F 10

```
MMM         MMM  PPPPPPPPPPPP
MMM         MMM  PPPPPP'PPPPP
MMM         MMM  PPPPPPPPPPPP
MMMMMM   MMMMMM  PPP         PFP
MMMMMM   MMMMMM  PPP         PPP
MMMMMM   MMMMMM  PPP         PPP
MMM   MMM   MMM  PPP         PPP
MMM   MMM   MMM  PPP         PPP
MMM   MMM   MMM  PPP         PPP
MMM         MMM  PPPPPPPPPPPP
MMM         MMM  PPPPPPPPPPPP
MMM         MMM  PPPPPPPPPPPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
```

```
**FILE**ID**MPCMOD                                   J 14








MM      MM  PPPPPPPP      CCCCCCCC  MM      MM    000000    DDDDDDDD
MM      MM  PPPPPPPP      CCCCCCCC  MM      MM    000000    DDDDDDDD
MMMM  MMMM  PP      PP  CC          MMMM  MMMM  00      00  DD      DD
MMMM  MMMM  PP      PP  CC          MMMM  MMMM  00      00  DD      DD
MM  MM  MM  PP      PP  CC          MM  MM  MM  00      00  DD      DD
MM  MM  MM  PP      PP  CC          MM  MM  MM  00      00  DD      DD
MM      MM  PPPPPPPP    CC          MM      MM  00      00  DD      DD
MM      MM  PPPPPPPP    CC          MM      MM  00      00  DD      DD
MM      MM  PP          CC          MM      MM  00      00  DD      DD
MM      MM  PP          CC          MM      MM  00      00  DD      DD
MM      MM  PP          CC          MM      MM  00      00  DD      DD      ....
MM      MM  PP          CC          MM      MM  00      00  DD      DD      ....
MM      MM  PP            CCCCCCCC  MM      MM    000000    DDDDDDDD        ....
MM      MM  PP            CCCCCCCC  MM      MM    000000    DDDDDDDD        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
                 0000      1 ;
                 0000      2 ; Version:       'V04-000'
                 0000      3 ;
                 0000      4
                 0000      5         .MCALL  MFPR
        00000001 0000      1 MPSWITCH = 1
                 0000      1         .NLIST  CND
                 0000     17         .TITLE  MPCMOD - MULTIPROCESSING KERNEL SYS SRV DISPATCHER FOR SECONDARY
                 0000     19         .IDENT  'V04-000'
                 0000     20
                 0000     21 ;
                 0000     22 ;****************************************************************************
                 0000     23 ;*                                                                          *
                 0000     24 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
                 0000     25 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
                 0000     26 ;*  ALL RIGHTS RESERVED.                                                    *
                 0000     27 ;*                                                                          *
                 0000     28 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
                 0000     29 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
                 0000     30 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
                 0000     31 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
                 0000     32 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
                 0000     33 ;*  TRANSFERRED.                                                            *
                 0000     34 ;*                                                                          *
                 0000     35 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
                 0000     36 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
                 0000     37 ;*  CORPORATION.                                                            *
                 0000     38 ;*                                                                          *
                 0000     39 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
                 0000     40 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
                 0000     41 ;*                                                                          *
                 0000     42 ;*                                                                          *
                 0000     43 ;****************************************************************************
                 0000     44 ;
                 0000     45 ; D. N. CUTLER 22-JUN-76
                 0000     46 ;
                 0000     47 ;  MODIFIED BY:
                 0000     48 ;
                 0000     49 ;       V03-041 LJK0287         Lawrence J. Kenah       27-Jun-1984
                 0000     50 ;               Add R5 to entry mask for $CANEXH syster service.
                 0000     51 ;
                 0000     52 ;       V03-040 LMP0239         L. Mark Pilant,         23-Apr-1984  9:21
                 0000     53 ;               Change $CHKPRO from an exec mode service to a kernel mode
                 0000     54 ;               service.  This was made necessary by the $CHKPRO (internal
                 0000     55 ;               entry point) interface change.
                 0000     56 ;
                 0000     57 ;       V03-039 MMD0250         Meg Dumont,     27-Feb-1984  17:49
                 0000     58 ;               Add support for $MTACCESS installation specific accessibility
                 0000     59 ;               routine
                 0000     60 ;
                 0000     61 ;       V03-038 DAS0001         David Solomon           20-Feb-1984
                 0000     62 ;               Implement new design for RMS echo SYS$INPUT to SYS$OUTPUT
                 0000     63 ;               (vs V03-019). Echo is now performed by a caller's mode AST
                 0000     64 ;               routine declared in RMS\RM$EXRMS. Change INCB/DECB of FAB/RAB
                 0000     65 ;               busy bit to BISB/BICB, now that we have room.
                 0000     66 ;
                 0000     67 ;       V03-037 SSA0004         Stan Amway              28-Dec-1983
```

```
0000  ;                 For $SETPFM, changed number of parameters from 1 to 4
0000  ;                 and changed entry mask to save R2-R11.
0000  ;
0000  ;         V03-036 TMK0002         Todd M. Katz            19-Nov-1983
0000  ;                 The entry point for $ASCTOID can no longer be reached as a
0000  ;                 branch destination from the executive mode dispatcher.
0000  ;                 A temporary entry point (EXE$ASCTOID) has been placed within
0000  ;                 this module, and a JMP is made from it to the real system
0000  ;                 service entry point (EXE$$ASCTOID).
0000  ;
0000  ;                 Also, change the entry mask for SYS$TRNLOG, so that R8 is
0000  ;                 now saved.
0000  ;
0000  ;         V03-035 TMK0001         Todd M. Katz            22-Oct-1983
0000  ;                 The entry points for $FINISH_RDB and $IDTOASC can no
0000  ;                 longer be reached as branch destinations from the executive
0000  ;                 mode dispatcher. Temporary entry points (EXE$FINISH_RDB and
0000  ;                 EXE$IDTOASC) have been placed within this module, and from
0000  ;                 each a JMP is made to the real system service entry points
0000  ;                 (EXE$$FINISH_RDB and EXE$$IDTOASC).
0000  ;
0000  ;         V03-034 PRB0254         Paul Beck       15-Sep-1983  14:49
0000  ;                 (1) Correct the way synchronous CJF services are defined.
0000  ;                 (2) Define loadable RUF services.
0000  ;
0000  ;         V03-033 WMC0029         Wayne Cardoza           31-Aug-1983
0000  ;                 Loadable services should not be unconditionally inhibited.
0000  ;                 Add an alternate CHMx argument to LDBSRV.
0000  ;
0000  ;         V03-032 DWT0125         David W. Thiel          22-Aug-1983
0000  ;                 Remove CHECKARGLIST and calls to same.
0000  ;
0000  ;         V03-031 MKL0167         Mary Kay Lyons          19-Aug-1983
0000  ;                 Generate loadable service vector for CJF$GETCJI.
0000  ;
0000  ;         V03-030 KBT0578         Keith B. Thompson       8-Aug-1983
0000  ;                 Add parameter to $FILESCAN
0000  ;
0000  ;         V03-029 RAS0178         Ron Schaefer            29-Jul-1983
0000  ;                 Add code to detect the AST/non-AST RMS FAB/RAB race
0000  ;                 condition where an RMS operation is initiated while
0000  ;                 the user FAB/RAB is still waiting for completion of
0000  ;                 previous operation.
0000  ;
0000  ;         V03-028 WMC0028         Wayne Cardoza           29-Jun-1983
0000  ;                 Add CJF services.
0000  ;
0000  ;         V03-027 WMC0027         Wayne Cardoza           23-Jun-1983
0000  ;                 Make old logical name services "all mode".
0000  ;                 Changes to image activator vectors.
0000  ;
0000  ;         V03-026 JWH0222         Jeffrey W. Horn          2-May-1983
0000  ;                 Add LDBSRV macro for vector definitions of loadable
0000  ;                 services.
0000  ;
0000  ;         V03-025 DMW4035         DMWalp                  26-May-1983
0000  ;                 Intergate new logical name structures.
```

```
0000  ;
0000  ;       V03-024 LMP0109         L. Mark Pilant,         28-Apr-1983  15:53
0000  ;               Make $CHKPRO an EXEC mode system service to allow examination
0000  ;               of various system data structures.
0000  ;
0000  ;       V03-024 RAS0147         Ron Schaefer            28-APR-1983
0000  ;               Add $FILESCAN. Add R8 and R9 to $SETPRN register mask.
0000  ;
0000  ;       V03-023 JLV0244         Jake VanNoy             27-APR-1983
0000  ;               Add $BRKTHRUW. Change $BRDCST to all mode service.
0000  ;               $BRDCST now uses $BRKTHRU to do real work.
0000  ;
0000  ;       V03-022 LMP0099         L. Mark Pilant,         13-Apr-1983  19:15
0000  ;               Add the $CHKPRO system service.
0000  ;
0000  ;       V03-021 ACG0319         Andrew C. Goldstein,    21-Mar-1983  13:51
0000  ;               Add $GRANTID and $REVOKID services
0000  ;
0000  ;       V03-020 JLV0234         Jake VanNoy              1-MAR-1983
0000  ;               Add $BRKTHRU service.
0000  ;
0000  ;       V03-019 RAS0120         Ron Schaefer            25-Feb-1983
0000  ;               Add support to echo SYS$INPUT to SYS$OUTPUT.
0000  ;               This involves examining the return code from RMS for $GET;
0000  ;               if the special status RMS$_ECHO (not returned to users)
0000  ;               is found, then create a RAB on the caller's stack and
0000  ;               execute a $PUT operation to echo the line.
0000  ;               A certain amount of RMS synchronization code was
0000  ;               shuffled around in order to make room for this.
0000  ;
0000  ;       V03-018 ACG0317         Andrew C. Goldstein,    22-Feb-1983  15:16
0000  ;               Fix off-by-one in kernel arg vector
0000  ;
0000  ;       V03-017 RSH0004         R. Scott Hanna          10-Feb-1983
0000  ;               Added $ASCTOID, $FINISH_RDB, and $IDTOASC to system service list
0000  ;
0000  ;       V03-016 RNG0016         Rod N. Gamache          1-Feb-1983
0000  ;               Added $GETLKI to system service list
0000  ;
0000  ;       V03-015 WMC0015         Wayne Cardoza           12-Jan-1983
0000  ;               Put back accidentally deleted space holder for RMS synchronization.
0000  ;
0000  ;       V03-014 DMW4023         DMWalp                   7-Jan-1983
0000  ;               Added $CRELNT, $CRELNM, $DELLNM and $TRNLNM
0000  ;
0000  ;       V03-013 KDM0033         Kathleen D. Morse       13-Dec-1982
0000  ;               Correct usage of an interlocked instruction to flush
0000  ;               the hardware cache queue.
0000  ;
0000  ;       V03-012 ROW0146         Ralph O. Weber           6-DEC-1982
0000  ;               Insert routine header comments for INHEXCP, CHECKARGLIST,
0000  ;               and EXE$CMODKRNLX (MPS$CMODKRNLX).  Move things around so
0000  ;               that EXE$CMODKRNL (MPS$CMODKRNL) header comments are near
0000  ;               EXE$CMODRKNL (MPS$CMODKRNL) and ASTEXIT comments are near
0000  ;               ASTEXIT.  Make basic kernal-mode .PSECT definition for Y$CMODK
0000  ;               or MP$CMOD1 immediately after executive mode code so that new
0000  ;               code can be inserted in a way that preserves routine headers,
0000   182 ;                  conditional assembly, and .PSECT definitions.  Backout ROW145,
0000   183 ;                  and in its place, correct conditional assembly of BGEQU 10$
0000   184 ;                  after ACCVIO_RET so that it is assembled only for MPCMOD and
0000   185 ;                  so that it is located before ACCVIO_RET.  Change PCB address
0000   186 ;                  lookup at KERDSP in MPCMOD to use CTL$GL_PCB so that it works
0000   187 ;                  correctly regardless of which processor executes it.
0000   188 ;
0000   189 ;          V03-011 ROW0145         Ralph O. Weber          29-NOV-1982
0000   190 ;                  Move EXE$EXCPTN (and MPS$EXCPTN) to before ASTEXIT (or
0000   191 ;                  MPS$ASTEXIT) in an attempt to make branch destinations in
0000   192 ;                  EXE$CMODKRNL reach.
0000   193 ;
0000   194 ;          V03-010 KDM0030         Kathleen D. Morse       18-Nov-1982
0000   195 ;                  Add logic to MPCMOD that allows the primary to execute
0000   196 ;                  secondary-specific code, without turning into a secondary.
0000   197 ;
0000   198 ;          V03-009 MLJ0099         Martin L. Jack, 20-Oct-1982  19:42
0000   199 ;                  Complete V03-002 by correcting mode and argument count of
0000   200 ;                  $SNDJBC and removing temporary stubs.
0000   201 ;
0000   202 ;          V03-008 RIH0001         Richard I. Hustvedt     1-Jun-1982
0000   203 ;                  Correct handling of AST queue by secondary processor to
0000   204 ;                  avoid losing some AST notifications by incorrectly computing
0000   205 ;                  PHD$B_ASTLVL.
0000   206 ;
0000   207 ;          V03-007 KDM0018         Kathleen D. Morse       30-Sep-1982
0000   208 ;                  Add MPSWITCH logic to create a kernel system service
0000   209 ;                  dispatcher for the secondary processor of an 11/782.
0000   210 ;
0000   211 ;          V03-006 STJ3028         Steven T. Jeffreys      26-Sep-1982
0000   212 ;                  Added $ERAPAT system service vector.
0000   213 ;
0000   214 ;          V03-005 DWT0058         David Thiel             11-Aug-1982
0000   215 ;                  Eliminate use of R2 while waiting for service
0000   216 ;                  completion.
0000   217 ;
0000   218 ;          V03-004 JWH0001         Jeffrey W. Horn         26-Jul-1982
0000   219 ;                  Add new RMS service, RMSRUHNDLR, an un-documented service
0000   220 ;                  which acts as the Recovery Unit handler for RMS.
0000   221 ;
0000   222 ;          V03-003 PHL0102         Peter H. Lipman         16-Jul-1982
0000   223 ;                  Fix new SYNCH logic to always return SS$_NORMAL,
0000   224 ;                  not access IOSB if error from service, and return
0000   225 ;                  error status from $SETEF if event flag cluster went away
0000   226 ;
0000   227 ;          V03-002 PHL0101         Peter H. Lipman         17-Jun-1982
0000   228 ;                  Add $SYNCH system service and fix $QIOW and $ENQW to use the
0000   229 ;                  new code for waiting for the combination of EFN and IOSB
0000   230 ;
0000   231 ;                  Improve readability of conditionals.
0000   232 ;
0000   233 ;                  Add $GETDVIW, $GETJPIW, $GETSYIW, $SNDJBC, $SNDJBCW, and
0000   234 ;                  $UPDSECW.  All the waiting versions use common code.
0000   235 ;
0000   236 ;
0000   237 ;
0000   238 ; CHANGE MODE SYSTEM SERVICE DISPATCHER
```

```
          0000  ;
          0000  ; MACRO LIBRARY CALLS
          0000  ;
          0000
          0000            $ACBDEF                              ;DEFINE AST CONTROL BLOCK OFFSETS
          0000            $CHFDEF                              ;DEFINE CONDITION HANDLING OFFSETS
          0000            $ENQDEF                              ;DEFINE ENQ SYSTEM SERVICE ARGS
          0000            $GETDVIDEF                           ;DEFINE GETDVI SYSTEM SERVICE ARGS
          0000            $GETJPIDEF                           ;DEFINE GETJPI SYSTEM SERVICE ARGS
          0000            $GETLKIDEF                           ;DEFINE GETLKI SYSTEM SERVICE ARGS
          0000            $GETSYIDEF                           ;DEFINE GETSYI SYSTEM SERVICE ARGS
          0000            $IPLDEF                              ;DEFINE INTERRUPT PRIORITY LEVELS
          0000            $LCKDEF                              ;DEFINE INTERLOCK BITS
          0000            $PCBDEF                              ;DEFINE PCB OFFSETS
          0000            $PHDDEF                              ;DEFINE PHD OFFSETS
          0000            $PRDEF                               ;DEFINE PROCESSOR REGISTERS
          0000            $PSLDEF                              ;DEFINE PROCESSOR STATUS FIELDS
          0000            $RABDEF                              ;DEFINE RMS RAB FIELDS
          0000            $RPBDEF                              ;DEFINE REBOOT PARAMETER BLOCK
          0000            $QIODEF                              ;DEFINE QIO SYSTEM SERVICE ARGS
          0000            $SGNDEF                              ;DEFINE SYSGEN PARAMETERS
          0000            $SNDJBCDEF                           ;DEFINE SNDJBC SYSTEM SERVICE ARGS
          0000            $SSDEF                               ;DEFINE SYSTEM STATUS VALUES
          0000            $SYNCHDEF                            ;DEFINE SYNCH SYSTEM SERVICE ARGS
          0000            $UPDSECDEF                           ;DEFINE UPDATE SECTION SYS SRV ARGS
          0000  ;
          0000  ; LOCAL EQUATES
          0000  ;
00000001  0000            CAT0 =          1@0
00000080  0000            CAT7 =          1@7
00000081  0000            DEF_MASK =      CAT0!CAT7            ;INHIBIT FOR 'ALL' AND 'NOT EXIT'
00000080  0000            EXC_MASK =      CAT7                 ;INHIBIT ONLY FOR 'ALL' CASE
          0000  ;
          0000  ; LOCAL MACROS
          0000  ;
          0000  ;         GSYSSRV - GENERATE SYSTEM SERVICE ENTRY VECTOR
          0000  ;
          0000  ;         GSYSSRV SRVNAME,MODE,NARG,REGISTERS,MASK,NOSYNC
          0000  ;
          0000  ;         WHERE:
          0000  ;                 SRVNAME - SERVICE NAME LESS ANY PREFIX (SYS$,EXE$,RMS$$)
          0000  ;                 MODE - MODE DESIGNATOR FOR SERVICE (K,E,ALL,R)
          0000  ;                 NARG - REQUIRED NUMBER OF ARGUMENTS
          0000  ;                 REGISTERS - REGISTER SAVE LIST
          0000  ;                 MASK - SERVICE INHIBIT MASK(BIT SET IN CAT INHIBITS)
          0000  ;                 NOSYNC - NON-ZERO IF RMS SYNCHRONIZATION CODE NOT TO BE INCLUDED
          0000  ;
          0000
          0000            .MACRO  GSYSSRV,SRVNAME,MODE,NARG,REGS,MASK=DEF_MASK,NOSYNC
          0000            .IF     NDF,RMSSWITCH
          0000            .IF     DF,LIBSWITCH
          0000            .PSECT  $$$0000,QUAD
          0000            .IFF
          0000            .PSECT  $$$000,QUAD
          0000            .ENDC
          0000            .ALIGN  QUAD
          0000            .IF DF  LIBSWITCH
```

```
0000    SYS$'SRVNAME::
0000            .IFF
0000            .IF     NDF,MPSWITCH
0000            .WORD   ^M<REGS>
0000            SRVNAME'_MASK = ^M<REGS>
0000            .IFTF   ;MPSWITCH
0000            .IF B   NOSYNC
0000            SRV'MODE        SRVNAME,NARG,MASK
0000            .IFF
0000            SRV'MODE        SRVNAME,NARG,MASK,NOSYNC
0000            .ENDC
0000            .ENDC   ;MPSWITCH
0000            .IFT
0000            .BLKL   2
0000            .ENDC
0000            .IFF
0000            SRV'MODE        SRVNAME,NARG,MASK
0000            .ENDC
0000            .ENDM   GSYSSRV
0000
0000    ;
0000    ;       GCOMPSRVB - GENERATE COMPOSITE SYSTEM SERVICE ENTRY VECTOR BEGIN
0000    ;
0000    ;       GCOMPSRVB  SRVNAME,REGISTER_MASK[,PREFIX]
0000    ;
0000    ;       WHERE:
0000    ;               SRVNAME - SERVICE NAME LESS ANY PREFIX (SYS$, EXE$)
0000    ;               REGISTER_MASK - SYMBOLIC REGISTER MASK, E.G QIO_MASK
0000    ;               PREFIX - IF SUPPLIED, THE PREFIX FOR THE SERVICE NAME.
0000    ;                       IF OMITTED, "SYS$" IS ASSUMED.
0000    ;
0000
0000            .MACRO  GCOMPSRVB,SRVNAME,REGMSK,PREFIX=SYS$
0000            .IF     NDF,MPSWITCH
0000            .IF     NDF,RMSSWITCH
0000            .IF     DF,LIBSWITCH
0000            .PSECT  $$$0000,QUAD
0000            .IFF
0000            .PSECT  $$$000,QUAD
0000            .ENDC
0000            .ALIGN  QUAD
0000            .IF DF  LIBSWITCH
0000            .IIF    NOT_BLANK, <SRVNAME>,-
0000    'PREFIX'SRVNAME::
0000            .IFF
0000            .ENABL  LSB
0000    COMPSTRT=.
0000            .IIF    NOT_BLANK, <REGMSK>,-
0000            .WORD   <REGMSK>
0000            .ENDC
0000            .ENDC
0000            .ENDC   ;MPSWITCH
0000            .ENDM   GCOMPSRVB
0000
0000    ;
0000    ;       GCOMPSRVE - GENERATE COMPOSITE SYSTEM SERVICE ENTRY VECTOR END
0000    ;
```

```
0000  ;       GCOMPSRVE       QUADWORDS
0000  ;
0000  ;       WHERE:
0000  ;               QUADWORDS - NUMBER OF QUADWORDS TO RESERVE FOR VECTOR
0000  ;
0000
0000          .MACRO  GCOMPSRVE,QUADS
0000          .IF     NDF,MPSWITCH
0000          .IF     NDF,RMSSWITCH
0000          .IF     DF,LIBSWITCH
0000          .BLKQ   QUADS
0000          .IFF
0000  COMPSIZE=.-COMPSTRT
0000          .IF     GE,QUADS*8-COMPSIZE
0000          .BLKB   QUADS*8-COMPSIZE
0000          .IFF
0000          .ERROR          ; VECTOR EXCEEDS ALLOCATED SIZE ;
0000          .ENDC
0000          .DSABL  LSB
0000          .ENDC
0000          .ENDC
0000          .ENDC   ;MPSWITCH
0000          .ENDM   GCOMPSRVE
0000
0000
0000  ;
0000  ;       SRVK - GENERATE ENTRY FOR KERNEL MODE SERVICE
0000  ;
0000  ;       SRVK    SRVNAME,NARG,MASK
0000  ;
0000
0000          .MACRO  SRVK,SRVNAME,NARG,MASK
0000          .IF     NDF,RMSSWITCH
0000          .IF     DF,MPSWITCH
0000  CMK$C_'SRVNAME==KCASCTR
0000          .IFF    ;MPSWITCH DEFINED
0000  CMK$C_'SRVNAME=KCASCTR
0000          CHMK    #SRVNAME
0000          RET
0000          .PSECT  Y$CMODKN,BYTE
0000          .=KCASCTR
0000          ASSUME NARG LE 127
0000          .BYTE   NARG
0000          .PSECT  Y$CMODKX,BYTE
0000          .=KCASCTR
0000          .BYTE   MASK
0000          .PSECT  Y$CMODK,BYTE
0000          .SIGNED_WORD    EXE$'SRVNAME-KCASE+2
0000          .IFTF   ;MPSWITCH
0000  SRVNAME=KCASCTR
0000  KCASCTR=KCASCTR+1
0000          .ENDC   ;MPSWITCH
0000          .ENDC
0000          .ENDM   SRVK
0000
0000  ;
0000  ;       SRVE - GENERATE ENTRY FOR EXECUTIVE MODE SERVICE
```

```
0000    ;
0000
0000            .MACRO  SRVE,SRVNAME,NARG,MASK
0000            .IF     NDF,MPSWITCH
0000            .IF     NDF,RMSSWITCH
0000    CME$C_'SRVNAME=ECASCTR
0000            CHME    #SRVNAME
0000            RET
0000            .PSECT  Y$CMODEN,BYTE
0000            .=ECASCTR
0000            ASSUME NARG LE 127
0000            .BYTE   NARG
0000            .PSECT  Y$CMODEX,BYTE
0000            .=ECASCTR
0000            .BYTE   MASK
0000            .PSECT  Y$CMODE,BYTE
0000            .SIGNED_WORD    EXE$'SRVNAME-ECASE+2
0000            .ENDC
0000    SRVNAME=ECASCTR
0000    ECASCTR=ECASCTR+1
0000            .ENDC   ;MPSWITCH
0000            .ENDM   SRVE
0000    ;
0000    ;
0000    ;       MACROS FOR GENERATING RMS SYSTEM VECTORS
0000    ;
0000            .MACRO  RMSSRV  SRVNAME NARG=1,REGS=<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>,-
0000                            MASK,NOSYNC=0
0000            GSYSSRV SRVNAME,R,NARG,<REGS>,MASK,NOSYNC
0000            .ENDM   RMSSRV
0000    ;
0000    ;   SRVR - GENERATE ENTRY FOR RMS SERVICE (EXEC MODE)
0000    ;
0000            .MACRO  SRVR    SRVNAME,NARG,MASK,NOSYNC
0000            .IF     NDF,MPSWITCH
0000            .IF     NDF,RMSSWITCH
0000    CME$C_'SRVNAME=RCASCTR
0000            CHME    #SRVNAME
0000            .IF EQ NOSYNC
0000            .IIF GT <.+2-RMSSYNC>-127,-
0000    RMSSYNC=RMSWBR                                  ;RESET BRANCH DESTINATION
0000    RMSWBR=.
0000            BRB     RMSSYNC
0000            .IFF
0000            RET
0000            .ENDC
0000            .PSECT  Y$CMODEN,BYTE
0000             =RCASCTR
0000            ASSUME NARG LE 127
0000            .BYTE   NARG
0000            .PSECT  Y$CMODEX,BYTE
0000            .=RCASCTR
0000            .BYTE   MASK
0000            .IFF
0000            .PSECT  $$$RMSVEC,BYTE,NOWRT
0000            .SIGNED_WORD    RMS$'SRVNAME-RCASE+2
0000            .ENDC
```

```
0000     SRVNAME=RCASCTR
0000     RCASCTR=RCASCTR+1
0000             .ENDC   ;MPSWITCH
0000             .ENDM   SRVR
0000
0000     ;
0000     ;       SRVALL - GENERATE ENTRY FOR ALL MODE SERVICE
0000     ;
0000
0000             .MACRO  SRVALL,SRVNAME,NARG,MASK
0000             .IF     NDF,MPSWITCH
0000             .IF NDF,RMSSWITCH
0000             JMP     @#EXE$'SRVNAME+2
0000             .ENDC
0000             .ENDC   ;MPSWITCH
0000             .ENDM   SRVALL
0000
```

```
0000          .SBTTL  Macros for Loadable Services
0000
0000  ;
0000  ;       LDBSRV - Generate Loadable Service Vector
0000  ;
0000  ;       LDBSRV  PREFIX,SRVNAME,MODE,REGS,SYN_EFN,SYN_IOSB,ALT_CHMX
0000  ;
0000  ;       Where:
0000  ;               PREFIX          - Prefix for system service vector entry point name
0000  ;               SRVNAME         - Service name less any prefix (SYS$,CJF$, etc.)
0000  ;               MODE            - Mode designator for service (K,E,ALL)
0000  ;               REGS            - Register save list
0000  ;               SYN_EFN         - Event flag argument number for $SYNCH
0000  ;               SYN_IOSB        - IOSB argument number for $SYNCH
0000  ;               ALT_CHMX        - Use same CHMx number as this service
0000  ;
0000
0000          .MACRO  LDBSRV,PREFIX,SRVNAME,MODE,REGS,SYN_EFN,SYN_IOSB,ALT_CHMX
0000          .IF NDF,RMSSWITCH
0000          .IF NDF,MPSWITCH
0000              .IF DF,LIBSWITCH
0000                  .PSECT  $$$0000,QUAD
0000                  .ALIGN  QUAD
0000  PREFIX''SRVNAME::
0000                  .IF BLANK SYN_EFN
0000                      .BLKL    2
0000                  .IFF
0000                      .BLKL    4
0000                  .ENDC
0000              .IFF
0000                  .PSECT  $$$000,QUAD
0000                  .ALIGN  QUAD
0000                  .WORD   ^M<REGS>
0000                  SRVNAME'_MASK = ^M<REGS>
0000                  LVEC_'MODE PREFIX,SRVNAME,SYN_EFN,SYN_IOSB,ALT_CHMX
0000              .ENDC
0000          .ENDC   ; MPSWITCH
0000          .ENDC   ; RMSSWITCH
0000          .ENDM   LDBSRV
0000
0000  ;
0000  ;       LVEC_K  - Kernel Mode Loadable System Service Vector
0000  ;
0000  ;       LVEC_K  PREFIX,SERVICE,EFN,IOSB
0000  ;
0000
0000          .MACRO  LVEC_K,PREFIX,SERVICE,EFN,IOSB,ALT_CHMK
0000          .IF BLANK ALT_CHMK
0000              CMK$C_'SERVICE = PREFIX'KCASCTR
0000          .IFF
0000              CMK$C_'SERVICE = ALT_CHMK
0000          .ENDC
0000          CHMK #SERVICE
0000          .IF NOT BLANK EFN
0000              PUSHL       #EFN
0000              PUSHL       #IOSB
0000              JMP         @#EXE$LDB_SYNCH
```

```
0000          .IFF
0000              RET
0000          .ENDC
0000          .IF BLANK ALT_CHMK
0000              SERVICE = PREFIX'KCASCTR
0000              PREFIX'KCASCTR = PREFIX'KCASCTR + 1
0000          .IFF
0000              SERVICE = ALT_CHMK
0000          .ENDC
0000          .ENDM   LVEC_K
0000
0000    ;
0000    ;     LVEC_E  - Exec Mode Loadable System Service Vector
0000    ;
0000    ;     LVEC_E  PREFIX,SERVICE,EFN,IOSB
0000    ;
0000
0000          .MACRO  LVEC_E,PREFIX,SERVICE,EFN,IOSB,ALT_CHME
0000          .IF BLANK ALT_CHME
0000              CME$C_'SERVICE = PREFIX'ECASCTR
0000          .IFF
0000              CME$C_'SERVICE = ALT_CHME
0000          .ENDC
0000          CHME    #SERVICE
0000          .IF NOT BLANK EFN
0000              PUSHL       #EFN
0000              PUSHL       #IOSB
0000              JMP         @#EXE$LDB_SYNCH
0000          .IFF
0000              RET
0000          .ENDC
0000          RET
0000          .IF BLANK ALT_CHME
0000              SERVICE = PREFIX'ECASCTR
0000              PREFIX'ECASCTR = PREFIX'ECASCTR + 1
0000          .IFF
0000              SERVICE = ALT_CHME
0000          .ENDC
0000          .ENDM   LVEC_E
0000
0000    ;
0000    ;     LVEC_ALL - Mode of caller Loadable System Service Vector
0000    ;
0000    ;     LVEC_ALL   PREFIX,SERVICE,EFN,IOSB
0000    ;
0000          .MACRO  LVEC_ALL,PREFIX,SERVICE,EFN,IOSB,ALT_CHMK
0000          JMP     @#EXE$'SERVICE
0000          .IF NOT BLANK EFN
0000              .ERROR      ; SYNCH NOT ALLOWED FOR ALL-MODE SERVICES
0000          .ENDC
0000          .ENDM   LVEC_ALL
0000
0000
0000
0000
0000
0000
```

```
    0000   698 ;
    0000   699 ; Establish .PSECT for kernel-mode servicing code which follows
    0000   700 ;
00000000   704         .PSECT  MP$CMOD1,QUAD
```

```
                          0000          .SBTTL  INHEXCP - Inhibited CHMK or CHME code handling
                          0000
                          0000     ;+
                          0000     ;
                          0000     ; INHEXCP - Inhibited CHMK or CHME code handling
                          0000     ;
                          0000     ; FUNCTIONAL DESCRIPTION:
                          0000     ;
                          0000     ; When tne ability to use specified system services is inhibited
                          0000     ; via the $SETSSF system service, this routine receives control
                          0000     ; when an attempt to execute an inhibited system service occurs.
                          0000     ;
                          0000     ; The exception condition is returned to the primary processor for execption
                          0000     ; handling.
                          0000     ;
                          0000     ; INPUTS:
                          0000     ;
                          0000     ;       R1      = SS error code (SS$_INHCHMK or SS$_INHCHME)
                          0000     ;       00(SP) = Change mode parameter code
                          0000     ;       04(SP) = Saved PC of exception
                          0000     ;       08(SP) = Saved PSL of exception
                          0000     ;
                          0000     ; ENVIRONMENT:
                          0000     ;
                          0000     ;       This code executes on the secondary processor.
                          0000     ;       If interrupted at any point, may continue on the primary processor.
                          0000     ;
                          0000     ;-
                          0000     INHEXCP:
                 51  DD   0000          PUSHL   R1                      ;PUSH THE EXECPTION CODE
                 04  DD   0002          PUSHL   #4                      ;PUSH THE NUMBER OF ARGUMENTS
                          0004          IFPRIMARY <JMP G^EXE$REFLECT>   ;IF PRIMARY, THEN CONTINUE RIGHT ALONG
                          001D                                          ;IF SECONDARY, RETURN PROCESS TO PRIMARY
 7E   10 AE   02   18  EF   001D          EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,16(SP),-(SP) ;CREATE PSL WITH PREV
      6E   6E   16  9C   0023          ROTL    #PSL$V_PRVMOD,(SP),(SP) ; MODE CORRECT AND CURRENT MODE = KERNEL
      00000000'GF   9F   0027          PUSHAB  G^EXE$REFLECT           ;REFLECT THE EXCEPTION
            FFD0'   31   002D          BRW     MPS$MPSCHED2            ; AND RETURN PROCESS TO PRIMARY
```

```
                                 0030          .SBTTL MPS$ASTEXIT - AST EXIT SYSTEM SERVICE FOR SECONDARY PROCESSOR
                                 0030    ;+
                                 0030    ; FUNCTIONAL DESCRIPTION:
                                 0030    ;
                                 0030    ; This is the AST exit system service routine for the secondary processor
                                 0030    ; only.  It clears the AST active bit for the appropriate mode, in the
                                 0030    ; process' PCB and then sets a new AST level (both in the PHD and the
                                 0030    ; secondary's processor register).  Because an AST may be delivered by
                                 0030    ; the primary while the secondary is executing this code, the routine
                                 0030    ; is repeated until the head of the AST queue is stable.
                                 0030    ;
                                 0030    ;
                                 0030    ; INPUTS:
                                 0030    ;
                                 0030    ;        (SP) - PC at time of interrupt
                                 0030    ;       4(SP) - PSL at time of interrupt
                                 0030    ;
                                 0030    ; ENVIRONMENT:
                                 0030    ;
                                 0030    ;       Executes on the secondary processor.
                                 0030    ;       If interrupted at any point, may continue on the primary processor.
                                 0030    ;
                                 0030    ;-
                                 0030
                             00000000          .PSECT  MP$CMOD2,BYTE
                                 0000    MPS$ASTEXIT:
50   04 AE    02    18    EF     0000          EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,4(SP),R0 ; Get previous mode
                    54    DD     0006          PUSHL   R4                         ; Save register
                    53    DD     0008          PUSHL   R3                         ; Save register (This is faster)
                    52    DD     000A          PUSHL   R2                         ; Save register (than a PUSHR.)
     54    0000'CF        D0     000C          MOVL    W^M S$GL_CURPCB,R4         ; Get address of current process' PCB
                                 0011          SETIPL  #IPL$_SYNCH                ; Disable system events
   00 0C A4    50         E7     0014          BBCCI   R0,PCB$B_ASTACT(R4),10$ ; Clear AST active bit for this mode
      50    10 A4         DE     0019    10$:  MOVAL   PCB$L_ASTQFL(R4),R0        ; Get address of AST queue
              52    04    D0     001D          MOVL    #4,R2                      ; Assume null AST level
              51    60    D0     0020          MOVL    (R0),R1                    ; Get flink
              51    50    D1     0023          CMPL    R0,R1                      ; Is the queue empty?
                    0D    13     0026          BEQL    20$                        ; Br on yes, set null AST level
                    52    D4     0028          CLRL    R2                         ; Assume kernel mode
                                 002A          ASSUME  ACB$V_KAST EQ 7
                 0B A1    95     002A          TSTB    ACB$B_RMOD(R1)             ; Check for kernel AST
                    06    19     002D          BLSS    20$                        ; Br if not kernel AST
  52    0B A1    FC 8F    8B     002F          BICB3   #^C<3>,ACB$B_RMOD(R1),R2 ; Get request mode
         53    6C A4      D0     0035    20$:  MOVL    PCB$L_PHD(R4),R3           ; Get address of PHD
              13    52    DA     0039          MTPR    R2,#PR$_ASTLVL             ; Set ASTLVL register
       00CF C3    52      90     003C          MOVB    R2,PHD$B_ASTLVL(R3)        ; Set ASTLVL in PHD
   00 0000'CF    00       E6     0041          BBSSI   #LCK$V_INTERLOCK,W^MPS$GL_INTERLOCK,30$ ; Flush cache queue
              51    60    D1     0047    30$:  CMPL    (R0),R1                    ; Has the head of the queue changed?
                    CD    12     004A          BNEQ    10$                        ; Yes, repeat ASTLVL computation
              52    8E    7D     004C          MOVQ    (SP)+,R2                   ; Restore registers
                 54 8ED0         004F          POPL    R4                         ; Restore register
                       02        0052          REI                                ; Return from interrupt
                             00000030          .PSECT  MP$CMOD1,QUAD
                    0030   872          .SBTTL  CHANGE MODE DETECTED ERROR HANDLING
                    0030   873  ;+
                    0030   874  ; ACCVIO - ACCESS VIOLATION DETECTED IN ARGUMENT LIST
                    0030   875  ; INSARG - INSUFFICIENT ARGUMENTS SUPPLIED FOR SERVICE
                    0030   876  ; SSFAIL - ABNORMAL STATUS RETURNED BY SERVICE ROUTINE
                    0030   877  ;
                    0030   878  ; THESE ROUTINES TAKE THE APPROPRIATE ACTION TO RETURN THE ERROR INDICATION
                    0030   879  ; TO THE ORIGINAL CALLER.
                    0030   880  ;
                    0030   881  ;-
                    0030   882          .ENABL  LSB
                    0030   883  ACCVIO:                                         ;
      5D    5E  D0  0030   884          MOVL    SP,FP                           ;SET FRAME POINTER BEFORE RET
0057'8F     50  B1  0033   885          CMPW    R0,#KCASCTR                     ;IS THIS AN UNRECOGNIZED CODE?
            09  1E  0038   887          BGEQU   10$                             ;YES, NOT NECESSARILY ACCVIO
      50    0C  3C  003A   892          MOVZWL  #SS$_ACCVIO,R0                  ;SET ACCESS VIOLATION
                04  003D   893          RET                                     ;
                    003E   894
0057'8F     50  B1  003E   895  KINSARG:CMPW    R0,#KCASCTR                     ;IS THIS AN UNRECOGNIZED CODE?
            72  1E  0043   896  10$:    BGEQU   KERDSP                          ;YES, NOT NECESSARILY INSARG
50    0114  8F  3C  0045   900          MOVZWL  #SS$_INSFARG,R0                 ;SET INSUFFICIENT NUMBER OF ARGUMENTS
                04  004A   902          RET                                     ;
                    004B   903  SRVEXIT:                                        ;SERVICE EXIT
         08 50  E9  004B   904          BLBC    R0,SSFAIL                       ;BR IF ABNORMAL COMPLETION
                02  004E   905  SRVREI: REI                                     ;
                    004F   909  MPS$EXCPTN::                                    ;SYSTEM SERVICE EXCEPTION
              0000  004F   911          .WORD   0                               ;ENTRY MASK
                    0051   915          SECBUG_CHECK SSRVEXCEPT,FATAL           ;UNEXPECTED SYSTEM SERVICE EXCEPTION
      50    07  D3  0056   917  SSFAIL: BITL    #7,R0                           ;TEST SEVERITY FIELD
            F3  13  0059   918          BEQL    SRVREI                          ;IF EQL WARNING
         FFF5'  31  005B   919          BRW     SSFAILMAIN                      ;GOTO MAIN SSFAIL LOGIC
                    005E   920          .DSABL  LSB
```

```
                                          005E   922          .SBTTL  Filtered Change Mode to Kernel Dispatcher
                                          005E   923  ;+
                                          005E   924  ;
                                          005E   928  ; MPS$CMODKRNLX - Secondary Filtered Change Mode to Kernel Dispatcher
                                          005E   930  ;
                                          005E   931  ; When inhibiting of user mode system service calls has been enabled via the
                                          005E   935  ; SSINHIBIT SYSGEN parameter, this routine -- not MPS$CMODKRNLX -- is called
                                          005E   937  ; whenever a CHMK instruction is executed.  The state of the stack on entry
                                          005E   938  ; is:
                                          005E   939  ;
                                          005E   940  ; INPUTS:
                                          005E   941  ;
                                          005E   942  ;       00(SP) = CHANGE MODE PARAMETER CODE.
                                          005E   943  ;       04(SP) = SAVED PC OF EXCEPTION.
                                          005E   944  ;       08(SP) = SAVED PSL OF EXCEPTION.
                                          005E   945  ;
                                          005E   946  ;       00(AP) = NUMBER OF SYSTEM SERVICE CALL ARGUMENTS.
                                          005E   947  ;       04(AP) = FIRST ARGUMENT.
                                          005E   948  ;                 .
                                          005E   949  ;                 .
                                          005E   950  ;                 .
                                          005E   951  ;       4*N(AP) = N'TH ARGUMENT.
                                          005E   952  ;
                                          005E   953  ; OUTPUTS:
                                          005E   954  ;
                                          005E   955  ;       THE APPROPRIATE KERNEL MODE SYSTEM SERVICE IS INVOKED.
                                          005E   956  ;-
                                          005E   957
                                      0000005E   964          .PSECT  MPS$CMOD1,QUAD
                                          005E   966
                                          005E   967          .ALIGN  QUAD
                                          0060   971 MPS$CMODKRNLX::
     50    03000000 8F    08 AE    CB     0060   973          BICL3   8(SP),#PSL$M_CURMOD,R0   ;CHECK THE PREVIOUS MODE
                             1D    12     0069   977          BNEQ    W^MPS$CMODKRNL           ;NO CHECK NEEDED FOR NON-USER MODE
                       50    6E    9A     006B   979          MOVZBL  (SP),R0                  ;PICK UP THE CHMK CODE
  00000000'GF     00000000'GF40    93     006E   984          BITB    G^SYS$GB_KMASK[R0],G^CTL$GB_SSFILTER ;'AND' WITH INHIBIT MASK
                             0C    13     007A   985          BEQL    W^MPS$CMODKRNL           ;THIS CODE IS ALLOWED
              51     04CC 8F       3C     007C   987          MOVZWL  #SS$_INHCHMK,R1          ;SET THE EXECPTION CODE
                           FF7C    31     0081   988          BRW     INHEXCP                  ;AND REFLECT IT
                                          0084   989
```

```
                              0084   991         .SBTTL  CHANGE MODE TO KERNEL DISPATCHER
                              0084   992 ;+
                              0084   996 ; MPS$CMODKRNL - SECONDARY CHANGE MODE TO KERNEL DISPATCHER
                              0084   998 ;
                              0084   999 ; THIS ROUTINE IS AUTOMATICALLY VECTORED TO WHEN A CHANGE MODE TO KERNEL
                              0084  1000 ; INSTRUCTION IS EXECUTED. THE STATE OF THE STACK ON ENTRY IS:
                              0084  1001 ;
                              0084  1002 ; INPUTS:
                              0084  1003 ;
                              0084  1004 ;        00(SP) = CHANGE MODE PARAMETER CODE.
                              0084  1005 ;        04(SP) = SAVED PC OF EXCEPTION.
                              0084  1006 ;        08(SP) = SAVED PSL OF EXCEPTION.
                              0084  1007 ;
                              0084  1008 ;        00(AP) = NUMBER OF SYSTEM SERVICE CALL ARGUMENTS.
                              0084  1009 ;        04(AP) = FIRST ARGUMENT.
                              0084  1010 ;                 .
                              0084  1011 ;                 .
                              0084  1012 ;                 .
                              0084  1013 ;        4*N(AP) = N'TH ARGUMENT.
                              0084  1014 ;
                              0084  1015 ; OUTPUTS:
                              0084  1016 ;
                              0084  1017 ;        THE APPROPRIATE KERNEL MODE SYSTEM SERVICE IS INVOKED.
                              0084  1018 ;-
                              0084  1019
                              0084  1020         .ALIGN  QUAD
                              0088  1024 MPS$CMODKRNL::                         ;2NDARY CHANGE MODE TO KERNEL DISPATCH
                              0088  1026                                        ;NOTE: MEMORY WRITING INSTRUCTIONS ARE
                              0088  1027                                        ;CAREFULLY INTERLACED WITH REGISTER
                              0088  1028                                        ;INSTRUCTIONS FOR SPEED.
                              0088  1029
                      50 8ED0 0088  1035         POPL    R0                     ;REMOVE CHANGE MODE PARAMETER FROM STACK
                      7E   13 008B  1039         BEQL    ASTEXIT                ;IF ZERO, AST EXIT SYSTEM SERVICE
                   BB AF   9F 008D  1041         PUSHAB  B^SRVEXIT              ;RETURN ADDRESS
                51    50   9A 0090  1042         MOVZBL  R0,R1                  ;BOUND RANGE OF CHMK CODES TO 0,255
                              0093  1043                                        ;AND 256 BYTES ACCESSIBLE FROM B_KRNLNARG
                      5D   DD 0093  1044         PUSHL   FP                     ;SAVE FP
51    00000000'GF41        9A 0095  1048         MOVZBL  G^SYS$GB_KRNLNARG[R1],R1 ;GET NUMBER OF REQUIRED ARGUMENTS
                      5C   DD 009D  1050         PUSHL   AP                     ;SAVE AP
5D    00000004 9F41        DE 009F  1051         MOVAL   @#4[R1],FP             ;CALCULATE LENGTH OF ARGUMENT LIST
                      7E   7C 00A7  1052         CLRQ    -(SP)                  ;PSW AND REGISTER SAVE MASK
                              00A9  1056         IFNORD  FP,(AP),ACCVIO1        ;DECLARE ACCESS VIOLATION
                5D    5E   D0 00AF  1058         MOVL    SP,FP                  ;SET FRAME POINTER FOR CALL FRAME
                51    6C   91 00B2  1059         CMPB    (AP),R1                ;CHECK FOR REQUIRED NUMBER OF ARGS
                      59   1F 00B5  1065         BLSSU   KINSARG1               ;IF LSSU, INSUFFICIENT ARGUMENTS
   54    00000000'GF       D0 00B7  1066 KERDSP: MOVL    G^CTL$GL_PCB,R4        ;GET CURRENT PROCESS PCB ADDRESS
      003B'8F         50   B1 00BE  1067         CMPW    R0,#WAITFR             ;IS THIS THE WAITFR SYSTEM SERVICE?
                      4E   13 00C3  1068         BEQL    MPS$WAITFR1            ;BR ON YES, EXECUTE SYS SRV ON SECONDARY
      003D'8F         50   B1 00C5  1069         CMPW    R0,#WFLAND             ;IS THIS THE WFLAND SYSTEM SERVICE?
                      4A   13 00CA  1070         BEQL    MPS$WFLAND1            ;BR ON YES, EXECUTE SYS SRV ON SECONDARY
      003E'8F         50   B1 00CC  1071         CMPW    R0,#WFLOR              ;IS THIS THE WFLOR SYSTEM SERVICE?
                      46   13 00D1  1072         BEQL    MPS$WFLOR1             ;BR ON YES, EXECUTE SYS SRV ON SECONDARY
                5E    08   C0 00D3  1073         ADDL    #8,SP                  ;CLEAN OFF PSW AND REG SAVE MASK
                      5C 8ED0 00D6  1074         POPL    AP                     ;RESTORE AP
                      5D 8ED0 00D9  1075         POPL    FP                     ;RESTORE FP
                6E    50   D0 00DC  1076         MOVL    R0,(SP)                ;REPLACE CHMK ON STACK OVER RET ADR
                              00DF  1077         IFPRIMARY <JMP G^EXE$CMODKRNL> ;IF PRIMARY, THEN CONTINUE RIGHT ALONG
```

```
                            00F8  1078                                                   ;IF SECONDARY, RETURN PROCESS TO PRIMARY
 /C   08 AE    02    18   EF  00F8  1079            EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,8(SP),-(SP) ;CREATE PSL WITH PREV
         6E    6E    16   9C  00FE  1080            ROTL    #PSL$V_PRVMOD,(SP),(SP) ; MODE CORRECT AND CURRENT MODE = KERNEL
          00000000'GF     9F  0102  1081            PUSHAB  G^EXE$CMODKRNL                   ;EXECUTE THE SERVICE ON PRIMARY
                   FEF5'  31  0108  1082            BRW     MPS$MPSCHED2                     ; AND RETURN PROCESS TO PRIMARY
                              010B  1083
                              010B  1084 ASTEXIT:
                     F3'  11  010B  1085            BRB     MPS$ASTEXIT                      ;BRANCH ASSIST
                              010D  1086 ACCVIO1:
                   FF20   31  010D  1087            BRW     ACCVIO                           ;BRANCH ASSIST
                              0110  1088 KINSARG1:
                   FF2B   31  0110  1089            BRW     KINSARG                          ;BRANCH ASSIST
                              0113  1090
                              0113  1091 ;
                              0113  1092 ; BRANCH ASSISTS TO REACH SYSTEM SERVICES.
                              0113  1093 ;
                              0113  1094 MPS$WAITFR1:
                   FEEC'  31  0113  1095            BRW     MPS$WAITFR+2                     ;BRANCH ASSIST (PAST REG SAVE MASK)
                              0116  1096 MPS$WFLAND1:
                   FEE9'  31  0116  1097            BRW     MPS$WFLAND+2                     ;BRANCH ASSIST (PAST REG SAVE MASK)
                              0119  1098 MPS$WFLOR1:
                   FEE6'  31  0119  1099            BRW     MPS$WFLOR+2                      ;BRANCH ASSIST (PAST REG SAVE MASK)
                              011C  1101 KCASE:                                              ;BASE OF CHMK CASE TABLE
               00000001       011C  1102 KCASCTR=1                                           ;CHMK CODES START AT 1
                              011C  1209            .ALIGN  QUAD
```

```
0120  ;
0120  ;       DEFINE REMAINING SERVICES
0120  ;
0120
0120          GSYSSRV ADJSTK,K,3,-                    ;ADJUST OUTER MODE STACK POINTER
0120                  <R2,R3,R4,R5,R6>,-              ;REGISTERS R2-R6
0120                  EXC_MASK                        ;EXCEPTION MASK
0000          GSYSSRV ADJWSL,K,2,-                    ;ADJUST WORKING SET LIMIT
0000                  <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000          GSYSSRV ALCDNP,K,4,-                    ;ALLOCATE DIAGNOSTIC PAGE
0000                  <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R7
0000          GSYSSRV ALLOC,K,4,-                     ;ALLOCATE DEVICE
0000                  <R2,R3,R4,R5,R6>                ;REGISTERS R2-R6
0000          GSYSSRV ASCEFC,K,4,-                    ;ASSOCIATE COMMON EVENT FLAG CLUSTER
0000                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000          GSYSSRV ASCTIM,ALL,3,-                  ;CONVERT TO ASCII TIME
0000                  <R2,R3,R4,R5,R6>                ;REGISTERS R2-R6
0000          GSYSSRV ASSIGN,K,4,-                    ;ASSIGN I/O CHANNEL
0000                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000          GSYSSRV BINTIM,ALL,2,-                  ;CONVERT TO BINARY TIME
0000                  <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000          GSYSSRV CANCEL,K,1,-                    ;CANCEL I/O ON CHANNEL
0000                  <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000          GSYSSRV CANTIM,K,2,-                    ;CANCEL TIMER REQUEST
0000                  <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000          GSYSSRV CANWAK,K,2,-                    ;CANCEL WAKE UP REQUESTS
0000                  <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000          GSYSSRV CRMPSC,K,12,-                   ;CREATE AND MAP SECTION
0000                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000          GSYSSRV CLRPAR,K,2,-                    ;CLEAR HARD PARITY ERROR
0000                  <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000          GSYSSRV CMEXEC,E,2,-                    ;CHANGE MODE TO EXECUTIVE
0000                  <R4>                            ;REGISTER R4
0000          GSYSSRV CMKRNL,K,2,-                    ;CHANGE MODE TO KERNEL
0000                  <R4>                            ;REGISTER R4
0000          GSYSSRV CLREF,K,1,-                     ;CLEAR EVENT FLAG
0000                  <R2,R3,R4,R5>                   ;REGISTERS R2-R5. SEE WAITFR COMMENTS.
0000          GSYSSRV CNTREG,K,4,-                    ;CONTRACT REGION
0000                  <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R7
0000          GSYSSRV GETPTI,K,5,-                    ;GET PAGE TABLE INFORMATION
0000                  <R2,R3,R4,R5,R6,R7,R8,R9,R10> ;REGISTERS R2-R10
0000          GSYSSRV CRELOG,ALL,4,-                  ;CREATE LOGICAL NAME
0000                  <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000          GSYSSRV CREMBX,K,7,-                    ;CREATE MAILBOX
0000                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000          GSYSSRV CREPRC,K,12,-                   ;CREATE PROCESS
0000                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000          GSYSSRV CRETVA,K,3,-                    ;CREATE VIRTUAL ADDRESS
0000                  <R2,R3,R4,R5,R6,R7,R8>,-;REGISTERS R2-R8
0000                  EXC_MASK                        ;EXCEPTION MASK
0000          GSYSSRV DACEFC,K,1,-                    ;DISASSOCIATE EVENT FLAG CLUSTER
0000                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000          GSYSSRV DALLOC,K,2,-                    ;DEALLOCATE DEVICE
0000                  <R2,R3,R4,R5,R8>                ;REGISTERS R2-R5,R8
0000          GSYSSRV DASSGN,K,1,-                    ;DEASSIGN I/O CHANNEL
0000                  <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000          GSYSSRV DCLAST,K,3,-                    ;DECLARE AST SYSTEM SERVICE
```

```
0000  1270                  <R2,R3,R4,R5>                 ;REGISTERS R2-R5
0000  1271          GSYSSRV DCLEXH,K,1,-                  ;DECLARE EXIT HANDLER
0000  1272                  <R2,R3,R4>                    ;REGISTERS R2-R4
0000  1273          GSYSSRV DELLOG,ALL,3,-                ;DELETE LOGICAL NAME
0000  1274                  <R2,R3,R4,R5,R6,R7,R8>        ;REGISTERS R2-R8
0000  1275          GSYSSRV DELMBX,K,1,-                  ;DELETE MAILBOX
0000  1276                  <R2,R3,R4,R5>                 ;REGISTERS R2-R5
0000  1277          GSYSSRV DELPRC,K,2,-                  ;DELETE PROCESS
0000  1278                  <R2,R3,R4,R5,R6,R7>           ;REGISTERS R2-R5
0000  1279          GSYSSRV DELTVA,K,3,-                  ;DELETE VIRTUAL ADDRESS
0000  1280                  <R2,R3,R4,R5,R6,R7>,-         ;REGISTERS R2-R7
0000  1281                  EXC_MASK                      ;EXCEPTION MASK
0000  1282          GSYSSRV DGBLSC,K,3,-                  ;DELETE GLOBAL SECTION
0000  1283                  <R2,R3,R4,R5,R6,R7,R8,R9,R10> ;REGISTERS R2-R10
0000  1284          GSYSSRV DLCDNP,K,2,-                  ;DEALLOCATE DIAGNOSTIC PAGE
0000  1285                  <R2,R3,R4,R5,R6,R7>           ;REGISTERS R2-R7
0000  1286          GSYSSRV DLCEFC,K,1,-                  ;DELETE COMMON EVENT CLUSTER
0000  1287                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1288          GSYSSRV UPDSEC,K,8,-                  ;UPDATE SECTION FILE
0000  1289                  <R2,R3,R4,R5,R6,R7,R8>        ;R2-R8
0000  1290          GSYSSRV SNDERR,K,1,-                  ;SEND MSG TO ERROR LOGGER
0000  1291                  <R2,R3,R4,R5>                 ;REGISTERS R2-R5
0000  1292          GSYSSRV EXIT,K,1,-                    ;IMAGE EXIT
0000  1293                  <R4>,0                        ;REGISTER R4, ALWAYS ALLOWED!
0000  1294          GSYSSRV EXPREG,K,4,-                  ;EXPAND PROGRAM REGION
0000  1295                  <R2,R3,R4,R5,R6,R7,R8>        ;REGISTERS R2-R8
0000  1296          GSYSSRV FAO,ALL,0,-                   ;FORMAT ASCII OUTPUT
0000  1297                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000  1298          GSYSSRV FAOL,ALL,0,-                  ;FORMAT ASCII OUTPUT WITH VALUE LIST
0000  1299                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1300          GSYSSRV FORCEX,K,3,-                  ;FORCE EXIT
0000  1301                  <R2,R3,R4,R5>                 ;REGISTERS R2-R5
0000  1302          GSYSSRV IMGSTA,ALL,6,-                ;IMAGE STARTUP
0000  1303                  <>                            ;REGISTERS NONE
0000  1304          GSYSSRV SNDJBC,E,7,-                  ;SEND TO JOB CONTROLLER
0000  1305                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1306          GSYSSRV GETTIM,E,1,-                  ;GET TIME
0000  1307                  <>                            ;NO REGISTERS
0000  1308          GCOMPSRVB UPDSECW,-                   ;UPDATE SECTION AND WAIT
0000  1309                  <UPDSEC_MASK ! GETJPI_SYNCH_MASK>
0000  1317          GCOMPSRVE      1
0000  1318          GSYSSRV HIBER,K,0,-                   ;HIBERNATE
0000  1319                  <R2,R3,R4,R5>                 ;REGISTERS R2-R5
0000  1320          GSYSSRV IMGACT,E,8,-                  ;IMAGE ACTIVATION
0000  1321                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1322          GSYSSRV LCKPAG,K,3,-                  ;LOCK PAGE IN MEMORY
0000  1323                  <R2,R3,R4,R5,R6,R7,R8>        ;REGISTERS R2-R8
0000  1324          GSYSSRV LKWSET,K,3,-                  ;LOCK PAGES IN WORKING SET
0000  1325                  <R2,R3,R4,R5,R6,R7,R8>        ;REGISTERS R2-R8
0000  1326          GSYSSRV MGBLSC,K,7,-                  ;MAP GLOBAL SECTION
0000  1327                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1328          GSYSSRV PURGWS,K,1,-                  ;PURGE WORKING SET
0000  1329                  <R2,R3,R4,R5,R6,R7,R8>        ;R2-R8
0000  1330          GSYSSRV NUMTIM,E,2,-                  ;CONVERT TIME TO NUMERIC
0000  1331                  <R2,R3,R4,R5,R6,R7>           ;REGISTERS R2-R7
0000  1332          GSYSSRV SNDOPR,E,2,-                  ;SEND MSG TO OPERATOR
0000  1333                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>       ;REGISTERS R2-R11
```

```
0000        GSYSSRV QIO,K,12,-                  ;QUEUE I/O REQUEST
                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000        GSYSSRV READEF,K,2,-                ;READ EVENT FLAG
0000                <R2,R3,R4,R5>               ;REGISTERS R2-R5
0000        GSYSSRV RESUME,K,2,-                ;RESUME PROCESS
0000                <R2,R3,R4,R5>               ;REGISTERS R2-R5
0000        GSYSSRV RUNDWN,K,1,-                ;RUNDOWN
0000                <R2,R3,R4,R5,R6,R7>         ;REGISTERS R2-R7
0000        GSYSSRV SNDSMB,E,2,-                ;SEND MSG TO SYMBIONT MANAGER
0000                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>       ;REGISTERS R2-R11
0000        GSYSSRV SCHDWK,K,4,-                ;SCHEDULE WAKEUP
0000                <R2,R3,R4,R5,R6,R7,R8,R9> ;REGISTERS R2-R9
0000        GSYSSRV SETAST,K,1,-                ;SET AST ENABLE SERVICE
0000                <R2,R3,R4,R5>               ;REGISTERS R2-R5
0000        GSYSSRV SETEF,K,1,-                 ;SET EVENT FLAG
0000                <R2,R3,R4,R5>               ;REGISTERS R2-R5. SEE WAITFR COMMENTS.
0000        GSYSSRV SETEXV,K,4,-                ;SET EXCEPTION VECTOR
0000                <R2,R3,R4,R5>               ;REGISTERS R2-R5
0000        GSYSSRV SETPRN,K,1,-                ;SET PROCESS  NAME
0000                <R2,R3,R4,R5,R6,R7,R8,R9> ;REGISTERS R2-R9
0000        GSYSSRV SETPRA,K,2,-                ;SET POWER RECOVERY AST
0000                <R2,R3,R4,R5>               ;REGISTERS R2-R5
0000        GSYSSRV SETIMR,K,4,-                ;SET TIMER
0000                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000        GSYSSRV SETPRI,K,4,-                ;SET PROCESS PRIORITY
0000                <R2,R3,R4,R5>               ;REGISTERS R2-R5
0000        GSYSSRV SETPRT,K,5,-                ;SET PAGE PROTECTION
0000                <R2,R3,R4,R5,R6,R7,R8,R9> ;REGISTERS R2-R9
0000        GSYSSRV SETRWM,K,1,-                ;SET RESOURCE WAIT MODE
0000                <R4>                        ;REGISTER R4
0000        GSYSSRV SETSFM,K,1,-                ;SET SYSTEM SERVICE FAILURE MODE
0000                <R4>,EXC_MASK               ;REGISTER R4, AND EXECPTION MASK
0000        GSYSSRV SETSWM,K,1,-                ;SET PROCESS SWAP MODE
0000                <R4>                        ;REGISTER R4
0000        GSYSSRV SUSPND,K,2,-                ;SUSPEND PROCESS
0000                <R2,R3,R4,R5>               ;REGISTERS R2-R5
0000        GSYSSRV TRNLOG,ALL,6,-              ;TRANSLATE LOGICAL NAME
0000                <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
0000        GSYSSRV ULKPAG,K,3,-                ;UNLOCK PAGE FROM MEMORY
0000                <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
0000        GSYSSRV ULWSET,K,3,-                ;UNLOCK PAGES FROM WORKING SET
0000                <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
0000        GSYSSRV UNWIND,ALL,2,-              ;UNWIND PROCEDURE CALL STACK
0000                <R2,R3,R4,R5>               ;REGISTERS R2-R5
0000        GSYSSRV WAITFR,K,1,-                ;WAIT FOR EVENT FLAG
0000                <R2,R3,R4,R5,R6>            ;REGISTERS R2-R6.  IF R8 IS EVER USED
0000                                            ;THE RMS SYCHRONIZATION CODE MUST BE
0000                                            ;MODIFIED TO SAVE IT ALSO.
0000        GSYSSRV WAKE,K,2,-                  ;WAKE PROCESS
0000                <R2,R3,R4,R5>               ;REGISTERS R2-R5
0000        GSYSSRV WFLAND,K,2,-                ;WAIT FOR LOGICAL AND OF EVENT FLAGS
0000                <R2,R3,R4,R5,R6>            ;REGISTERS R2-R6
0000        GSYSSRV WFLOR,K,2,-                 ;WAIT FOR LOGICAL OR OF EVENT FLAGS
0000                <R2,R3,R4,R5,R6>            ;REGISTERS R2-R5
0000        GSYSSRV BRDCST,ALL,2,-              ;BROADCAST TO TERMINALS
0000                <R2,R3,R4,R5,R6>            ;REGISTERS R2-R6
0000        GSYSSRV DCLCMH,K,3,-                ;DECLARE CHANGE MODE HANDLER
```

```
0000  1391                  <R4>                            ;SAVE R4
0000  1392          GSYSSRV SETPFM,K,4,-                    ;SET PAGE FAULT MONITORING
0000  1393                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1394          GSYSSRV GETMSG,ALL,5,-                  ;GET MESSAGE
0000  1395                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1396          GSYSSRV DERLMB,K,1,-                    ;DECLARE ERROR LOG MAILBOX
0000  1397                  <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000  1398          GSYSSRV CANEXH,K,1,-                    ;CANCEL EXIT HANDLER
0000  1399                  <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000  1400          GSYSSRV GETCHN,K,5,-                    ;GET CHANNEL INFORMATION
0000  1401                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1402          GSYSSRV GETDEV,K,5,-                    ;GET DEVICE INFORMATION
0000  1403                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1404          GSYSSRV GETJPI,K,7,-                    ;GET JOB PROCESS INFORMATION
0000  1405                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1406          GSYSSRV PUTMSG,ALL,3,-                  ;PUT FORMATTED ERROR MESSAGE
0000  1407                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1408          GSYSSRV EXCMSG,ALL,2,-                  ;OUTPUT EXCEPTION SUMMARY MESSAGE
0000  1409                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1410          GSYSSRV SNDACC,E,2,-                    ;SEND MSG TO ACOUNTING MANAGER
0000  1411                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1412          GSYSSRV SETIME,K,1,-                    ;SET SYSTEM TIME
0000  1413                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1414          GSYSSRV SETPRV,K,4,-                    ;SET PRIVILEGES
0000  1415                  <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
```

```
0000  1417 ;
0000  1418 ;       SPECIAL VECTORS FOR AST DELIVERY AND CLEARING
0000  1419 ;
0000  1420 ;       SYS$CLRAST CLEARS THE CURRENTLY ACTIVE AST STATUS
0000  1421 ;
0000  1422 ;       SYS$GL_ASTRET CONTAINS THE VALUE OF THE RETURN ADDRESS FROM
0000  1423 ;       THE CALL INSTRUCTION USED TO DISPATCH AN AST. THIS VALUE CAN
0000  1424 ;       BE USED WHEN SEARCHING UP THE STACK FOR THE AST CALL FRAME.
0000  1425 ;
0000  1732
```

```
0000            .SBTTL  REGION 2 OF SYS. SERV. VECTOR DEFINITIONS
0000
0000    ;
0000    ; Note: Service codes for exec mode services in this region are
0000    ; reserved by the offset defined above between RCASCTR and ECASCTR.
0000    ; If the ASSUME at the end of this section breaks, the offset must
0000    ; be increased.
0000    ;
0000
0000            GSYSSRV ENQ,K,11,-                      ; ENQUEUE
                        <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000            GSYSSRV DEQ,K,4,-                       ; DEQUEUE
                        <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000            GCOMPSRVB ENQW,-                        ; ENQUEUE AND WAIT
                        <ENQ_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
0000            GCOMPSRVE       3                       ; RESERVE 3 QUADWORDS FOR VECTOR
0000            GSYSSRV SETSSF,K,1,-                    ; SET SYSTEM SERVICE FILTER MASK
                        <R4>                            ; REGISTER R4
0000            GSYSSRV SETSTK,K,3,-                    ; SET STACK LIMITS
                        <R2,R3,R4>                      ; REGISTERS R2,R3,R4
0000            GSYSSRV GETSYI,K,7,-                    ; GET SYSTEM INFORMATION
                        <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000            GSYSSRV IMGFIX,ALL,0,-                  ; IMAGE ADDRESS RELOCATION FIXUP
                        <R2,R3,R4,R5>                   ; REGISTERS R2-R5
0000            GCOMPSRVB       IMGFIX_2,-              ; ********** TEMP **********
                        <0>
0000            GCOMPSRVE       1                       ; ********** TEMP **********
0000            GSYSSRV GETDVI,K,8,-                    ; GET DEVICE AND VOLUME INFORMATION
                        <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000            GCOMPSRVB GETDVIW,-                     ; GET DEVICE INFORMATION AND WAIT
                        <GETDVI_MASK ! GETJPI_SYNCH_MASK>
0000            GCOMPSRVE       1
0000            GCOMPSRVB GETJPIW,-                     ; GET JOB/PROCESS INFORMATION AND WAIT
                        <GETJPI_MASK ! GETJPI_SYNCH_MASK>
0000            GCOMPSRVE       2
0000            GCOMPSRVB GETSYIW,-                     ; GET SYSTEM INFORMATION AND WAIT
                        <GETSYI_MASK ! GETJPI_SYNCH_MASK>
0000            GCOMPSRVE       1
0000            GCOMPSRVB SNDJBCW,-                     ; SEND TO JOB CONTROLLER AND WAIT
                        <SNDJBC_MASK ! GETJPI_SYNCH_MASK>
0000            GCOMPSRVE       1
0000            GCOMPSRVB SYNCH,-                       ; SYNCHRONIZE EFN AND IOSB
                        <WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
0000            GCOMPSRVE       6                       ; RESERVE 6 QUADWORDS FOR VECTOR
0000            GSYSSRV ERAPAT,K,3,-                    ; GENERATE A SECURITY  ERASE PATTERN
                        <R4>                            ; SAVE R4
0000            GSYSSRV CRELNT,K,8,-                    ; CREATE LOGICAL NAME TABLE
                        <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000            GSYSSRV CRELNM,K,5,-                    ; CREATE LOGICAL NAME
                        <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000            GSYSSRV DELLNM,K,3,-                    ; DELETE LOGICAL NAME
                        <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000            GSYSSRV TRNLNM,K,5,-                    ; TRANSLATE LOGICAL NAME
                        <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000            GSYSSRV GETLKI,K,7,-                    ; GET LOCK INFORMATION
                        <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000            GCOMPSRVB GETLKIW,-                     ; GET LOCK INFORMATION AND WAIT
```

```
          0000              <GETLKI_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
          0000      GCOMPSRVE       2                   ; RESERVE 2 QUADWORDS FOR VECTOR
          0000
          0000      GSYSSRV ASCTOID,E,3,-               ;ASCII TO IDENTIFIER CONVERSION
          0000              <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
          0000      GSYSSRV FINISH_RDB,E,1,-            ;FINISH RDB CONTEXT STREAM
          0000              <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
          0000      GSYSSRV IDTOASC,E,6,-               ;IDENTIFIER TO ASCII CONVERSION
          0000              <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
          0000      GSYSSRV BRKTHRU,K,11,-              ;BREAK THROUGH WRITES
          0000              <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
          0000      GSYSSRV GRANTID,ALL,5,-             ;GRANT IDENTIFIER TO PROCESS
          0000              <R2,R3>                     ;REGISTERS R2-R3
          0000      GSYSSRV REVOKID,ALL,5,-             ;REVOKE IDENTIFIER FROM PROCESS
          0000              <R2,R3>                     ;REGISTERS R2-R3
          0000      GSYSSRV CHKPRO,K,1,-                ;GENERAL PROTECTION CHECK ROUTINE
          0000              <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
          0000      GCOMPSRVB BRKTHRUW,-                ; BREAK THOUGH WRITE AND WAIT
          0000              <BRKTHRU_MASK ! GETJPI_SYNCH_MASK>
          0000      GCOMPSRVE       2
          0000      GSYSSRV GETQUI,E,7,-                ;GET QUEUE INFORMATION
          0000              <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
          0000      GCOMPSRVB GETQUIW,-                 ;GET QUEUE INFORMATION AND WAIT
          0000              <GETQUI_MASK ! GETJPI_SYNCH_MASK>
          0000      GCOMPSRVE       2
          0000
          0000  ;
00004028  0000      CJF$KCASCTR = 16424
          0000  ;
          0000      LDBSRV  CJF$, ALLJDR,       K,    <R4>
          0000      LDBSRV  CJF$, ASSJNL,       K,    <R4>
          0000      LDBSRV  CJF$, CONUIC,       K,    <R4>
          0000      LDBSRV  CJF$, CREJNL,       K,    <R4>
          0000      LDBSRV  CJF$, DEALJDR,      K,    <R4>
          0000      LDBSRV  CJF$, DEASJNL,      ALL,  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
          0000      LDBSRV  CJF$, DEASJNL_INT,  K,    <R4>
          0000      LDBSRV  CJF$, DELJNL,       K,    <R4>
          0000      LDBSRV  CJF$, DMTJMD,       K,    <R4>
          0000      LDBSRV  CJF$, DSPJNL,       K,    <R4>
          0000      LDBSRV  CJF$, GETJNL,       K,    <R4>
          0000      LDBSRV  CJF$, GETRUI,       K,    <R4>
          0000      LDBSRV  CJF$, MODFLT,       K,    <R4>
          0000      LDBSRV  CJF$, POSJNL,       K,    <R4>
          0000      LDBSRV  CJF$, READJNL,      K,    <R4>
          0000      LDBSRV  CJF$, RECOVER,      K,    <R4>
          0000      LDBSRV  CJF$, MNTJMD,       K,    <R4>
          0000      LDBSRV  CJF$, CRENWV,       K,    <R4>
          0000      LDBSRV  CJF$, CONJNLF,      K,    <R4>
          0000      LDBSRV  CJF$, DCNJNLF,      K,    <R4>
          0000      LDBSRV  CJF$, FORCEJNL,     ALL,  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
          0000      LDBSRV  CJF$, FORCEJNLW,    ALL,  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
          0000      LDBSRV  CJF$, WRITEJNL,     ALL,  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
          0000      LDBSRV  CJF$, WRITEJNLW,    ALL,  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
          0000      LDBSRV  CJF$, GETCJI,       K,    <R4>
          0000      LDBSRV  CJF$, DMTJMDW,      K,    <R4>, 4, 5, DMTJMD
          0000      LDBSRV  CJF$, MODFLTW,      K,    <R4>, 4, 5, MODFLT
          0000      LDBSRV  CJF$, POSJNLW,      K,    <R4>, 4, 5, POSJNL
          0000        LDBSRV  CJF$, READJNLW,    K,   <R4>, 4, 5, READJNL
          0000        LDBSRV  CJF$, RECOVERW,    K,   <R4>, 5, 6, RECOVER
          0000
          0000  ;
00004010  0000        RUF$KCASCTR = 16400
          0000  ;
          0000        LDBSRV  RUF$,  REENTERRU,    K,  <R2,R3,R4,R5,R6>
          0000        LDBSRV  RUF$,  STARTRU,      K,  <R2,R3,R4,R5,R6>
          0000        LDBSRV  RUF$,  PHASE1,       K,  <R2,R3,R4,R5,R6>
          0000        LDBSRV  RUF$,  PHASE2,       K,  <R2,R3,R4,R5,R6>
          0000        LDBSRV  RUF$,  CANCELRU,     K,  <R2,R3,R4,R5,R6>
          0000        LDBSRV  RUF$,  MARKPOINTRU,  K,  <R2,R3,R4,R5,R6>
          0000        LDBSRV  RUF$,  RESETRU,      K,  <R2,R3,R4,R5,R6>
          0000        LDBSRV  RUF$,  DCLRUH,       K,  <R2,R3,R4,R5,R6>
          0000        LDBSRV  RUF$,  CANRUH,       K,  <R2,R3,R4,R5,R6>
          0000        LDBSRV  RUF$,  RUSTATUS,     K,  <R2,R3,R4,R5,R6>
          0000  ;
          0000  ; End Recovery Unit consists of a two-phase commit, so we call each
          0000  ; phase separately.
          0000  ;
          0000        GCOMPSRVB ENDRU, <PHASE1_MASK ! PHASE2_MASK>, RUF$ ; End Recovery Unit
          0000        GCOMPSRVE       2
          0000        GSYSSRV MTACCESS,K,6,-          ;Mag tape installation specific access routi
                              <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
          0000
          0000  ;
          0000  ; End of system service vector definitions. New system services are
          0000  ; to be added at this point.
          0000  ;
          0000
```

```
                                00000053  2167              .PSECT  MP$CMOD2,BYTE
                                    0053  2169 SSFAILMAIN:                                   ;SSFAIL MAIN LOGIC
          51   00000000'GF     D0   0053  2170              MOVL    G^CTL$GL_PCB,R1          ;GET PCB ADDRESS
                       0E A1   B5   005A  2171              TSTW    PCB$W_MTXCNT(R1)         ;MUTEX COUNT ZERO?
                          47   12   005D  2172              BNEQ    20$                      ;IF NEQ NO
                  02      18   EF   005F  2173              EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,- ;EXTRACT PREVIOUS MODE FROM
                7E     04 AE        0062  2174                      4(SP),-(SP)              ;SAVED PSL
                  6E      06   C0   0065  2175              ADDL    #PCB$V_SSFEXC,(SP)       ;ADD IN BASE BIT NUMBER
              ° 24 A1     8E   E1   0068  2176              BBC     (SP)+,PCB$L_STS(R1),10$ ;IF CLEAR, FAILURE EXCEPTION DISABLED
                          7E   DC   006D  2177              MOVPSL  -(SP)                    ;GET CURRENT PSL
       8E     6E    02    18   EF   006F  2178              EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,(SP),(SP)+ ;IF CURRENT MODE IS
                          03   12   0074  2179              BNEQ    5$                       ;NOT KERNEL, THEN BRANCH
                                    0076  2180              SETIPL  #0                       ;FORCE IPL TO 0 FOR ERROR PATH
                                    0079  2190 5$:          IFPRIMARY <JMP G^EXE$SSFAIL>     ;IF PRIMARY, THEN CONTINUE RIGHT ALONG
                                    0092  2191                                               ;IF SECONDARY, RETURN PROCESS TO PRIMARY
    7E    04 AE     02    18   EF   0092  2192              EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,4(SP),-(SP) ;CREATE PSL WITH PREV
              6E    6E    16   9C   0098  2193              ROTL    #PSL$V_PRVMOD,(SP),(SP) ; MODE CORRECT AND CURRENT MODE = KERNEL
               00000000'GF     9F   009C  2194              PUSHAB  G^EXE$SSFAIL             ;REFLECT THE EXCEPTION
                       FF5B'   31   00A2  2195              BRW     MPS$MPSCHED2             ; AND RETURN PROCESS TO PRIMARY
                               02   00A5  2196 10$:         REI                              ;RETURN FROM SERVICE WITH ERROR STATUS
                                    00A6  2197 20$:         IFPRIMARY <BUG_CHECK MTXCNTNONZ,FATAL>  ;PRIMARY VERSION OF BUGCHECK
                                    00BD  2198              SECBUG_CHECK MTXCNTNONZ,FATAL    ;MUTEX COUNT NONZERO AT SERVICE EXIT
                      00000055      00C2  2265 KCASMAX=KCASCTR-2
                                    00C2  2266
                                    00C2  2269
```

```
         00C2  234           .END
```

```
$$ARGS                          = 00000008                 CMK$C_DELTVA                    = 0000001A  G
$$T1                            = 00000024                 CMK$C_DEQ                       = 00000049  G
ACB$B_RMOD                      = 0000000B                 CMK$C_DERLMB                    = 00000041  G
ACB$V_KAST                      = 00000007                 CMK$C_DGBLSC                    = 0000001B  G
ACCVIO                            00000030 R      02       CMK$C_DLCDNP                    = 0000001C  G
ACCVIO1                           0000010D R      02       CMK$C_DLCEFC                    = 0000001D  G
ADJSTK                          = 00000001                 CMK$C_ENQ                       = 00000048  G
ADJWSL                          = 00000002                 CMK$C_ERAPAT                    = 0000004E  G
ALCDNP                          = 00000003                 CMK$C_EXIT                      = 00000020  G
ALLOC                           = 00000004                 CMK$C_EXPREG                    = 00000021  G
ASCEFC                          = 00000005                 CMK$C_FORCEX                    = 00000022  G
ASSIGN                          = 00000006                 CMK$C_GETCHN                    = 00000043  G
ASTEXIT                           0000010B R      02       CMK$C_GETDEV                    = 00000044  G
BRKTHRU                         = 00000054                 CMK$C_GETDVI                    = 0000004D  G
BUG$_MTXCNTNONZ                   ********   X    03       CMK$C_GETJPI                    = 00000045  G
BUG$_SSRVEXCEPT                   ********   X    02       CMK$C_GETLKI                    = 00000053  G
CANCEL                          = 00000007                 CMK$C_GETPTI                    = 0000000F  G
CANEXH                          = 00000042                 CMK$C_GETSYI                    = 0000004C  G
CANTIM                          = 00000008                 CMK$C_HIBER                     = 00000023  G
CANWAK                          = 00000009                 CMK$C_LCKPAG                    = 00000024  G
CAT0                            = 00000001                 CMK$C_LKWSET                    = 00000025  G
CAT7                            = 00000080                 CMK$C_MGBLSC                    = 00000026  G
CHKPRO                          = 00000055                 CMK$C_MTACCESS                  = 00000056  G
CJF$KCASCTR                     = 00004028                 CMK$C_PURGWS                    = 00000027  G
CLREF                           = 0000000D                 CMK$C_QIO                       = 00000028  G
CLRPAR                          = 0000000B                 CMK$C_READEF                    = 00000029  G
CMK$C_ADJSTK                    = 00000001  G              CMK$C_RESUME                    = 0000002A  G
CMK$C_ADJWSL                    = 00000002  G              CMK$C_RUNDWN                    = 0000002B  G
CMK$C_ALCDNP                    = 00000003  G              CMK$C_SCHDWK                    = 0000002C  G
CMK$C_ALLOC                     = 00000004  G              CMK$C_SETAST                    = 0000002D  G
CMK$C_ASCEFC                    = 00000005  G              CMK$C_SETEF                     = 0000002E  G
CMK$C_ASSIGN                    = 00000006  G              CMK$C_SETEXV                    = 0000002F  G
CMK$C_BRKTHRU                   = 00000054  G              CMK$C_SETIME                    = 00000046  G
CMK$C_CANCEL                    = 00000007  G              CMK$C_SETIMR                    = 00000032  G
CMK$C_CANEXH                    = 00000042  G              CMK$C_SETPFM                    = 00000040  G
CMK$C_CANTIM                    = 00000008  G              CMK$C_SETPRA                    = 00000031  G
CMK$C_CANWAK                    = 00000009  G              CMK$C_SETPRI                    = 00000033  G
CMK$C_CHKPRO                    = 00000055  G              CMK$C_SETPRN                    = 00000030  G
CMK$C_CLREF                     = 0000000D  G              CMK$C_SETPRT                    = 00000034  G
CMK$C_CLRPAR                    = 0000000B  G              CMK$C_SETPRV                    = 00000047  G
CMK$C_CMKRNL                    = 0000000C  G              CMK$C_SETRWM                    = 00000035  G
CMK$C_CNTREG                    = 0000000E  G              CMK$C_SETSFM                    = 00000036  G
CMK$C_CRELNM                    = 00000050  G              CMK$C_SETSSF                    = 0000004A  G
CMK$C_CRELNT                    = 0000004F  G              CMK$C_SETSTK                    = 0000004B  G
CMK$C_CREMBX                    = 00000010  G              CMK$C_SETSWM                    = 00000037  G
CMK$C_CREPRC                    = 00000011  G              CMK$C_SNDERR                    = 0000001F  G
CMK$C_CRETVA                    = 00000012  G              CMK$C_SUSPND                    = 00000038  G
CMK$C_CRMPSC                    = 0000000A  G              CMK$C_TRNLNM                    = 00000052  G
CMK$C_DACEFC                    = 00000013  G              CMK$C_ULKPAG                    = 00000039  G
CMK$C_DALLOC                    = 00000014  G              CMK$C_ULWSET                    = 0000003A  G
CMK$C_DASSGN                    = 00000015  G              CMK$C_UPDSEC                    = 0000001E  G
CMK$C_DCLAST                    = 00000016  G              CMK$C_WAITFR                    = 0000003B  G
CMK$C_DCLCMH                    = 0000003F  G              CMK$C_WAKE                      = 0000003C  G
CMK$C_DCLEXH                    = 00000017  G              CMK$C_WFLAND                    = 0000003D  G
CMK$C_DELLNM                    = 00000051  G              CMK$C_WFLOR                     = 0000003E  G
CMK$C_DELMBX                    = 00000018  G              CMKRNL                          = 0000000C
CMK$C_DELPRC                    = 00000019  G              CNTREG                          = 0000000E
```

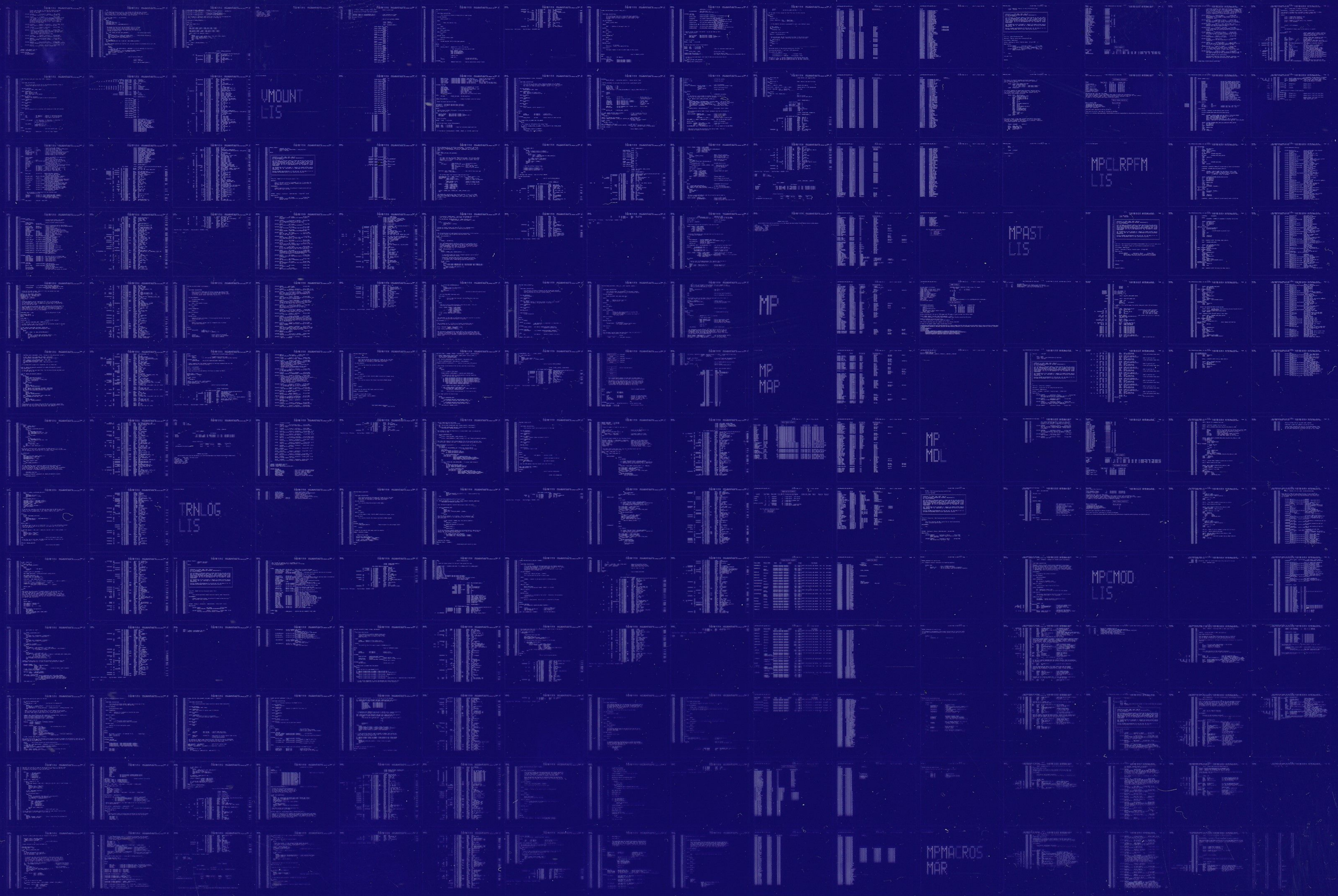
```
CRELNM                          = 00000050                GETDVI$_NULLARG                 = 00000020
CRELNT                          = 0000004F                GETJPI                          = 00000045
CREMBX                          = 00000010                GETJPI$_ASTADR                  = 00000018
CREPRC                          = 00000011                GETJPI$_ASTPRM                  = 0000001C
CRETVA                          = 00000012                GETJPI$_EFN                     = 00000004
CRMPSC                          = 0000000A                GETJPI$_IOSB                    = 00000014
CTL$GB_SSFILTER                   ********   X   02       GETJPI$_ITMLST                  = 00000010
CTL$GL_PCB                        ********   X   02       GETJPI$_NARGS                   = 00000007
DACEFC                          = 00000013                GETJPI$_PIDADR                  = 00000008
DALLOC                          = 00000014                GETJPI$_PRCNAM                  = 0000000C
DASSGN                          = 00000015                GETLKI                          = 00000053
DCLAST                          = 00000016                GETLKI$_ASTADR                  = 00000014
DCLCMH                          = 0000003F                GETLKI$_ASTPRM                  = 00000018
DCLEXH                          = 00000017                GETLKI$_EFN                     = 00000004
DEF_MASK                        = 00000081                GETLKI$_IOSB                    = 00000010
DELLNM                          = 00000051                GETLKI$_ITMLST                  = 0000000C
DELMBX                          = 00000018                GETLKI$_LKIDADR                 = 00000008
DELPRC                          = 00000019                GETLKI$_NARGS                   = 00000007
DELTVA                          = 0000001A                GETLKI$_RESERVED                = 0000001C
DEQ                             = 00000049                GETPTI                          = 0000000F
DERLMB                          = 00000041                GETSYI                          = 0000004C
DGBLSC                          = 0000001B                GETSYI$_ASTADR                  = 00000018
DLCDNP                          = 0000001C                GETSYI$_ASTPRM                  = 0000001C
DLCEFC                          = 0000001D                GETSYI$_CSIDADR                 = 00000008
ENQ                             = 00000048                GETSYI$_EFN                     = 00000004
ENQ$_ACMODE                     = 00000028                GETSYI$_IOSB                    = 00000014
ENQ$_ASTADR                     = 0000001C                GETSYI$_ITMLST                  = 00000010
ENQ$_ASTPRM                     = 00000020                GETSYI$_NARGS                   = 00000007
ENQ$_BLKAST                     = 00000024                GETSYI$_NODENAME                = 0000000C
ENQ$_EFN                        = 00000004                HIBER                           = 00000023
ENQ$_FLAGS                      = 00000010                INHEXCP                           00000000 R      02
ENQ$_LKMODE                     = 00000008                IPL$_SYNCH                      = 00000008
ENQ$_LKSB                       = 0000000C                KCASCTR                         = 00000057
ENQ$_NARGS                      = 0000000B                KCASE                             0000011C R      02
ENQ$_PARID                      = 00000018                KCASMAX                         = 00000055
ENQ$_PROT                       = 0000002C                KERDSP                            000000B7 R      02
ENQ$_RESNAM                     = 00000014                KINSARG                           0000003E R      02
ERAPAT                          = 0000004E                KINSARG1                          00000110 R      02
EXC_MASK                        = 00000080                LCK$V_INTERLOCK                 = 00000000
EXE$CMODKRNL                      ********   X   02       LCKPAG                          = 00000024
EXE$GL_RPB                        ********   X   02       LKWSET                          = 00000025
EXE$REFLECT                       ********   X   02       MGBLSC                          = 00000026
EXE$SSFAIL                        ********   X   03       MPS$ASTEXIT                       00000000 R      03
EXIT                            = 00000020                MPS$CMODKRNL                      00000088 RG     02
EXPREG                          = 00000021                MPS$CMODKRNLX                     00000060 RG     02
FORCEX                          = 00000022                MPS$EXCPTN                        0000004F RG     02
GETCHN                          = 00000043                MPS$GL_CURPCB                     ********   X    03
GETDEV                          = 00000044                MPS$GL_INTERLOCK                  ********   X    03
GETDVI                          = 0000004D                MPS$MPSCHED2                      ********   X    02
GETDVI$_ASTADR                  = 00000018                MPS$SECBUGCHK                     ********   X    02
GETDVI$_ASTPRM                  = 0000001C                MPS$WAITFR                        ********   X    02
GETDVI$_CHAN                    = 00000008                MPS$WAITFR1                       00000113 R      02
GETDVI$_DEVNAM                  = 0000000C                MPS$WFLAND                        ********   X    02
GETDVI$_EFN                     = 00000004                MPS$WFLAND1                       00000116 R      02
GETDVI$_IOSB                    = 00000014                MPS$WFLOR                         ********   X    02
GETDVI$_ITMLST                  = 00000010                MPS$WFLOR1                        00000119 R      02
GETDVI$_NARGS                   = 00000008                MPSWITCH                        = 00000001
```

```
MTACCESS                       = 00000056            SNDJBC$_IOSB                   = 00000014
PCB$B_ASTACT                   = 0000000C            SNDJBC$_ITMLST                 = 00000010
PCB$L_ASTQFL                   = 00000010            SNDJBC$_NARGS                  = 00000007
PCB$L_PHD                      = 0000006C            SNDJBC$_NULLARG                = 0000000C
PCB$L_STS                      = 00000024            SRVEXIT                          0000004B R      02
PCB$V_SSFEXC                   = 00000006            SRVREI                           0000004E R      02
PCB$W_MTXCNT                   = 0000000E            SS$_ACCVIO                     = 0000000C
PHD$B_ASTLVL                   = 000000CF            SS$_INHCHMK                    = 000004CC
PR$_ASTLVL                     = 00000013            SS$_INSFARG                    = 00000114
PR$_IPL                        = 00000012            SSFAIL                           00000056 R      02
PR$_SCBB                       = 00000011            SSFAILMAIN                       00000053 R      03
PSL$M_CURMOD                   = 03000000            SUSPND                         = 00000038
PSL$S_CURMOD                   = 00000002            SYNCH$_EFN                     = 00000004
PSL$V_CURMOD                   = 00000018            SYNCH$_IOSB                    = 00000008
PSL$V_PRVMOD                   = 00000016            SYNCH$_NARGS                   = 00000002
PURGWS                         = 00000027            SYS$GB_KMASK                     ********    X   02
QIO                            = 00000028            SYS$GB_KRNLNARG                  ********    X   02
QIO$_ASTADR                    = 00000014            TRNLNM                         = 00000052
QIO$_ASTPRM                    = 00000018            ULKPAG                         = 00000039
QIO$_CHAN                      = 00000008            ULWSET                         = 0000003A
QIO$_EFN                       = 00000004            UPDSEC                         = 0000001E
QIO$_FUNC                      = 0000000C            UPDSEC$_ACMODE                 = 0000000C
QIO$_IOSB                      = 00000010            UPDSEC$_ASTADR                 = 0000001C
QIO$_NARGS                     = 0000000C            UPDSEC$_ASTPRM                 = 00000020
QIO$_P1                        = 0000001C            UPDSEC$_EFN                    = 00000014
QIO$_P2                        = 00000020            UPDSEC$_INADR                  = 00000004
QIO$_P3                        = 00000024            UPDSEC$_IOSB                   = 00000018
QIO$_P4                        = 00000028            UPDSEC$_NARGS                  = 00000008
QIO$_P5                        = 0000002C            UPDSEC$_RETADR                 = 00000008
QIO$_P6                        = 00000030            UPDSEC$_UPDFLG                 = 00000010
READEF                         = 00000029            WAITFR                         = 0000003B
RESUME                         = 0000002A            WAKE                           = 0000003C
RPB$L_SCBB                     = 000000B0            WFLAND                         = 0000003D
RUF$KCASCTR                    = 00004010            WFLOR                          = 0000003E
RUNDWN                         = 0000002B
SCHDWK                         = 0000002C
SETAST                         = 0000002D
SETEF                          = 0000002E
SETEXV                         = 0000002F
SETIME                         = 00000046
SETIMR                         = 00000032
SETPFM                         = 00000040
SETPRA                         = 00000031
SETPRI                         = 00000033
SETPRN                         = 00000030
SETPRT                         = 00000034
SETPRV                         = 00000047
SETRWM                         = 00000035
SETSFM                         = 00000036
SETSSF                         = 0000004A
SETSTK                         = 0000004B
SETSWM                         = 00000037
SNDERR                         = 0000001F
SNDJBC$_ASTADR                 = 00000018
SNDJBC$_ASTPRM                 = 0000001C
SNDJBC$_EFN                    = 00000004
SNDJBC$_FUNC                   = 00000008
```

```
                                                     +----------------+
                                                     ! Psect synopsis !
                                                     +----------------+

PSECT name                       Allocation          PSECT No.  Attributes
----------                       ----------          ---------  ----------
.  ABS  .                        00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                            00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
MP$CMOD1                         00000120  (  288.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC QUAD
MP$CMOD2                         000000C2  (  194.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$$000                           00000000  (    0.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC QUAD

                                                 +------------------------+
                                                 ! Performance indicators !
                                                 +------------------------+

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                      29    00:00:00.07    00:00:00.82
Command processing                 157    00:00:01.25    00:00:07.91
Pass 1                             668    00:00:23.94    00:00:58.52
Symbol table sort                    0    00:00:02.07    00:00:03.17
Pass 2                             225    00:00:06.60    00:00:20.10
Symbol table output                 38    00:00:00.29    00:00:01.29
Psect synopsis output                2    00:00:00.03    00:00:00.03
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals              1121    00:00:34.25    00:01:31.85

The working set limit was 2250 pages.
210745 bytes (412 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1356 non-local and 11 local symbols.
2351 source lines were read in Pass 1, producing 23 object records in Pass 2.
51 pages of virtual memory were used to define 47 macros.

                                                +--------------------------+
                                                ! Macro library statistics !
                                                +--------------------------+

Macro library name                               Macros defined
------------------                               --------------
_$255$DUA28:[MP.OBJ]MP.MLB;1                                  8
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                                9
_$255$DUA28:[SYSLIB]STARLET.MLB;2                            19
TOTALS (all libraries)                                       36

1362 GETS were required to define 36 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:MPCMOD/OBJ=OBJ$:MPCMOD MSRC$:MPPREFIX/UPDATE=(ENH$:MPPREFIX)+MSRC$:MPSWT/UPDATE=(ENH$:MPSWT)+MASD$:[SYS.SRC]CMODSSDSP
```

0247 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0248 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MPERRLOG
LIS

MPSCBVEC
LIS

MPINT
LIS

MPPFM
LIS

MPDAT
LIS

MPPWRFAIL
LIS

MPMCHECK
LIS

MPINTEXC
LIS

MPLOG
LIS

MPERRMSG
LIS

MPSCHED
LIS

MPSHWPFM
LIS

MPLOAD
LIS

MPINIT
LIS