```
TSKMN1 -- TSX-Plus Keyboard Mon MACRO V05.04  Monday 21-Dec-87 11:40
Table of contents
```

```
    6-    1       Table of command keywords
   10-    1       Job Initialization
   11-    1       Entry to KMON
   14-    1       Get keyboard command
   18-    1       Identify command
   21-    1       CALUKM -- Start user-written command processor
   22-    1       CMDIND -- IND command
   23-    1       INDINI -- Start IND program
   24-    1       Process CCL commands
   25-    1       INSCF  -- See if a command file is installed with priv
   26-    1       TRMINI -- Perform terminal-dependent initialization
   27-    1       VIRINI -- Virtual line initialization
   28-    1       CPYPRN -- Copy context info from parent job
   29-    1       PRTGRT -- Print the logon greeting message
   30-    1       SETSUF -- Set up a start-up command file
```

```
                                    .TITLE  TSKMN1 -- TSX-Plus Keyboard Monitor
                                    .ENABL  LC
                                    .DSABL  GBL
000000                              .ASECT
          000042           .        =       42
000042    003736'                   .WORD   KMSTK           ;DEFINE INITIAL STACK POINTER
          000300           .        =       300
000300    000000G                   .WORD   MDT             ;# MINUTES OF SYSTEM UPTIME IF DEMO VERSION
000000                              .CSECT  TSKMON
000000                     TSKMON:
000000                     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.
                           ;  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  CTMSG,LGOVER,PPNMSG,BOTEMT,KEYBUF
                                    .GLOBL  START,CPUMSG,CF2DEP,PBUFND,KMFTXT
                                    .GLOBL  REMNDR,R50CHR,KEYEND,KMSTK,BADBOT
                                    .GLOBL  ALDEMT,DLCEMT,TALEMT,ALCDEV,CDBUF
                                    .GLOBL  MONTAB,TSKMON,ASNOVF,AMBOPT,TSKMN1
                                    .GLOBL  WRNHED,EM$NUK,ESC,CSHMSG,UCIDEF
                                    .GLOBL  EM$DAA,GENMON,CDGEMT,CDGADR,CDPEMT,CDPADR
                           ;
                           ;  Global references
                           ;
                                    .GLOBL  $NOVLN,P2$VIR,PRIVA2,PRIVS2,SC$ERR,SUCF2
```

```
        .GLOBL  INSSRC,II$PRV,II$NPV,IIBUF,AFCF,II$FLG,EM$NUC
        .GLOBL  ISPF5,ISPF6,ISPF7,ISPF9,PRIVFO,SKPSPC,ISPF11
        .GLOBL  LBSPRI,GETSYP,JPWDEV,JPWTYP,SBPSUF,CFSTRT
        .GLOBL  ABRTAD,ABRTCD,CINFLG,$VNOTT,INIUKD,R50KMN,ABRTOV
        .GLOBL  CORUSR,LSW,SERFLG,IOABFL,KMPRMT,MXPRMT
        .GLOBL  UTRPAD,JSWLOC,ERRLOC,MAXMEM,MAXPRI,JPWFLG
        .GLOBL  USRSTK,$KINIT,CFSTK,MXJMEM,DFJMEM,$SUCF
        .GLOBL  SPUBUF,SXBPNT,MXJADR,LWINDO,CMDSPN,CMDRSM
        .GLOBL  TMTOTH,TMTOTL,TMUSRH,TMIOWH,RUNCHN,RUNNAM
        .GLOBL  TMSWTH,TMIDLH,TMIOH,TMSWPH,CMDHD,$SYSPS
        .GLOBL  VEDIT,WILDFL,$NOIN,$NOWTT,$HITTY,$CHACT
        .GLOBL  PO$DBG,PRIVCO,PRIVAO,PRIVSO,PO$SPV,PVNPW,RUNARG
        .GLOBL  JS$KMN,JS$ON,LMONHD,EM$OVL,II$$SZ,RSTPRV,EM$SFP
        .GLOBL  MSTALC,ALCERR,ABM24,ABM23,ABM22,ABM21,ABM20,SMONHD
        .GLOBL  ABM17,ABM16,ABM15,ABM14,ABM13,ABM12,ABM11,JS$LOG
        .GLOBL  ABM10,ABM7,ABM6,ABM5,ABM4,ABM3,ABM2,ABM1,AB1
        .GLOBL  AB2,AB3,AB4,AB5,AB6,AB10,AB11,AB12,AB13,AB14,AB15,AB16
        .GLOBL  REGEN,STDNAM,IMVT10,IMVT52,IMADM3,IMHAZL,MONTXT,FILERM
        .GLOBL  LOCMSG,TOOLNG,NOCF,NUMPRM,STRLEN,ABCMD,MISUCL
        .GLOBL  URCMD,TM$LNI,UNSUP,LINNTX,CMDINS
        .GLOBL  TECO,EDIT,KED,K52,$1STLG,$DIBOL,HANCHN
        .GLOBL  SH$VAL,SH$NAM,SH$$SZ,SH$RTN,SH$FLG
        .GLOBL  SO$NVL,SO$OCT,SO$NO,HANENT,HANSIZ,CMDYEL
        .GLOBL  CMDRSY,CMDRUN,CMDRST,CMDSHO,CMDSET,CMDSND
        .GLOBL  CMDWHO,CMDFMT,CMDDAT,CMDTIM,CMDMNT,CMDMON
        .GLOBL  CMDDMT,CMDDSP,CMDASN,CMDALC,CMDDLC,CMDREM
        .GLOBL  CMDMEM,OPRCMD,CMDSPO,CMDKIL,CMDPAU,CMDDET,CMDRCL
        .GLOBL  CMDACC,CMDUSE,CMDINI,CMDSQZ,UCLCMD,CMDBOT,CMDBOT
        .GLOBL  CMDSHT,PROFLG,CMDDEF
        .GLOBL  LSW9,$DEBUG,LSW11
        .GLOBL  HAZEL,HAZLFL,HAZLNO,$MLOCK,MDT
        .GLOBL  LINBUF,LINNXT,LSTACT,PRGTOP,PRGSIZ,KMNHI
        .GLOBL  KMNTOP,KMNPGS,KMNSTK,KMNSTR,CXTPAG
        .GLOBL  LINPNT,LINCNT,LACTIV,LRDTIM,CS$RON
        .GLOBL  LOTBUF,LOTNXT,LOTPNT,$VTESC
        .GLOBL  LOTSIZ,LOTSPC,LCOL,UCISPC,$MAPOK
        .GLOBL  LAFSIZ,LFWLIM,LINCUR,NUMON,ILSW2
        .GLOBL  $CARUP,$SLON,$SLTTY,$SLLET,$SLKED
        .GLOBL  LSUCF,$CCLRN,$UKMON,UKMNAM,$UKMRN
        .GLOBL  KL3CLR,$PRGLK,LSW5,PVON,S$SPND
        .GLOBL  S$TWFN,S$TTFN,S$OTFN,S$IOFN,S$OTLO
        .GLOBL  LSTDL,FSTDL,$DETCH,UMSYTP
        .GLOBL  $DISCN,LPROJ,LPROG,LUNAME,UPPN
        .GLOBL  LCPUHI,LCPULO,LCONTM,$CTRLS,$SPLJB
        .GLOBL  STPFLG,TOTON,USPLCH,SPLCHN
        .GLOBL  S$INWT,S$OTWT,S$TMWT,S$SFWT
        .GLOBL  S$MSWT,CFBUF,CFEND,CCLSAV,KMNCHN
        .GLOBL  MINTIM,LSECPT,MAXSEC,$EMTTR
        .GLOBL  OKFILE,OKFEND,$CLTST,$NOINT
        .GLOBL  LJSW,CTRLTT,NEWJSW,JSTKND,ERRSPC
        .GLOBL  USTART,GENTOP,BOTDEV,BOTUNI,CMDOFF
        .GLOBL  $CTRLC,LSW2,$INKMN,CHAIN,UFORM,$INDAB
        .GLOBL  MAXASN,$CFABT,INDSTA,INDERR
        .GLOBL  RUNDEV,LNBLKS,CXTBAS,CXTWDS,UHIMEM
        .GLOBL  ASNTBL,$DILUP,LNSBLK,RUNFLG
        .GLOBL  ASNEND,LSW3,LSW2S,$DUPRN
        .GLOBL  $FORM,$TAB,LSCCA,$CFSOT,$RNIOP,$SETRN
        .GLOBL  $PAGE,$SCOPE,$ECHO,$LC
        .GLOBL  UCHAN,$FORMO,$CFALL,$CFDCC,$CFCCL
        .GLOBL  LNPRIM,LNMAP,CW$50H,CONFIG,$NTGCC
        .GLOBL  $DOOFF,NUCHN,LRBFIL,CFIND,LPARNT
        .GLOBL  C.CSW,C.DEVQ,C.SBLK,NLINES
        .GLOBL  CD$NAM,CD$DVU,CD$BAS,CD$JOB,CD$$SZ,CD$$UB
        .GLOBL  LTSCMD,LNSPAC,CFNEST,VU$CL,UCLNAM
        .GLOBL  $CFOPN,CFSEND,PBFEND,CFSP,$TTGAG,VUCLMC
        .GLOBL  UFPTRP,SDSFCB,SD$DEL,CFLFL4,$UCLCF,VUCLOR
        .GLOBL  SDFLAG,SD$FLK,SD$WFM,SDFORM,$UCLRN
        .GLOBL  SDBUF1,SDBLK,NSPLDV,LD$RON,$UCLCM,$UCLCL
        .GLOBL  LDNAME,LDSIZE,LDFLAG,LDBASE,LDPDEV
        .GLOBL  $DEFER,CFCHAN,SCHAIN,LDDEVX
        .GLOBL  CFPNT,CFBLK,$QUIET,DIABFL,KDOCIN,CFSQEZ
        .GLOBL  DIABNO,VT52NO,LA36NO,LA36FL,POPCF
        .GLOBL  LSW4,KL4CLR,SDSKIP,SDBU,SD$BAK
        .GLOBL  $INCOR,$KED
        .GLOBL  SF$BSY,SFFORM,SD$SNG,SFNMBL,NFRESB
        .GLOBL  SD$HLD,SF$HLD,CURPRM,PRMPNT,SF$1ST
        .GLOBL  LSTPRM,PRMBUF,PRMEND,CFSPND
        .GLOBL  SDFHD,SFFLAG,SFQLNK,CFHOLD
        .GLOBL  LCOL,$QTSET,$TECO,CD$TOP
        .GLOBL  $WILD,ERRSEV,UERSEV,PASLIN
        .GLOBL  LSTPL,SDCB,SDCBND
        .GLOBL  VQUAN1,VQUN1A,VQUAN2,VHIPCT
        .GLOBL  DCTRD,DCCRD,DCTWR,DCCWR
        .GLOBL  VCORTM,KMPRMT,MXPRMT
        .GLOBL  RDB,RDBEND,RT$NAM,RT$$SZ
        .GLOBL  SDCBSZ,LSTSL,LSTATE,SJEMT,RJEMT
        .GLOBL  TK1VAL,CINDAT,SYSDAT,SYTIMH,SYTIML
        .GLOBL  BASMAP,LOMAP,HIMAP,JCXPGS
        .GLOBL  TSXLN,TSXSIT,GRT1,TRGRET,LICTXT,SUPCOD,NAMTOP,SUMS,SUCS
        .GLOBL  LPRG1,LPRG2,S$QUSR,S$IOWT,S$SFWT
        .GLOBL  S$SPDB,S$SPCB,SFUSER,SFFILE
        .GLOBL  LCBIT,LA36,LA120,VT52,VT100,DIABLO,QUME
        .GLOBL  ADM3A,LTRMTP,LA12FL,LA12NO,VT52FL
        .GLOBL  VT10FL,VT1ONO,QUMEFL,QUMENO,ADM3FL
        .GLOBL  ADM3NO,SYINDX,SYUNIT,NUMDEV,PNAME
        .GLOBL  OF$DEV,OF$UNT,OF$FIL,OF$FLG,SYNAME
        .GLOBL  OF$$SZ,OT$RON,RESDEV,$TAPE
        .GLOBL  KMNBAS,VT2007,VT2008,VT20NO,VT20FL
        .GLOBL  LSW6,$SNWTT,PF$SYS,PF$IOW
        .GLOBL  RSR,TSR,LMXNUM,LSTMX,MXDTR,ZCLR,MXCSR
        .GLOBL  $INDDF,$INDRN,IN$ACT,IN$CNT,IN$CMD,INDSAV
        .GLOBL  $PHONE,INVEC,LMXLN,MXVEC,$INIT,$DEAD
        .GLOBL  ITRMTP,LMXPRM,LSW7,$CFKIL,CFSTS,CF$IND,CF$QUT
        .GLOBL  CFABLV,MONVEC,MAXPRI,MXJPRI
        .GLOBL  LOGCHN,LOGFLG,LOGPTR,LOGBUF,LOGBLK
        .GLOBL  LF$OPN,LF$WRT,UCLBLK,UCLDAT
        .GLOBL  CSHHD,FC$CDX,FC$LNK,FD$NAM,UC$NDC,UC$MDC
        .GLOBL  PLOAD,DORUN,CMDFRM,CMDDSN,STLGCN,DATTIM,PRGALL
        .GLOBL  LDCLEN,R50TSX,R50UCL,INDABT
        .GLOBL  CMDBUF,PAUMSG,RDCMD,DKSAV,SYSAV,CVTTAB,RUNHD,SEARCH
        .GLOBL  INVOPT,FKILL,ABRTCF,ACRFN,XAREA,FILNAM,NOPRG,FPRINT
        .GLOBL  PUSHCF,TRMSTR,FILNAM,R50DIR,R50SY,R50IND,R50SAV
        .GLOBL  INDACT,R50DUP,R50PIP,R50KED,R50K52,R50KEX,R50COM
        .GLOBL  BLKO,RDERM,R50VIR,NOSTRT,RUNEMT,OVRCOR
        .GLOBL  BADSAV,LDNAM,NOPRG,NOCIN,SIZVAL,ASKLNM,BADCMD,KCSIBF
        .GLOBL  ASDEX,KCSIMS,GTRD50,R50BUF,R50LDO,MNTDEV,DMTARG
        .GLOBL  DEADEV,CHKMNT,CHKMTX,INFOMT,NOFLAG,MTOPHD,INVOPT,ILLCMD
        .GLOBL  R50LD,INVLDN,R50DSK,ACRFIL,BDFNAM,LOGASN,MNTFUL,R50LD7
        .GLOBL  TBLOVF,SETHD,CSIMS2,CKPRIV,R50NO,AMBOPT,ACRDEC
        .GLOBL  MAXAVL,PRTDEC,DEVUNT,PNAME,HANIDX,HNBUF
        .GLOBL  ACROCT,HANBSY,CSIMS1,MISSEQ,NOIND
        .GLOBL  BADPMT,BADPRI,TOTXT,CRLF,HIPRI,STLGHD,LOGCLS,R50LOG
        .GLOBL  BDLGOP,SPLHLA,LDOPHD,PRTFIX,PRTSPC
        .GLOBL  DLTXT,OCTFIX,PRTTTP,NATXT,NOTXT,YESTXT,NINTXT
        .GLOBL  PRTUNM,SYHD1,SYHD2,PRTLN,SPACE2,DETTXT,SPACE3,RNMS
        .GLOBL  SWPTX,LOCKTX,SPACE5,PRTDC3,KBMSG,DIVIDE,PRTDC2
        .GLOBL  COLOO,CPUAH,CPUAL,PRTTMV,NOFIL,CMDBUF,CALUCL
        .GLOBL  NOUDC,DEVHD1,ASNHD1,ASNHD2,SHMTH1,SHMTH2
        .GLOBL  CVDVNM,SPACE6,PRTBUF,PRTFNM,NONEMS,NODAT,NOLDMT
        .GLOBL  SUBARO,EDTFIL,RONTXT,NOTAVL,KBTX,MNFLGS,MNBPC
        .GLOBL  DELSPC,MNBASE,MNTOP,MONHD,MONAR1,NOPMGN,PMBUSY,MONAR2
        .GLOBL  NSWPMS,MAXMTX,CURMTX,CHKDLM,SPLHD,INVOPT
        .GLOBL  DEVIDL,COAL,ALDEX,COAD,SPACTV,SPWFM,DEVIDL,SPSNG
        .GLOBL  COAL,ALDEX,ALDBLK,COAD,SPACTV,SPWFM,DEVIDL
        .GLOBL  SPSNG,SPFUL,SPCF,SPFLK,NOFIL,SPGEMT,NOOPTT
        .GLOBL  BDLIN,MSGBUF,MSGEND,NOTON,GAGMSG,MHNSMS
        .GLOBL  YELEMT,LINFRE,DJABMS,DLMSG,INVTIM,DMTALL
        .GLOBL  SHTMSG,AUTHFN,SPLACT,DOSTOP,OFFEMT,KILEMT,UPTMMS
        .GLOBL  TMTOTH,DIVSOR,TMTOTL,PRTPCT,SUM1,SUM2,SUM3,SUM4
        .GLOBL  SUM5,SUM6,SUM7,OTHRON,SPLPND,STPASK,SRTSMS
        .GLOBL  SIZEMT,ASNOVF,INVLDM,CSIMS4,MNTARG,HUPARG,R50TT
        .GLOBL  KMNNAM,CCLNAM,OTRMNT,CHKDEV,DMTSUB,CMDCCL,CHKTTD
        .GLOBL  SHOHD,SUBTXT,MNTTXT,SRTTXT,TOTMMS,UMSSMS,SSRMAP
        .GLOBL  TSXSMS,USRMMS,JCXSMS,DZTXT,OCTPRT,CSIMS3,OVLYEQ
        .GLOBL  PRTR50,PRTDAT,PRTTOD,PRTTIM,INVDEV,ALFN,R50DK
        .GLOBL  DETHD,DETARG,RUNMS,NOFRDL,R50MON,INVDAT,MUL32,COAF
```

```
      1                               ;
      2                               ;   Assembly parameters
      3                               ;
      4                               ;   Greeting message control option
      5                               ;   If GREET=2, full greeting message is printed.
      6                               ;   If GREET=1, truncated greeting message is printed.
      7                               ;   If GREET=0, greeting message is not printed. (SHOULD NEVER BE USED)
      8                               ;
      9         000002                GREET   =       2
     10                               ;
     11                               ;   Assembly constants
     12                               ;
     13         000012                LF      =       12              ;LINE FEED
     14         000015                CR      =       15              ;CARRIAGE RETURN
     15         000040                BLANK   =       40              ;ASCII SPACE
     16         000007                BELL    =       07              ;ASCII BELL
     17         000011                TAB     =       11              ;HORIZONTAL TAB
     18         000014                FF      =       14              ;FORM FEED
     19         000033                ESC     =       33              ;Escape character
     20         000400                BLKWDS  =       256.            ;# OF WORDS IN DISK BLOCK
     21         000016                HANCHN  =       16              ;Channel used to access handler files
     22                               ;
     23                               ;   Flags which are passed to CCL in chain location 510.
     24                               ;
     25         000001                C$TECO  =       1               ;USE TECO AS DEFAULT EDITOR
     26         000002                C$WILD  =       2               ;USE IMPLICIT WILDCARD NAMES
     27         000004                C$QUIT  =       4               ;SET TT QUIET IS IN EFFECT
     28         000010                C$TEST  =       10              ;DISPLAY CCL GENERATED COMMANDS
     29         000020                C$KED   =       20              ;DEFAULT EDITOR IS KED
     30         000040                C$K52   =       40              ;DEFAULT EDITOR IS K52
     31         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   US*E,CMDUSE
     70 000366                                      CMDDEF   INI*TIALIZE,CMDINI
     71 000372                                      CMDDEF   SQ*UEEZE,CMDSQZ
     72 000376                                      CMDDEF   LIB*RARY,CMDCCL
     73 000402                                      CMDDEF   TE*CO,CMDCCL
     74 000406                                      CMDDEF   UC*L,UCLCMD
     75 000412                                      CMDDEF   MU*NG,CMDCCL
     76 000416                                      CMDDEF   CL*OSE,RDCMD
     77 000422                                      CMDDEF   GT,RDCMD
     78 000426                                      CMDDEF   INS*TALL,CMDINS
     79 000432                                      CMDDEF   LO*AD,RDCMD
     80 000436                                      CMDDEF   SU*SPEND,CMDSPN
     81 000442                                      CMDDEF   UNL*OAD,RDCMD
     82 000446                                      CMDDEF   BO*OT,CMDBOT
     83 000452                                      CMDDEF   $ST*OP,CMDBOT
     84 000456                                      CMDDEF   $SH*UTDOWN,CMDSHT
     85 000462                                      TBLEND
```

```
                                        ;----------------------------------------------------------------
 000234  000000  000000  000000  ASDEX:  .WORD   0,0,0,0
 000242  000000
 000244  075250  012445          SYCOM:  .RAD50  /SY COM/
 000250  015270  012445          DKCOM:  .RAD50  /DK COM/
 000254  075250  073376          SYSAV:  .RAD50  /SY SAV/
 000260  015270  073376          DKSAV:  .RAD50  /DK SAV/
 000264  100040                  R50TT:  .RAD50  /TT /
 000266  015270                  R50DK:  .RAD50  /DK /
 000270  075250                  R50SY:  .RAD50  /SY /
 000272  046537                  R50LOG: .RAD50  /LOG/
 000274  016003                  R50DSK: .RAD50  /DSK/
 000276  042614                  R50KED: .RAD50  /KED/
 000300  045130                  R50K52: .RAD50  /K52/
 000302  042640                  R50KEX: .RAD50  /KEX/
 000304  062570                  R50PIP: .RAD50  /PIP/
 000306  015172                  R50DIR: .RAD50  /DIR/
 000310  016130                  R50DUP: .RAD50  /DUP/
 000312  035164                  R50IND: .RAD50  /IND/
 000314  100020                  R50TSX: .RAD50  /TSX/
 000316  101704                  R50UCL: .RAD50  /UCL/
 000320  105372                  R50VIR: .RAD50  /VIR/
 000322  045640                  R50LD:  .RAD50  /LD /
 000324  045676                  R50LD0: .RAD50  /LD0/
 000326  045705                  R50LD7: .RAD50  /LD7/
 000330  012445                  R50COM: .RAD50  /COM/
 000332  073376                  R50SAV: .RAD50  /SAV/
 000334  015270  016003          MNTDEF: .RAD50  /DK DSK/
 000340  004056                  ALDEX:  .RAD50  /ALN/
 000342  015270  004051  026760  ALDBLK: .RAD50  /DK ALIGN ALN/
 000350  004056
 000352  075250  003273  021113  AUTHFN: .RAD50  /SY ACCESSTSX/
 000360  100020
 000362  075250  100003  051646  KMNNAM: .RAD50  /SY TSKMONSAV/
 000370  073376
 000372  075250  011504  000000  CCLNAM: .RAD50  /SY CCL   SAV/
 000400  073376
 000402  075250  035164  000000  INDNAM: .RAD50  /SY IND   SAV/
 000410  073376
 000412  075250  045640  000000  LDNAM:  .RAD50  /SY LD    SYS/
 000420  075273
 000422  043327  053600          R50KMN: .RAD50  /KMON  /
 000426  075250  114730  000000  HNBUF:  .RAD50  /SY XXX   TSX/
 000434  100020
 000436  054730                  R50NO:  .RAD50  /NO /
                                 ;
 000440  000000                  DEVUNT: .WORD   0
 000442  000000                  ASKLNM: .WORD   0
 000444  000000                  INDRFL: .WORD   0
 000446  000000                  QUOTFL: .WORD   0
 000450                          FILNAM: .BLKW   5
 000462                          XAREA:  .BLKW   8.
 000502                          R50BUF: .BLKW   2
 000506                          LSTFRM: .BLKW   3
 000514  000000                  CPUAH:  .WORD   0
 000516  000000                  CPUAL:  .WORD   0
 000520  000000                  HANIDX: .WORD   0
```

```
     50 000522  000000  000000  000000  MNTDEV: .WORD   0,0,0,0
        000530  000000
     51 000532  000000                  CMDEND: .WORD   0       ;Pointer past end of current command string
     52 000534  000000  000000  000000  ALCDEV: .WORD   0,0,0,0 ;Name of dev being allocated or deallocated
        000542  000000
```

```
                                ;
                                ;  Emt arg block to run a program.
                                ;
000544     000     126          RUNEMT: .BYTE   0,126
000546  000000                          .WORD   0
                                ;
                                ;  Emt arg block to start a spooler.
                                ;
000550     000     126          SPGEMT: .BYTE   0,126
000552  000001                          .WORD   1
000554  000000                          .WORD   0
                                ;
                                ;  Emt arg block to control detached jobs.
                                ;
000556     000     132          DETARG: .BYTE   0,132
000560  000000                          .WORD   0
                                ;
                                ;  Emt argument block to set the size of the job
                                ;
000562     000     141          SIZEMT: .BYTE   0,141
000564  000000                  SIZVAL: .WORD   0
                                ;
                                ;  Emt arg block to mount a file structure.
                                ;
000566     000     134          MNTARG: .BYTE   0,134
000570  000522'                         .WORD   MNTDEV
                                ;
                                ;  Emt arg block to dismount a file structure.
                                ;
000572     000     135          DMTARG: .BYTE   0,135
000574  000522'                         .WORD   MNTDEV
                                ;
                                ;  Emt arg block to clean out all entries in directory cache
                                ;
000576     001     135          CSHCLN: .BYTE   1,135
                                ;
                                ;  Emt arg block to copy the file status and privileges from a parent job
                                ;
000600     001     160          CPYCXT: .BYTE   1,160
000602  000000                          .WORD   0
                                ;
                                ;  Emt arg block to send a message to a line.
                                ;
000604     000     127          YELEMT: .BYTE   0,127
000606  000000                          .WORD   0
000610  000612'                         .WORD   MSGBUF
000612                          MSGBUF: .BLKB   84.
000736                          MSGEND:
                                ;
                                ;  Emt to allocate a device
                                ;
000736     000     156          ALDEMT: .BYTE   0,156
000740  000534'                         .WORD   ALCDEV          ;Pointer to device spec
                                ;
                                ;  Emt to deallocate a device
                                ;
000742     001     156          DLCEMT: .BYTE   1,156
```

```
000744  000534'                         .WORD   ALCDEV
                                ;
                                ;  Emt to determine if a device is allocated to another user
                                ;
000746     002     156          TALEMT: .BYTE   2,156
000750  000534'                         .WORD   ALCDEV
                                ;
                                ;  Emt argument block to read a cached device descriptor block into CDBUF
                                ;
000752     000     126          CDGEMT: .BYTE   0,126
000754  000010                          .WORD   10                      ;Subfunction code (peek)
000756  000000                  CDGADR: .WORD   0                       ;Address of block within kernel
000760  000000G                         .WORD   CD$$SZ                  ;Amt of data to get
000762  003402'                         .WORD   CDBUF                   ;Destination buffer
                                ;
                                ;  Emt argument block to copy a cached device descriptor block from CDBUF
                                ;  back into the kernel data area.
                                ;
000764     000     126          CDPEMT: .BYTE   0,126
000766  000011                          .WORD   11                      ;Subfunction code (poke)
000770  000000                  CDPADR: .WORD   0                       ;Address of block within kernel
000772  000000G                         .WORD   CD$$SZ
000774  003402'                         .WORD   CDBUF                   ;Destination buffer
                                ;
                                ;  Emt argument block to move INSTALL entry from kernel into IIBUF
                                ;
000776     000     126          INGEMT: .BYTE   0,126
001000  000010                          .WORD   10                      ;Subfunction code (peek)
001002  000000                  INGADR: .WORD   0                       ;Address of block within kernel
001004  000000G                         .WORD   II$$SZ                  ;Amt of data to get
001006  000060'                         .WORD   IIBUF                   ;Destination buffer
                                ;
                                ;  Emt argument block to store INSTALL entry into kernel data
                                ;
001010     000     126          INPEMT: .BYTE   0,126
001012  000011                          .WORD   11                      ;Subfunction code (poke)
001014  000000                  INPADR: .WORD   0                       ;Address of block within kernel
001016  000000G                         .WORD   II$$SZ                  ;Amt of data to move
001020  000060'                         .WORD   IIBUF                   ;Source buffer
                                ;
                                ;  Emt arg block to log off the current job.
                                ;
001022     000     126          OFFEMT: .BYTE   0,126
001024  000002                          .WORD   2
001026  000000                          .WORD   0                       ;0==>Default time to drop DTR
                                ;
                                ;  Emt to update running copy of a handler.
                                ;
001030     000     126          HUPARG: .BYTE   0,126
001032  000003                          .WORD   3
001034                                  .BLKW   2
                                ;
                                ;  Emt to reboot the system.
                                ;
001040     000     126          BOTEMT: .BYTE   0,126
001042  000004                          .WORD   4
001044  004362'                         .WORD   START
```

```
                                        ;
                                        ;   Emt to force logoff of a job.
                                        ;
    001046       000      126           KILEMT: .BYTE   0,126
    001050    000005                            .WORD   5
    001052    000000                            .WORD   0
                                        ;
                                        ;   Emt to suspend execution of a job
                                        ;
    001054       000      126           SJEMT:  .BYTE   0,126
    001056    000012                            .WORD   12
    001060    000000                            .WORD   0
                                        ;
                                        ;   Emt to resume execution of a job
                                        ;
    001062       001      126           RJEMT:  .BYTE   1,126
    001064    000012                            .WORD   12
    001066    000000                            .WORD   0
                                        ;
                                        ;   EMT arg block to set "hold" mode for a spooled log file
                                        ;
    001070       000G     151           SPLHLA: .BYTE   LOGCHN,151
    001072    000000                            .WORD   0
    001074    000001                            .WORD   1
                                        ;
                                        ;   EMT arg block to copy data from the context block of another job
                                        ;   into our context block.
                                        ;
    001076       000      126           EMCXCP: .BYTE   0,126
    001100    000014                            .WORD   14
    001102    000000                            .WORD   0               ;# of job we are copying from
    001104    000000                            .WORD   0               ;Address of item being copied
    001106    000000                            .WORD   0               ;Number of bytes to copy
                                        ;
                                        ;   EMT arg block to set the job execution priority
                                        ;
    001110       000      150           PRIEMT: .BYTE   0,150
    001112    000000                            .WORD   0
                                        ;
                                        ;   EMT arg block to create a window for this job
                                        ;
    001114       000      161           MAKWIN: .BYTE   0,161
    001116       001      001                   .BYTE   1,1
    001120       120      020                   .BYTE   80.,16.
    001122       001      000                   .BYTE   1,0
    001124    000000                            .WORD   0
                                        ;
                                        ;   EMT arg block to select window 1 as the current window
                                        ;
    001126       001      161           MAPWIN: .BYTE   1,161
    001130       001      000                   .BYTE   1,0
                                        ;
                                        ;   EMT arg block to delete all temporary display windows for the job
                                        ;
    001132       002      161           DELWIN: .BYTE   2,161
    001134       000      000                   .BYTE   0,0
                                        ;
```

```
                                   ;  Emt arg block to initiate performance monitoring.
                                   ;
001136     000      136            MONAR1: .BYTE   0,136
001140  000000                     MNBASE: .WORD   0
001142  000000                     MNTOP:  .WORD   0
001144  000000                     MNBPC:  .WORD   0
001146  000000                     MNFLGS: .WORD   0
                                   ;
                                   ;  Emt arg block to start performance monitoring.
                                   ;
001150     001      136            MONAR2: .BYTE   1,136
                                   ;
                                   ;  Emt arg block to broadcast job status change info to monitoring jobs
                                   ;
001152     002      157            GENMON: .BYTE   2,157
001154  000000                             .WORD   0               ;Store válue to broadcase here
```

```
001156                          CMDBUF: .BLKW   66.
001362                          CBFEND:                         ;END OF CMDBUF
001362                          KEYBUF: .BLKW   8.              ;Used by SEARCH to hold keywords
001402                          KEYEND:                         ;End of KEYBUF
001402                          BLKO:   .BLKW   512.
       001402'                  KCSIBF  =       BLKO            ;OVERLAY WITH BLKO
       001402'                  INBUF   =       BLKO            ;OVERLAY WITH BLKO
003402                          CDBUF:  .BLKW   10.             ;Should be at least as large as CD$$SZ
                                ;   TSKMON STACK
003426                                  .BLKW   100.
       003736'                  KMSTK   =       .
                                ;
                                ;   BYTE DATA
                                ;
                                ;   PRINT BUFFER
003736                          PRTBUF: .BLKB   256.
004336     200                  PBUFND: .BYTE   200
004337     000                  NOFLAG: .BYTE   0
004340     000                  DOLRAT: .BYTE   0               ;INDICATES $@ SCANNED
004341     000                  DOTAT:  .BYTE   0               ;INDICATES #@ SCANNED
004342     000                  COLEQL: .BYTE   0               ;INDICATES := SCANNED
004343     000                  NOUCL:  .BYTE   0               ;Non-zero==>Don't call UCL for this command
004344     000                  NUMMNT: .BYTE   0               ;Counts number of mounted devices
004345     123    131    072    SYTXT:  .ASCIZ  /SY:/
004350     000
004351     123    131    072    SYINTX: .ASCIZ  /SY:IND /
004354     111    116    104
004357     040    000
                                ;
                                        .EVEN
                                ;
```

```
                                          .SBTTL  Job Initialization
                                  ;---------------------------------------------
                                  ;  ENTER TSKMON AT KMNSTR.
                                  ;
004362  000240                    START:  NOP                     ;DEBUGGING BPT INSERTION POINT
004364  116701  000000G                   MOVB    CORUSR,R1       ;GET USER INDEX #
                                  ;
                                  ;  INITIALIZATION PERFORMED FOR LINE THE FIRST TIME
                                  ;  TSKMON IS ENTERED.
                                  ;
004370  032761  000000G 000000G           BIT     #$KINIT,LSW(R1) ;IS THIS 1ST ENTRY FOR THIS LINE?
004376  001402                            BEQ     10$             ;BR IF YES
004400  000167  001026                    JMP     KMNOT1
                                  ;
                                  ;  Make sure TSGEN hasn't been modified without relinking TSKMON.
                                  ;
004404  026727  000000G 123456    10$:    CMP     GENTOP,#123456  ;DOES THIS LOOK LIKE THE TOP OF TSGEN?
004412  001406                            BEQ     22$             ;BR IF OK
004414                                    FERR    #REGEN          ;NEED TO RELINK TSKMON
004430  016761  000000G 000000G   22$:    MOV     MINTIM,LCONTM(R1);SET JOB LOG-ON TIME
004436  042761  000000G 000000G           BIC     #$CFKIL,LSW6(R1) ;IN CASE OF ^C^C DURING GETSYP
004444  042761  000000G 000000G           BIC     #$CTRLC,LSW(R1) ;DO NOT DO CONTROL-C ABORT
                                  ;
                                  ;  Initially set the job name to all blanks
                                  ;
004452  010105                            MOV     R1,R5           ;GET JOB INDEX #
004454  070527  000006                    MUL     #6,R5           ;* 12. BYTES PER ENTRY
004460  062705  000000G                   ADD     #LUNAME,R5      ;POINT TO USER NAME ENTRY FOR THIS LINE
004464  012704  000014                    MOV     #12.,R4
004470  112725  000040            24$:    MOVB    #' ,(R5)+       ;SET USER NAME TO BLANKS
004474  077403                            SOB     R4,24$
                                  ;
                                  ;  Initialize cells in job context block
                                  ;
004476  012767  000000G 000000G           MOV     #CFSTK,CFSP     ;COMMAND FILE STACK
004504  016700  000000G                   MOV     DFJMEM,R0       ;GET MAX # KB JOB IS ALLOWED TO USE
004510  072027  000012                    ASH     #10.,R0         ;CONVERT TO ADDRESS
004514  001002                            BNE     25$             ;BR IF DIDN'T OVERFLOW 64KB
004516  012700  177774                    MOV     #177774,R0      ;SET MAX MEMORY AS 64KB
004522  010067  000000G           25$:    MOV     R0,MAXMEM       ;SET AS TOP OF MEMORY FOR JOB
004526  012767  000000G 000000G           MOV     #NAMTOP,UHIMEM  ;HIGHEST LEGAL ADDRESS FOR JOB
004534  012767  000040G 000000G           MOV     #SPUBUF+4,SXBPNT;SPOOL FILE BUFFER
004542  112767  000000G 000000G           MOVB    #MAXPRI,MXJPRI  ;SET MAXIMUM JOB PRIORITY
004550  112767  000056  000000G           MOVB    #'.,KMPRMT      ;SET UP DEFAULT KMON PROMPT (".")
004556  112767  000200  000001G           MOVB    #200,KMPRMT+1   ;TERMINATE PROMPT STRING
004564  012703  000000G                   MOV     #STDNAM,R3      ;SET UP CURRENT (DEFAULT) SPOOL FORM NAME
004570  012704  000000G                   MOV     #UFORM,R4
004574  012700  000006                    MOV     #6,R0
004600  112324                    3$:     MOVB    (R3)+,(R4)+
004602  077002                            SOB     R0,3$
004604  112767  000000G 000000G           MOVB    #SC$ERR,ERRSEV  ;ERROR ABORT SEVERITY LEVEL
                                  ;
                                  ;  Start job with all privileges granted.
                                  ;  Privileges may be changed by running LOGON or by use of
                                  ;  the SET PROCESS/AUTH/PRIV=(list)  command.
                                  ;
004612  012700  000000G                   MOV     #PVNPW,R0       ;Get # words in each privilege vector
```

```
004616  005002                          CLR     R2              ;Set index for 1st word
004620  012703  177777                  MOV     #177777,R3      ;Get all privilege flags turned on
004624  010362  000000G         77$:    MOV     R3,PRIVA0(R2)   ;Authorized privileges
004630  010362  000000G                 MOV     R3,PRIVS0(R2)   ;Set privileges
004634  010362  000000G                 MOV     R3,PRIVF0(R2)   ;Command file privileges
004640  010362  000000G                 MOV     R3,PRIVC0(R2)   ;Current (program) privileges
004644  062702  000002                  ADD     #2,R2           ;Increment vector index
004650  077013                          SOB     R0,77$          ;Set all privileges
004652  020127  000000G                 CMP     R1,#LSTPL       ;Is this a primary line?
004656  003016                          BGT     32$             ;Br if not
004660  032761  000000G 000000G         BIT     #$NOVLN,JLSW2(R1);Disallow subprocess usage?
004666  001412                          BEQ     32$             ;Br if don't need to disallow subprocesses
004670  012700  000000G                 MOV     #P2$VIR,R0      ;Get subprocess privilege flag
004674  040067  000000G                 BIC     R0,PRIVA2       ;Remove subprocess privilege
004700  040067  000000G                 BIC     R0,PRIVS2
004704  040067  000000G                 BIC     R0,PRIVF0
004710  040067  000000G                 BIC     R0,PRIVC0
004714  004767  000000G         32$:    CALL    RSTPRV          ;Setup some other privilege flags
                                ;
                                ;  Open spool file channel
                                ;
004720  105767  000000G                 TSTB    NSPLDV          ;ARE THERE ANY SPOOLED DEVICES?
004724  001410                          BEQ     28$             ;BR IF NOT
004726                                  .REOPEN #XAREA,#USPLCH,#SPLCHN ;OPEN CHANNEL FOR WRITES TO SPOOL FILE
                                ;
                                ;  Initialize TSXUCL data base for this job
                                ;  (Say that there are no user-defined commands yet)
                                ;
004746  016705  000000G         28$:    MOV     UCLBLK,R5       ;Get # blocks in file per job
004752  001446                          BEQ     2$              ;Br if TSXUCL data file not needed
004754                                  .LOOKUP #XAREA,#1,#UCLDAT ;Lookup TSXUCL data file
004774  103435                          BCS     2$              ;Br if not there
004776  010100                          MOV     R1,R0           ;Get job index number
005000  006200                          ASR     R0              ;Convert to job number
005002  005300                          DEC     R0              ;Make first job # 0
005004  070500                          MUL     R0,R5           ;Compute block number of data for this job
005006  012702  001402'                 MOV     #BLK0,R2        ;Point to data buffer
005012  012762  177777  000000G         MOV     #-1,UC$NDC(R2)  ;Set flag saying TSXUCL should init data
005020  016762  000000G 000000G         MOV     VUCLMC,UC$MDC(R2) ;Set maximum allowed number of commands
005026                                  .WRITW  #XAREA,#1,R2,#256.,R5 ;Write out data for job
005062                                  .CLOSE  #1              ;Close TSXUCL file
                                ;
                                ;  Set default editor
                                ;
005070  126727  000000G 000000G 2$:     CMPB    VEDIT,#EDIT     ;IS DEFAULT EDITOR EDIT?
005076  001413                          BEQ     7$              ;BR IF YES
005100  126727  000000G 000000G         CMPB    VEDIT,#TECO     ;IS DEFAULT EDITOR TECO?
005106  001004                          BNE     21$             ;BR IF NOT
005110  052761  000000G 000000G         BIS     #$TECO,LSW5(R1) ;SET DEFAULT EDITOR FOR JOB
005116  000403                          BR      7$
005120  052761  000000G 000000G 21$:    BIS     #$KED,LSW5(R1)  ;DEFAULT EDITOR MUST BE KED
                                ;
                                ;  Set default UCL FIRST/MIDDLE/LAST
                                ;
005126  126727  000000G 000001  7$:     CMPB    VUCLOR,#1       ;SET UCL FIRST?
005134  001003                          BNE     30$             ;BR IF NOT
005136  052761  000000G 000000G         BIS     #$UCLCF,LSW7(R1)
```

```
 005144  126727  000000G 000002  30$:    CMPB    VUCLOR,#2          ;SET UCL MIDDLE?
 005152  001003                          BNE     29$                ;BR IF NOT
 005154  052761  000000G 000000G         BIS     #$UCLCM,LSW7(R1)
 005162  126727  000000G 000003  29$:    CMPB    VUCLOR,#3          ;SET UCL LAST?
 005170  001003                          BNE     27$                ;BR IF NOT
 005172  052761  000000G 000000G         BIS     #$UCLCL,LSW7(R1)
                                 ;
                                 ;  Set default SL mode
                                 ;
 005200  052761  000000G 000000G 27$:    BIS     #$SLLET,LSW7(R1);Default SL substitution on
                                 ;
                                 ;  Set default wildcard flag
                                 ;
 005206  005727  000000G         31$:    TST     #WILDFL            ;EXPLICIT OR IMPLICIT WILDCARDS?
 005212  001403                          BEQ     8$                 ;BR IF EXPLICIT
 005214  052761  000000G 000000G         BIS     #$WILD,LSW5(R1)    ;SET IMPLICIT WILDCARD FLAG
 005222  020127  000000G         8$:     CMP     R1,#LSTDL          ;REAL OR VIRTUAL LINE?
 005226  003022                          BGT     12$                ;BR IF VIRTUAL
 005230  020127  000000G                 CMP     R1,#LSTPL          ;REAL OR DETACHED?
 005234  003021                          BGT     1$                 ;BR DETACHED
                                 ;
                                 ;  See if we need to get a system password (primary line only).
                                 ;
 005236  032761  000000G 000000G         BIT     #$SYSPS,LSW2(R1);Do we need to accept a system password?
 005244  001402                          BEQ     13$                ;Br if not
 005246  004767  000000G                 CALL    GETSYP             ;Accept system password
                                 ;
                                 ;  Mount SY device for this job (primary line only).
                                 ;
 005252  016767  173012  173242  13$:    MOV     R50SY,MNTDEV       ;SET SY AS DEVICE TO MOUNT
 005260  012700  000566'                 MOV     #MNTARG,R0         ;MOUNT SY
 005264  104375                          EMT     375
                                 ;
                                 ;  Terminal dependent initialization for primary lines.
                                 ;
 005266  004767  004702                  CALL    TRMINI             ;Do terminal-dependent initialization
 005272  000402                          BR      1$
                                 ;
                                 ;  Initialization for subprocesses only.
                                 ;  Print n> to show which subprocess we are switching to.
                                 ;
 005274  004767  005156          12$:    CALL    VIRINI             ;Do virtual line initialization
                                 ;
                                 ;  Do some initialization for detached jobs
                                 ;
 005300  032761  000000G 000000G 1$:     BIT     #$DETCH,LSW(R1) ;Is this a detached job?
 005306  001403                          BEQ     GRTINI             ;Br if not
 005310  012761  000000C 000000G         MOV     #$ECHO!$LC!$DEFER,LSW2(R1) ;Init control flags
                                 ;
                                 ;  Print TSX greeting message
                                 ;
 005316  004767  005774          GRTINI: CALL    PRTGRT             ;Print the TSX-Plus logon greeting
 005322  016161  000000G 000000G         MOV     LSW2(R1),LSW2S(R1) ;SAVE IN CASE OF CTRL-C REENTRY
                                 ;
                                 ;  If this job has a parent job, copy file context and privilege from parent
                                 ;
 005330  004767  005254                  CALL    CPYPRN             ;Copy info from parent job
```

```
                                        ;
                                        ;  Set flag saying job initialization has been done
                                        ;
005334  052761  000000G 000000G                 BIS     #$KINIT,LSW(R1) ;SAY LINE HAS BEEN INITIALIZED
                                        ;
                                        ;  See if this line has an associated start-up command file.
                                        ;
005342  020127  000000G                         CMP     R1,#LSTDL       ;Is this a virtual line?
005346  101004                                  BHI     1$              ;Br if yes
005350  016102  000000G                         MOV     LSUCF(R1),R2    ;IS THERE A START-UP COMMAND FILE?
005354  001407                                  BEQ     61$             ;BR IF NOT
005356  000402                                  BR      2$
005360  012702  000000G                 1$:     MOV     #SBPSUF,R2      ;Point to job context cell with file name
005364  105712                          2$:     TSTB    (R2)            ;IS FILE NAME NULL?
005366  001402                                  BEQ     61$             ;BR IF YES
005370  004767  006324                          CALL    SETSUF          ;Set up start-up command file for execution
                                        ;
                                        ;  Broadcast status message to monitoring jobs telling them that
                                        ;  this job logged on.
                                        ;
005374  012767  000000G 173552          61$:    MOV     #JS$ON,GENMON+2 ;Set logged-on status code
005402  012700  001152'                         MOV     #GENMON,R0      ;Point to EMT argument block
005406  104375                                  EMT     375             ;Tell monitoring jobs that we are initiated
005410  005761  000000G                         TST     LPARNT(R1)      ;Do we have a parent job?
005414  001406                                  BEQ     KMNOT1          ;Br if not
005416  012767  000000G 173530                  MOV     #JS$LOG,GENMON+2;Set status code saying we have logged on
005424  012700  001152'                         MOV     #GENMON,R0      ;Point to EMT argument block
005430  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.
                                         ;
 005432  016761  172764  000000G KMNOT1: MOV     R50KMN,LPRG1(R1);SET "KMON" AS RUNNING PROGRAM NAME
 005440  016761  172760  000000G         MOV     R50KMN+2,LPRG2(R1)
 005446  005067  000000G                 CLR     UTRPAD          ;CLEAR USER TRAP CONTROL
 005452  005067  000000G                 CLR     UFPTRP          ;RESET FLOATING POINT TRAP CONTROL
 005456  005061  000000G                 CLR     LSCCA(R1)       ;RESET .SCCA TRAP CONTROL
 005462  105067  000000G                 CLRB    SERFLG          ;DO .HERR
 005466  105067  000000G                 CLRB    RUNARG          ;No argument string from RUN command
 005472  005767  000000G                 TST     UCHAN           ;DID USER DO A .CDFN?
 005476  001404                          BEQ     4$              ;BR IF NOT
 005500  005067  000000G                 CLR     UCHAN           ;UNDO THE .CDFN
 005504  004767  000000G                 CALL    PRGALL          ;PURGE ALL CHANNELS
 005510  005061  000000G         4$:     CLR     LTSCMD(R1)      ;NO PENDING SPECIAL COMMAND
 005514  042761  000000G 000000G         BIC     #$SETRN,LSW9(R1);Say SETUP is no longer running
 005522                                  .CLOSE  #RUNCHN         ;CLOSE PROGRAM SAV FILE
 005530  004767  000000G                 CALL    RSTPRV          ;Reset job privileges
 005534  032761  000000G 000000G         BIT     #$DUPRN,LSW6(R1);WAS DUP THE PROGRAM THAT EXITED?
 005542  001407                          BEQ     7$              ;BR IF NOT
 005544  004767  000000G                 CALL    PRGALL          ;PURGE ALL CHANNELS
 005550  004767  000000G                 CALL    LDCLEN          ;DO "LD CLEAN" OPERATION
 005554  042761  000000G 000000G         BIC     #$DUPRN,LSW6(R1);SAY DUP IS NO LONGER RUNNING
 005562  032761  000000G 000000G 7$:     BIT     #$DOOFF,LSW(R1) ;SHOULD WE LOG USER OFF?
 005570  001402                          BEQ     6$              ;BRANCH IF NOT
 005572  000167  000000G                 JMP     CMDOFF          ;FORCE LOGOFF
                                         ;  CHECK FOR SYSTEM ABORT
 005576  116702  000000G         6$:     MOVB    ABRTCD,R2       ;WAS USER ABORTED?
 005602  001467                          BEQ     NOABRT          ;BRANCH IF NOT
                                         ;
                                         ;  USER WAS ABORTED -- PUT OUT ABORT MESSAGE.
                                         ;
 005604  112767  000010  000000G         MOVB    #10,UERSEV      ;SAY ERROR SEVERITY LEVEL = SEVERE
 005612  006302                          ASL     R2              ;CVT ERROR CODE TO WORD INDEX
 005614  016203  000176'                 MOV     ABTMSG(R2),R3   ;GET ADDR OF ERROR MESSAGE
 005620                                  .PRINT  #MONTXT
 005626                                  .PRINT  R3
                                         ;  Print name of file that caused the error
 005632  005767  000000G                 TST     ERRSPC          ;Do we have a file spec to print?
 005636  001414                          BEQ     1$              ;Br if not
 005640                                  .PRINT  #FILERM         ;Print heading message
 005646  012703  001402'                 MOV     #BLKO,R3        ;Point to buffer where result is to be stored
 005652  012704  000000G                 MOV     #ERRSPC,R4      ;Point to RAD50 file spec
 005656  004767  000000G                 CALL    EDTFIL          ;Convert file spec to ascii
 005662                                  .PRINT  #BLKO           ;Print the file spec
                                         ;  PUT OUT LOCATION OF ABORTED INSTRUCTION.
 005670                          1$:     .PRINT  #LOCMSG
 005676  016702  000000G                 MOV     ABRTAD,R2       ;GET ADDR OF ABORT INST
 005702  004767  000000G                 CALL    OCTPRT          ;PRINT THE OCTAL VALUE
 005706  020227  120000                  CMP     R2,#120000      ;IN OVERLAY REGION?
 005712  000407                          BR      10$             ;*** skip overlay this version ***
                                         ;       BLOS    10$             ;BR IF NOT
 005714                                  .PRINT  #OVLYEQ         ;PRINT "  OVERLAY = "
 005722  016700  000000G                 MOV     ABRTOV,R0       ;GET OVERLAY NAME
 005726  004767  000000G                 CALL    PRTR50          ;AND DISPLAY IT
```

```
     58 005732                              10$:    .PRINT  #CRLF
     59 005740  005067  000000G                     CLR     ABRTAD          ;CLEAN ABORT ADDR
     60 005744  105067  000000G                     CLRB    ABRTCD          ;AND CODE
     61 005750  042737  004440  000000G             BIC     #004440,@#JSWLOC;RELEASE CHAIN AREA, PASLIN, AND SCHAIN
     62 005756  105067  000000G                     CLRB    CINFLG          ;KILL ANY CHAIN REQUEST
```

```
                                        ;
                                        ;  User did not abort.  See if this is a .CHAIN request
                                        ;
 005762  005067  000000G                NOABRT: CLR     ERRSPC             ;CLEAR ANY ERROR FILE SPEC
 005766  032761  000000C 000000G                BIT     #$CFKIL!$CFABT,LSW6(R1) ;REQUEST TO ABORT ALL COM FILES?
 005774  001405                                 BEQ     1$                 ;BR IF NOT
 005776  105067  000000G                        CLRB    CINFLG             ;CLEAR ANY CHAIN REQUEST
 006002  042737  000000C 000000G                BIC     #PASLIN!SCHAIN,@#JSWLOC ;KILL ANY PASSED COMMAND
 006010  105767  000000G                1$:     TSTB    CINFLG             ;IS THIS .CHAIN REQUEST?
 006014  001402                                 BEQ     NOCIN              ;BRANCH IF NOT
 006016  000167  000000G                        JMP     KDOCIN             ;GO DO THE CHAIN
                                        ;
                                        ;  Not .CHAIN request so enter normal KMON.
                                        ;
                                        ;  Delete any temporary display windows for this job
                                        ;
 006022  012700  001132'                NOCIN:  MOV     #DELWIN,R0         ;Point to EMT argument block
 006026  104375                                 EMT     375                ;Delete all temporary windows for job
                                        ;
                                        ;  Clean up various cells
                                        ;
 006030  012761  000040  000000G                MOV     #BLANK,LRBFIL(R1) ;USE SPACE FOR RUBOUT FOR KMON
 006036  016161  000000G 000000G                MOV     LSW2S(R1),LSW2(R1);RESET STATUS OF LSW2
 006044  042761  000000G 000000G                BIC     #KL4CLR,LSW4(R1)
 006052  042761  000000G 000000G                BIC     #$VTESC,LSW5(R1);TURN OFF VT52 ESC-LETTER ACTIVATION
 006060  042761  000000C 000000G                BIC     #$DEBUG!$RNIOP,LSW9(R1);Not debugger, I/O page
 006066  005061  000000G                        CLR     LNSPAC(R1)         ;NO SPECIAL ACTIVATION CHARS
 006072  005061  000000G                        CLR     LRDTIM(R1)         ;CLEAR TT READ TIME-OUT VALUE
 006076  005061  000000G                        CLR     LAFSIZ(R1)         ;CLEAR FIELD WIDTH ACTIVATION
 006102  005061  000000G                        CLR     LFWLIM(R1)         ;CLEAR FIELD WIDTH LIMIT
 006106  042761  000000G 000000G                BIC     #KL3CLR,LSW3(R1);CLEAR MISC BITS IN LSW3
 006114  042761  000000G 000000G                BIC     #$CFDCC,LSW4(R1);CLEAR DEFERRED-CTRL-C FLAG
 006122  042761  000000C 000000G                BIC     #<$NOWTT!$CHACT>,LSW5(R1);TURN OFF NO WAIT AND SINGLE CHAR ACT
 006130  042761  000000G 000000G                BIC     #$NOINT,LSW7(R1);Reset non-interactive run switch
 006136  042761  000000G 000000G                BIC     #$NTGCC,LSW9(R1);Clear non-terminating .GTLIN ctrl-C flag
 006144  032761  000000G 000000G                BIT     #$SNWTT,LSW6(R1);DID USER DO "SET TT NOWAIT"?
 006152  001403                                 BEQ     3$                 ;BR IF NOT
 006154  052761  000000G 000000G                BIS     #$NOWTT,LSW5(R1);SET NO-WAIT FLAG
 006162  016700  000000G                3$:     MOV     CFSPND,R0          ;WAS A COMMAND FILE SUSPENDED?
 006166  001404                                 BEQ     4$                 ;BR IF NOT
 006170  004767  000000G                        CALL    CFSTRT             ;Restart it
 006174  005067  000000G                        CLR     CFSPND
                                        ;
                                        ;  Check if program specified an error severity level.
                                        ;
 006200  156767  000000G 000000G 4$:            BISB    UERSEV,INDERR      ;SAVE ERROR STATUS FOR IND
 006206  126767  000000G 000000G                CMPB    UERSEV,ERRSEV      ;DID PROGRAM SPECIFY ERROR SEVERITY?
 006214  103413                                 BLO     1$                 ;BR IF OK
 006216  004767  000000G                        CALL    ABRTCF             ;SEVER ERROR -- ABORT COMMAND FILE
 006222  042761  000000G 000000G                BIC     #$CCLRN,LSW5(R1);SAY CCL NOT RUNNING
 006230  032761  000000G 000000G                BIT     #$INDAB,LSW7(R1);DOES HE WANT TO ABORT IND COMMAND FILES?
 006236  001402                                 BEQ     1$                 ;BR IF NOT
 006240  004767  000000G                        CALL    INDABT             ;Abort IND execution
 006244  105067  000000G                1$:     CLRB    UERSEV             ;CLEAR ERROR STATUS
                                        ;
                                        ;  Purge all of user's channels
                                        ;
 006250  004767  000000G                 CALL    PRGALL           ;PURGE ALL CHANNELS
 006254  032761  000000G 000000G         BIT     #$CFKIL,LSW6(R1);Should we abort IND and command files?
 006262  001403                          BEQ     2$               ;Br if not
 006264  004767  000000G                 CALL    INDABT           ;Abort IND and nested command files
 006270  000406                          BR      5$
 006272  032761  000000G 000000G 2$:     BIT     #$CFABT,LSW6(R1);SHOULD WE ABORT ALL ACTIVE COMMAND FILES?
 006300  001405                          BEQ     6$               ;BR IF NOT
 006302  004767  000000G                 CALL    ABRTCF           ;ABORT ALL ACTIVE COMMAND FILES
 006306  042761  000000G 000000G 5$:     BIC     #$CCLRN,LSW5(R1);STOP EXECUTION OF CCL
                                 ;
                                 ;   Tell any jobs that are monitoring us that we just entered TSKMON
                                 ;
 006314  005761  000000G         6$:     TST     LMONHD(R1)       ;Are we being monitored?
 006320  001003                          BNE     7$               ;Br if yes
 006322  005767  000000G                 TST     SMONHD           ;Anyone monitoring all jobs?
 006326  001406                          BEQ     CKPASL           ;Br if not
 006330  012700  001152'         7$:     MOV     #GENMON,R0       ;Point to EMT argument block
 006334  012760  000000G 000002          MOV     #JS$KMN,2(R0)    ;Set status code
 006342  104375                          EMT     375              ;Broadcast status message
```

```
                                        ;
                                        ;  See if program passed a set of command lines to Kmon
                                        ;  when it exited.
                                        ;
006344  032737  000000C 000000G CKPASL: BIT     #PASLIN!SCHAIN,@#JSWLOC;DID PROGRAM PASS US A COMMAND?
006352  001546                          BEQ     4$                ;BR IF NOT
                                        ;
                                        ;  Program did pass a set of commands to Kmon.
                                        ;  Set it up to look like a fake command file.
                                        ;  That is, the commands are stored in the command file buffer so that
                                        ;  they are read as if they came from a command file, but no actual
                                        ;  file is open.
                                        ;
                                        ;  Determine if we should abort the currently open command file.
                                        ;
006354  032761  000000C 000000G         BIT     #$CCLRN!$INDRN,LSW5(R1);IS CCL OR IND RUNNING?
006362  001006                          BNE     5$                ;BR IF YES
006364  032737  000000G 000000G         BIT     #SCHAIN,@#JSWLOC;SHOULD BE ABORT CURRENT COMMAND FILE?
006372  001002                          BNE     5$                ;BR IF NOT
006374  004767  000000G                 CALL    ABRTCF            ;ABORT ALL CURRENTLY OPEN COMMAND FILES
006400  005767  000010G         5$:     TST     CINDAT+10         ;DID HE PASS US A NULL COMMAND FILE?
006404  001531                          BEQ     4$                ;IF YES THEN WE ARE FINISHED
                                        ;
                                        ;  If command line is being passed to us by the TSXUCL program,
                                        ;  check to see if this is a case where TSXUCL could not recognize
                                        ;  the command and is passing it back to us for processing.
                                        ;  If so, just move the command to the command buffer and then proceed
                                        ;  with normal KMON command checking.
                                        ;
006406  032761  000000G 000000G         BIT     #$UCLRN,LSW7(R1);Is TSXUCL program running?
006414  001432                          BEQ     15$               ;Br if not
006416  126727  000012G 000077          CMPB    CINDAT+12,#'?     ;Is TSXUCL throwing command back to us?
006424  001026                          BNE     15$               ;Br if not
006426  012703  000013G                 MOV     #CINDAT+13,R3     ;Point to chain data area (past "?" char)
006432  012702  001156'                 MOV     #CMDBUF,R2        ;Point to command buffer
006436  112322                  16$:    MOVB    (R3)+,(R2)+       ;Move command to command buffer
006440  001376                          BNE     16$               ;Loop till asciz null moved
006442  005302                          DEC     R2                ;Make R2 point to null at end of command
006444  010267  172062                  MOV     R2,CMDEND         ;Save pointer to end of command
006450  042761  000000G 000000G         BIC     #$UCLRN,LSW7(R1);Say TSXUCL program no longer running
006456  042737  000000C 000000G         BIC     #PASLIN!SCHAIN,@#JSWLOC ;Clear command-passed flag
006464  052737  000000G 000000G         BIS     #LCBIT,@#JSWLOC   ;Enable lower-case input
006472  105267  175645                  INCB    NOUCL             ;Set flag saying not to call TSXUCL again
006476  000167  001762                  JMP     IDNCMD            ;Go process the command
                                        ;
                                        ;  If we have pending commands in the command file buffer, compress
                                        ;  them to make room for new commands.
                                        ;  If input is coming from a real command file we do not need to
                                        ;  compress since we will reread buffer when we hit end of new commands.
                                        ;
006502  032761  000000G 000000G 15$:    BIT     #$CFOPN,LSW4(R1);IS A COMMAND FILE OPEN?
006510  001403                          BEQ     11$               ;BR IF NOT
006512  012705  001000G                 MOV     #CFBUF+512.,R5    ;SAY ENTIRE COMMAND FILE BUFFER IS FREE
006516  000403                          BR      12$
006520  004767  000000G         11$:    CALL    CFSQEZ            ;COMPRESS INFO IN CURRENT COMMAND FILE BUFFER
006524  010005                          MOV     R0,R5             ;SAVE ADDRESS OF END OF FREE SPACE IN BUFFER
006526  004767  000000G         12$:    CALL    PUSHCF            ;OPEN A NEW COMMAND FILE (PUSH CURRENT ONE)
```

```
                                         ;
                                         ;  Move command line from chain area to command file buffer
                                         ;
 006532  012703  0000100                         MOV     #CINDAT+10,R3    ;POINT TO CELL WITH CHAR COUNT
 006536  012302                                  MOV     (R3)+,R2         ;GET COUNT OF # CHARS IN COMMAND
 006540  020227  000270                          CMP     R2,#<1000-510>   ;CHECK LENGTH OF COMMAND STRING
 006544  003407                                  BLE     6$               ;BR IF COMMAND IS SMALL ENOUGH
 006546  042737  000000C 000000G 10$:            BIC     #PASLIN!SCHAIN,@#JSWLOC ;CLEAR FLAGS BEFORE ABORT
 006554                                          FABORT  #TOOLNG          ;LINE TOO LONG ERROR MESSAGE
 006564  012704  000000G         6$:             MOV     #CFBUF,R4        ;POINT TO START OF COMMAND FILE BUFFER
 006570  112324                  2$:             MOVB    (R3)+,(R4)+      ;MOVE COMMAND TO BUFFER
 006572  001005                                  BNE     1$               ;BR IF NORMAL CHARACTER
                                         ;  Put cr-lf in place of asciz nulls
 006574  112764  000015  177777                  MOVB    #CR,-1(R4)
 006602  112724  000012                          MOVB    #LF,(R4)+
 006606  077210                  1$:             SOB     R2,2$            ;MOVE ALL CHARACTERS
 006610  020405                                  CMP     R4,R5            ;DID WE OVERFLOW BUFFER SPACE?
 006612  101355                                  BHI     10$              ;BR IF YES -- COMMAND TOO LONG
                                         ;
                                         ;  Null fill remainder of buffer
                                         ;
 006614  020405                  14$:            CMP     R4,R5            ;HAVE WE REACHED THE END OF THE BUFFER?
 006616  103002                                  BHIS    13$              ;BR IF YES
 006620  105024                                  CLRB    (R4)+            ;NULL FILL REST OF BUFFER
 006622  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.
                                         ;
 006624  032761  000000C 000000G 13$:            BIT     #$CCLRN!$INDRN,LSW5(R1) ;IS CCL OR IND RUNNING?
 006632  001004                                  BNE     8$               ;BR IF YES
 006634  032737  000000G 000000G                 BIT     #SCHAIN,@#JSWLOC;IS THIS A SPECIAL CHAIN EXIT?
 006642  001403                                  BEQ     9$               ;BR IF NOT
 006644  052761  000000G 000000G 8$:             BIS     #$QUIET,LSW4(R1);SET FLAG TO SUPPRESS LISTING COMMAND LINES
 006652  032761  000000G 000000G 9$:             BIT     #$CCLRN,LSW5(R1);IS THIS AN EXPANDED CCL COMMAND?
 006660  001403                                  BEQ     4$               ;BR IF NOT
 006662  052761  000000G 000000G                 BIS     #$CFCCL,LSW4(R1);REMEMBER THAT THIS IS A CCL COMMAND
                                         ;  Clear flags which say CCL or IND is running.
 006670  042761  000000C 000000G 4$:             BIC     #$CCLRN!$INDRN,LSW5(R1);CCL AND IND ARE NO LONGER RUNNING
 006676  042737  000000C 000000G                 BIC     #PASLIN!SCHAIN,@#JSWLOC ;CLEAR COMMAND-PASSED FLAGS
 006704  042761  000000C 000000G                 BIC     #$UCLRN!$UKMRN,LSW7(R1);SAY TSXUCL IS NO LONGER RUNNING
                                         ;
                                         ;  See if we are exiting from a locked program
                                         ;
 006712  032761  000000G 000000G CKLK:           BIT     #$PRGLK,LSW5(R1);IS A LOCKED PROGRAM EXITING?
 006720  001402                                  BEQ     CKSF2            ;BR IF NOT
 006722  000167  000000G                         JMP     CMDOFF           ;EXIT FROM LOCKED PROGRAM==>LOGOFF
                                         ;
                                         ;  See if we need to start execution of a secondary start-up command file
                                         ;  (The secondary start-up command file runs without privilege after the
                                         ;  initial start-up command file finishes).
                                         ;
 006726  105767  000000G         CKSF2:          TSTB    SUCF2            ;Is there a pending secondary command file?
 006732  001417                                  BEQ     NEWCMD           ;Br if not
 006734  005767  000000G                         TST     CFPNT            ;Are we currently in another command file?
 006740  001014                                  BNE     NEWCMD           ;Br if yes -- Wait for it to finish
 006742  042761  000000G 000000G                 BIC     #$SUCF,LSW9(R1) ;Say we are finished with 1st startup file
```

```
    115 006750  042761  000000G 000000G         BIC     #$NOIN,LSW3(R1) ;Allow input to be accepted for line
    116 006756  012702  000000G                 MOV     #SUCF2,R2       ;Point to name of secondary command file
    117 006762  004767  004732                  CALL    SETSUF          ;Set up command file
    118 006766  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.
                                    ;
006772  052737  000000G 000000G NEWCMD: BIS     #LCBIT,@#JSWLOC ;ENABLE LOWER-CASE INPUT
007000  116701  000000G                 MOVB    CORUSR,R1       ;GET JOB INDEX NUMBER
007004  105761  000000G                 TSTB    LCOL(R1)        ;ARE WE ALREADY AT LEFT MARGIN?
007010  001403                          BEQ     RDCMD           ;BR IF YES -- NO CR-LF NEEDED
007012                                  .PRINT  #CRLF           ;PRINT CR-LF
                                    ;
                                    ;  See if IND is in control and we need to call it to get the next command
                                    ;
007020  116701  000000G         RDCMD:  MOVB    CORUSR,R1       ;GET JOB INDEX NUMBER
007024  012702  000000G                 MOV     #INDSTA,R2      ;GET POINTER TO IND STATUS BYTE
007030  132712  000000G                 BITB    #IN$ACT,@R2     ;IS IND ACTIVE?
007034  001406                          BEQ     2$              ;BR IF NOT
007036  142712  000000C                 BICB    #IN$ACT!IN$CMD,@R2 ;CLEAR STATUS FLAGS FOR IND
007042  152712  000000G                 BISB    #IN$CNT,@R2     ;SAY WE ARE CONTINUING EXECUTION OF IND
007046  000167  002476                  JMP     INDRUN          ;GO RUN IND PROGRAM
                                    ;
                                    ;  If user-written command interface program is active, call it to
                                    ;  get the next command.
                                    ;
007052  032761  000000G 000000G 2$:     BIT     #$UXMON,LSW7(R1);Are we to use user-written command program?
007060  001405                          BEQ     3$              ;Br if not
007062  005767  000000G                 TST     CFPNT           ;Are we getting commands from a command file?
007066  001002                          BNE     3$              ;Br if yes -- Don't call user program till end
007070  000167  002162                  JMP     CALUKM          ;Enter user command processor program
                                    ;
                                    ;  Read next command line
                                    ;
007074                          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.
                                        ;
007114  012702  001156'         PRSCMD: MOV     #CMDBUF,R2      ;MOVE FINISHED COMMAND HERE
007120  005067  171320                  CLR     INDRFL          ;SAY NO INDIRECT FILE YET
007124  105067  171316                  CLRB    QUOTFL          ;Say we are not in quoted string
007130  105067  175204                  CLRB    DOLRAT          ;SAY HAVE NOT SEEN "$@"
007134  105067  175201                  CLRB    DOTAT           ;SAY HAVE NOT SEEN "#@"
007140  105067  175176                  CLRB    COLEQL          ;SAY HAVE NOT SEEN ":="
007144  105067  175173                  CLRB    NOUCL           ;SAY WE MAY CALL UCL FOR THIS COMMAND
007150  012704  001402'         SCNCMD: MOV     #INBUF,R4       ;SCAN FROM HERE
007154  112400                  8$:     MOVB    (R4)+,R0        ;GET NEXT CHAR FROM INPUT LINE
007156  001546                          BEQ     4$              ;BR IF END OF LINE HIT
007160  120027  000040                  CMPB    R0,#'           ;SKIP OVER LEADING SPACES
007164  001773                          BEQ     8$
007166  120027  000011                  CMPB    R0,#TAB         ;SKIP LEADING TABS
007172  001770                          BEQ     8$
007174  120027  000014                  CMPB    R0,#FF          ;SKIP LEADING FORM FEEDS
007200  001765                          BEQ     8$
007202  000402                          BR      9$              ;BEGIN SCANNING REAL COMMAND
                                ;
                                ;  Get next character from input line
                                ;
007204  112400                  6$:     MOVB    (R4)+,R0        ;GET NEXT CHAR FROM INPUT LINE
007206  001530                          BEQ     11$             ;BR IF END OF LINE HIT
                                ;
                                ;  See if we are in a quoted string
                                ;
007210  120067  171232          9$:     CMPB    R0,QUOTFL       ;Is this the terminating quote mark?
007214  001003                          BNE     17$             ;Br if not
007216  105067  171224                  CLRB    QUOTFL          ;Say not within quoted field now
007222  000511                          BR      5$              ;Go store terminating quote character
007224  105767  171216          17$:    TSTB    QUOTFL          ;Are we inside a quoted string now?
007230  001106                          BNE     5$              ;Br if yes -- Go store char without checking
007232  120027  000047                  CMPB    R0,#47          ;Apostrophe character?
007236  001403                          BEQ     19$             ;Br if yes
007240  120027  000042                  CMPB    R0,#42          ;Quote character?
007244  001003                          BNE     18$             ;Br if not
007246  110067  171174          19$:    MOVB    R0,QUOTFL       ;Remember we are inside a quoted string
007252  000475                          BR      5$              ;Go store character without further checking
                                ;
                                ;  Check for start of comments
                                ;
007254  120027  000041          18$:    CMPB    R0,#'!          ;START OF COMMENT FIELD?
007260  001503                          BEQ     11$             ;BR IF YES
                                ;
                                ;  Check for :=
                                ;
007262  105767  175054                  TSTB    COLEQL          ;Have we already seen :=?
007266  001067                          BNE     5$              ;Br if yes -- Ignore @'s after :=
007270  120027  000072                  CMPB    R0,#':          ;Start of := sequence?
007274  001010                          BNE     12$             ;Br if not
007276  110022                          MOVB    R0,(R2)+        ;Store into result string
007300  112400                          MOVB    (R4)+,R0        ;Get character following colon
007302  001472                          BEQ     11$             ;Br if end of line hit
```

```
007304  120027  000075                   CMPB    R0,#'=          ;Is this := ?
007310  001002                           BNE     12$             ;Br if not
007312  105267  175024                   INCB    COLEQL          ;Remember := seen within command string
                                 ;
                                 ;   Check for $@
                                 ;
007316  120027  000044           12$:    CMPB    R0,#'$          ;COULD THIS BE START OF "$@"?
007322  001014                           BNE     10$             ;BR IF NOT
007324  005767  171114                   TST     INDRFL          ;HAVE WE ALREADY SEEN @?
007330  001046                           BNE     5$              ;BR IF YES
007332  121427  000100                   CMPB    (R4),#'@        ;IS THIS "$@"?
007336  001043                           BNE     5$              ;BR IF NOT
007340  010267  171100                   MOV     R2,INDRFL       ;REMEMBER INDIRECT FILE NAME LOCATION
007344  105267  174770                   INCB    DOLRAT          ;REMEMBER PREFIX WAS "$@"
007350  005204                           INC     R4              ;SKIP PAST "$"
007352  000714                           BR      6$              ;START GETTING FILE NAME
                                 ;
                                 ;   Check for #@
                                 ;
007354  120027  000043           10$:    CMPB    R0,#'#          ;COULD THIS BE START OF "#@"?
007360  001014                           BNE     3$              ;BR IF NOT
007362  005767  171056                   TST     INDRFL          ;HAVE WE ALREADY SEEN @?
007366  001027                           BNE     5$              ;BR IF YES
007370  121427  000100                   CMPB    (R4),#'@        ;IS THIS "#@"?
007374  001024                           BNE     5$              ;BR IF NOT
007376  010267  171042                   MOV     R2,INDRFL       ;REMEMBER INDIRECT FILE NAME LOCATION
007402  105267  174733                   INCB    DOTAT           ;REMEMBER PREFIX WAS "#@"
007406  005204                           INC     R4              ;POINT PAST "."
007410  000675                           BR      6$              ;START GETTING FILE NAME
                                 ;
                                 ;   Check for @ and @@
                                 ;
007412  120027  000100           3$:     CMPB    R0,#'@          ;START OF INDIRECT FILE REFERENCE?
007416  001013                           BNE     5$              ;BR IF NOT
007420  121427  000100                   CMPB    (R4),#'@        ;Is this "@@"?
007424  001002                           BNE     20$             ;Br if not
007426  005204                           INC     R4              ;Skip past second at-sign
007430  000406                           BR      5$              ;Translate "@@" to "@" and pass with command
007432  005767  171006           20$:    TST     INDRFL          ;ALREADY SEEN @ BEFORE?
007436  001003                           BNE     5$              ;IF YES THEN IGNORE THIS ONE FOR NOW
007440  010267  171000                   MOV     R2,INDRFL       ;SAVE POINTER TO START OF FILE NAME
007444  000657                           BR      6$              ;DON'T BOTHER STORING @ IN CMDBUF
                                 ;
                                 ;   Move character to buffer
                                 ;
007446  020227  001362'          5$:     CMP     R2,#CBFEND      ;MAKE SURE WE DON'T OVERFLOW BUFFER
007452  103404                           BLO     1$              ;BR IF OK
007454                                   FABORT  #ILLCMD         ;COMMAND TOO LONG
007464  110022                   1$:     MOVB    R0,(R2)+        ;MOVE CHAR TO CMDBUF
007466  000646                           BR      6$              ;GO GET REST OF LINE
                                 ;
                                 ;   Reached end of line -- strip off any trailing spaces and tabs
                                 ;
007470  010267  171036           11$:    MOV     R2,CMDEND       ;SAVE POINTER PAST END OF COMMAND STRING
007474  020227  001156'          4$:     CMP     R2,#CMDBUF      ;HAVE WE GONE PAST START OF BUFFER?
007500  001002                           BNE     16$             ;BR IF NOT
007502  000167  177312                   JMP     RDCMD           ;YES -- THIS IS A NULL COMMAND
```

```
 007506  124227  000040          16$:    CMPB    -(R2),#'        ;IS NEXT CHARACTER A SPACE?
 007512  001770                          BEQ     4$              ;LOOP BACKWARD OVER SPACES
 007514  121227  000011                  CMPB    (R2),#TAB       ;IS THIS A TAB?
 007520  001765                          BEQ     4$              ;LOOP IF YES
 007522  005202                          INC     R2              ;POINT BEYOND LAST NON-BLANK CHARACTER
                                 ;
                                 ;  See if command line is continued
                                 ;
 007524  126227  177777  000055          CMPB    -1(R2),#'-      ;IS LINE CONTINUED?
 007532  001013                          BNE     GOTCML          ;BR IF NOT
                                 ;  Line is continued -- get more
 007534  005302                          DEC     R2              ;POINT BACK TO -
 007536                                  .GTLIN  #INBUF,#KMPRMT  ;READ COMMAND CONTINUATION LINE
 007556  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.
                                        ;
 007562  105012                         GOTCML: CLRB    (R2)            ;store null at end of command
 007564  026727  170654  001156'                CMP     INDRFL,#CMDBUF  ;Was "@" first character of command?
 007572  001443                                 BEQ     INDCMD          ;Br if yes -- Don't call UCL for this
 007574  105767  174542                         TSTB    COLEQL          ;Was := seen within command line?
 007600  001411                                 BEQ     13$             ;Br if not
 007602  005767  000000G                        TST     UCLBLK          ;Are we supporting user-defined commands?
 007606  001004                                 BNE     1$              ;Br if yes
 007610                                         FABORT  #EM$NUC         ;No user-defined commands allowed
 007620  000167  001306                 1$:     JMP     CALUCL          ;Go call UCL to process it
                                        ;
                                        ;  See if command line contains an indirect command file reference
                                        ;
 007624  016703  170614                 13$:    MOV     INDRFL,R3       ;Did we have indirect file reference?
 007630  001024                                 BNE     INDCMD          ;Br if yes
                                        ;
                                        ;  If we are to call TSXUCL before normal processing, determine if we
                                        ;  should do it for this command.
                                        ;
 007632  126727  171320  000137                 CMPB    CMDBUF,#'_      ;Was "_" specified as first char of command?
 007640  001411                                 BEQ     15$             ;Br if yes -- Don't call TSXUCL for _command
 007642  032761  000000G 000000G                BIT     #$UCLCF,LSW7(R1);Should we call TSXUCL before normal commands?
 007650  001412                                 BEQ     16$             ;Br if not
 007652  005767  000000G                        TST     UCLBLK          ;Are we allowing user-defined commands?
 007656  001407                                 BEQ     16$             ;Br if not
 007660  000167  001246                         JMP     CALUCL          ;Call TSXUCL to try to process this command
 007664  112767  000040  171264         15$:    MOVB    #' ,CMDBUF      ;Replace leading underscore with space
 007672  105267  174445                         INCB    NOUCL           ;Remember not to call TSXUCL for this command
 007676  000167  000562                 16$:    JMP     IDNCMD          ;Go try to identify the command
```

```
                                        ;
                                        ;  Command line contains indirect file reference.
                                        ;
                                        ;  Accrue indirect file name and try to open the file.
                                        ;
 007702  116701  000000G                INDCMD: MOVB    CORUSR,R1           ;GET JOB INDEX #
 007706  016703  170532                         MOV     INDRFL,R3           ;GET POINTER TO COMMAND FILE NAME
                                        ;
                                        ;  Accrue the command file spec
                                        ;
 007712  010304                                 MOV     R3,R4               ;Save pointer to start of command file name
 007714  012705  000250'                        MOV     #DKCOM,R5           ;SET DEFAULT DEV AND EXT
 007720  004767  000000G                        CALL    ACRFN               ;ACCRUE THE FILE NAME
 007724  103002                                 BCC     4$                  ;BR IF GOT NAME OK
 007726  000167  177066                         JMP     RDCMD               ;ERROR WHILE GETTING FILE NAME
                                        ;
                                        ;  See if we should let IND process the indirect command file
                                        ;
 007732  026727  170506  001156' 4$:            CMP     INDRFL,#CMDBUF      ;Was "@" first character of command?
 007740  001031                                 BNE     1$                  ;Br if not
 007742  105767  174373                         TSTB    DOTAT               ;WAS PREFIX "#@"?
 007746  001407                                 BEQ     2$                  ;BR IF NOT
 007750  005767  000000G                        TST     INDSAV              ;IS IND AVAILABLE ON SYSTEM?
 007754  001013                                 BNE     3$                  ;BR IF YES
 007756                                         FABORT  #NOIND              ;IND IS NOT AVAILABLE
 007766  032761  000000G 000000G 2$:            BIT     #$INDDF,LSW5(R1);IS IND TO BE CALLED BY DEFAULT?
 007774  001413                                 BEQ     1$                  ;BR IF NOT
 007776  105767  174336                         TSTB    DOLRAT              ;WAS COMMAND PREFIX "$@"?
 010002  001010                                 BNE     1$                  ;BR IF YES -- DON'T USE IND
 010004  132767  000000G 000000G 3$:            BITB    #IN$ACT,INDSTA      ;IS IND ACTIVE NOW?
 010012  001004                                 BNE     1$                  ;BR IF YES
 010014  010403                                 MOV     R4,R3               ;Get back pointer to start of file spec
 010016  005005                                 CLR     R5                  ;NO DEFAULT DEVICE
 010020  000167  001372                         JMP     INDINI              ;GO ENTER IND
                                        ;
                                        ;  Process this indirect command file directly rather than calling IND
                                        ;
 010024  016700  170420          1$:            MOV     FILNAM,R0           ;Get the device name
 010030  004767  000000G                        CALL    CHKTTD              ;Is the device TT?
 010034  103441                                 BCS     NOICMD              ;Br if yes -- Error
 010036  004767  000000G                        CALL    PUSHCF              ;PUSH CURRENT @FILE ON STACK
 010042  112767  000001  000000G                MOVB    #1,SERFLG           ;DO .SERR
 010050                                         .LOOKUP #XAREA,#CFCHAN,#FILNAM
 010070  112767  000000  000000G                MOVB    #0,SERFLG           ;DO .HERR, DON'T CLEAR CARRY FLAG
 010076  103024                                 BCC     CFOPEN              ;BR IF OPEN OK
 010100  116701  000000G                        MOVB    CORUSR,R1           ;GET JOB INDEX NUMBER
 010104  032761  000000G 000000G                BIT     #$SUCF,LSW9(R1)     ;ARE WE WITHIN A START-UP COMMAND FILE?
 010112  001410                                 BEQ     5$                  ;BR IF NOT
 010114                                         FERR    #NOCF               ;PRINT ERROR MESSAGE
 010130  000167  000000G                        JMP     CMDOFF              ;LOG THE JOB OFF
 010134  004767  000000G         5$:            CALL    POPCF               ;POP PUSHED COMMAND FILE STATUS
 010140                                 NOICMD: FABORT  #NOCF               ;COULDN'T OPEN @FILE
                                        ;
                                        ;  We have successfully opened the indirect command file.
                                        ;
 010150  116701  000000G                CFOPEN: MOVB    CORUSR,R1
 010154  052761  000000G 000000G                BIS     #$CFOPN,LSW4(R1);SAY CFCHAN IS OPEN
```

```
                                   ;
                                   ;  See if command file was installed with any privileges.
                                   ;
 010162  004767  001726                  CALL    INSCF           ;See if command file was installed
                                   ;
                                   ;  READ IN 1ST BLOCK FROM INDIRECT FILE
                                   ;
 010166                                  .READW  #XAREA,#CFCHAN,#CFBUF,#256.,#0
 010224  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.
 010226  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
 010234  105713                    CFPRM:  TSTB    (R3)            ;HIT END OF COMMAND?
 010236  001437                          BEQ     RDREST          ;BR IF END HIT
 010240  122327  000040                  CMPB    (R3)+,#'        ;SKIP OVER LEADING SPACES
 010244  001773                          BEQ     CFPRM
                                   ;  SEE IF WE SHOULD USE SPACE OR \ AS PARAMETER DELIMITER
 010246  012701  000040                  MOV     #' ,R1          ;ASSUME SPACE IS DELIMITER
 010252  124327  000134                  CMPB    -(R3),#'\       ;DOES HE WANT TO USE \?
 010256  001001                          BNE     8$              ;BR IF NOT -- USE SPACE
 010260  112301                          MOVB    (R3)+,R1        ;USE \ AS PARAM SEPARATOR
                                   ;  THERE ARE SOME PARAMETERS -- ACCRUE THEM
 010262  012704  000000G           8$:   MOV     #PRMPNT,R4      ;POINT TO PARAM POINTER CELLS
 010266  012705  000000G                 MOV     #PRMBUF,R5      ;POINT TO PARAMETER STRING BUFFER
 010272  020427  000000G           7$:   CMP     R4,#LSTPRM      ;TOO MANY PARAMETERS?
 010276  103062                          BHIS    TOMPRM          ;BR IF TOO MANY
 010300  010524                          MOV     R5,(R4)+        ;SET PARAMETER STRING POINTER
 010302  121301                    6$:   CMPB    (R3),R1         ;REACHED PARAM DELIMITER YET?
 010304  001407                          BEQ     4$              ;BR IF YES
 010306  105713                          TSTB    (R3)            ;HIT END OF COMMAND?
 010310  001405                          BEQ     4$              ;BR IF YES
 010312  020527  000000G                 CMP     R5,#PRMEND      ;HIT END OF PARAM STRING BUFFER?
 010316  103056                          BHIS    PTL             ;BR IF PARAM STRING TOO LONG
 010320  112325                          MOVB    (R3)+,(R5)+     ;MOVE PARAMETER TO BUFFER
 010322  000767                          BR      6$
                                   ;  HIT END OF PARAMETER -- STORE NULL IN STRING TO MARK END.
 010324  105025                    4$:   CLRB    (R5)+           ;FLAG END OF THIS PARAMETER STRING
 010326  105723                          TSTB    (R3)+           ;MORE TO ACCRUE?
 010330  001360                          BNE     7$              ;BR IF YES
 010332  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.
                                   ;
 010336  016702  170102            RDREST: MOV     INDRFL,R2       ;GET POINTER TO START OF @SPECIFICATION
 010342  001436                          BEQ     CFJMP           ;BR IF IMPLICIT @FILE EXECUTION
 010344  020227  001156'                 CMP     R2,#CMDBUF      ;IS @FILE 1ST THING IN COMMAND?
```

```
010350  001433                     BEQ     CFJMP           ;BR IF YES
                           ;
                           ;  THIS INDIRECT FILE REFERENCE IS PART OF ANOTHER COMMAND SO
                           ;  SUPPRESS THE LISTING OF THE 1ST LINE OF @FILE AS WE READ IT.
                           ;
010352  116701  000000G            MOVB    CORUSR,R1       ;GET USER INDEX #
010356  016146  000000G            MOV     LSW4(R1),-(SP)  ;SAVE COMMAND FILE STATUS FLAGS
010362  012705  000000G            MOV     #$QUIET,R5      ;GET QUIET FLAG INTO REGISTER
010366  050561  000000G            BIS     R5,LSW4(R1)     ;TURN QUIET ON
010372                             .GTLIN  #INBUF          ;READ LINE FROM @FILE
010410  030526                     BIT     R5,(SP)+        ;NEED TO SET OR RESET QUIET MODE?
010412  001002                     BNE     10$             ;BR IF WANT TO LEAVE IT SET
010414  040561  000000G            BIC     R5,LSW4(R1)     ;RESET QUIET MODE
010420  005067  170020     10$:    CLR     INDRFL          ;SAY @FILE REFERENCE HAS BEEN RESOLVED
010424  105067  173710             CLRB    DOLRAT          ;SAY HAVEN'T SEEN "$@"
010430  105067  173705             CLRB    DOTAT           ;SAY HAVEN'T SEEN "#@"
010434  000167  176510             JMP     SCNCMD          ;GO GET REST OF COMMAND
010440  000167  176354     CFJMP:  JMP     RDCMD           ;GO READ NEXT COMMAND
                           ;  ERROR -- TOO MANY PARAMETERS
010444                     TOMPRM: FABORT  #NUMPRM
                           ;  ERROR -- PARAMETER STRING TOO LONG
010454                     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.
                                      ;
 010464  012704  000000'              IDNCMD: MOV     #CMDHD,R4       ;POINT TO TABLE OF SYSTEM COMMANDS
 010470  012703  001156'                      MOV     #CMDBUF,R3      ;POINT TO OUR COMMAND
 010474  004767  000000G                      CALL    SEARCH          ;LOOK UP THE COMMAND KEYWORD
 010500  103006                               BCC     FNDCMD          ;BR IF IDENTIFIED THE COMMAND
                                      ;   ERROR DURING SEARCH.  SEE IF UNRECOGNIZED OR AMBIGUOUS.
 010502  005704                               TST     R4              ;AMBIGUOUS OR UNRECOGNIZED?
 010504  001407                               BEQ     NOCMD           ;BR IF UNRECOGNIZED
 010506                                       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.
                                      ;
 010516  116701  000000G              FNDCMD: MOVB    CORUSR,R1       ;GET USER INDEX #
 010522  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.
                                        ;
 010524  116701  000000G                NOCMD:  MOVB    CORUSR,R1        ;GET CURRENT JOB INDEX NUMBER
 010530  032761  000000G 000000G                BIT     #$UCLCM,LSW7(R1);SET UCL MIDDLE?
 010536  001410                                 BEQ     7$               ;BR IF NOT
 010540  105767  173577                         TSTB    NOUCL            ;DID UCL ALREADY REJECT THIS COMMAND?
 010544  001005                                 BNE     7$               ;BR IF YES
 010546  005767  000000G                        TST     UCLBLK           ;ARE WE ALLOWING USER-DEFINED COMMANDS?
 010552  001402                                 BEQ     7$               ;BR IF NOT
 010554  000167  000352                         JMP     CALUCL           ;SEND COMMAND TO TSXUCL
                                        ;
                                        ;  See if there is a command file on DK device with command name.
                                        ;
 010560  012703  001156'                7$:     MOV     #CMDBUF,R3       ;POINT TO KEYWORD NAME
 010564  012705  000250'                        MOV     #DKCOM,R5        ;GET DEV AND EXT DEFAULTS
 010570  004767  000000G                        CALL    ACRFN            ;ACCRUE THE FILE NAME
 010574  103513                                 BCS     6$               ;BR IF ERROR IN GETTING FILE NAME
 010576  016700  167646                         MOV     FILNAM,R0        ;Get the device name
 010602  004767  000000G                        CALL    CHKTTD           ;Is it TT?
 010606  103506                                 BCS     6$               ;Error if yes
 010610  004767  000000G                        CALL    PUSHCF           ;PUSH CURRENT @FILE STATUS
 010614  112767  000001  000000G                MOVB    #1,SERFLG        ;DON'T ABORT ON ERRORS
 010622                                         .LOOKUP #XAREA,#CFCHAN,#FILNAM
 010642  112767  000000  000000G                MOVB    #0,SERFLG        ;DO .HERR -- DON'T CLEAR CARRY FLAG
 010650  103402                                 BCS     3$               ;BR IF NOT FOUND
 010652  005005                                 CLR     R5               ;SAY NO DEFAULT DEVICE NAME
 010654  000424                                 BR      4$               ;GO SEE IF WE SHOULD RUN IND
                                        ;
                                        ;  See if there is a command file on SY device with command name.
                                        ;
 010656  016767  167406  167564         3$:     MOV     R50SY,FILNAM     ;CHANGE DEVICE NAME TO BE "SY"
 010664  112767  000001  000000G                MOVB    #1,SERFLG        ;DO .SERR
 010672                                         .LOOKUP #XAREA,#CFCHAN,#FILNAM
 010712  112767  000000  000000G                MOVB    #0,SERFLG        ;DO .HERR, DON'T CLEAR CARRY FLAG
 010720  103430                                 BCS     1$               ;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.
                                        ;
 010722  012705  004345'                        MOV     #SYTXT,R5        ;GET DEFAULT DEVICE FOR IND FILE
 010726  132767  000000G 000000G        4$:     BITB    #IN$ACT,CFIND    ;IS IND ACTIVE NOW?
 010734  001015                                 BNE     2$               ;BR IF ALREADY ACTIVE
 010736  032761  000000G 000000G                BIT     #$INDDF,LSW5(R1);IS IND WANTED?
 010744  001411                                 BEQ     2$               ;BR IF NOT
 010746                                         .PURGE  #CFCHAN          ;PURGE CHANNEL WE OPENED TO FILE
 010754  004767  000000G                        CALL    POPCF            ;POP UP TO OLD FILE
 010760  012703  001156'                        MOV     #CMDBUF,R3       ;POINT TO START OF COMMAND LINE
```

```
010764  000167  000426                      JMP     INDINI          ;GO START UP IND
                                    ;
                                    ;   This is an implicit command file execution.
                                    ;
010770  052761  000000G 000000G 2$:         BIS     #$QUIET,LSW4(R1);ALWAYS SET QUIET IF IMPLICIT RUN
010776  000167  177146                      JMP     CFOPEN          ;CONTINUE PROCESSING STARTUP OF @FILE
                                    ;
                                    ;   This is not an implicit @file call.
                                    ;
011002  004767  000000G             1$:     CALL    POPCF           ;REOPEN PREVIOUS @FILE
                                    ;
                                    ;   See if there is a program on "SY" with command name.
                                    ;
011006  012703  001156'                     MOV     #CMDBUF,R3      ;POINT TO COMMAND KEYWORD
011012  012705  000254'                     MOV     #SYSAV,R5       ;SET DEFAULT DEV AND EXT
011016  004767  000000G                     CALL    ACRFN           ;ACCRUE FILE NAME
011022  103002                              BCC     5$              ;BR IF GOT FILE NAME OK
011024  000167  175770              6$:     JMP     RDCMD           ;ERROR IN GETTING FILE NAME
011030  016700  167414              5$:     MOV     FILNAM,R0       ;Get the device name
011034  004767  000000G                     CALL    CHKTTD          ;Is the device TT?
011040  103421                              BCS     BADCMD          ;Error if yes
011042  112767  000001  000000G             MOVB    #1,SERFLG       ;DO .SERR
011050                                      .LOOKUP #XAREA,#RUNCHN,#FILNAM
011070  112767  000000  000000G             MOVB    #0,SERFLG       ;DO .HERR, DON'T CLEAR CARRY FLAG
011076  103406                              BCS     TRYUCL          ;COULDN'T FIND PROGRAM
011100  000167  000000G                     JMP     DORUN           ;START RUNNING THE PROGRAM
                                    ;
011104                              BADCMD: FABORT  #ILLCMD         ;INVALID COMMAND
```

```
                                        ;------------------------------------------------------------------
                                        ;  See if there is a SY:TSXUCL program to process the command
                                        ;
  011114  032761  000000G 000000G TRYUCL: BIT     #$UCLCL,LSW7(R1);SHOULD WE CALL TSXUCL LAST?
  011122  001451                          BEQ     URERR           ;BR IF NOT
  011124  105767  173213                  TSTB    NOUCL           ;DID UCL ALREADY REJECT THIS COMMAND?
  011130  001046                          BNE     URERR           ;BR IF YES
                                        ;
                                        ;  Call the TSXUCL program to process this command
                                        ;
  011132  105767  000000G         CALUCL: TSTB    VU$CL           ;IS THERE A UCL PROGRAM?
  011136  001443                          BEQ     URERR           ;BR IF NOT
                                        ;
                                        ;  UCL option is genned in.  See if we can find UCL program.
                                        ;
  011140                                  .LOOKUP #XAREA,#RUNCHN,#UCLNAM ;TRY TO FIND SY:UCL.SAV
  011160  103426                          BCS     9$              ;BR IF CAN'T FIND UCL PROGRAM
                                        ;
                                        ;  We found the UCL program.
                                        ;  Pass command line to it in chain area.
                                        ;
  011162  012703  000012G                 MOV     #CINDAT+12,R3   ;POINT TO CHAIN AREA
  011166  012704  001156'                 MOV     #CMDBUF,R4      ;POINT TO COMMAND LINE BUFFER
  011172  112423                  1$:     MOVB    (R4)+,(R3)+     ;MOVE COMMAND TO CHAIN DATA AREA
  011174  001376                          BNE     1$              ;LOOP TILL ASCIZ NULL MOVED
  011176  162703  000013G                 SUB     #CINDAT+13,R3   ;COMPUTE LENGTH OF COMMAND LINE
  011202  010367  000010G                 MOV     R3,CINDAT+10    ;AND STORE LENGTH INTO 510
  011206  012704  000000G                 MOV     #UCLNAM,R4      ;POINT TO NAME OF UCL PROGRAM
  011212  105267  000000G                 INCB    CINFLG          ;SIMULATE .CHAIN
  011216  116701  000000G                 MOVB    CORUSR,R1       ;GET CURRENT JOB INDEX NUMBER
  011222  052761  000000G 000000G         BIS     #$UCLRN,LSW7(R1);SAY THAT UCL PROGRAM IS RUNNING
  011230  005000                          CLR     R0              ;NO RUN OPTION FLAGS FOR PLOAD
  011232  000167  000000G                 JMP     PLOAD           ;LOAD AND RUN UCL PROGRAM
                                        ;
                                        ;  Cannot find TSXUCL program
                                        ;
  011236                          9$:     FABORT  #MISUCL         ;SAY UCL IS MISSING
                                        ;
                                        ;  Could not identify command
                                        ;
  011246                          URERR:  FABORT  #URCMD          ;UNRECOGNIZABLE COMMAND
```

```
                                               .SBTTL  CALUKM -- Start user-written command processor
                                       ;-----------------------------------------------------------------------
                                       ;  Call user-written command interface program.
                                       ;
 011256  116701  000000G               CALUKM: MOVB    CORUSR,R1          ;Get job index number
 011262  012704  000000G                       MOV     #UCISPC,R4         ;Get pointer to program name for PLOAD
 011266                                        .LOOKUP #XAREA,#RUNCHN,R4;Try to lookup program
 011304  103406                                BCS     9$                 ;Br if cannot find program
                                       ;
                                       ;  We found the program.  Enter it.
                                       ;
 011306  052761  000000G 000000G               BIS     #$UKMRN,LSW7(R1);Say user command processor is running
 011314  005000                                CLR     R0                 ;No run option flags for PLOAD
 011316  000167  000000G                       JMP     PLOAD              ;Enter the program
                                       ;
                                       ;  Cannot find the program.
                                       ;
 011322  042761  000000G 000000G 9$:           BIC     #$UKMON,LSW7(R1);Don't try to use the program again
 011330                                        FABORT  #EM$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.
                                  ;
   011340                         CMDIND:
                                  ;
                                  ;  See if IND is available
                                  ;
   011340  005767  000000G                TST     INDSAV          ;Is IND available on system?
   011344  001004                         BNE     1$              ;Br if yes
   011346                                 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.
                                  ;
   011356  004767  000000G        1$:     CALL    SKPSPC          ;Skip over any spaces
   011362  105713                         TSTB    (R3)            ;Any file specified?
   011364  001411                         BEQ     4$              ;Br if not
   011366  010304                         MOV     R3,R4           ;Save pointer to start of command file
   011370  012705  000250'                MOV     #DKCOM,R5       ;Set default device and extension
   011374  004767  000000G                CALL    ACRFN           ;Accrue the file name
   011400  103002                         BCC     2$              ;Br if got file name ok
   011402  000167  175412                 JMP     RDCMD           ;Error accruing file name
   011406  010403                 2$:     MOV     R4,R3           ;Restore pointer to start of file spec
                                  ;
                                  ;  Start execution of IND
                                  ;
   011410  005005                 4$:     CLR     R5              ;No default device string
   011412  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
                                ;
 011416                         INDINI:
                                ;
                                ;  Error if IND is already running
                                ;
 011416  132767  000000G 000000G         BITB    #IN$ACT,INDSTA  ;Is IND running now?
 011424  001404                          BEQ     6$              ;Br if not
 011426                                  FABORT  #INDACT         ;IND is already active
                                ;
                                ;  Set flag that says IND is being started
                                ;
 011436  116701  000000G        6$:      MOVB    CORUSR,R1       ;Get job index number
 011442  052761  000000G 000000G         BIS     #$INDRN,LSW5(R1);Say IND is running
                                ;
                                ;  See if command file was installed with any privileges
                                ;
 011450  004767  000440                  CALL    INSCF           ;See if command file was installed
                                ;
                                ;  Build command line of the form "SY:IND file-name"
                                ;
 011454  012702  001402'                 MOV     #BLKO,R2        ;Point to area where we will build command
                                ;
                                ;  Insert "SY:IND "
                                ;
 011460  012704  004351'                 MOV     #SYINTX,R4      ;Point to "SY:IND " text string
 011464  112422                 1$:      MOVB    (R4)+,(R2)+     ;Copy text string
 011466  001376                          BNE     1$              ;Loop till null hit
 011470  005302                          DEC     R2              ;Point back to null
                                ;
                                ;  See if we need to insert default device name
                                ;
 011472  005705                          TST     R5              ;Do we have a default device name?
 011474  001403                          BEQ     2$              ;Br if not
 011476  112522                 3$:      MOVB    (R5)+,(R2)+     ;Move in default device name
 011500  001376                          BNE     3$
 011502  005302                          DEC     R2              ;Point back over terminating null
                                ;
                                ;  Copy command text string
                                ;
 011504  010304                 2$:      MOV     R3,R4           ;Get pointer to command buffer
 011506  112422                 4$:      MOVB    (R4)+,(R2)+     ;Copy command text string
 011510  001376                          BNE     4$
                                ;
                                ;  Now move string back to command buffer
                                ;
 011512  012702  001402'                 MOV     #BLKO,R2        ;Point to start of new string
 011516  010304                          MOV     R3,R4           ;Point to destination area
 011520  112224                 5$:      MOVB    (R2)+,(R4)+     ;Copy string
 011522  001376                          BNE     5$
```

```
                                        ;
                                        ;  Now process like a R command
                                        ;
 011524  156767  000000G 000000G                BISB    UERSEV,INDERR   ;Pass error sev level to IND
 011532  142767  000000G 000000G                BICB    #IN$CMD,INDSTA  ;Say no IND command pending
 011540  012705  000254'                        MOV     #SYSAV,R5       ;Point to default device (SY)
 011544  000167  000000G                        JMP     RUNNAM          ;Enter RUN code
                                        ;
                                        ;  Reopen channel to IND.SAV file
                                        ;
 011550                                 INDRUN: .REOPEN #XAREA,#RUNCHN,#INDSAV ;REOPEN CHANNEL 16 TO IND
 011570  012704  000402'                        MOV     #INDNAM,R4      ;POINT TO NAME OF IND
 011574  105267  000000G                        INCB    CINFLG          ;SAY A .CHAIN IS IN PROGRESS
 011600  156767  000000G 000000G                BISB    UERSEV,INDERR   ;PASS ERROR SEV LEVEL TO IND
 011606  116701  000000G                        MOVB    CORUSR,R1       ;GET JOB INDEX NUMBER
 011612  052761  000000G 000000G                BIS     #$INDRN,LSW5(R1);SAY IND IS RUNNING
 011620  142767  000000G 000000G                BICB    #IN$CMD,INDSTA  ;SAY NO IND COMMAND PENDING
 011626  005000                                 CLR     R0              ;No run option flags for PLOAD
 011630  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.
                                        ;
 011634  012703  000012G                CMDCCL: MOV     #CINDAT+12,R3   ;POINT TO CHAIN DATA AREA
 011640  012704  001156'                        MOV     #CMDBUF,R4      ;POINT TO COMMAND BUFFER
 011644  112400                         1$:     MOVB    (R4)+,R0        ;GET NEXT CHAR FROM COMMAND LINE
 011646  120027  000011                         CMPB    R0,#TAB         ;IS CHARACTER TAB?
 011652  001403                                 BEQ     10$             ;BR IF YES
 011654  120027  000014                         CMPB    R0,#FF          ;FORM-FEED?
 011660  001002                                 BNE     11$
 011662  112700  000040                 10$:    MOVB    #' ,R0          ;TRANSLATE TAB AND FF TO SPACE
 011666  120027  000141                 11$:    CMPB    R0,#141         ;SEE IF IT IS A LOWER CASE LETTER
 011672  103405                                 BLO     6$              ;BR IF NOT
 011674  120027  000172                         CMPB    R0,#172         ;LOWER CASE Z
 011700  101002                                 BHI     6$              ;BR IF DELIMITER
 011702  162700  000040                         SUB     #40,R0          ;CONVERT LOWER-CASE UP UPPER-CASE
 011706  110023                         6$:     MOVB    R0,(R3)+        ;MOVE CHAR TO CHAIN AREA
 011710  001355                                 BNE     1$              ;LOOP TILL ASCIZ NULL MOVED
                                        ;  NOW SET SOME FLAGS IN LOCATION 510
 011712  005003                                 CLR     R3              ;FORM FLAG WORD IN R3
 011714  016104  000000G                        MOV     LSW5(R1),R4     ;PICK UP STATUS WORD
 011720  032704  000000G                        BIT     #$TECO!$KED,R4  ;IS DEFAULT EDITOR TECO OR KED?
 011724  001417                                 BEQ     2$              ;BR IF NOT (MUST BE EDIT THEN)
 011726  032704  000000G                        BIT     #$TECO,R4       ;IS IT TECO?
 011732  001403                                 BEQ     7$              ;BR IF NOT
 011734  052703  000001                         BIS     #C$TECO,R3      ;TELL CCL TO USE TECO
 011740  000411                                 BR      2$
 011742  026127  000000G 000000G 7$:     CMP     LTRMTP(R1),#VT52;IS TERMINAL TYPE VT52?
 011750  001403                                 BEQ     8$              ;BR IF YES (EDITOR = K52)
 011752  052703  000020                         BIS     #C$KED,R3       ;DEFAULT EDITOR = KED
 011756  000402                                 BR      2$
 011760  052703  000040                 8$:     BIS     #C$K52,R3       ;DEFAULT EDITOR = K52
 011764  032761  000000G 000000G 2$:     BIT     #$DIBOL,LSW6(R1);IS DIBOL OR DBL DEFAULT COMPILER?
 011772  001402                                 BEQ     9$              ;BR IF DBL IS DEFAULT
 011774  052703  000100                         BIS     #C$DIBL,R3      ;REMEMBER DIBOL IS DEFAULT
 012000  032704  000000G                9$:     BIT     #$WILD,R4       ;WANT IMPLICIT/EXPLICIT WILDCARDS?
 012004  001402                                 BEQ     3$              ;BR IF WANTS EXPLICIT
 012006  052703  000002                         BIS     #C$WILD,R3      ;SET IMPLICIT-WILDCARD FLAG
 012012  032704  000000G                3$:     BIT     #$CLTST,R4      ;DO CCL IN TEST MODE?
 012016  001402                                 BEQ     5$              ;BR IF NOT
 012020  052703  000010                         BIS     #C$TEST,R3      ;SET TEST MODE FLAG
 012024  032761  000000G 000000G 5$:     BIT     #$QUIET,LSW4(R1);ARE COMMAND FILES IN QUIET MODE?
 012032  001402                                 BEQ     4$              ;BR IF NOT
 012034  052703  000004                         BIS     #C$QUIT,R3      ;SET QUIET FLAG
 012040  110337  000010G                4$:     MOVB    R3,@#CINDAT+10  ;STORE FLAG WORD INTO CHAIN DATA AREA
 012044  110137  000011G                        MOVB    R1,@#CINDAT+11  ;STORE USER INDEX # IN 511 FOR CCL
                                        ;  Reopen channel 16 with CCL file status saved during initialization.
 012050                                         .REOPEN #XAREA,#RUNCHN,#CCLSAV ;OPEN CHANNEL TO CCL.SAV FILE
 012070  012704  000372'                        MOV     #CCLNAM,R4      ;POINT TO CELL WITH CCL NAME
 012074  105267  000000G                        INCB    CINFLG          ;SIMULATE CHAIN
```

```
     58 012100  052761  000000G 000000G          BIS     #$CCLRN,LSW5(R1);REMEMBER CCL.SAV IS RUNNING
     59 012106  005000                           CLR     R0              ;No run option flags for PLOAD
     60 012110  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
                                ;
012114  010246                  INSCF:  MOV     R2,-(SP)
                                ;
                                ;  See if this command file is installed
                                ;
012116  012700  000450'                 MOV     #FILNAM,R0      ;Point to file spec
012122  004767  000000G                 CALL    INSSRC          ;See if file is in install table
012126  103420                          BCS     9$              ;Br if not
                                ;
                                ;  Command file is installed.
                                ;  Apply any privilege changes.
                                ;
012130  012702  000000C                 MOV     #2*<PVNPW-1>,R2 ;Get index to last privilege word
012134  056262  000000C 000000G 1$:     BIS     II$PRV+IIBUF(R2),PRIVFO(R2) ;Set some flags
012142  046262  000000C 000000G         BIC     II$NPV+IIBUF(R2),PRIVFO(R2) ;Clear some flags
012150  162702  000002                  SUB     #2,R2           ;More to do?
012154  002367                          BGE     1$              ;Loop if yes
012156  056767  000000C 000000G         BIS     II$FLG+IIBUF,AFCF ;Set command file attribute flags
012164  004767  000000G                 CALL    RSTPRV          ;Set current attributes for command file
                                ;
                                ;  Finished
                                ;
012170  012602                  9$:     MOV     (SP)+,R2
012172  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.
                                ;
012174  010246                  TRMINI: MOV     R2,-(SP)
                                ;
                                ;  Get initial LSW2 flags
                                ;
012176  016100  000000G                 MOV     LTRMTP(R1),R0   ;GET TERMINAL TYPE FLAGS
012202  016102  000000G                 MOV     LSW2(R1),R2     ;GET LSW2 FLAGS
                                ;  VT100
012206  032700  000000G                 BIT     #VT100,R0       ;IS THIS A VT100 TERMINAL?
012212  001410                          BEQ     14$             ;BR IF NOT
012214  042702  000000G                 BIC     #VT10NO,R2      ;SET FLAGS FOR LSW2
012220  052702  000000G                 BIS     #VT10FL,R2
012224                                  .PRINT  #IMVT10         ;CLEAR SCREEN
012232  000503                          BR      15$
                                ;  VT200
012234  032700  000000G         14$:    BIT     #VT2007!VT2008,R0 ;VT200 terminal?
012240  001410                          BEQ     32$             ;Br if not
012242  042702  000000G                 BIC     #VT20NO,R2      ;Set flags for VT200
012246  052702  000000G                 BIS     #VT20FL,R2
012252                                  .PRINT  #IMVT10         ;Clear screen
012260  000470                          BR      15$
                                ;  VT52
012262  032700  000000G         32$:    BIT     #VT52,R0        ;IS THIS A VT52 TERMINAL?
012266  001410                          BEQ     16$             ;BR IF NOT
012270                                  .PRINT  #IMVT52         ;CLEAR SCREEN
012276  042702  000000G                 BIC     #VT52NO,R2      ;SET FLAGS IN LSW2
012302  052702  000000G                 BIS     #VT52FL,R2
012306  000455                          BR      15$
                                ;  ADM3A
012310  032700  000000G         16$:    BIT     #ADM3A,R0       ;IS THIS AN ADM3A TERMINAL?
012314  001410                          BEQ     17$             ;BR IF NOT
012316                                  .PRINT  #IMADM3         ;CLEAR SCREEN
012324  042702  000000G                 BIC     #ADM3NO,R2      ;SET LSW2 FLAGS
012330  052702  000000G                 BIS     #ADM3FL,R2
012334  000442                          BR      15$
                                ;  LA36
012336  032700  000000G         17$:    BIT     #LA36,R0        ;IS THIS AN LA36?
012342  001405                          BEQ     18$             ;BR IF NOT
012344  042702  000000G                 BIC     #LA36NO,R2      ;SET FLAGS
012350  052702  000000G                 BIS     #LA36FL,R2
012354  000432                          BR      15$
                                ;  LA120
012356  032700  000000G         18$:    BIT     #LA120,R0       ;IS THIS AN LA120?
012362  001405                          BEQ     19$             ;BR IF NOT
012364  042702  000000G                 BIC     #LA12NO,R2      ;SET FLAGS
012370  052702  000000G                 BIS     #LA12FL,R2
012374  000422                          BR      15$
                                ;  Hazeltine
012376  032700  000000G         19$:    BIT     #HAZEL,R0       ;HAZELTINE TERMINAL?
012402  001410                          BEQ     20$             ;BR IF NOT
```

```
   012404                                  .PRINT  #IMHAZL          ;CLEAR SCREEN
   012412  042702  000000G                 BIC     #HAZLNO,R2       ;SET TERMINAL CONTROL FLAGS
   012416  052702  000000G                 BIS     #HAZLFL,R2
   012422  000407                          BR      15$
                                   ;  DIABLE & QUME
   012424  032700  000000C         20$:    BIT     #DIABLO!QUME,R0 ;DIABLO OR QUME TERMINAL?
   012430  001410                          BEQ     1$               ;BR IF NOT
   012432  042702  000000G                 BIC     #DIABNO,R2       ;SET FLAGS
   012436  052702  000000G                 BIS     #DIABFL,R2
                                   ;  Store updated LSW2 flags.
   012442  010261  000000G         15$:    MOV     R2,LSW2(R1)
   012446  010261  000000G                 MOV     R2,LSW2S(R1)
                                   ;
                                   ;  Finished
                                   ;
   012452  012602                  1$:     MOV     (SP)+,R2
   012454  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
                              ;
012456  010246                VIRINI: MOV     R2,-(SP)
012460  010346                        MOV     R3,-(SP)
012462  010446                        MOV     R4,-(SP)
                              ;
                              ;  Get job index number of our primary job
                              ;
012464  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.
                              ;
012470  005762  000000G               TST     LWINDO(R2)       ;Is primary line using a display window?
012474  001413                        BEQ     1$               ;Br if not
012476  012700  001114'               MOV     #MAKWIN,R0       ;Point to make-window argument block
012502  110260  000007                MOVB    R2,7(R0)         ;Set index # of primary job
012506  106260  000007                ASRB    7(R0)            ;Convert index to job number
012512  104375                        EMT     375              ;Try to create a window
012514  103403                        BCS     1$               ;Br if unable to create a window
012516  012700  001126'               MOV     #MAPWIN,R0       ;Point to map-window argument block
012522  104375                        EMT     375              ;Select this window
                              ;
                              ;  Print n>
                              ;
012524  016204  000000G       1$:     MOV     LSECPT(R2),R4    ;GET ADDRESS OF TABLE WITH VIRTUAL LINES
012530  005003                        CLR     R3               ;COUNT VIRTUAL LINE #'S
012532  005203                13$:    INC     R3
012534  120124                        CMPB    R1,(R4)+         ;LOOK UP OUR VIRTUAL LINE #
012536  001375                        BNE     13$
012540  062703  000060                ADD     #'0,R3           ;CONVERT NUMBER TO ASCII
012544                                .PRINT  #CRLF            ;PRINT CR-LF
012552                                .TTYOUT R3               ;PRINT LINE #
012560                                .TTYOUT #76              ;PRINT '>'
                              ;
                              ;  Try to copy any key definitions from primary line
                              ;
012570  016102  000000G               MOV     LNPRIM(R1),R2    ;Get our primary line number
012574  004767  000000G               CALL    INIUKD           ;Copy any user-defined keys
                              ;
                              ;  Finished
                              ;
012600  012604                9$:     MOV     (SP)+,R4
012602  012603                        MOV     (SP)+,R3
012604  012602                        MOV     (SP)+,R2
012606  000207                        RETURN
```

```
                                            .SBTTL  CPYPRN -- Copy context info from parent job
                                    ;-----------------------------------------------------------------------
                                    ;  Copy context information from a parent job.
                                    ;
 012610  010246                     CPYPRN: MOV     R2,-(SP)
 012612  010346                             MOV     R3,-(SP)
 012614  010546                             MOV     R5,-(SP)
                                    ;
                                    ;  Return immediately if there is no parent job
                                    ;
 012616  016102  000000G                    MOV     LPARNT(R1),R2   ;Get # of parent job
 012622  001002                             BNE     2$              ;Br if there is a parent job
 012624  000167  000456                     JMP     9$              ;No parent job -- Nothing to copy
                                    ;
                                    ;  Copy file context and privilege information
                                    ;
 012630  010267  165746             2$:     MOV     R2,CPYCXT+2     ;Get # of parent job
 012634  006267  165742                     ASR     CPYCXT+2        ;Convert to job number
 012640  012700  000600'                    MOV     #CPYCXT,R0      ;Point to EMT arg block
 012644  104375                             EMT     375             ;Copy context from parent job
 012646  004767  000000G                    CALL    LDCLEN          ;Reinit LD status
                                    ;
                                    ;  Copy execution priority
                                    ;
 012652  012700  001110'                    MOV     #PRIEMT,R0      ;Point to emt argument block
 012656  116260  000000G 000002             MOVB    LBSPRI(R2),2(R0);Get base priority from parent job
 012664  104375                             EMT     375             ;Set the job priority
                                    ;
                                    ;  Copy user name and PPN
                                    ;
 012666  010203                             MOV     R2,R3           ;Get # of parent job
 012670  010105                             MOV     R1,R5           ;Get # of our job
 012672  070327  000006                     MUL     #6.,R3          ;Get offset to name of parent job
 012676  070527  000006                     MUL     #6.,R5          ;Get offset to name of our job
 012702  062703  000000G                    ADD     #LUNAME,R3      ;Point to tables
 012706  062705  000000G                    ADD     #LUNAME,R5
 012712  012700  000014                     MOV     #12.,R0         ;Get # chars to move
 012716  112325                     1$:     MOVB    (R3)+,(R5)+     ;Copy user names
 012720  077002                             SOB     R0,1$
 012722  016261  000000G 000000G            MOV     LPROJ(R2),LPROJ(R1);Copy project number
 012730  016261  000000G 000000G            MOV     LPROG(R2),LPROG(R1);Copy programmer number
 012736  016167  000000G 000000G            MOV     LPROJ(R1),UPPN  ;Set project number
 012744  016167  000000G 000002G            MOV     LPROG(R1),UPPN+2;Set programmer number
                                    ;
                                    ;  Copy flags from LSW2
                                    ;
 012752  016261  000000G 000000G            MOV     LSW2(R2),LSW2(R1) ;Copy LSW2 flags
 012760  016261  000000G 000000G            MOV     LSW2S(R2),LSW2S(R1)
                                    ;
                                    ;  Copy some flags from LSW5
                                    ;
 012766  016200  000000G                    MOV     LSW5(R2),R0     ;Get flags from parent's LSW5
 012772  042700  000000C                    BIC     #^C<ISPF5>,R0   ;Clear all but selected flags
 012776  042761  000000G 000000G            BIC     #ISPF5,LSW5(R1) ;Clear those flags in our LSW5
 013004  050061  000000G                    BIS     R0,LSW5(R1)     ;Transfer flags from parent job
                                    ;
                                    ;  Copy some flags from LSW6
```

```
                                        ;
 013010  016200  000000G                        MOV     LSW6(R2),R0      ;Get flags from parent's LSW6
 013014  042700  000000C                        BIC     #^C<ISPF6>,R0    ;Clear all but selected flags
 013020  042761  000000G 000000G                BIC     #ISPF6,LSW6(R1) ;Clear those flags in our LSW6
 013026  050061  000000G                        BIS     R0,LSW6(R1)      ;Transfer flags from parent job
                                        ;
                                        ;   Copy some flags from LSW7
                                        ;
 013032  016200  000000G                        MOV     LSW7(R2),R0      ;Get flags from parent's LSW7
 013036  042700  000000C                        BIC     #^C<ISPF7>,R0    ;Clear all but selected flags
 013042  042761  000000G 000000G                BIC     #ISPF7,LSW7(R1) ;Clear those flags in our LSW7
 013050  050061  000000G                        BIS     R0,LSW7(R1)      ;Transfer flags from parent job
                                        ;
                                        ;   Copy some flags from LSW9
                                        ;
 013054  016200  000000G                        MOV     LSW9(R2),R0      ;Get flags from parent's LSW9
 013060  042700  000000C                        BIC     #^C<ISPF9>,R0    ;Clear all but selected flags
 013064  042761  000000G 000000G                BIC     #ISPF9,LSW9(R1) ;Clear those flags in our LSW9
 013072  050061  000000G                        BIS     R0,LSW9(R1)      ;Transfer flags from parent job
                                        ;
                                        ;   Copy some flags from LSW11
                                        ;
 013076  016200  000000G                        MOV     LSW11(R2),R0     ;Get flags from parent's LSW11
 013102  042700  000000C                        BIC     #^C<ISPF11>,R0   ;Clear all but selexted flags
 013106  042761  000000G 000000G                BIC     #ISPF11,LSW11(R1);Clear those flags in our LSW11
 013114  050061  000000G                        BIS     R0,LSW11(R1)     ;Transfer flags from parent job
                                        ;
                                        ;   Copy Kmon prompt string
                                        ;
 013120  010267  165756                         MOV     R2,EMCXCP+4      ;Set # of job we are copying from
 013124  012767  000000G 165752                 MOV     #KMPRMT,EMCXCP+6;Set address of item
 013132  012767  000000C 165746                 MOV     #<<MXPRMT+2>&76>,EMCXCP+8.;Set # bytes to copy
 013140  012700  001076'                        MOV     #EMCXCP,R0       ;Point to EMT arg block
 013144  104375                                 EMT     375              ;Copy Kmon prompt string
                                        ;
                                        ;   Copy default printer form name
                                        ;
 013146  012767  000000G 165730                 MOV     #UFORM,EMCXCP+6 ;Set address of item
 013154  012767  000006  165724                 MOV     #6,EMCXCP+8.     ;Set # of bytes to copy
 013162  012700  001076'                        MOV     #EMCXCP,R0       ;Point to EMT arg block
 013166  104375                                 EMT     375              ;Copy default form name
                                        ;
                                        ;   Copy User Command Interface (UCI) file spec
                                        ;
 013170  012767  000000G 165706                 MOV     #UCISPC,EMCXCP+6;Set address of item to copy
 013176  012767  000010  165702                 MOV     #8.,EMCXCP+8.    ;Set # bytes to copy
 013204  012700  001076'                        MOV     #EMCXCP,R0       ;Point to EMT arg block
 013210  104375                                 EMT     375              ;Copy UCI file spec
                                        ;
                                        ;   Copy subprocess start-up command file spec
                                        ;
 013212  012767  000000G 165664                 MOV     #SBPSUF,EMCXCP+6;Set address of item to copy
 013220  012767  000020  165660                 MOV     #16.,EMCXCP+8.   ;Set # bytes to copy
 013226  012700  001076'                        MOV     #EMCXCP,R0       ;Point to EMT arg block
 013232  104375                                 EMT     375              ;Copy subprocess start-up file spec
                                        ;
                                        ;   Copy print-window device information
                                        ;
    013234  012767  000000G 165642               MOV     #JPWDEV,EMCXCP+6;Set address of print-window device
    013242  012767  000002  165636               MOV     #2,EMCXCP+8.    ;Set # bytes to copy
    013250  012700  001076'                      MOV     #EMCXCP,R0      ;Point to EMT arg block
    013254  104375                               EMT     375             ;Copy JPWDEV
    013256  012767  000000G 165620               MOV     #JPWTYP,EMCXCP+6;Set address of print-window dev type
    013264  012700  001076'                      MOV     #EMCXCP,R0      ;Point to EMT arg block
    013270  104375                               EMT     375             ;Copy JPWTYP
    013272  012767  000000G 165604               MOV     #JPWFLG,EMCXCP+6;Set address of print-window flag word
    013300  012700  001076'                      MOV     #EMCXCP,R0      ;Point to EMT arg block
    013304  104375                               EMT     375             ;Copy JPWFLG
                                        ;
                                        ;  Finihsed
                                        ;
    013306  012605                      9$:      MOV     (SP)+,R5
    013310  012603                               MOV     (SP)+,R3
    013312  012602                               MOV     (SP)+,R2
    013314  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).
                                        ;
013316  020127  000000G                 PRTGRT: CMP     R1,#LSTPL       ;IS THIS A REAL OR VIRTUAL LINE?
013322  003157                                  BGT     GRTVIR          ;BR IF VIRTUAL -- ABBREVIATE GREETING
013324  005727  000002                          TST     #GREET          ;IS PRINTED GREETING WANTED?
013330  001403                                  BEQ     23$             ;BR IF NOT
013332                                          .PRINT  #GRT1           ;TSX NAME AND VERSION NUMBER
013340  032761  000000G 000000G 23$:            BIT     #$1STLG,LSW6(R1);IS THIS THE 1ST TIME THIS LINE LOGGED ON?
013346  001064                                  BNE     GRTFIN          ;BR IF NOT
013350  052761  000000G 000000G                 BIS     #$1STLG,LSW6(R1);REMEMBER 1ST LOGON HAS OCCURED
013356  020127  000000G                         CMP     R1,#LSTPL       ;IS THIS A REAL OR VIRTUAL LINE?
013362  101056                                  BHI     GRTFIN          ;BR IF VIRTUAL -- SKIP LONG MESSAGE
013364  005727  000002                          TST     #GREET          ;SHOULD WE PRINT GREETING MESSAGE?
013370  001453                                  BEQ     GRTFIN          ;BR IF NOT
                                        ;
                                        ;  Calculate the message checksum.
                                        ;
013372  012702  176532G                         MOV     #SUMS-1246,R2   ;GET ADDRESS OF START OF MESSAGE
013376  005003                                  CLR     R3              ;FORM CHECKSUM IN R3
013400  012704  001402'                         MOV     #BLK0,R4        ;TEMPORARILY STORE GREETING HERE
013404  112200                          2$:     MOVB    (R2)+,R0        ;GET CHARACTER FROM MESSAGE
013406  060003                                  ADD     R0,R3           ;FORM CHECKSUM
013410  005400                                  NEG     R0              ;DECRIPT CHARACTER
013412  001424                                  BEQ     3$              ;BR IF END OF MESSAGE TEXT
013414  020227  000000G                         CMP     R2,#LICTXT      ;CHECK FOR LICENSE TEXT
013420  103371                                  BHIS    2$              ;BR IF INTO THE LICENSE TEXT
                                        ;
                                        ;  Check for various flags that indicated message variations.
                                        ;
013422  105767  000000G                         TSTB    PROFLG          ;ARE WE RUNNING ON A PROFESSIONAL?
013426  001003                                  BNE     4$              ;BR IF YES -- PRINT TRUNCATED GREETING
013430  005727  000001                          TST     #GREET-1        ;CHECK FOR TRUNCATED GREETING MESSAGE
013434  001013                                  BNE     3$              ;BR IF FULL MESSAGE IS PRINTED
013436  020227  000000G                 4$:     CMP     R2,#TRGRET      ;CHECK FOR TRUNCATED MESSAGE END
013442  103410                                  BLO     3$              ;CONTINUE MOVING THE GREETING MESSAGE
013444  101357                                  BHI     2$              ;GET THE NEXT CHARACTER
013446  112724  000056                          MOVB    #'.,(R4)+       ;PLACE PERIOD AT THE END OF TRUNCATED MESSAGE
013452  112724  000015                          MOVB    #CR,(R4)+       ;PLACE CARRIAGE RETURN AFTER PERIOD
013456  112724  000012                          MOVB    #LF,(R4)+       ;PLACE LINE-FEED AFTER PERIOD
013462  000750                                  BR      2$              ;GET THE NEXT CHARACTER
013464  110024                          3$:     MOVB    R0,(R4)+        ;PUT IN TEMP BUFFER
013466  001346                                  BNE     2$              ;BR IF MORE OF MESSAGE TEXT
013470                                  1$:     .PRINT  #BLK0           ;PRINT GREETING MESSAGE
013476  005403                                  NEG     R3              ;DECRIPT CHECKSUM
013500  005203                                  INC     R3
                                        ;*************************************************************
                                        ;  If the following range of instructions are altered --
                                        ;  change the program that assigns micro TSX-Plus license numbers.
013502  020327  000000G                         CMP     R3,#SUCS        ;CHECK CHECKSUM
013506  001404                                  BEQ     GRTFIN          ;BR IF OK
                                        ;  Checksum failure -- kill TSX
013510  005037  000100                          CLR     @#100           ;DIE HORRIBLY...
013514  005037  000060                          CLR     @#60
```

```
                                        ;
                                        ;  Checksum ok, set up site number.
                                        ;
013520  105767  0000000G                GRTFIN: TSTB    PROFLG          ;ARE WE RUNNING ON A PROFESSIONAL?
013524  001415                                  BEQ     1$              ;BR IF NOT
                                        ;
                                        ;  Print license number on a Professional
                                        ;
013526                                          .PRINT  #TM$LN1         ;"License # = mmmm-TPS-"
013534  016705  0000000G                        MOV     TSXSIT,R5       ;Get the license number
013540  004767  0000000G                        CALL    PRTDEC          ;Print it
013544                                          .PRINT  #CRLF           ;Terminate the print line
013552  004767  0000000G                        CALL    DATTIM          ;Print date and time
013556  000441                                  BR      GRTVIR          ;Go print line number
                                        ;
                                        ;  Print license number if not on a professional
                                        ;
013560  012767  0000000G 0000000G 1$:           MOV     #TSXLN,TSXSIT   ;STORE SITE NUMBER
                                        ;  If the preceeding range of instructions are altered --
                                        ;  change the program that assigns micro TSX-Plus license numbers.
                                        ;****************************************************************
013566  005727  000001                          TST     #GREET-1        ;SHOULD WE PRINT DATE AND TIME?
013572  002451                                  BLT     GRTEND          ;BR IF NOT
013574  012704  001402'                         MOV     #BLKO,R4        ;TEMPORARILY STORE GREETING HERE
013600  012702  0000000G                        MOV     #LICTXT,R2      ;OBTAIN LICENSE TEXT ADDRESS
013604  112200                          21$:    MOVB    (R2)+,R0        ;GET CHARACTER FROM MESSAGE
013606  005400                                  NEG     R0              ;DECRIPT CHARACTER
013610  110024                          31$:    MOVB    R0,(R4)+        ;PUT IN TEMP BUFFER
013612  001374                                  BNE     21$             ;BR IF MORE OF MESSAGE TEXT
013614  112724  000200                          MOVB    #200,(R4)+      ;NO CR/LF FOLLOWING TEXT
013620                                          .PRINT  #BLKO           ;PRINT GREETING MESSAGE
                                        ;
                                        ;  If not supported ("S" or "s"), then print license is unsupported.
                                        ;
013626  116700  0000000G                PRTSUP: MOVB    SUPCOD,R0       ;OBTAIN THE SUPPORT CODE (SIGN EXTEND)
013632  005400                                  NEG     R0              ;DECRIPT THE CHARACTER
013634  120027  000123                          CMPB    R0,#'S          ;CHECK FOR SUPPORTED LICENSE
013640  001406                                  BEQ     61$             ;BR IF LICENSE SUPPORTED
013642  120027  000163                          CMPB    R0,#'s          ;CHECK FOR SUPPORTED LICENSE
013646  001403                                  BEQ     61$             ;BR IF LICENSE SUPPORTED
013650                                          .PRINT  #UNSUP          ;PRINT UNSUPPORTED LICENSE TEXT
013656  004767  0000000G                61$:    CALL    DATTIM          ;PRINT CURRENT DATE & TIME
                                        ;
                                        ;  Print line number.
                                        ;
013662                                  GRTVIR: .PRINT  #LINNTX         ;DISPLAY LINE #
013670  010105                                  MOV     R1,R5
013672  006205                                  ASR     R5
013674  004767  0000000G                        CALL    PRTDEC
013700                                          .PRINT  #CRLF
013706                                          .TTYOUT #LF
013716  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.
                                  ;
 013720  010246                   SETSUF: MOV     R2,-(SP)
 013722  010346                           MOV     R3,-(SP)
                                  ;
                                  ;  Start execution of a new command file
                                  ;
 013724  004767  000000G                  CALL    PUSHCF          ;SET UP FOR NEW COMMAND FILE
                                  ;
                                  ;  Move command file name to buffer
                                  ;
 013730  012703  000000G                  MOV     #CFBUF,R3       ;POINT TO COMMAND FILE BUFFER
 013734  032761  000000G 000000G          BIT     #$DETCH,LSW(R1) ;Is this a detached job?
 013742  001004                           BNE     63$             ;Br if yes
 013744  112723  000136                   MOVB    #'^,(R3)+       ;PUT '^(@ AT FRONT OF COMMAND
 013750  112723  000050                   MOVB    #'(,(R3)+
 013754  112723  000100           63$:    MOVB    #'@,(R3)+
 013760  112223                   5$:     MOVB    (R2)+,(R3)+     ;MOVE FILE NAME TO COMMAND AREA
 013762  001376                           BNE     5$              ;LOOP TILL ALL MOVED
 013764  112763  000015  177777           MOVB    #CR,-1(R3)      ;END LINE WITH CR-LF
 013772  112723  000012                   MOVB    #LF,(R3)+
 013776  105023                   6$:     CLRB    (R3)+           ;FILL REST OF BUFFER WITH NULLS
 014000  020327  000000G                  CMP     R3,#CFEND
 014004  103774                           BLO     6$
                                  ;
                                  ;  Finished
                                  ;
 014006  012603                           MOV     (SP)+,R3
 014010  012602                           MOV     (SP)+,R2
 014012  000207                           RETURN
                                  ;
         004362'                          .END    START
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 11432 Words  ( 45 Pages)
Size of core pool: 17920 Words  ( 70 Pages)
Operating  system: RT-11

Elapsed time: 00:01:37.58
DK:TSKMN1,LP:TSKMN1=DK:TSKMN1.MAC/C/N:SYM
```

```
$1STLG   1-78     29-12    29-14
$CARUP   1-95
$CCLRN   1-96     12-50    12-66    13-16    13-87    13-92    13-96    24-58
$CFABT   1-110    12-5     12-63
$CFALL   1-116
$CFCCL   1-116    13-94
$CFDCC   1-116    12-32
$CFKIL   1-160    10-21    12-5     12-59
$CFOPN   1-122    13-51    17-57
$CFSOT   1-114
$CHACT   1-68     12-33
$CLTST   1-106    24-46
$CTRLC   1-109    10-22
$CTRLS   1-101
$DEAD    1-159
$DEBUG   1-87     12-26
$DEFER   1-127    10-162
$DETCH   1-99     10-160   30-19
$DIBOL   1-78     24-40
$DILUP   1-112
$DISCN   1-100
$DOOFF   1-118    11-26
$DUPRN   1-113    11-21    11-25
$ECHO    1-115    10-162
$EMTTR   1-105
$FORM    1-114
$FORMO   1-116
$HITTY   1-68
$INCOR   1-131
$INDAB   1-109    12-51
$INDDF   1-158    17-26    19-53
$INDRN   1-158    13-16    13-87    13-96    23-21    23-73
$INIT    1-159
$INKMN   1-109
$KED     1-131    10-108   24-29
$KINIT   1-64     10-11    10-175
$LC      1-115    10-162
$MAPOK   1-93
$MLOCK   1-88
$NOIN    1-68     13-115
$NOINT   1-106    12-34
$NOVLN   1-57     10-68
$NOWTT   1-68     12-33    12-38
$NTGCC   1-117    12-35
$PAGE    1-115
$PHONE   1-159
$PRGLK   1-97     13-102
$QTSET   1-136
$QUIET   1-128    13-91    17-122   19-62    24-49
$RNIOP   1-114    12-26
$SCOPE   1-115
$SETRN   1-114    11-18
$SLKED   1-95
$SLLET   1-95     10-124
$SLON    1-95
$SLTTY   1-95
```

```
TSKMN1 -- TSX-Plus Keyboard Mon MACRO V05.04  Monday 21-Dec-87 11:40 Page S-2
Cross reference table (CREF V05.04)


$SNWTT   1-156    12-36
$SPLJB   1-101
$SUCF    1-64     13-114   17-47
$SYSPS   1-67     10-138
$TAB     1-114
$TAPE    1-154
$TECO    1-136    10-106   24-29    24-31
$TTGAG   1-122
$UCLCF   1-123    10-114   16-26
$UCLCL   1-125    10-120   20-4
$UCLCM   1-125    10-117   19-14
$UCLRN   1-124    13-30    13-40    13-98    20-31
$UKMON   1-96     14-24    21-18
$UKMRN   1-96     13-98    21-12
$VNOTT   1-61
$VTESC   1-92     12-25
$WILD    1-137    10-130   24-43
...V1   10-81    10-81    10-88    10-88    10-88    10-97    10-97    10-98    10-98    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   10-81    10-81    10-81#   10-81#   10-88    10-88    10-88#   10-88#   10-97    10-97    10-97#   10-97#
        10-98    10-98#   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#
AB1      1-73      4-30
AB10     1-74      4-37
AB11     1-74      4-38
AB12     1-74      4-39
AB13     1-74      4-40
AB14     1-74      4-41
AB15     1-74      4-42
AB16     1-74      4-43
AB2      1-74      4-31
AB3      1-74      4-32
AB4      1-74      4-33
AB5      1-74      4-34
AB6      1-74      4-35
ABCMD    1-76     18-17
ABM1     1-73      4-28
ABM10    1-73      4-21
ABM11    1-72      4-20
ABM12    1-72      4-19
ABM13    1-72      4-18
ABM14    1-72      4-17
ABM15    1-72      4-16
ABM16    1-72      4-15
ABM17    1-72      4-14
ABM2     1-73      4-27
ABM20    1-71      4-13
ABM21    1-71      4-12
ABM22    1-71      4-11
ABM23    1-71      4-10
```

```
TSKMN1 -- TSX-Plus Keyboard Mon MACRO V05.04  Monday 21-Dec-87 11:40 Page S-3
Cross reference table (CREF V05.04)


ABM24    1-71      4-9
ABM3     1-73      4-26
ABM4     1-73      4-25
ABM5     1-73      4-24
ABM6     1-73      4-23
ABM7     1-73      4-22
ABRTAD   1-61     11-50     11-59*
ABRTCD   1-61     11-30     11-60*
ABRTCF   1-168    12-49     12-65     13-20
ABRTOV   1-61     11-56
ABTMSG   4-29#    11-37
ACRDEC   1-176
ACRFIL   1-175
ACRFN    1-168    17-13     19-26     19-73     22-22
ACROCT   1-178
ADM3A    1-150    26-37
ADM3FL   1-151    26-41
ADM3NO   1-152    26-40
AFCF     1-58     25-28*
ALCDEV   1-50      7-52#     8-53      8-58      8-63
ALCERR   1-71      4-7
ALDBLK   1-191     7-29#
ALDEMT   1-50      8-52#
ALDEX    1-190     1-191     7-28#
ALFN     1-202
AMBOPT   1-51      1-176
ASDEX    1-173     7-2#
ASKLNM   1-172     7-40#
ASNEND   1-113
ASNHD1   1-185
ASNHD2   1-185
ASNOVF   1-51      1-198
ASNTBL   1-112
AUTHFN   1-195     7-30#
BADBOT   1-49
BADCMD   1-172    19-78     19-85#
BADPMT   1-179
BADPRI   1-179
BADSAV   1-172
BASMAP   1-145
BDFNAM   1-175
BDLGOP   1-180
BDLIN    1-193
BELL     2-16#
BLANK    2-15#    12-22
BLKO     1-171     9-5#      9-6       9-7      10-94     11-44     11-47     23-29     23-54     29-24     29-47     29-81
        29-88
BLKWDS   2-20#
BOTDEV   1-108
BOTEMT   1-47      8-112#
BOTUNI   1-108
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
C.CSW    1-119
C.DEVQ   1-119
C.SBLK   1-119
CALUCL   1-184    16-14     16-30     19-20     20-11#
CALUKM  14-28     21-5#
CBFEND   9-2#     15-103
CCLNAM   1-199     7-32#    24-56
CCLSAV   1-104    24-55
CD$$SZ   1-120     8-70      8-79
CD$$UB   1-120
CD$BAS   1-120
CD$DVU   1-120
CD$JOB   1-120
CD$NAM   1-120
CD$TOP   1-136
CDBUF    1-50      8-71      8-80      9-8#
CDGADR   1-53      8-69#
CDGEMT   1-53      8-67#
CDPADR   1-53      8-78#
CDPEMT   1-53      8-76#
CF$IND   1-160
CF$QUT   1-160
CF2DEP   1-48
CFABLV   1-161
CFBLK    1-128
CFBUF    1-104    13-53     13-67     17-65     30-18
CFCHAN   1-127    17-43     17-43     17-65     17-65     19-33     19-33     19-43     19-43     19-55     19-55
CFEND    1-104    17-70     30-29
CFHOLD   1-135
CFIND    1-118    19-51
CFJMP   17-113    17-115    17-132#
CFLFL4   1-123
CFNEST   1-121
CFOPEN  17-45     17-56#    19-63
CFPNT    1-128    13-112    14-26     17-70*
CFPRM   17-66     17-76#    17-79
CFSEND   1-122
CFSP     1-122    10-35*
CFSPND   1-134    12-39     12-42*
CFSQEZ   1-128    13-55
CFSTK    1-64     10-35
CFSTRT   1-60     12-41
CFSTS    1-160
CHAIN    1-109
CHKDEV   1-199
CHKDLM   1-189
CHKMNT   1-174
CHKMTX   1-174
CHKTTD   1-199    17-39     19-29     19-77
CINDAT   1-144    13-21     13-32     13-34     13-61     20-22     20-26     20-27*    24-11     24-52*    24-53*
CINFLG   1-61     11-62*    12-7*     12-9      20-29*    23-70*    24-57*
CKLK    13-102#
CKPASL  12-73     13-5#
CKPRIV   1-176
```

```
CKSF2    13-103     13-110#
CMDACC    1-85       6-68
CMDALC    1-83       6-51
CMDASN    1-83       6-48
CMDBOT    1-85       1-85       6-82       6-83
CMDBUF    1-167      1-184      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    1-199      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       6-72
          6-73       6-75      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
CMDDSN    1-165      6-49       6-50
CMDDSP    1-83       6-42
CMDEND    7-51#     13-39*     15-111*
CMDFMT    1-82       6-33
CMDFRM    1-165      6-32
CMDHD     1-67       6-8#      18-10
CMDIND    6-53      22-6#
CMDINI    1-85       6-70
CMDINS    1-77       6-78
CMDKIL    1-84       6-64
CMDMEM    1-84       6-57
CMDMNT    1-82       6-39
CMDMON    1-82       6-40
CMDOFF    1-108      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    1-65       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       6-84
CMDSND    1-81       6-22
CMDSPN    1-65       6-80
CMDSPO    1-84       6-62
CMDSQZ    1-85       6-71
CMDTIM    1-82       6-38
CMDUSE    1-85       6-69
CMDWHO    1-82       6-24       6-25
CMDYEL    1-80       6-23
COAD      1-190      1-191
COAF      1-203
COAL      1-190      1-191
COLOO     1-184
COLEQL    9-21#     15-11*     15-51      15-60*     16-9
CONFIG    1-117
CORUSR    1-62      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     1-184      7-47#
```

```
CPUAL     1-184     7-48#
CPUMSG    1-48
CPYCXT    8-39#    28-17*    28-18*    28-19
CPYPRN   10-171    28-5#
CR        2-14#    13-71     29-42     30-26
CRLF      1-179    11-58     14-9      27-39     29-69     29-107
CS$RON    1-91
CSHCLN    8-35#
CSHHD     1-164
CSHMSG    1-52
CSIMS1    1-178     3-8
CSIMS2    1-176     3-8
CSIMS3    1-201     3-8
CSIMS4    1-198     3-8
CTMSG     1-47
CTRLTT    1-107
CURMTX    1-189
CURPRM    1-133
CVDVNM    1-186
CVTTAB    1-167
CW$50H    1-117
CXTBAS    1-111
CXTPAG    1-90
CXTWDS    1-111
DATTIM    1-165    29-70     29-99
DCCRD     1-140
DCCWR     1-140
DCTRD     1-140
DCTWR     1-140
DEADEV    1-174
DELSPC    1-188
DELWIN    8-169#   12-17
DETARG    1-203     8-15#
DETHD     1-203
DETTXT    1-182
DEVHD1    1-185
DEVIDL    1-190     1-190     1-191
DEVUNT    1-177     7-39#
DFJMEM    1-64     10-36
DIABFL    1-128    26-66
DIABLO    1-149    26-63
DIABNO    1-129    26-65
DIVIDE    1-183
DIVSOR    1-196     3-5#
DJABMS    1-194
DKCOM     7-4#     17-12     19-25     22-21
DKSAV     1-167     7-6#
DLCEMT    1-50      8-57#
DLMSG     1-194
DLTXT     1-181
DMTALL    1-194
DMTARG    1-173     8-30#
DMTSUB    1-199
DOLRAT    9-19#    15-9*     15-71*    17-28     17-129*
DORUN     1-165    19-83
DOSTOP    1-195
```

```
DOTAT     9-20#     15-10*    15-84*    17-21     17-130*
DZTXT     1-201
EDIT      1-78      10-102
EDTFIL    1-187     11-46
EM$DAA    1-53
EM$NUC    1-58      16-13
EM$NUK    1-52      21-19
EM$OVL    1-70       4-5
EM$SFP    1-70       4-4
EMCXCP    8-143#    28-87*    28-88*    28-89*    28-90     28-95*    28-96*    28-97     28-102*   28-103*   28-104    28-109*
         28-110*    28-111    28-116*   28-117*   28-118    28-120*   28-121    28-123*   28-124
ERRLOC    1-63
ERRSEV    1-137     10-51*    12-47
ERRSPC    1-107     11-41     11-45     12-4*
ESC       1-52       2-19#
FC$CDX    1-164
FC$LNK    1-164
FD$NAM    1-164
FF        2-18#     15-20     24-16
FILERM    1-75      11-43
FILNAM    1-168      1-169     7-43#    17-38     17-43     19-28     19-33     19-41*    19-43     19-76     19-80     25-16
FKILL     1-168     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    1-168     10-19     17-49
FSTDL     1-99
GAGMSG    1-193
GENMON    1-53       8-186#   10-192*   10-193    10-197*   10-198    12-74
GENTOP    1-108     10-17
GETSYP    1-60      10-140
GOTCML   15-124     16-6#
GREET     1-46       2-9#     29-9      29-17     29-36     29-79
GRT1      1-146     29-11
GRTEND   29-80      29-109#
GRTFIN    1-46      29-13     29-16     29-18     29-54     29-61#
GRTINI   10-161     10-166#
GRTVIR   29-8       29-71     29-103#
GTRD50    1-173
HANBSY    1-178
HANCHN    1-78       2-21#
HANENT    1-80
HANIDX    1-177      7-49#
HANSIZ    1-80
HAZEL     1-88      26-56
HAZLFL    1-88      26-60
HAZLNO    1-88      26-59
HIMAP     1-145
HIPRI     1-179
HNBUF     1-177      7-36#
HUPARG    1-198      8-106#
IDNCMD   13-44      16-33     18-10#
II$$SZ    1-70       8-87      8-95
II$FLG    1-58      25-28
II$NPV    1-58      25-25
II$PRV    1-58      25-24
IIBUF     1-45       1-58      3-9#      8-88      8-96     25-24     25-25     25-28
```

```
ILLCMD   1-174    15-105   19-85
ILSW2    1-94     10-68
IMADM3   1-75     26-39
IMHAZL   1-75     26-58
IMVT10   1-75     26-20    26-27
IMVT52   1-75     26-32
IN$ACT   1-158    14-15    14-17    17-30    19-51    23-14
IN$CMD   1-158    14-17    23-62    23-74
IN$CNT   1-158    14-18
INBUF    9-7#     14-32    15-13    15-127   17-124
INDABT   1-166    12-53    12-61
INDACT   1-170    23-16
INDCMD  16-8      16-19    17-6#
INDERR   1-110    12-46*   23-61*   23-71*
INDINI  17-34     19-58    22-30    23-10#
INDNAM   7-33#    23-69
INDRFL   7-41#    15-7*    15-66    15-70*   15-79    15-83*   15-96    15-98*   16-7     16-18    17-7     17-19
        17-112    17-128*
INDRUN  14-19     23-68#
INDSAV   1-158    17-23    22-10    23-68
INDSTA   1-110    14-14    17-30    23-14    23-62*   23-74*
INFOMT   1-174
INGADR   1-45      8-86#
INGEMT   1-45      8-84#
INIUKD   1-61     27-46
INPADR   1-45      8-94#
INPEMT   1-45      8-92#
INSCF   17-61     23-25    25-12#
INSSRC   1-58     25-17
INVDAT   1-203
INVDEV   1-202
INVEC    1-159
INVLDM   1-198
INVLDN   1-175
INVOPT   1-168     1-174    1-189
INVTIM   1-194
IOABFL   1-62
ISPF11   1-59     28-81    28-82
ISPF5    1-59     28-53    28-54
ISPF6    1-59     28-60    28-61
ISPF7    1-59     28-67    28-68
ISPF9    1-59     28-74    28-75
ITRMTP   1-160
JCXPGS   1-145
JCXSMS   1-201
JPWDEV   1-60     28-116
JPWFLG   1-63     28-123
JPWTYP   1-60     28-120
JS$KMN   1-70     12-75
JS$LOG   1-72     10-197
JS$ON    1-70     10-192
JSTKND   1-107
JSWLOC   1-63     11-61*   12-8*    13-5     13-18    13-41*   13-42*   13-65*   13-89    13-97*   14-5*
K52      1-78
KBMSG    1-183
KBTX     1-187
```

Cross reference table (CREF V05.04)

```
KCSIBF   1-172      9-6#
KCSIMS   1-173      3-8#
KDOCIN   1-128     12-11
KED      1-78
KEYBUF   1-47       9-3#
KEYEND   1-49       9-4#
KILEMT   1-195      8-118#
KL3CLR   1-97      12-31
KL4CLR   1-130     12-24
KMFTXT   1-48
KMNBAS   1-155
KMNCHN   1-104
KMNHI    1-89
KMNNAM   1-199      7-31#
KMNOT1   1-46      10-13     10-196    11-6#
KMNPGS   1-90
KMNSTK   1-90
KMNSTR   1-90
KMNTOP   1-90
KMPRMT   1-62       1-141    10-44*    10-45*    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#
         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#
LA120    1-149     26-50
LA12FL   1-150     26-53
LA12NO   1-150     26-52
LA36     1-149     26-44
LA36FL   1-129     26-47
LA36NO   1-129     26-46
LACTIV   1-91
LAFSIZ   1-94      12-29*
LBSPRI   1-60      28-26
LCBIT    1-149     13-42     14-5
LCOL     1-93       1-136    14-7
LCONTM   1-101     10-20*
LCPUHI   1-101
LCPULO   1-101
LD$RON   1-125
LDBASE   1-126
LDCLEN   1-166     11-24     28-21
LDDEVX   1-127
LDFLAG   1-126
LDNAM    1-172      7-34#
LDNAME   1-126
LDOPHD   1-180
```

```
LDPDEV    1-126
LDSIZE    1-126
LF        2-13#     13-72     29-43     29-108    30-27
LF$OPN    1-163
LF$WRT    1-163
LFWLIM    1-94      12-30*
LGOVER    1-47       4-8
LICTXT    1-146     29-29     29-82
LINBUF    1-89
LINCNT    1-91
LINCUR    1-94
LINFRE    1-194
LINNTX    1-77      29-103
LINNXT    1-89
LINPNT    1-91
LJSW      1-107
LMONHD    1-70      12-70
LMXLN     1-159
LMXNUM    1-157
LMXPRM    1-160
LNBLKS    1-111
LNMAP     1-117
LNPRIM    1-117     27-16     27-45
LNSBLK    1-112
LNSPAC    1-121     12-27*
LOCKTX    1-183
LOCMSG    1-76      11-49
LOGASN    1-175
LOGBLK    1-162
LOGBUF    1-162
LOGCHN    1-162      8-136
LOGCLS    1-179
LOGFLG    1-162
LOGPTR    1-162
LOMAP     1-145
LOTBUF    1-92
LOTNXT    1-92
LOTPNT    1-92
LOTSIZ    1-93
LOTSPC    1-93
LPARNT    1-118     10-175    28-11
LPRG1     1-147     11-6*
LPRG2     1-147     11-7*
LPROG     1-100     28-41     28-41*    28-43
LPROJ     1-100     28-40     28-40*    28-42
LRBFIL    1-118     12-22*
LRDTIM    1-91      12-28*
LSCCA     1-114     11-10*
LSECPT    1-105     27-33
LSTACT    1-89
LSTATE    1-143
LSTDL     1-99      10-131    10-179
LSTFRM    7-46#
LSTMX     1-157
LSTPL     1-138     10-66     10-133    29-7      29-15
LSTPRM    1-134     17-88
```

```
LSTSL    1-143
LSUCF    1-96      10-181
LSW      1-62      10-11     10-22*    10-160    10-175*   11-26     30-19
LSW11    1-87      28-80     28-82*    28-83*
LSW2     1-109     10-138    10-162*   10-167    12-23*    26-14     26-68*    28-47     28-47*
LSW2S    1-113     10-167*   12-23     26-69*    28-48     28-48*
LSW3     1-113     12-31*    13-115*
LSW4     1-130     12-24*    12-32*    13-51     13-91*    13-94*    17-57*    17-121    17-123*   17-127*   19-62*    24-49
LSW5     1-97      10-106*   10-108*   10-130*   12-25*    12-33*    12-38*    12-50*    12-66*    13-16     13-87     13-92
        13-96*     13-102    17-26     19-53     23-21*    23-73*    24-28     24-58*    28-52     28-54*    28-55*
LSW6     1-156     10-21*    11-21     11-25*    12-5      12-36     12-59     12-63     24-40     28-59     28-61*    28-62*
        29-12      29-14*
LSW7     1-160     10-114*   10-117*   10-120*   10-124*   12-34*    12-51     13-30     13-40*    13-98*    14-24     16-26
        19-14      20-4      20-31*    21-12*    21-18*    28-66     28-68*    28-69*
LSW9     1-87      11-18*    12-26*    12-35*    13-114*   17-47     28-73     28-75*    28-76*
LTRMTP   1-150     24-35     26-13
LTSCMD   1-121     11-17*
LUNAME   1-100     10-28     28-35     28-36
LWINDO   1-65      27-21
MAKWIN   8-156#    27-23
MAPWIN   8-164#    27-28
MAXASN   1-110
MAXAVL   1-177
MAXMEM   1-63      10-40*
MAXMTX   1-189
MAXPRI   1-63       1-161    10-43
MAXSEC   1-105
MDT      1-8        1-88
MHNSMS   1-193
MINTIM   1-105     10-20
MISSEQ   1-178
MISUCL   1-76      20-37
MNBASE   1-188      8-175#
MNBPC    1-187      8-177#
MNFLGS   1-187      8-178#
MNTARG   1-198      8-25#    10-145
MNTDEF   7-27#
MNTDEV   1-173      7-50#     8-26      8-31     10-144*
MNTFUL   1-175
MNTOP    1-188      8-176#
MNTTXT   1-200
MONAR1   1-188      8-174#
MONAR2   1-188      8-182#
MONHD    1-188
MONTAB   1-51
MONTXT   1-75      11-38
MONVEC   1-161
MSGBUF   1-193      8-46      8-47#
MSGEND   1-193      8-48#
MSTALC   1-71       4-6
MTOPHD   1-174
MUL32    1-203
MXCSR    1-157
MXDTR    1-157
MXJADR   1-65
MXJMEM   1-64
```

```
MXJPRI    1-161     10-43#
MXPRMT    1-62       1-141     28-89
MXVEC     1-159
NAMTOP    1-146     10-41
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
NATXT     1-181
NEWCMD   13-111     13-113    14-5#
NEWJSW    1-107
NFRESB    1-132
NINTXT    1-181
NLINES    1-119
NOABRT   11-31      12-4#
NOCF      1-76      17-49     17-52
NOCIN     1-172     12-10     12-17#
NOCMD    18-16      19-13#
NODAT     1-186
NOFIL     1-184      1-192
NOFLAG    1-174      9-18#
NOFRDL    1-203
NOICMD   17-40      17-52#
NOIND     1-178     17-25     22-12
NOLDMT    1-186
NONEMS    1-186
NOOPTT    1-192
NOPMGN    1-188
NOPRG     1-168      1-172
NOSTRT    1-171
NOTAVL    1-187
NOTON     1-193
NOTXT     1-181
NOUCL     9-22#     13-43*    15-12*    16-32*    19-16     20-6
NOUDC     1-185
NSPLDV    1-125     10-79
NSWPMS    1-189
NUCHN     1-118
NUMDEV    1-152
NUMMNT    9-23#
NUMON     1-94
NUMPRM    1-76      17-134
OCTFIX    1-181
OCTPRT    1-201     11-51
OF$$SZ    1-154
OF$DEV    1-153
OF$FIL    1-153
```

```
OF$FLG   1-153
OF$UNT   1-153
OFFEMT   1-195     8-100#
OKFEND   1-106
OKFILE   1-106
OPRCMD   1-84      6-61
OT$RON   1-154
OTHRON   1-197
OTRMNT   1-199
OVLYEQ   1-201    11-55
OVRCOR   1-171
PO$DBG   1-69
PO$SPV   1-69
P2$VIR   1-57     10-70
PASLIN   1-137    12-8      13-5      13-41     13-65     13-97
PAUMSG   1-167
PBFEND   1-122    17-103*
PBUFND   1-48      9-17#
PF$IOW   1-156
PF$SYS   1-156
PLOAD    1-165    20-33     21-14     23-76     24-60
PMBUSY   1-188
PNAME    1-152     1-177
POPCF    1-129    17-51     19-56     19-67
PPNMSG   1-47
PRGALL   1-165    11-16     11-23     12-58
PRGSIZ   1-89
PRGTOP   1-89
PRIEMT   8-151#   28-25
PRIVAO   1-69     10-60*
PRIVA2   1-57     10-71*
PRIVCO   1-69     10-63*    10-74*
PRIVFO   1-59     10-62*    10-73*    25-24*    25-25*
PRIVSO   1-69     10-61*
PRIVS2   1-57     10-72*
PRMBUF   1-134    17-87
PRMEND   1-134    17-95
PRMPNT   1-133    17-86
PROFLG   1-86     29-34     29-61
PRSCMD  15-6#
PRTBUF   1-186     9-16#
PRTDAT   1-202
PRTDC2   1-183
PRTDC3   1-183
PRTDEC   1-177    29-68     29-106
PRTFIX   1-180
PRTFNM   1-186
PRTGRT   1-46     10-166    29-7#
PRTLN    1-182
PRTPCT   1-196
PRTR50   1-202    11-57
PRTSPC   1-180
PRTSUP  29-92#
PRTTIM   1-202
PRTTMV   1-184
PRTTOD   1-202
```

```
PRTTTP   1-181
PRTUNM   1-182
PTL     17-96     17-136#
PUSHCF   1-169    13-57     17-41     19-31     30-14
PVNPW    1-69     10-57     25-23
PVON     1-97
QUME     1-149    26-63
QUMEFL   1-151
QUMENO   1-151
QUOTFL   7-42#    15-8*     15-31     15-33*    15-35     15-41*
R50BUF   1-173     7-45#
R50CHR   1-49
R50COM   1-170     7-25#
R50DIR   1-169     7-16#
R50DK    1-202     7-8#
R50DSK   1-175     7-11#
R50DUP   1-170     7-17#
R50IND   1-169     7-18#
R50K52   1-170     7-13#
R50KED   1-170     7-12#
R50KEX   1-170     7-14#
R50KMN   1-61      7-35#    11-6      11-7
R50LD    1-175     7-22#
R50LDO   1-173     7-23#
R50LD7   1-175     7-24#
R50LOG   1-179     7-10#
R50MON   1-203     3-7#
R50NO    1-176     7-37#
R50PIP   1-170     7-15#
R50SAV   1-169     7-26#
R50SY    1-169     7-9#     10-144    19-41
R50TSX   1-166     7-19#
R50TT    1-198     7-7#
R50UCL   1-166     7-20#
R50VIR   1-171     7-21#
RDB      1-142
RDBEND   1-142
RDCMD    1-167     6-76      6-77      6-79      6-81     14-8      14-13#    15-114    17-15     17-132    19-75     22-24
RDERM    1-171     4-36
RDREST  17-77     17-112#
REGEN    1-75     10-19
REMNDR   1-49      3-6#
RESDEV   1-154
RJEMT    1-143     8-130#
RNMS     1-182
RONTXT   1-187
RSR      1-157
RSTPRV   1-70     10-75     11-20     25-29
RT$$SZ   1-142
RT$NAM   1-142
RUNARG   1-69     11-12*
RUNCHN   1-66     11-19     11-19     19-80     19-80     20-16     20-16     21-7      21-7      23-68     23-68     24-55
        24-55
RUNDEV   1-111
RUNEMT   1-171     8-4#
RUNFLG   1-112
```

```
RUNHD     1-167
RUNMS     1-203
RUNNAM    1-66      23-64
S$INWT    1-103
S$IOFN    1-98
S$IOWT    1-147
S$MSWT    1-104
S$OTFN    1-98
S$OTLO    1-98
S$OTWT    1-103
S$QUSR    1-147
S$SFWT    1-103      1-147
S$SPCB    1-148
S$SPDB    1-148
S$SPND    1-97
S$TMWT    1-103
S$TTFN    1-98
S$TWFN    1-98
SBPSUF    1-60      10-184    28-109
SC$ERR    1-57      10-51
SCHAIN    1-127     12-8      13-5      13-18     13-41     13-65     13-89     13-97
SCNCMD   15-13#     15-128    17-131
SD$BAK    1-130
SD$DEL    1-123
SD$FLK    1-124
SD$HLD    1-133
SD$SNG    1-132
SD$WFM    1-124
SDBLK     1-125
SDBU      1-130
SDBUF1    1-125
SDCB      1-138
SDCBND    1-138
SDCBSZ    1-143
SDFHD     1-135
SDFLAG    1-124
SDFORM    1-124
SDSFCB    1-123
SDSKIP    1-130
SEARCH    1-167     18-12
SERFLG    1-62      11-11*    17-42*    17-44*    19-32*    19-34*    19-42*    19-44*    19-79*    19-81*
SETHD     1-176
SETSUF   10-187     13-117    30-9#
SF$1ST    1-133
SF$BSY    1-132
SF$HLD    1-133
SFFILE    1-148
SFFLAG    1-135
SFFORM    1-132
SFNMBL    1-132
SFQLNK    1-135
SFUSER    1-148
SH$$SZ    1-79
SH$FLG    1-79
SH$NAM    1-79
SH$RTN    1-79
```

```
SH$VAL    1-79
SHMTH1    1-185
SHMTH2    1-185
SHOHD     1-200
SHTMSG    1-195
SIZEMT    1-198     8-20#
SIZVAL    1-172     8-21#
SJEMT     1-143     8-124#
SKPSPC    1-59     22-17
SMONHD    1-71     12-72
SO$NO     1-80
SO$NVL    1-80
SO$OCT    1-80
SPACE2    1-182
SPACE3    1-182
SPACE5    1-183
SPACE6    1-186
SPACTV    1-190     1-191
SPCF      1-192
SPFLK     1-192
SPFUL     1-192
SPGEMT    1-192     8-9#
SPLACT    1-195
SPLCHN    1-102    10-81
SPLHD     1-189
SPLHLA    1-180     8-136#
SPLPND    1-197
SPSNG     1-190     1-192
SPUBUF    1-65     10-42
SPWFM     1-190     1-191
SRTSMS    1-197
SRTTXT    1-200
SSRMAP    1-200
START     1-48      8-114     10-5#      30-38
STDNAM    1-75     10-46
STLGCN    1-165
STLGHD    1-179
STPASK    1-197
STPFLG    1-102
STRLEN    1-76     17-136
SUBARO    1-187
SUBTXT    1-200
SUCF2     1-57     13-110     13-116     13-118*
SUCS      1-146    29-53
SUM1      1-196
SUM2      1-196
SUM3      1-196
SUM4      1-196
SUM5      1-197
SUM6      1-197
SUM7      1-197
SUMS      1-146    29-22
SUPCOD    1-146    29-92
SWPTX     1-183
SXBPNT    1-65     10-42*
SYCOM     7-3#
```

```
SYHD1     1-182
SYHD2     1-182
SYINDX    1-152
SYINTX    9-25#     23-33
SYNAME    1-153
SYSAV     1-167      7-5#      19-72      23-63
SYSDAT    1-144
SYTIMH    1-144
SYTIML    1-144
SYTXT     9-24#     19-50
SYUNIT    1-152
TAB       2-17#     15-18      15-117     24-14
TALEMT    1-50       8-62#
TBLOVF    1-176
TECO      1-78      10-104
TK1VAL    1-144
TM$LN1    1-77      29-66
TMIDLH    1-67
TMIOH     1-67
TMIOWH    1-66
TMSWPH    1-67
TMSWTH    1-67
TMTOTH    1-66       1-196
TMTOTL    1-66       1-196
TMUSRH    1-66
TOMPRM   17-89      17-134#
TOOLNG    1-76      13-66
TOTMMS    1-200
TOTON     1-102
TOTXT     1-179
TRGRET    1-146     29-38
TRMINI   10-150     26-9#
TRMSTR    1-169
TRYUCL   19-82      20-4#
TSKMN1    1-11#      1-51
TSKMON    1-10#      1-46       1-51
TSR       1-157
TSXLN     1-146     29-75
TSXSIT    1-146     29-67      29-75*
TSXSMS    1-201
UC$MDC    1-164     10-96*
UC$NDC    1-164     10-95*
UCHAN     1-116     11-13      11-15*
UCIDEF    1-52       3-4#
UCISPC    1-93      21-6       28-102
UCLBLK    1-163     10-86      16-11      16-28      19-18
UCLCMD    1-85       6-74
UCLDAT    1-163     10-88
UCLNAM    1-121     20-16      20-28
UERSEV    1-137     11-35*     12-46      12-47      12-54*     23-61      23-71
UFORM     1-109     10-47      28-95
UFPTRP    1-123     11-9*
UHIMEM    1-111     10-41*
UKMNAM    1-96
UMSSMS    1-200
UMSYTP    1-99
```

```
TSKMN1 -- TSX-Plus Keyboard Mon MACRO V05.04  Monday 21-Dec-87 11:40 Page S-18
Cross reference table (CREF V05.04)


UNSUP    1-77     29-98
UPPN     1-100    28-42#    28-43#
UPTMMS   1-195
URCMD    1-77     20-41
URERR   20-5      20-7      20-12     20-41#
USPLCH   1-102    10-81     10-81
USRMMS   1-201
USRSTK   1-64
USTART   1-108
UTRPAD   1-63     11-8#
VCORTM   1-141
VEDIT    1-68     10-102    10-104
VHIPCT   1-139
VIRINI  10-156    27-10#
VQUAN1   1-139
VQUAN2   1-139
VQUN1A   1-139
VT100    1-149    26-16
VT10FL   1-151    26-19
VT10NO   1-151    26-18
VT2007   1-155    26-23
VT2008   1-155    26-23
VT20FL   1-155    26-26
VT20NO   1-155    26-25
VT52     1-149    24-35     26-30
VT52FL   1-150    26-34
VT52NO   1-129    26-33
VU$CL    1-121    20-11
VUCLMC   1-122    10-96
VUCLOR   1-123    10-112    10-115    10-118
WILDFL   1-68     10-128
WRNHED   1-52
XAREA    1-168     7-44#    10-81     10-88     10-97     17-43     17-65     19-33     19-43     19-80     20-16     21-7
        23-68     24-55
YELEMT   1-194     8-44#
YESTXT   1-181
ZCLR     1-157
```

```
TSKMN1 -- TSX-Plus Keyboard Mon MACRO V05.04  Monday 21-Dec-87 11:40 Page M-1
Cross reference table (CREF V05.04)

...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  10-81    10-88    10-97    17-43    17-65    19-33    19-43    19-80    20-16    21-7     23-68    24-55
...CM2  10-81    10-88    10-88    10-97    10-97    10-97    10-97    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  10-98    11-19    19-55
...CM5  10-81    10-88    10-97    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  10-97    17-65
.CLOSE   1-39#   10-98    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#   10-88    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#   10-81    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#   10-97
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
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#    10-19    17-49
TBLDEF   5-26#    6-8
TBLEND   5-52#    6-85
```