-108   10-108   10-109   10-109   11-19    11-19    11-38
          11-39    11-43    11-47    11-49    11-55    11-58    14-9     14-32    15-127   17-43    17-43    17-43
          17-65    17-65    17-124   19-33    19-33    19-33    19-43    19-43    19-43    19-55    19-80    19-80
          19-80    20-16    20-16    20-16    21-7     21-7     21-7     23-68    23-68    24-55    24-55    26-20
          26-27    26-32    26-39    26-58    27-39    27-40    27-41    29-11    29-47    29-66    29-69    29-88
          29-98    29-103   29-107   29-108
...V2     10-92    10-92    10-92#   10-92#   10-99    10-99    10-99#   10-99#   10-108   10-108   10-108#  10-108#
          10-109   10-109#  11-19    11-19#   17-43    17-43    17-43#   17-43#   17-65    17-65    17-65#   17-65#
          19-33    19-33    19-33#   19-33#   19-43    19-43    19-43#   19-43#   19-55    19-55#   19-80    19-80
          19-80#   19-80#   20-16    20-16    20-16#   20-16#   21-7     21-7     21-7#    21-7#    23-68    23-68
          23-68#   23-68#   24-55    24-55    24-55#   24-55#
```

Cross reference table (CREF V05.05)

```
AB1       1-74       4-30
AB10      1-75       4-37
AB11      1-75       4-38
AB12      1-75       4-39
AB13      1-75       4-40
AB14      1-75       4-41
AB15      1-75       4-42
AB16      1-75       4-43
AB2       1-75       4-31
AB3       1-75       4-32
AB4       1-75       4-33
AB5       1-75       4-34
AB6       1-75       4-35
ABCMD     1-77      18-17
ABM1      1-74       4-28
ABM10     1-74       4-21
ABM11     1-73       4-20
ABM12     1-73       4-19
ABM13     1-73       4-18
ABM14     1-73       4-17
ABM15     1-73       4-16
ABM16     1-73       4-15
ABM17     1-73       4-14
ABM2      1-74       4-27
ABM20     1-72       4-13
ABM21     1-72       4-12
ABM22     1-72       4-11
ABM23     1-72       4-10
ABM24     1-72       4-9
ABM3      1-74       4-26
ABM4      1-74       4-25
ABM5      1-74       4-24
ABM6      1-74       4-23
ABM7      1-74       4-22
ABRTAD    1-62      11-50      11-59*
ABRTCD    1-62      11-30      11-60*
ABRTCF    1-151     12-49      12-65      13-20
ABRTOV    1-62      11-56
ABTMSG    4-29#     11-37
ACRFN     1-151     17-13      19-26      19-73      22-22
ADM3A     1-136     26-37
ADM3FL    1-137     26-41
ADM3NO    1-138     26-40
AFCF      1-59      25-28*
ALCDEV    1-50       7-49#      8-49       8-54       8-59
ALCERR    1-72       4-7
ALDBLK    1-168      7-27#
ALDEMT    1-50       8-48#
ALDEX     1-168      7-26#
ALFN      1-177
AMBOPT    1-159
ASDEX     1-156      7-2#
ASKLNM    1-155      7-38#
AUTHFN    1-172      7-28#
BADCMD    1-155     19-78      19-85#
BELL      2-16#
```

```
BLANK    2-15#    12-22
BLKO     1-154     9-5#      9-6       9-7       10-105    11-44     11-47     23-29     23-54     29-24     29-47     29-81
        29-88
BLKWDS   2-20#
BOTEMT   1-47      8-108#
C$DIBL   2-31#    24-42
C$K52    2-30#    24-39
C$KED    2-29#    24-37
C$QUIT   2-27#    24-51
C$TECO   2-25#    24-33
C$TEST   2-28#    24-48
C$WILD   2-26#    24-45
CALUCL   1-164    16-14     16-30     19-20     20-11#
CALUKM  14-28     21-5#
CBFEND   9-2#     15-103
CCLNAM   1-175     7-30#    24-56
CCLSAV   1-100    24-55
CD$$SZ   1-114     8-66      8-75
CDBUF    1-50      8-67      8-76      9-8#
CDGADR   1-53      8-65#
CDGEMT   1-53      8-63#
CDPADR   1-53      8-74#
CDPEMT   1-53      8-72#
CFBUF    1-100    13-53     13-67     17-65     30-18
CFCHAN   1-120    17-43     17-43     17-65     17-65     19-33     19-33     19-43     19-43     19-55     19-55
CFEND    1-100    17-70     30-29
CFIND    1-113    19-51
CFJMP   17-113    17-115    17-132#
CFOPEN  17-45     17-56#    19-63
CFPNT    1-121    13-112    14-26     17-70*
CFPRM   17-66     17-76#    17-79
CFSP     1-116    10-43*
CFSPND   1-126    12-39     12-42*
CFSQEZ   1-121    13-55
CFSTK    1-65     10-43
CFSTRT   1-61     12-41
CHKTTD   1-175    17-39     19-29     19-77
CINDAT   1-132    13-21     13-32     13-34     13-61     20-22     20-26     20-27*    24-11     24-52*    24-53*
CINFLG   1-62     11-62*    12-7*     12-9      20-29*    23-70*    24-57*
CKLK    13-102#
CKPASL  12-73     13-5#
CKSF2   13-103    13-110#
CMDACC   1-85      6-68
CMDALC   1-83      6-51
CMDASN   1-83      6-48
CMDBOT   1-85      6-83      6-84
CMDBUF   1-164     9-1#     13-35     15-6      15-112    16-7      16-24     16-31*    17-19     17-114    18-11     19-24
        19-57     19-71     20-23     24-12
CMDCCL   1-175     6-11      6-15      6-16      6-17      6-18      6-19      6-26      6-27      6-28      6-29      6-30
         6-31      6-34      6-35      6-36      6-54      6-55      6-56      6-58      6-59      6-60      6-63      6-73
         6-74      6-76     24-11#
CMDDAT   1-82      6-37
CMDDEF   1-86      6-43
CMDDET   1-84      6-67
CMDDLC   1-83      6-52
CMDDMT   1-83      6-41
```

Cross reference table (CREF V05.05)

```
CMDDSN    1-148     6-49      6-50
CMDDSP    1-83      6-42
CMDEND    7-48#     13-39*    15-111*
CMDFMT    1-82      6-33
CMDFRM    1-148     6-32
CMDHD     1-68      6-8#      18-10
CMDIND    6-53      22-6#
CMDINI    1-85      6-71
CMDINS    1-78      6-79
CMDKIL    1-84      6-64
CMDMEM    1-84      6-57
CMDMNT    1-82      6-39
CMDMON    1-82      6-40
CMDNAC    1-85      6-69
CMDOFF    1-104     6-44      6-45      6-46      6-47      11-28     13-104    17-50
CMDPAU    1-84      6-65
CMDRCL    1-84      6-66
CMDREM    1-83      6-12
CMDRSM    1-66      6-13
CMDRST    1-81      6-14
CMDRSY    1-81      6-9
CMDRUN    1-81      6-10
CMDSET    1-81      6-21
CMDSHO    1-81      6-20
CMDSHT    1-86      6-85
CMDSND    1-81      6-22
CMDSPN    1-66      6-81
CMDSPO    1-84      6-62
CMDSQZ    1-85      6-72
CMDTIM    1-82      6-38
CMDUSE    1-85      6-70
CMDWHO    1-82      6-24      6-25
CMDYEL    1-80      6-23
COLEQL    9-21#     15-11*    15-51     15-60*    16-9
CORUSR    1-63      10-6      14-6      14-13     17-6      17-46     17-56     17-120    18-27     19-13     20-30     21-5
          23-20     23-72
CPUAH     1-164     7-44#
CPUAL     1-164     7-45#
CPYCXT    8-35#     28-17*    28-18*    28-19
CPYPRN    10-186    28-5#
CR        2-14#     13-71     29-42     30-26
CRLF      1-162     11-58     14-9      27-39     29-69     29-107
CSIMS1    1-161     3-8
CSIMS2    1-159     3-8
CSIMS3    1-176     3-8
CSIMS4    1-174     3-8
DATTIM    1-148     29-70     29-99
DELWIN    8-165#    12-17
DETARG    1-178     8-15#
DEVUNT    1-160     7-37#
DFJMEM    1-65      10-44
DIABFL    1-121     26-66
DIABLO    1-135     26-63
DIABNO    1-122     26-65
DIVSOR    1-173     3-5#
DKCOM     7-3#      17-12     19-25     22-21
```

Cross reference table (CREF V05.05)

```
DKSAV    1-150     7-5#
DLCEMT   1-50      8-53#
DMTARG   1-156     8-30#
DOLRAT   9-19#     15-9*     15-71*    17-28     17-129*
DORUN    1-148     19-83
DOTAT    9-20#     15-10*    15-84*    17-21     17-130*
EDIT     1-79      10-113
EDTFIL   1-166     11-46
EM$NUC   1-59      16-13
EM$NUK   1-52      21-19
EM$OVL   1-71      4-5
EM$SFP   1-71      4-4
EMCXCP   8-139#    28-87*    28-88*    28-89*    28-90     28-95*    28-96*    28-97     28-102*   28-103*   28-104    28-109*
         28-110*   28-111    28-116*   28-117*   28-118    28-120*   28-121    28-123*   28-124
ERRSEV   1-128     10-59*    12-47
ERRSPC   1-103     11-41     11-45     12-4*
ESC      1-52      2-19#
EXCJOB   1-57      10-14*
FF       2-18#     15-20     24-16
FILERM   1-76      11-43
FILNAM   1-152     7-41#     17-38     17-43     19-28     19-33     19-41*    19-43     19-76     19-80     25-16
FKILL    1-151     13-66     15-105    16-13     17-25     17-52     17-134    17-136    18-17     19-85     20-37     20-41
         21-19     22-12     23-16
FNDCMD   18-13     18-27#
FPRINT   1-151     10-27     17-49
GENMON   1-53      8-182#    10-207*   10-208    10-212*   10-213    12-74
GENTOP   1-104     10-25
GETSYP   1-61      10-151
GOTCML   15-124    16-6#
GREET    1-46      2-9#      29-9      29-17     29-36     29-79
GRT1     1-133     29-11
GRTEND   29-80     29-109#
GRTFIN   1-46      29-13     29-16     29-18     29-54     29-61#
GRTINI   10-172    10-178#
GRTVIR   29-8      29-71     29-103#
HANCHN   1-79      2-21#
HANIDX   1-160     7-46#
HAZEL    1-88      26-56
HAZLFL   1-88      26-60
HAZLNO   1-88      26-59
HNBUF    1-160     7-34#
HUPARG   1-174     8-102#
IDNCMD   13-44     16-33     18-10#
II$$SZ   1-71      8-83      8-91
II$FLG   1-59      25-28
II$NPV   1-59      25-25
II$PRV   1-59      25-24
IIBUF    1-45      3-9#      8-84      8-92      25-24     25-25     25-28
ILLCMD   1-157     15-105    19-85
ILSW2    1-92      10-79
IMADM3   1-76      26-39
IMHAZL   1-76      26-58
IMVT10   1-76      26-20     26-27
IMVT52   1-76      26-32
IN$ACT   1-142     14-15     14-17     17-30     19-51     23-14
IN$CMD   1-142     14-17     23-62     23-74
```

Cross reference table (CREF V05.05)

```
IN$CNT    1-142     14-18
INBUF     9-7#      14-32     15-13     15-127    17-124
INDABT    1-149     12-53     12-61
INDACT    1-153     23-16
INDCMD   16-8       16-19     17-6#
INDERR    1-106     12-46*    23-61*    23-71*
INDINI   17-34      19-58     22-30     23-10#
INDNAM    7-31#     23-69
INDRFL    7-39#     15-7*     15-66     15-70*    15-79     15-83*    15-96     15-98*    16-7      16-18     17-7      17-19
         17-112     17-128*
INDRUN   14-19      23-68#
INDSAV    1-142     17-23     22-10     23-68
INDSTA    1-106     14-14     17-30     23-14     23-62*    23-74*
INGADR    1-45       8-82#
INGEMT    1-45       8-80#
INIUKD    1-62      27-46
INPADR    1-45       8-90#
INPEMT    1-45       8-88#
INSCF    17-61      23-25     25-12#
INSSRC    1-59      25-17
ISPF11    1-60      28-81     28-82
ISPF5     1-60      28-53     28-54
ISPF6     1-60      28-60     28-61
ISPF7     1-60      28-67     28-68
ISPF9     1-60      28-74     28-75
JPWDEV    1-61      28-116
JPWFLG    1-64      28-123
JPWTYP    1-61      28-120
JS$KMN    1-71      12-75
JS$LOG    1-73      10-212
JS$ON     1-71      10-207
JSWLOC    1-64      11-61*    12-8*     13-5      13-18     13-41*    13-42*    13-65*    13-89     13-97*    14-5*
KCSIBF    1-155      9-6#
KCSIMS    1-156      3-8#
KDOCIN    1-121     12-11
KEYBUF    1-47       9-3#
KEYEND    1-49       9-4#
KILEMT    1-172      8-114#
KL3CLR    1-95      12-31
KL4CLR    1-123     12-24
KMINIT    1-57      10-180
KMNNAM    1-175      7-29#
KMNOT1    1-46      10-21     10-211    11-6#
KMONCE    1-57      10-12     10-178
KMPRMT    1-63       1-130    10-52*    10-53*    14-32     15-127    28-88
KMSTK     1-6        1-49      9-11#
L         6-9        6-9#      6-10      6-10#     6-11      6-11#     6-12      6-12#     6-13      6-13#     6-14      6-14#
          6-15       6-15#     6-16      6-16#     6-17      6-17#     6-18      6-18#     6-19      6-19#     6-20      6-20#
          6-21       6-21#     6-22      6-22#     6-23      6-23#     6-24      6-24#     6-25      6-25#     6-26      6-26#
          6-27       6-27#     6-28      6-28#     6-29      6-29#     6-30      6-30#     6-31      6-31#     6-32      6-32#
          6-33       6-33#     6-34      6-34#     6-35      6-35#     6-36      6-36#     6-37      6-37#     6-38      6-38#
          6-39       6-39#     6-40      6-40#     6-41      6-41#     6-42      6-42#     6-43      6-43#     6-44      6-44#
          6-45       6-45#     6-46      6-46#     6-47      6-47#     6-48      6-48#     6-49      6-49#     6-50      6-50#
          6-51       6-51#     6-52      6-52#     6-53      6-53#     6-54      6-54#     6-55      6-55#     6-56      6-56#
          6-57       6-57#     6-58      6-58#     6-59      6-59#     6-60      6-60#     6-61      6-61#     6-62      6-62#
          6-63       6-63#     6-64      6-64#     6-65      6-65#     6-66      6-66#     6-67      6-67#     6-68      6-68#
```

```
MAKWIN   8-152#   27-23
MAPWIN   8-160#   27-28
MAXMEM   1-64     10-48*
MAXPRI   1-64     10-51
MDT      1-8       1-88
MINTIM   1-101    10-28
MISUCL   1-77     20-37
MNBASE   1-167     8-171#
MNBPC    1-166     8-173#
MNFLGS   1-166     8-174#
MNTARG   1-174     8-25#    10-156
MNTDEV   1-156     7-47#     8-26      8-31      10-155*
MNTOP    1-167     8-172#
MONAR1   1-167     8-170#
MONAR2   1-167     8-178#
MONTXT   1-76     11-38
MSGBUF   1-170     8-42      8-43#
MSGEND   1-170     8-44#
MSTALC   1-72      4-6
MXJPRI   1-144    10-51*
MXPRMT   1-130    28-89
NAMTOP   1-133    10-49
NARGS    6-8#      6-9       6-9       6-10      6-10      6-11      6-11      6-12      6-12      6-13      6-13      6-14
         6-14      6-15      6-15      6-16      6-16      6-17      6-17      6-18      6-18      6-19      6-19      6-20
         6-20      6-21      6-21      6-22      6-22      6-23      6-23      6-24      6-24      6-25      6-25      6-26
         6-26      6-27      6-27      6-28      6-28      6-29      6-29      6-30      6-30      6-31      6-31      6-32
         6-32      6-33      6-33      6-34      6-34      6-35      6-35      6-36      6-36      6-37      6-37      6-38
         6-38      6-39      6-39      6-40      6-40      6-41      6-41      6-42      6-42      6-43      6-43      6-44
         6-44      6-45      6-45      6-46      6-46      6-47      6-47      6-48      6-48      6-49      6-49      6-50
         6-50      6-51      6-51      6-52      6-52      6-53      6-53      6-54      6-54      6-55      6-55      6-56
         6-56      6-57      6-57      6-58      6-58      6-59      6-59      6-60      6-60      6-61      6-61      6-62
         6-62      6-63      6-63      6-64      6-64      6-65      6-65      6-66      6-66      6-67      6-67      6-68
         6-68      6-69      6-69      6-70      6-70      6-71      6-71      6-72      6-72      6-73      6-73      6-74
         6-74      6-75      6-75      6-76      6-76      6-77      6-77      6-78      6-78      6-79      6-79      6-80
         6-80      6-81      6-81      6-82      6-82      6-83      6-83      6-84      6-84      6-85      6-85
NEWCMD   13-111    13-113    14-5#
NOABRT   11-31     12-4#
NOCF      1-77     17-49     17-52
NOCIN     1-155    12-10     12-17#
NOCMD    18-16     19-13#
NOFLAG    1-157     9-18#
NOICMD   17-40     17-52#
NOIND     1-161    17-25     22-12
NOUCL     9-22#    13-43*    15-12*    16-32*    19-16     20-6
NSPLDV    1-119    10-90
NUMPRM    1-77     17-134
OCTPRT    1-176    11-51
OFFEMT    1-172     8-96#
OKFAND    1-139    10-66*
OKFEND    1-102    10-67
OKFILE    1-102    10-66
OKFNND    1-139    10-67*
OPRCMD    1-84      6-61
OVLYEQ    1-176    11-55
P2$VIR    1-58     10-81
PASLIN    1-128    12-8      13-5      13-41     13-65     13-97
```

Cross reference table (CREF V05.05)

```
PBFEND    1-116     17-103*
PBUFND    1-48       9-17#
PLOAD     1-148     20-33     21-14     23-76     24-60
POPCF     1-122     17-51     19-56     19-67
PRGALL    1-148     11-16     11-24     12-58
PRIEMT    8-147#    28-25
PRIVA0    1-70      10-71*
PRIVA2    1-58      10-82*
PRIVC0    1-70      10-74*    10-85*
PRIVF0    1-60      10-73*    10-84*    25-24*    25-25*
PRIVS0    1-70      10-72*
PRIVS2    1-58      10-83*
PRMBUF    1-126     17-87
PRMEND    1-126     17-95
PRMPNT    1-125     17-86
PROFLG    1-86      29-34     29-61
PRSCMD   15-6#
PRTBUF    1-165      9-16#
PRTDEC    1-160     29-68     29-106
PRTGRT    1-46      10-181    29-7#
PRTR50    1-177     11-57
PRTSUP   29-92#
PTL      17-96      17-136#
PUSHCF    1-152     13-57     17-41     19-31     30-14
PVNPW     1-70      10-68     25-23
QUME      1-135     26-63
QUOTFL    7-40#     15-8*     15-31     15-33*    15-35     15-41*
R50BUF    1-156      7-43#
R50COM    1-153      7-24#
R50DIR    1-152      7-15#
R50DK     1-177      7-7#
R50DSK    1-158      7-10#
R50DUP    1-153      7-16#
R50IND    1-152      7-17#
R50K52    1-153      7-12#
R50KED    1-153      7-11#
R50KEX    1-153      7-13#
R50KMN    1-62       7-33#     11-6      11-7
R50LD     1-158      7-21#
R50LDO    1-156      7-22#
R50LD7    1-158      7-23#
R50LOG    1-162      7-9#
R50MON    1-178      3-7#
R50NO     1-159      7-35#
R50PIP    1-153      7-14#
R50SAV    1-152      7-25#
R50SY     1-152      7-8#      10-155    19-41
R50TSX    1-149      7-18#
R50TT     1-174      7-6#
R50UCL    1-149      7-19#
R50VIR    1-154      7-20#
RDCMD     1-150      6-77      6-78      6-80      6-82      14-8      14-13#    15-114    17-15     17-132    19-75     22-24
RDERM     1-154      4-36
RDREST   17-77      17-112#
REGEN     1-76      10-27
REMNDR    1-49       3-6#
```

```
RESDEV    1-139     10-65*
RJEMT     1-131      8-126#
RSTPRV    1-71      10-86     11-20     25-29
RUNARG    1-70      11-12*
RUNCHN    1-67      11-19     11-19     19-80     19-80     20-16     20-16     21-7      21-7      23-68     23-68     24-55
         24-55
RUNEMT    1-154      8-4#
RUNNAM    1-67      23-64
SBPSUF    1-61      10-199    28-109
SC$ERR    1-58      10-59
SCHAIN    1-120     12-8      13-5      13-18     13-41     13-65     13-89     13-97
SCNCMD   15-13#     15-128    17-131
SEARCH    1-150     18-12
SERFLG    1-63      11-11*    17-42*    17-44*    19-32*    19-34*    19-42*    19-44*    19-79*    19-81*
SETSUF   10-202     13-117    30-9#
SIZEMT    1-174      8-20#
SIZVAL    1-155      8-21#
SJEMT     1-131      8-120#
SKPSPC    1-60      22-17
SMONHD    1-72      12-72
SPGEMT    1-169      8-9#
SPLCHN    1-99      10-92
SPLHLA    1-163      8-132#
SPUBUF    1-66      10-50
START     1-48       8-110     10-5#     30-38
STDNAM    1-76      10-54
STRLEN    1-77      17-136
SUCF2     1-58      13-110    13-116    13-118*
SUCS      1-133     29-53
SUMS      1-133     29-22
SUPCOD    1-133     29-92
SXBPNT    1-66      10-50*
SYINTX    9-24#     23-33
SYSAV     1-150      7-4#     19-72     23-63
SYTXT     9-23#     19-50
TAB       2-17#     15-18     15-117    24-14
TALEMT    1-50       8-58#
TECO      1-79      10-115
TM$LN1    1-78      29-66
TOMPRM   17-89      17-134#
TOOLNG    1-77      13-66
TRGRET    1-133     29-38
TRMINI   10-161     26-9#
TRYUCL   19-82      20-4#
TSKMN1    1-11#      1-51
TSKMON    1-10#      1-46      1-51
TSXLN     1-133     29-75
TSXSIT    1-133     29-67     29-75*
UC$MDC    1-147     10-107*
UC$NDC    1-147     10-106*
UCHAN     1-111     11-13     11-15*
UCIDEF    1-52       3-4#
UCISPC    1-91      21-6      28-102
UCLBLK    1-146     10-97     16-11     16-28     19-18
UCLCMD    1-85       6-75
UCLDAT    1-146     10-99
```

Cross reference table (CREF V05.05)

```
UCLNAM    1-115     20-16     20-28
UERSEV    1-128     11-35*    12-46     12-47     12-54*    23-61     23-71
UFORM     1-105     10-55     28-95
UFPTRP    1-117     11-9*
UHIMEM    1-107     10-49*
UNSUP     1-78      29-98
UPPN      1-97      28-42*    28-43*
URCMD     1-78      20-41
URERR    20-5       20-7      20-12     20-41#
USPLCH    1-99      10-92     10-92
UTRPAD    1-64      11-8*
VEDIT     1-69      10-113    10-115
VIRINI   10-167     27-10#
VT100     1-135     26-16
VT10FL    1-137     26-19
VT10NO    1-137     26-18
VT2007    1-140     26-23
VT2008    1-140     26-23
VT20FL    1-140     26-26
VT20NO    1-140     26-25
VT52      1-135     24-35     26-30
VT52FL    1-136     26-34
VT52NO    1-122     26-33
VU$CL     1-115     20-11
VUCLMC    1-116     10-107
VUCLOR    1-117     10-123    10-126    10-129
WILDFL    1-69      10-139
XAREA     1-151      7-42#    10-92     10-99     10-108    17-43     17-65     19-33     19-43     19-80     20-16     21-7
         23-68      24-55
YELEMT    1-171      8-40#
```

Cross reference table (CREF V05.05)

```
...CM0   14-32   14-32   14-32   14-32   15-127  15-127  15-127  15-127  17-124  17-124  17-124  17-124
...CM1   10-92   10-99   10-108  17-43   17-65   19-33   19-43   19-80   20-16   21-7    23-68   24-55
...CM2   10-92   10-99   10-99   10-108  10-108  10-108  10-108  17-43   17-43   17-65   17-65   17-65
         17-65   19-33   19-33   19-43   19-43   19-80   19-80   20-16   20-16   21-7    21-7    23-68
         24-55
...CM3   10-109  11-19   19-55
...CM5   10-92   10-99   10-108  11-38   11-39   11-43   11-47   11-49   11-55   11-58   14-9    17-43
         17-65   19-33   19-43   19-80   20-16   21-7    23-68   24-55   26-20   26-27   26-32   26-39
         26-58   27-39   27-40   27-41   29-11   29-47   29-66   29-69   29-88   29-98   29-103  29-107
         29-108
...CM7   10-108  17-65
.CLOSE    1-39#  10-109  11-19
.CSIGE    1-37#
.CSISP    1-35#
.DATE     1-38#
.ENTER    1-40#
.EXIT     1-40#
.FPROT    1-41#
.GTIM     1-38#
.GTLIN    1-38#  14-32   15-127  17-124
.GVAL     1-41#
.HERR     1-41#
.LOOKU    1-39#  10-99   17-43   19-33   19-43   19-80   20-16   21-7
.PRINT    1-39#  11-38   11-39   11-43   11-47   11-49   11-55   11-58   14-9    26-20   26-27   26-32
         26-39   26-58   27-39   29-11   29-47   29-66   29-69   29-88   29-98   29-103  29-107
.PURGE    1-36#  19-55
.PVAL     1-41#
.READW    1-36#  17-65
.REOPE    1-37#  10-92   23-68   24-55
.SAVES    1-37#
.SERR     1-41#
.SRESE    1-35#
.TTOUT    1-35#
.TTYIN    1-36#
.TTYOU    1-36#  27-40   27-41   29-108
.WRITW    1-40#  10-108
CMDDEF    5-38#   6-9     6-10    6-11    6-12    6-13    6-14    6-15    6-16    6-17    6-18    6-19
          6-20    6-21    6-22    6-23    6-24    6-25    6-26    6-27    6-28    6-29    6-30    6-31
          6-32    6-33    6-34    6-35    6-36    6-37    6-38    6-39    6-40    6-41    6-42    6-43
          6-44    6-45    6-46    6-47    6-48    6-49    6-50    6-51    6-52    6-53    6-54    6-55
          6-56    6-57    6-58    6-59    6-60    6-61    6-62    6-63    6-64    6-65    6-66    6-67
          6-68    6-69    6-70    6-71    6-72    6-73    6-74    6-75    6-76    6-77    6-78    6-79
          6-80    6-81    6-82    6-83    6-84    6-85
FABORT    5-16#  13-66   15-105  16-13   17-25   17-52   17-134  17-136  18-17   19-85   20-37   20-41
         21-19   22-12   23-16
FERR      5-5#   10-27   17-49
TBLDEF    5-26#   6-8
TBLEND    5-52#   6-86
```