TSKSHO -- Keyboard SHOW Command MACRO V05.05 Friday 20-Jan-89 10:51
Table of contents


```
    4-    1       Data areas
    5-    1       SHOW command
    6-    1               .       ALL
    8-    1               .       MEMORY
    9-    1               .       TERMINALS
   10-    1               .       CL
   11-    1               .       JOBS
   12-    1               .       QUEUE
   13-    1               .       COMMANDS
   14-    1               .       DEVICES
   18-    1               .       ASSIGNS
   19-    1               .       ALLOCATIONS
   20-    1               .       MOUNTS
   21-    1               .       MODEM
   22-    1               .       DATE
   22-   15               .       TIME
   22-   23               .       VERSION
   22-   39               .       USE
   23-    1               .       INSTALL
   25-    1               .       REGIONS
   27-    1               .       PRIVILEGES
   28-    1               .       SL
   30-    1               .       RUN-TIMES
   31-    1               .       SPOOL
   32-    1               .       SUBSET
   33-    1               .       VM
   34-    1       SYSTAT (& WHO) command
   34-    9       USE command
   35-    1       PRTUSE -- Print system usage statistics
   36-    1       MEMORY command
```

```
                                        .TITLE  TSKSHO -- Keyboard SHOW Command Routines
                                        .ENABL  LC
                                        .DSABL  GBL
000000                                  .CSECT  TSKSHO
000000                          TSKSHO:
                                ;
                                ;  TSKSHO is the portion of TSKMON that contains the code
                                ;  to implement the SHOW command.
                                ;
                                ;  Copyright 1978, 1979, 1980, 1981, 1982, 1983, 1984,1985.
                                ;  S&H Computer Systems, Inc,
                                ;  Nashville, Tennessee
                                ;
                                ;  Macro calls
                                ;
                                        .MCALL  .CSISPC,.TTOUTR,.SRESET
                                        .MCALL  .READW,.TTYIN,.TTYOUT,.PURGE
                                        .MCALL  .CSIGEN,.SAVEST,.REOPEN
                                        .MCALL  .GTLIN,.GTIM,.DATE,.SPFUN
                                        .MCALL  .PRINT,.CLOSE,.LOOKUP
                                        .MCALL  .WRITW,.ENTER,.EXIT
                                        .MCALL  .SERR,.HERR,.FPROT,.GVAL,.PVAL
                                ;
                                ;  Global definitions
                                ;
                                        .GLOBL  TSKSHO,CMDHD,CMDOFF,KDOCIN,SKPSPC,UCLCMD
                                        .GLOBL  DORUN,CMDFRM,CMDDSN,STLGCN,DATTIM,PRGALL
                                        .GLOBL  DLCEMT,ALCDEV,CMDSHO,CMDSET,CMDWHO,CMDMEM,CMDUSE
                                ;
                                ;  Global references
                                ;
                                        .GLOBL  AF$DUP,AF$IND,AF$UCL,AF$SET,AF$CCA,TSXVRS,AF$NPW
                                        .GLOBL  CKCLUS,SHOKEY,HANIOC,SYPSWD,TM$NSP,TM$SL1,TM$OFF,TM$ON
                                        .GLOBL  TM$KED,TM$TTY,TM$SUB,TM$NO,VMXMSG,VMAXMC,VMXMRB
                                        .GLOBL  PEKEMT,PEKADR,PEKSIZ,TM$NNR,CDBUF,CDGET,TM$IN1,TM$IN2
                                        .GLOBL  CLRPRV,OPTLST,PFSO,PFCO,PVNPW,TSXVER
                                        .GLOBL  PO$SYS,PO$SPV,PRIVSO,PRVOPT,KBPARN,PARNNL
                                        .GLOBL  TM$PVA,TM$PVC,PRIVAO,EM$CNO,EM$CPO,EM$CAP,RSTPRV
                                        .GLOBL  CHKEQ,CKACOJ,PO$OPR,CKSYPV,AF$BYA,TM$PVL,DMYDEV,AF$TPO
                                        .GLOBL  INSTBL,INGADR,INGEMT,IIBUF,II$NAM,II$FLG,II$$SZ,EM$NAD
                                        .GLOBL  INSTBN,AF$SCA,AF$NOW,AF$MEM,PO$DBG,PRIVCO,PRVLST
                                        .GLOBL  ABRTAD,ABRTCD,CINFLG,$VNOTT,VPRILO,VPRIHI,II$PRV,II$NPV
                                        .GLOBL  TM$RD1,TM$RD2,TM$LCL,TM$GBL,SPACE1,RC$OWN,RC$CNT,RC$BAS
                                        .GLOBL  RC$EXC,RC$AGE,RC$AEP,RC$USE,RC$FLG,RC$GBL,RC$NAM
                                        .GLOBL  RC$LEN,RC$PVT,RCBBAS,RCBEND,RC$$SZ,SHRRCB,SHRRCN
                                        .GLOBL  LP$SPD,LP$PAR,LP$ODD,LP$7BT,EM$ICL,PROSLT,RC$LCG
                                        .GLOBL  EM$NPD,EM$ILN,EM$CIP,EM$NSF,EM$IUN,EM$CLN
                                        .GLOBL  EM$ILN,EM$ACL,EM$TSL,EM$CLB,EM$NSL,EM$SLT,EM$SLW
                                        .GLOBL  SLKDON,SLKDOF,EM$UIO,TM$PR1,TM$PR2,TM$LPR,TM$HPR
                                        .GLOBL  TM$HPE,TM$CNG,TM$CDS,TM$CEN,TRMHD1,TRMHD2
                                        .GLOBL  OPRTXT,CLLINE,LCLTXT,REMTXT,TM$AUT,CLFREE,CLUNIT,CLVERS
                                        .GLOBL  TM$CL0,TM$CL1,TM$CL2,TM$CL3,TM$CL4,TM$CL5,TM$CL6,TM$CL7
                                        .GLOBL  TM$CL8,QHDMS1,QHDMS2,DVSHH1,DVSHH2,DVSHH3,SYASHD,DKASHD
                                        .GLOBL  TM$NAD,ALCHD1,ALCHD2,TM$NSD,TM$SDN,LNAME,TM$C13
                                        .GLOBL  CORUSR,LSW,$CTRLO,SERFLG,IOABFL,$CHACT,$STSNG
                                        .GLOBL  LSTHL,LCLUNT,FSTIOL,LSTIOL,CL$LIX,CW$PRO,CONFG2
                                        .GLOBL  CL$RQH,CL$WQH,MAXALC,ALCTBL,ALCEND
```

```
                .GLOBL  AD$DVU, AD$JOB, AD$$SZ, UCIDEF, HANCHN
                .GLOBL  NEDCHR, LOUTIR, LINIR, LINRTS, CLOTIR
                .GLOBL  CO$DEF, CL$COL, LCDTYP, SOPALC, SOPDAT, SOPTIM
                .GLOBL  UTRPAD, JSWLOC, ERRLOC, MAXMEM, MAXPRI
                .GLOBL  USRSTK, $KINIT, CFSTK, MXJMEM, DFJMEM, EM$HNI
                .GLOBL  SPUBUF, SXBPNT, VSWPFL, MXJADR, CLSFCH
                .GLOBL  TMTOTH, TMTOTL, TMUSRH, TMIOWH, LDMNT, EM$CSE
                .GLOBL  TMSWTH, TMIDLH, TMIOH, TMSWPH, LDCLEN
                .GLOBL  WILDFL, $NOIN, $NOWTT, $HITTY
                .GLOBL  TECO, EDIT, KED, K52, $1STLG, $DIBOL
                .GLOBL  SH$VAL, SH$NAM, SH$$SZ, SH$RTN, SH$FLG
                .GLOBL  SO$NVL, SO$OCT, SO$NO, HANENT, HANSIZ
                .GLOBL  H.CSR, H.VEC, DVSTAT, SFID, ACRSPD, HANPAR, LSTSPL
                .GLOBL  HAZEL, HAZLFL, HAZLNO, $MLOCK, MDT, GETKCH
                .GLOBL  LINBUF, LINNXT, LSTACT, PRGTOP, PRGSIZ, KMNHI
                .GLOBL  KMNTOP, KMNPGS, KMNSTK, KMNSTR, CXTPAG, FSTIOL
                .GLOBL  LINPNT, LINCNT, LACTIV, LRDTIM, CS$RON
                .GLOBL  LOTBUF, LOTNXT, LOTPNT, $VTESC
                .GLOBL  LOTSIZ, LOTSPC, LCOL, $SLKED, ESC
                .GLOBL  LAFSIZ, LFWLIM, LINCUR, NUMON, ILSW2
                .GLOBL  VPRIDF, VPRIVR, $DBKMN
                .GLOBL  $CARUP, DOASGN, UKMNAM, $UKMON, LSW9
                .GLOBL  LSUCF, $CCLRN, EM$NUK, S$QMIO, S$$RUN
                .GLOBL  KL3CLR, $PRGLK, LSW5, PVON, S$SPND, $AUTO
                .GLOBL  S$TWFN, S$TTFN, S$OTFN, S$IOFN, S$OTLO
                .GLOBL  LSTDL, FSTDL, $DETCH, UMSYTP, S$TTSC
                .GLOBL  $DISCN, LPROJ, LPROG, LUNAME, S$RT, S$LOW
                .GLOBL  LCPUHI, LCPULO, LCONTM, $CTRLS, $SPLJB, TXTCL
                .GLOBL  STPFLG, TOTON, USPLCH, SPLCHN, S$HICP, TXTC1
                .GLOBL  S$INWT, S$OTWT, S$TMWT, S$SFWT, S9600
                .GLOBL  S$MSWT, CFBUF, CFEND, CCLSAV, KMNCHN
                .GLOBL  MINTIM, LSECPT, MAXSEC, $EMTTR, VCSHNB
                .GLOBL  OKFILE, OKFEND, $CLTST, UCISPC, MHNSIZ
                .GLOBL  CASTBR, CASCBR, CASTBW, CASCUP, MHNSMS
                .GLOBL  CASTRO, CASTWO, CLTOTL, CO$DTR, CLSFSP
                .GLOBL  CO$CR, CO$FF, CO$FFO, CO$LC, CO$TAB, CO$CTL
                .GLOBL  CO$LFI, CO$LFO, CO$BNI, CO$BNO, CL$OPT
                .GLOBL  CL$LEN, CL$SKP, CL$WID, CL$LIN, PHYMEM
                .GLOBL  LJSW, CTRLTT, NEWJSW, JSTKND, VIMAGE
                .GLOBL  USTART, GENTOP, BOTDEV, BOTUNI, CSHALC
                .GLOBL  $CTRLC, LSW2, $INKMN, CHAIN, UFORM
                .GLOBL  $SGQO, $SGQ3, LITIME
                .GLOBL  MAXASN, AT$$SZ, $CFABT, INDSTA, INDERR
                .GLOBL  RUNDEV, LNBLKS, CXTBAS, CXTWDS, UHIMEM
                .GLOBL  ASNTBL, $DILUP, CSHDEV, CSHDVN, LNSBLK
                .GLOBL  ASNEND, LSW3, LSW2S, $DUPRN
                .GLOBL  $FORM, $TAB, LSCCA, $CFSOT, LOFSPC, R50COM
                .GLOBL  $PAGE, $SCOPE, $ECHO, $LC, $8BIT, CHKALC
                .GLOBL  UCHAN, $FORMO, $CFALL, $CFDCC, $CFCCL
                .GLOBL  LNPRIM, LNMAP, CW$50H, CONFIG, $SUCF
                .GLOBL  $DOOFF, NUCHN, LRBFIL, CFIND, TALEMT
                .GLOBL  C.CSW, C.DEVQ, C.SBLK, NLINES, CO$8BT
                .GLOBL  CD$NAM, CD$DVU, CD$BAS, CD$JOB, CD$$SZ, CD$$UB
                .GLOBL  LTSCMD, LNSPAC, CFNEST, UCLNAM
                .GLOBL  $CFOPN, CFSEND, PBFEND, CFSP, $TTGAG
                .GLOBL  UFPTRP, SDSFCB, SD$DEL, CFLFL4, $UCLCF
                .GLOBL  SDFLAG, SD$FLK, SD$WFM, SDFORM, $UCLRN
```

```
                .GLOBL  SDBUF1,SDBLK,NSPLDV,LD$RON,$UCLCM,$UCLCL
                .GLOBL  LDNAME,LDSIZE,LDFLAG,LDBASE,LDPDEV
                .GLOBL  LSW8,$SGQ1,$SGQ1A,$SGQ1B,$SGQ1C,$SGQ2,$SGIIO,$SGHIO
                .GLOBL  $DEFER,CFCHAN,SCHAIN,LDDEVX,$SGALL
                .GLOBL  CFPNT,CFBLK,$QUIET,DIABFL
                .GLOBL  DIABNO,VT52NO,LA36NO,LA36FL
                .GLOBL  LSW4,KL4CLR,SDSKIP,SDBU,SD$BAK
                .GLOBL  $INCOR,$KED,VQUN1B,VINTIO,VQUN1C
                .GLOBL  SF$BSY,SFFORM,SD$SNG,SFNMBL,NFRESB
                .GLOBL  SD$HLD,SF$HLD,CURPRM,PRMPNT,SF$1ST
                .GLOBL  LSTPRM,PRMBUF,PRMEND,CFSPND
                .GLOBL  SDFHD,SFFLAG,SFQLNK,CFHOLD,LOGDVU,LOGBAS
                .GLOBL  LCOL,$QTSET,$TECO,CD$TOP,LOGCHK
                .GLOBL  $WILD,ERRSEV,UERSEV,PASLIN,LOGBAS,LOGDVU
                .GLOBL  LSTPL,SDCB,SDCBND,VQUAN0,VQUAN3
                .GLOBL  VQUAN1,VQUN1A,VQUAN2,VHIPCT,VQUAN0,VQUAN3
                .GLOBL  DCTRD,DCCRD,DCTWR,DCCWR
                .GLOBL  AT$LOG,AT$SIZ,AT$DEV,AT$FIL,AT$EXT,AT$$SZ
                .GLOBL  VCORTM,NUMDCD,KMPRMT,MXPRMT,C1DEVX
                .GLOBL  RDB,RDBEND,RT$DEV,RT$NAM,RT$$SZ,CLDEVX,SDDVU
                .GLOBL  SDNAME,SDCBSZ,LSTSL,LSTATE
                .GLOBL  TK1VAL,CINDAT,SYSDAT,SYTIMH,SYTIML
                 GLOBL  BASMAP,LOMAP,HIMAP,JCXPGS
                .GLOBL  SMRSIZ,SRTSIZ,CSHSIZ,TK1SEC
                .GLOBL  TSXLN,TSXSIT,GRT1,TRGRET,LICTXT,SUPCOD,NAMTOP,SUMS,SUCS
                .GLOBL  LPRG1,LPRG2,S$QUSR,S$IOWT
                .GLOBL  S$SPDB,S$SPCB,SFUSER,SFFILE,VT200,VT2007,VT2008
                .GLOBL  LCBIT,LA36,LA120,VT52,VT100,DIABLO,QUME
                .GLOBL  ADM3A,LTRMTP,LA12FL,LA12NO,VT52FL
                .GLOBL  VT10FL,VT10NO,QUMEFL,QUMENO,ADM3FL
                .GLOBL  VT20FL,VT20NO
                .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,ODTBAS,$CTRLD
                .GLOBL  LSW6,$SNWTT,PF$SYS,PF$IOW,$DEBUG
                .GLOBL  RSR,TSR,LMXNUM,LSTMX,MXDTR,ZCLR,MXCSR
                .GLOBL  $INDDF,$INDRN,IN$ACT,IN$CNT,IN$CMD,INDSAV
                .GLOBL  $PHONE,INVEC,LMXLN,MXVEC,$INIT,$DEAD,$HARD
                .GLOBL  ITRMTP,LMXPRM,LSW7,$INDAB,CFSTS,CF$IND,CF$QUT
                .GLOBL  CFABLV,MONVEC,LBSPRI,MAXPRI,MXJPRI,LPRI
                .GLOBL  LOGCHN,LOGFLG,LOGPTR,LOGBUF,LOGBLK
                .GLOBL  LF$OPN,LF$WRT,UCLBLK,UCLDAT
                .GLOBL  CSHHD,FC$CDX,FC$LNK,FD$NAM,UC$NDC,UC$MDC,CVTUC
                .GLOBL  CMDBUF,PAUMSG,RDCMD,DKSAV,SYSAV,CVTTAB,RUNHD,SEARCH
                .GLOBL  FKILL,ABRTCF,ACRFN,XAREA,FILNAM,NOPRG,FPRINT
                .GLOBL  PUSHCF,TRMSTR,FILNAM,R50DIR,R50SY,R50IND,R50SAV
                .GLOBL  INDACT,R50DUP,R50PIP,R50KED,R50K52,R50KEX,R50TSX,R50UCL
                .GLOBL  BLKO,RDERM,R50VIR,NOSTRT,RUNEMT,OVRCOR
                .GLOBL  BADSAV,LDNAM,NOPRG,NOCIN,SIZVAL,ASKLNM,BADCMD,KCSIBF
                .GLOBL  ASDEX,KCSIMS,ASNOVF,GTRD50,R50BUF,R50LDO,MNTDEV,DMTARG
                .GLOBL  DEADEV,CHKMNT,CHKMTX,INFOMT,NOFLAG,MTOPHD,ILLCMD
                 GLOBL  R50LD,INVLDN,R50DSK,ACRFIL,BDFNAM,LOGASN,MNTFUL,R50LD7
                .GLOBL  TBLOVF,SETHD,CSIMS2,CKPRIV,R50NO,AMBOPT,ACRDEC
                .GLOBL  MAXAVL,PRTDEC,DEVUNT,PNAME,HNBUF
                .GLOBL  ACROCT,HANBSY,CSIMS1,MISSEQ,NOIND,POPCF
                .GLOBL  BADPMT,BADPRI,TOTXT,CRLF,HIPRI,STLGHD,LOGCLS,R50LOG
```

```
                        .GLOBL  BDLGOP,SPLHLA,NOCCL,LDOPHD,PRTFIX,PRTSPC
                        .GLOBL  DLTXT,OCTFIX,PRTTTP,NATXT,SPDTX1,NOTXT,YESTXT,NINTXT
                        .GLOBL  PRTUNM,SYHD1,SYHD2,PRTLN,SPACE2,DETTXT,SPACE3,RNMS,WTMS
                        .GLOBL  SWPTX,LOCKTX,SPACE5,PRTDC3,KBMSG,DIVIDE,PRTDC2
                        .GLOBL  COLOO,CPUAH,CPUAL,PRTTMV,NOFIL,CMDBUF,CALUCL
                        .GLOBL  NOUDC,DEVHD1,ASNHD1,ASNHD2,SHMTH1,SHMTH2,PRTTMD
                        .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,AMBOPT,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
                        .GLOBL  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,NOKMON,CCLNAM,OTRMNT,CHKDEV,DMTSUB,CMDCCL
                        .GLOBL  SHOHD,SUBTXT,MNTTXT,SRTTXT,TOTMMS,UMSSMS,SSRMAP
                        .GLOBL  TSXSMS,USRMMS,JCXSMS,DZTXT,OCTPRT
                        .GLOBL  PRTR50,PRTDAT,PRTTOD,PRTTIM,INVDEV,ALFN,R50DK
                        .GLOBL  DETHD,DETARG,RUNMS,NOFRDL,R50MON,INVDAT,MUL32,COAF
                        .GLOBL  AR$PRJ,AR$PRG,AR$CON,AR$CNT,AR$CPH,AR$CPL,AR$UNM
                        .GLOBL  AR$DMY,AR$$SZ,ARNRPB,$SLON,$SLTTY,$SLLET
                        .GLOBL  PRTWRN,SLMXLN,$LOFCF,CSHMSG,$CARMN,VUSPHN
                        .GLOBL  AF$HIE,AF$NOI,$NOINT,AF$PLK,AF$DBG
                        .GLOBL  AF$IOP,$RNIOP,SHVTX1,SHVTX2,SHVTX3,SHVTX4,SJSPPN
                        .GLOBL  VONTM,VOFFTM,VTMIN,VTMOUT,VTMLOC
                        .GLOBL  MDMTXT,OFTTXT,ONTTXT,TMITXT,TMOTXT,TMLTXT,PHNTXT
```

```
                        ;
                        ;  Assembly constants
                        ;
000012          LF      =       12              ;LINE FEED
000015          CR      =       15              ;CARRIAGE RETURN
000040          BLANK   =       40              ;ASCII SPACE
000007          BELL    =       07              ;ASCII BELL
000011          TAB     =       11              ;HORIZONTAL TAB
000014          FF      =       14              ;FORM FEED
000054          COMMA   =       54              ;COMMA
000400          BLKWDS  =       256.            ;# OF WORDS IN DISK BLOCK
132500          WLDNAM  =       132500          ;RAD50 /*/ (WILDCARD)
```

```
                        ;-----------------------------------------------------
                        ;  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 print a warning message
                        ;
                              .MACRO  FWARN   MSG
                              MOV     R5,-(SP)
                              MOV     MSG,R5
                              CALL    PRTWRN
                              MOV     (SP)+,R5
                              .ENDM   FWARN
                        ;
                        ;-------------------------------------------------------------
                        ;  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  CMDVSH
                        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  NAMESH
                        L     =       .
                              .ASCIZ  /STRNG/
                              .CSECT  CMDVSH
                              .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  CMDVSH
                                        .WORD   0
                                        .CSECT  TSKSHO
                                        .ENDM   TBLEND
                                ;-----------------------------------------------------------------------
                                ;  Define options for SHOW command
                                ;
000000                                  TBLDEF  SHO,1
000002                                  CMDDEF  ALL,SOPALL
000006                                  CMDDEF  ALLO*CATE,SOPALC
000012                                  CMDDEF  ALLO*CATIONS,SOPALC
000016                                  CMDDEF  AS*SIGNS,SOPASN
000022                                  CMDDEF  CA*CHE,SOPCSH
000026                                  CMDDEF  C*ONFIGURATION,SOPCON
000032                                  CMDDEF  CL,SOPCL
000036                                  CMDDEF  COM*MANDS,SOPCMD
000042                                  CMDDEF  COR*TIM,SOPCTM
000046                                  CMDDEF  C1,SOPCL
000052                                  CMDDEF  D*EVICES,SOPDEV
000056                                  CMDDEF  DA*TE,SOPDAT
000062                                  CMDDEF  E*RRORS,RDCMD
000066                                  CMDDEF  H*IPRCT,SOPHIO
000072                                  CMDDEF  INS*TALL,SOPINS
000076                                  CMDDEF  INT*IOC,SOPIIO
000102                                  CMDDEF  J*OBS,SOPJOB
000106                                  CMDDEF  KEY*S,SHOKEY
000112                                  CMDDEF  LD,SOPSUB
000116                                  CMDDEF  M*EMORY,SOPMEM
000122                                  CMDDEF  MAXMC,SOPMC
000126                                  CMDDEF  MAXMRB,SOPMR
000132                                  CMDDEF  MAXMSG,SOPMB
000136                                  CMDDEF  MO*UNTS,SOPMNT
000142                                  CMDDEF  MOD*EM,SOPMDM           ;Synonym with PHONE
000146                                  CMDDEF  N*UMDC,SOPNDC
000152                                  CMDDEF  PH*ONE,SOPMDM           ;Synonym with MODEM
000156                                  CMDDEF  PRI*ORITY,SOPPRI
000162                                  CMDDEF  PRIL*OW,SOPPLO
000166                                  CMDDEF  PRIH*I,SOPPHI
000172                                  CMDDEF  PRID*EF,SOPPDF
000176                                  CMDDEF  PRIV*ILEGES,SOPPRV
000202                                  CMDDEF  PRIVIR,SOPPVR
000206                                  CMDDEF  Q*UEUE,SOPQUE
000212                                  CMDDEF  QUAN0,SOPQ0
000216                                  CMDDEF  QUAN1,SOPQ1
000222                                  CMDDEF  QUAN1A,SOPQ1A
000226                                  CMDDEF  QUAN1B,SOPQ1B
000232                                  CMDDEF  QUAN1C,SOPQ1C
000236                                  CMDDEF  QUAN2,SOPQ2
000242                                  CMDDEF  QUAN3,SOPQ3
000246                                  CMDDEF  REG*IONS,SOPREG
000252                                  CMDDEF  RUN*-TIMES,SOPSRT
000256                                  CMDDEF  S*UBSETS,SOPSUB
000262                                  CMDDEF  SL*E,SOPSLE
```

```
    115 000266                          CMDDEF  SP*OOL,SOPSPL
    116 000272                          CMDDEF  SYSP*ASSWORD,SOPSYP
    117 000276                          CMDDEF  T*ERMINALS,SOPTRM
    118 000302                          CMDDEF  TI*ME,SOPTIM
    119 000306                          CMDDEF  TT*Y,SOPTRM
    120 000312                          CMDDEF  USE,SOPUSE
    121 000316                          CMDDEF  USERS,SOPJOB
    122 000322                          CMDDEF  VE*RSION,SOPVER
    123 000326                          CMDDEF  VM,SOPVM
    124 000332                          TBLEND
```

```
                                        .SBTTL  Data areas
                                ;-------------------------------------------------------------------------------
                                ;  Data areas
                                ;
000000  075250  014644  000000  HANNAM: .RAD50  /SY ddd   TSX/  ;File spec for device handler
000006  100020
000010  000000                  DVEC:   .WORD   0
000012  000000                  DCSR:   .WORD   0
000014  012276                  R50CLO: .RAD50  /CLO/
000016  000000                  CDBASE: .WORD   0
000020  000000                  CDDVU:  .WORD   0
000022  000000                  CDBAS5: .WORD   0
                                ;
                                ;  Table of terminal device type names.
                                ;  Note, the order of the entries in this table must match the order
                                ;  of the values of the CDX$xx entries as defined in TSGEN.
                                ;
000024  015340                  CTLNAM: .RAD50  /DL /           ;CDX$DL
000026  016420                          .RAD50  /DZ /           ;CDX$DZ
000030  015100                          .RAD50  /DH /           ;CDX$DZ
000032  015126                          .RAD50  /DHV/           ;CDX$VH
000034  062170                          .RAD50  /PC /           ;CDX$PI
000036  012500                          .RAD50  /CP /           ;CDX$PC
000040  063200                          .RAD50  /PP /           ;CDX$PP
000042  066300                          .RAD50  /QP /           ;CDX$QP
000044  114700                          .RAD50  /XX /
```

```
                                        .SBTTL  SHOW command
                                ;-----------------------------------------------------------------------
                                ;  Process the SHOW command
                                ;
000046  000240                  CMDSHO: NOP
000050  004767  000000G                CALL    CVTTAB          ;CONVERT TABS AND FF'S TO SPACES
000054  105713                        TSTB    @R3             ;IF NO OPTION WAS SPECIFED WITH COMMAND,
000056  001555                        BEQ     SOPASN          ;TREAT COMMAND LIKE "SHOW ASSIGNS"
000060  012704  000000'              MOV     #SHOHD,R4       ;POINT TO TABLE OF SHOW OPTIONS
000064  004767  000000G              CALL    SEARCH          ;LOOK UP OPTION
000070  103401                        BCS     BDSO            ;BR IF INVALID OPTION
000072  000134                        JMP     @(R4)+          ;JUMP TO PROCESSING ROUTINE
                                ;
                                ;  Invalid option
                                ;
000074  005704                  BDSO:   TST     R4              ;Invalid or ambiguous option?
000076  001404                        BEQ     1$              ;Br if invalid
000100                                  FABORT  #AMBOPT         ;Ambigious option
000110                          1$:     FABORT  #INVOPT         ;Invalid option
```

. ALL

```
                                        .SBTTL  .       ALL
                                ;-------------------------------------------------------------------------
                                ;  SHOW ALL
                                ;
000120  004767  010022          SOPALL: CALL    SHOVER          ;SHOW VERSION
000124                                  .PRINT  #CRLF
000132  004767  005052                  CALL    SHODEV          ;SHOW DEVICES
000136                                  .PRINT  #CRLF
000144  004767  006074                  CALL    SHOASN          ;SHOW ASSIGNS
000150                                  .PRINT  #CRLF
000156  004767  006430                  CALL    SHOALC          ;SHOW ALLOCATIONS
000162                                  .PRINT  #CRLF
000170  004767  003534                  CALL    SHOJOB          ;SHOW JOBS
000174  004767  001446                  CALL    SHOTRM          ;SHOW TERMINALS
000200                                  .PRINT  #CRLF
000206  004767  002630                  CALL    SHOCL           ;SHOW CL
000212                                  .PRINT  #CRLF
000220  004767  001056                  CALL    SHOMEM          ;SHOW MEMORY
000224                                  .PRINT  #CRLF
000232  004767  012012                  CALL    SHOSPL          ;SHOW SPOOL
000236                                  .PRINT  #CRLF
000244                                  .PRINT  #SUBTXT
000252  004767  012102                  CALL    SHOSUB          ;SHOW SUBSETS
000256                                  .PRINT  #CRLF
000264                                  .PRINT  #MNTTXT
000272  004767  006532                  CALL    SHOMNT          ;SHOW MOUNTS
000276                                  .PRINT  #CRLF
000304                                  .PRINT  #SRTTXT
000312  004767  011630                  CALL    SHOSRT          ;SHOW RUN-TIMES
000316                                  .PRINT  #CRLF
000324  004767  010560                  CALL    SHOREG          ;SHOW REGIONS
000330                                  .PRINT  #CRLF
000336  032767  0000000G 0000000G       BIT     #PO$SYS,PRIVCO          ;System privilege?
000344  001405                          BEQ     1$                      ;Branch if not
000346  004767  007656                  CALL    SHOINS          ;SHOW INSTALL
000352                                  .PRINT  #CRLF
000360  004767  011264          1$:     CALL    SHOPRV          ;SHOW PRIVILEGES
000364                                  .PRINT  #CRLF
000372  004767  011372                  CALL    SHOSLE          ;SHOW SL
000376                                  .PRINT  #CRLF
000404  004767  012220                  CALL    SHOWVM          ;SHOW VM (Base, Top and Size)
000410  000426                          BR      SOPJMP
                                ;
                                ;  SHOW ASSIGNS
                                ;
000412  004767  005626          SOPASN: CALL    SHOASN          ;SHOW ASSIGNS
000416  000423                          BR      SOPJMP
                                ;
                                ;  SHOW COMMANDS
                                ;
000420                          UCLCMD:
000420  004767  004526          SOPCMD: CALL    SHOCMD          ;SHOW COMMANDS
000424  000420                          BR      SOPJMP
                                ;
                                ;  SHOW CONFIGURATION
                                ;
000426  004767  004556          SOPCON: CALL    SHODEV          ;SHOW DEVICES
```

```
        ALL

     58 000432  004767  001210                  CALL    SHOTRM          ;SHOW TERMINALS
     59 000436  000413                          BR      SOPJMP
     60                                 ;
     61                                 ;  SHOW DATE
     62                                 ;
     63 000440  004767  007430          SOPDAT: CALL    SHODAT          ;SHOW DATE
     64 000444  000410                          BR      SOPJMP
     65                                 ;
     66                                 ;  SHOW DEVICES
     67                                 ;
     68 000446  004767  004536          SOPDEV: CALL    SHODEV          ;SHOW DEVICES
     69 000452  000405                          BR      SOPJMP
     70                                 ;
     71                                 ;  SHOW JOBS
     72                                 ;
     73 000454  004767  003250          SOPJOB: CALL    SHOJOB          ;SHOW JOBS
     74 000460  000402                          BR      SOPJMP
     75                                 ;
     76                                 ;  SHOW MEMORY
     77                                 ;
     78 000462  004767  000614          SOPMEM: CALL    SHOMEM          ;SHOW MEMORY
     79 000466  000167  000000G         SOPJMP: JMP     RDCMD
     80                                 ;
     81                                 ;  SHOW ALLOCATIONS
     82                                 ;
     83 000472  004767  006114          SOPALC: CALL    SHOALC          ;SHOW ALLOCATIONS
     84 000476  000167  000000G                 JMP     RDCMD
     85                                 ;
     86                                 ;  SHOW CL
     87                                 ;
     88 000502  004767  002334          SOPCL:  CALL    SHOCL           ;SHOW CL
     89 000506  000767                          BR      SOPJMP
     90                                 ;
     91                                 ;  SHOW INSTALL
     92                                 ;
     93 000510  004767  007514          SOPINS: CALL    SHOINS          ;SHOW INSTALL
     94 000514  000764                          BR      SOPJMP
     95                                 ;
     96                                 ;  SHOW MOUNTS
     97                                 ;
     98 000516  004767  006306          SOPMNT: CALL    SHOMNT          ;SHOW MOUNTS
     99 000522  000761                          BR      SOPJMP
    100                                 ;
    101                                 ;  SHOW MODEM (ONTIM, OFFTIM, TIMIN, TIMOUT, TIMLOC)
    102                                 ;
    103 000524  004767  007172          SOPMDM: CALL    SHOMDM          ;SHOW MODEM
    104 000530  000756                          BR      SOPJMP
    105                                 ;
    106                                 ;  SHOW PRIVILEGES
    107                                 ;
    108 000532  004767  011112          SOPPRV: CALL    SHOPRV          ;SHOW PRIVILEGES
    109 000536  000753                          BR      SOPJMP
    110                                 ;
    111                                 ;  SHOW QUEUE
    112                                 ;
    113 000540  004767  004310          SOPQUE: CALL    SHOQUE          ;SHOW QUEUE
    114 000544  000750                          BR      SOPJMP
```

```
.       ALL

                                        ;
                                        ;   SHOW RUN-TIMES
                                        ;
000546  004767  011374          SOPSRT: CALL    SHOSRT          ;SHOW RUN-TIMES
000552  000745                          BR      SOPJMP
                                        ;
                                        ;   SHOW SL
                                        ;
000554  004767  011210          SOPSLE: CALL    SHOSLE          ;SHOW SL
000560  000742                          BR      SOPJMP
                                        ;
                                        ;   SHOW VM
                                        ;
000562  004767  012042          SOPVM:  CALL    SHOWVM          ;SHOW VM
000566  000737                          BR      SOPJMP
                                        ;
                                        ;   SHOW REGIONS
                                        ;
000570  004767  010314          SOPREG: CALL    SHOREG          ;SHOW REGIONS
000574  000734                          BR      SOPJMP
                                        ;
                                        ;   SHOW SUBSETS
                                        ;
000576  004767  011556          SOPSUB: CALL    SHOSUB          ;SHOW SUBSETS
000602  000731                          BR      SOPJMP
                                        ;
                                        ;   SHOW SPOOL
                                        ;
000604  004767  011440          SOPSPL: CALL    SHOSPL          ;SHOW SPOOL
000610  000726                          BR      SOPJMP
                                        ;
                                        ;   SHOW SYSPASSWORD
                                        ;
000612  004767  000432          SOPSYP: CALL    SHOSYP          ;SHOW SYSPASSWORD
000616  000723                          BR      SOPJMP
                                        ;
                                        ;   SHOW TERMINALS
                                        ;
000620  004767  001022          SOPTRM: CALL    SHOTRM          ;SHOW TERMINALS
000624  000720                          BR      SOPJMP
                                        ;
                                        ;   SHOW TIME
                                        ;
000626  004767  007300          SOPTIM: CALL    SHOTIM          ;SHOW TIME
000632  000715                          BR      SOPJMP
                                        ;
                                        ;   SHOW USE
                                        ;
000634  004767  007362          SOPUSE: CALL    SHOUSE          ;SHOW USE
000640  000712                          BR      SOPJMP
                                        ;
                                        ;   SHOW VERSION
                                        ;
000642  004767  007300          SOPVER: CALL    SHOVER          ;SHOW VERSION
000646  000707                          BR      SOPJMP
```

```
.       ALL

                                        ;
                                        ;   SHOW MAXMSG
                                        ;
000650  016705  000000G        SOPMB:  MOV     VMXMSG,R5
000654  000427                        BR      SHOVAL
                                        ;
                                        ;   SHOW MAXMC
                                        ;
000656  016705  000000G        SOPMC:  MOV     VMAXMC,R5
000662  000424                        BR      SHOVAL
                                        ;
                                        ;   SHOW MAXMRB
                                        ;
000664  016705  000000G        SOPMR:  MOV     VMXMRB,R5
000670  000421                        BR      SHOVAL
                                        ;
                                        ;   SHOW QUANO
                                        ;
000672  016705  000000G        SOPQO:  MOV     VQUANO,R5       ;Get value
000676  000416                        BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN1
                                        ;
000700  016705  000000G        SOPQ1:  MOV     VQUAN1,R5       ;GET VALUE
000704  000413                        BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN1A
                                        ;
000706  016705  000000G        SOPQ1A: MOV     VQUN1A,R5
000712  000410                        BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN1B
                                        ;
000714  016705  000000G        SOPQ1B: MOV     VQUN1B,R5
000720  000405                        BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN1C
                                        ;
000722  016705  000000G        SOPQ1C: MOV     VQUN1C,R5
000726  000402                        BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN2
                                        ;
000730  016705  000000G        SOPQ2:  MOV     VQUAN2,R5
000734                         SHOVAL: .PRINT  #SPACE3         ;Print 3 spaces
000742  004767  000000G                CALL    PRTDEC
000746                                 .PRINT  #CRLF
000754  000167  000000G                JMP     RDCMD
                                        ;
                                        ;   SHOW QUAN3
                                        ;
000760  016705  000000G        SOPQ3:  MOV     VQUAN3,R5       ;Get QUAN3 value
000764  000763                        BR      SHOVAL          ;Display it
                                        ;
                                        ;   SHOW CORTIM
                                        ;
000766  016705  000000G        SOPCTM: MOV     VCORTM,R5
```

```
000772  000760                                  BR      SHOVAL
                                        ;
                                        ;   SHOW INTIOC
                                        ;
000774  016705  000000G         SOPIIO: MOV     VINTIO,R5
001000  000755                          BR      SHOVAL
                                        ;
                                        ;   SHOW HIPRCT
                                        ;
001002  016705  000000G         SOPHIO: MOV     VHIPCT,R5
001006  000752                          BR      SHOVAL
                                        ;
                                        ;   SHOW NUMDC
                                        ;
001010  016705  000000G         SOPNDC: MOV     NUMDCD,R5
001014  000747                          BR      SHOVAL
                                        ;
                                        ;   SHOW PRILOW
                                        ;
001016  116705  000000G         SOPPLO: MOVB    VPRILO,R5
001022  000744                          BR      SHOVAL
                                        ;
                                        ;   SHOW PRIHI
                                        ;
001024  116705  000000G         SOPPHI: MOVB    VPRIHI,R5
001030  000741                          BR      SHOVAL
                                        ;
                                        ;   SHOW PRIDEF
                                        ;
001032  116705  000000G         SOPPDF: MOVB    VPRIDF,R5
001036  000736                          BR      SHOVAL
                                        ;
                                        ;   SHOW PRIVIR
                                        ;
001040  116705  000000G         SOPPVR: MOVB    VPRIVR,R5
001044  000733                          BR      SHOVAL
                                        ;
                                        ;   SHOW PRIORITY
                                        ;
001046                          SOPPRI: .PRINT  #TM$PR1         ;"Current priority = "
001054  116105  000000G                 MOVB    LPRI(R1),R5     ;Get current priority
001060  004767  000000G                 CALL    PRTDEC          ;Print it
001064                                  .PRINT  #TM$PR2         ;"; maximum authorized priority = "
001072  116705  000000G                 MOVB    MXJPRI,R5       ;Get max authorized priority
001076  004767  000000G                 CALL    PRTDEC          ;Print it
001102                                  .PRINT  #CRLF           ;End of line
001110                                  .PRINT  #TM$LPR         ;"Low priority range = 0 to "
001116  116705  000000G                 MOVB    VPRILO,R5       ;Get top of low priority range
001122  004767  000000G                 CALL    PRTDEC          ;Print the value
001126                                  .PRINT  #CRLF
001134                                  .PRINT  #TM$HPR         ;"High priority range = "
001142  116705  000000G                 MOVB    VPRIHI,R5       ;Get base of high priority range
001146  004767  000000G                 CALL    PRTDEC          ;Print it
001152                                  .PRINT  #TM$HPE         ;Finish message
001160  000167  000000G                 JMP     RDCMD
                                        ;
                                        ;   SHOW CACHE
```

```
                                   ;
001164  016705  000000G            SOPCSH: MOV     CSHALC,R5       ;Was caching genned into system?
001170  001004                             BNE     1$              ;Br if yes
001172                                     .PRINT  #TM$CNG         ;Caching not genned into system
001200  000421                             BR      9$
001202  005767  000000G            1$:     TST     VCSHNB          ;Is caching enabled?
001206  001004                             BNE     2$              ;Br if yes
001210                                     .PRINT  #TM$CDS         ;Caching is disabled
001216  000412                             BR      9$
001220                             2$:     .PRINT  #TM$CEN         ;Data cache size is ...
001226  016705  000000G                    MOV     VCSHNB,R5       ;Get current number of blocks for cache
001232  004767  000000G                    CALL    PRTDEC          ;Print size of cache
001236                                     .PRINT  #CRLF           ;Terminate print line
001244  000167  000000G            9$:     JMP     RDCMD

                                   ;
                                   ;  SHOW SYSPASSWORD
                                   ;
001250  004767  000000G            SHOSYP: CALL    CKSYPV          ;Make sure user has SYSPRV privilege
001254  105767  000000G                    TSTB    SYPSWD          ;Is there a system password?
001260  001004                             BNE     1$              ;Br if yes
001262                                     .PRINT  #TM$NSP         ;No system password
001270  000403                             BR      9$
001272                             1$:     .PRINT  #SYPSWD         ;Print the password
001300  000207                     9$:     RETURN
```

```
                                        .SBTTL  .       MEMORY
                                ;-----------------------------------------------------------------------
                                ;  SHOW MEMORY
                                ;
001302  010546                  SHOMEM: MOV     R5,-(SP)
                                ;  Total installed memory
001304                                  .PRINT  #TOTMMS         ;TOTAL MEMORY
001312  016705  000000G                 MOV     PHYMEM,R5       ;Get total # 64-byte blocks of memory
001316  072527  177775                  ASH     #-3,R5          ;Convert to # 512-byte pages
001322  042705  160000                  BIC     #160000,R5      ;Kill possible sign extension
001326  004767  012230                  CALL    PRTKB           ;DISPLAY THE VALUE
                                ;  Size of unmapped system & unmapped handlers
001332                                  .PRINT  #UMSSMS         ;SIZE OF UNMAPPED PORTION OF SYSTEM
001340  016705  000000G                 MOV     UMSYTP,R5       ;GET ADDRESS OF TOP OF TSX
001344  062705  001777                  ADD     #1777,R5        ;ROUND UP TO KB
001350  000241                          CLC                     ;CVT TO KB
001352  006005                          ROR     R5              ;/2 without propagating sign
001354  072527  177767                  ASH     #-9.,R5         ;/2/512.=/1024.
001360  004767  000000G                 CALL    PRTDEC          ;Display #Kb
001364                                  .PRINT  #KBPARN         ;"Kb  ("
001372  016702  000000G                 MOV     UMSYTP,R2       ;Get limit again
001376  004767  000000G                 CALL    OCTPRT          ;Display high address
001402                                  .PRINT  #PARNNL         ;")<CR><LF>"
                                ;  Size of mapped system regions
001410                                  .PRINT  #SSRMAP         ;SIZE OF SYSTEM MAPPED REGION
001416  016705  000000G                 MOV     SMRSIZ,R5       ;# 64-byte blocks for mapped regions
001422  062705  000007                  ADD     #7,R5           ;Round up
001426  072527  177775                  ASH     #-3,R5          ;Convert to 512-byte block units
001432  004767  012124                  CALL    PRTKB
001436  010546                          MOV     R5,-(SP)        ;SAVE MAPPED REGION SIZE
                                ;  Total TSX-Plus size
001440                                  .PRINT  #TSXSMS         ;TSX SIZE
001446  016705  000000G                 MOV     LOMAP,R5        ;PAGE AT TOP OF TSX
001452  166705  000000G                 SUB     BASMAP,R5       ;# PAGES FOR TSX AND HANDLERS
001456  005305                          DEC     R5
001460  062605                          ADD     (SP)+,R5        ;ADD MAPPED REGION SIZE
001462  004767  012074                  CALL    PRTKB
                                ;  Mapped handlers
001466                                  .PRINT  #MHNSMS         ;Size of mapped handlers
001474  016705  000000G                 MOV     MHNSIZ,R5       ;# 64-byte blocks for mapped handlers
001500  062705  000007                  ADD     #7,R5           ;Round up
001504  072527  177775                  ASH     #-3,R5          ;Convert to 512-byte block units
001510  004767  012046                  CALL    PRTKB
                                ;  Shared run-time systems
001514                                  .PRINT  #SRTSMS         ;SIZE OF SHARABLE RUN-TIME SYSTEMS
001522  016705  000000G                 MOV     SRTSIZ,R5       ;# 64-byte blocks for shared run-times
001526  062705  000007                  ADD     #7,R5           ;Round up
001532  072527  177775                  ASH     #-3,R5          ;Convert to 512-byte block units
001536  004767  012020                  CALL    PRTKB
                                ;  Data cache buffers
001542                                  .PRINT  #CSHMSG         ;Size of data cache
001550  016705  000000G                 MOV     CSHSIZ,R5       ;# 64-byte blocks for data cache
001554  062705  000007                  ADD     #7,R5           ;Round up
001560  000241                          CLC                     ;Clear carry to convert to 512 byte
001562  006005                          ROR     R5              ; block units without sign extension
001564  072527  177776                  ASH     #-2,R5          ; in case cache >=2Mb (100000 64bytes)
001570  004767  011766                  CALL    PRTKB
```

```
                                        ;       User memory space
   001574                                       .PRINT  #USRMMS         ;USER MEMORY SPACE
   001602  016705  000000G                      MOV     HIMAP,R5
   001606  166705  000000G                      SUB     LOMAP,R5
   001612  004767  011744                       CALL    PRTKB
                                        ;       Job context area size
   001616                                       .PRINT  #JCXSMS         ;SIZE OF JOB CONTEXT AREA
   001624  016705  000000G                      MOV     JCXPGS,R5
   001630  005205                               INC     R5              ;ROUND UP
   001632  004767  011724                       CALL    PRTKB
   001636  004767  011620                       CALL    DSPMEM          ;SHOW JOB MEMORY LIMITS
   001642  012605                               MOV     (SP)+,R5
   001644  000207                               RETURN
```

```
                                        .SBTTL  .       TERMINALS
                                ;-----------------------------------------------------------------------
                                ;  Display information about terminals.
                                ;
001646  010146                  SHOTRM: MOV     R1,-(SP)
001650  010246                          MOV     R2,-(SP)
001652  010346                          MOV     R3,-(SP)
001654  010446                          MOV     R4,-(SP)
001656  010546                          MOV     R5,-(SP)
                                ;
                                ;  Print heading lines
                                ;
001660                                  .PRINT  #CRLF           ;Print a blank line
001666                                  .PRINT  #TRMHD1         ;Heading line 1
001674                                  .PRINT  #TRMHD2         ;Heading line 2
                                ;
                                ;  Begin loop to print info for each terminal
                                ;
001702  012701  000002                  MOV     #2,R1           ;Get first terminal index number
                                ;
                                ;  If we are on a pro, do not display info about uninstalled lines
                                ;
001706  032761  000000G 000000G 1$:     BIT     #$DEAD,LSW3(R1) ;Is this line installed?
001714  001406                          BEQ     33$             ;Br if yes
001716  032767  000000G 000000G         BIT     #CW$PRO,CONFG2  ;Are we running on a pro?
001724  001402                          BEQ     33$             ;Br if not
001726  000167  001044                  JMP     30$             ;Go check next line
                                ;
                                ;  Display unit number
                                ;
001732  010105                  33$:    MOV     R1,R5           ;Get terminal index #
001734  006205                          ASR     R5              ;Convert to unit #
001736  012703  000002                  MOV     #2,R3           ;Print using 2 columns
001742  004767  000000G                 CALL    PRTFIX          ;Print unit number
                                ;
                                ;  Print "*" if this is the current line
                                ;
001746  116700  000000G                 MOVB    CORUSR,R0       ;Get current job index #
001752  116000  000000G                 MOVB    LNPRIM(R0),R0   ;Get primary line #
001756  012703  000004                  MOV     #4,R3           ;Set to print 4 spaces
001762  020100                          CMP     R1,R0           ;Is this our line?
001764  001005                          BNE     15$             ;Br if not
001766                                  .TTYOUT #'*             ;Flag our line
001776  005303                          DEC     R3              ;Print only 3 more spaces
002000  004767  000000G         15$:    CALL    PRTSPC          ;Print spaces
                                ;
                                ;  Display type of terminal
                                ;
002004  012703  000000G                 MOV     #OPRTXT,R3      ;Assume this is operator's console
002010  120167  000000G                 CMPB    R1,CTRLTT       ;Is this operator's console?
002014  001430                          BEQ     2$              ;Br if yes
002016  012703  000000G                 MOV     #CLLINE,R3      ;Assume this is a CL line
002022  020127  000000G                 CMP     R1,#LSTPL       ;Is this a CL line?
002026  101023                          BHI     2$              ;Br if yes
002030  012703  000000G                 MOV     #LCLTXT,R3      ;Assume this is a local terminal
002034  032761  000000G 000000G         BIT     #$PHONE,ILSW2(R1);Is this a dial-up terminal?
002042  001415                          BEQ     2$              ;Br if not dial-up
```

```
     58 002044  032761  000000G 000000G          BIT     #$INIT,LSW(R1)  ;Is line in use now?
     59 002052  001407                          BEQ     17$             ;Print "PHONE" if inactive dial-up line
     60 002054  105767  000000G                 TSTB    VUSPHN          ;Are phone lines always phone?
     61 002060  001004                          BNE     17$             ;If so, print "PHONE"
     62 002062  032761  000000G 000000G          BIT     #$CARMN,LSW5(R1) ;Is line being used as dial-up?
     63 002070  001402                          BEQ     2$              ;Print "LOCAL" if active but no carrier
     64 002072  012703  000000G          17$:    MOV     #REMTXT,R3      ;This is a remote terminal
     65 002076                           2$:     .PRINT  R3              ;Print type
     66                                  ;
     67                                  ;  Display vector address and DL/DZ type
     68                                  ;
     69 002102  016100  000000G                  MOV     LCDTYP(R1),R0   ;Get device type code for this line
     70 002106  016000  000024'                  MOV     CTLNAM(R0),R0   ;Get RAD50 name of controlling device
     71 002112  004767  000000G                  CALL    PRTR50          ;Print the device name
     72 002116  016105  000000G                  MOV     LMXLN(R1),R5    ;Get line # within mux
     73 002122  016104  000000G                  MOV     LMXNUM(R1),R4   ;Get mux index number
     74 002126  001007                           BNE     3$              ;Br if this is a mux line
     75                                  ;  This is not a mux line
     76 002130  012703  000004                   MOV     #4,R3           ;Print 4 spaces
     77 002134  004767  000000G                  CALL    PRTSPC
     78 002140  016105  000000G                  MOV     INVEC(R1),R5    ;Get input vector address
     79 002144  000420                           BR      4$
     80                                  ;  This is a mux line
     81 002146                           3$:     .TTYOUT #'-             ;Put in hyphen
     82 002156  016105  000000G                  MOV     LMXLN(R1),R5    ;Get line # within the mux
     83 002162  012703  000002                   MOV     #2,R3           ;Print in 2 column field
     84 002166  004767  000000G                  CALL    PRTFIX          ;Print line number
     85 002172  012703  000001                   MOV     #1,R3           ;Print 1 space
     86 002176  004767  000000G                  CALL    PRTSPC
     87 002202  016405  000000G                  MOV     MXVEC(R4),R5    ;Get DZ-11 interrupt vector address
     88 002206  012703  000003           4$:     MOV     #3,R3           ;Print 3 digits
     89 002212  004767  000000G                  CALL    OCTFIX          ;Print vector address
     90 002216  012703  000002                   MOV     #2,R3           ;Print 2 spaces
     91 002222  004767  000000G                  CALL    PRTSPC
     92                                  ;
     93                                  ;  Print address of Control Status Register (CSR)
     94                                  ;
     95 002226  016104  000000G                  MOV     LMXNUM(R1),R4   ;Get mux index number
     96 002232  001003                           BNE     10$             ;Br if DZ-11
     97                                  ;  This is a DL-11 line
     98 002234  016102  000000G                  MOV     RSR(R1),R2      ;Get address of receiver status register
     99 002240  000402                           BR      11$
    100                                  ;  This is a DZ-11 line
    101 002242  016402  000000G          10$:    MOV     MXCSR(R4),R2    ;Get address of CSR
    102 002246  004767  000000G          11$:    CALL    OCTPRT          ;Print the address
    103 002252  012703  000002                   MOV     #2,R3           ;Print 2 spaces
    104 002256  004767  000000G                  CALL    PRTSPC
    105                                 ;
    106                                 ;  Print information about terminal type
    107                                 ;
    108 002262  004767  000000G                  CALL    PRTTTP          ;Print terminal type
    109 002266  012703  000002                   MOV     #2,R3           ;Print 2 spaces
    110 002272  004767  000000G                  CALL    PRTSPC
    111                                 ;
    112                                 ;  Print information about speed
    113                                 ;
    114 002276  032761  000000G 000000G          BIT     #$AUTO,ILSW2(R1);Is autobaud specified for this line?
```

```
     115 002304  001413                          BEQ     24$             ;Br if not
     116 002306  005761  000000G                 TST     LCLUNT(R1)      ;Is this line being used as a CL unit?
     117 002312  002010                          BGE     24$             ;Br if yes -- No autobaud for CL
     118 002314  032761  000000G 000000G         BIT     #$DILUP,LSW(R1) ;Is a job active on this line?
     119 002322  001004                          BNE     24$             ;Br if yes -- speed is known if job on
     120 002324                                  .PRINT  #TM$AUT         ;Print "auto"
     121 002332  000424                          BR      7$
     122 002334  016103  000000G         24$:    MOV     LMXPRM(R1),R3   ;Get line parameters for mux line
     123 002340  000303                          SWAB    R3              ;Right justify speed code
     124 002342  042703  177760                  BIC     #177760,R3      ;Clear all but speed code
     125 002346  001004                          BNE     5$              ;Br if we know the speed
     126 002350                                  .PRINT  #NATXT          ;Print "N/A"
     127 002356  000412                          BR      7$
     128 002360  070327  000005          5$:     MUL     #5,R3           ;Values are stored 5 characters each
     129 002364  062703  000000G                 ADD     #SPDTX1,R3      ;Point to text string
     130 002370  012704  000005                  MOV     #5,R4           ;Print 5 characters
     131 002374  112300                  6$:     MOVB    (R3)+,R0        ;Get next char of speed
     132 002376                                  .TTYOUT                 ;Print it
     133 002402  077404                          SOB     R4,6$           ;Loop till all printed
     134 002404                          7$:     .TTYOUT #40             ;Print a space
     135                                         ;
     136                                         ;  Print information about character length and parity
     137                                         ;
     138 002414  116103  000001G                 MOVB    LMXPRM+1(R1),R3 ;Get line parameter flags
     139 002420  032703  000000G                 BIT     #LP$7BT,R3      ;7 bit characters wanted?
     140 002424  001005                          BNE     27$             ;Br if yes
     141 002426                                  .TTYOUT #'8             ;Say length is 8 bits
     142 002436  000404                          BR      28$
     143 002440                          27$:    .TTYOUT #'7             ;Say length is 7 bits
     144 002450  032703  000000G         28$:    BIT     #LP$PAR,R3      ;Parity wanted?
     145 002454  001005                          BNE     29$             ;Br if yes
     146 002456                                  .TTYOUT #'N             ;Say no parity
     147 002466  000414                          BR      31$
     148 002470  032703  000000G         29$:    BIT     #LP$ODD,R3      ;Odd parity wanted?
     149 002474  001005                          BNE     32$             ;Br if yes
     150 002476                                  .TTYOUT #'E             ;Say even parity
     151 002506  000404                          BR      31$
     152 002510                          32$:    .TTYOUT #'O             ;Say odd parity
     153 002520  012703  000004          31$:    MOV     #4,R3           ;Print 4 spaces
     154 002524  004767  000000G                 CALL    PRTSPC
     155                                         ;
     156                                         ;  Indicate if line is active now
     157                                         ;
     158 002530  012703  000000G                 MOV     #NOTXT,R3       ;Assume line is not active
     159 002534  032761  000000G 000000G         BIT     #$DEAD,LSW3(R1) ;Is line installed?
     160 002542  001035                          BNE     12$             ;Br if not installed
     161 002544  016105  000000G                 MOV     LCLUNT(R1),R5   ;Is a CL unit assigned to this line?
     162 002550  002012                          BGE     16$             ;Br if yes
     163 002552  020127  000000G                 CMP     R1,#LSTPL       ;Is this a real or CL line?
     164 002556  101027                          BHI     12$             ;Br if CL line
     165 002560  032761  000000G 000000G         BIT     #$INIT,LSW(R1)  ;Is line active now?
     166 002566  001423                          BEQ     12$             ;Br if not
     167 002570  012703  000000G                 MOV     #YESTXT,R3      ;Point to "Yes" text
     168 002574  000420                          BR      12$             ;Print yes
     169 002576  006205                  16$:    ASR     R5              ;Convert CL unit index to unit number
     170 002600  020527  000007                  CMP     R5,#7.          ;Units 0-7 go with CL device
     171 002604  101406                          BLOS    34$
```

```
    172 002606  162705  000010                      SUB     #8.,R5          ;Remove C1 unit bias
    173 002612                                      .PRINT  #TXTC1          ;Print "C1"
    174 002620  000403                              BR      35$
    175 002622                              34$:    .PRINT  #TXTCL          ;Print "CL"
    176 002630  004767  000000G             35$:    CALL    PRTDEC          ;Print CL unit number
    177 002634  000402                              BR      25$
    178 002636                              12$:    .PRINT  R3              ;Print Yes/No
    179 002642  012703  000003              25$:    MOV     #3,R3           ;Print 3 spaces
    180 002646  004767  000000G                     CALL    PRTSPC
    181                                     ;
    182                                     ;  Print name of user who is using this line
    183                                     ;
    184 002652  032761  000000G 000000G             BIT     #$DEAD,LSW3(R1) ;Is this line installed?
    185 002660  001404                              BEQ     26$             ;Br if line is installed
    186 002662                                      .PRINT  #NINTXT         ;Print Not installed
    187 002670  000437                              BR      14$
    188 002672  016105  000000G             26$:    MOV     LNAME(R1),R5    ;Is there a descriptive name for this line?
    189 002676  001405                              BEQ     13$             ;Br if not
    190 002700  105715                              TSTB    (R5)            ;Is the line blank?
    191 002702  001403                              BEQ     13$             ;Branch if no line name
    192 002704                                      .PRINT  R5              ;Print asciz name string
    193 002710  000432                              BR      30$
    194 002712  016105  000000G             13$:    MOV     LCLUNT(R1),R5   ;Is this line used as a CL unit?
    195 002716  002016                              BGE     21$             ;Br if yes
    196 002720  020127  000000G                     CMP     R1,#LSTPL       ;Is this a real or CL line?
    197 002724  101007                              BHI     23$             ;Br if CL line
    198 002726  032761  000000G 000000G             BIT     #$KINIT,LSW(R1) ;Is job logged on?
    199 002734  001415                              BEQ     14$             ;Br if not
    200 002736  004767  000000G                     CALL    PRTUNM          ;Print user's name
    201 002742  000412                              BR      14$
    202 002744                              23$:    .PRINT  #CLFREE         ;This is a free CL line
    203 002752  000406                              BR      14$
    204 002754                              21$:    .PRINT  #CLUNIT         ;Print "CL line "
    205 002762  006205                              ASR     R5              ;Convert to unit number
    206 002764  004767  000000G                     CALL    PRTDEC          ;Print decimal value
    207 002770                              14$:
    208                                     ;
    209                                     ;  Terminate print line and loop if more lines to print
    210                                     ;
    211 002770                                      .PRINT  #CRLF           ;Terminate print line
    212 002776  062701  000002              30$:    ADD     #2,R1           ;Advance line index number
    213 003002  020127  000000G                     CMP     R1,#LSTHL       ;Are there more lines?
    214 003006  101007                              BHI     20$             ;Br if not
    215 003010  020127  000002G                     CMP     R1,#LSTPL+2     ;Have we stepped past last real line?
    216 003014  001002                              BNE     22$             ;Br if not
    217 003016  012701  000000G                     MOV     #FSTIOL,R1      ;Skip up to 1st CL line
    218 003022  000167  176660              22$:    JMP     1$              ;Go display info for next line
    219                                     ;
    220                                     ;  Finished
    221                                     ;
    222 003026  012605                      20$:    MOV     (SP)+,R5
    223 003030  012604                              MOV     (SP)+,R4
    224 003032  012603                              MOV     (SP)+,R3
    225 003034  012602                              MOV     (SP)+,R2
    226 003036  012601                              MOV     (SP)+,R1
    227 003040  000207                              RETURN
```

```
                                        .SBTTL  .       CL
                                ;-------------------------------------------------------------------------------
                                ;  Display information about CL units.
                                ;
003042  010146                  SHOCL:  MOV     R1,-(SP)
003044  010346                          MOV     R3,-(SP)
003046  010446                          MOV     R4,-(SP)
003050  010546                          MOV     R5,-(SP)
                                ;
                                ;  See if there are any CL units
                                ;
003052  005727  000000G                 TST     #CLTOTL         ;Are there any CL units?
003056  001005                          BNE     11$             ;Br if yes
003060                                  .PRINT  #TM$CLO         ;There are no CL units
003066  000167  000466                  JMP     9$
                                ;
                                ;  There are some CL units.
                                ;  Print version number.
                                ;
003072                          11$:    .PRINT  #TM$CL7         ;"CL emulating XL version "
003100  116705  000000G                 MOVB    CLVERS,R5       ;Set number to be printed
003104  004767  000000G                 CALL    PRTDEC          ;Print the decimal value
003110                                  .PRINT  #TM$CL8         ;" "
                                ;
                                ;  Print heading lines.
                                ;
003116                          1$:     .PRINT  #TM$CL1         ;Heading line 1
003124                                  .PRINT  #TM$CL2         ;Heading line 2
                                ;
                                ;  Begin loop to print info about each unit.
                                ;
003132  005001                          CLR     R1              ;Init CL unit index number
                                ;
                                ;  Print CL unit name
                                ;
003134  010105                  2$:     MOV     R1,R5           ;Get CL unit index
003136  006205                          ASR     R5              ;Convert to unit number
003140  020527  000007                  CMP     R5,#7.          ;Unit # in range 0-7 get name CL
003144  101406                          BLOS    19$
003146                                  .PRINT  #TM$C13         ;Print " C1"
003154  162705  000010                  SUB     #8.,R5          ;Remove C1 unit number bias
003160  000403                          BR      20$
003162                          19$:    .PRINT  #TM$CL3         ;Print " CL"
003170  004767  000000G         20$:    CALL    PRTDEC          ;Print unit #
003174  012703  000002                  MOV     #2,R3           ;Print 2 spaces
003200  004767  000000G                 CALL    PRTSPC
                                ;
                                ;  Print the number of the associated line
                                ;
003204  016105  000000G                 MOV     CL$LIX(R1),R5   ;Get associated line number
003210  001412                          BEQ     3$              ;Br if there is no associated line
003212  006205                          ASR     R5              ;Get line #
003214  012703  000003                  MOV     #3.,R3          ;Get print field width
003220  004767  000000G                 CALL    PRTFIX          ;Print the line number
003224  012703  000003                  MOV     #3.,R3          ;Print 3 spaces
003230  004767  000000G                 CALL    PRTSPC
003234  000407                          BR      4$
```

```
     58 003236                          3$:     .PRINT  #TM$CL4         ;Print "none"
     59 003244  012703  000002                  MOV     #2,R3           ;Print 2 spaces
     60 003250  004767  0000000G                CALL    PRTSPC
     61                                 ;
     62                                 ;  Print number of any job that is using this CL unit
     63                                 ;
     64 003254  010100                  4$:     MOV     R1,R0           ;Get CL index number
     65 003256  004767  0000000G                CALL    CKCLUS          ;See if any job is using CL unit
     66 003262  010005                          MOV     R0,R5           ;Get # of job that is using the device
     67 003264  001411                          BEQ     18$             ;Br if device is free
     68 003266  012703  000003                  MOV     #3,R3           ;Print in 3 character field
     69 003272  004767  0000000G                CALL    PRTFIX          ;Print the job number
     70 003276  012703  000002                  MOV     #2,R3           ;Print 2 spaces
     71 003302  004767  0000000G                CALL    PRTSPC
     72 003306  000407                          BR      17$
     73 003310                          18$:    .PRINT  #TM$CL4         ;Print "none"
     74 003316  012703  000001                  MOV     #1,R3           ;Print 1 space
     75 003322  004767  0000000G                CALL    PRTSPC
     76                                 ;
     77                                 ;  See if this CL unit is spooled
     78                                 ;
     79 003326  016705  0000000G        17$:    MOV     CLDEVX,R5       ;Get CL device index number
     80 003332  010100                          MOV     R1,R0           ;Get CL unit # index
     81 003334  006200                          ASR     R0              ;Convert to unit #
     82 003336  020027  000007                  CMP     R0,#7           ;CL or C1 type unit?
     83 003342  101404                          BLOS    21$             ;Br if CL
     84 003344  162700  000010                  SUB     #8.,R0          ;Remove C1 unit bias
     85 003350  016705  0000000G                MOV     C1DEVX,R5       ;Get C1 device index number
     86 003354  000300                  21$:    SWAB    R0              ;Put unit number in high order byte
     87 003356  050005                          BIS     R0,R5           ;Combine with device index #
     88 003360  012700  0000000G                MOV     #SDCB,R0        ;Point to first SDCB
     89 003364  020027  0000000G        5$:     CMP     R0,#SDCBND      ;Have we checked all spooled devices?
     90 003370  103011                          BHIS    7$              ;Br if yes -- This CL unit is not spooled
     91 003372  020560  0000000G                CMP     R5,SDDVU(R0)    ;Is this SDCB for this CL unit?
     92 003376  001403                          BEQ     6$              ;Br if yes -- CL unit is spooled
     93 003400  062700  0000000G                ADD     #SDCBSZ,R0      ;Point to next SDCB
     94 003404  000767                          BR      5$
     95 003406                          6$:     .PRINT  #TM$CL5         ;Print "(spooled)"
     96                                 ;
     97                                 ;  Print which options are selected for this line
     98                                 ;
     99 003414  005761  0000000G        7$:     TST     CL$LIX(R1)      ;Is this CL unit assigned to a line?
    100 003420  001445                          BEQ     10$             ;Br if not -- Don't display options
    101 003422  016105  0000000G                MOV     CL$OPT(R1),R5   ;Get option flags for this CL unit
    102 003426  012703  003572'                 MOV     #CLOPTB,R3      ;Point to option name table
    103 003432  005004                          CLR     R4              ;Say no option names printed yet
    104 003434  032305                 15$:    BIT     (R3)+,R5        ;Is this option selected?
    105 003436  001420                          BEQ     16$             ;Br if not
    106 003440  005704                          TST     R4              ;Is this the 1st option?
    107 003442  001004                          BNE     12$             ;Br if not
    108 003444                                  .PRINT  #TM$CL6         ;Print " ["
    109 003452  000404                          BR      13$
    110 003454                         12$:    .TTYOUT #54             ;Print comma
    111 003464  010304                 13$:    MOV     R3,R4           ;Point to option name string
    112 003466  112400                 14$:    MOVB    (R4)+,R0        ;Get next char from option name string
    113 003470  001403                          BEQ     16$             ;Br if hit end of name string
    114 003472                                  .TTYOUT                 ;Print next char of option name
```

```
     115 003476  000773                          BR      14$             ;Loop to print rest of option name
     116 003500  105723                  16$:    TSTB    (R3)+           ;Search for end of option name
     117 003502  001376                          BNE     16$             ;Loop till end found
     118 003504  005203                          INC     R3              ;Point to next word
     119 003506  042703  000001                  BIC     #1,R3
     120 003512  020327  003730'                 CMP     R3,#CLOPND      ;Checked all options?
     121 003516  103746                          BLO     15$             ;Loop if not
     122 003520  005704                          TST     R4              ;Were any options printed?
     123 003522  001404                          BEQ     10$             ;Br if not
     124 003524                                  .TTYOUT #']             ;Close option list
     125                                 ;
     126                                 ;  Terminate the print line
     127                                 ;
     128 003534                          10$:    .PRINT  #CRLF           ;End of print line
     129                                 ;
     130                                 ;  See if there are more CL units
     131                                 ;
     132 003542  062701  000002                  ADD     #2,R1           ;Advance CL unit index
     133 003546  020127  000000C                 CMP     R1,#2*CLTOTL    ;Have we done all CL units?
     134 003552  103002                          BHIS    9$              ;Br if yes
     135 003554  000167  177354                  JMP     2$              ;Go show info for next unit
     136                                 ;
     137                                 ;  Finished
     138                                 ;
     139 003560  012605                  9$:     MOV     (SP)+,R5
     140 003562  012604                          MOV     (SP)+,R4
     141 003564  012603                          MOV     (SP)+,R3
     142 003566  012601                          MOV     (SP)+,R1
     143 003570  000207                          RETURN
     144                                 ;
     145                                 ;  Table of CL option flags and option names
     146                                 ;
     147                                         .MACRO  CLOP    FLAG,NAME
     148                                         .WORD   FLAG
     149                                         .ASCIZ  /'NAME'/
     150                                         .EVEN
     151                                         .ENDM   CLOP
     152                                 ;
     153 003572                          CLOPTB:
     154 003572                                  CLOP    CO$FF,FORM
     155 003602                                  CLOP    CO$TAB,TAB
     156 003610                                  CLOP    CO$LC,LC
     157 003616                                  CLOP    CO$8BT,EIGHTBIT
     158 003632                                  CLOP    CO$LFO,LFOUT
     159 003642                                  CLOP    CO$LFI,LFIN
     160 003652                                  CLOP    CO$FFO,FORMO
     161 003662                                  CLOP    CO$BNO,BINOUT
     162 003674                                  CLOP    CO$BNI,BININ
     163 003704                                  CLOP    CO$CR,CR
     164 003712                                  CLOP    CO$CTL,CTRL
     165 003722                                  CLOP    CO$DTR,DTR
     166 003730                          CLOPND:
```

```
                                               .SBTTL          JOBS
                                       ;--------------------------------------------------------------------------
                                       ;  Show information about all jobs.
                                       ;
                                       ;  Define /PPN option for SHOW JOBS command
                                       ;
 003730                                        TBLDEF  SJO,1
 000336                                        CMDDEF  P*PN,0
 000342                                        TBLEND
                                       ;
 003730  010146                        SHOJOB: MOV     R1,-(SP)
 003732  010246                                MOV     R2,-(SP)
 003734  010346                                MOV     R3,-(SP)
 003736  010446                                MOV     R4,-(SP)
 003740  010546                                MOV     R5,-(SP)
                                       ;
                                       ;  See if we should display PPN instead of User names
                                       ;
 003742  105713                                TSTB    (R3)            ;ANY COMMAND OPTIONS?
 003744  001420                                BEQ     40$             ;BRANCH IF NOT
 003746  121327  000057                        CMPB    (R3),#'/        ;VALID SWITCH LEADIN?
 003752  001015                                BNE     40$             ;IGNORE IF NOT
 003754  005203                                INC     R3              ;SKIP /
 003756  012704  000334'                       MOV     #SJOHD,R4       ;POINT TO SHOW JOBS OPTION TABLE
 003762  004767  000000G                       CALL    SEARCH          ;VALID OPTION?
 003766  103004                                BCC     41$             ;REMEMBER /PPN IF SO
 003770                                        FABORT  #INVOPT
 004000  112767  000001  000000G       41$:    MOVB    #1,SJSPPN       ;/PPN ONLY VALID OPTION AT THIS TIME
                                       ;
                                       ;  Print system usage statistics
                                       ;
 004006  004767  007122                40$:    CALL    PRTUSE          ;PRINT SYSTEM USAGE STATISTICS
                                       ;
                                       ;  Print heading lines
                                       ;
 004012                                        .PRINT  #SYHD1          ;PRINT FIRST LINE OF JOB HEADING
 004020                                        .PRINT  #SYHD2          ;AND UNDERLINE IT
                                       ;
                                       ;  Begin loop to display information about each active line
                                       ;
 004026  012701  000002                        MOV     #2,R1           ;GET 1ST LINE INDEX #
                                       ;
                                       ;  See if line is active
                                       ;
 004032  032761  000000G 000000G       1$:     BIT     #$KINIT,LSW(R1) ;HAS LINE BEEN INITIALIZED?
 004040  001002                                BNE     20$             ;BR IF YES
 004042  000167  000624                        JMP     30$             ;TRY NEXT LINE
                                       ;
                                       ;  Print line number
                                       ;
 004046  010105                        20$:    MOV     R1,R5           ;GET LINE INDEX #
 004050  004767  000000G                       CALL    PRTLN           ;PRINT THE LINE #
                                       ;
                                       ;  Print '*' if this is our line
                                       ;
 004054  112700  000040                        MOVB    #' ,R0          ;ASSUME THAT IT'S NOT OUR LINE
 004060  120167  000000G                       CMPB    R1,CORUSR       ;IS THIS OUR LINE?
```

```
     58 004064  001002                        BNE     2$              ;BR IF NOT
     59 004066  112700  000052                MOVB    #'*,R0          ;PRINT * IF YES
     60 004072                          2$:   .TTYOUT
     61 004076                                .PRINT  #SPACE2         ;PRINT 2 SPACES
     62                                 ;
     63                                 ;  Print information about owner jobs
     64                                 ;
     65 004104  020127  000000G                CMP     R1,#LSTPL       ;IS THIS A PRIMARY LINE?
     66 004110  101407                         BLOS    3$              ;BR IF PRIMARY LINE
     67 004112  020127  000000G                CMP     R1,#LSTDL       ;DETACHED JOB?
     68 004116  101004                         BHI     3$              ;BR IF NOT
     69 004120                                .PRINT  #DETTXT         ;PRINT "Detached"
     70 004126  000443                         BR      18$
     71 004130  116105  000000G          3$:   MOVB    LNPRIM(R1),R5   ;GET OWNER LINE NUMBER
     72 004134  004767  000000G                CALL    PRTLN           ;PRINT OWNER LINE NUMBER
     73 004140                                .TTYOUT #'(             ;ENCLOSE VIRTUAL LINE # IN ( )
     74 004150  020127  000000G                CMP     R1,#LSTPL       ;IS THIS A PRIMARY LINE?
     75 004154  101412                         BLOS    19$             ;BR IF YES
     76 004156  016504  000000G                MOV     LSECPT(R5),R4   ;POINT TO VIRTUAL LINE # TABLE FOR OWNER JOB
     77 004162  012700  000000G                MOV     #MAXSEC,R0      ;GET HIGHEST VIRTUAL LINE #
     78 004166  012705  000001                 MOV     #1,R5           ;START WITH VIRTUAL LINE #1
     79 004172  120124                   21$:  CMPB    R1,(R4)+        ;SEARCH FOR LINE WITHIN TABLE
     80 004174  001403                         BEQ     22$             ;BR IF FOUND RIGHT ONE
     81 004176  005205                         INC     R5              ;INC VIRTUAL LINE #
     82 004200  077004                         SOB     R0,21$          ;LOOP TILL FOUND
     83 004202  005005                   19$:  CLR     R5              ;SAY THIS IS LINE 0
     84 004204  004767  000000G          22$:  CALL    PRTDEC          ;PRINT VIRTUAL LINE #
     85 004210                                .TTYOUT #')             ;CLOSE VIRTUAL LINE #
     86 004220  020527  000011                 CMP     R5,#9.          ;1 OR 2 DIGIT VIRTUAL LINE #?
     87 004224  101004                         BHI     18$             ;BR IF 2 DIGIT
     88 004226                                .TTYOUT #40             ;PRINT A SPACE
     89 004236                           18$:  .PRINT  #SPACE2         ;PRINT 2 SPACES
     90                                 ;
     91                                 ;  Print job priority
     92                                 ;
     93 004244  116105  000000G                MOVB    LPRI(R1),R5     ;GET JOB'S PRIORITY VALUE
     94 004250  012703  000003                 MOV     #3,R3           ;PRINT IN 3 SPACES
     95 004254  004767  000000G                CALL    PRTFIX
     96 004260                                .PRINT  #SPACE2         ;PRINT 2 SPACES
     97                                 ;
     98                                 ;  Print current line state
     99                                 ;
    100 004266  005003                         CLR     R3
    101 004270  016102  000000G                MOV     LSTATE(R1),R2   ;GET USER'S CURRENT EXECUTION STATE
    102 004274  020263  004730'          4$:   CMP     R2,STBIT(R3)    ;LOOK UP THE STATE
    103 004300  001416                         BEQ     6$              ;BR WHEN FOUND
    104 004302  062703  000002                 ADD     #2,R3           ;TRY NEXT FLAG
    105 004306  020327  000050                 CMP     R3,#NUMST       ;DONE ALL?
    106 004312  103770                         BLO     4$              ;BR IF NOT
    107 004314  012700  000000G                MOV     #RNMS,R0        ;ASSUME 'RN' STATE
    108 004320  020227  000000G                CMP     R2,#S$$RUN      ;IS JOB EXECUTABLE?
    109 004324  101402                         BLOS    25$             ;USE 'RN' IF SO
    110 004326  012700  000000G                MOV     #WTMS,R0        ;GENERIC WAIT IF STATE NOT RN OR IN TABLE
    111 004332                           25$:  .PRINT                  ;PRINT 'RN' OR 'WT' AS STATE
    112 004334  000406                         BR      5$
    113 004336  016367  005000' 000504   6$:   MOV     STNAM(R3),STPBUF;MOVE STATE NAME TO PRINT BUFFER
    114 004344                                .PRINT  #STPBUF         ;PRINT STATE NAME
```

```
                                        ;
                                        ;  Print "SWP" if job is outswapped
                                        ;
004352  032761  000000G 000000G 5$:     BIT     #$INCOR,LSW(R1) ;IS JOB IN MEMORY NOW?
004360  001004                          BNE     12$             ;BR IF YES
004362                                  .PRINT  #SWPTX          ;PRINT "SWP"
004370  000413                          BR      13$
                                        ;
                                        ;  Print "Lock" if job is locked in memory
                                        ;
004372  032761  000000G 000000G 12$:    BIT     #$MLOCK,LSW6(R1);IS JOB LOCKED IN MEMORY?
004400  001404                          BEQ     14$             ;BR IF NOT
004402                                  .PRINT  #LOCKTX         ;PRINT "LOCK"
004410  000403                          BR      13$
004412                                  14$:    .PRINT  #SPACE5         ;PRINT SPACES
                                        ;
                                        ;  Pring current memory size
                                        ;
004420                                  13$:    .TTYOUT #40             ;PRINT A SPACE
004430                                          .TTYOUT #40             ;PRINT ANOTHER SPACE
004440  016105  000000G                 MOV     LNBLKS(R1),R5   ;GET # 256-WORD BLOCKS ASSIGNED TO JOB
004444  066105  000000G                 ADD     LNSBLK(R1),R5   ;ADD # BLOCKS USED FOR PLAS REGIONS
004450  005205                          INC     R5              ;ROUND UP
004452  006205                          ASR     R5              ;CVT TO # K BYTES
004454  004767  000000G                 CALL    PRTDC3          ;PRINT # K-BYTES OF MEMORY USED
004460                                  .PRINT  #KBMSG          ;PRINT "Kb"
004466                                  .PRINT  #SPACE2         ;PUT IN 2 SPACES
                                        ;
                                        ;  Print job connect time
                                        ;
004474  016705  000000G                 MOV     MINTIM,R5       ;GET CURRENT MINUTE TIMER VALUE
004500  166105  000000G                 SUB     LCONTM(R1),R5   ;CALCULATE CONNECT TIME FOR LINE
004504  005205                          INC     R5              ;CHARGE A MINIMUM OF 1 MINUTE
004506  005004                          CLR     R4              ;CLEAR HIGH-ORDER FOR DIVIDE
004510  012703  000074                 MOV     #60.,R3         ;SET TO DIVIDE BY 60.
004514  004767  000000G                 CALL    DIVIDE          ;DIVIDE BY 60 TO GET HOURS AND MINUTES
004520  010046                          MOV     R0,-(SP)        ;SAVE NUMBER OF MINUTES
004522  020527  000144                 CMP     R5,#100.        ;DO WE NEED MORE THAN 2 DIGITS FOR HOURS?
004526  103405                          BLO     23$             ;BR IF NOT
004530  012703  000003                 MOV     #3,R3           ;PRINT 3 DIGITS
004534  004767  000000G                 CALL    PRTFIX          ;PRINT # HOURS
004540  000406                          BR      24$
004542                                  23$:    .TTYOUT #40             ;PRINT A SPACE
004552  004767  000000G                 CALL    PRTDC2          ;PRINT # HOURS WITH 2 DIGITS
004556  012605                          24$:    MOV     (SP)+,R5        ;GET # MINUTES CONNECTED
004560                                          .TTYOUT #':             ;PRINT COLON AFTER HOURS
004570  004767  000000G                 CALL    PRTDC2          ;PRINT # MINUTES CONNECTED
004574                                  .PRINT  #COL00          ;PRINT ':00' SECONDS
                                        ;
                                        ;  Print cpu time
                                        ;
004602                                  .PRINT  #SPACE2         ;PRINT 2 SPACES
004610  016104  000000G                 MOV     LCPUHI(R1),R4   ;GET HIGH ORDER CPU TIME (CLOCK TICKS)
004614  016105  000000G                 MOV     LCPULO(R1),R5   ;GET LOW ORDER CPU TIME (CLOCK TICKS)
004620  004767  000000G                 CALL    PRTTMV          ;PRINT TIME VALUE
                                        ;
                                        ;  Print name of running program
```

```
                                        ;
004624                                          .PRINT  #SPACE2
004632  016100  000000G                         MOV     LPRG1(R1),R0    ;GET 1ST 3 CHARS OF NAME (RAD50)
004636  004767  000000G                         CALL    PRTR50          ;PRINT THEM
004642  016100  000000G                         MOV     LPRG2(R1),R0    ;PRINT 2ND 3 CHARS
004646  004767  000000G                         CALL    PRTR50
                                        ;
                                        ;  Print user name or PPN
                                        ;
004652                                          .PRINT  #SPACE3         ;PRINT 3 SPACES
004660  004767  000000G                         CALL    PRTUNM          ;PRINT USER NAME
004664                                          .PRINT  #CRLF
                                        ;
                                        ;  Do next line
                                        ;
004672  062701  000002          30$:    ADD     #2,R1           ;GET NEXT LINE INDEX #
004676  020127  000000G                 CMP     R1,#LSTSL       ;CHECKED ALL?
004702  101002                          BHI     99$             ;BR IF YES
004704  000167  177122                  JMP     1$              ;DO NEXT LINE
                                        ;
                                        ;  Finished
                                        ;
004710  105067  000000G          99$:    CLRB    SJSPPN          ;TURN OFF /PPN SWITCH
004714  012605                          MOV     (SP)+,R5
004716  012604                          MOV     (SP)+,R4
004720  012603                          MOV     (SP)+,R3
004722  012602                          MOV     (SP)+,R2
004724  012601                          MOV     (SP)+,R1
004726  000207                          RETURN

                                ;
                                ;-------------------------
                                ;  JOB STATE CODES
                                ;
004730  000000G                 STBIT:  .WORD   S$RT            ;REAL-TIME
004732  000000G                         .WORD   S$TTSC          ;TT INPUT DONE AND SINGLE CHAR ACTIVATION
004734  000000G                         .WORD   S$TTFN          ;TT INPUT DONE
004736  000000G                         .WORD   S$OTFN          ;TT OUTPUT BUFFER EMPTY
004740  000000G                         .WORD   S$HICP          ;INTERACTIVE JOB COMPUTATION
004742  000000G                         .WORD   S$TWFN          ;TIMED WAIT FINISH
004744  000000G                         .WORD   S$OTLO          ;OUTPUT BUFFER ALMOST EMPTY
004746  000000G                         .WORD   S$IOFN          ;I/O WAIT COMPLETION
004750  000000G                         .WORD   S$LOW           ;LOW PRIORITY COMPUTATION
004752  000000G                         .WORD   S$INWT          ;INPUT WAIT
004754  000000G                         .WORD   S$OTWT          ;OUTPUT WAIT
004756  000000G                         .WORD   S$TMWT          ;TIMED WAIT
004760  000000G                         .WORD   S$SPND          ;JOB HAS DONE A .SPND
004762  000000G                         .WORD   S$MSWT          ;MESSAGE WAIT
004764  000000G                         .WORD   S$QUSR          ;WAITING FOR USR
004766  000000G                         .WORD   S$IOWT          ;WAITING FOR I/O TO FINISH
004770  000000G                         .WORD   S$SFWT          ;WAITING FOR SHARED FILE RECORD
004772  000000G                         .WORD   S$QMIO          ;WAITING FOR MAPPED I/O BUFFER
004774  000000G                         .WORD   S$SPDB          ;WAITING FOR SPOOLED FILE SPACE
004776  000000G                         .WORD   S$SPCB          ;WAITING FOR SPOOLED FILE
000050                          NUMST   =       <.-STBIT>
                                ;
                                ;  PARALLEL VECTOR OF STATE NAMES
```

```
    229                                         ;
    230 005000      122     124         STNAM:  .ASCII  /RT/            ;S$RT
    231 005002      111     116                 .ASCII  /IN/            ;S$TTSC
    232 005004      111     116                 .ASCII  /IN/            ;S$TTFN
    233 005006      111     116                 .ASCII  /IN/            ;S$OTFN
    234 005010      111     116                 .ASCII  /IN/            ;S$HICP
    235 005012      110     111                 .ASCII  /HI/            ;S$TWFN
    236 005014      110     111                 .ASCII  /HI/            ;S$OTLO
    237 005016      110     111                 .ASCII  /HI/            ;S$IOFN
    238 005020      114     117                 .ASCII  /LO/            ;S$LOW
    239 005022      124     111                 .ASCII  /TI/            ;S$INWT
    240 005024      124     117                 .ASCII  /TO/            ;S$OTWT
    241 005026      123     114                 .ASCII  /SL/            ;S$TMWT
    242 005030      123     114                 .ASCII  /SL/            ;S$SPND
    243 005032      115     123                 .ASCII  /MS/            ;S$MSWT
    244 005034      125     123                 .ASCII  /US/            ;S$QUSR
    245 005036      111     117                 .ASCII  /IO/            ;S$IOWT
    246 005040      123     106                 .ASCII  /SF/            ;S$SFWT
    247 005042      115     111                 .ASCII  /MI/            ;S$QMIO
    248 005044      123     120                 .ASCII  /SP/            ;S$SPDB
    249 005046      123     120                 .ASCII  /SP/            ;S$SPCB
    250                                         ;
    251 005050      130     130     200 STPBUF: .ASCII  /XX/<200>
    252                                 .EVEN
```

```
                                        .SBTTL  .       QUEUE
                                ;-------------------------------------------------------------------------------
                                ;
                                ;   SHOW QUEUE
                                ;
005054  010146                  SHOQUE: MOV     R1,-(SP)
005056  012701  000000G                 MOV     #SDCB,R1        ;POINT TO FIRST SDCB
005062  020127  000000G         4$:     CMP     R1,#SDCBND      ;CHECKED ALL SDCB'S?
005066  103024                          BHIS    3$              ;BR IF YES -- THERE ARE NO QUEUED FILES
005070  005761  000000G                 TST     SDFHD(R1)       ;ANY FILES QUEUED FOR THIS DEVICE?
005074  001003                          BNE     1$              ;BR IF YES
005076  062701  000000G                 ADD     #SDCBSZ,R1      ;POINT TO NEXT SDCB
005102  000767                          BR      4$              ;GO CHECK IT
005104                          1$:     .PRINT  #QHDMS1         ;PRINT QUEUE HEAD MESSAGE
005112                                  .PRINT  #QHDMS2         ;Underline the heading
005120  004767  000000G         5$:     CALL    LSTSPL          ;LIST INFO ABOUT FILES QUEUED FOR THIS DEV
005124  062701  000000G                 ADD     #SDCBSZ,R1      ;POINT TO NEXT SDCB
005130  020127  000000G                 CMP     R1,#SDCBND      ;DONE ALL SDCB'S?
005134  103771                          BLO     5$              ;BR IF MORE TO DO
005136  000403                          BR      2$              ;FINISHED
005140                          3$:     .PRINT  #NOFIL          ;THERE ARE NO PENDING SPOOL FILES
005146  012601                  2$:     MOV     (SP)+,R1
005150  000207                          RETURN
```

```
                                        .SBTTL  .       COMMANDS
                                ;-----------------------------------------------------------------------
                                ;  SHOW COMMANDS Calls TSXUCL to display specific or all command definitions.
                                ;
                                ;  Inputs:
                                ;       R3      Points to next keyword in command buffer or end of command
                                ;       R4      Points to next word in SHOW command table
                                ;
                                ;  Outputs:
                                ;       R3      Points to end of input command in CMDBUF
                                ;       R4      Points to end of string to be passed to TSXUCL in CMDBUF
                                ;
005152                          SHOCMD:
                                ;
                                ;  See if there are any user defined commands
                                ;
005152  005767  0000000G                TST     UCLBLK          ;Any user defined commands?
005156  001410                          BEQ     1$              ;Br if not
                                ;
                                ;  There may be user defined commands.
                                ;  Pass any specific command to be displayed to TSXUCL.
                                ;  Enter TSXUCL to show the commands.
                                ;
005160  012704  0000000G                MOV     #CMDBUF,R4      ;Point to beginning of command buffer
                                                                ;Probably holds "SHOW ... " at this point
005164  112724  000077                  MOVB    #'?,(R4)+       ;Tell TSXUCL we want to show commands
005170  112324                  3$:     MOVB    (R3)+,(R4)+     ;Pass next character from command buffer
005172  001376                          BNE     3$              ;Until remainder of command line passed
005174  000167  0000000G                JMP     CALUCL          ;Call TSXUCL
                                ;
                                ;  There are no user defined commands
                                ;
005200                          1$:     .PRINT  #NOUDC          ;No user defined commands
005206  000207                          RETURN
```

```
                                        .SBTTL  .       DEVICES
                                ;-----------------------------------------------------------------------
                                ;  SHOW DEVICES
                                ;
005210  010246                  SHODEV: MOV     R2,-(SP)
                                ;
                                ;  Print heading lines
                                ;
005212                                  .PRINT  #DVSHH1
005220                                  .PRINT  #DVSHH2
005226                                  .PRINT  #DVSHH3
                                ;
                                ;  List status of each device that is installed in the system
                                ;
005234  005002                          CLR     R2              ;Start with the first device
005236  004767  000016          1$:     CALL    DEVDSP          ;Display status about this device
005242  062702  000002                  ADD     #2,R2           ;Get next device index
005246  020267  000000G                 CMP     R2,NUMDEV       ;Finished all devices?
005252  101771                          BLOS    1$              ;Loop if not
                                ;
                                ;  Finished
                                ;
005254  012602                          MOV     (SP)+,R2
005256  000207                          RETURN
```

```
                                        ;-------------------------------------------------------------------------
                                        ;  Display a line of status information for a device.
                                        ;
                                        ;  Inputs:
                                        ;   R2 = Device index number.
                                        ;
005260  010346                          DEVDSP: MOV     R3,-(SP)
005262  010546                                  MOV     R5,-(SP)
                                        ;
                                        ;  Print the device name
                                        ;
005264  012703  000002                          MOV     #2,R3           ;Print 2 spaces
005270  004767  000000G                         CALL    PRTSPC
005274  016200  000000G                         MOV     PNAME(R2),R0    ;Get RAD50 device name
005300  004767  000000G                         CALL    PRTR50          ;Print it
                                        ;
                                        ;  Print the current I/O count
                                        ;
005304  012703  000003                          MOV     #3,R3           ;Print spaces
005310  004767  000000G                         CALL    PRTSPC
005314  016205  000000G                         MOV     HANIOC(R2),R5   ;Get # pending I/O operations for dev
005320  012703  000005                          MOV     #5,R3           ;Print in 5 column field
005324  004767  000000G                         CALL    PRTFIX          ;Print it
                                        ;
                                        ;  Print the device status word
                                        ;
005330  012703  000003                          MOV     #3,R3           ;Print spaces
005334  004767  000000G                         CALL    PRTSPC
005340  016205  000000G                         MOV     DVSTAT(R2),R5   ;Get status word value
005344  012703  000006                          MOV     #6,R3           ;Print 6 digits
005350  004767  000000G                         CALL    OCTFIX          ;Print octal value
                                        ;
                                        ;  Print the handler virtual base address
                                        ;
005354  012703  000003                          MOV     #3,R3           ;Print spaces
005360  004767  000000G                         CALL    PRTSPC
005364  016205  000000G                         MOV     HANENT(R2),R5   ;Get base address of handler
005370  020527  000100                          CMP     R5,#100         ;Is this a real or pseudo handler?
005374  103440                                  BLO     9$              ;Branch of pseudo handler
005376  162705  000006                          SUB     #6,R5           ;Get actual handler base address
005402  012703  000006                          MOV     #6,R3           ;Print 6 digits
005406  004767  000000G                         CALL    OCTFIX          ;Print octal value
                                        ;
                                        ;  Print the handler physical base address
                                        ;
005412  012703  000003                          MOV     #3,R3           ;Print spaces
005416  004767  000000G                         CALL    PRTSPC
005422  016205  000000G                         MOV     HANPAR(R2),R5   ;Get physical base address of handler
005426  012703  000006                          MOV     #6,R3           ;Print 6 digits
005432  004767  000000G                         CALL    OCTFIX          ;Print octal value
                                        ;
                                        ;  Print the handler size
                                        ;
005436  012703  000002                          MOV     #2,R3           ;Print 2 spaces
005442  004767  000000G                         CALL    PRTSPC
005446  016205  000000G                         MOV     HANSIZ(R2),R5   ;Get the handler size
005452  012703  000005                          MOV     #5,R3           ;Print 5 digits
```

```
      58 005456  004767  000000G                 CALL    PRTFIX          ;Print decimal value
      59 005462  012703  000002                  MOV     #2,R3           ;Print spaces
      60 005466  004767  000000G                 CALL    PRTSPC
      61                                ;
      62                                ;  Print CSR and Vector addresses for this device
      63                                ;
      64 005472  004767  000014                  CALL    DSPCSR
      65                                ;
      66                                ;  Terminate this print line
      67                                ;
      68 005476                         9$:     .PRINT  #CRLF           ;End the print line
      69                                ;
      70                                ;  Finished
      71                                ;
      72 005504  012605                          MOV     (SP)+,R5
      73 005506  012603                          MOV     (SP)+,R3
      74 005510  000207                          RETURN
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Extract CSR and Vector addresses from a device handler and display them.
                                ;
                                ;  Inputs:
                                ;   R2 = Device index number.
                                ;
005512  010146                  DSPCSR: MOV     R1,-(SP)
005514  010246                          MOV     R2,-(SP)
005516  010346                          MOV     R3,-(SP)
005520  010446                          MOV     R4,-(SP)
005522  010546                          MOV     R5,-(SP)
                                ;
                                ;  Build file spec for device handler
                                ;
005524  016767  0000000G 172246         MOV     SYNAME,HANNAM   ;Set physical device name for SY device
005532  016267  0000000G 172242         MOV     PNAME(R2),HANNAM+2 ;Set RAD50 device name
                                ;
                                ;  Try to open the device handler file
                                ;
005540                                  .SERR                   ;Don't abort on lookup errors
005546                                  .LOOKUP #XAREA,#1,#HANNAM ;Try to open handler file
005566  103547                          BCS     9$              ;Br if unable to open handler
                                ;
                                ;  Read block 0 of handler and save information about CSR address
                                ;
005570                                  .READW  #XAREA,#1,#BLK0,#256.,#0 ;Read block 0 of handler
005626  103527                          BCS     9$
005630  016767  000001C 172154         MOV     BLK0+H.CSR,DCSR ;Save CSR info
                                ;
                                ;  Read blocks 1 and 2 of handler
                                ;
005636                                  .READW  #XAREA,#1,#BLK0,#512.,#1 ;Read blocks 1 and 2 of handler
005676  016767  000001C 172104         MOV     BLK0-1000+H.VEC,DVEC    ;Save vector info
                                ;
                                ;  If both the CSR and vector are zero, don't display anything
                                ;
005704  016700  172102                  MOV     DCSR,R0         ;Get CSR value
005710  056700  172074                  BIS     DVEC,R0         ;Add Vector value
005714  001474                          BEQ     9$              ;Br if both are zero
                                ;
                                ;  Determine if this is a handler for a PRO device with floating
                                ;  vector and CSR addresses.
                                ;
005716  005001                          CLR     R1              ;Assume this is not a PRO device
005720  012704  000000C                 MOV     #BLK0-1000+H.VEC,R4     ;Point to handler vector cell
005724  012400                          MOV     (R4)+,R0        ;Get address of vector
005726  002013                          BGE     3$              ;Br if not pointer to vector list
005730  006300                          ASL     R0              ;Get byte offset to vector list
005732  060004                          ADD     R0,R4           ;Point to start of vector list
005734  005714                          TST     (R4)            ;Is this a PRO device with floating addresses?
005736  002007                          BGE     3$              ;Br if not
005740  005724                          TST     (R4)+           ;Point to word with PRO device ID
005742  012401                          MOV     (R4)+,R1        ;Get PRO device ID
005744  010100                          MOV     R1,R0           ;Get PRO device ID
005746  004767  000226                 CALL    PIDCSR          ;Get CSR address for PRO device
005752  010005                          MOV     R0,R5           ;Get to R5 for OCTFIX
005754  000402                          BR      4$              ;Go display it
```

```
                                        ;
                                        ;  Display the CSR value
                                        ;
006756  016705  172030        3$:     MOV     DCSR,R5         ;Get CSR value
005762  012703  000002        4$:     MOV     #2,R3           ;Print 2 spaces
005766  004767  000000G               CALL    PRTSPC
005772  012703  000006               MOV     #6,R3           ;Print 6 digits
005776  004767  000000G               CALL    OCTFIX          ;Print octal value
                                        ;
                                        ;  Display vector addresses
                                        ;
006002  012703  000002               MOV     #2,R3           ;Print 2 spaces
006006  004767  000000G               CALL    PRTSPC
006012  016705  171772               MOV     DVEC,R5         ;Single vector or vector list?
006016  001433                       BEQ     9$              ;Br if no vector
006020  002405                       BLT     1$              ;Br if multiple vector
006022  012703  000003               MOV     #3,R3           ;Print 3 digits
006026  004767  000000G               CALL    OCTFIX          ;Print octal value
006032  000425                       BR      9$
                                        ;
                                        ;  We have multiple vectors (and possibly floating PRO vectors)
                                        ;
006034  005701                1$:     TST     R1              ;Do we have floating vectors for PRO device?
006036  001404                       BEQ     5$              ;Br if not
006040  010100                       MOV     R1,R0           ;Get PRO device ID
006042  004767  000154               CALL    PIDVEC          ;Get base vector based on device ID
006046  010001                       MOV     R0,R1           ;Save base vector location for device
006050  011405                5$:     MOV     (R4),R5         ;Get address of next vector
006052  060105                       ADD     R1,R5           ;Add base vector address
006054  012703  000003               MOV     #3,R3           ;Print 3 digits
006060  004767  000000G               CALL    OCTFIX          ;Print octal value
006064  062704  000006               ADD     #6,R4           ;Point to next vector entry
006070  005714                       TST     (R4)            ;Are there more vectors?
006072  003405                       BLE     9$              ;Br if not
006074  012703  000001               MOV     #1,R3           ;Print 1 space
006100  004767  000000G               CALL    PRTSPC
006104  000761                       BR      5$              ;Print next vector
                                ;
                                ;  Finished
                                ;
006106                        9$:     .CLOSE  #1              ;Close handler file
006114                                .HERR                   ;Reset error aborts
006122  012605                       MOV     (SP)+,R5
006124  012604                       MOV     (SP)+,R4
006126  012603                       MOV     (SP)+,R3
006130  012602                       MOV     (SP)+,R2
006132  012601                       MOV     (SP)+,R1
006134  000207                       RETURN
```

```
                                        ;-----------------------------------------------------------------------
                                        ;  Convert a PRO device ID number into the option slot # for the device.
                                        ;
                                        ;  Inputs:
                                        ;   R0 = PRO device ID.
                                        ;
                                        ;  Outputs:
                                        ;   C-flag set ==> Could not find device ID.
                                        ;   R0 = Option slot number.
                                        ;
006136  010146                  PIDSLT: MOV     R1,-(SP)
006140  012701  0000000                 MOV     #PROSLT,R1      ;Point to entry for slot 0
006144  021127  177777          1$:     CMP     (R1),#-1        ;Checked all entries?
006150  001410                          BEQ     2$              ;Br if yes
006152  020021                          CMP     R0,(R1)+        ;Search for correct entry
006154  001373                          BNE     1$              ;Loop if this is not it
006156  162701  0000020                 SUB     #PROSLT+2,R1    ;Get byte index for slot
006162  006201                          ASR     R1              ;Get word index
006164  010100                          MOV     R1,R0           ;Return in R0
006166  000241                          CLC                     ;Signal success on return
006170  000401                          BR      9$
006172  000261                  2$:     SEC                     ;Signal failure on return
006174  012601                  9$:     MOV     (SP)+,R1
006176  000207                          RETURN

                                        ;-----------------------------------------------------------------------
                                        ;  Determine the address of the CSR for a PRO device given the device ID.
                                        ;
                                        ;  Inputs:
                                        ;   R0 = PRO device ID
                                        ;
                                        ;  Outputs:
                                        ;   C-flag set ==> Do not recognize the device.
                                        ;   R0 = CSR address
                                        ;
006200  004767  177732          PIDCSR: CALL    PIDSLT          ;Determine slot where device is installed
006204  103405                          BCS     9$              ;Br if don't recognize device
006206  072027  000007                  ASH     #7.,R0          ;CSR addresses are 200 apart per slot
006212  062700  174000                  ADD     #174000,R0      ;CSR for slot 0 is here
006216  000241                          CLC                     ;Signal success on return
006220  000207                  9$:     RETURN

                                        ;-----------------------------------------------------------------------
                                        ;  Determine the base vector for a PRO device given the device ID.
                                        ;
                                        ;  Inputs:
                                        ;   R0 = PRO device ID
                                        ;
                                        ;  Outputs:
                                        ;   C-flag set ==> Did not recognize device.
                                        ;   R0 = Base vector location for device.
                                        ;
006222  004767  177710          PIDVEC: CALL    PIDSLT          ;Determine what slot has controller
006226  103405                          BCS     9$              ;Br if don't recognize device
006230  072027  000003                  ASH     #3.,R0          ;Vectors are 8 bytes apart per slot
006234  062700  000300                  ADD     #300,R0         ;Vector for slot 0 is at 300
006240  000241                          CLC                     ;Signal success on return
```

```
      58 006242  000207                    9$:     RETURN
```

```
                                        .SBTTL  .       ASSIGNS
                                ;-------------------------------------------------------------------------------
                                ;  Display assignments
                                ;
006244  010246                  SHOASN: MOV     R2,-(SP)
006246                                  .PRINT  #ASNHD1         ;PRINT HEADING
                                ;
                                ;  If there is no assignment for SY, print default assignment
                                ;
006254  012702  000000G                 MOV     #ASNTBL,R2      ;Point to assign table
006260  026762  000000G 000000G 5$:     CMP     R50SY,AT$LOG(R2);Is this entry for SY?
006266  001423                          BEQ     6$              ;Br if yes
006270  062702  000000G                 ADD     #AT$$SZ,R2      ;Point to next entry
006274  020227  000000G                 CMP     R2,#ASNEND      ;Checked all assign entries?
006300  103767                          BLO     5$              ;Loop if not
006302                                  .PRINT  #SYASHD         ;Print "SY --> "
006310  016700  000000G                 MOV     SYNAME,R0       ;Get default device
006314  004767  000000G                 CALL    PRTR50
006320                                  .TTYOUT #':
006330                                  .PRINT  #CRLF
                                ;
                                ;  If there is no assignment for DK, print default assignment
                                ;
006336  012702  000000G         6$:     MOV     #ASNTBL,R2      ;Point to assign table
006342  026762  000000G 000000G /$:     CMP     R50DK,AT$LOG(R2);Is this entry for DK?
006350  001423                          BEQ     8$              ;Br if yes
006352  062702  000000G                 ADD     #AT$$SZ,R2      ;Point to next entry
006356  020227  000000G                 CMP     R2,#ASNEND      ;Checked all assign entries?
006362  103767                          BLO     7$              ;Loop if not
006364                                  .PRINT  #DKASHD         ;Print "DK --> "
006372  016700  000000G                 MOV     SYNAME,R0       ;Get default device
006376  004767  000000G                 CALL    PRTR50          ;Print it
006402                                  .TTYOUT #':
006412                                  .PRINT  #CRLF
                                ;
                                ;  Now print user assign table entries
                                ;
006420  012702  000000G         8$:     MOV     #ASNTBL,R2      ;POINT TO ASSIGN TABLE
006424  005762  000000G         2$:     TST     AT$LOG(R2)      ;IS THIS ASSIGN TABLE ENTRY IN USE?
006430  001461                          BEQ     3$              ;BR IF NOT
006432  016200  000000G                 MOV     AT$LOG(R2),R0           ;GET LOGICAL DEVICE NAME
006436  004767  000000G                 CALL    PRTR50          ;DISPLAY LOGICAL DEVICE NAME
006442                                  .PRINT  #ASNHD2         ;PRINT ARROW
006450  016200  000000G                 MOV     AT$DEV(R2),R0   ;GET PHYSICAL DEVICE NAME
006454  004767  000000G                 CALL    PRTR50          ;PRINT IT
006460                                  .TTYOUT #':             ;PUT IN COLON
006470  016200  000000G                 MOV     AT$FIL(R2),R0   ;GET 1ST 3 CHARS OF FILE NAME
006474  001434                          BEQ     4$              ;BR IF NO NAME
006476  004767  000000G                 CALL    PRTR50          ;PRINT 1ST PART OF NAME
006502  016200  000002G                 MOV     AT$FIL+2(R2),R0 ;GET 2ND 3 CHARS OF FILE NAME
006506  004767  000000G                 CALL    PRTR50          ;PRINT THE NAME
006512                                  .TTYOUT #'.             ;PUT IN A PERIOD
006522  016200  000000G                 MOV     AT$EXT(R2),R0   ;GET EXTENSION
006526  001417                          BEQ     4$              ;BR IF NONE SPECIFIED
006530  004767  000000G                 CALL    PRTR50          ;DISPLAY THE EXTENSION
006534  016205  000000G                 MOV     AT$SIZ(R2),R5   ;GET FILE SIZE
006540  001412                          BEQ     4$              ;BR IF NONE SPECIFIED
```

```
      58 006542                                        .TTYOUT #'[             ;START SIZE SPEC
      59 006552  004767  000000G                       CALL    PRTDEC          ;DISPLAY THE FILE SIZE
      60 006556                                        .TTYOUT #']             ;TERMINATE THE SIZE SPEC
      61 006566                              4$:       .PRINT  #CRLF           ;TERMINATE THE LINE
      62 006574  062702  000000G             3$:       ADD     #AT$$SZ,R2      ;POINT TO NEXT ASSIGN ENTRY
      63 006600  020227  000000G                       CMP     R2,#ASNEND      ;HAVE WE REACHED END OF TABLE?
      64 006604  103707                                BLO     2$              ;BR IF NOT
      65 006606  012602                                MOV     (SP)+,R2
      66 006610  000207                                RETURN
```

```
                                        .SBTTL  .       ALLOCATIONS
                                ;-------------------------------------------------------------------------------
                                ;  Show allocations
                                ;
006612  010146                  SHOALC: MOV     R1,-(SP)
006614  010246                          MOV     R2,-(SP)
006616  010346                          MOV     R3,-(SP)
006620  010546                          MOV     R5,-(SP)
                                ;
                                ;  Make a fast scan to see if there are any allocated devices
                                ;
006622  012702  000000G                 MOV     #ALCTBL,R2      ;Point to start of allocation table
006626  105762  000000G         1$:     TSTB    AD$JOB(R2)      ;Is this entry used?
006632  001011                          BNE     2$              ;Br if yes
006634  062702  000000G                 ADD     #AD$$SZ,R2      ;Point to next entry
006640  020227  000000G                 CMP     R2,#ALCEND      ;Checked all entries?
006644  103770                          BLO     1$              ;Loop if not
                                ;
                                ;  There are no allocated devices
                                ;
006646                                  .PRINT  #TM$NAD         ;No allocated devices
006654  000460                          BR      9$              ;Finished
                                ;
                                ;  There are some allocated devices.
                                ;  Print heading lines.
                                ;
006656                          2$:     .PRINT  #CRLF           ;Print a blank line
006664                                  .PRINT  #ALCHD1         ;Print heading line 1
006672                                  .PRINT  #ALCHD2         ;Print heading line 2
                                ;
                                ;  Begin loop to print information about each allocated device
                                ;
006700  012702  000000G                 MOV     #ALCTBL,R2      ;Point to start of allocation table
006704  116205  000000G         3$:     MOVB    AD$JOB(R2),R5   ;Is this entry used?
006710  001435                          BEQ     4$              ;Br if not
                                ;
                                ;  Print the device name
                                ;
006712                                  .TTYOUT #40             ;Put a space in front of the device name
006722  016200  000000G                 MOV     AD$DVU(R2),R0   ;Get device and unit number
006726  004767  000000G                 CALL    CVDVNM          ;Convert dev and unit # to device name
006732  004767  000000G                 CALL    PRTR50          ;Print the device name
006736  012703  000004                  MOV     #4.,R3          ;Print 4 spaces
006742  004767  000000G                 CALL    PRTSPC
                                ;
                                ;  Print the number of the job to which the device is allocated
                                ;
006746  012703  000002                  MOV     #2.,R3          ;Print job number in 2 col field
006752  010501                          MOV     R5,R1           ;Save job index number in R1
006754  006205                          ASR     R5              ;Convert job index number to job number
006756  004767  000000G                 CALL    PRTFIX          ;Print the job number
006762  012703  000003                  MOV     #3.,R3          ;Space over 3 columns
006766  004767  000000G                 CALL    PRTSPC
                                ;
                                ;  Print the user's name
                                ;
006772  004767  000000G                 CALL    PRTUNM          ;Print the user's name
```

```
                                        ;
                                        ;  Terminate this print line
                                        ;
006776                                          .PRINT  #CRLF           ;End of line
                                        ;
                                        ;  See if there are more allocated devices to display
                                        ;
007004  062702  000000G         4$:     ADD     #AD$$SZ,R2      ;Point to next allocation entry
007010  020227  000000G                 CMP     R2,#ALCEND      ;Finished all entries?
007014  103733                          BLO     3$              ;Br if more to do
                                        ;
                                        ;  Finished
                                        ;
007016  012605                  9$:     MOV     (SP)+,R5
007020  012603                          MOV     (SP)+,R3
007022  012602                          MOV     (SP)+,R2
007024  012601                          MOV     (SP)+,R1
007026  000207                          RETURN
```

```
                                        .SBTTL  MOUNTS
                                ;-------------------------------------------------------------------------------
                                ;  Show mounts
                                ;
007030  010146                  SHOMNT: MOV     R1,-(SP)
007032  010246                          MOV     R2,-(SP)
007034  010346                          MOV     R3,-(SP)
007036  010446                          MOV     R4,-(SP)
007040  010546                          MOV     R5,-(SP)
                                ;
                                ;  Set up a table of pointers to active mount entries in BLKO area.
                                ;
007042  016704  000000G                 MOV     CSHDEV,R4       ;Point to start of mount table
007046  012705  000000G                 MOV     #BLKO,R5        ;Point to area where we will build table
007052  010400                  20$:    MOV     R4,R0           ;Get address of next mount entry
007054  004767  000000G                 CALL    CDGET           ;Get mount entry into CDBUF
007060  005767  000001C                 TST     CDBUF+CD$DVU    ;Is this mount entry in use?
007064  001404                          BEQ     21$             ;Br if not
007066  020527  001774G                 CMP     R5,#BLKO+1020.  ;Is table full?
007072  103001                          BHIS    21$             ;Br if yes
007074  010425                          MOV     R4,(R5)+        ;Save pointer to active mount entry
007076  062704  000000G          21$:    ADD     #CD$$SZ,R4      ;Point to next entry in mount table
007102  020467  000000G                 CMP     R4,CSHDVN       ;Have we checked all mount entries?
007106  103761                          BLO     20$             ;Loop if not
007110  005015                          CLR     (R5)            ;Put null pointer at end of table
                                ;
                                ;  Now sort the list of mount table pointers using two keys:
                                ;  key 1 -- Device # & unit #;  key 2 -- base block number of logical disks
                                ;
007112  012705  000000G                 MOV     #BLKO,R5        ;Point to first entry in table
007116  012503                  24$:    MOV     (R5)+,R3        ;Get pointer to a mount entry
007120  001437                          BEQ     26$             ;Br if finished sorting table
007122  010504                          MOV     R5,R4           ;Get pointer to next entry in table
007124  010300                  23$:    MOV     R3,R0           ;Get address of mount entry
007126  004767  000000G                 CALL    CDGET           ;Copy entry into CDBUF
007132  016702  000001C                 MOV     CDBUF+CD$DVU,R2 ;Get device # and unit # of mounted device
007136  000302                          SWAB    R2              ;Put dev # in high-order, unit # in low-order
007140  016767  000001C 170650         MOV     CDBUF+CD$BAS,CDBASE ;Save base block number
007146  012401                  25$:    MOV     (R4)+,R1        ;Get pointer to another mount entry
007150  001762                          BEQ     24$             ;Br if checked all beyond reference entry
007152  010100                          MOV     R1,R0           ;Get address of mount entry
007154  004767  000000G                 CALL    CDGET           ;Copy entry into CDBUF
007160  016700  000001C                 MOV     CDBUF+CD$DVU,R0 ;Get device # and unit # of this entry
007164  000300                          SWAB    R0              ;Rearrange dev # and unit # order
007166  020002                          CMP     R0,R2           ;Look for smallest dev # and unit #
007170  101366                          BHI     25$             ;Br if this device is higher
007172  103404                          BLO     22$             ;Br if this device is smaller than ref one
007174  026767  000001C 170614         CMP     CDBUF+CD$BAS,CDBASE;Same device, now check base block #'s
007202  103361                          BHIS    25$             ;Br if not smaller
007204  010364  177776          22$:    MOV     R3,-2(R4)       ;Swap table entries
007210  010165  177776                  MOV     R1,-2(R5)
007214  010103                          MOV     R1,R3           ;Set new reference pointer
007216  000742                          BR      23$             ;Continue sort
                                ;
                                ;  If there are any mounted devices, print table heading
                                ;
007220  005767  000000G          26$:    TST     BLKO            ;Are there any mounted devices?
```

```
     58 007224  001406                        BEQ     30$             ;Br if not
     59 007226                                .PRINT  #SHMTH1         ;Print heading line
     60 007234                                .PRINT  #SHMTH2         ;Underline it
     61                                ;
     62                                ;  Now print information about each device.
     63                                ;
     64 007242  012702  000000G       30$:    MOV     #BLK0,R2        ;Get pointer to table of sorted pointers
     65 007246  012201                1$:     MOV     (R2)+,R1        ;Get next mount table pointer
     66 007250  001002                        BNE     27$             ;Br if got another entry
     67 007252  000167  000414                JMP     2$              ;Finished all devices
     68 007256  010100                27$:    MOV     R1,R0           ;Get address of mount entry
     69 007260  004767  000000G               CALL    CDGET           ;Move entry into CDBUf
     70 007264  016700  000001C               MOV     CDBUF+CD$DVU,R0 ;GET UNIT # / DEVICE # OF MOUNTED DEVICE
     71 007270  001576                        BEQ     8$              ;BR IF NULL -- THIS ENTRY IS FREE
     72                                ;
     73                                ;  Found a mounted device.
     74                                ;  Print device name and unit number
     75                                ;
     76 007272  004767  000000G               CALL    CVDVNM          ;CONVERT DEVICE & UNIT #'S TO DEVICE NAME
     77 007276  004767  000000G               CALL    PRTR50          ;PRINT THE DEVICE NAME
     78 007302                                .TTYOUT #':             ;PRINT ":"
     79                                ;
     80                                ;  If this is a logical disk, print the LD file name
     81                                ;
     82 007312  005767  000001C               TST     CDBUF+CD$BAS    ;IS THIS A LOGICAL DISK?
     83 007316  001005                        BNE     29$             ;BR IF YES
     84 007320                                .PRINT  #SPACE6         ;SPACE OVER 6 COLUMNS
     85 007326  000167  000234                JMP     4$
     86 007332  010146                29$:    MOV     R1,-(SP)        ;SAVE ORIGINAL MOUNT TABLE INDEX
     87 007334  005046                        CLR     -(SP)           ;PUT NULL ON STACK TO SIGNAL END
     88 007336  010146                12$:    MOV     R1,-(SP)        ;SAVE POINTER TO INNER-MOST LOGICAL DISK
     89 007340  016703  000001C               MOV     CDBUF+CD$BAS,R3 ;GET BASE BLOCK NUMBER OF THIS LD
     90 007344  005005                        CLR     R5              ;SAY NO ENCLOSING LD FOUND YET
     91 007346  016767  000001C 170444        MOV     CDBUF+CD$DVU,CDDVU ;Save device and unit info
     92 007354  016704  000000G               MOV     CSHDEV,R4       ;SEARCH THROUGH MOUNTED DEVICE TABLE
     93 007360  010400                13$:    MOV     R4,R0           ;Get address of mount entry
     94 007362  004767  000000G               CALL    CDGET           ;Move into CDBUF
     95 007366  026767  170426  000001C       CMP     CDDVU,CDBUF+CD$DVU; LOOKING FOR ONE WITH SAME PHYS DEVICE
     96 007374  001023                        BNE     14$             ;BR IF NOT THIS ONE
     97 007376  005767  000001C               TST     CDBUF+CD$BAS    ;IS THIS ALSO A LOGICAL DISK?
     98 007402  001420                        BEQ     14$             ;BR IF NOT
     99 007404  020367  000001C               CMP     R3,CDBUF+CD$BAS ;SEE IF THIS LD ENCLOSES INNER LD
    100 007410  101415                        BLOS    14$             ;BR IF NOT
    101 007412  020367  000001C               CMP     R3,CDBUF+CD$TOP ;CHECK UPPER RANGE
    102 007416  103012                        BHIS    14$             ;BR IF NOT ENCLOSING
    103 007420  005705                        TST     R5              ;ANY OTHER ENCLOSING LD'S FOUND SO FAR?
    104 007422  001404                        BEQ     15$             ;BR IF NOT
    105 007424  026767  000001C 170370        CMP     CDBUF+CD$BAS,CDBAS5; SAVE ONE WITH HIGHEST BASE
    106 007432  101404                        BLOS    14$
    107 007434  010405                15$:    MOV     R4,R5           ;THIS IS AN ENCLOSING LOGICAL DISK
    108 007436  016767  000001C 170356        MOV     CDBUF+CD$BAS,CDBAS5
    109 007444  062704  000000G       14$:    ADD     #CD$$SZ,R4      ;POINT TO NEXT MOUNT TABLE ENTRY
    110 007450  020467  000000G               CMP     R4,CSHDVN       ;CHECKED ALL?
    111 007454  103741                        BLO     13$             ;LOOP IF NOT
    112 007456  010501                        MOV     R5,R1           ;DID WE FIND AN ENCLOSING LOGICAL DISK?
    113 007460  001404                        BEQ     16$             ;BR IF NOT
    114 007462  010500                        MOV     R5,R0           ;WE FOUND AN OUTER LD
```

```
    115 007464  004767  000000G                 CALL    CDGET           ;GET OUTER LD DATA INTO CDBUF
    116 007470  000722                          BR      12$             ;KEEP SCANNING OUTWARD
    117 007472  012605                  16$:    MOV     (SP)+,R5        ;GET POINTER TO OUTER-MOST LOGICAL DISK
    118 007474  012703  000000G                 MOV     #PRTBUF,R3      ;GET POINTER TO TEXT BUFFER FOR PRTFNM
    119 007500  010500                  17$:    MOV     R5,R0           ;Get address of mount entry
    120 007502  004767  000000G                 CALL    CDGET           ;Get mount entry into CDBUF
    121 007506  012700  000000C                 MOV     #CDBUF+CD$NAM,R0;GET POINTER TO FILE NAME
    122 007512  004767  000000G                 CALL    PRTFNM          ;FORMAT THE FILE NAME
    123 007516  112723  000072                  MOVB    #':,(R3)+       ;PUT COLON AFTER FILE NAME
    124 007522  012605                          MOV     (SP)+,R5        ;IS THERE ANOTHER NAME?
    125 007524  001365                          BNE     17$             ;BR IF YES
    126 007526  005303                          DEC     R3              ;POINT BACK TO LAST COLON
    127 007530  020327  0000060         18$:    CMP     R3,#PRTBUF+6.   ;TAB UP TO COLUMN 6
    128 007534  103003                          BHIS    19$
    129 007536  112723  000040                  MOVB    #40,(R3)+       ;FILL WITH TRAILING SPACES
    130 007542  000772                          BR      18$
    131 007544  112713  000200          19$:    MOVB    #200,(R3)       ;PUT IN END OF STRING MARKER
    132 007550                                  .PRINT  #PRTBUF         ;PRINT THE LOGICAL DISK SPECIFICATION
    133 007556  012601                          MOV     (SP)+,R1        ;GET BACK MOUNT TABLE INDEX
    134 007560  010100                          MOV     R1,R0           ;Get address of mount entry
    135 007562  004767  000000G                 CALL    CDGET           ;Move entry into CDBUF
    136                                 ;
    137                                 ;  Print numbers of jobs that have this device mounted
    138                                 ;
    139 007566  012705  000001          4$:     MOV     #1,R5           ;START WITH JOB # 1
    140 007572  012703  000001                  MOV     #1,R3           ;GET MOUNT FLAG FOR JOB # 1
    141 007576  012704  000000C                 MOV     #CDBUF+CD$JOB,R4;POINT TO TABLE WITH JOB BIT FLAGS
    142 007602  130314                  7$:     BITB    R3,(R4)         ;IS DEVICE MOUNTED BY THIS JOB?
    143 007604  001415                          BEQ     5$              ;BR IF NOT
    144 007606                                  .TTYOUT #40             ;PRINT A SPACE
    145 007616  020527  000012                  CMP     R5,#10.         ;IS THIS A 2 DIGIT JOB NUMBER?
    146 007622  103004                          BHIS    10$             ;BR IF YES
    147 007624                                  .TTYOUT #40             ;PRINT EXTRA SPACE FOR COLUMN ALIGNMENT
    148 007634  004767  000000G         10$:    CALL    PRTDEC          ;PRINT JOB NUMBER
    149 007640  106303                  5$:     ASLB    R3              ;SHIFT OVER MOUNT FLAG
    150 007642  103002                          BCC     6$              ;BR IF DID NOT SHIFT OUT OF BYTE
    151 007644  005204                          INC     R4              ;POINT TO NEXT BYTE
    152 007646  006103                          ROL     R3              ;RESET FLAG BIT
    153 007650  005205                  6$:     INC     R5              ;ADVANCE JOB NUMBER
    154 007652  020527  000000G                 CMP     R5,#NLINES      ;CHECKED ALL JOBS?
    155 007656  101751                          BLOS    7$              ;LOOP IF MORE TO CHECK
    156                                 ;
    157                                 ;  Finished with this mount entry
    158                                 ;
    159 007660                                  .PRINT  #CRLF           ;END LINE
    160 007666  000167  177354          8$:     JMP     1$              ;Go back and print info for next device
    161                                 ;
    162                                 ;  Finished checking all entries in mount table
    163                                 ;
    164 007672  005767  000000G         2$:     TST     BLK0            ;WERE THERE ANY MOUNTED DEVICES?
    165 007676  001003                          BNE     3$              ;BR IF YES
    166 007700                                  .PRINT  #NONEMS         ;PRINT "(NONE)"
    167                                 ;
    168                                 ;  Finished
    169                                 ;
    170 007706  012605                  3$:     MOV     (SP)+,R5
    171 007710  012604                          MOV     (SP)+,R4
```

.	MOUNTS

```
    172 007712  012603                         MOV     (SP)+,R3
    173 007714  012602                         MOV     (SP)+,R2
    174 007716  012601                         MOV     (SP)+,R1
    175 007720  000207                         RETURN
```

```
                                        .SBTTL  .       MODEM
                                ;-----------------------------------------------------------------------
                                ;  Display modem control parameters
                                ;
007722                          SHOMDM: .PRINT  #PHNTXT         ;Start with PHONE
007730  012700  000060                  MOV     #'0,R0          ;Assume phone = 0
007734  105767  0000000G                TSTB    VUSPHN          ;Is phone set?
007740  001401                          BEQ     1$              ;Br if not
007742  005200                          INC     R0              ;PHONE=1
007744                          1$:     .TTYOUT R0
007750                                  .PRINT  #MDMTXT         ;Print modem header
007756                                  .PRINT  #ONTTXT         ;ONTIM
007764  016705  0000000G                MOV     VONTM,R5
007770  004767  0000000G                CALL    PRTDEC
007774                                  .PRINT  #OFTTXT         ;OFFTIM
010002  016705  0000000G                MOV     VOFFTM,R5
010006  004767  0000000G                CALL    PRTDEC
010012                                  .PRINT  #TMITXT         ;TIMIN
010020  016705  0000000G                MOV     VTMIN,R5
010024  004767  0000000G                CALL    PRTDEC
010030                                  .PRINT  #TMLTXT         ;TIMLOC
010036  016705  0000000G                MOV     VTMLOC,R5
010042  004767  0000000G                CALL    PRTDEC
010046                                  .PRINT  #TMOTXT         ;TIMOUT
010054  016705  0000000G                MOV     VTMOUT,R5
010060  004767  0000000G                CALL    PRTDEC
010064                                  .PRINT  #CRLF
010072  000207                          RETURN
```

```
                                        .SBTTL  .       DATE
                                ;-------------------------------------------------------------------------
                                ;  Display current date.
                                ;
010074                          SHODAT: .DATE                   ;GET CURRENT DATE
010102  005700                          TST     R0              ;IS SYSTEM DATE KNOWN?
010104  001406                          BEQ     1$              ;BR IF NO DATE ENTERED
010106  004767  0000006                 CALL    PRTDAT          ;DISPLAY DATE
010112                                  .PRINT  #CRLF
010120  000403                          BR      9$
                                ;  Date is unknown.
010122                          1$:     .PRINT  #NODAT          ;NO DATE
010130  000207                  9$:     RETURN

                                        .SBTTL  .       TIME
                                ;-------------------------------------------------------------------------
                                ;  Display current time of day.
                                ;
010132  004767  0000006         SHOTIM: CALL    PRTTOD          ;DISPLAY CURRENT TIME
010136                                  .PRINT  #CRLF
010144  000207                          RETURN

                                        .SBTTL  .       VERSION
                                ;-------------------------------------------------------------------------
                                ;  Display the TSX-Plus system version number.
                                ;
010146                          SHOVER: .PRINT  #TSXVER         ;"TSX-Plus Version="
010154  012703  0000006                 MOV     #TSXVRS,R3      ;Get system version number
010160  005002                          CLR     R2              ;Clear high-order for divide
010162  071227  000144                  DIV     #100.,R2        ;Divide R2-R3 by 100
010166  010205                          MOV     R2,R5           ;Get major version number
010170  004767  0000006                 CALL    PRTDEC          ;Print major version number
010174                                  .TTYOUT #'.             ;Print decimal point
010204  010305                          MOV     R3,R5           ;Get fractional version number
010206  004767  0000006                 CALL    PRTDC2          ;Print fractional version number
010212                                  .PRINT  #CRLF           ;Terminate print line
010220  000207                          RETURN

                                        .SBTTL  .       USE
                                ;-------------------------------------------------------------------------
                                ;  Display computer usage information for current job.
                                ;
010222  004767  0000006         SHOUSE: CALL    PRTTIM          ;PRINT CONNECT AND CPU TIME FOR JOB
010226  000207                          RETURN
```

```
                                        .SBTTL  .       INSTALL
                                ;-----------------------------------------------------------------------
                                ;  Show information about installed programs
                                ;
010230  010546                  SHOINS: MOV     R5,-(SP)
010232  004767  000000G                 CALL    CKSYPV          ;Require SYSPRV privilege to do this
                                ;
                                ;  Print title lines
                                ;
010236                                  .PRINT  #TM$IN1
010244                                  .PRINT  #TM$IN2
                                ;
                                ;  Begin loop to print information about each installed program
                                ;
010252  016705  000000G                 MOV     INSTBL,R5       ;Point to 1st install table entry
010256  010567  000000G         1$:     MOV     R5,INGADR       ;Set address of entry to get
010262  012700  000000G                 MOV     #INGEMT,R0      ;Get EMT arg block
010266  104375                          EMT     375             ;Get the entry to IIBUF
010270  005767  000001C                 TST     IIBUF+II$NAM    ;Is this entry in use?
010274  001402                          BEQ     2$              ;Br if not
010276  004767  000016                  CALL    INSPRT          ;Print info about this program
010302  062705  000000G         2$:     ADD     #II$$SZ,R5      ;Point to next install table entry
010306  020567  000000G                 CMP     R5,INSTBN       ;Done all entries?
010312  103761                          BLO     1$              ;Loop if not
                                ;
                                ;  Finished
                                ;
010314  012605                  MOV     (SP)+,R5
010316  000207                  RETURN
```

```
                                 ;-----------------------------------------------------------------------
                                 ;  Print information about the program whose install entry is in IIBUF.
                                 ;
                                 ;  Inputs:
                                 ;   IIBUF contains install entry for program.
                                 ;
010320  010146                   INSPRT: MOV     R1,-(SP)
010322  010246                           MOV     R2,-(SP)
010324  010346                           MOV     R3,-(SP)
010326  010446                           MOV     R4,-(SP)
                                 ;
                                 ;  Print the file spec for the program.
                                 ;
010330  012704  000000C                  MOV     #IIBUF+II$NAM,R4;Point to file spec to be converted
010334  012703  000000G                  MOV     #BLKO,R3        ;Point to result area
010340  004767  000000G                  CALL    EDTFIL          ;Convert file spec to asciz form
010344  112723  000040           1$:     MOVB    #' ,(R3)+       ;Store space following file spec
010350  020327  000021G                  CMP     R3,#BLKO+17.    ;Filled out to flag area?
010354  103773                           BLO     1$              ;Loop if not
010356  112713  000200                   MOVB    #200,(R3)       ;Store null at end of name
010362                                   .PRINT  #BLKO           ;Print the name
                                 ;
                                 ;  Print names of attribute flags
                                 ;
010370  012704  177756                   MOV     #-18.,R4        ;No attributes printed yet
010374  012703  010620'                  MOV     #INSANT,R3      ;Point to attribute name table
010400  032367  000001C          2$:     BIT     (R3)+,II$FLG+IIBUF ;Is this attribute set?
010404  001422                           BEQ     4$              ;Br if not
010406  005704                           TST     R4              ;Is this the first attribute for program?
010410  002406                           BLT     8$              ;Br if yes
010412                                   .TTYOUT #'/             ;Print slash
010422  005204                           INC     R4              ;Count column used by slash
010424  000402                           BR      3$
010426  012704  000022           8$:     MOV     #18.,R4         ;Initialize column number
010432  011301                   3$:     MOV     (R3),R1         ;Get address of name string
010434                                   .PRINT  R1              ;Print attribute name
010440  122127  000200           10$:    CMPB    (R1)+,#200      ;End of keyword?
010444  001402                           BEQ     4$              ;Br if yes
010446  005204                           INC     R4              ;Count columns
010450  000773                           BR      10$
010452  005723                   4$:     TST     (R3)+           ;Skip name pointer
010454  005713                           TST     (R3)            ;Any more attributes to check?
010456  001350                           BNE     2$              ;Loop if yes
                                 ;
                                 ;  See if program has any associated privileges
                                 ;
010460  012702  000000C                  MOV     #II$PRV+IIBUF,R2;Point to privilege flag info
010464  012703  000000C                  MOV     #II$NPV+IIBUF,R3;Point to negative privilege flags
010470  012700  000000G                  MOV     #PVNPW,R0       ;Get # privilege words
010474  005722                   12$:    TST     (R2)+           ;Any privilege flags set?
010476  001004                           BNE     13$             ;Br if yes
010500  005723                           TST     (R3)+           ;Any negative privilege flags set?
010502  001002                           BNE     13$             ;Br if yes
010504  077005                           SOB     R0,12$          ;Check all privilege words
010506  000434                           BR      11$             ;No privileges specified for program
                                 ;
                                 ;  Print information about special privileges associated with program
```

```
                                                ;
010510                                  13$:    .PRINT  #TM$PVL         ;Print "/PRIV=("
010516  005704                                  TST     R4              ;Any attributes printed?
010520  002401                                  BLT     14$             ;Br if not
010522  005404                                  NEG     R4              ;Make R4 negative to suppress leading comma
010524  162704  000007                14$:    SUB     #7.,R4          ;Increase column count
010530  012702  000000C               15$:    MOV     #II$PRV+IIBUF,R2;Point to words with privilege flags
010534  012703  000001                        MOV     #+1,R3          ;Select positive flags this time
010540  012700  000022                        MOV     #18.,R0         ;Wrap-around to column 18
010544  004767  0000000                       CALL    PRVLST          ;Print selected privileges
010550  012702  000000C                       MOV     #II$NPV+IIBUF,R2;Point to words with deselected priv flags
010554  012703  177777                        MOV     #-1,R3          ;Select negative flags
010560  012700  000022                        MOV     #18.,R0         ;Set wrap-around column
010564  004767  0000000                       CALL    PRVLST          ;Print deselected privileges
010570                                        .TTYOUT #')             ;Terminate privilege list
                                        ;
                                        ;  Terminate the print line
                                        ;
010600                                  11$:    .PRINT  #CRLF           ;End line
                                        ;
                                        ;  Finished
                                        ;
010606  012604                                  MOV     (SP)+,R4
010610  012603                                  MOV     (SP)+,R3
010612  012602                                  MOV     (SP)+,R2
010614  012601                                  MOV     (SP)+,R1
010616  000207                                  RETURN
                                        ;
                                        ;  Table of install program attributes and names
                                        ;
010620  0000000 010722'                 INSANT: .WORD   AF$SCA,1$
010624  0000000 010735'                         .WORD   AF$NOW,2$
010630  0000000 010744'                         .WORD   AF$HIE,3$
010634  0000000 010751'                         .WORD   AF$NOI,4$
010640  0000000 010770'                         .WORD   AF$IOP,5$
010644  0000000 010777'                         .WORD   AF$MEM,6$
010650  0000000 011007'                         .WORD   AF$PLK,7$
010654  0000000 011014'                         .WORD   AF$DBG,8$
010660  0000000 011022'                         .WORD   AF$BYA,9$
010664  0000000 011031'                         .WORD   AF$TPO,10$
010670  0000000 011045'                         .WORD   AF$DUP,11$
010674  0000000 011051'                         .WORD   AF$IND,12$
010700  0000000 011055'                         .WORD   AF$UCL,13$
010704  0000000 011064'                         .WORD   AF$SET,14$
010710  0000000 011072'                         .WORD   AF$CCA,15$
010714  0000000 011077'                         .WORD   AF$NPW,16$
010720  000000                                  .WORD   0
                                        ;
010722      123     111     116     1$:     .ASCII  /SINGLECHAR/<200>
010725      107     114     105
010730      103     110     101
010733      122     200
010735      116     117     127     2$:     .ASCII  /NOWAIT/<200>
010740      101     111     124
010743      200
010744      110     111     107     3$:     .ASCII  /HIGH/<200>
010747      110     200
```

```
    109 010751      116     117     116  4$:     .ASCII  /NONINTERACTIVE/<200>
        010754      111     116     124
        010757      105     122     101
        010762      103     124     111
        010765      126     105     200
    110 010770      111     117     120  5$:     .ASCII  /IOPAGE/<200>
        010773      101     107     105
        010776      200
    111 010777      115     105     115  6$:     .ASCII  /MEMLOCK/<200>
        011002      114     117     103
        011005      113     200
    112 011007      114     117     103  7$:     .ASCII  /LOCK/<200>
        011012      113     200
    113 011014      104     105     102  8$:     .ASCII  /DEBUG/<200>
        011017      125     107     200
    114 011022      102     131     120  9$:     .ASCII  /BYPASN/<200>
        011025      101     123     116
        011030      200
    115 011031      124     122     101  10$:    .ASCII  /TRANSPARENT/<200>
        011034      116     123     120
        011037      101     122     105
        011042      116     124     200
    116 011045      104     125     120  11$:    .ASCII  /DUP/<200>
        011050      200
    117 011051      111     116     104  12$:    .ASCII  /IND/<200>
        011054      200
    118 011055      124     123     130  13$:    .ASCII  /TSXUCL/<200>
        011060      125     103     114
        011063      200
    119 011064      123     105     124  14$:    .ASCII  /SETUP/<200>
        011067      125     120     200
    120 011072      123     103     103  15$:    .ASCII  /SCCA/<200>
        011075      101     200
    121 011077      116     117     127  16$:    .ASCII  /NOWINDOW/<200>
        011102      111     116     104
        011105      117     127     200
    122                                  .EVEN
```

```
                                        .SBTTL  .       REGIONS
                                ;-------------------------------------------------------------------------------
                                ;  Show named regions
                                ;
 011110  010246                 SHOREG: MOV     R2,-(SP)
 011112  010446                         MOV     R4,-(SP)
 011114  005002                         CLR     R2              ;Clear count of # regions shown
                                ;
                                ;  First show information about all private regions
                                ;
 011116  012704  000000G                MOV     #RCBBAS,R4      ;Point to first private Region Control Blk
 011122  004767  000106         1$:     CALL    RGNDSP          ;Display info about this region
 011126  062704  000000G                ADD     #RC$$SZ,R4      ;Point to next RCB
 011132  020427  000000G                CMP     R4,#RCBEND      ;Checked all RCB's?
 011136  103771                         BLO     1$              ;Br if not
                                ;
                                ;  Now show information about all shared regions
                                ;
 011140  016704  000000G                MOV     SHRRCB,R4       ;Point to 1st shared RCB
 011144  020467  000000G        2$:     CMP     R4,SHRRCN       ;Checked all shared RCB's?
 011150  103021                         BHIS    3$              ;Br if yes
 011152  010467  000000G                MOV     R4,PEKADR       ;Set address of RCB
 011156  012767  000000G 000000G       MOV     #RC$$SZ,PEKSIZ  ;Get # bytes to fetch
 011164  012700  000000G                MOV     #PEKEMT,R0      ;Point to EMT arg block
 011170  104375                         EMT     375             ;Move RCB to BLKO buffer
 011172  012704  000000G                MOV     #BLKO,R4        ;Point to RCB in our buffer
 011176  004767  000032                 CALL    RGNDSP          ;Display info about the region
 011202  016704  000000G                MOV     PEKADR,R4       ;Recover real RCB address
 011206  062704  000000G                ADD     #RC$$SZ,R4      ;Point to next RCB
 011212  000754                         BR      2$
                                ;
                                ;  If there were no named regions, print a message.
                                ;
 011214  005702                 3$:     TST     R2              ;Were there any named regions?
 011216  001003                         BNE     9$              ;Br if yes
 011220                                 .PRINT  #TM$NNR         ;No named regions
                                ;
                                ;  Finished
                                ;
 011226  012604                 9$:     MOV     (SP)+,R4
 011230  012602                         MOV     (SP)+,R2
 011232  000207                         RETURN
```

```
                                        ;-------------------------------------------------------------------------------
                                        ;  Display a line of information about a named region.
                                        ;
                                        ;  Inputs:
                                        ;   R4 = Pointer to Region Control Block.
                                        ;   R2 = Count of number of active regions so far.
                                        ;
                                        ;  Outputs:
                                        ;   R2 = Incremented if this is an active region.
                                        ;
011234                                  RGNDSP:
                                        ;
                                        ;  See if this RCB is for an active, named region
                                        ;
011234  032764  000000G 000000G         1$:     BIT     #RC$USE,RC$FLG(R4) ;Is this an active RCB?
011242  001410                                  BEQ     1$                 ;Br if not
011244  032764  000000G 000000G                 BIT     #RC$LCG,RC$FLG(R4) ;Is this a local copy of a global RCB?
011252  001004                                  BNE     1$                 ;Br if yes -- We will list global RCB
011254  032764  000000G 000000G                 BIT     #RC$GBL,RC$FLG(R4) ;Is this a named region?
011262  001001                                  BNE     2$                 ;Br if yes
011264  000207                                  RETURN
                                        ;
                                        ;  This is an RCB for an active named region.
                                        ;
011266  010146                         2$:     MOV     R1,-(SP)
011270  010346                                 MOV     R3,-(SP)
011272  010546                                 MOV     R5,-(SP)
                                        ;
                                        ;  See if this is the 1st region
                                        ;
011274  005702                                 TST     R2                 ;Is this the first region?
011276  001006                                 BNE     3$                 ;Br if not
011300                                          .PRINT  #TM$RD1            ;Display title line 1
011306                                          .PRINT  #TM$RD2            ;Display title line 2
011314  005202                         3$:     INC     R2                 ;Count another region displayed
                                        ;
                                        ;  Display name of the region
                                        ;
011316  016400  000000G                         MOV     RC$NAM(R4),R0      ;Get 1st 3 chars of name
011322  004767  000000G                         CALL    PRTR50             ;Print them
011326  016400  000002G                         MOV     RC$NAM+2(R4),R0    ;Get 2nd 3 chars of name
011332  004767  000000G                         CALL    PRTR50             ;Print them
                                        ;
                                        ;  Display size of region
                                        ;
011336  016401  000000G                         MOV     RC$LEN(R4),R1      ;Get # 64-byte blocks allocated for region
011342  005000                                  CLR     R0                 ;Clear high-order for divide
011344  071027  000020                         DIV     #16.,R0            ;Conver to # Kb
011350  010005                                  MOV     R0,R5              ;Get # whole K
011352  012703  000006                         MOV     #6.,R3             ;Print in 6 digit field
011356  004767  000000G                         CALL    PRTFIX             ;Print # whole Kb
011362  116105  011630'                        MOVB    FRAC64(R1),R5      ;Convert to decimal Kb fraction
011366                                          .TTYOUT #'.                ;Print decimal point
011376  004767  000000G                         CALL    PRTDEC             ;Print decimal digit
                                        ;
                                        ;  Display type of region
                                        ;
```

```
      58 011402                                         .PRINT  #SPACE2          ;Print 2 spaces
      59 011410  032764  000000G 000000G               BIT     #RC$PVT,RC$FLG(R4) ;Is this a private or shared region?
      60 011416  001407                                 BEQ     4$               ;Br if shared
      61 011420                                         .PRINT  #TM$LCL          ;Say region is local to job
      62 011426                                         .PRINT  #SPACE1          ;Print extra space
      63 011434  000403                                 BR      5$
      64 011436                                 4$:     .PRINT  #TM$GBL          ;Say region is global
      65                                                ;
      66                                                ;  Print number of job that created region
      67                                                ;
      68 011444  116405  000000G                5$:     MOVB    RC$OWN(R4),R5    ;Get # of job that created region
      69 011450  006205                                 ASR     R5               ;Convert to #
      70 011452  012703  000005                         MOV     #5.,R3           ;Print 5 digit field
      71 011456  004767  000000G                       CALL    PRTFIX           ;Print job #
      72                                                ;
      73                                                ;  Print attachment (use) count
      74                                                ;
      75 011462  116405  000000G                       MOVB    RC$CNT(R4),R5    ;Get attachment count
      76 011466  012703  000005                         MOV     #5.,R3           ;Print in 5 digit field
      77 011472  004767  000000G                       CALL    PRTFIX           ;Print use count
      78                                                ;
      79                                                ;  Show if shared
      80                                                ;
      81 011476  012703  000004                         MOV     #4.,R3           ;Print 4 spaces
      82 011502  004767  000000G                       CALL    PRTSPC
      83 011506  032764  000000G 000000G               BIT     #RC$EXC,RC$FLG(R4) ;Is this region sharable?
      84 011514  001404                                 BEQ     6$               ;Br if yes
      85 011516                                         .PRINT  #NOTXT           ;Print 'No'
      86 011524  000403                                 BR      7$
      87 011526                                 6$:     .PRINT  #YESTXT          ;Print 'Yes'
      88                                                ;
      89                                                ;  Show if AGE is set
      90                                                ;
      91 011534                                 7$:     .PRINT  #SPACE3          ;Print 3 spaces
      92 011542  032764  000000C 000000G               BIT     #RC$AGE!RC$AEP,RC$FLG(R4) ;Is AGE enabled?
      93 011550  001004                                 BNE     8$               ;Br if yes
      94 011552                                         .PRINT  #NOTXT           ;Print 'No'
      95 011560  000403                                 BR      9$
      96 011562                                 8$:     .PRINT  #YESTXT          ;Print 'Yes'
      97                                                ;
      98                                                ;  Show region base 64-byte block #
      99                                                ;
     100 011570                                 9$:     .PRINT  #SPACE2          ;Print 2 spaces
     101 011576  010246                                 MOV     R2,-(SP)
     102 011600  016402  000000G                        MOV     RC$BAS(R4),R2    ;Get base block #
     103 011604  004767  000000G                        CALL    OCTPRT           ;Display it
     104 011610  012602                                 MOV     (SP)+,R2
     105                                                ;
     106                                                ;  Terminate print line
     107                                                ;
     108 011612                                         .PRINT  #CRLF
     109                                                ;
     110                                                ;  Finished
     111                                                ;
     112 011620  012605                                 MOV     (SP)+,R5
     113 011622  012603                                 MOV     (SP)+,R3
     114 011624  012601                                 MOV     (SP)+,R1
```

```
     115 011626  000207                            RETURN
     116                                 ;
     117                                 ;  Values to convert # fractional 64-byte blocks to tenths of Kb
     118                                 ;
     119 011630      000     001     001 FRAC64: .BYTE   0.,1.,1.,2.,2.,3.,4.,4.,5.,6.,6.,7.,7.,8.,9.,9.
         011633      002     002     003
         011636      004     004     005
         011641      006     006     007
         011644      007     010     011
         011647      011
     120                                 .EVEN
```

```
                                        .SBTTL  .       PRIVILEGES
                                ;-----------------------------------------------------------------------
                                ;  Show privileges
                                ;
011650  010246                  SHOPRV: MOV     R2,-(SP)
011652  010346                          MOV     R3,-(SP)
011654  010446                          MOV     R4,-(SP)
                                ;
                                ;  List authorized privileges
                                ;
011656                                  .PRINT  #TM$PVA         ;Authorized privileges
011664  012702  0000000G                MOV     #PRIVA0,R2      ;Point to priv flag vector
011670  012703  000001                  MOV     #+1,R3          ;Show positive privileges only
011674  012704  177750                  MOV     #-24.,R4        ;Start at column 24
011700  012700  000030                  MOV     #24.,R0         ;Wrap around to col 24
011704  004767  0000000G                CALL    PRVLST          ;List privileges
011710                                  .PRINT  #CRLF           ;Terminate last line
011716                                  .PRINT  #CRLF           ;Put in a blank line
                                ;
                                ;  List current privileges
                                ;
011724                                  .PRINT  #TM$PVC         ;Current privileges
011732  012702  0000000G                MOV     #PRIVC0,R2      ;Current privilege flags
011736  012704  177750                  MOV     #-24.,R4        ;Start at column 24
011742  012700  000030                  MOV     #24.,R0         ;Wrap around to col 24
011746  004767  0000000G                CALL    PRVLST          ;List privileges
011752                                  .PRINT  #CRLF
                                ;
                                ;  Finished
                                ;
011760  012604                          MOV     (SP)+,R4
011762  012603                          MOV     (SP)+,R3
011764  012602                          MOV     (SP)+,R2
011766  000207                          RETURN
```

```
                                        .SBTTL  .       SL
                                ;-----------------------------------------------------------------------------
                                ;  Display single line editor status.
                                ;
011770  010246                 SHOSLE: MOV     R2,-(SP)
011772  010346                         MOV     R3,-(SP)
011774  010446                         MOV     R4,-(SP)
                                ;
                                ;  Print message heading
                                ;
011776                                  .PRINT  #TM$SL1         ;"SL status:  "
                                ;
                                ;  Print ON or OFF
                                ;
012004  032761  000000G 000000G         BIT     #$SLON,LSW7(R1) ;Is SL turned on?
012012  001003                          BNE     1$              ;Br if yes
012014  012702  000000G                 MOV     #TM$OFF,R2      ;Point to OFF message
012020  000402                          BR      2$
012022  012702  000000G         1$:     MOV     #TM$ON,R2       ;Point to ON message
012026                          2$:     .PRINT  R2              ;Print text
                                ;
                                ;  Check for KED mode
                                ;
012032  012704  000000G                 MOV     #$SLKED,R4      ;Get Ked flag
012036  012703  000000G                 MOV     #TM$KED,R3
012042  004767  000046                  CALL    SLCKFL
                                ;
                                ;  Check for TTY mode
                                ;
012046  012704  000000G                 MOV     #$SLTTY,R4      ;Get flag
012052  012703  000000G                 MOV     #TM$TTY,R3      ;Point to text string
012056  004767  000032                  CALL    SLCKFL
                                ;
                                ;  Check for SUBSTITUTE mode
                                ;
012062  012704  000000G                 MOV     #$SLLET,R4      ;Get flag
012066  012703  000000G                 MOV     #TM$SUB,R3      ;Point to text string
012072  004767  000016                  CALL    SLCKFL
                                ;
                                ;  Terminate the print line
                                ;
012076                                  .PRINT  #CRLF           ;Terminate print line
                                ;
                                ;  Finished
                                ;
012104  012604                          MOV     (SP)+,R4
012106  012603                          MOV     (SP)+,R3
012110  012602                          MOV     (SP)+,R2
012112  000207                          RETURN
```

```
                                ;*****************************************************************************
                                ;  Check of a SL option flag is set in LSW7, and if it is print that the
                                ;  option is on; otherwise, print that the option is off.
                                ;
                                ;  Inputs:
                                ;   R1 = Job index number
                                ;   R3 = Pointer to text string for option name
                                ;   R4 = Flag bit to be tested in LSW7
                                ;
012114                          SLCKFL:
                                ;
                                ;  Print a leading comma
                                ;
012114                                  .TTYOUT #54             ;Print comma
                                ;
                                ;  If option flag is not set, print "NO"
                                ;
012124  030461  0000000                 BIT     R4,LSW7(R1)     ;Is the option flag set?
012130  001003                          BNE     1$              ;Br if yes
012132                                  .PRINT  #TM$NO          ;Print NO
                                ;
                                ;  Print the name of the option
                                ;
012140                          1$:     .PRINT  R3              ;Print the option name
                                ;
                                ;  Finished
                                ;
012144  000207                  RETURN
```

```
                                                .SBTTL  .       RUN-TIMES
                                        ;-------------------------------------------------------------------------------
                                        ;  Show run-times
                                        ;
012146  010146                          SHOSRT: MOV     R1,-(SP)
012150  012701  000000G                         MOV     #RDB,R1         ;POINT TO FIRST RUN-TIME DESCRIPTOR BLOCK
012154  005005                                  CLR     R5              ;Count run-times in R5
012156  020127  000000G                 1$:     CMP     R1,#RDBEND      ;ANY RUN-TIMES?
012162  103023                                  BHIS    2$              ;BR IF FINISHED
012164  026127  000000G 000000G                 CMP     RT$DEV(R1),#DMYDEV ;Is this dummy entry for patching?
012172  001414                                  BEQ     4$              ;Br if yes
012174  016100  000000G                         MOV     RT$NAM(R1),R0   ;GET 1ST 3 CHARS OF RUN-TIME NAME
012200  005205                                  INC     R5              ;Count another run-time
012202  004767  000000G                         CALL    PRTR50          ;PRINT THEM
012206  016100  000002G                         MOV     RT$NAM+2(R1),R0 ;GET 2ND 3 CHARS OF NAME
012212  004767  000000G                         CALL    PRTR50          ;PRINT THEM
012216                                          .PRINT  #CRLF           ;END LINE
012224  062701  000000G                 4$:     ADD     #RT$$SZ,R1      ;POINT TO NEXT DESCRIPTOR BLOCK
012230  000752                                  BR      1$
012232  005705                          2$:     TST     R5              ;WERE THERE ANY RUN-TIMES?
012234  001003                                  BNE     3$              ;BR IF YES
012236                                          .PRINT  #NONEMS         ;PRINT NONE
012244  012601                          3$:     MOV     (SP)+,R1
012246  000207                                  RETURN
```

```
                                        .SBTTL  .       SPOOL
                                ;-------------------------------------------------------------------------
                                ;  SHOW SPOOL
                                ;  List the names of the spooled devices.
                                ;
012250  010246                  SHOSPL: MOV     R2,-(SP)
                                ;
                                ;  See if there are any spooled devices
                                ;
012252  105767  000000G                 TSTB    NSPLDV          ;Are there any spooled devices?
012256  001405                          BEQ     3$              ;Br if not
012260  012702  000000G                 MOV     #SDCB,R2        ;Point to first spooled device control block
012264  020227  000000G                 CMP     R2,#SDCBND      ;Are there any spooled devices?
012270  103404                          BLO     1$              ;Br if there are spooled devices
012272                          3$:     .PRINT  #TM$NSD         ;There are no spooled devices
012300  000425                          BR      9$
                                ;
                                ;  There are spooled devices, print their names
                                ;
012302                          1$:     .PRINT  #TM$SDN         ;Print heading
012310  016200  000000G         2$:     MOV     SDNAME(R2),R0   ;Get RAD50 name of spooled device
012314  020027  000000G                 CMP     R0,#DMYDEV      ;Uninstalled device?
012320  001405                          BEQ     4$              ;Br if yes
012322  004767  000000G                 CALL    PRTR50          ;Print the name
012326                                  .PRINT  #SPACE2         ;Print 2 spaces
012334  062702  000000G         4$:     ADD     #SDCBSZ,R2      ;Point to next SDCB
012340  020227  000000G                 CMP     R2,#SDCBND      ;Are there more?
012344  103761                          BLO     2$              ;Br if yes
012346                                  .PRINT  #CRLF           ;Terminate the line
                                ;
                                ;  Finished
                                ;
012354  012602                  9$:     MOV     (SP)+,R2
012356  000207                          RETURN
```

```
                                        .SBTTL  SUBSET
                                ;-------------------------------------------------------------------------------
                                ;  SHOW SUBSET
                                ;  Display information about logical disks.
                                ;
012360  010146                  SHOSUB: MOV     R1,-(SP)
012362  010246                          MOV     R2,-(SP)
012364  010346                          MOV     R3,-(SP)
012366  010446                          MOV     R4,-(SP)
012370  010546                          MOV     R5,-(SP)
                                ;
                                ;  First do a SET LD CLEAN to update logical disk information
                                ;
012372  004767  000000G                 CALL    LDCLEN          ;DO SET LD CLEAN
                                ;
                                ;  Make a fast scan and see if any logical disks are mounted
                                ;
012376  012703  000010                  MOV     #8.,R3          ;GET # LOGICAL DISK ENTRIES
012402  012704  000000G                  MOV     #LDNAME,R4      ;POINT TO FILE NAME TABLE
012406  005714                  5$:     TST     (R4)            ;IS THIS DISK ASSIGNED TO A FILE?
012410  001007                          BNE     6$              ;BR IF YES
012412  062704  000010                  ADD     #8.,R4          ;POINT TO NEXT ENTRY IN TABLE
012416  077305                          SOB     R3,5$           ;LOOP IF MORE TO CHECK
012420                                  .PRINT  #NOLDMT         ;NO LOGICAL DISKS MOUNTED
012426  000472                          BR      9$
                                ;
                                ;  Now begin to display logical disk information
                                ;
012430  016702  000000G         6$:     MOV     R50LDO,R2       ;GET "LDO" NAME
012434  005003                          CLR     R3              ;INIT DISK TABLE INDEX
012436  012704  000000G                  MOV     #LDNAME,R4      ;POINT TO FILE NAME TABLE
                                ;
                                ;  See if next logical disk unit is assigned
                                ;
012442  005714                  2$:     TST     (R4)            ;IS DISK ASSIGNED TO A FILE?
012444  001453                          BEQ     1$              ;BR IF NOT
                                ;
                                ;  Print logical disk name
                                ;
012446  010200                          MOV     R2,R0           ;GET DISK NAME
012450  004767  000000G                 CALL    PRTR50          ;PRINT THE DISK NAME
012454                                  .PRINT  #SUBARO         ;" --> "
                                ;
                                ;  Print the file name
                                ;
012462  010346                          MOV     R3,-(SP)        ;SAVE R3
012464  012703  000000G                 MOV     #BLK0,R3        ;EDIT FILE NAME INTO BLK0
012470  004767  000000G                 CALL    EDTFIL          ;EDIT FILE NAME
012474  112723  000133                  MOVB    #133,(R3)+      ;"[" START OF FILE SIZE
012500  112713  000200                  MOVB    #200,(R3)       ;TERMINATE NAME STRING
012504                                  .PRINT  #BLK0           ;PRINT NAME
012512  012603                          MOV     (SP)+,R3
012514  016305  000000G                 MOV     LDSIZE(R3),R5   ;GET FILE SIZE
012520  004767  000000G                 CALL    PRTDEC          ;PRINT IT
012524                                  .TTYOUT #135            ;"]"
                                ;
                                ;  Print "read only" if that is the case
```

```
                                ;
012534  032763  000000G 000000G        BIT     #LD$RON,LDFLAG(R3) ;IS IS MOUNTED READ-ONLY
012542  001403                        BEQ     3$                 ;BR IF NOT
012544                                .PRINT  #RONTXT            ;PRINT "(read only)"
                                ;
                                ;  Print "not available" if that is the case
                                ;
012552  005763  000000G         3$:    TST     LDPDEV(R3)         ;IS FILE CURRENTLY ACTIVE?
012556  001003                        BNE     4$                 ;BR IF YES
012560                                .PRINT  #NOTAVL            ;NOT AVAILABLE
012566                          4$:   .PRINT  #CRLF              ;PRINT THE LINE
                                ;
                                ;  Advance to next logical disk
                                ;
012574  005202                  1$:    INC     R2                 ;ADVANCE LOGICAL DISK NAME
012576  062704  000010                ADD     #8.,R4             ;ADVANCE NAME POINTER
012602  062703  000002                ADD     #2,R3              ;ADVANCE TABLE POINTER
012606  020327  000016                CMP     R3,#14.            ;DONE ALL?
012612  101713                        BLOS    2$                 ;BR IF NOT
                                ;
                                ;  Finished
                                ;
012614  012605                  9$:    MOV     (SP)+,R5
012616  012604                        MOV     (SP)+,R4
012620  012603                        MOV     (SP)+,R3
012622  012602                        MOV     (SP)+,R2
012624  012601                        MOV     (SP)+,R1
012626  000207                        RETURN
```

```
                                        .SBTTL  .       VM
                                ;-------------------------------------------------------
                                ; SHOW CURRENT VM BASE, TOP AND DEVICE SIZE
                                ;
012630  012767  105646  000000G SHOWVM: MOV     #^RVMO,FILNAM   ;Set device name VM:
012636  005067  000002G                 CLR     FILNAM+2        ;No file name
012642                                  .SERR                   ;Trap .LOOKUP errors
012650                                  .LOOKUP #XAREA,#1,#FILNAM ;Get channel to VMO:
012670  103004                          BCC     1$              ;Branch if we got VM
012672                                  .PRINT  #SHVTX4         ;"VM not installed"
012700  000476                          BR      9$              ;Exit if no VM
                                ;
                                ;  Get current base and top
                                ;
012702                          1$:     .SPFUN  #XAREA,#1,#372,#BLKO,#0,#0 ;Request VM base and top
012744  103451                          BCS     8$              ;Ignore command on error
012746                                  .PRINT  #SHVTX1         ;" VM Base="
012754  016705  000000G                 MOV     BLKO,R5         ;Retrieve base
012760  012703  000006                  MOV     #6,R3           ;Six digit display
012764  004767  000000G                 CALL    OCTFIX          ;Display it
012770                                  .PRINT  #SHVTX2         ;" Top="
012776  016705  000002G                 MOV     BLKO+2,R5       ;Retrieve top
013002  004767  000000G                 CALL    OCTFIX          ;Display it
                                ;
                                ;  Get current device size in blocks from handler
                                ;
013006                                  .SPFUN  #XAREA,#1,#373,#BLKO,#0,#0 ;Request VM device size
013050  103407                          BCS     8$              ;Skip size if bad
013052                                  .PRINT  #SHVTX3         ;" Size="
013060  016705  000000G                 MOV     BLKO,R5         ;Retrieve size
013064  004767  000000G                 CALL    PRTDEC          ;Display decimal size in blocks
                                ;
013070                          8$:     .CLOSE  #1              ;Only close if successful open
013076                          9$:     .PRINT  #CRLF           ;Format display
013104                                  .HERR                   ;Give back error trapping
013112  000207                          RETURN
```

```
                                        .SBTTL  SYSTAT (& WHO) command
                                ;-----------------------------------------------------
                                ;  THE WHO COMMAND PRINTS OUT A LIST OF ALL LINE NUMBERS
                                ;  WHICH ARE LOGGED ON.
                                ;
013114  004767  170610          CMDWHO: CALL    SHOJOB          ;SHOW JOB INFORMATION
013120  000167  000000G                  JMP     RDCMD

                                        .SBTTL  USE command
                                ;-----------------------------------------------------------------------
                                ;  Process the USE command.
                                ;
013124  004767  000000G         CMDUSE: CALL    PRTTIM          ;PRINT CONNECT AND CPU TIME FOR JOB
013130  000167  000000G                  JMP     RDCMD
```

```
                                        .SBTTL  PRTUSE -- Print system usage statistics
                                ;-----------------------------------------------------------------------
                                ;  PRTUSE is called to print the system usage statistics.
                                ;
013134  010146                  PRTUSE: MOV     R1,-(SP)
013136  010246                          MOV     R2,-(SP)
013140  010346                          MOV     R3,-(SP)
013142  010446                          MOV     R4,-(SP)
                                ;  Print system up-time.
013144                                  .PRINT  #UPTMMS         ;"UPTIME: "
013152  016704  000000G                 MOV     TMTOTH,R4       ;GET TOTAL UP-TIME (OR TIME SINCE LAST RESET)
013156  016705  000000G                 MOV     TMTOTL,R5
013162  000241                          CLC                     ;DIVIDE TIME VALUE BY 2 TO GET 1/10 SEC UNITS
013164  006004                          ROR     R4              ;SHIFT HIGH-ORDER PART
013166  006005                          ROR     R5              ;AND LOW-ORDER PART
013170  004767  000000G                 CALL    PRTTMD          ;PRINT TIME VALUE
013174                                  .PRINT  #CRLF           ;TERMINATE LINE
                                ;  Print system usage statistics
013202  016767  000000G 000000G         MOV     TMTOTH,DIVSOR   ;SET TOTAL UP-TIME AS DIVISOR FOR PERCENTAGE
013210  016767  000000G 000002G         MOV     TMTOTL,DIVSOR+2
013216  012702  013254'                 MOV     #SUMVEC,R2      ;POINT TO DRIVER VECTOR TABLE
013222  012200                  1$.     MOV     (R2)+,R0        ;GET ADDRESS OF ASCIZ STRING TO PRINT
013224  001406                          BEQ     2$              ;BR IF END OF LIST HIT
013226                                  .PRINT                  ;PRINT TEXT MESSAGE
013230  012201                          MOV     (R2)+,R1        ;GET ADDRESS OF TIME CELL TO PRINT
013232  001773                          BEQ     1$              ;BR IF NONE WITH THIS TEXT
013234  004767  000000G                 CALL    PRTPCT          ;CONVERT TO PERCENTAGE AND PRINT VALUE
013240  000770                          BR      1$
                                ;
                                ;  Finished
                                ;
013242  012604                  2$:     MOV     (SP)+,R4
013244  012603                          MOV     (SP)+,R3
013246  012602                          MOV     (SP)+,R2
013250  012601                          MOV     (SP)+,R1
013252  000207                          RETURN
                                ;
                                ;  Driver vector for system usage printout.
                                ;  First entry of each pair is the address of an asciz string to print.
                                ;  Second entry is address of 32-bit time value to be printed as percentage.
                                ;
013254  000000G 000000G         SUMVEC: .WORD   SUM1,TMUSRH     ;RUN-TIME
013260  000000G 000000G                 .WORD   SUM2,TMIOWH     ;I/O WAIT
013264  000000G 000000G                 .WORD   SUM3,TMSWTH     ;SWAP WAIT
013270  000000G 000000G                 .WORD   SUM4,TMIDLH     ;IDLE TIME
013274  000000G 000000                  .WORD   SUM5,0          ;END OF LINE
013300  000000G 000000G                 .WORD   SUM6,TMIOH      ;USER I/O TIME
013304  000000G 000000G                 .WORD   SUM7,TMSWPH     ;SWAP TIME
013310  000000G 000000                  .WORD   SUM5,0          ;END OF LINE
013314  000000G 000000                  .WORD   SUM5,0
013320  000000                          .WORD   0               ;END OF LIST
```

```
                                        .SBTTL  MEMORY command
                                ;-------------------------------------------------------------------------
                                ;  The MEMORY command is used to set or display the maximum memory limit
                                ;  for the current job.
                                ;
013322  004767  000000G        CMDMEM: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
013326  111300                  MOVB    (R3),R0         ;WAS A MEMORY LIMIT SPECIFIED WITH COMMAND?
013330  001004                  BNE     SETMEM          ;BR IF YES
                                ;
                                ;  Display current memory values
                                ;
013332  004767  000124          CALL    DSPMEM          ;DISPLAY JOB MEMORY LIMITS
013336  000167  000000G         JMP     RDCMD           ;GO GET NEXT COMMAND
                                ;
                                ;  Set a new memory limit for the job.
                                ;
013342  105767  000000G        SETMEM: TSTB    VSWPFL          ;IS SWAPPING ALLOWED?
013346  001005                  BNE     4$              ;BR IF YES
013350                          .PRINT  #NSWPMS         ;CAN'T CHANGE MEMORY SIZE OF NON-SWAP SYSTEM
013356  000167  000000G         JMP     RDCMD
013362  004767  000000G        4$:     CALL    ACRDEC          ;ACCRUE THE VALUE
013366  120027  000113          CMPB    R0,#'K          ;DID HE SPECIFY K-SOMETHING?
013372  001001                  BNE     1$              ;BR IF NOT
013374  005203                  INC     R3              ;SKIP "K"
013376  121327  000127          1$:     CMPB    (R3),#'W        ;WAS IT "KW"?
013402  001001                  BNE     2$              ;BR IF NOT
013404  006301                  ASL     R1              ;DOUBLE MEMORY VALUE
                                ;  Compare request with max limit.
013406  020167  000000G        2$:     CMP     R1,MXJMEM       ;IS REQUEST LARGET THAN MAX ALLOWED?
013412  101413                  BLOS    3$              ;BR IF NOT
013414  016701  000000G         MOV     MXJMEM,R1       ;SET TO MAX ALLOWED
013420  010105                  MOV     R1,R5
013422                          .PRINT  #MAXMTX         ;DISPLAY MAX ALLOWED
013430  004767  000000G         CALL    PRTDEC
013434                          .PRINT  #KBTX
013442  072127  000012          3$:     ASH     #10.,R1         ;CONVERT # KB TO ADDRESS
013446  001002                  BNE     5$              ;BR IF DIDN'T OVERFLOW 64KB
013450  012701  177774          MOV     #177774,R1      ;SET TO 64KB
013454  010167  000000G         5$:     MOV     R1,MAXMEM       ;SET AS MAX ADDRESS FOR JOB
013460                          .EXIT                   ;EXIT TO ACTUALLY DO THE MEMORY SIZE CHANGE

                                ;-------------------------------------------------------------------------
                                ;  Display information about job memory limits.
                                ;
013462  010546                 DSPMEM: MOV     R5,-(SP)
013464                          .PRINT  #CURMTX         ;CURRENT MEMORY =
013472  016705  000000G         MOV     MAXMEM,R5       ;GET CURRENT HIGH-MEMORY LIMIT FOR JOB
013476  020527  177770          CMP     R5,#177770      ;64KB?
013502  103403                  BLO     1$              ;BR IF NOT
013504  012705  000100          MOV     #64.,R5         ;DO THIS TO AVOID OVERFLOW IN CONVERSION
013510  000403                  BR      2$
013512  000305                  1$:     SWAB    R5              ;CONVERT TO # KB
013514  072527  177776          ASH     #-2,R5
013520  004767  000000G         2$:     CALL    PRTDEC          ;DISPLAY THE VALUE
013524                          .PRINT  #KBTX           ;PRINT "KB"
013532                          .PRINT  #MAXMTX         ;MAX MEMORY =
013540  016705  000000G         MOV     MXJMEM,R5       ;MAX SIZE ALLOWED
```

```
     58 013544  004767  000000G                  CALL    PRTDEC          ;DISPLAY THE VALUE
     59 013550                                   .PRINT  #KBTX           ;"KB"
     60 013556  012605                           MOV     (SP)+,R5
     61 013560  000207                           RETURN
```

```
      1                                 ;-------------------------------------------------------------------------------
      2                                 ;  PRTKB is called to convert a value from # of 256-word memory pages
      3                                 ;  to # k-bytes and print the value followed by "Kb<cr><lf>"
      4                                 ;
      5 013562  010546                  PRTKB:  MOV     R5,-(SP)
      6 013564  006205                          ASR     R5              ;CONVERT # PAGES TO # KB
      7 013566  004767  0000006                 CALL    PRTDEC          ;PRINT THE VALUE
      8 013572                                  .PRINT  #KBTX           ;PRINT "KB<CR><LF>"
      9 013600  012605                          MOV     (SP)+,R5
     10 013602  000207                          RETURN
     11         000001                          .END
Errors detected:  0

*** Assembler statistics


Work  file  reads:  0
Work  file writes:  0
Size of work file:  12332 Words  ( 49 Pages)
Size of core pool:  18176 Words  ( 71 Pages)
Operating  system:  RT-11

Elapsed time:  00:01:15.26
,LP:TSKSHO=DK:TSKSHO/C/N:SYM
```

Cross reference table (CREF V05.05)

$1STLG 1-67
$8BIT 1-105
$AUTO 1-81 9-114
$CARMN 1-198 9-62
$CARUP 1-79
$CCLRN 1-80
$CFABT 1-100
$CFALL 1-106
$CFCCL 1-106
$CFDCC 1-106
$CFOPN 1-112
$CFSOT 1-104
$CHACT 1-55
$CLTST 1-90
$CTRLC 1-98
$CTRLD 1-149
$CTRLO 1-55
$CTRLS 1-85
$DBKMN 1-78
$DEAD 1-153 9-23 9-159 9-184
$DEBUG 1-150
$DEFER 1-118
$DETCH 1-83
$DIBOL 1-67
$DILUP 1-102 9-118
$DISCN 1-84
$DOOFF 1-108
$DUPRN 1-103
$ECHO 1-105
$EMTTR 1-89
$FORM 1-104
$FORMO 1-106
$HARD 1-153
$HITTY 1-66
$INCOR 1-122 11-118
$INDAB 1-154
$INDDF 1-152
$INDRN 1-152
$INIT 1-153 9-58 9-165
$INKMN 1-98
$KED 1-122
$KINIT 1-62 9-198 11-45
$LC 1-105
$LOFCF 1-198
$MLOCK 1-71 11-125
$NOIN 1-66
$NOINT 1-199
$NOWTT 1-66
$PAGE 1-105
$PHONE 1-153 9-56
$PRGLK 1-81
$QTSET 1-127
$QUIET 1-119
$RNIOP 1-200
$SCOPE 1-105
$SGALL 1-118

Cross reference table (CREF V05.05)


```
$SGHIO    1-117
$SGIIO    1-117
$SGQO     1-99
$SGQ1     1-117
$SGQ1A    1-117
$SGQ1B    1-117
$SGQ1C    1-117
$SGQ2     1-117
$SGQ3     1-99
$SLKED    1-76      28-24
$SLLET    1-197     28-36
$SLON     1-197     28-15
$SLTTY    1-197     28-30
$SNWTT    1-150
$SPLJB    1-85
$STSNG    1-55
$SUCF     1-107
$TAB      1-104
$TAPE     1-148
$TECO     1-127
$TTGAG    1-112
$UCLCF    1-113
$UCLCL    1-115
$UCLCM    1-115
$UCLRN    1-114
$UKMON    1-79
$VNOTT    1-42
$VTESC    1-75
$WILD     1-128
...V1     6-6       6-8       6-10      6-12      6-15      6-17      6-19      6-21      6-22      6-24      6-25      6-27
          6-28      6-30      6-32      6-36      6-38      6-40      7-45      7-47      7-97      7-100     7-103     7-104
          7-107     7-108     7-111     7-118     7-122     7-124     7-127     7-136     7-138     8-7       8-13      8-20
          8-23      8-25      8-32      8-39      8-45      8-51      8-59      8-64      9-13      9-14      9-15      9-43
          9-65      9-81      9-120     9-126     9-132     9-134     9-141     9-143     9-146     9-150     9-152     9-173
          9-175     9-178     9-186     9-192     9-202     9-204     9-211     10-14     10-20     10-23     10-27     10-28
          10-40     10-43     10-58     10-73     10-95     10-108    10-110    10-114    10-124    10-128    11-36     11-37
          11-60     11-61     11-69     11-73     11-85     11-88     11-89     11-96     11-111    11-114    11-120    11-127
          11-129    11-133    11-134    11-140    11-141    11-157    11-160    11-162    11-166    11-173    11-181    11-183
          12-14     12-15     12-21     13-33     14-9      14-10     14-11     15-68     16-21     16-21     16-21     16-26
          16-26     16-32     16-32     16-98     16-98     18-6      18-16     18-19     18-20     18-30     18-33     18-34
          18-43     18-46     18-52     18-58     18-60     18-61     19-21     19-27     19-28     19-29     19-39     19-61
          20-59     20-60     20-78     20-84     20-132    20-144    20-147    20-159    20-166    21-5      21-10     21-11
          21-12     21-15     21-18     21-21     21-24     21-27     22-9      22-12     22-20     22-27     22-33     22-36
          23-10     23-11     24-21     24-31     24-36     24-59     24-72     24-76     25-36     26-33     26-34     26-53
          26-58     26-61     26-62     26-64     26-85     26-87     26-91     26-94     26-96     26-100    26-108    27-11
          27-17     27-18     27-22     27-27     28-11     28-20     28-42     29-14     29-20     29-24     30-17     30-22
          31-15     31-20     31-25     31-29     32-24     32-42     32-51     32-55     32-61     32-67     32-68     33-8
          33-8      33-8      33-10     33-15     33-17     33-21     33-27     33-29     33-33     33-33     33-34     35-10
          35-17     35-24     36-19     36-33     36-35     36-46     36-55     36-56     36-59     37-8
...V2     16-21     16-21     16-21#    16-21#    16-26     16-26     16-26#    16-26#    16-32     16-32     16-32#    16-32#
          16-98     16-98#    33-8      33-8      33-8#     33-8#     33-15     33-15     33-15     33-15#    33-15#    33-15#
          33-27     33-27     33-27     33-27#    33-27#    33-27#    33-33     33-33#
ABRTAD    1-42
ABRTCD    1-42
ABRTCF    1-160
ACRDEC    1-168     36-21
```

Cross reference table (CREF V05.05)

```
ACRFIL   1-167
ACRFN    1-160
ACROCT   1-170
ACRSPD   1-70
AD$$SZ   1-58     19-15     19-65
AD$DVU   1-58     19-40
AD$JOB   1-58     19-13     19-34
ADM3A    1-143
ADM3FL   1-144
ADM3NO   1-146
AF$BYA   1-39     24-96
AF$CCA   1-32     24-102
AF$DBG   1-199    24-95
AF$DUP   1-32     24-98
AF$HIE   1-199    24-90
AF$IND   1-32     24-99
AF$IOP   1-200    24-92
AF$MEM   1-41     24-93
AF$NOI   1-199    24-91
AF$NOW   1-41     24-89
AF$NPW   1-32     24-103
AF$PLK   1-199    24-94
AF$SCA   1-41     24-88
AF$SET   1-32     24-101
AF$TPO   1-39     24-97
AF$UCL   1-32     24-100
ALCDEV   1-28
ALCEND   1-57     19-16     19-66
ALCHD1   1-54     19-28
ALCHD2   1-54     19-29
ALCTBL   1-57     19-12     19-33
ALDBLK   1-183
ALDEX    1-182     1-183
ALFN     1-194
AMBOPT   1-168     1-181     5-18
AR$$SZ   1-197
AR$CNT   1-196
AR$CON   1-196
AR$CPH   1-196
AR$CPL   1-196
AR$DMY   1-197
AR$PRG   1-196
AR$PRJ   1-196
AR$UNM   1-196
ARNRPB   1-197
ASDEX    1-165
ASKLNM   1-164
ASNEND   1-103    18-14     18-28     18-63
ASNHD1   1-177    18-6
ASNHD2   1-177    18-43
ASNOVF   1-165     1-190
ASNTBL   1-102    18-10     18-24     18-38
AT$$SZ   1-100     1-132    18-13     18-27     18-62
AT$DEV   1-132    18-44
AT$EXT   1-132    18-53
AT$FIL   1-132    18-47     18-50
```

Cross reference table (CREF V05.05)

```
AT$LOG    1-132    18-11    18-25    18-39    18-41
AT$SIZ    1-132    18-56
AUTHFN    1-187
BADCMD    1-164
BADPMT    1-171
BADPRI    1-171
BADSAV    1-164
BASMAP    1-137     8-34
BDFNAM    1-167
BDLGOP    1-172
BDLIN     1-185
BDSO      5-11      5-16#
BELL      2-7#
BLANK     2-6#
BLKO      1-163    16-26    16-28    16-32    16-33    16-45    20-14    20-19    20-30    20-57    20-64    20-164
         24-15    24-18    24-21    25-26    32-47    32-51    33-15    33-18    33-22    33-27    33-30
BLKWDS    2-11#
BOTDEV    1-97
BOTUNI    1-97
C.CSW     1-109
C.DEVQ    1-109
C.SBLK    1-109
C1DEVX    1-133    10-85
CALUCL    1-176    13-29
CASCBR    1-91
CASCUP    1-91
CASTBR    1-91
CASTBW    1-91
CASTRO    1-92
CASTWO    1-92
CCLNAM    1-191
CCLSAV    1-88
CD$$SZ    1-110    20-22    20-109
CD$$UB    1-110
CD$BAS    1-110    20-38    20-48    20-82    20-89    20-97    20-99    20-105    20-108
CD$DVU    1-110    20-17    20-36    20-43    20-70    20-91    20-95
CD$JOB    1-110    20-141
CD$NAM    1-110    20-121
CD$TOP    1-127    20-101
CDBAS5    4-11#    20-105    20-108*
CDBASE    4-9#     20-38*    20-48
CDBUF     1-35     20-17    20-36    20-38    20-43    20-48    20-70    20-82    20-89    20-91    20-95    20-97
         20-99    20-101    20-105    20-108    20-121    20-141
CDDVU     4-10#    20-91*    20-95
CDGET     1-35     20-16    20-35    20-42    20-69    20-94    20-115    20-120    20-135
CF$IND    1-154
CF$QUT    1-154
CFABLV    1-155
CFBLK     1-119
CFBUF     1-88
CFCHAN    1-118
CFEND     1-88
CFHOLD    1-126
CFIND     1-108
CFLFL4    1-113
CFNEST    1-111
```

Cross reference table (CREF V05.05)

```
CFPNT     1-119
CFSEND    1-112
CFSP      1-112
CFSPND    1-125
CFSTK     1-62
CFSTS     1-154
CHAIN     1-98
CHKALC    1-105
CHKDEV    1-191
CHKDLM    1-181
CHKEQ     1-39
CHKMNT    1-166
CHKMTX    1-166
CINDAT    1-136
CINFLG    1-42
CKACOJ    1-39
CKCLUS    1-33      10-65
CKPRIV    1-168
CKSYPV    1-39       7-133     23-6
CL$COL    1-60
CL$LEN    1-95
CL$LIN    1-95
CL$LIX    1-56      10-50      10-99
CL$OPT    1-94      10-101
CL$RQH    1-57
CL$SKP    1-95
CL$WID    1-95
CL$WQH    1-57
CLDEVX    1-134     10-79
CLFREE    1-51       9-202
CLLINE    1-51       9-52
CLOPND   10-120     10-166#
CLOPTB   10-102     10-153#
CLOTIR    1-59
CLRPRV    1-36
CLSFCH    1-63
CLSFSP    1-92
CLTOTL    1-92      10-12      10-133
CLUNIT    1-51       9-204
CLVERS    1-51      10-21
CMDBUF    1-159      1-176     13-24
CMDCCL    1-191
CMDDSN    1-27
CMDFRM    1-27
CMDHD     1-26
CMDMEM    1-28      36-6#
CMDOFF    1-26
CMDSET    1-28
CMDSHO    1-28       5-5#
CMDUSE    1-28      34-13#
CMDWHO    1-28      34-6#
CO$BBT    1-109     10-157
CO$BNI    1-94      10-162
CO$BNO    1-94      10-161
CO$CR     1-93      10-163
CO$CTL    1-93      10-164
```

Cross reference table (CREF V05.05)

```
CO$DEF    1-60
CO$DTR    1-92      10-165
CO$FF     1-93      10-154
CO$FFO    1-93      10-160
CO$LC     1-93      10-156
CO$LFI    1-94      10-159
CO$LFO    1-94      10-158
CO$TAB    1-93      10-155
COAD      1-182      1-183
COAF      1-195
COAL      1-182      1-183
COLOO     1-176     11-162
COMMA     2-10#
CONFG2    1-56       9-25
CONFIG    1-107
CORUSR    1-55       9-38      11-57
CPUAH     1-176
CPUAL     1-176
CR        2-5#
CRLF      1-171      6-6       6-8       6-10      6-12      6-15      6-17      6-19      6-21      6-24      6-27      6-30
          6-32       6-36      6-38      6-40      7-47      7-103     7-107     7-127     9-13      9-211     10-128    11-183
         15-68      18-20     18-34     18-61     19-27     19-61     20-159    21-27     22-9      22-20     22-36     24-76
         26-108     27-17     27-18     27-27     28-42     30-17     31-29     32-68     33-34     35-17
CS$RON    1-74
CSHALC    1-97       7-116
CSHDEV    1-102     20-13     20-92
CSHDVN    1-102     20-23     20-110
CSHHD     1-158
CSHMSG    1-198      8-51
CSHSIZ    1-138      8-52
CSIMS1    1-170
CSIMS2    1-168
CSIMS4    1-190
CTLNAM    4-17#      9-70
CTRLTT    1-96       9-50
CURMTX    1-181     36-46
CURPRM    1-124
CVDVNM    1-178     19-41     20-76
CVTTAB    1-159      5-6      36-6
CVTUC     1-158
CW$50H    1-107
CW$PRO    1-56       9-25
CXTBAS    1-101
CXTPAG    1-73
CXTWDS    1-101
DATTIM    1-27
DCCRD     1-131
DCCWR     1-131
DCSR      4-7#      16-28*    16-37     16-61
DCTRD     1-131
DCTWR     1-131
DEADEV    1-166
DELSPC    1-180
DETARG    1-195
DETHD     1-195
DETTXT    1-174     11-69
```

Cross reference table (CREF V05.05)

```
DEVDSP  14-16     15-7#
DEVHD1   1-177
DEVIDL   1-182     1-182     1-183
DEVUNT   1-169
DFJMEM   1-62
DIABFL   1-119
DIABLO   1-142
DIABNO   1-120
DIVIDE   1-175    11-150
DIVSOR   1-188    35-19*    35-20*
DJABMS   1-186
DKASHD   1-53     18-30
DKSAV    1-159
DLCEMT   1-28
DLMSG    1-186
DLTXT    1-173
DMTALL   1-186
DMTARG   1-165
DMTSUB   1-191
DMYDEV   1-39     30-10     31-22
DOASGN   1-79
DORUN    1-27
DOSTOP   1-187
DSPCSR  15-64     16-7#
DSPMEM   8-68     36-12     36-45#
DVEC     4-6#     16-33*    16-38     16-71
DVSHH1   1-53     14-9
DVSHH2   1-53     14-10
DVSHH3   1-53     14-11
DVSTAT   1-70     15-29
DZTXT    1-193
EDIT     1-67
EDTFIL   1-179    24-16     32-48
EM$ACL   1-48
EM$CAP   1-38
EM$CIP   1-47
EM$CLB   1-48
EM$CLN   1-47
EM$CNO   1-38
EM$CPO   1-38
EM$CSE   1-64
EM$HNI   1-62
EM$ICL   1-46
EM$ILN   1-47      1-48
EM$IUN   1-47
EM$NAD   1-40
EM$NPD   1-47
EM$NSF   1-47
EM$NSL   1-48
EM$NUK   1-80
EM$SLT   1-48
EM$SLW   1-48
EM$TSL   1-48
EM$UIO   1-49
ERRLOC   1-61
ERRSEV   1-128
```

Cross reference table (CREF V05.05)

ESC 1-76
FC$CDX 1-158
FC$LNK 1-158
FD$NAM 1-158
FF 2-9#
FILNAM 1-160 1-161 33-5* 33-6* 33-8
FKILL 1-160 5-18 5-19 11-27
FPRINT 1-160
FRAC64 26-52 26-119#
FSTDL 1-83
FSTIOL 1-56 1-73 9-217
GAGMSG 1-185
GENTOP 1-97
GETKCH 1-71
GRT1 1-139
GTRD50 1-165
H.CSR 1-70 16-28
H.VEC 1-70 16-33 16-45
HANBSY 1-170
HANCHN 1-58
HANENT 1-69 15-37
HANIOC 1-33 15-21
HANNAM 4-5# 16-15* 16-16* 16-21
HANPAR 1-70 15-48
HANSIZ 1-69 15-56
HAZEL 1-71
HAZLFL 1-71
HAZLNO 1-71
HIMAP 1-137 8-60
HIPRI 1-171
HNBUF 1-169
HUPARG 1-190
II$$SZ 1-40 23-22
II$FLG 1-40 24-27
II$NAM 1-40 23-19 24-14
II$NPV 1-42 24-48 24-68
II$PRV 1-42 24-47 24-64
IIBUF 1-40 23-19 24-14 24-27 24-47 24-48 24-64 24-68
ILLCMD 1-166
ILSW2 1-77 9-56 9-114
IN$ACT 1-152
IN$CMD 1-152
IN$CNT 1-152
INDACT 1-162
INDERR 1-100
INDSAV 1-152
INDSTA 1-100
INFOMT 1-166
INGADR 1-40 23-16*
INGEMT 1-40 23-17
INSANT 24-26 24-88#
INSPRT 23-21 24-7#
INSTBL 1-40 23-15
INSTBN 1-41 23-23
INVDAT 1-195
INVDEV 1-194

Cross reference table (CREF V05.05)

INVEC 1-153 9-78
INVLDM 1-190
INVLDN 1-167
INVOPT 1-181 5-19 11-27
INVTIM 1-186
IOABFL 1-55
ITRMTP 1-154
JCXPGS 1-137 8-65
JCXSMS 1-193 8-64
JSTKND 1-96
JSWLOC 1-61
K52 1-67
KBMSG 1-175 11-140
KBPARN 1-37 8-20
KBTX 1-179 36-35 36-55 36-59 37-8
KCSIBF 1-164
KCSIMS 1-165
KDOCIN 1-26
KED 1-67
KILEMT 1-187
KL3CLR 1-81
KL4CLR 1-121
KMNBAS 1-149
KMNCHN 1-88
KMNHI 1-72
KMNNAM 1-191
KMNPGS 1-73
KMNSTK 1-73
KMNSTR 1-73
KMNTOP 1-73
KMPRMT 1-133
L 3-70 3-70# 3-71 3-71# 3-72 3-72# 3-73 3-73# 3-74 3-74# 3-75 3-75#
3-76 3-76# 3-77 3-77# 3-78 3-78# 3-79 3-79# 3-80 3-80# 3-81 3-81#
3-82 3-82# 3-83 3-83# 3-84 3-84# 3-85 3-85# 3-86 3-86# 3-87 3-87#
3-88 3-88# 3-89 3-89# 3-90 3-90# 3-91 3-91# 3-92 3-92# 3-93 3-93#
3-94 3-94# 3-95 3-95# 3-96 3-96# 3-97 3-97# 3-98 3-98# 3-99 3-99#
3-100 3-100# 3-101 3-101# 3-102 3-102# 3-103 3-103# 3-104 3-104# 3-105 3-105#
3-106 3-106# 3-107 3-107# 3-108 3-108# 3-109 3-109# 3-110 3-110# 3-111 3-111#
3-112 3-112# 3-113 3-113# 3-114 3-114# 3-115 3-115# 3-116 3-116# 3-117 3-117#
3-118 3-118# 3-119 3-119# 3-120 3-120# 3-121 3-121# 3-122 3-122# 3-123 3-123#
11-8 11-8#
LA120 1-142
LA12FL 1-143
LA12NO 1-143
LA36 1-142
LA36FL 1-120
LA36NO 1-120
LACTIV 1-74
LAFSIZ 1-77
LBSPRI 1-155
LCBIT 1-142
LCDTYP 1-60 9-69
LCLTXT 1-51 9-55
LCLUNT 1-56 9-116 9-161 9-194
LCOL 1-76 1-127
LCONTM 1-85 11-146

Cross reference table (CREF V05.05)


```
LCPUHI    1-85      11-167
LCPULO    1-85      11-168
LD$RON    1-115     32-59
LDBASE    1-116
LDCLEN    1-65      32-14
LDDEVX    1-118
LDFLAG    1-116     32-59
LDMNT     1-64
LDNAM     1-164
LDNAME    1-116     32-19     32-31
LDOPHD    1-172
LDPDEV    1-116     32-65
LDSIZE    1-116     32-53
LF        2-4#
LF$OPN    1-157
LF$WRT    1-157
LFWLIM    1-77
LICTXT    1-139
LINBUF    1-72
LINCNT    1-74
LINCUR    1-77
LINFRE    1-186
LINIR     1-59
LINNXT    1-72
LINPNT    1-74
LINRTS    1-59
LITIME    1-99
LJSW      1-96
LMXLN     1-153      9-72      9-82
LMXNUM    1-151      9-73      9-95
LMXPRM    1-154      9-122     9-138
LNAME     1-54       9-188
LNBLKS    1-101     11-135
LNMAP     1-107
LNPRIM    1-107      9-39     11-71
LNSBLK    1-102     11-136
LNSPAC    1-111
LOCKTX    1-175     11-127
LOFSPC    1-104
LOGASN    1-167
LOGBAS    1-126      1-128
LOGBLK    1-156
LOGBUF    1-156
LOGCHK    1-127
LOGCHN    1-156
LOGCLS    1-171
LOGDVU    1-126      1-128
LOGFLG    1-156
LOGPTR    1-156
LOMAP     1-137      8-33      8-61
LOTBUF    1-75
LOTNXT    1-75
LOTPNT    1-75
LOTSIZ    1-76
LOTSPC    1-76
LOUTIR    1-59
```

Cross reference table (CREF V05.05)


```
LP$7BT    1-46      9-139
LP$ODD    1-46      9-148
LP$PAR    1-46      9-144
LP$SPD    1-46
LPRG1     1-140    11-174
LPRG2     1-140    11-176
LPRI      1-155     7-98     11-93
LPROG     1-84
LPROJ     1-84
LRBFIL    1-108
LRDTIM    1-74
LSCCA     1-104
LSECPT    1-89     11-76
LSTACT    1-72
LSTATE    1-135    11-101
LSTDL     1-83     11-67
LSTHL     1-56      9-213
LSTIOL    1-56
LSTMX     1-151
LSTPL     1-129     9-53      9-163     9-196     9-215     11-65     11-74
LSTPRM    1-125
LSTSL     1-135    11-188
LSTSPL    1-70     12-16
LSUCF     1-80
LSW       1-55      9-58      9-118     9-165     9-198     11-45     11-118
LSW2      1-98
LSW2S     1-103
LSW3      1-103     9-23      9-159     9-184
LSW4      1-121
LSW5      1-81      9-62
LSW6      1-150    11-125
LSW7      1-154    28-15     29-18
LSW8      1-117
LSW9      1-79
LTRMTP    1-143
LTSCMD    1-111
LUNAME    1-84
MAXALC    1-57
MAXASN    1-100
MAXAVL    1-169
MAXMEM    1-61     36-39*    36-47
MAXMTX    1-181    36-33     36-56
MAXPRI    1-61      1-155
MAXSEC    1-89     11-77
MDMTXT    1-202    21-11
MDT       1-71
MHNSIZ    1-90      8-40
MHNSMS    1-91      8-39
MINTIM    1-89     11-145
MISSEQ    1-170
MNBASE    1-180
MNBPC     1-179
MNFLGS    1-179
MNTARG    1-190
MNTDEV    1-165
MNTFUL    1-167
```

Cross reference table (CREF V05.05)

MNTOP 1-180
MNTTXT 1-192 6-25
MONAR1 1-180
MONAR2 1-180
MONHD 1-180
MONVEC 1-155
MSGBUF 1-185
MSGEND 1-185
MTOPHD 1-166
MUL32 1-195
MXCSR 1-151 9-101
MXDTR 1-151
MXJADR 1-63
MXJMEM 1-62 36-29 36-31 36-57
MXJPRI 1-155 7-101
MXPRMT 1-133
MXVEC 1-153 9-87
NAMTOP 1-139
NARGS 3-69# 3-70 3-70 3-71 3-71 3-72 3-72 3-73 3-73 3-74 3-74 3-75
3-75 3-76 3-76 3-77 3-77 3-78 3-78 3-79 3-79 3-80 3-80 3-81
3-81 3-82 3-82 3-83 3-83 3-84 3-84 3-85 3-85 3-86 3-86 3-87
3-87 3-88 3-88 3-89 3-89 3-90 3-90 3-91 3-91 3-92 3-92 3-93
3-93 3-94 3-94 3-95 3-95 3-96 3-96 3-97 3-97 3-98 3-98 3-99
3-99 3-100 3-100 3-101 3-101 3-102 3-102 3-103 3-103 3-104 3-104 3-105
3-105 3-106 3-106 3-107 3-107 3-108 3-108 3-109 3-109 3-110 3-110 3-111
3-111 3-112 3-112 3-113 3-113 3-114 3-114 3-115 3-115 3-116 3-116 3-117
3-117 3-118 3-118 3-119 3-119 3-120 3-120 3-121 3-121 3-122 3-122 3-123
3-123 11-7# 11-8 11-8
NATXT 1-173 9-126
NEDCHR 1-59
NEWJSW 1-96
NFRESB 1-123
NINTXT 1-173 9-186
NLINES 1-109 20-154
NOCCL 1-172
NOCIN 1-164
NODAT 1-178 22-12
NOFIL 1-176 1-184 12-21
NOFLAG 1-166
NOFRDL 1-195
NOIND 1-170
NOKMON 1-191
NOLDMT 1-178 32-24
NONEMS 1-178 20-166 30-22
NOOPTT 1-184
NOPMGN 1-180
NOPRG 1-160 1-164
NOSTRT 1-163
NOTAVL 1-179 32-67
NOTON 1-185
NOTXT 1-173 9-158 26-85 26-94
NOUDC 1-177 13-33
NSPLDV 1-115 31-10
NSWPMS 1-181 36-19
NUCHN 1-108
NUMDCD 1-133 7-72

Cross reference table (CREF V05.05)

```
NUMDEV    1-146     14-18
NUMON     1-77
NUMST    11-105     11-226#
OCTFIX    1-173      9-89      15-31      15-42      15-50      16-65      16-75      16-88      33-20      33-23
OCTPRT    1-193      8-22       9-102     26-103
ODTBAS    1-149
OF$$SZ    1-148
OF$DEV    1-147
OF$FIL    1-147
OF$FLG    1-147
OF$UNT    1-147
OFFEMT    1-187
OFTTXT    1-202     21-15
OKFEND    1-90
OKFILE    1-90
ONTTXT    1-202     21-12
OPRTXT    1-51       9-49
OPTLST    1-36
OT$RON    1-148
OTHRON    1-189
OTRMNT    1-191
OVRCOR    1-163
PO$DBG    1-41
PO$OPR    1-39
PO$SPV    1-37
PO$SYS    1-37       6-33
PARNNL    1-37       8-23
PASLIN    1-128
PAUMSG    1-159
PBFEND    1-112
PEKADR    1-35      25-22*     25-28
PEKEMT    1-35      25-24
PEKSIZ    1-35      25-23*
PF$IOW    1-150
PF$SYS    1-150
PFCO      1-36
PFSO      1-36
PHNTXT    1-202     21-5
PHYMEM    1-95       8-8
PIDCSR   16-55      17-36#
PIDSLT   17-11#     17-36      17-53
PIDVEC   16-83      17-53#
PMBUSY    1-180
PNAME     1-146      1-169     15-14      16-16
POPCF     1-170
PRGALL    1-27
PRGSIZ    1-72
PRGTOP    1-72
PRIVAO    1-38      27-12
PRIVCO    1-41       6-33      27-23
PRIVSO    1-37
PRMBUF    1-125
PRMEND    1-125
PRMPNT    1-124
PROSLT    1-46      17-12      17-17
PRTBUF    1-178     20-118     20-127     20-132
```

Cross reference table (CREF V05.05)

```
PRTDAT    1-194    22-8
PRTDC2    1-175    11-158    11-161    22-35
PRTDC3    1-175    11-139
PRTDEC    1-169     7-46      7-99      7-102     7-106     7-110     7-126     8-19      9-176     9-206     10-22     10-44
         11-84     18-59     20-148    21-14     21-17     21-20     21-23     21-26     22-32     26-54     32-54     33-31
         36-34     36-54     36-58     37-7
PRTFIX    1-172     9-34      9-84      10-54     10-69     11-95     11-155    15-23     15-58     19-51     26-51     26-71
         26-77
PRTFNM    1-178    20-122
PRTKB     8-11      8-29      8-37      8-43      8-49      8-57      8-62      8-67      37-5#
PRTLN     1-174    11-52     11-72
PRTPCT    1-188    35-27
PRTR50    1-194     9-71     11-175    11-177    15-15     18-18     18-32     18-42     18-45     18-49     18-51     18-55
         19-42    20-77     26-40     26-42     30-14     30-16     31-24     32-41
PRTSPC    1-172     9-45      9-77      9-86      9-91      9-104     9-110     9-154     9-180     10-46     10-56     10-60
         10-71    10-75     15-13     15-20     15-28     15-36     15-47     15-55     15-60     16-63     16-70     16-93
         19-44    19-53     26-82
PRTTIM    1-194    22-43     34-13
PRTTMD    1-177    35-16
PRTTMV    1-176    11-169
PRTTOD    1-194    22-19
PRTTTP    1-173     9-108
PRTUNM    1-174     9-200     11-182    19-57
PRTUSE   11-32     35-5#
PRTWRN    1-198
PRVLST    1-41     24-67     24-71     27-16     27-26
PRVOPT    1-37
PUSHCF    1-161
PVNPW     1-36     24-49
PVON      1-81
QHDMS1    1-53     12-14
QHDMS2    1-53     12-15
QUME      1-142
QUMEFL    1-144
QUMENO    1-144
R50BUF    1-165
R50CLO    4-8#
R50COM    1-104
R50DIR    1-161
R50DK     1-194    18-25
R50DSK    1-167
R50DUP    1-162
R50IND    1-161
R50K52    1-162
R50KED    1-162
R50KEX    1-162
R50LD     1-167
R50LDO    1-165    32-29
R50LD7    1-167
R50LOG    1-171
R50MON    1-195
R50NO     1-168
R50PIP    1-162
R50SAV    1-161
R50SY     1-161    18-11
R50TSX    1-162
```

Cross reference table (CREF V05.05)

```
R50TT    1-190
R50UCL   1-162
R50VIR   1-163
RC$$SZ   1-45    25-13    25-23    25-29
RC$AEP   1-44    26-92
RC$AGE   1-44    26-92
RC$BAS   1-43    26-102
RC$CNT   1-43    26-75
RC$EXC   1-44    26-83
RC$FLG   1-44    26-15    26-17    26-19    26-59    26-83    26-92
RC$GBL   1-44    26-19
RC$LCG   1-46    26-17
RC$LEN   1-45    26-46
RC$NAM   1-44    26-39    26-41
RC$OWN   1-43    26-68
RC$PVT   1-45    26-59
RC$USE   1-44    26-15
RCBBAS   1-45    25-11
RCBEND   1-45    25-14
RDB      1-134   30-6
RDBEND   1-134   30-8
RDCMD    1-159    3-82     6-79     6-84     7-48     7-112    7-128    34-7     34-14    36-13    36-20
RDERM    1-163
REMTXT   1-51     9-64
RESDEV   1-148
RGNDSP  25-12    25-27    26-11#
RNMS     1-174   11-107
RONTXT   1-179   32-61
RSR      1-151    9-98
RSTPRV   1-38
RT$$SZ   1-134   30-18
RT$DEV   1-134   30-10
RT$NAM   1-134   30-12    30-15
RUNDEV   1-101
RUNEMT   1-163
RUNHD    1-159
RUNMS    1-195
S$$RUN   1-80    11-108
S$HICP   1-86    11-210
S$INWT   1-87    11-215
S$IOFN   1-82    11-213
S$IOWT   1-140   11-221
S$LOW    1-84    11-214
S$MSWT   1-88    11-219
S$OTFN   1-82    11-209
S$OTLO   1-82    11-212
S$OTWT   1-87    11-216
S$QMIO   1-80    11-223
S$QUSR   1-140   11-220
S$RT     1-84    11-206
S$SFWT   1-87    11-222
S$SPCB   1-141   11-225
S$SPDB   1-141   11-224
S$SPND   1-81    11-218
S$TMWT   1-87    11-217
S$TTFN   1-82    11-208
```

Cross reference table (CREF V05.05)


```
S$TTSC    1-83      11-207
S$TWFN    1-82      11-211
S9600     1-87
SCHAIN    1-118
SD$BAK    1-121
SD$DEL    1-113
SD$FLK    1-114
SD$HLD    1-124
SD$SNG    1-123
SD$WFM    1-114
SDBLK     1-115
SDBU      1-121
SDBUF1    1-115
SDCB      1-129     10-88     12-7      31-12
SDCBND    1-129     10-89     12-8      12-18     31-13     31-27
SDCBSZ    1-135     10-93     12-12     12-17     31-26
SDDVU     1-134     10-91
SDFHD     1-126     12-10
SDFLAG    1-114
SDFORM    1-114
SDNAME    1-135     31-21
SDSFCB    1-113
SDSKIP    1-121
SEARCH    1-159      5-10     11-25
SERFLG    1-55
SETHD     1-168
SETMEM    36-8      36-17#
SF$1ST     1-124
SF$BSY     1-123
SF$HLD     1-124
SFFILE     1-141
SFFLAG     1-126
SFFORM     1-123
SFID       1-70
SFNMBL     1-123
SFQLNK     1-126
SFUSER     1-141
SH$$SZ     1-68
SH$FLG     1-68
SH$NAM     1-68
SH$RTN     1-68
SH$VAL     1-68
SHMTH1     1-177     20-59
SHMTH2     1-177     20-60
SHOALC     6-11       6-83      19-5#
SHOASN     6-9        6-46      18-5#
SHOCL      6-16       6-88      10-5#
SHOCMD     6-52      13-13#
SHODAT     6-63      22-5#
SHODEV     6-7        6-57       6-68      14-5#
SHOHD      1-192      3-69#      5-9
SHOINS     6-35       6-93      23-5#
SHOJOB     6-13       6-73      11-11#     34-6
SHOKEY     1-33       3-87
SHOMDM     6-103     21-5#
SHOMEM     6-18       6-78       8-5#
```

Cross reference table (CREF V05.05)

```
SHOMNT   6-26     6-98     20-5#
SHOPRV   6-37     6-108    27-5#
SHOQUE   6-113    12-6#
SHOREQ   6-31     6-133    25-5#
SHOSLE   6-39     6-123    28-5#
SHOSPL   6-20     6-143    31-6#
SHOSRT   6-29     6-118    30-5#
SHOSUB   6-23     6-138    32-6#
SHOSYP   6-148    7-133#
SHOTIM   6-158    22-19#
SHOTRM   6-14     6-58     6-153    9-5#
SHOUSE   6-163    22-43#
SHOVAL   7-5      7-10     7-15     7-20     7-25     7-30     7-35     7-40     7-45#    7-53     7-58     7-63
         7-68     7-73     7-78     7-83     7-88     7-93
SHOVER   6-5      6-168    22-27#
SHOWVM   6-41     6-128    33-5#
SHRRCB   1-45     25-19
SHRRCN   1-45     25-20
SHTMSG   1-187
SHVTX1   1-200    33-17
SHVTX2   1-200    33-21
SHVTX3   1-200    33-29
SHVTX4   1-200    33-10
SIZEMT   1-190
SIZVAL   1-164
SJOHD    11-7#    11-24
SJSPPN   1-200    11-28*   11-194#
SKPSPC   1-26
SLCKFL   28-26    28-32    28-38    29-10#
SLKDOF   1-49
SLKDON   1-49
SLMXLN   1-198
SMRSIZ   1-138    8-26
SO$NO    1-69
SO$NVL   1-69
SO$OCT   1-69
SOPALC   1-60     3-71     3-72     6-83#
SOPALL   3-70     6-5#
SOPASN   3-73     5-8      6-46#
SOPCL    3-76     3-79     6-88#
SOPCMD   3-77     6-52#
SOPCON   3-75     6-57#
SOPCSH   3-74     7-116#
SOPCTM   3-78     7-57#
SOPDAT   1-60     3-81     6-63#
SOPDEV   3-80     6-68#
SOPHIO   3-83     7-67#
SOPIIO   3-85     7-62#
SOPINS   3-84     6-93#
SOPJMP   6-42     6-47     6-53     6-59     6-64     6-69     6-74     6-79#    6-89     6-94     6-99     6-104
         6-109    6-114    6-119    6-124    6-129    6-134    6-139    6-144    6-149    6-154    6-159    6-164
         6-169
SOPJOB   3-86     3-121    6-73#
SOPMB    3-92     7-4#
SOPMC    3-90     7-9#
SOPMDM   3-94     3-96     6-103#
```

Cross reference table (CREF V05.05)

```
SOPMEM    3-89       6-78#
SOPMNT    3-93       6-98#
SOPMR     3-91       7-14#
SOPNDC    3-95       7-72#
SOPPDF    3-100      7-87#
SOPPHI    3-99       7-82#
SOPPLO    3-98       7-77#
SOPPRI    3-97       7-97#
SOPPRV    3-101      6-108#
SOPPVR    3-102      7-92#
SOPQ0     3-104      7-19#
SOPQ1     3-105      7-24#
SOPQ1A    3-106      7-29#
SOPQ1B    3-107      7-34#
SOPQ1C    3-108      7-39#
SOPQ2     3-109      7-44#
SOPQ3     3-110      7-52#
SOPQUE    3-103      6-113#
SOPREG    3-111      6-133#
SOPSLE    3-114      6-123#
SOPSPL    3-115      6-143#
SOPSRT    3-112      6-118#
SOPSUB    3-88       3-113      6-138#
SOPSYP    3-116      6-148#
SOPTIM    1-60       3-118      6-158#
SOPTRM    3-117      3-119      6-153#
SOPUSE    3-120      6-163#
SOPVER    3-122      6-168#
SOPVM     3-123      6-128#
SPACE1    1-43      26-62
SPACE2    1-174     11-61      11-89      11-96      11-141     11-166     11-173     26-58      26-100     31-25
SPACE3    1-174      7-45      11-181     26-91
SPACE5    1-175     11-129
SPACE6    1-178     20-84
SPACTV    1-182      1-183
SPCF      1-184
SPDTX1    1-173      9-129
SPFLK     1-184
SPFUL     1-184
SPGEMT    1-184
SPLACT    1-187
SPLCHN    1-86
SPLHD     1-181
SPLHLA    1-172
SPLPND    1-189
SPSNG     1-182      1-184
SPUBUF    1-63
SPWFM     1-182      1-183
SRTSIZ    1-138      8-46
SRTSMS    1-189      8-45
SRTTXT    1-192      6-28
SSRMAP    1-192      8-25
STBIT    11-102     11-206#    11-226
STLGCN    1-27
STLGHD    1-171
STNAM    11-113     11-230#
```

Cross reference table (CREF V05.05)

```
STPASK    1-189
STPBUF   11-113*    11-114     11-251#
STPFLG    1-86
SUBARG    1-179     32-42
SUBTXT    1-192      6-22
SUCS      1-139
SUM1      1-188     35-42
SUM2      1-188     35-43
SUM3      1-188     35-44
SUM4      1-188     35-45
SUM5      1-189     35-46      35-49      35-50
SUM6      1-189     35-47
SUM7      1-189     35-48
SUMS      1-139
SUMVEC   35-21      35-42#
SUPCOD    1-139
SWPTX     1-175     11-120
SXBPNT    1-63
SYASHD    1-53      18-16
SYHD1     1-174     11-36
SYHD2     1-174     11-37
SYINDX    1-146
SYNAME    1-147     16-15      18-17      18-31
SYPSWD    1-33       7-134      7-138
SYSAV     1-159
SYSDAT    1-136
SYTIMH    1-136
SYTIML    1-136
SYUNIT    1-146
TAB       2-8#
TALEMT    1-108
TBLOVF    1-168
TECO      1-67
TK1SEC    1-138
TK1VAL    1-136
TM$AUT    1-51        9-120
TM$C13    1-54       10-40
TM$CDS    1-50        7-122
TM$CEN    1-50        7-124
TM$CLO    1-52       10-14
TM$CL1    1-52       10-27
TM$CL2    1-52       10-28
TM$CL3    1-52       10-43
TM$CL4    1-52       10-58      10-73
TM$CL5    1-52       10-95
TM$CL6    1-52       10-108
TM$CL7    1-52       10-20
TM$CL8    1-53       10-23
TM$CNG    1-50        7-118
TM$GBL    1-43       26-64
TM$HPE    1-50        7-111
TM$HPR    1-49        7-108
TM$IN1    1-35       23-10
TM$IN2    1-35       23-11
TM$KED    1-34       28-25
TM$LCL    1-43       26-61
```

Cross reference table (CREF V05.05)

```
TM$LPR    1-49      7-104
TM$NAD    1-54     19-21
TM$NNR    1-35     25-36
TM$NO     1-34     29-20
TM$NSD    1-54     31-15
TM$NSP    1-33      7-136
TM$OFF    1-33     28-17
TM$ON     1-33     28-19
TM$PR1    1-49      7-97
TM$PR2    1-49      7-100
TM$PVA    1-38     27-11
TM$PVC    1-38     27-22
TM$PVL    1-39     24-59
TM$RD1    1-43     26-33
TM$RD2    1-43     26-34
TM$SDN    1-54     31-20
TM$SL1    1-33     28-11
TM$SUB    1-34     28-37
TM$TTY    1-34     28-31
TMIDLH    1-65     35-45
TMIOH     1-65     35-47
TMIOWH    1-64     35-43
TMITXT    1-202    21-18
TMLTXT    1-202    21-21
TMOTXT    1-202    21-24
TMSWPH    1-65     35-48
TMSWTH    1-65     35-44
TMTOTH    1-64      1-188    35-11     35-19
TMTOTL    1-64      1-188    35-12     35-20
TMUSRH    1-64     35-42
TOTMMS    1-192     8-7
TOTON     1-86
TOTXT     1-171
TRGRET    1-139
TRMHD1    1-50      9-14
TRMHD2    1-50      9-15
TRMSTR    1-161
TSKSHO    1-5#      1-26
TSR       1-151
TSXLN     1-139
TSXSIT    1-139
TSXSMS    1-193     8-32
TSXVER    1-36     22-27
TSXVRS    1-32     22-28
TXTC1     1-86      9-173
TXTCL     1-85      9-175
UC$MDC    1-158
UC$NDC    1-158
UCHAN     1-106
UCIDEF    1-58
UCISPC    1-90
UCLBLK    1-157    13-17
UCLCMD    1-26      6-51#
UCLDAT    1-157
UCLNAM    1-111
UERSEV    1-128
```

Cross reference table (CREF V05.05)

UFORM	1-98						
UFPTRP	1-113						
UHIMEM	1-101						
UKMNAM	1-79						
UMSSMS	1-192	8-13					
UMSYTP	1-83	8-14	8-21				
UPTMMS	1-187	35-10					
USPLCH	1-86						
USRMMS	1-193	8-59					
USRSTK	1-62						
USTART	1-97						
UTRPAD	1-61						
VCORTM	1-133	7-57					
VCSHNB	1-89	7-120	7-125				
VHIPCT	1-130	7-67					
VIMAGE	1-96						
VINTIO	1-122	7-62					
VMAXMC	1-34	7-9					
VMXMRB	1-34	7-14					
VMXMSG	1-34	7-4					
VOFFTM	1-201	21-16					
VONTM	1-201	21-13					
VPRIDF	1-78	7-87					
VPRIHI	1-42	7-82	7-109				
VPRILO	1-42	7-77	7-105				
VPRIVR	1-78	7-92					
VQUAN0	1-129	1-130	7-19				
VQUAN1	1-130	7-24					
VQUAN2	1-130	7-44					
VQUAN3	1-129	1-130	7-52				
VQUN1A	1-130	7-29					
VQUN1B	1-122	7-34					
VQUN1C	1-122	7-39					
VSWPFL	1-63	36-17					
VT100	1-142						
VT10FL	1-144						
VT10NO	1-144						
VT200	1-141						
VT2007	1-141						
VT2008	1-141						
VT20FL	1-145						
VT20NO	1-145						
VT52	1-142						
VT52FL	1-143						
VT52NO	1-120						
VTMIN	1-201	21-19					
VTMLOC	1-201	21-22					
VTMOUT	1-201	21-25					
VUSPHN	1-198	9-60	21-7				
WILDFL	1-66						
WLDNAM	2-12#						
WTMS	1-174	11-110					
XAREA	1-160	16-21	16-26	16-32	33-8	33-15	33-27
YESTXT	1-173	9-167	26-87	26-96			
ZCLR	1-151						

Cross reference table (CREF V05.05)

```
...CM1   16-21   16-26   16-32   33-8    33-15   33-27
...CM2   16-21   16-21   16-26   16-26   16-26   16-26   16-32   16-32   16-32   16-32   33-8    33-8
         33-15   33-15   33-15   33-15   33-15   33-27   33-27   33-27   33-27   33-27
...CM3   16-98   33-33
...CM5    6-6     6-8     6-10    6-12    6-15    6-17    6-19    6-21    6-22    6-24    6-25    6-27
          6-28    6-30    6-32    6-36    6-38    6-40    7-45    7-47    7-97    7-100   7-103   7-104
          7-107   7-108   7-111   7-118   7-122   7-124   7-127   7-136   7-138   8-7     8-13    8-20
          8-23    8-25    8-32    8-39    8-45    8-51    8-59    8-64    9-13    9-14    9-15    9-43
          9-65    9-81    9-120   9-126   9-132   9-134   9-141   9-143   9-146   9-150   9-152   9-173
          9-175   9-178   9-186   9-192   9-202   9-204   9-211   10-14   10-20   10-23   10-27   10-28
         10-40   10-43   10-58   10-73   10-95   10-108  10-110  10-114  10-124  10-128  11-36   11-37
         11-60   11-61   11-69   11-73   11-85   11-88   11-89   11-96   11-111  11-114  11-120  11-127
         11-129  11-133  11-134  11-140  11-141  11-157  11-160  11-162  11-166  11-173  11-181  11-183
         12-14   12-15   12-21   13-33   14-9    14-10   14-11   15-68   16-21   16-26   16-32   18-6
         18-16   18-19   18-20   18-30   18-33   18-34   18-43   18-46   18-52   18-58   18-60   18-61
         19-21   19-27   19-28   19-29   19-39   19-61   20-59   20-60   20-78   20-84   20-132  20-144
         20-147  20-159  20-166  21-5    21-10   21-11   21-12   21-15   21-18   21-21   21-24   21-27
         22-9    22-12   22-20   22-27   22-33   22-36   23-10   23-11   24-21   24-31   24-36   24-59
         24-72   24-76   25-36   26-33   26-34   26-53   26-58   26-61   26-62   26-64   26-85   26-87
         26-91   26-94   26-96   26-100  26-108  27-11   27-17   27-18   27-22   27-27   28-11   28-20
         28-42   29-14   29-20   29-24   30-17   30-22   31-15   31-20   31-25   31-29   32-24   32-42
         32-51   32-55   32-61   32-67   32-68   33-8    33-10   33-15   33-17   33-21   33-27   33-29
         33-34   35-10   35-17   35-24   36-19   36-33   36-35   36-46   36-55   36-56   36-59   37-8
...CM7   16-26   16-32
.CLOSE    1-20#  16-98   33-33
.CSIGE    1-18#
.CSISP    1-16#
.DATE     1-19#  22-5
.ENTER    1-21#
.EXIT     1-21#  36-40
.FPROT    1-22#
.GTIM     1-19#
.GTLIN    1-19#
.GVAL     1-22#
.HERR     1-22#  16-99   33-35
.LOOKU    1-20#  16-21   33-8
.PRINT    1-20#   6-6     6-8     6-10    6-12    6-15    6-17    6-19    6-21    6-22    6-24    6-25
          6-27    6-28    6-30    6-32    6-36    6-38    6-40    7-45    7-47    7-97    7-100   7-103
          7-104   7-107   7-108   7-111   7-118   7-122   7-124   7-127   7-136   7-138   8-7     8-13
          8-20    8-23    8-25    8-32    8-39    8-45    8-51    8-59    8-64    9-13    9-14    9-15
          9-65    9-120   9-126   9-173   9-175   9-178   9-186   9-192   9-202   9-204   9-211   10-14
         10-20   10-23   10-27   10-28   10-40   10-43   10-58   10-73   10-95   10-108  10-128  11-36
         11-37   11-61   11-69   11-89   11-96   11-111  11-114  11-120  11-127  11-129  11-140  11-141
         11-162  11-166  11-173  11-181  11-183  12-14   12-15   12-21   13-33   14-9    14-10   14-11
         15-68   18-6    18-16   18-20   18-30   18-34   18-43   18-61   19-21   19-27   19-28   19-29
         19-61   20-59   20-60   20-84   20-132  20-159  20-166  21-5    21-11   21-12   21-15   21-18
         21-21   21-24   21-27   22-9    22-12   22-20   22-27   22-36   23-10   23-11   24-21   24-36
         24-59   24-76   25-36   26-33   26-34   26-58   26-61   26-62   26-64   26-85   26-87   26-91
         26-94   26-96   26-100  26-108  27-11   27-17   27-18   27-22   27-27   28-11   28-20   28-42
         29-20   29-24   30-17   30-22   31-15   31-20   31-25   31-29   32-24   32-42   32-51   32-61
         32-67   32-68   33-10   33-17   33-21   33-29   33-34   35-10   35-17   35-24   36-19   36-33
         36-35   36-46   36-55   36-56   36-59   37-8
.PURGE    1-17#
.PVAL     1-22#
.READW    1-17#  16-26   16-32
.REOPE    1-18#
```

Cross reference table (CREF V05.05)

.SAVES	1-18#											
.SERR	1-22#	16-20	33-7									
.SPFUN	1-19#	33-15	33-27									
.SRESE	1-16#											
.TTOUT	1-16#											
.TTYIN	1-17#											
.TTYOU	1-17#	9-43	9-81	9-132	9-134	9-141	9-143	9-146	9-150	9-152	10-110	10-114
	10-124	11-60	11-73	11-85	11-88	11-133	11-134	11-157	11-160	18-19	18-33	18-46
	18-52	18-58	18-60	19-39	20-78	20-144	20-147	21-10	22-33	24-31	24-72	26-53
	29-14	32-55										
.WRITW	1-21#											
CLOP	10-147#	10-154	10-155	10-156	10-157	10-158	10-159	10-160	10-161	10-162	10-163	10-164
	10-165											
CMDDEF	3-47#	3-70	3-71	3-72	3-73	3-74	3-75	3-76	3-77	3-78	3-79	3-80
	3-81	3-82	3-83	3-84	3-85	3-86	3-87	3-88	3-89	3-90	3-91	3-92
	3-93	3-94	3-95	3-96	3-97	3-98	3-99	3-100	3-101	3-102	3-103	3-104
	3-105	3-106	3-107	3-108	3-109	3-110	3-111	3-112	3-113	3-114	3-115	3-116
	3-117	3-118	3-119	3-120	3-121	3-122	3-123	11-8				
FABORT	3-15#	5-18	5-19	11-27								
FERR	3-4#											
FWARN	3-23#											
TBLDEF	3-35#	3-69	11-7									
TBLEND	3-61#	3-124	11-9									