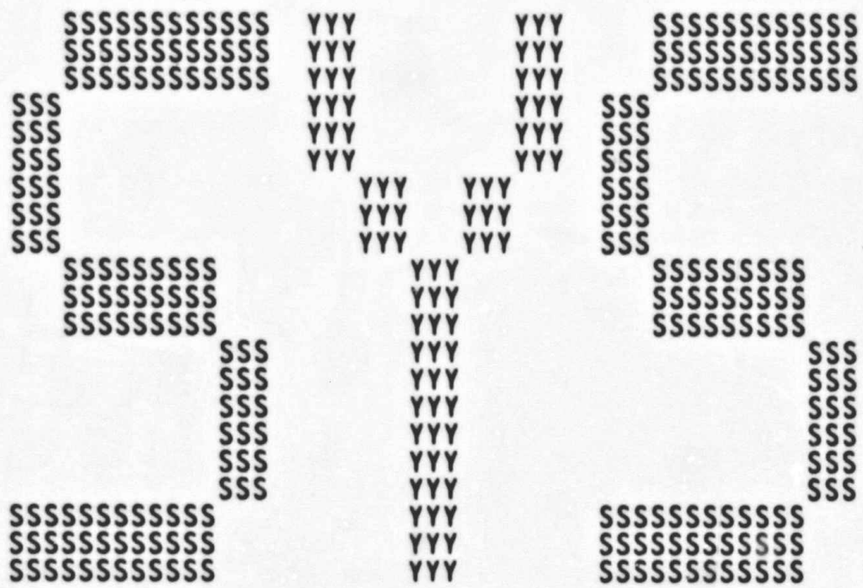
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS

```
  SSSSSSSS  YY      YY   SSSSSSSS  VV      VV  EEEEEEEEEE   CCCCCCCC  TTTTTTTTTT    OOOOOO    RRRRRRRR
  SSSSSSSS  YY      YY   SSSSSSSS  VV      VV  EEEEEEEEEE   CCCCCCCC  TTTTTTTTTT    OOOOOO    RRRRRRRR
SS          YY      YY SS          VV      VV  EE          CC            TT      OO      OO  RR      RR
SS          YY      YY SS          VV      VV  EE          CC            TT      OO      OO  RR      RR
SS            YY  YY   SS          VV      VV  EE          CC            TT      OO      OO  RR      RR
SS            YY  YY   SS          VV      VV  EE          CC            TT      OO      OO  RR      RR
  SSSSSS        YY       SSSSSS    VV      VV  EEEEEEEE    CC            TT      OO      OO  RRRRRRRR
  SSSSSS        YY       SSSSSS    VV      VV  EEEEEEEE    CC            TT      OO      OO  RRRRRRRR
        SS      YY             SS  VV      VV  EE          CC            TT      OO      OO  RR  RR
        SS      YY             SS  VV      VV  EE          CC            TT      OO      OO  RR  RR
        SS      YY             SS    VV  VV    EE          CC            TT      OO      OO  RR    RR      ....
        SS      YY             SS    VV  VV    EE          CC            TT      OO      OO  RR    RR      ....
SSSSSSSS        YY     SSSSSSSS        VV      EEEEEEEEEE   CCCCCCCC      TT        OOOOOO    RR      RR    ....
SSSSSSSS        YY     SSSSSSSS        VV      EEEEEEEEEE   CCCCCCCC      TT        OOOOOO    RR      RR    ....
```

```
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     1 LIBSWITCH=1                                   ;GENERATE LIBRARY FORM OF SERVICE VECTOR
          0000     1         .NLIST  CND
          0000     8         .TITLE  SYS$VECTOR - SYSTEM SERVICE VECTOR DEFINITIONS
          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 system 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    68 ;               For $SETPFM, changed number of parameters from 1 to 4
          0000    69 ;               and changed entry mask to save R2-R11.
          0000    70 ;
          0000    71 ;       V03-036 TMK0002         Todd M. Katz            19-Nov-1983
          0000    72 ;               The entry point for $ASCTOID can no longer be reached as a
```

```
0000     73 ;               branch destination from the executive mode dispatcher.
0000     74 ;               A temporary entry point (EXE$ASCTOID) has been placed within
0000     75 ;               this module, and a JMP is made from it to the real system
0000     76 ;               service entry point (EXE$$ASCTOID).
0000     77 ;
0000     78 ;               Also, change the entry mask for SYS$TRNLOG, so that R8 is
0000     79 ;               now saved.
0000     80 ;
0000     81 ;       V03-035 TMK0001         Todd M. Katz            22-Oct-1983
0000     82 ;               The entry points for $FINISH_RDB and $IDTOASC can no
0000     83 ;               longer be reached as branch destinations from the executive
0000     84 ;               mode dispatcher. Temporary entry points (EXE$FINISH_RDB and
0000     85 ;               EXE$IDTOASC) have been placed within this module, and from
0000     86 ;               each a JMP is made to the real system service entry points
0000     87 ;               (EXE$$FINISH_RDB and EXE$$IDTOASC).
0000     88 ;
0000     89 ;       V03-034 PRB0254         Paul Beck       15-Sep-1983  14:49
0000     90 ;               (1) Correct the way synchronous CJF services are defined.
0000     91 ;               (2) Define loadable RUF services.
0000     92 ;
0000     93 ;       V03-033 WMC0029         Wayne Cardoza           31-Aug-1983
0000     94 ;               Loadable services should not be unconditionally inhibited.
0000     95 ;               Add an alternate CHMx argument to LDBSRV.
0000     96 ;
0000     97 ;       V03-032 DWT0125         David W. Thiel          22-Aug-1983
0000     98 ;               Remove CHECKARGLIST and calls to same.
0000     99 ;
0000    100 ;       V03-031 MKL0167         Mary Kay Lyons          19-Aug-1983
0000    101 ;               Generate loadable service vector for CJF$GETCJI.
0000    102 ;
0000    103 ;       V03-030 KBT0578         Keith B. Thompson       8-Aug-1983
0000    104 ;               Add parameter to $FILESCAN
0000    105 ;
0000    106 ;       V03-029 RAS0178         Ron Schaefer            29-Jul-1983
0000    107 ;               Add code to detect the AST/non-AST RMS FAB/RAB race
0000    108 ;               condition where an RMS operation is initiated while
0000    109 ;               the user FAB/RAB is still waiting for completion of
0000    110 ;               previous operation.
0000    111 ;
0000    112 ;       V03-028 WMC0028         Wayne Cardoza           29-Jun-1983
0000    113 ;               Add CJF services.
0000    114 ;
0000    115 ;       V03-027 WMC0027         Wayne Cardoza           23-Jun-1983
0000    116 ;               Make old logical name services "all mode".
0000    117 ;               Changes to image activator vectors.
0000    118 ;
0000    119 ;       V03-026 JWH0222         Jeffrey W. Horn          2-May-1983
0000    120 ;               Add LDBSRV macro for vector definitions of loadable
0000    121 ;               services.
0000    122 ;
0000    123 ;       V03-025 DMW4035         DMWalp                  26-May-1983
0000    124 ;               Intergate new logical name structures.
0000    125 ;
0000    126 ;       V03-024 LMP0109         L. Mark Pilant,         28-Apr-1983  15:53
0000    127 ;               Make $CHKPRO an EXEC mode system service to allow examination
0000    128 ;               of various system data structures.
0000    129 ;
```

```
0000   130 ;       V03-024 RAS0147         Ron Schaefer            28-APR-1983
0000   131 ;               Add $FILESCAN. Add R8 and R9 to $SETPRN register mask.
0000   132 ;
0000   133 ;       V03-023 JLV0244         Jake VanNoy             27-APR-1983
0000   134 ;               Add $BRKTHRUW. Change $BRDCST to all mode service.
0000   135 ;               $BRDCST now uses $BRKTHRU to do real work.
0000   136 ;
0000   137 ;       V03-022 LMP0099         L. Mark Pilant,         13-Apr-1983  19:15
0000   138 ;               Add the $CHKPRO system service.
0000   139 ;
0000   140 ;       V03-021 ACG0319         Andrew C. Goldstein,    21-Mar-1983  13:51
0000   141 ;               Add $GRANTID and $REVOKID services
0000   142 ;
0000   143 ;       V03-020 JLV0234         Jake VanNoy              1-MAR-1983
0000   144 ;               Add $BRKTHRU service.
0000   145 ;
0000   146 ;       V03-019 RAS0120         Ron Schaefer            25-Feb-1983
0000   147 ;               Add support to echo SYS$INPUT to SYS$OUTPUT.
0000   148 ;               This involves examining the return code from RMS for $GET;
0000   149 ;               if the special status RMS$_ECHO (not returned to users)
0000   150 ;               is found, then create a RAB on the caller's stack and
0000   151 ;               execute a $PUT operation to echo the line.
0000   152 ;               A certain amount of RMS synchronization code was
0000   153 ;               shuffled around in order to make room for this.
0000   154 ;
0000   155 ;       V03-018 ACG0317         Andrew C. Goldstein,    22-Feb-1983  15:16
0000   156 ;               Fix off-by-one in kernel arg vector
0000   157 ;
0000   158 ;       V03-017 RSH0004         R. Scott Hanna          10-Feb-1983
0000   159 ;               Added $ASCTOID, $FINISH_RDB, and $IDTOASC to system service list
0000   160 ;
0000   161 ;       V03-016 RNG0016         Rod N. Gamache           1-Feb-1983
0000   162 ;               Added $GETLKI to system service list
0000   163 ;
0000   164 ;       V03-015 WMC0015         Wayne Cardoza           12-Jan-1983
0000   165 ;               Put back accidentally deleted space holder for RMS synchronization.
0000   166 ;
0000   167 ;       V03-014 DMW4023         DMWalp                   7-Jan-1983
0000   168 ;               Added $CRELNT, $CRELNM, $DELLNM and $TRNLNM
0000   169 ;
0000   170 ;       V03-013 KDM0033         Kathleen D. Morse       13-Dec-1982
0000   171 ;               Correct usage of an interlocked instruction to flush
0000   172 ;               the hardware cache queue.
0000   173 ;
0000   174 ;       V03-012 ROW0146         Ralph O. Weber           6-DEC-1982
0000   175 ;               Insert routine header comments for INHEXCP, CHECKARGLIST,
0000   176 ;               and EXE$CMODKRNLX (MPS$CMODKRNLX).  Move things around so
0000   177 ;               that EXE$CMODKRNL (MPS$CMODKRNL) header comments are near
0000   178 ;               EXE$CMODRKNL (MPS$CMODKRNL) and ASTEXIT comments are near
0000   179 ;               ASTEXIT.  Make basic kernal-mode .PSECT definition for Y$CMODK
0000   180 ;               or MP$CMOD1 immediately after executive mode code so that new
0000   181 ;               code can be inserted in a way that preserves routine headers,
0000   182 ;               conditional assembly, and .PSECT definitions.  Backout ROW145,
0000   183 ;               and in its place, correct conditional assembly of BGEQU 10$
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   239 ;
0000   240 ; MACRO LIBRARY CALLS
0000   241 ;
0000   242
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   303          .IFTF   ;MPSWITCH
```

```
0000    304          .IF B   NOSYNC
0000    305          SRV'MODE        SRVNAME,NARG,MASK
0000    306          .IFF
0000    307          SRV'MODE        SRVNAME,NARG,MASK,NOSYNC
0000    308          .ENDC
0000    309          .ENDC   ;MPSWITCH
0000    310          .IFT
0000    311          .BLKL   2
0000    312          .ENDC
0000    313          .IFF
0000    314          SRV'MODE        SRVNAME,NARG,MASK
0000    315          .ENDC
0000    316          .ENDM   GSYSSRV
0000    317
0000    318 ;
0000    319 ;        GCOMPSRVB - GENERATE COMPOSITE SYSTEM SERVICE ENTRY VECTOR BEGIN
0000    320 ;
0000    321 ;        GCOMPSRVB  SRVNAME,REGISTER_MASK[,PREFIX]
0000    322 ;
0000    323 ;        WHERE:
0000    324 ;                SRVNAME - SERVICE NAME LESS ANY PREFIX (SYS$, EXE$)
0000    325 ;                REGISTER_MASK - SYMBOLIC REGISTER MASK, E.G QIO_MASK
0000    326 ;                PREFIX - IF SUPPLIED, THE PREFIX FOR THE SERVICE NAME.
0000    327 ;                         IF OMITTED, "SYS$" IS ASSUMED.
0000    328 ;
0000    329
0000    330          .MACRO  GCOMPSRVB,SRVNAME,REGMSK,PREFIX=SYS$
0000    331          .IF     NDF,MPSWITCH
0000    332          .IF     NDF,RMSSWITCH
0000    333          .IF     DF,LIBSWITCH
0000    334          .PSECT  $$$0000,QUAD
0000    335          .IFF
0000    336          .PSECT  $$$000,QUAD
0000    337          .ENDC
0000    338          .ALIGN  QUAD
0000    339          .IF DF  LIBSWITCH
0000    340          .IIF    NOT_BLANK, <SRVNAME>,-
0000    341 'PREFIX'SRVNAME::
0000    342          .IFF
0000    343          .ENABL  LSB
0000    344 COMPSTRT=.
0000    345          .IIF    NOT_BLANK, <REGMSK>,-
0000    346          .WORD   <REGMSK>
0000    347          .ENDC
0000    348          .ENDC
0000    349          .ENDC   ;MPSWITCH
0000    350          .ENDM   GCOMPSRVB
0000    351
0000    352 ;
0000    353 ;        GCOMPSRVE - GENERATE COMPOSITE SYSTEM SERVICE ENTRY VECTOR END
0000    354 ;
0000    355 ;        GCOMPSRVE       QUADWORDS
0000    356 ;
0000    357 ;        WHERE:
0000    358 ;                QUADWORDS - NUMBER OF QUADWORDS TO RESERVE FOR VECTOR
0000    359 ;
0000    360
```

```
0000   361          .MACRO  GCOMPSRVE,QUADS
0000   362          .IF     NDF,MPSWITCH
0000   363          .IF     NDF,RMSSWITCH
0000   364          .IF     DF,LIBSWITCH
0000   365          .BLKQ   QUADS
0000   366          .IFF
0000   367 COMPSIZE=.-COMPSTRT
0000   368          .IF     GE,QUADS*8-COMPSIZE
0000   369          .BLKB   QUADS*8-COMPSIZE
0000   370          .IFF
0000   371          .ERROR          ; VECTOR EXCEEDS ALLOCATED SIZE ;
0000   372          .ENDC
0000   373          .DSABL  LSB
0000   374          .ENDC
0000   375          .ENDC
0000   376          .ENDC   ;MPSWITCH
0000   377          .ENDM   GCOMPSRVE
0000   378
0000   379
0000   380 ;
0000   381 ;        SRVK - GENERATE ENTRY FOR KERNEL MODE SERVICE
0000   382 ;
0000   383 ;        SRVK    SRVNAME,NARG,MASK
0000   384 ;
0000   385
0000   386          .MACRO  SRVK,SRVNAME,NARG,MASK
0000   387          .IF     NDF,RMSSWITCH
0000   388          .IF     DF,MPSWITCH
0000   389 CMK$C_'SRVNAME==KCASCTR
0000   390          .IFF    ;MPSWITCH DEFINED
0000   391 CMK$C_'SRVNAME=KCASCTR
0000   392          CHMK    #SRVNAME
0000   393          RET
0000   394          .PSECT  Y$CMODKN,BYTE
0000   395          .=KCASCTR
0000   396          ASSUME NARG LE 127
0000   397          .BYTE   NARG
0000   398          .PSECT  Y$CMODKX,BYTE
0000   399          .=KCASCTR
0000   400          .BYTE   MASK
0000   401          .PSECT  Y$CMODK,BYTE
0000   402          .SIGNED_WORD    EXE$'SRVNAME-KCASE+2
0000   403          .IFTF   ;MPSWITCH
0000   404 SRVNAME=KCASCTR
0000   405 KCASCTR=KCASCTR+1
0000   406          .ENDC   ;MPSWITCH
0000   407          .ENDC
0000   408          .ENDM   SRVK
0000   409
0000   410 ;
0000   411 ;        SRVE - GENERATE ENTRY FOR EXECUTIVE MODE SERVICE
0000   412 ;
0000   413
0000   414          .MACRO  SRVE,SRVNAME,NARG,MASK
0000   415          .IF     NDF,MPSWITCH
0000   416          .IF     NDF,RMSSWITCH
0000   417 CME$C_'SRVNAME=ECASCTR
```

```
0000   418          CHME    #SRVNAME
0000   419          RET
0000   420          .PSECT  Y$CMODEN,BYTE
0000   421          .=ECASCTR
0000   422          ASSUME NARG LE 127
0000   423          .BYTE   NARG
0000   424          .PSECT  Y$CMODEX,BYTE
0000   425          .=ECASCTR
0000   426          .BYTE   MASK
0000   427          .PSECT  Y$CMODE,BYTE
0000   428          .SIGNED_WORD    EXE$'SRVNAME-ECASE+2
0000   429          .ENDC
0000   430 SRVNAME=ECASCTR
0000   431 ECASCTR=ECASCTR+1
0000   432          .ENDC   ;MPSWITCH
0000   433          .ENDM   SRVE
0000   434 ;
0000   435 ;
0000   436 ;    MACROS FOR GENERATING RMS SYSTEM VECTORS
0000   437 ;
0000   438          .MACRO  RMSSRV  SRVNAME NARG=1,REGS=<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>,-
0000   439                          MASK,NOSYNC=0
0000   440          GSYSSRV SRVNAME,R,NARG,<REGS>,MASK,NOSYNC
0000   441          .ENDM   RMSSRV
0000   442 ;
0000   443 ;  SRVR - GENERATE ENTRY FOR RMS SERVICE (EXEC MODE)
0000   444 ;
0000   445          .MACRO  SRVR    SRVNAME,NARG,MASK,NOSYNC
0000   446          .IF     NDF,MPSWITCH
0000   447          .IF     NDF,RMSSWITCH
0000   448 CME$C_'SRVNAME=RCASCTR
0000   449          CHME    #SRVNAME
0000   450          .IF EQ NOSYNC
0000   451          .IIF GT <.+2-RMSSYNC>-127,-
0000   452 RMSSYNC=RMSWBR                              ;RESET BRANCH DESTINATION
0000   453 RMSWBR=.
0000   454          BRB     RMSSYNC
0000   455          .IFF
0000   456          RET
0000   457          .ENDC
0000   458          .PSECT  Y$CMODEN,BYTE
0000   459          .=RCASCTR
0000   460          ASSUME NARG LE 127
0000   461          .BYTE   NARG
0000   462          .PSECT  Y$CMODEX,BYTE
0000   463          .=RCASCTR
0000   464          .BYTE   MASK
0000   465          .IF
0000   466          .PSECT  $$$RMSVEC,BYTE,NOWRT
0000   467          .SIGNED_WORD    RMS$'SRVNAME-RCASE+2
0000   468          .ENDC
0000   469 SRVNAME=RCASCTR
0000   470 RCASCTR=RCASCTR+1
0000   471          .ENDC   ;MPSWITCH
0000   472          .ENDM   SRVR
0000   473
0000   474 ;
```

```
0000    475 ;       SRVALL - GENERATE ENTRY FOR ALL MODE SERVICE
0000    476 ;
0000    477
0000    478         .MACRO  SRVALL,SRVNAME,NARG,MASK
0000    479         .IF     NDF,MPSWITCH
0000    480         .IF NDF,RMSSWITCH
0000    481         JMP     @#EXE$'SRVNAME+2
0000    482         .ENDC
0000    483         .ENDC   ;MPSWITCH
0000    484         .ENDM   SRVALL
0000    485
```

```
0000   487          .SBTTL  Macros for Loadable Services
0000   488
0000   489 ;
0000   490 ;       LDBSRV - Generate Loadable Service Vector
0000   491 ;
0000   492 ;       LDBSRV  PREFIX,SRVNAME,MODE,REGS,SYN_EFN,SYN_IOSB,ALT_CHMX
0000   493 ;
0000   494 ;       Where:
0000   495 ;               PREFIX          - Prefix for system service vector entry point name
0000   496 ;               SRVNAME         - Service name less any prefix (SYS$,CJF$, etc.)
0000   497 ;               MODE            - Mode designator for service (K,E,ALL)
0000   498 ;               REGS            - Register save list
0000   499 ;               SYN_EFN         - Event flag argument number for $SYNCH
0000   500 ;               SYN_IOSB        - IOSB argument number for $SYNCH
0000   501 ;               ALT_CHMX        - Use same CHMx number as this service
0000   502 ;
0000   503
0000   504          .MACRO  LDBSRV,PREFIX,SRVNAME,MODE,REGS,SYN_EFN,SYN_IOSB,ALT_CHMX
0000   505          .IF NDF,RMSSWITCH
0000   506          .IF NDF,MPSWITCH
0000   507              .IF DF,LIBSWITCH
0000   508                  .PSECT  $$$0000,QUAD
0000   509                  .ALIGN  QUAD
0000   510 PREFIX''SRVNAME::
0000   511                  .IF BLANK SYN_EFN
0000   512                      .BLKL    2
0000   513                  .IFF
0000   514                      .BLKL    4
0000   515                  .ENDC
0000   516              .IFF
0000   517                  .PSECT  $$$000,QUAD
0000   518                  .ALIGN  QUAD
0000   519                  .WORD   ^M<REGS>
0000   520                  SRVNAME' MASK = ^M<REGS>
0000   521                  LVEC_'MODE PREFIX,SRVNAME,SYN_EFN,SYN_IOSB,ALT_CHMX
0000   522              .ENDC
0000   523          .ENDC   ; MPSWITCH
0000   524          .ENDC   ; RMSSWITCH
0000   525          .ENDM   LDBSRV
0000   526
0000   527 ;
0000   528 ;       LVEC_K  - Kernel Mode Loadable System Service Vector
0000   529 ;
0000   530 ;       LVEC_K  PREFIX,SERVICE,EFN,IOSB
0000   531 ;
0000   532
0000   533          .MACRO  LVEC_K,PREFIX,SERVICE,EFN,IOSB,ALT_CHMK
0000   534          .IF BLANK ALT_CHMK
0000   535              CMK$C_'SERVICE = PREFIX'KCASCTR
0000   536          .IFF
0000   537              CMK$C_'SERVICE = ALT_CHMK
0000   538          .ENDC
0000   539          CHMK #SERVICE
0000   540          .IF NOT_BLANK EFN
0000   541              PUSHL       #EFN
0000   542              PUSHL       #IOSB
0000   543              JMP         @#EXE$LDB_SYNCH
```

```
0000   544          .IFF
0000   545              RET
0000   546          .ENDC
0000   547          .IF BLANK ALT_CHMK
0000   548              SERVICE = PREFIX'KCASCTR
0000   549              PREFIX'KCASCTR = PREFIX'KCASCTR + 1
0000   550          .IFF
0000   551              SERVICE = ALT_CHMK
0000   552          .ENDC
0000   553          .ENDM   LVEC_K
0000   554
0000   555 ;
0000   556 ;        LVEC_E  - Exec Mode Loadable System Service Vector
0000   557 ;
0000   558 ;        LVEC_E  PREFIX,SERVICE,EFN,IOSB
0000   559 ;
0000   560
0000   561          .MACRO  LVEC_E,PREFIX,SERVICE,EFN,IOSB,ALT_CHME
0000   562          .IF BLANK ALT_CHME
0000   563              CME$C_'SERVICE = PREFIX'ECASCTR
0000   564          .IFF
0000   565              CME$C_'SERVICE = ALT_CHME
0000   566          .ENDC
0000   567          CHME    #SERVICE
0000   568          .IF NOT_BLANK EFN
0000   569              PUSHL       #EFN
0000   570              PUSHL       #IOSB
0000   571              JMP         @#EXE$LDB_SYNCH
0000   572          .IFF
0000   573              RET
0000   574          .ENDC
0000   575          RET
0000   576          .IF BLANK ALT_CHME
0000   577              SERVICE = PREFIX'ECASCTR
0000   578              PREFIX'ECASCTR = PREFIX'ECASCTR + 1
0000   579          .IFF
0000   580              SERVICE = ALT_CHME
0000   581          .ENDC
0000   582          .ENDM   LVEC_E
0000   583
0000   584 ;
0000   585 ;        LVEC_ALL - Mode of caller Loadable System Service Vector
0000   586 ;
0000   587 ;        LVEC_ALL   PREFIX,SERVICE,EFN,IOSB
0000   588 ;
0000   589          .MACRO  LVEC_ALL,PREFIX,SERVICE,EFN,IOSB,ALT_CHMK
0000   590          JMP     @#EXE$'SERVICE
0000   591          .IF NOT_BLANK EFN
0000   592              .ERROR      ; SYNCH NOT ALLOWED FOR ALL-MODE SERVICES
0000   593          .ENDC
0000   594          .ENDM   LVEC_ALL
0000   595
0000   596
```

```
                     0000  1112          .SBTTL  SYSTEM SERVICE VECTOR DEFINITION
                     0000  1113 ;
                     0000  1114 ;
                     0000  1115 ;       DEFINE ALL SYSTEM SERVICE VECTOR POSITIONS
                     0000  1116 ;
                     0000  1117 ;
                     0000  1118
                     0000  1122          .PSECT  $$$0000,QUAD,ABS
80000000             0000  1124 .=^X80000000                         ;BIASED AT THE START OF SYSEM SPACE
                     0000  1132 VECBASE:                             ;VECTOR AREA BASE
                     0000  1133
                     0000  1134 ;
                     0000  1135 ;       QIO AND WAIT COMPOSITE SERVICE
                     0000  1136 ;
                     0000  1137 ;       THE QIO AND WAITFR COMPOSITE SERVICE OCCUPIES THE FIRST TWO
                     0000  1138 ;       SYSTEM SERVICE VECTOR POSITIONS.  IT IS CONSTRUCTED BY
                     0000  1139 ;       FROM TWO DISCRETE CHMK INSTRUCTIONS, ONE PERFORMING THE QIO
                     0000  1140 ;       AND THE OTHER PERFORMING THE WAITFR, WHICH RELY UPON THE
                     0000  1141 ;       COMPATIBLE ARGUMENT LISTS OF THESE TWO SERVICES. WAITFR HAS
                     0000  1142 ;       A SINGLE ARGUMENT, THE EVENT FLAG, WHICH IS THE FIRST ARGUMENT
                     0000  1143 ;       IN THE QIO ARGUMENT LIST.
                     0000  1144 ;
                     0000  1145
                     0000  1146          GCOMPSRVB QIOW,-                ;QIO AND WAIT
                     0000  1147                  <QIO_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
                     0000  1154          GCOMPSRVE       2               ;RESERVE 2 QUADWORDS FOR VECTOR
                     0010  1158
                     0010  1159 ;
                     0010  1160 ; CONDITION HANDLER DISPATCH VECTOR
                     0010  1161 ;
                     0010  1162 ; THE FOLLOWING VECTOR IS INCLUDED IN THE SYSTEM VECTOR SPACE SO THAT BOTH
                     0010  1163 ; HARDWARE-DETECTED (EXCEPTIONS) AND SOFTWARE-DETECTED (SIGNALS) CONDITIONS
                     0010  1164 ; CAN BE DISPATCHED FROM THE SAME CALL INSTRUCTION. THIS IS NECESSARY SO
                     0010  1165 ; THAT THE STACK SEARCH ALGORITHM AND THE UNWIND SYSTEM SERVICE CAN DETECT
                     0010  1166 ; AND PROPERLY PROCESS MULTIPLE ACTIVE SIGNALS AND/OR EXCEPTIONS.
                     0010  1167 ;
                     0010  1168
                     0010  1169          .ALIGN  QUAD
                     0010  1174 SYS$CALL_HANDL::                     ;CONDITION HANDLER DISPATCH
80000018             0010  1185          .BLKQ   1                   ;RESERVE SPACE
                     0018  1190
                     0018  1191 ;
                     0018  1192 ; COMMAND INTERPRETER DISPATCH VECTOR
                     0018  1193 ;
                     0018  1194 ; THE FOLLOWING VECTOR IS INCLUDED IN THE SYSTEM VECTOR SPACE SO THAT DIRECT
                     0018  1195 ; CALLS CAN BE MADE TO THE CURRENT COMMAND INTERPRETER WITHOUT HAVING TO KNOW
                     0018  1196 ; THE ADDRESS OF ITS SERVICE ROUTINE.
                     0018  1197 ;
                     0018  1198
                     0018  1199          .ALIGN  QUAD
                     0018  1201 SYS$CLI::                            ;COMMAND INTERPRETER DISPATCH
80000020             0018  1206          .BLKQ   1                   ;RESERVE SPACE
```

```
0020  1213 ;
0020  1214 ;       DEFINE REMAINING SERVICES
0020  1215 ;
0020  1216
0020  1217          GSYSSRV ADJSTK,K,3,-                ;ADJUST OUTER MODE STACK POINTER
0020  1218                  <R2,R3,R4,R5,R6>,-          ;REGISTERS R2-R6
0020  1219                  EXC_MASK                    ;EXCEPTION MASK
0028  1220          GSYSSRV ADJWSL,K,2,-                ;ADJUST WORKING SET LIMIT
0028  1221                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
0030  1222          GSYSSRV ALCDNP,K,4,-                ;ALLOCATE DIAGNOSTIC PAGE
0030  1223                  <R2,R3,R4,R5,R6,R7>         ;REGISTERS R2-R7
0038  1224          GSYSSRV ALLOC,K,4,-                 ;ALLOCATE DEVICE
0038  1225                  <R2,R3,R4,R5,R6>            ;REGISTERS R2-R6
0040  1226          GSYSSRV ASCEFC,K,4,-                ;ASSOCIATE COMMON EVENT FLAG CLUSTER
0040  1227                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0048  1228          GSYSSRV ASCTIM,ALL,3,-              ;CONVERT TO ASCII TIME
0048  1229                  <R2,R3,R4,R5,R6>            ;REGISTERS R2-R6
0050  1230          GSYSSRV ASSIGN,K,4,-                ;ASSIGN I/O CHANNEL
0050  1231                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0058  1232          GSYSSRV BINTIM,ALL,2,-              ;CONVERT TO BINARY TIME
0058  1233                  <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
0060  1234          GSYSSRV CANCEL,K,1,-                ;CANCEL I/O ON CHANNEL
0060  1235                  <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
0068  1236          GSYSSRV CANTIM,K,2,-                ;CANCEL TIMER REQUEST
0068  1237                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
0070  1238          GSYSSRV CANWAK,K,2,-                ;CANCEL WAKE UP REQUESTS
0070  1239                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
0078  1240          GSYSSRV CRMPSC,K,12,-               ;CREATE AND MAP SECTION
0078  1241                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0080  1242          GSYSSRV CLRPAR,K,2,-                ;CLEAR HARD PARITY ERROR
0080  1243                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
0088  1244          GSYSSRV CMEXEC,E,2,-                ;CHANGE MODE TO EXECUTIVE
0088  1245                  <R4>                        ;REGISTER R4
0090  1246          GSYSSRV CMKRNL,K,2,-                ;CHANGE MODE TO KERNEL
0090  1247                  <R4>                        ;REGISTER R4
0098  1248          GSYSSRV CLREF,K,1,-                 ;CLEAR EVENT FLAG
0098  1249                  <R2,R3,R4,R5>               ;REGISTERS R2-R5. SEE WAITFR COMMENTS.
00A0  1250          GSYSSRV CNTREG,K,4,-                ;CONTRACT REGION
00A0  1251                  <R2,R3,R4,R5,R6,R7>         ;REGISTERS R2-R7
00A8  1252          GSYSSRV GETPTI,K,5,-                ;GET PAGE TABLE INFORMATION
00A8  1253                  <R2,R3,R4,R5,R6,R7,R8,R9,R10> ;REGISTERS R2-R10
00B0  1254          GSYSSRV CRELOG,ALL,4,-              ;CREATE LOGICAL NAME
00B0  1255                  <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
00B8  1256          GSYSSRV CREMBX,K,7,-                ;CREATE MAILBOX
00B8  1257                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
00C0  1258          GSYSSRV CREPRC,K,12,-               ;CREATE PROCESS
00C0  1259                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
00C8  1260          GSYSSRV CRETVA,K,3,-                ;CREATE VIRTUAL ADDRESS
00C8  1261                  <R2,R3,R4,R5,R6,R7,R8>,-    ;REGISTERS R2-R8
00C8  1262                  EXC_MASK                    ;EXCEPTION MASK
00D0  1263          GSYSSRV DACEFC,K,1,-                ;DISASSOCIATE EVENT FLAG CLUSTER
00D0  1264                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
00D8  1265          GSYSSRV DALLOC,K,2,-                ;DEALLOCATE DEVICE
00D8  1266                  <R2,R3,R4,R5,R8>            ;REGISTERS R2-R5,R8
00E0  1267          GSYSSRV DASSGN,K,1,-                ;DEASSIGN I/O CHANNEL
00E0  1268                  <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
00E8  1269          GSYSSRV DCLAST,K,3,-                ;DECLARE AST SYSTEM SERVICE
```

```
                                                I 3
00E8  1270                      <R2,R3,R4,R5>                   ;REGISTERS R2-R5
00F0  1271          GSYSSRV DCLEXH,K,1,-                        ;DECLARE EXIT HANDLER
00F0  1272                      <R2,R3,R4>                      ;REGISTERS R2-R4
00F8  1273          GSYSSRV DELLOG,ALL,3,-                      ;DELETE LOGICAL NAME
00F8  1274                      <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0100  1275          GSYSSRV DELMBX,K,1,-                        ;DELETE MAILBOX
0100  1276                      <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0108  1277          GSYSSRV DELPRC,K,2,-                        ;DELETE PROCESS
0108  1278                      <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R5
0110  1279          GSYSSRV DELTVA,K,3,-                        ;DELETE VIRTUAL ADDRESS
0110  1280                      <R2,R3,R4,R5,R6,R7>,-           ;REGISTERS R2-R7
0110  1281                      EXC_MASK                        ;EXCEPTION MASK
0118  1282          GSYSSRV DGBLSC,K,3,-                        ;DELETE GLOBAL SECTION
0118  1283                      <R2,R3,R4,R5,R6,R7,R8,R9,R10> ;REGISTERS R2-R10
0120  1284          GSYSSRV DLCDNP,K,2,-                        ;DEALLOCATE DIAGNOSTIC PAGE
0120  1285                      <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R7
0128  1286          GSYSSRV DLCEFC,K,1,-                        ;DELETE COMMON EVENT CLUSTER
0128  1287                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0130  1288          GSYSSRV UPDSEC,K,8,-                        ;UPDATE SECTION FILE
0130  1289                      <R2,R3,R4,R5,R6,R7,R8>          ;R2-R8
0138  1290          GSYSSRV SNDERR,K,1,-                        ;SEND MSG TO ERROR LOGGER
0138  1291                      <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0140  1292          GSYSSRV EXIT,K,1,-                          ;IMAGE EXIT
0140  1293                      <R4>,0                          ;REGISTER R4, ALWAYS ALLOWED!
0148  1294          GSYSSRV EXPREG,K,4,-                        ;EXPAND PROGRAM REGION
0148  1295                      <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0150  1296          GSYSSRV FAO,ALL,0,-                         ;FORMAT ASCII OUTPUT
0150  1297                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0158  1298          GSYSSRV FAOL,ALL,0,-                        ;FORMAT ASCII OUTPUT WITH VALUE LIST
0158  1299                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0160  1300          GSYSSRV FORCEX,K,3,-                        ;FORCE EXIT
0160  1301                      <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0168  1302          GSYSSRV IMGSTA,ALL,6,-                      ;IMAGE STARTUP
0168  1303                      <>                              ;REGISTERS NONE
0170  1304          GSYSSRV SNDJBC,E,7,-                        ;SEND TO JOB CONTROLLER
0170  1305                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0178  1306          GSYSSRV GETTIM,E,1,-                        ;GET TIME
0178  1307                      <>                              ;NO REGISTERS
0180  1308          GCOMPSRVB UPDSECW,-                         ;UPDATE SECTION AND WAIT
0180  1309                      <UPDSEC_MASK ! GETJPI_SYNCH_MASK>
0180  1317          GCOMPSRVE       1
0188  1318          GSYSSRV HIBER,K,0,-                         ;HIBERNATE
0188  1319                      <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0190  1320          GSYSSRV IMGACT,E,8,-                        ;IMAGE ACTIVATION
0190  1321                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0198  1322          GSYSSRV LCKPAG,K,3,-                        ;LOCK PAGE IN MEMORY
0198  1323                      <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
01A0  1324          GSYSSRV LKWSET,K,3,-                        ;LOCK PAGES IN WORKING SET
01A0  1325                      <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
01A8  1326          GSYSSRV MGBLSC,K,7,-                        ;MAP GLOBAL SECTION
01A8  1327                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
01B0  1328          GSYSSRV PURGWS,K,1,-                        ;PURGE WORKING SET
01B0  1329                      <R2,R3,R4,R5,R6,R7,R8>          ;R2-R8
01B8  1330          GSYSSRV NUMTIM,E,2,-                        ;CONVERT TIME TO NUMERIC
01B8  1331                      <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R7
01C0  1332          GSYSSRV SNDOPR,E,2,-                        ;SEND MSG TO OPERATOR
01C0  1333                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>       ;REGISTERS R2-R11
```

```
                                                                ;  3
01C8  1334          GSYSSRV QIO,K,12,-                          ;QUEUE I/O REQUEST
01C8  1335                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
01D0  1336          GSYSSRV READEF,K,2,-                        ;READ EVENT FLAG
01D0  1337                  <R2,R3,R4,R5>                       ;REGISTERS R2-R5
01D8  1338          GSYSSRV RESUME,K,2,-                        ;RESUME PROCESS
01D8  1339                  <R2,R3,R4,R5>                       ;REGISTERS R2-R5
01E0  1340          GSYSSRV RUNDWN,K,1,-                        ;RUNDOWN
01E0  1341                  <R2,R3,R4,R5,R6,R7>                 ;REGISTERS R2-R7
01E8  1342          GSYSSRV SNDSMB,E,2,-                        ;SEND MSG TO SYMBIONT MANAGER
01E8  1343                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>       ;REGISTERS R2-R11
01F0  1344          GSYSSRV SCHDWK,K,4,-                        ;SCHEDULE WAKEUP
01F0  1345                  <R2,R3,R4,R5,R6,R7,R8,R9> ;REGISTERS R2-R9
01F8  1346          GSYSSRV SETAST,K,1,-                        ;SET AST ENABLE SERVICE
01F8  1347                  <R2,R3,R4,R5>                       ;REGISTERS R2-R5
0200  1348          GSYSSRV SETEF,K,1,-                         ;SET EVENT FLAG
0200  1349                  <R2,R3,R4,R5>                       ;REGISTERS R2-R5. SEE WAITFR COMMENTS.
0208  1350          GSYSSRV SETEXV,K,4,-                        ;SET EXCEPTION VECTOR
0208  1351                  <R2,R3,R4,R5>                       ;REGISTERS R2-R5
0210  1352          GSYSSRV SETPRN,K,1,-                        ;SET PROCESS  NAME
0210  1353                  <R2,R3,R4,R5,R6,R7,R8,R9> ;REGISTERS R2-R9
0218  1354          GSYSSRV SETPRA,K,2,-                        ;SET POWER RECOVERY AST
0218  1355                  <R2,R3,R4,R5>                       ;REGISTERS R2-R5
0220  1356          GSYSSRV SETIMR,K,4,-                        ;SET TIMER
0220  1357                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0228  1358          GSYSSRV SETPRI,K,4,-                        ;SET PROCESS PRIORITY
0228  1359                  <R2,R3,R4,R5>                       ;REGISTERS R2-R5
0230  1360          GSYSSRV SETPRT,K,5,-                        ;SET PAGE PROTECTION
0230  1361                  <R2,R3,R4,R5,R6,R7,R8,R9> ;REGISTERS R2-R9
0238  1362          GSYSSRV SETRWM,K,1,-                        ;SET RESOURCE WAIT MODE
0238  1363                  <R4>                                ;REGISTER R4
0240  1364          GSYSSRV SETSFM,K,1,-                        ;SET SYSTEM SERVICE FAILURE MODE
0240  1365                  <R4>,EXC_MASK                       ;REGISTER R4, AND EXECPTION MASK
0248  1366          GSYSSRV SETSWM,K,1,-                        ;SET PROCESS SWAP MODE
0248  1367                  <R4>                                ;REGISTER R4
0250  1368          GSYSSRV SUSPND,K,2,-                        ;SUSPEND PROCESS
0250  1369                  <R2,R3,R4,R5>                       ;REGISTERS R2-R5
0258  1370          GSYSSRV TRNLOG,ALL,6,-                      ;TRANSLATE LOGICAL NAME
0258  1371                  <R2,R3,R4,R5,R6,R7,R8>              ;REGISTERS R2-R8
0260  1372          GSYSSRV ULKPAG,K,3,-                        ;UNLOCK PAGE FROM MEMORY
0260  1373                  <R2,R3,R4,R5,R6,R7,R8>              ;REGISTERS R2-R8
0268  1374          GSYSSRV ULWSET,K,3,-                        ;UNLOCK PAGES FROM WORKING SET
0268  1375                  <R2,R3,R4,R5,R6,R7,R8>              ;REGISTERS R2-R8
0270  1376          GSYSSRV UNWIND,ALL,2,-                      ;UNWIND PROCEDURE CALL STACK
0270  1377                  <R2,R3,R4,R5>                       ;REGISTERS R2-R5
0278  1378          GSYSSRV WAITFR,K,1,-                        ;WAIT FOR EVENT FLAG
0278  1379                  <R2,R3,R4,R5,R6>                    ;REGISTERS R2-R6.  IF R8 IS EVER USED
0280  1380                                                      ;THE RMS SYCHRONIZATION CODE MUST BE
0280  1381                                                      ;MODIFIED TO SAVE IT ALSO.
0280  1382          GSYSSRV WAKE,K,2,-                          ;WAKE PROCESS
0280  1383                  <R2,R3,R4,R5>                       ;REGISTERS R2-R5
0288  1384          GSYSSRV WFLAND,K,2,-                        ;WAIT FOR LOGICAL AND OF EVENT FLAGS
0288  1385                  <R2,R3,R4,R5,R6>                    ;REGISTERS R2-R6
0290  1386          GSYSSRV WFLOR,K,2,-                         ;WAIT FOR LOGICAL OR OF EVENT FLAGS
0290  1387                  <R2,R3,R4,R5,R6>                    ;REGISTERS R2-R5
0298  1388          GSYSSRV BRDCST,ALL,2,-                      ;BROADCAST TO TERMINALS
0298  1389                  <R2,R3,R4,R5,R6>                    ;REGISTERS R2-R6
02A0  1390          GSYSSRV DCLCMH,K,3,-                        ;DECLARE CHANGE MODE HANDLER
```

```
                                                K 3
02A0  1391                      <R4>                        ;SAVE R4
02A8  1392              GSYSSRV SETPFM,K,4,-                ;SET PAGE FAULT MONITORING
02A8  1393                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
02B0  1394              GSYSSRV GETMSG,ALL,5,-              ;GET MESSAGE
02B0  1395                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
02B8  1396              GSYSSRV DERLMB,K,1,-                ;DECLARE ERROR LOG MAILBOX
02B8  1397                      <R2,R3,R4,R5>               ;REGISTERS R2-R5
02C0  1398              GSYSSRV CANEXH,K,1,-                ;CANCEL EXIT HANDLER
02C0  1399                      <R2,R3,R4,R5>               ;REGISTERS R2-R5
02C8  1400              GSYSSRV GETCHN,K,5,-                ;GET CHANNEL INFORMATION
02C8  1401                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
02D0  1402              GSYSSRV GETDEV,K,5,-                ;GET DEVICE INFORMATION
02D0  1403                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
02D8  1404              GSYSSRV GETJPI,K,7,-                ;GET JOB PROCESS INFORMATION
02D8  1405                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
02E0  1406              GSYSSRV PUTMSG,ALL,3,-              ;PUT FORMATTED ERROR MESSAGE
02E0  1407                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
02E8  1408              GSYSSRV EXCMSG,ALL,2,-              ;OUTPUT EXCEPTION SUMMARY MESSAGE
02E8  1409                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
02F0  1410              GSYSSRV SNDACC,E,2,-                ;SEND MSG TO ACOUNTING MANAGER
02F0  1411                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
02F8  1412              GSYSSRV SETIME,K,1,-                ;SET SYSTEM TIME
02F8  1413                      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0300  1414              GSYSSRV SETPRV,K,4,-                ;SET PRIVILEGES
0300  1415                      <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
```

```
                     0308  1417 ;
                     0308  1418 ;       SPECIAL VECTORS FOR AST DELIVERY AND CLEARING
                     0308  1419 ;
                     0308  1420 ;       SYS$CLRAST CLEARS THE CURRENTLY ACTIVE AST STATUS
                     0308  1421 ;
                     0308  1422 ;       SYS$GL_ASTRET CONTAINS THE VALUE OF THE RETURN ADDRESS FROM
                     0308  1423 ;       THE CALL INSTRUCTION USED TO DISPATCH AN AST. THIS VALUE CAN
                     0308  1424 ;       BE USED WHEN SEARCHING UP THE STACK FOR THE AST CALL FRAME.
                     0308  1425 ;
                     0308  1429         .PSECT  $$$0000,QUAD
                     0308  1433         .ALIGN  QUAD
                     0308  1435 SYS$CLRAST::                            ;CLEAR ACTIVE AST
80000310             0308  1436         .BLKL   2                       ;
                     0310  1443         .ALIGN  QUAD                    ;
                     0310  1445 SYS$GL_ASTRET::                         ;
80000314             0310  1446         .BLKL   1
                     0314  1447 SYS$GL_COMMON::                         ;ADDRESS OF CORE COMMON DESCRIPTOR
80000318             0314  1448         .BLKL   1                       ;
                     0318  1454
                     0318  1455 ;
                     0318  1456 ; ENTRY VECTOR FOR CONDITION HANDLER SEARCH. LIB$SIGNAL USES THIS VECTOR
                     0318  1457 ; TO SHARE EXCEPTION'S CODE TO SEARCH FOR AND CALL CONDITION HANDLERS.
                     0318  1458 ; THIS ENTRY IS NOT CALLED; RATHER, IT IS JUMPED TO. NO RETURN IS MADE.
                     0318  1459 ;
                     0318  1460
                     0318  1461         .ALIGN  QUAD
                     0318  1463 SYS$SRCHANDLER::
80000320             0318  1467         .BLKQ   1                       ;RESERVE SPACE
                     0320  1469
                     0320  1471 ;
                     0320  1472 ;  NOTE THAT THE CODE IN PSECT $$$000 AT THIS POINT CANNOT EXCEED 320 (HEX)
                     0320  1473 ;  WITHOUT MODIFYING THE RMS SYNCHRONIZATION CODE WHICH PRECEDES THE RMS
                     0320  1474 ;  VECTORS WHICH CANNOT BE MOVED.
                     0320  1475 ;
                     0320  1476
```

```
0320  1478 ;
0320  1479 ; Set up the base for the RMS service codes. We leave a hole so that
0320  1480 ; other exec mode system services can be defined later in this module.
0320  1481 ; The hole is defined by the offset between ECASCTR and RCASCTR; it
0320  1482 ; is checked with an ASSUME at the end of all service definitions.
0320  1483 ;
0320  1487
```

```
                     0320  1507 ;++
                     0320  1508 ;
                     0320  1509 ;  RMS SERVICES
                     0320  1510 ;
                     0320  1511 ;
                     0320  1512 ;  RMS SYNCHRONIZATION ROUTINE
                     0320  1513 ;
                     0320  1514 ;  THE FOLLOWING ROUTINE IS USED BY THE VARIOUS RMS SERVICES IN ORDER
                     0320  1515 ;  TO AWAIT I/O COMPLETION.  THE ROUTINE IS IN THE VECTOR AREA IN ORDER
                     0320  1516 ;  TO WAIT AT THE CALLER'S MODE,  THUS ALLOWING AST ACTIVITY FOR EITHER
                     0320  1517 ;  USER OR SUPERVISOR MODE, OR BOTH.
                     0320  1518 ;
                     0320  1519 ;  THE FAB/RAB IS CHECKED FOR A LEGAL BLOCK ID, I.E., A 1 OR 3, AND
                     0320  1520 ;  AN ERROR RETURNED IF INVALID.  THE STRUCTURE IS NOT REPROBED.
                     0320  1521 ;
                     0320  1522 ;  NOTE THAT EACH RMS SERVICE VECTOR TERMINATES WITH A BRANCH TO THIS
                     0320  1523 ;  ROUTINE.
                     0320  1524 ;
                     0320  1525 ;  THIS ROUTINE ASSUMES THAT THE FOLLOWING REGISTERS HAVE BEEN SET BY THE
                     0320  1526 ;  EXITING RMS EXEC-LEVEL CODE WHENEVER A STALL IS REQUIRED:
                     0320  1527 ;
                     0320  1528 ;       R3      EFN TO WAIT ON
                     0320  1529 ;       R8      RAB/FAB ADDRESS TO WAIT ON
                     0320  1530 ;       R4      (RMSWAIT_BR ENTRY POINT ONLY, $WAIT SERVICE) FLAG FOR WAIT TYPE
                     0320  1531 ;               (0 = SAME RAB, 1 = DIFFERENT RABS)
                     0320  1532 ;
                     0320  1533 ;--
                     0320  1537          .PSECT  $$$0000,QUAD
80000320             0320  1539          .BLKB   ^X320-<.-VECBASE>
80000368             0320  1616          .BLKB   ^X48                    ;THIS TAKES THE SPACE OF THE CODE
                     0368  1617                                          ;WHEN GENERATING THE GLOBAL SYMBOLS
```

```
0368  1621 ;
0368  1622 ;
0368  1623 ;  DEFINE RMS SERVICES
0368  1624 ;
0368  1629 ;
0368  1630 ;  HIGH USE RECORD OPERATIONS
0368  1631 ;
0368  1632          RMSSRV  DELETE          ;DELETE A RECORD
0370  1633          .NLIST CND
0370  1634          RMSSRV  FIND            ;FIND RECORD
0378  1635          RMSSRV  FREE            ;RELEASE LOCK ON ALL RECORDS
0380  1636          RMSSRV  GET             ;GET A RECORD
0388  1637          RMSSRV  PUT             ;PUT A RECORD
0390  1638          RMSSRV  READ            ;READ A BLOCK
0398  1639          RMSSRV  RELEASE         ;RELEASE LOCK ON NAMED RECORD
03A0  1640          RMSSRV  UPDATE          ;REWRITE EXISTING RECORD
03A8  1646          RMSSRV  WAIT            ;STALL FOR RECORD OPERATION COMPLETE
03B0  1652          RMSSRV  WRITE           ;WRITE BLOCK
03B8  1653 ;
03B8  1654 ;  LOWER USAGE OPERATIONS
03B8  1655 ;
03B8  1656          RMSSRV  CLOSE           ;CLOSE FILE
03C0  1657          RMSSRV  CONNECT         ;CONNECT RAB
03C8  1658          RMSSRV  CREATE          ;CREATE FILE
03D0  1659          RMSSRV  DISCONNECT      ;DISCONNECT RAB
03D8  1660          RMSSRV  DISPLAY         ;DISPLAY FILE INFORMATION
03E0  1661          RMSSRV  ERASE           ;ERASE (DELETE) FILE
03E8  1662          RMSSRV  EXTEND          ;EXTEND FILE ALLOCATION
03F0  1663          RMSSRV  FLUSH           ;FINISH I/O ACTIVITY FOR STREAM
03F8  1664          RMSSRV  MODIFY          ;MODIFY FILE ATTRIBUTES
0400  1665          RMSSRV  NXTVOL          ;NEXT VOLUME
0408  1666          RMSSRV  OPEN            ;OPEN FILE
0410  1667          RMSSRV  REWIND          ;REWIND FILE
0418  1668          RMSSRV  SPACE           ;POSITION FOR TRANSFER
0420  1669          RMSSRV  TRUNCATE        ;TRUNCATE FILE
0428  1670          RMSSRV  ENTER           ;ENTER FILENAME INTO DIRECTORY
0430  1671          RMSSRV  PARSE           ;PARSE FILENAME SPECIFICATION
0438  1672          RMSSRV  REMOVE          ;REMOVE FILENAME FROM DIRECTORY
0440  1673          RMSSRV  RENAME,NARG=4   ;RENAME A FILE
0448  1674          RMSSRV  SEARCH          ;SEARCH A FILE DIRECTORY
0450  1675          RMSSRV  SETDDIR,NARG=3,NOSYNC=1
0458  1676                                  ;SET DEFAULT DIRECTORY STRING
0458  1677          RMSSRV  SETDFPROT,REGS=<R2,R3>,NARG=2,NOSYNC=1
0460  1678                                  ;SET DEFAULT FILE PROTECTION MASK
0460  1679          RMSSRV  SSVEXC,REGS=<>,NOSYNC=1
0468  1680                                  ;GENERATE SYS SERV EXCEPTION
0468  1681          RMSSRV  RMSRUNDWN,NARG=2,NOSYNC=1
0470  1682                                  ;PERFORM RUNDOWN ON RMS FILES
0470  1683          RMSSRV  RMSRUHNDLR,NARG=5,NOSYNC=1
0478  1684                                  ;RMS Recovery Unit Handler
0478  1685          RMSSRV  FILESCAN,NARG=3,NOSYNC=1
0480  1686                                  ;Perform syntax check for file specs
0480  1687 ;
0480  1688 ;  ADD NEW RMS SERVICES IN FRONT OF THIS CODE!
0480  1689 ;
0480  1690 ;  Now we add special non-vector code.  Because of the CASE instruction
0480  1691 ;  used at the front of RMS, this code (and any future additional code)
```

```
                     0480  1692 ; must be the last element of the RMS area.
                     0480  1693 ;
                     0480  1694
                     0480  1695          GCOMPSRVB                   ;Helper branch to error processing
                     0480  1704          GCOMPSRVE       1
                     0488  1705
                     0488  1707
                     0488  1708 ; NOTE:  RMSVECEND MARKS THE END OF THE CURRENTLY DEFINED RMS VECTORS.
                     0488  1709 ;        SSVECREG2 MARKS THE START OF THE SECOND REGION OF SYSTEM
                     0488  1710 ;        SERVICE VECTORS.  THERE IS EMPTY SPACE BETWEEN THESE REGIONS
                     0488  1711 ;        FOR FUTURE RMS VECTORS.  IF NECESSARY, THIS SPACE CAN ALSO
                     0488  1712 ;        BE USED FOR SYSTEM SERVICE VECTORS BY BACKING UP SSVECREG2
                     0488  1713 ;        (TOWARDS THE RMS VECTORS) AND ADDING NEW SYSTEM SERVICE VECTORS
                     0488  1714 ;        BEFORE THE ALREADY DEFINED ONES.  IN OTHER WORDS, THESE TWO
                     0488  1715 ;        VECTOR REGIONS MAY GROW TOWARDS EACH OTHER.  IF THEY COLLIDE,
                     0488  1716 ;        AN ASSEMBLY ERROR IS GENERATED.
                     0488  1717
                     0488  1719          .PSECT  $$$0000,QUAD
                     0488  1723
                     0488  1724 RMSVECEND:
800005C0             0488  1725 .=VECBASE+^X5C0
                     05C0  1726 SSVECREG2:                           ; START OF SYSTEM SERVICE VECTOR REGION 2
                     05C0  1732
```

```
                        .SBTTL  REGION 2 OF SYS. SERV. VECTOR DEFINITIONS

;
; Note: Service codes for exec mode services in this region are
; reserved by the offset defined above between RCASCTR and ECASCTR.
; If the ASSUME at the end of this section breaks, the offset must
; be increased.
;

        GSYSSRV ENQ,K,11,-                      ; ENQUEUE
                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
        GSYSSRV DEQ,K,4,-                       ; DEQUEUE
                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
        GCOMPSRVB ENQW,-                        ; ENQUEUE AND WAIT
                <ENQ_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
        GCOMPSRVE       3                       ; RESERVE 3 QUADWORDS FOR VECTOR
        GSYSSRV SETSSF,K,1,-                    ; SET SYSTEM SERVICE FILTER MASK
                <R4>                            ; REGISTER R4
        GSYSSRV SETSTK,K,3,-                    ; SET STACK LIMITS
                <R2,R3,R4>                      ; REGISTERS R2,R3,R4
        GSYSSRV GETSYI,K,7,-                    ; GET SYSTEM INFORMATION
                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
        GSYSSRV IMGFIX,ALL,0,-                  ; IMAGE ADDRESS RELOCATION FIXUP
                <R2,R3,R4,R5>                   ; REGISTERS R2-R5
        GCOMPSRVB       IMGFIX_2,-              ; ********** TEMP **********
                <0>
        GCOMPSRVE       1                       ; ********** TEMP **********
        GSYSSRV GETDVI,K,8,-                    ; GET DEVICE AND VOLUME INFORMATION
                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
        GCOMPSRVB GETDVIW,-                     ; GET DEVICE INFORMATION AND WAIT
                <GETDVI_MASK ! GETJPI_SYNCH_MASK>
        GCOMPSRVE       1
        GCOMPSRVB GETJPIW,-                     ; GET JOB/PROCESS INFORMATION AND WAIT
                <GETJPI_MASK ! GETJPI_SYNCH_MASK>
        GCOMPSRVE       2
        GCOMPSRVB GETSYIW,-                     ; GET SYSTEM INFORMATION AND WAIT
                <GETSYI_MASK ! GETJPI_SYNCH_MASK>
        GCOMPSRVE       1
        GCOMPSRVB SNDJBCW,-                     ; SEND TO JOB CONTROLLER AND WAIT
                <SNDJBC_MASK ! GETJPI_SYNCH_MASK>
        GCOMPSRVE       1
        GCOMPSRVB SYNCH,-                       ; SYNCHRONIZE EFN AND IOSB
                <WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
        GCOMPSRVE       6                       ; RESERVE 6 QUADWORDS FOR VECTOR
        GSYSSRV ERAPAT,K,3,-                    ; GENERATE A SECURITY  ERASE PATTERN
                <R4>                            ; SAVE R4
        GSYSSRV CRELNT,K,8,-                    ; CREATE LOGICAL NAME TABLE
                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
        GSYSSRV CRELNM,K,5,-                    ; CREATE LOGICAL NAME
                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
        GSYSSRV DELLNM,K,3,-                    ; DELETE LOGICAL NAME
                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
        GSYSSRV TRNLNM,K,5,-                    ; TRANSLATE LOGICAL NAME
                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
        GSYSSRV GETLKI,K,7,-                    ; GET LOCK INFORMATION
                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
        GCOMPSRVB GETLKIW,-                     ; GET LOCK INFORMATION AND WAIT
```

```
                     06A0  1875                <GETLKI_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
                     06A0  1887        GCOMPSRVE       2                ; RESERVE 2 QUADWORDS FOR VECTOR
                     06B0  1888
                     06B0  1889        GSYSSRV ASCTOID,E,3,-            ;ASCII TO IDENTIFIER CONVERSION
                     06B0  1890                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                     06B8  1891        GSYSSRV FINISH_RDB,E,1,-         ;FINISH RDB CONTEXT STREAM
                     06B8  1892                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                     06C0  1893        GSYSSRV IDTOASC,E,6,-            ;IDENTIFIER TO ASCII CONVERSION
                     06C0  1894                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                     06C8  1895        GSYSSRV BRKTHRU,K,11,-           ;BREAK THROUGH WRITES
                     06C8  1896                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                     06D0  1897        GSYSSRV GRANTID,ALL,5,-          ;GRANT IDENTIFIER TO PROCESS
                     06D0  1898                <R2,R3>                  ;REGISTERS R2-R3
                     06D8  1899        GSYSSRV REVOKID,ALL,5,-          ;REVOKE IDENTIFIER FROM PROCESS
                     06D8  1900                <R2,R3>                  ;REGISTERS R2-R3
                     06E0  1901        GSYSSRV CHKPRO,K,1,-             ;GENERAL PROTECTION CHECK ROUTINE
                     06E0  1902                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                     06E8  1903        GCOMPSRVB BRKTHRUW,-             ; BREAK THOUGH WRITE AND WAIT
                     06E8  1904                <BRKTHRU_MASK ! GETJPI_SYNCH_MASK>
                     06E8  1913        GCOMPSRVE       2
                     06F8  1914        GSYSSRV GETQUI,E,7,-             ;GET QUEUE INFORMATION
                     06F8  1915                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                     0700  1916        GCOMPSRVB GETQUIW,-              ;GET QUEUE INFORMATION AND WAIT
                     0700  1917                <GETQUI_MASK ! GETJPI_SYNCH_MASK>
                     0700  1926        GCOMPSRVE       2
                     0710  1927
                     0710  1928 ;
00004028             0710  1929        CJF$KCASCTR = 16424
                     0710  1930 ;
                     0710  1931        LDBSRV  CJF$, ALLJDR,       K,   <R4>
                     0718  1932        LDBSRV  CJF$, ASSJNL,       K,   <R4>
                     0720  1933        LDBSRV  CJF$, CONUIC,       K,   <R4>
                     0728  1934        LDBSRV  CJF$, CREJNL,       K,   <R4>
                     0730  1935        LDBSRV  CJF$, DEALJDR,      K,   <R4>
                     0738  1936        LDBSRV  CJF$, DEASJNL,      ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                     0740  1937        LDBSRV  CJF$, DEASJNL_INT,  K,   <R4>
                     0748  1938        LDBSRV  CJF$, DELJNL,       K,   <R4>
                     0750  1939        LDBSRV  CJF$, DMTJMD,       K,   <R4>
                     0758  1940        LDBSRV  CJF$, DSPJNL,       K,   <R4>
                     0760  1941        LDBSRV  CJF$, GETJNL,       K,   <R4>
                     0768  1942        LDBSRV  CJF$, GETRUI,       K,   <R4>
                     0770  1943        LDBSRV  CJF$, MODFLT,       K,   <R4>
                     0778  1944        LDBSRV  CJF$, POSJNL,       K,   <R4>
                     0780  1945        LDBSRV  CJF$, READJNL,      K,   <R4>
                     0788  1946        LDBSRV  CJF$, RECOVER,      K,   <R4>
                     0790  1947        LDBSRV  CJF$, MNTJMD,       K,   <R4>
                     0798  1948        LDBSRV  CJF$, CRENWV,       K,   <R4>
                     07A0  1949        LDBSRV  CJF$, CONJNLF,      K,   <R4>
                     07A8  1950        LDBSRV  CJF$, DCNJNLF,      K,   <R4>
                     07B0  1951        LDBSRV  CJF$, FORCEJNL,     ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                     07B8  1952        LDBSRV  CJF$, FORCEJNLW,    ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                     07C0  1953        LDBSRV  CJF$, WRITEJNL,     ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                     07C8  1954        LDBSRV  CJF$, WRITEJNLW,    ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                     07D0  1955        LDBSRV  CJF$, GETCJI,       K,   <R4>
                     07D8  1956        LDBSRV  CJF$, DMTJMDW,      K,   <R4>, 4, 5, DMTJMD
                     07E8  1957        LDBSRV  CJF$, MODFLTW,      K,   <R4>, 4, 5, MODFLT
                     07F8  1958        LDBSRV  CJF$, POSJNLW,      K,   <R4>, 4, 5, POSJNL
```

```
                     0808  1959          LDBSRV  CJF$, READJNLW,    K,   <R4>, 4, 5, READJNL
                     0818  1960          LDBSRV  CJF$, RECOVERW,    K,   <R4>, 5, 6, RECOVER
                     0828  1961
                     0828  1962 ;
00004010             0828  1963          RUF$KCASCTR = 16400
                     0828  1964 ;
                     0828  1965          LDBSRV  RUF$,  REENTERRU,    K,  <R2,R3,R4,R5,R6>
                     0830  1966          LDBSRV  RUF$,  STARTRU,      K,  <R2,R3,R4,R5,R6>
                     0838  1967          LDBSRV  RUF$,  PHASE1,       K,  <R2,R3,R4,R5,R6>
                     0840  1968          LDBSRV  RUF$,  PHASE2,       K,  <R2,R3,R4,R5,R6>
                     0848  1969          LDBSRV  RUF$,  CANCELRU,     K,  <R2,R3,R4,R5,R6>
                     0850  1970          LDBSRV  RUF$,  MARKPOINTRU,  K,  <R2,R3,R4,R5,R6>
                     0858  1971          LDBSRV  RUF$,  RESETRU,      K,  <R2,R3,R4,R5,R6>
                     0860  1972          LDBSRV  RUF$,  DCLRUH,       K,  <R2,R3,R4,R5,R6>
                     0868  1973          LDBSRV  RUF$,  CANRUH,       K,  <R2,R3,R4,R5,R6>
                     0870  1974          LDBSRV  RUF$,  RUSTATUS,     K,  <R2,R3,R4,R5,R6>
                     0878  1975 ;
                     0878  1976 ; End Recovery Unit consists of a two-phase commit, so we call each
                     0878  1977 ; phase separately.
                     0878  1978 ;
                     0878  1979          GCOMPSRVB ENDRU, <PHASE1_MASK ! PHASE2_MASK>, RUF$ ; End Recovery Unit
                     0878  1990          GCOMPSRVE       2
                     0888  1991          GSYSSRV MTACCESS,K,6,-          ;Mag tape installation specific access routi
                     0888  1992                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                     0890  1993
                     0890  1994 ;
                     0890  1995 ; End of system service vector definitions. New system services are
                     0890  1996 ; to be added at this point.
                     0890  1997 ;
                     0890  2003
```

0890 2269

```
0890  2345          .END
```

```
$$ARGS                  =  00000008
$$T1                    =  00000024
CAT0                    =  00000001
CAT7                    =  00000080
CJF$ALLJDR                 80000710  G
CJF$ASSJNL                 80000718  G
CJF$CONJNLF                800007A0  G
CJF$CONUIC                 80000720  G
CJF$CREJNL                 80000728  G
CJF$CRENWV                 80000798  G
CJF$DCNJNLF                800007A8  G
CJF$DEALJDR                80000730  G
CJF$DEASJNL                80000738  G
CJF$DEASJNL_INT            80000740  G
CJF$DELJNL                 80000748  G
CJF$DMTJMD                 80000750  G
CJF$DMTJMDW                800007D8  G
CJF$DSPJNL                 80000758  G
CJF$FORCEJNL               800007B0  G
CJF$FORCEJNLW              800007B8  G
CJF$GETCJI                 800007D0  G
CJF$GETJNL                 80000760  G
CJF$GETRUI                 80000768  G
CJF$KCASCTR             =  00004028
CJF$MNTJMD                 80000790  G
CJF$MODFLT                 80000770  G
CJF$MODFLTW                800007E8  G
CJF$POSJNL                 80000778  G
CJF$POSJNLW                800007F8  G
CJF$READJNL                80000780  G
CJF$READJNLW               80000808  G
CJF$RECOVER                80000788  G
CJF$RECOVERW               80000818  G
CJF$WRITEJNL               800007C0  G
CJF$WRITEJNLW              800007C8  G
DEF_MASK                =  00000081
ENQ$_ACMODE             =  00000028
ENQ$_ASTADR             =  0000001C
ENQ$_ASTPRM             =  00000020
ENQ$_BLKAST             =  00000024
ENQ$_EFN                =  00000004
ENQ$_FLAGS              =  00000010
ENQ$_LKMODE             =  00000008
ENQ$_LKSB               =  0000000C
ENQ$_NARGS              =  0000000B
ENQ$_PARID              =  00000018
ENQ$_PROT               =  0000002C
ENQ$_RESNAM             =  00000014
EXC_MASK                =  00000080
GETDVI$_ASTADR          =  00000018
GETDVI$_ASTPRM          =  0000001C
GETDVI$_CHAN            =  00000008
GETDVI$_DEVNAM          =  0000000C
GETDVI$_EFN             =  00000004
GETDVI$_IOSB            =  00000014
GETDVI$_ITMLST          =  00000010
GETDVI$_NARGS           =  00000008
GETDVI$_NULLARG         =  00000020
GETJPI$_ASTADR          =  00000018
GETJPI$_ASTPRM          =  0000001C
GETJPI$_EFN             =  00000004
GETJPI$_IOSB            =  00000014
GETJPI$_ITMLST          =  00000010
GETJPI$_NARGS           =  00000007
GETJPI$_PIDADR          =  00000008
GETJPI$_PRCNAM          =  0000000C
GETLKI$_ASTADR          =  00000014
GETLKI$_ASTPRM          =  00000018
GETLKI$_EFN             =  00000004
GETLKI$_IOSB            =  00000010
GETLKI$_ITMLST          =  0000000C
GETLKI$_LKIDADR         =  00000008
GETLKI$_NARGS           =  00000007
GETLKI$_RESERVED        =  0000001C
GETSYI$_ASTADR          =  00000018
GETSYI$_ASTPRM          =  0000001C
GETSYI$_CSIDADR         =  00000008
GETSYI$_EFN             =  00000004
GETSYI$_IOSB            =  00000014
GETSYI$_ITMLST          =  00000010
GETSYI$_NARGS           =  00000007
GETSYI$_NODENAME        =  0000000C
LIBSWITCH               =  00000001
QIO$_ASTADR             =  00000014
QIO$_ASTPRM             =  00000018
QIO$_CHAN               =  00000008
QIO$_EFN                =  00000004
QIO$_FUNC               =  0000000C
QIO$_IOSB               =  00000010
QIO$_NARGS              =  0000000C
QIO$_P1                 =  0000001C
QIO$_P2                 =  00000020
QIO$_P3                 =  00000024
QIO$_P4                 =  00000028
QIO$_P5                 =  0000002C
QIO$_P6                 =  00000030
RMSVECEND                  80000488
RUF$CANCELRU               80000848  G
RUF$CANRUH                 80000868  G
RUF$DCLRUH                 80000860  G
RUF$ENDRU                  80000878  G
RUF$KCASCTR             =  00004010
RUF$MARKPOINTRU            80000850  G
RUF$PHASE1                 80000838  G
RUF$PHASE2                 80000840  G
RUF$REENTERRU              80000828  G
RUF$RESETRU                80000858  G
RUF$RUSTATUS               80000870  G
RUF$STARTRU                80000830  G
SNDJBC$_ASTADR          =  00000018
SNDJBC$_ASTPRM          =  0000001C
SNDJBC$_EFN             =  00000004
SNDJBC$_FUNC            =  00000008
SNDJBC$_IOSB            =  00000014
```

SNDJBC$_ITMLST	=	00000010		SYS$DISCONNECT		800003D0	G
SNDJBC$_NARGS	=	00000007		SYS$DISPLAY		800003D8	G
SNDJBC$_NULLARG	=	0000000C		SYS$DLCDNP		80000120	G
SSVECREG2		800005C0		SYS$DLCEFC		80000128	G
SYNCH$_EFN	=	00000004		SYS$ENQ		800005C0	G
SYNCH$_IOSB	=	00000008		SYS$ENQW		800005D0	G
SYNCH$_NARGS	=	00000002		SYS$ENTER		80000428	G
SYS$ADJSTK		80000020	G	SYS$ERAPAT		80000670	G
SYS$ADJWSL		80000028	G	SYS$ERASE		800003E0	G
SYS$ALCDNP		80000030	G	SYS$EXCMSG		800002E8	G
SYS$ALLOC		800C0038	G	SYS$EXIT		80000140	G
SYS$ASCEFC		80000040	G	SYS$EXPREG		80000148	G
SYS$ASCTIM		80000048	G	SYS$EXTEND		800003E8	G
SYS$ASCTOID		800006B0	G	SYS$FAO		80000150	G
SYS$ASSIGN		80000050	G	SYS$FAOL		80000158	G
SYS$BINTIM		80000058	G	SYS$FILESCAN		80000478	G
SYS$BRDCST		80000298	G	SYS$FIND		80000370	G
SYS$BRKTHRU		800006C8	G	SYS$FINISH_RDB		800006B8	G
SYS$BRKTHRUW		800006E8	G	SYS$FLUSH		800003F0	G
SYS$CALL_HANDL		80000010	G	SYS$FORCEX		80000160	G
SYS$CANCEL		80000060	G	SYS$FREE		80000378	G
SYS$CANEXH		800002C0	G	SYS$GET		80000380	G
SYS$CANTIM		80000068	G	SYS$GETCHN		800002C8	G
SYS$CANWAK		80000070	G	SYS$GETDEV		800002D0	G
SYS$CHKPRO		800006E0	G	SYS$GETDVI		80000610	G
SYS$CLI		80000018	G	SYS$GETDVIW		80000618	G
SYS$CLOSE		800003B8	G	SYS$GETJPI		800002D8	G
SYS$CLRAST		80000308	G	SYS$GETJPIW		80000620	G
SYS$CLREF		80000098	G	SYS$GETLKI		80000698	G
SYS$CLRPAR		80000080	G	SYS$GETLKIW		800006A0	G
SYS$CMEXEC		80000088	G	SYS$GETMSG		800002B0	G
SYS$CMKRNL		80000090	G	SYS$GETPTI		800000A8	G
SYS$CNTREG		800000A0	G	SYS$GETQUI		800006F8	G
SYS$CONNECT		800003C0	G	SYS$GETQUIW		80000700	G
SYS$CREATE		800003C8	G	SYS$GETSYI		800005F8	G
SYS$CRELNM		80000680	G	SYS$GETSYIW		80000630	G
SYS$CRELNT		80000678	G	SYS$GETTIM		80000178	G
SYS$CRELOG		800000B0	G	SYS$GL_ASTRET		80000310	G
SYS$CREMBX		800000B8	G	SYS$GL_COMMON		80000314	G
SYS$CREPRC		800000C0	G	SYS$GRANTID		800006D0	G
SYS$CRETVA		800000C8	G	SYS$HIBER		80000188	G
SYS$CRMPSC		80000078	G	SYS$IDTOASC		800006C0	G
SYS$DACEFC		800000D0	G	SYS$IMGACT		80000190	G
SYS$DALLOC		800000D8	G	SYS$IMGFIX		80000600	G
SYS$DASSGN		800000E0	G	SYS$IMGFIX_2		80000608	G
SYS$DCLAST		800000E8	G	SYS$IMGSTA		80000168	G
SYS$DCLCMH		800002A0	G	SYS$LCKPAG		80000198	G
SYS$DCLEXH		800000F0	G	SYS$LKWSET		800001A0	G
SYS$DELETE		80000368	G	SYS$MGBLSC		800001A8	G
SYS$DELLNM		80000688	G	SYS$MODIFY		800003F8	G
SYS$DELLOG		800000F8	G	SYS$MTACCESS		80000888	G
SYS$DELMBX		80000100	G	SYS$NUMTIM		800001B8	G
SYS$DELPRC		80000108	G	SYS$NXTVOL		80000400	G
SYS$DELTVA		80000110	G	SYS$OPEN		80000408	G
SYS$DEQ		800005C8	G	SYS$PARSE		80000430	G
SYS$DERLMB		800002B8	G	SYS$PURGWS		800001B0	G
SYS$DGBLSC		80000118	G	SYS$PUT		80000388	G

SYS$PUTMSG	800002E0	G	SYS$WFLAND	80000288	G
SYS$QIO	800001C8	G	SYS$WFLOR	80000290	G
SYS$QIOW	80000000	G	SYS$WRITE	800003B0	G
SYS$READ	80000390	G	UPDSEC$_ACMODE	= 0000000C	
SYS$READEF	800001D0	G	UPDSEC$_ASTADR	= 0000001C	
SYS$RELEASE	80000398	G	UPDSEC$_ASTPRM	= 00000020	
SYS$PEMOVE	80000438	G	UPDSEC$_EFN	= 00000014	
SYS$RENAME	80000440	G	UPDSEC$_INADR	= 00000004	
SYS$RESUME	800001D8	G	UPDSEC$_IOSB	= 00000018	
SYS$REVOKID	800006D8	G	UPDSEC$_NARGS	= 00000008	
SYS$REWIND	80000410	G	UPDSEC$_RETADR	= 00000008	
SYS$RMSRUHNDLR	80000470	G	UPDSEC$_UPDFLG	= 00000010	
SYS$RMSRUNDWN	80000468	G	VECBASE	80000000	
SYS$RUNDWN	800001E0	G			
SYS$SCHDWK	800001F0	G			
SYS$SEARCH	80000448	G			
SYS$SETAST	800001F8	G			
SYS$SETDDIR	80000450	G			
SYS$SETDFPROT	80000458	G			
SYS$SETEF	80000200	G			
SYS$SETEXV	80000208	G			
SYS$SETIME	800002F8	G			
SYS$SETIMR	80000220	G			
SYS$SETPFM	800002A8	G			
SYS$SETPRA	80000218	G			
SYS$SETPRI	80000228	G			
SYS$SETPRN	80000210	G			
SYS$SETPRT	80000230	G			
SYS$SETPRV	80000300	G			
SYS$SETRWM	80000238	G			
SYS$SETSFM	80000240	G			
SYS$SETSSF	800005E8	G			
SYS$SETSTK	800005F0	G			
SYS$SETSWM	80000248	G			
SYS$SNDACC	800002F0	G			
SYS$SNDERR	80000138	G			
SYS$SNDJBC	80000170	G			
SYS$SNDJBCW	80000638	G			
SYS$SNDOPR	800001C0	G			
SYS$SNDSMB	800001E8	G			
SYS$SPACE	80000418	G			
SYS$SRCHANDLER	80000318	G			
SYS$SSVEXC	80000460	G			
SYS$SUSPND	80000250	G			
SYS$SYNCH	80000640	G			
SYS$TRNLNM	80000690	G			
SYS$TRNLOG	80000258	G			
SYS$TRUNCATE	80000420	G			
SYS$ULKPAG	80000260	G			
SYS$ULWSET	80000268	G			
SYS$UNWIND	80000270	G			
SYS$UPDATE	800003A0	G			
SYS$UPDSEC	80000130	G			
SYS$UPDSECW	80000180	G			
SYS$WAIT	800003A8	G			
SYS$WAITFR	80000278	G			
SYS$WAKE	80000280	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	80000890 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC QUAD

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.07	00:00:01.21
Command processing	111	00:00:00.62	00:00:05.49
Pass 1	592	00:00:20.55	00:01:10.23
Symbol table sort	0	00:00:02.02	00:00:06.40
Pass 2	206	00:00:05.66	00:00:20.02
Symbol table output	35	00:00:00.26	00:00:00.80
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	979	00:00:29.22	00:01:44.18

The working set limit was 2100 pages.
183647 bytes (359 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1355 non-local and 0 local symbols.
2346 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
_255DUA28:[SYS.OBJ]LIB.MLB;1	6
_255DUA28:[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$:SYSVECTOR/OBJ=OBJ$:SYSVECTOR MSRC$:LBSW/UPDATE=(ENH$:LBSW)+MSRC$:CMODSSDSP/UPDATE=(ENH$:CMODSSDSP)+EXECML$/LIB
```

WRTMFYPAG
LIS

UCBCREDEL
LIS

USRVECTOR
LIS

SYSVECTOR
LIS

SYSINIT
LIS

VERSION
LIS

SYSINI

UPCASEDAT
LIS

SYSINIT
MAP

SYSWAIT
LIS

TIMESCHDL
LIS