L 12

```
   SSSSSSSSSSSS  YYY         YYY      SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY      SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY      SSSSSSSSSSSS
SSS              YYY         YYY   SSS
SSS              YYY         YYY   SSS
SSS              YYY         YYY   SSS
SSS                 YYY   YYY      SSS
SSS                 YYY   YYY      SSS
SSS                 YYY   YYY      SSS
   SSSSSSSSS           YYY            SSSSSSSSS
   SSSSSSSSS           YYY            SSSSSSSSS
   SSSSSSSSS           YYY            SSSSSSSSS
            SSS        YYY                     SSS
            SSS        YYY                     SSS
            SSS        YYY                     SSS
            SSS        YYY                     SSS
            SSS        YYY                     SSS
            SSS        YYY                     SSS
SSSSSSSSSSSS           YYY         SSSSSSSSSSSS
SSSSSSSSSSSS           YYY         SSSSSSSSSSSS
SSSSSSSSSSSS           YYY         SSSSSSSSSSSS
```

```
_$

Ps
--
YZ

Z$

Z$

Z$

Z$

Z$

Z$
Z$
Z$
Z$
Z$
Z$

Z$
```

```
**FILE**ID**P1SYSVECT                                J 2
```

```
PPPPPPPP        11        SSSSSSSS  YY      YY    SSSSSSSS  VV      VV  EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT
PPPPPPPP        11        SSSSSSSS  YY      YY    SSSSSSSS  VV      VV  EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT
PP      PP    1111      SS          YY      YY  SS          VV      VV  EE          CC              TT
PP      PP    1111      SS          YY      YY  SS          VV      VV  EE          CC              TT
PP      PP      11      SS            YY  YY    SS          VV      VV  EE          CC              TT
PP      PP      11      SS            YY  YY    SS          VV      VV  EE          CC              TT
PPPPPPPP        11        SSSSSS        YY        SSSSSS    VV      VV  EEEEEEEE    CC              TT
PPPPPPPP        11        SSSSSS        YY        SSSSSS    VV      VV  EEEEEEEE    CC              TT
PP              11              SS      YY              SS  VV      VV  EE          CC              TT
PP              11              SS      YY              SS  VV      VV  EE          CC              TT
PP              11              SS      YY              SS    VV  VV    EE          CC              TT          ....
PP              11              SS      YY              SS    VV  VV    EE          CC              TT          ....
PP            111111    SSSSSSSS        YY      SSSSSSSS        VV      EEEEEEEEEE    CCCCCCCC      TT          ....
PP            111111    SSSSSSSS        YY      SSSSSSSS        VV      EEEEEEEEEE    CCCCCCCC      TT          ....

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

```
00000001  0000   LIBSWITCH=1                             ;GENERATE LIBRARY FORM OF SERVICE VECTOR
00000001  0000   P1VSWITCH=1                             ;GENERATE P1 SPACE VECTORS
          0000           .NLIST  CND
          0000           .TITLE  SYS$P1_VECTOR - P1 SYSTEM SERVICE VECTOR DEFINITIONS
          0000           .IDENT  'V04-000'
          0000
          0000   ;
          0000   ;****************************************************************************
          0000   ;*                                                                          *
          0000   ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
          0000   ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
          0000   ;*  ALL RIGHTS RESERVED.                                                    *
          0000   ;*                                                                          *
          0000   ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
          0000   ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
          0000   ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
          0000   ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
          0000   ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
          0000   ;*  TRANSFERRED.                                                            *
          0000   ;*                                                                          *
          0000   ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
          0000   ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
          0000   ;*  CORPORATION.                                                            *
          0000   ;*                                                                          *
          0000   ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
          0000   ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
          0000   ;*                                                                          *
          0000   ;*                                                                          *
          0000   ;****************************************************************************
          0000   ;
          0000   ; D. N. CUTLER 22-JUN-76
          0000   ;
          0000   ;  MODIFIED BY:
          0000   ;
          0000   ;       V03-041 LJK0287         Lawrence J. Kenah       27-Jun-1984
          0000   ;               Add R5 to entry mask for $CANEXH system service.
          0000   ;
          0000   ;       V03-040 LMP0239         L. Mark Pilant,         23-Apr-1984  9:21
          0000   ;               Change $CHKPRO from an exec mode service to a kernel mode
          0000   ;               service.  This was made necessary by the $CHKPRO (internal
          0000   ;               entry point) interface change.
          0000   ;
          0000   ;       V03-039 MMD0250         Meg Dumont,     27-Feb-1984  17:49
          0000   ;               Add support for $MTACCESS installation specific accessibility
          0000   ;               routine
          0000   ;
          0000   ;       V03-038 DAS0001         David Solomon           20-Feb-1984
          0000   ;               Implement new design for RMS echo SYS$INPUT to SYS$OUTPUT
          0000   ;               (vs V03-019). Echo is now performed by a caller's mode AST
          0000   ;               routine declared in RMS\RM$EXRMS. Change INCB/DECB of FAB/RAB
          0000   ;               busy bit to BISB/BICB, now that we have room.
          0000   ;
          0000   ;       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  ;               Addeu $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$CMOD1 immediately after executive mode code so that new
0000  ;               code can be inserted in a way that preserves routine headers,
0000  ;               conditional assembly, and .PSECT definitions.  Backout ROW145,
0000  ;               and in its place, correct conditional assembly of BGEQU 10$
0000  ;               after ACCVIO_RET so that it is assembled only for MPCMOD and
0000  ;               so that it is located before ACCVIO_RET.  Change PCB address
```

B 3

```
0000  ;                 lookup at KERDSP in MPCMOD to use CTL$GL_PCB so that it works
0000  ;                 correctly regardless of which processor executes it.
0000  ;
0000  ;         V03-011 ROW0145         Ralph O. Weber          29-NOV-1982
0000  ;                 Move EXE$EXCPTN (and MPS$EXCPTN) to before ASTEXIT (or
0000  ;                 MPS$ASTEXIT) in an attempt to make branch destinations in
0000  ;                 EXE$CMODKRNL reach.
0000  ;
0000  ;         V03-010 KDM0030         Kathleen D. Morse       18-Nov-1982
0000  ;                 Add logic to MPCMOD that allows the primary to execute
0000  ;                 secondary-specific code  without turning into a secondary.
0000  ;
0000  ;         V03-009 MLJ0099         Martin L. Jack, 20-Oct-1982  19:42
0000  ;                 Complete V03-002 by correcting mode and argument count of
0000  ;                 $SNDJBC and removing temporary stubs.
0000  ;
0000  ;         V03-008 RIH0001         Richard I. Hustvedt     1-Jun-1982
0000  ;                 Correct handling of AST queue by secondary processor to
0000  ;                 avoid losing some AST notifications by incorrectly computing
0000  ;                 PHD$B_ASTLVL.
0000  ;
0000  ;         V03-007 KDM0018         Kathleen D. Morse       30-Sep-1982
0000  ;                 Add MPSWITCH logic to create a kernel system service
0000  ;                 dispatcher for the secondary processor of an 11/782.
0000  ;
0000  ;         V03-006 STJ3028         Steven T. Jeffreys      26-Sep-1982
0000  ;                 Added $ERAPAT system service vector.
0000  ;
0000  ;         V03-005 DWT0058         David Thiel             11-Aug-1982
0000  ;                 Eliminate use of R2 while waiting for service
0000  ;                 completion.
0000  ;
0000  ;         V03-004 JWH0001         Jeffrey W. Horn         26-Jul-1982
0000  ;                 Add new RMS service, RMSRUHNDLR, an un-documented service
0000  ;                 which acts as the Recovery Unit handler for RMS.
0000  ;
0000  ;         V03-003 PHL0102         Peter H. Lipman         16-Jul-1982
0000  ;                 Fix new SYNCH logic to always return SS$_NORMAL,
0000  ;                 not access IOSB if error from service, and return
0000  ;                 error status from $SETEF if event flag cluster went away
0000  ;
0000  ;         V03-002 PHL0101         Peter H. Lipman         17-Jun-1982
0000  ;                 Add $SYNCH system service and fix $QIOW and $ENQW to use the
0000  ;                 new code for waiting for the combination of EFN and IOSB
0000  ;
0000  ;                 Improve readability of conditionals.
0000  ;
0000  ;                 Add $GETDVIW, $GETJPIW, $GETSYIW, $SNDJBC, $SNDJBCW, and
0000  ;                 $UPDSECW.  All the waiting versions use common code.
0000  ;
0000  ;
0000  ;
0000  ; CHANGE MODE SYSTEM SERVICE DISPATCHER
0000  ;
0000  ; MACRO LIBRARY CALLS
0000  ;
0000
```

```
          0000    243          $ACBDEF                              ;DEFINE AST CONTROL BLOCK OFFSETS
          0000    244          $CHFDEF                              ;DEFINE CONDITION HANDLING OFFSETS
          0000    245          $ENQDEF                              ;DEFINE ENQ SYSTEM SERVICE ARGS
          0000    246          $GETDVIDEF                           ;DEFINE GETDVI SYSTEM SERVICE ARGS
          0000    247          $GETJPIDEF                           ;DEFINE GETJPI SYSTEM SERVICE ARGS
          0000    248          $GETLKIDEF                           ;DEFINE GETLKI SYSTEM SERVICE ARGS
          0000    249          $GETSYIDEF                           ;DEFINE GETSYI SYSTEM SERVICE ARGS
          0000    250          $IPLDEF                              ;DEFINE INTERRUPT PRIORITY LEVELS
          0000    254          $PCBDEF                              ;DEFINE PCB OFFSETS
          0000    255          $PHDDEF                              ;DEFINE PHD OFFSETS
          0000    256          $PRDEF                               ;DEFINE PROCESSOR REGISTERS
          0000    257          $PSLDEF                              ;DEFINE PROCESSOR STATUS FIELDS
          0000    258          $RABDEF                              ;DEFINE RMS RAB FIELDS
          0000    259          $RPBDEF                              ;DEFINE REBOOT PARAMETER BLOCK
          0000    260          $QIODEF                              ;DEFINE QIO SYSTEM SERVICE ARGS
          0000    261          $SGNDEF                              ;DEFINE SYSGEN PARAMETERS
          0000    262          $SNDJBCDEF                           ;DEFINE SNDJBC SYSTEM SERVICE ARGS
          0000    263          $SSDEF                               ;DEFINE SYSTEM STATUS VALUES
          0000    264          $SYNCHDEF                            ;DEFINE SYNCH SYSTEM SERVICE ARGS
          0000    265          $UPDSECDEF                           ;DEFINE UPDATE SECTION SYS SRV ARGS
          0000    266 ;
          0000    267 ; LOCAL EQUATES
          0000    268 ;
00000001  0000    269          CAT0 =          1@0
00000080  0000    270          CAT7 =          1@7
00000081  0000    271          DEF_MASK =      CAT0!CAT7            ;INHIBIT FOR 'ALL' AND 'NOT EXIT'
00000080  0000    272          EXC_MASK =      CAT7                 ;INHIBIT ONLY FOR 'ALL' CASE
          0000    273 ;
          0000    274 ; LOCAL MACROS
          0000    275 ;
          0000    276 ;        GSYSSRV - GENERATE SYSTEM SERVICE ENTRY VECTOR
          0000    277 ;
          0000    278 ;        GSYSSRV SRVNAME,MODE,NARG,REGISTERS,MASK,NOSYNC
          0000    279 ;
          0000    280 ;        WHERE:
          0000    281 ;                SRVNAME - SERVICE NAME LESS ANY PREFIX (SYS$,EXE$,RMS$$)
          0000    282 ;                MODE - MODE DESIGNATOR FOR SERVICE (K,E,ALL,R)
          0000    283 ;                NARG - REQUIRED NUMBER OF ARGUMENTS
          0000    284 ;                REGISTERS - REGISTER SAVE LIST
          0000    285 ;                MASK - SERVICE INHIBIT MASK(BIT SET IN CAT INHIBITS)
          0000    286 ;                NOSYNC - NON-ZERO IF RMS SYNCHRONIZATION CODE NOT TO BE INCLUDED
          0000    287 ;
          0000    288
          0000    289          .MACRO  GSYSSRV,SRVNAME,MODE,NARG,REGS,MASK=DEF_MASK,NOSYNC
          0000    290          .IF     NDF,RMSSWITCH
          0000    291          .IF     DF,LIBSWITCH
          0000    292          .PSECT  $$$0000,QUAD
          0000    293          .IFF
          0000    294          .PSECT  $$$000,QUAD
          0000    295          .ENDC
          0000    296          .ALIGN  QUAD
          0000    297          .IF DF  LIBSWITCH
          0000    298 SYS$'SRVNAME::
          0000    299          .IFF
          0000    300          .IF     NDF,MPSWITCH
          0000    301          .WORD   ^M<REGS>
          0000    302          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        .SBTTL  SYSTEM SERVICE VECTOR DEFINITION
          0000  ;
          0000  ;
          0000  ;     DEFINE ALL SYSTEM SERVICE VECTOR POSITIONS
          0000  ;
          0000  ;
          0000
          0000        .PSECT  $$$0000,QUAD,ABS
7FFEDE00  0000  .=^X7FFEDE00                        ;BIASED IN P1 SPACE
          DE00  VECBASE:                            ;VECTOR AREA BASE
          DE00
          DE00  ;
          DE00  ;     QIO AND WAIT COMPOSITE SERVICE
          DE00  ;
          DE00  ;     THE QIO AND WAITFR COMPOSITE SERVICE OCCUPIES THE FIRST TWO
          DE00  ;     SYSTEM SERVICE VECTOR POSITIONS.  IT IS CONSTRUCTED BY
          DE00  ;     FROM TWO DISCRETE CHMK INSTRUCTIONS, ONE PERFORMING THE QIO
          DE00  ;     AND THE OTHER PERFORMING THE WAITFR, WHICH RELY UPON THE
          DE00  ;     COMPATIBLE ARGUMENT LISTS OF THESE TWO SERVICES. WAITFR HAS
          DE00  ;     A SINGLE ARGUMENT, THE EVENT FLAG, WHICH IS THE FIRST ARGUMENT
          DE00  ;     IN THE QIO ARGUMENT LIST.
          DE00  ;
          DE00
          DE00        GCOMPSRVB QIOW,-              ;QIO AND WAIT
          DE00            <QIO_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
          DE00        GCOMPSRVE     2               ;RESERVE 2 QUADWORDS FOR VECTOR
          DE10
          DE10  ;
          DE10  ; CONDITION HANDLER DISPATCH VECTOR
          DE10  ;
          DE10  ; THE FOLLOWING VECTOR IS INCLUDED IN THE SYSTEM VECTOR SPACE SO THAT BOTH
          DE10  ; HARDWARE-DETECTED (EXCEPTIONS) AND SOFTWARE-DETECTED (SIGNALS) CONDITIONS
          DE10  ; CAN BE DISPATCHED FROM THE SAME CALL INSTRUCTION. THIS IS NECESSARY SO
          DE10  ; THAT THE STACK SEARCH ALGORITHM AND THE UNWIND SYSTEM SERVICE CAN DETECT
          DE10  ; AND PROPERLY PROCESS MULTIPLE ACTIVE SIGNALS AND/OR EXCEPTIONS.
          DE10  ;
          DE10
          DE10        .ALIGN  QUAD
80000010  DE10  SYS$CALL_HANDL == . - ^X7FFEDE00 + ^X80000000
7FFEDE18  DE10        .BLKQ   1                     ;RESERVE SPACE
          DE18
          DE18  ;
          DE18  ; COMMAND INTERPRETER DISPATCH VECTOR
          DE18  ;
          DE18  ; THE FOLLOWING VECTOR IS INCLUDED IN THE SYSTEM VECTOR SPACE SO THAT DIRECT
          DE18  ; CALLS CAN BE MADE TO THE CURRENT COMMAND INTERPRETER WITHOUT HAVING TO KNOW
          DE18  ; THE ADDRESS OF ITS SERVICE ROUTINE.
          DE18  ;
          DE18
          DE18        .ALIGN  QUAD
          DE18  SYS$CLI::                           ;COMMAND INTERPRETER DISPATCH
7FFEDE20  DE18        .BLKQ   1                     ;RESERVE SPACE
```

```
DE20   ;
DE20   ;       DEFINE REMAINING SERVICES
DE20   ;
DE20
DE20           GSYSSRV ADJSTK,K,3,-            ;ADJUST OUTER MODE STACK POINTER
DE20                   <R2,R3,R4,R5,R6>,-      ;REGISTERS R2-R6
DE20                   EXC_MASK                ;EXCEPTION MASK
DE28           GSYSSRV ADJWSL,K,2,-            ;ADJUST WORKING SET LIMIT
DE28                   <R2,R3,R4,R5>           ;REGISTERS R2-R5
DE30           GSYSSRV ALCDNP,K,4,-            ;ALLOCATE DIAGNOSTIC PAGE
DE30                   <R2,R3,R4,R5,R6,R7>     ;REGISTERS R2-R7
DE38           GSYSSRV ALLOC,K,4,-             ;ALLOCATE DEVICE
DE38                   <R2,R3,R4,R5,R6>        ;REGISTERS R2-R6
DE40           GSYSSRV ASCEFC,K,4,-            ;ASSOCIATE COMMON EVENT FLAG CLUSTER
DE40                   <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DE48           GSYSSRV ASCTIM,ALL,3,-          ;CONVERT TO ASCII TIME
DE48                   <R2,R3,R4,R5,R6>        ;REGISTERS R2-R6
DE50           GSYSSRV ASSIGN,K,4,-            ;ASSIGN I/O CHANNEL
DE50                   <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DE58           GSYSSRV BINTIM,ALL,2,-          ;CONVERT TO BINARY TIME
DE58                   <R2,R3,R4,R5,R6,R7,R8>  ;REGISTERS R2-R8
DE60           GSYSSRV CANCEL,K,1,-            ;CANCEL I/O ON CHANNEL
DE60                   <R2,R3,R4,R5,R6,R7,R8>  ;REGISTERS R2-R8
DE68           GSYSSRV CANTIM,K,2,-            ;CANCEL TIMER REQUEST
DE68                   <R2,R3,R4,R5>           ;REGISTERS R2-R5
DE70           GSYSSRV CANWAK,K,2,-            ;CANCEL WAKE UP REQUESTS
DE70                   <R2,R3,R4,R5>           ;REGISTERS R2-R5
DE78           GSYSSRV CRMPSC,K,12,-           ;CREATE AND MAP SECTION
DE78                   <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DE80           GSYSSRV CLRPAR,K,2,-            ;CLEAR HARD PARITY ERROR
DE80                   <R2,R3,R4,R5>           ;REGISTERS R2-R5
DE88           GSYSSRV CMEXEC,E,2,-            ;CHANGE MODE TO EXECUTIVE
DE88                   <R4>                    ;REGISTER R4
DE90           GSYSSRV CMKRNL,K,2,-            ;CHANGE MODE TO KERNEL
DE90                   <R4>                    ;REGISTER R4
DE98           GSYSSRV CLREF,K,1,-             ;CLEAR EVENT FLAG
DE98                   <R2,R3,R4,R5>           ;REGISTERS R2-R5. SEE WAITFR COMMENTS.
DEA0           GSYSSRV CNTREG,K,4,-            ;CONTRACT REGION
DEA0                   <R2,R3,R4,R5,R6,R7>     ;REGISTERS R2-R7
DEA8           GSYSSRV GETPTI,K,5,-            ;GET PAGE TABLE INFORMATION
DEA8                   <R2,R3,R4,R5,R6,R7,R8,R9,R10> ;REGISTERS R2-R10
DEB0           GSYSSRV CRELOG,ALL,4,-          ;CREATE LOGICAL NAME
DEB0                   <R2,R3,R4,R5,R6,R7,R8>  ;REGISTERS R2-R8
DEB8           GSYSSRV CREMBX,K,7,-            ;CREATE MAILBOX
DEB8                   <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DEC0           GSYSSRV CREPRC,K,12,-           ;CREATE PROCESS
DEC0                   <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DEC8           GSYSSRV CRETVA,K,3,-            ;CREATE VIRTUAL ADDRESS
DEC8                   <R2,R3,R4,R5,R6,R7,R8>,-;REGISTERS R2-R8
DEC8                   EXC_MASK                ;EXCEPTION MASK
DED0           GSYSSRV DACEFC,K,1,-            ;DISASSOCIATE EVENT FLAG CLUSTER
DED0                   <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DED8           GSYSSRV DALLOC,K,2,-            ;DEALLOCATE DEVICE
DED8                   <R2,R3,R4,R5,R8>        ;REGISTERS R2-R5,R8
DEE0           GSYSSRV DASSGN,K,1,-            ;DEASSIGN I/O CHANNEL
DEE0                   <R2,R3,R4,R5,R6,R7,R8>  ;REGISTERS R2-R8
DEE8           GSYSSRV DCLAST,K,3,-            ;DECLARE AST SYSTEM SERVICE
```

```
DEE8             <R2,R3,R4,R5>                   ;REGISTERS R2-R5
DEF0     GSYSSRV DCLEXH,K,1,-                    ;DECLARE EXIT HANDLER
DEF0             <R2,R3,R4>                      ;REGISTERS R2-R4
DEF8     GSYSSRV DELLOG,ALL,3,-                  ;DELETE LOGICAL NAME
DEF8             <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
DF00     GSYSSRV DELMBX,K,1,-                    ;DELETE MAILBOX
DF00             <R2,R3,R4,R5>                   ;REGISTERS R2-R5
DF08     GSYSSRV DELPRC,K,2,-                    ;DELETE PROCESS
DF08             <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R5
DF10     GSYSSRV DELTVA,K,3,-                    ;DELETE VIRTUAL ADDRESS
DF10             <R2,R3,R4,R5,R6,R7>,-           ;REGISTERS R2-R7
DF10             EXC_MASK                        ;EXCEPTION MASK
DF18     GSYSSRV DGBLSC,K,3,-                    ;DELETE GLOBAL SECTION
DF18             <R2,R3,R4,R5,R6,R7,R8,R9,R10> ;REGISTERS R2-R10
DF20     GSYSSRV DLCDNP,K,2,-                    ;DEALLOCATE DIAGNOSTIC PAGE
DF20             <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R7
DF28     GSYSSRV DLCEFC,K,1,-                    ;DELETE COMMON EVENT CLUSTER
DF28             <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DF30     GSYSSRV UPDSEC,K,8,-                    ;UPDATE SECTION FILE
DF30             <R2,R3,R4,R5,R6,R7,R8>          ;R2-R8
DF38     GSYSSRV SNDERR,K,1,-                    ;SEND MSG TO ERROR LOGGER
DF38             <R2,R3,R4,R5>                   ;REGISTERS R2-R5
DF40     GSYSSRV EXIT,K,1,-                      ;IMAGE EXIT
DF40             <R4>,0                          ;REGISTER R4, ALWAYS ALLOWED!
DF48     GSYSSRV EXPREG,K,4,-                    ;EXPAND PROGRAM REGION
DF48             <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
DF50     GSYSSRV FAO,ALL,0,-                     ;FORMAT ASCII OUTPUT
DF50             <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
DF58     GSYSSRV FAOL,ALL,0,-                    ;FORMAT ASCII OUTPUT WITH VALUE LIST
DF58             <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DF60     GSYSSRV FORCEX,K,3,-                    ;FORCE EXIT
DF60             <R2,R3,R4,R5>                   ;REGISTERS R2-R5
DF68     GSYSSRV IMGSTA,ALL,6,-                  ;IMAGE STARTUP
DF68             <>                              ;REGISTERS NONE
DF70     GSYSSRV SNDJBC,E,7,-                    ;SEND TO JOB CONTROLLER
DF70             <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DF78     GSYSSRV GETTIM,E,1,-                    ;GET TIME
DF78             <>                              ;NO REGISTERS
DF80     GCOMPSRVB UPDSECW,-                     ;UPDATE SECTION AND WAIT
DF80             <UPDSEC_MASK ! GETJPI_SYNCH_MASK>
DF80     GCOMPSRVE        1
DF88     GSYSSRV HIBER,K,0,-                     ;HIBERNATE
DF88             <R2,R3,R4,R5>                   ;REGISTERS R2-R5
DF90     GSYSSRV IMGACT,E,8,-                    ;IMAGE ACTIVATION
DF90             <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DF98     GSYSSRV LCKPAG,K,3,-                    ;LOCK PAGE IN MEMORY
DF98             <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
DFA0     GSYSSRV LKWSET,K,3,-                    ;LOCK PAGES IN WORKING SET
DFA0             <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
DFA8     GSYSSRV MGBLSC,K,7,-                    ;MAP GLOBAL SECTION
DFA8             <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DFB0     GSYSSRV PURGWS,K,1,-                    ;PURGE WORKING SET
DFB0             <R2,R3,R4,R5,R6,R7,R8>          ;R2-R8
DFB8     GSYSSRV NUMTIM,E,2,-                    ;CONVERT TIME TO NUMERIC
DFB8             <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R7
DFC0     GSYSSRV SNDOPR,E,2,-                    ;SEND MSG TO OPERATOR
DFC0             <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>       ;REGISTERS R2-R11
```

```
DFC8  1334        $SYSSRV QIO,K,12,-                    ;QUEUE I/O REQUEST
DFC8  1335                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
DFD0  1336        $SYSSRV READEF,K,2,-                  ;READ EVENT FLAG
DFD0  1337                <R2,R3,R4,R5>                 ;REGISTERS R2-R5
DFD8  1338        $SYSSRV RESUME,K,2,-                  ;RESUME PROCESS
DFD8  1339                <R2,R3,R4,R5>                 ;REGISTERS R2-R5
DFE0  1340        $SYSSRV RUNDWN,K,1,-                  ;RUNDOWN
DFE0  1341                <R2,R3,R4,R5,R6,R7>           ;REGISTERS R2-R7
DFE8  1342        $SYSSRV SNDSMB,E,2,-                  ;SEND MSG TO SYMBIONT MANAGER
DFE8  1343                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>      ;REGISTERS R2-R11
DFF0  1344        $SYSSRV SCHDWK,K,4,-                  ;SCHEDULE WAKEUP
DFF0  1345                <R2,R3,R4,R5,R6,R7,R8,R9> ;REGISTERS R2-R9
DFF8  1346        $SYSSRV SETAST,K,1,-                  ;SET AST ENABLE SERVICE
DFF8  1347                <R2,R3,R4,R5>                 ;REGISTERS R2-R5
E000  1348        $SYSSRV SETEF,K,1,-                   ;SET EVENT FLAG
E000  1349                <R2,R3,R4,R5>                 ;REGISTERS R2-R5. SEE WAITFR COMMENTS.
E008  1350        $SYSSRV SETEXV,K,4,-                  ;SET EXCEPTION VECTOR
E008  1351                <R2,R3,R4,R5>                 ;REGISTERS R2-R5
E010  1352        $SYSSRV SETPRN,K,1,-                  ;SET PROCESS  NAME
E010  1353                <R2,R3,R4,R5,R6,R7,R8,R9> ;REGISTERS R2-R9
E018  1354        $SYSSRV SETPRA,K,2,-                  ;SET POWER RECOVERY AST
E018  1355                <R2,R3,R4,R5>                 ;REGISTERS R2-R5
E020  1356        $SYSSRV SETIMR,K,4,-                  ;SET TIMER
E020  1357                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E028  1358        $SYSSRV SETPRI,K,4,-                  ;SET PROCESS PRIORITY
E028  1359                <R2,R3,R4,R5>                 ;REGISTERS R2-R5
E030  1360        $SYSSRV SETPRT,K,5,-                  ;SET PAGE PROTECTION
E030  1361                <R2,R3,R4,R5,R6,R7,R8,R9> ;REGISTERS R2-R9
E038  1362        $SYSSRV SETRWM,K,1,-                  ;SET RESOURCE WAIT MODE
E038  1363                <R4>                          ;REGISTER R4
E040  1364        $SYSSRV SETSFM,K,1,-                  ;SET SYSTEM SERVICE FAILURE MODE
E040  1365                <R4>,EXC_MASK                 ;REGISTER R4, AND EXECPTION MASK
E048  1366        $SYSSRV SETSWM,K,1,-                  ;SET PROCESS SWAP MODE
E048  1367                <R4>                          ;REGISTER R4
E050  1368        $SYSSRV SUSPND,K,2,-                  ;SUSPEND PROCESS
E050  1369                <R2,R3,R4,R5>                 ;REGISTERS R2-R5
E058  1370        $SYSSRV TRNLOG,ALL,6,-                ;TRANSLATE LOGICAL NAME
E058  1371                <R2,R3,R4,R5,R6,R7,R8>        ;REGISTERS R2-R8
E060  1372        $SYSSRV ULKPAG,K,3,-                  ;UNLOCK PAGE FROM MEMORY
E060  1373                <R2,R3,R4,R5,R6,R7,R8>        ;REGISTERS R2-R8
E068  1374        $SYSSRV ULWSET,K,3,-                  ;UNLOCK PAGES FROM WORKING SET
E068  1375                <R2,R3,R4,R5,R6,R7,R8>        ;REGISTERS R2-R8
E070  1376        $SYSSRV UNWIND,ALL,2,-                ;UNWIND PROCEDURE CALL STACK
E070  1377                <R2,R3,R4,R5>                 ;REGISTERS R2-R5
E078  1378        $SYSSRV WAITFR,K,1,-                  ;WAIT FOR EVENT FLAG
E078  1379                <R2,R3,R4,R5,R6>              ;REGISTERS R2-R6.  IF R8 IS EVER USED
E080  1380                                              ;THE RMS SYCHRONIZATION CODE MUST BE
E080  1381                                              ;MODIFIED TO SAVE IT ALSO.
E080  1382        $SYSSRV WAKE,K,2,-                    ;WAKE PROCESS
E080  1383                <R2,R3,R4,R5>                 ;REGISTERS R2-R5
E088  1384        $SYSSRV WFLAND,K,2,-                  ;WAIT FOR LOGICAL AND OF EVENT FLAGS
E088  1385                <R2,R3,R4,R5,R6>              ;REGISTERS R2-R6
E090  1386        $SYSSRV WFLOR,K,2,-                   ;WAIT FOR LOGICAL OR OF EVENT FLAGS
E090  1387                <R2,R3,R4,R5,R6>              ;REGISTERS R2-R5
E098  1388        $SYSSRV BRDCST,ALL,2,-                ;BROADCAST TO TERMINALS
E098  1389                <R2,R3,R4,R5,R6>              ;REGISTERS R2-R6
E0A0  1390        $SYSSRV DCLCMH,K,3,-                  ;DECLARE CHANGE MODE HANDLER
```

```
E0A0  1391                  <R4>                             ;SAVE R4
E0A8  1392          GSYSSRV SETPFM,K,4,-                     ;SET PAGE FAULT MONITORING
E0A8  1393                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E0B0  1394          GSYSSRV GETMSG,ALL,5,-                   ;GET MESSAGE
E0B0  1395                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E0B8  1396          GSYSSRV DERLMB,K,1,-                     ;DECLARE ERROR LOG MAILBOX
E0B8  1397                  <R2,R3,R4,R5>                    ;REGISTERS R2-R5
E0C0  1398          GSYSSRV CANEXH,K,1,-                     ;CANCEL EXIT HANDLER
E0C0  1399                  <R2,R3,R4,R5>                    ;REGISTERS R2-R5
E0C8  1400          GSYSSRV GETCHN,K,5,-                     ;GET CHANNEL INFORMATION
E0C8  1401                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E0D0  1402          GSYSSRV GETDEV,K,5,-                     ;GET DEVICE INFORMATION
E0D0  1403                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E0D8  1404          GSYSSRV GETJPI,K,7,-                     ;GET JOB PROCESS INFORMATION
E0D8  1405                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E0E0  1406          GSYSSRV PUTMSG,ALL,3,-                   ;PUT FORMATTED ERROR MESSAGE
E0E0  1407                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E0E8  1408          GSYSSRV EXCMSG,ALL,2,-                   ;OUTPUT EXCEPTION SUMMARY MESSAGE
E0E8  1409                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E0F0  1410          GSYSSRV SNDACC,E,2,-                     ;SEND MSG TO ACOUNTING MANAGER
E0F0  1411                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E0F8  1412          GSYSSRV SETIME,K,1,-                     ;SET SYSTEM TIME
E0F8  1413                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E100  1414          GSYSSRV SETPRV,K,4,-                     ;SET PRIVILEGES
E100  1415                  <R2,R3,R4,R5,R6,R7,R8>           ;REGISTERS R2-R8
```

```
          E108   ;
          E108   ;       SPECIAL VECTORS FOR AST DELIVERY AND CLEARING
          E108   ;
          E108   ;       SYS$CLRAST CLEARS THE CURRENTLY ACTIVE AST STATUS
          E108   ;
          E108   ;       SYS$GL_ASTRET CONTAINS THE VALUE OF THE RETURN ADDRESS FROM
          E108   ;       THE CALL INSTRUCTION USED TO DISPATCH AN AST. THIS VALUE CAN
          E108   ;       BE USED WHEN SEARCHING UP THE STACK FOR THE AST CALL FRAME.
          E108   ;
          E108           .PSECT  $$$0000,QUAD
          E108           .ALIGN  QUAD
          E108   SYS$CLRAST::                            ;CLEAR ACTIVE AST
7FFEE110  E108           .BLKL   2                       ;
          E110           .ALIGN  QUAD                    ;
          E110   SYS$GL_ASTRET::                         ;
7FFEE114  E110           .BLKL   1                       ;
          E114   SYS$GL_COMMON::                         ;ADDRESS OF CORE COMMON DESCRIPTOR
7FFEE118  E114           .BLKL   1                       ;
          E118
          E118   ;
          E118   ; ENTRY VECTOR FOR CONDITION HANDLER SEARCH. LIB$SIGNAL USES THIS VECTOR
          E118   ; TO SHARE EXCEPTION'S CODE TO SEARCH FOR AND CALL CONDITION HANDLERS.
          E118   ; THIS ENTRY IS NOT CALLED; RATHER, IT IS JUMPED TO. NO RETURN IS MADE.
          E118   ;
          E118
          E118           .ALIGN  QUAD
          E118   SYS$SRCHANDLER::
7FFEE120  E118           .BLKQ   1                       ;RESERVE SPACE
          E120
          E120   ;
          E120   ;  NOTE THAT THE CODE IN PSECT $$$000 AT THIS POINT CANNOT EXCEED 320 (HEX)
          E120   ;  WITHOUT MODIFYING THE RMS SYNCHRONIZATION CODE WHICH PRECEDES THE RMS
          E120   ;  VECTORS WHICH CANNOT BE MOVED.
          E120   ;
          E120
```

```
SYS
Pse

PSE
---
.
$AB
$$$

Pha
---
Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
183
The
234
43

Mac
---
_$2
_$2
TOT

120

The

MAC
```

```
E120  1478 ;
E120  1479 ; Set up the base for the RMS service codes. We leave a hole so that
E120  1480 ; other exec mode system services can be defined later in this module.
E120  1481 ; The hole is defined by the offset between ECASCTR and RCASCTR; it
E120  1482 ; is checked with an ASSUME at the end of all service definitions.
E120  1483 ;
E120  1487
```

**F

```
                 E120  ;++
                 E120  ;
                 E120  ;  RMS SERVICES
                 E120  ;
                 E120  ;
                 E120  ;  RMS SYNCHRONIZATION ROUTINE
                 E120  ;
                 E120  ;  THE FOLLOWING ROUTINE IS USED BY THE VARIOUS RMS SERVICES IN ORDER
                 E120  ;  TO AWAIT I/O COMPLETION.  THE ROUTINE IS IN THE VECTOR AREA IN ORDER
                 E120  ;  TO WAIT AT THE CALLER'S MODE,  THUS ALLOWING AST ACTIVITY FOR EITHER
                 E120  ;  USER OR SUPERVISOR MODE, OR BOTH.
                 E120  ;
                 E120  ;  THE FAB/RAB IS CHECKED FOR A LEGAL BLOCK ID, I.E., A 1 OR 3, AND
                 E120  ;  AN ERROR RETURNED IF INVALID.  THE STRUCTURE IS NOT REPROBED.
                 E120  ;
                 E120  ;  NOTE THAT EACH RMS SERVICE VECTOR TERMINATES WITH A BRANCH TO THIS
                 E120  ;  ROUTINE.
                 E120  ;
                 E120  ;  THIS ROUTINE ASSUMES THAT THE FOLLOWING REGISTERS HAVE BEEN SET BY THE
                 E120  ;  EXITING RMS EXEC-LEVEL CODE WHENEVER A STALL IS REQUIRED:
                 E120  ;
                 E120  ;       R3      EFN TO WAIT ON
                 E120  ;       R8      RAB/FAB ADDRESS TO WAIT ON
                 E120  ;       R4      (RMSWAIT BR ENTRY POINT ONLY, $WAIT SERVICE) FLAG FOR WAIT TYPE
                 E120  ;               (0 = SAME RAB, 1 = DIFFERENT RABS)
                 E120  ;
                 E120  ;--
                 E120          .PSECT  $$$0000,QUAD
       7FFEE120  E120          .BLKB   ^X320-<.-VECBASE>
       7FFEE168  E120          .BLKB   ^X48                    ;THIS TAKES THE SPACE OF THE CODE
                 E168                                          ;WHEN GENERATING THE GLOBAL SYMBOLS
```

```
E168  ;
E168  ;
E168  ;  DEFINE RMS SERVICES
E168  ;
E168  ;
E168  ;  HIGH USE RECORD OPERATIONS
E168  ;
E168         RMSSRV  DELETE           ;DELETE A RECORD
E170         .NLIST CND
E170         RMSSRV  FIND             ;FIND RECORD
E178         RMSSRV  FREE             ;RELEASE LOCK ON ALL RECORDS
E180         RMSSRV  GET              ;GET A RECORD
E188         RMSSRV  PUT              ;PUT A RECORD
E190         RMSSRV  READ             ;READ A BLOCK
E198         RMSSRV  RELEASE          ;RELEASE LOCK ON NAMED RECORD
E1A0         RMSSRV  UPDATE           ;REWRITE EXISTING RECORD
E1A8         RMSSRV  WAIT             ;STALL FOR RECORD OPERATION COMPLETE
E1B0         RMSSRV  WRITE            ;WRITE BLOCK
E1B8  ;
E1B8  ;  LOWER USAGE OPERATIONS
E1B8  ;
E1B8         RMSSRV  CLOSE            ;CLOSE FILE
E1C0         RMSSRV  CONNECT          ;CONNECT RAB
E1C8         RMSSRV  CREATE           ;CREATE FILE
E1D0         RMSSRV  DISCONNECT       ;DISCONNECT RAB
E1D8         RMSSRV  DISPLAY          ;DISPLAY FILE INFORMATION
E1E0         RMSSRV  ERASE            ;ERASE (DELETE) FILE
E1E8         RMSSRV  EXTEND           ;EXTEND FILE ALLOCATION
E1F0         RMSSRV  FLUSH            ;FINISH I/O ACTIVITY FOR STREAM
E1F8         RMSSRV  MODIFY           ;MODIFY FILE ATTRIBUTES
E200         RMSSRV  NXTVOL           ;NEXT VOLUME
E208         RMSSRV  OPEN             ;OPEN FILE
E210         RMSSRV  REWIND           ;REWIND FILE
E218         RMSSRV  SPACE            ;POSITION FOR TRANSFER
E220         RMSSRV  TRUNCATE         ;TRUNCATE FILE
E228         RMSSRV  ENTER            ;ENTER FILENAME INTO DIRECTORY
E230         RMSSRV  PARSE            ;PARSE FILENAME SPECIFICATION
E238         RMSSRV  REMOVF           ;REMOVE FILENAME FROM DIRECTORY
E240         RMSSRV  RENAME,NARG=4    ;RENAME A FILE
E248         RMSSRV  SEARCH           ;SEARCH A FILE DIRECTORY
E250         RMSSRV  SETDDIR,NARG=3,NOSYNC=1
E258                                  ;SET DEFAULT DIRECTORY STRING
E258         RMSSRV  SETDFPROT,REGS=<R2,R3>,NARG=2,NOSYNC=1
E260                                  ;SET DEFAULT FILE PROTECTION MASK
E260         RMSSRV  SSVEXC,REGS=<>,NOSYNC=1
E268                                  ;GENERATE SYS SERV EXCEPTION
E268         RMSSRV  RMSRUNDWN,NARG=2,NOSYNC=1
E270                                  ;PERFORM RUNDOWN ON RMS FILES
E270         RMSSRV  RMSRUHNDLR,NARG=5,NOSYNC=1
E278                                  ;RMS Recovery Unit Handler
E278         RMSSRV  FILESCAN,NARG=3,NOSYNC=1
E280                                  ;Perform syntax check for file specs
E280  ;
E280  ; ADD NEW RMS SERVICES IN FRONT OF THIS CODE!
E280  ;
E280  ; Now we add special non-vector code.  Because of the CASE instruction
E280  ; used at the front of RMS, this code (and any future additional code)
```

```
               E280  1692 ; must be the last element of the RMS area.
               E280  1693 ;
               E280  1694
               E280  1695          GCOMPSRVB                ;Helper branch to error processing
               E280  1704          GCOMPSRVE       1
               E288  1705
               E288  1707
               E288  1708 ; NOTE:  RMSVECEND MARKS THE END OF THE CURRENTLY DEFINED RMS VECTORS.
               E288  1709 ;        SSVECREG2 MARKS THE START OF THE SECOND REGION OF SYSTEM
               E288  1710 ;        SERVICE VECTORS.  THERE IS EMPTY SPACE BETWEEN THESE REGIONS
               E288  1711 ;        FOP FUTURE RMS VECTORS.  IF NECESSARY, THIS SPACE CAN ALSO
               E288  1712 ;        BE USED FOR SYSTEM SERVICE VECTORS BY BACKING UP SSVECREG2
               E288  1713 ;        (TOWARDS THE RMS VECTORS) AND ADDING NEW SYSTEM SERVICE VECTORS
               E288  1714 ;        BEFORE THE ALREADY DEFINED ONES.  IN OTHER WORDS, THESE TWO
               E288  1715 ;        VECTOR REGIONS MAY GROW TOWARDS EACH OTHER.  IF THEY COLLIDE,
               E288  1716 ;        AN ASSEMBLY ERROR IS GENERATED.
               E288  1717
               E288  1719          .PSECT  $$$0000,QUAD
               E288  1723
               E288  1724 RMSVECEND:
     7FFEE3C0  E288  1725 .=VECBASE+^X5C0
               E3C0  1726 SSVECREG2:                        ; START OF SYSTEM SERVICE VECTOR REGION 2
               E3C0  1732
```

```
E3C0  1734          .SBTTL  REGION 2 OF SYS. SERV. VECTOR DEFINITIONS
E3C0  1735
E3C0  1736  ;
E3C0  1737  ; Note: Service codes for exec mode services in this region are
E3C0  1738  ; reserved by the offset defined above between RCASCTR and ECASCTR.
E3C0  1739  ; If the ASSUME at the end of this section breaks, the offset must
E3C0  1740  ; be increased.
E3C0  1741  ;
E3C0  1742
E3C0  1743          GSYSSRV ENQ,K,11,-               ; ENQUEUE
E3C0  1744                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
E3C8  1745          GSYSSRV DEQ,K,4,-                ; DEQUEUE
E3C8  1746                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
E3D0  1747          GCOMPSRVB ENQW,-                 ; ENQUEUE AND WAIT
E3D0  1748                  <ENQ_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
E3D0  1762          GCOMPSRVE     3                  ; RESERVE 3 QUADWORDS FOR VECTOR
E3E8  1763          GSYSSRV SETSSF,K,1,-             ; SET SYSTEM SERVICE FILTER MASK
E3E8  1764                  <R4>                     ; REGISTER R4
E3F0  1765          GSYSSRV SETSTK,K,3,-             ; SET STACK LIMITS
E3F0  1766                  <R2,R3,R4>               ; REGISTERS R2,R3,R4
E3F8  1767          GSYSSRV GETSYI,K,7,-             ; GET SYSTEM INFORMATION
E3F8  1768                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
E400  1769          GSYSSRV IMGFIX,ALL,0,-           ; IMAGE ADDRESS RELOCATION FIXUP
E400  1770                  <R2,R3,R4,R5>            ; REGISTERS R2-R5
E408  1771          GCOMPSRVB       IMGFIX_2,-       ; ********** TEMP **********
E408  1772                  <0>
E408  1773          GCOMPSRVE      1                 ; ********** TEMP **********
E410  1774          GSYSSRV GETDVI,K,8,-             ; GET DEVICE AND VOLUME INFORMATION
E410  1775                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
E418  1776          GCOMPSRVB GETDVIW,-              ; GET DEVICE INFORMATION AND WAIT
E418  1777                  <GETDVI_MASK ! GETJPI_SYNCH_MASK>
E418  1786          GCOMPSRVE     1
E420  1787          GCOMPSRVB GETJPIW,-              ; GET JOB/PROCESS INFORMATION AND WAIT
E420  1788                  <GETJPI_MASK ! GETJPI_SYNCH_MASK>
E420  1798          GCOMPSRVE     2
E430  1799          GCOMPSRVB GETSYIW,-              ; GET SYSTEM INFORMATION AND WAIT
E430  1800                  <GETSYI_MASK ! GETJPI_SYNCH_MASK>
E430  1809          GCOMPSRVE     1
E438  1810          GCOMPSRVB SNDJBCW,-              ; SEND TO JOB CONTROLLER AND WAIT
E438  1811                  <SNDJBC_MASK ! GETJPI_SYNCH_MASK>
E438  1820          GCOMPSRVE     1
E440  1821          GCOMPSRVB SYNCH,-                ; SYNCHRONIZE EFN AND IOSB
E440  1822                  <WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
E440  1861          GCOMPSRVE     6                  ; RESERVE 6 QUADWORDS FOR VECTOR
E470  1862          GSYSSRV ERAPAT,K,3,-             ; GENERATE A SECURITY  ERASE PATTERN
E470  1863                  <R4>                     ; SAVE R4
E478  1864          GSYSSRV CRELNT,K,8,-             ; CREATE LOGICAL NAME TABLE
E478  1865                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E480  1866          GSYSSRV CRELNM,K,5,-             ; CREATE LOGICAL NAME
E480  1867                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E488  1868          GSYSSRV DELLNM,K,3,-             ; DELETE LOGICAL NAME
E488  1869                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E490  1870          GSYSSRV TRNLNM,K,5,-             ; TRANSLATE LOGICAL NAME
E490  1871                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E498  1872          GSYSSRV GETLKI,K,7,-             ; GET LOCK INFORMATION
E498  1873                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
E4A0  1874          GCOMPSRVB GETLKIW,-              ; GET LOCK INFORMATION AND WAIT
```

```
             E4A0  1875               <GETLKI_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
             E4A0  1887       GCOMPSRVE      2                 ; RESERVE 2 QUADWORDS FOR VECTOR
             E4B0  1888
             E4B0  1889       GSYSSRV ASCTOID,E,3,-            ;ASCII TO IDENTIFIER CONVERSION
             E4B0  1890               <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
             E4B8  1891       GSYSSRV FINISH_RDB,E,1,-         ;FINISH RDB CONTEXT STREAM
             E4B8  1892               <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
             E4C0  1893       GSYSSRV IDTOASC,E,6,-            ;IDENTIFIER TO ASCII CONVERSION
             E4C0  1894               <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
             E4C8  1895       GSYSSRV BRKTHRU,K,11,-           ;BREAK THROUGH WRITES
             E4C8  1896               <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
             E4D0  1897       GSYSSRV GRANTID,ALL,5,-          ;GRANT IDENTIFIER TO PROCESS
             E4D0  1898               <R2,R3>                  ;REGISTERS R2-R3
             E4D8  1899       GSYSSRV REVOKID,ALL,5,-          ;REVOKE IDENTIFIER FROM PROCESS
             E4D8  1900               <R2,R3>                  ;REGISTERS R2-R3
             E4E0  1901       GSYSSRV CHKPRO,K,1,-             ;GENERAL PROTECTION CHECK ROUTINE
             E4E0  1902               <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
             E4E8  1903       GCOMPSRVB BRKTHRUW,-             ; BREAK THOUGH WRITE AND WAIT
             E4E8  1904               <BRKTHRU_MASK ! GETJPI_SYNCH_MASK>
             E4E8  1913       GCOMPSRVE      2
             E4F8  1914       GSYSSRV GETQUI,E,7,-             ;GET QUEUE INFORMATION
             E4F8  1915               <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
             E500  1916       GCOMPSRVB GETQUIW,-              ;GET QUEUE INFORMATION AND WAIT
             E500  1917               <GETQUI_MASK ! GETJPI_SYNCH_MASK>
             E500  1926       GCOMPSRVE      2
             E510  1927
             E510  1928 ;
    00004028 E510  1929       CJF$KCASCTR = 16424
             E510  1930 ;
             E510  1931       LDBSRV  CJF$, ALLJDR,      K,   <R4>
             E518  1932       LDBSRV  CJF$, ASSJNL,      K,   <R4>
             E520  1933       LDBSRV  CJF$, CONUIC,      K,   <R4>
             E528  1934       LDBSRV  CJF$, CREJNL,      K,   <R4>
             E530  1935       LDBSRV  CJF$, DEALJDR,     K,   <R4>
             E538  1936       LDBSRV  CJF$, DEASJNL,     ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
             E540  1937       LDBSRV  CJF$, DEASJNL_INT, K,   <R4>
             E548  1938       LDBSRV  CJF$, DELJNL,      K,   <R4>
             E550  1939       LDBSRV  CJF$, DMTJMD,      K,   <R4>
             E558  1940       LDBSRV  CJF$, DSPJNL,      K,   <R4>
             E560  1941       LDBSRV  CJF$, GETJNL,      K,   <R4>
             E568  1942       LDBSRV  CJF$, GETRUI,      K,   <R4>
             E570  1943       LDBSRV  CJF$, MODFLT,      K,   <R4>
             E578  1944       LDBSRV  CJF$, POSJNL,      K,   <R4>
             E580  1945       LDBSRV  CJF$, READJNL,     K,   <R4>
             E588  1946       LDBSRV  CJF$, RECOVER,     K,   <R4>
             E590  1947       LDBSRV  CJF$, MNTJMD,      K,   <R4>
             E598  1948       LDBSRV  CJF$, CRENWV,      K,   <R4>
             E5A0  1949       LDBSRV  CJF$, CONJNLF,     K,   <R4>
             E5A8  1950       LDBSRV  CJF$, DCNJNLF,     K,   <R4>
             E5B0  1951       LDBSRV  CJF$, FORCEJNL,    ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
             E5B8  1952       LDBSRV  CJF$, FORCEJNLW,   ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
             E5C0  1953       LDBSRV  CJF$, WRITEJNL,    ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
             E5C8  1954       LDBSRV  CJF$, WRITEJNLW,   ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
             E5D0  1955       LDBSRV  CJF$, GETCJI,      K,   <R4>
             E5D8  1956       LDBSRV  CJF$, DMTJMDW,     K,   <R4>, 4, 5, DMTJMD
             E5E8  1957       LDBSRV  CJF$, MODFLTW,     K,   <R4>, 4, 5, MODFLT
             E5F8  1958       LDBSRV  CJF$, POSJNLW,     K,   <R4>, 4, 5, POSJNL
```

```
            E608  1959          LDBSRV  CJF$, READJNLW,     K,   <R4>, 4, 5, READJNL
            E618  1960          LDBSRV  CJF$, RECOVERW,     K,   <R4>, 5, 6, RECOVER
            E628  1961
            E628  1962 ;
   00004010 E628  1963          RUF$KCASCTR = 16400
            E628  1964 ;
            E628  1965          LDBSRV  RUF$,  REENTERRU,    K,  <R2,R3,R4,R5,R6>
            E630  1966          LDBSRV  RUF$,  STARTRU,      K,  <R2,R3,R4,R5,R6>
            E638  1967          LDBSRV  RUF$,  PHASE1,       K,  <R2,R3,R4,R5,R6>
            E640  1968          LDBSRV  RUF$,  PHASE2,       K,  <R2,R3,R4,R5,R6>
            E648  1969          LDBSRV  RUF$,  CANCELRU,     K,  <R2,R3,R4,R5,R6>
            E650  1970          LDBSRV  RUF$,  MARKPOINTRU,  K,  <R2,R3,R4,R5,R6>
            E658  1971          LDBSRV  RUF$,  RESETRU,      K,  <R2,R3,R4,R5,R6>
            E660  1972          LDBSRV  RUF$,  DCLRUH,       K,  <R2,R3,R4,R5,R6>
            E668  1973          LDBSRV  RUF$,  CANRUH,       K,  <R2,R3,R4,R5,R6>
            E670  1974          LDBSRV  RUF$,  RUSTATUS,     K,  <R2,R3,R4,R5,R6>
            E678  1975 ;
            E678  1976 ; End Recovery Unit consists of a two-phase commit, so we call each
            E678  1977 ; phase separately.
            E678  1978 ;
            E678  1979          GCOMPSRVB ENDRU, <PHASE1_MASK ! PHASE2_MASK>, RUF$ ; End Recovery Unit
            E678  1990          GCOMPSRVE       2
            E688  1991          GSYSSRV MTACCESS,K,6,-          ;Mag tape installation specific access routi
            E688  1992                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
            E690  1993
            E690  1994 ;
            E690  1995 ; End of system service vector definitions. New system services are
            E690  1996 ; to be added at this point.
            E690  1997 ;
            E690  2003
```

```
     E690  2269
     E690  2345          .END
```

```
$$ARGS                           = 00000008            GETDVI$_NULLARG                  = 00000020
$$T1                             = 00000024            GETJPI$_ASTADR                   = 00000018
CAT0                             = 00000001            GETJPI$_ASTPRM                   = 0000001C
CAT7                             = 00000080            GETJPI$_EFN                      = 00000004
CJF$ALLJDR                         7FFEE510  G         GETJPI$_IOSB                     = 00000014
CJF$ASSJNL                         7FFEE518  G         GETJPI$_ITMLST                   = 00000010
CJF$CONJNLF                        7FFEE5A0  G         GETJPI$_NARGS                    = 00000007
CJF$CONUIC                         7FFEE520  G         GETJPI$_PIDADR                   = 00000008
CJF$CREJNL                         7FFEE528  G         GETJPI$_PRCNAM                   = 0000000C
CJF$CRENWV                         7FFEE598  G         GETLKI$_ASTADR                   = 00000014
CJF$DCNJNLF                        7FFEE5A8  G         GETLKI$_ASTPRM                   = 00000018
CJF$DEALJDR                        7FFEE530  G         GETLKI$_EFN                      = 00000004
CJF$DEASJNL                        7FFEE538  G         GETLKI$_IOSB                     = 00000010
CJF$DEASJNL_INT                    7FFEE540  G         GETLKI$_ITMLST                   = 0000000C
CJF$DELJNL                         7FFEE548  G         GETLKI$_LKIDADR                  = 00000008
CJF$DMTJMD                         7FFEE550  G         GETLKI$_NARGS                    = 00000007
CJF$DMTJMDW                        7FFEE5D8  G         GETLKI$_RESERVED                 = 0000001C
CJF$DSPJNL                         7FFEE558  G         GETSYI$_ASTADR                   = 00000018
CJF$FORCEJNL                       7FFEE5B0  G         GETSYI$_ASTPRM                   = 0000001C
CJF$FORCEJNLW                      7FFEE5B8  G         GETSYI$_CSIDADR                  = 00000008
CJF$GETCJI                         7FFEE5D0  G         GETSYI$_EFN                      = 00000004
CJF$GETJNL                         7FFEE560  G         GETSYI$_IOSB                     = 00000014
CJF$GETRUI                         7FFEE568  G         GETSYI$_ITMLST                   = 00000010
CJF$KCASCTR                      = 00004028            GETSYI$_NARGS                    = 00000007
CJF$MNTJMD                         7FFEE590  G         GETSYI$_NODENAME                 = 0000000C
CJF$MODFLT                         7FFEE570  G         LIBSWITCH                        = 00000001
CJF$MODFLTW                        7FFEE5E8  G         P1VSWITCH                        = 00000001
CJF$POSJNL                         7FFEE578  G         QIO$_ASTADR                      = 00000014
CJF$POSJNLW                        7FFEE5F8  G         QIO$_ASTPRM                      = 00000018
CJF$READJNL                        7FFEE580  G         QIO$_CHAN                        = 00000008
CJF$READJNLW                       7FFEE608  G         QIO$_EFN                         = 00000004
CJF$RECOVER                        7FFEE588  G         QIO$_FUNC                        = 0000000C
CJF$RECOVERW                       7FFEE618  G         QIO$_IOSB                        = 00000010
CJF$WRITEJNL                       7FFEE5C0  G         QIO$_NARGS                       = 0000000C
CJF$WRITEJNLW                      7FFEE5C8  G         QIO$_P1                          = 0000001C
DEF_MASK                         = 00000081            QIO$_P2                          = 00000020
ENQ$_ACMODE                      = 00000028            QIO$_P3                          = 00000024
ENQ$_ASTADR                      = 0000001C            QIO$_P4                          = 00000028
ENQ$_ASTPRM                      = 00000020            QIO$_P5                          = 0000002C
ENQ$_BLKAST                      = 00000024            QIO$_P6                          = 00000030
ENQ$_EFN                         = 00000004            RMSVECEND                          7FFEE288
ENQ$_FLAGS                       = 00000010            RUF$CANCELRU                       7FFEE648  G
ENQ$_LKMODE                      = 00000008            RUF$CANRUH                         7FFEE668  G
ENQ$_LKSB                        = 0000000C            RUF$DCLRUH                         7FFEE660  G
ENQ$_NARGS                       = 0000000B            RUF$ENDRU                          7FFEE678  G
ENQ$_PARID                       = 00000018            RUF$KCASCTR                      = 00004010
ENQ$_PROT                        = 0000002C            RUF$MARKPOINTRU                    7FFEE650  G
ENQ$_RESNAM                      = 00000014            RUF$PHASE1                         7FFEE638  G
EXC_MASK                         = 00000080            RUF$PHASE2                         7FFEE640  G
GETDVI$_ASTADR                   = 00000018            RUF$REENTERRU                      7FFEE628  G
GETDVI$_ASTPRM                   = 0000001C            RUF$RESETRU                        7FFEE658  G
GETDVI$_CHAN                     = 00000008            RUF$RUSTATUS                       7FFEE670  G
GETDVI$_DEVNAM                   = 0000000C            RUF$STARTRU                        7FFEE630  G
GETDVI$_EFN                      = 00000004            SNDJBC$_ASTADR                   = 00000018
GETDVI$_IOSB                     = 00000014            SNDJBC$_ASTPRM                   = 0000001C
GETDVI$_ITMLST                   = 00000010            SNDJBC$_EFN                      = 00000004
GETDVI$_NARGS                    = 00000008            SNDJBC$_FUNC                     = 00000008
```

```
SNDJBC$_IOSB                     = 00000014             SYS$DGBLSC                       7FFEDF18  G
SNDJBC$_ITMLST                   = 00000010             SYS$DISCONNECT                   7FFEE1D0  G
SNDJBC$_NARGS                    = 00000007             SYS$DISPLAY                      7FFEE1D8  G
SNDJBC$_NULLARG                  = 0000000C             SYS$DLCDNP                       7FFEDF20  G
SSVECREG2                          7FFEE3C0             SYS$DLCEFC                       7FFEDF28  G
SYNCH$_EFN                       = 00000004             SYS$ENQ                          7FFEE3C0  G
SYNCH$_IOSB                      = 00000008             SYS$ENQW                         7FFEE3D0  G
SYNCH$_NARGS                     = 00000002             SYS$ENTER                        7FFEE228  G
SYS$ADJSTK                         7FFEDE20  G          SYS$ERAPAT                       7FFEE470  G
SYS$ADJWSL                         7FFEDE28  G          SYS$ERASE                        7FFEE1E0  G
SYS$ALCDNP                         7FFEDE30  G          SYS$EXCMSG                       7FFEE0E8  G
SYS$ALLOC                          7FFEDE38  G          SYS$EXIT                         7FFEDF40  G
SYS$ASCEFC                         7FFEDE40  G          SYS$EXPREG                       7FFEDF48  G
SYS$ASCTIM                         7FFEDE48  G          SYS$EXTEND                       7FFEE1E8  G
SYS$ASCTOID                        7FFEE4B0  G          SYS$FAO                          7FFEDF50  G
SYS$ASSIGN                         7FFEDE50  G          SYS$FAOL                         7FFEDF58  G
SYS$BINTIM                         7FFEDE58  G          SYS$FILESCAN                     7FFEE278  G
SYS$BRDCST                         7FFEE098  G          SYS$FIND                         7FFEE170  G
SYS$BRKTHRU                        7FFEE4C8  G          SYS$FINISH_RDB                   7FFEE4B8  G
SYS$BRKTHRUW                       7FFEE4E8  G          SYS$FLUSH                        7FFEE1F0  G
SYS$CALL_HANDL                   = 80000010  G          SYS$FORCEX                       7FFEDF60  G
SYS$CANCEL                         7FFEDE60  G          SYS$FREE                         7FFEE178  G
SYS$CANEXH                         7FFEE0C0  G          SYS$GET                          7FFEE180  G
SYS$CANTIM                         7FFEDE68  G          SYS$GETCHN                       7FFEE0C8  G
SYS$CANWAK                         7FFEDE70  G          SYS$GETDEV                       7FFEE0D0  G
SYS$CHKPRO                         7FFEE4E0  G          SYS$GETDVI                       7FFEE410  G
SYS$CLI                            7FFEDE18  G          SYS$GETDVIW                      7FFEE418  G
SYS$CLOSE                          7FFEE1B8  G          SYS$GETJPI                       7FFEE0D8  G
SYS$CLRAST                         7FFEE108  G          SYS$GETJPIW                      7FFEE420  G
SYS$CLREF                          7FFEDE98  G          SYS$GETLKI                       7FFEE498  G
SYS$CLRPAR                         7FFEDE80  G          SYS$GETLKIW                      7FFEE4A0  G
SYS$CMEXEC                         7FFEDE88  G          SYS$GETMSG                       7FFEE0B0  G
SYS$CMKRNL                         7FFEDE90  G          SYS$GETPTI                       7FFEDEA8  G
SYS$CNTREG                         7FFEDEA0  G          SYS$GETQUI                       7FFEE4F8  G
SYS$CONNECT                        7FFEE1C0  G          SYS$GETQUIW                      7FFEE500  G
SYS$CREATE                         7FFEE1C8  G          SYS$GETSYI                       7FFEE3F8  G
SYS$CRELNM                         7FFEE480  G          SYS$GETSYIW                      7FFEE430  G
SYS$CRELNT                         7FFEE478  G          SYS$GETTIM                       7FFEDF78  G
SYS$CRELOG                         7FFEDEB0  G          SYS$GL_ASTRET                    7FFEE110  G
SYS$CREMBX                         7FFEDEB8  G          SYS$GL_COMMON                    7FFEE114  G
SYS$CREPRC                         7FFEDEC0  G          SYS$GRANTID                      7FFEE4D0  G
SYS$CRETVA                         7FFEDEC8  G          SYS$HIBER                        7FFEDF88  G
SYS$CRMPSC                         7FFEDE78  G          SYS$IDTOASC                      7FFEE4C0  G
SYS$DACEFC                         7FFEDED0  G          SYS$IMGACT                       7FFEDF90  G
SYS$DALLOC                         7FFEDED8  G          SYS$IMGFIX                       7FFEE400  G
SYS$DASSGN                         7FFEDEE0  G          SYS$IMGFIX_2                     7FFEE408  G
SYS$DCLAST                         7FFEDEE8  G          SYS$IMGSTA                       7FFEDF68  G
SYS$DCLCMH                         7FFEE0A0  G          SYS$LCKPAG                       7FFEDF98  G
SYS$DCLEXH                         7FFEDEF0  G          SYS$LKWSET                       7FFEDFA0  G
SYS$DELETE                         7FFEE168  G          SYS$MGBLSC                       7FFEDFA8  G
SYS$DELLNM                         7FFEE488  G          SYS$MODIFY                       7FFEE1F8  G
SYS$DELLOG                         7FFEDEF8  G          SYS$MTACCESS                     7FFEE688  G
SYS$DELMBX                         7FFEDF00  G          SYS$NUMTIM                       7FFEDFB8  G
SYS$DELPRC                         7FFEDF08  G          SYS$NXTVOL                       7FFEE200  G
SYS$DELTVA                         7FFEDF10  G          SYS$OPEN                         7FFEE208  G
SYS$DEQ                            7FFEE3C8  G          SYS$PARSE                        7FFEE230  G
SYS$DERLMB                         7FFEE0B8  G          SYS$PURGWS                       7FFEDFB0  G
```

```
SYS$PUT                          7FFEE188  G            SYS$WAKE                      7FFEE080  G
SYS$PUTMSG                       7FFEE0E0  G            SYS$WFLAND                    7FFEE088  G
SYS$QIO                          7FFEDFC8  G            SYS$WFLOR                     7FFEE090  G
SYS$QIOW                         7FFEDE00  G            SYS$WRITE                     7FFEE1B0  G
SYS$READ                         7FFEE190  G            UPDSEC$_ACMODE              = 0000000C
SYS$READEF                       7FFEDFD0  G            UPDSEC$_ASTADR              = 0000001C
SYS$RELEASE                      7FFEE198  G            UPDSEC$_ASTPRM              = 00000020
SYS$REMOVE                       7FFEE238  G            UPDSEC$_EFN                 = 00000014
SYS$RENAME                       7FFEE240  G            UPDSEC$_INADR               = 00000004
SYS$RESUME                       7FFEDFD8  G            UPDSEC$_IOSB                = 00000018
SYS$REVOKID                      7FFEE4D8  G            UPDSEC$_NARGS               = 00000008
SYS$REWIND                       7FFEE210  G            UPDSEC$_RETADR              = 00000008
SYS$RMSRUHNDLR                   7FFEE270  G            UPDSEC$_UPDFLG              = 00000010
SYS$RMSRUNDWN                    7FFEE268  G            VECBASE                       7FFEDE00
SYS$RUNDWN                       7FFEDFE0  G
SYS$SCHDWK                       7FFEDFF0  G
SYS$SEARCH                       7FFEE248  G
SYS$SETAST                       7FFEDFF8  G
SYS$SETDDIR                      7FFEE250  G
SYS$SETDFPROT                    7FFEE258  G
SYS$SETEF                        7FFEE000  G
SYS$SETEXV                       7FFEE008  G
SYS$SETIME                       7FFEE0F8  G
SYS$SETIMR                       7FFEE020  G
SYS$SETPFM                       7FFEE0A8  G
SYS$SETPRA                       7FFEE018  G
SYS$SETPRI                       7FFEE028  G
SYS$SETPRN                       7FFEE010  G
SYS$SETPRT                       7FFEE030  G
SYS$SETPRV                       7FFEE100  G
SYS$SETRWM                       7FFEE038  G
SYS$SETSFM                       7FFEE040  G
SYS$SETSSF                       7FFEE3E8  G
SYS$SETSTK                       7FFEE3F0  G
SYS$SETSWM                       7FFEE048  G
SYS$SNDACC                       7FFEE0F0  G
SYS$SNDERR                       7FFEDF38  G
SYS$SNDJBC                       7FFEDF70  G
SYS$SNDJBCW                      7FFEE438  G
SYS$SNDOPR                       7FFEDFC0  G
SYS$SNDSMB                       7FFEDFE8  G
SYS$SPACE                        7FFEE218  G
SYS$SRCHANDLER                   7FFEE118  G
SYS$SSVEXC                       7FFEE260  G
SYS$SUSPND                       7FFEE050  G
SYS$SYNCH                        7FFEE440  G
SYS$TRNLNM                       7FFEE490  G
SYS$TRNLOG                       7FFEE058  G
SYS$TRUNCATE                     7FFEE220  G
SYS$ULKPAG                       7FFEE060  G
SYS$ULWSET                       7FFEE068  G
SYS$UNWIND                       7FFEE070  G
SYS$UPDATE                       7FFEE1A0  G
SYS$UPDSEC                       7FFEDF30  G
SYS$UPDSECW                      7FFEDF80  G
SYS$WAIT                         7FFEE1A8  G
SYS$WAITFR                       7FFEE078  G
```

```
                                            +----------------+
                                            ! 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
$$$0000                         7FFEE690  (*****.)   02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC QUAD

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

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       -------- ---
Initialization                    35    00:00:00.03    00:00:01.89
Command processing               117    00:00:00.81    00:00:07.40
Pass 1                           597    00:00:20.90    00:01:22.01
Symbol table sort                  0    00:00:02.16    00:00:05.76
Pass 2                           207    00:00:05.87    00:00:21.08
Symbol table output               35    00:00:00.24    00:00:01.41
Psect synopsis output              3    00:00:00.02    00:00:00.43
Cross-reference output             0    00:00:00.00    00:00:00.00
Assembler run totals             996    00:00:30.03    00:01:59.98

The working set limit was 2250 pages.
183693 bytes (359 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 0 local symbols.
2347 source lines were read in Pass 1, producing 18 object records in Pass 2.
43 pages of virtual memory were used to define 39 macros.

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

Macro library name                       Macros defined
------------------                       --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                    6
_$255$DUA28:[SYSLIB]STARLET.MLB;2                18
TOTALS (all libraries)                           24

1204 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:P1SYSVECT/OBJ=OBJ$:P1SYSVECT MSRC$:LBSW/UPDATE=(ENH$:LBSW)+MSRC$:P1SW/UPDATE=(ENH$:P1SW)+MSRC$:CMODSSDSP/UPDATE=(ENH$
```

0378 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY