



```

PPPPPPPP      11      SSSSSSSS  YY      YY      SSSSSSSS  VV      VV      EEEEEEEEEEE  CCCCCCCC  TTTTTTTTTT
PPPPPPPP      11      SSSSSSSS  YY      YY      SSSSSSSS  VV      VV      EEEEEEEEEEE  CCCCCCCC  TTTTTTTTTT
PP      PP    1111    SS      SS      YY      YY      SS      VV      VV      EE      EE      CC      CC      TT      TT
PP      PP    1111    SS      SS      YY      YY      SS      VV      VV      EE      EE      CC      CC      TT      TT
PP      PP    11      SS      SS      YY      YY      SS      VV      VV      EE      EE      CC      CC      TT      TT
PP      PP    11      SS      SS      YY      YY      SS      VV      VV      EE      EE      CC      CC      TT      TT
PPPPPPPP      11      SSSSSS      YY      YY      SSSSSS  VV      VV      EEEEEEEE  CCCCCC  TTTT
PPPPPPPP      11      SSSSSS      YY      YY      SSSSSS  VV      VV      EEEEEEEE  CCCCCC  TTTT
PP      PP    11      SS      SS      YY      YY      SS      VV      VV      EE      EE      CC      CC      TT      TT
PP      PP    11      SS      SS      YY      YY      SS      VV      VV      EE      EE      CC      CC      TT      TT
PP      PP    11      SS      SS      YY      YY      SS      VV      VV      EE      EE      CC      CC      TT      TT
PP      PP    11      SS      SS      YY      YY      SS      VV      VV      EE      EE      CC      CC      TT      TT
PP      PP    111111  SSSSSSSS  YY      YY      SSSSSSSS  VV      VV      EEEEEEEEEEE  CCCCCCCC  TTTT
PP      PP    111111  SSSSSSSS  YY      YY      SSSSSSSS  VV      VV      EEEEEEEEEEE  CCCCCCCC  TTTT

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
LLLLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLLLL  IIIIII  SSSSSSSS

```



(1) 487  
(1) 1112  
(1) 1734

Macros for Loadable Services  
SYSTEM SERVICE VECTOR DEFINITION  
REGION 2 OF SYS. SERV. VECTOR DEFINITIONS

```
00000001 0000 1 LIBSWITCH=1 ;GENERATE LIBRARY FORM OF SERVICE VECTOR
00000001 0000 1 P1VSWITCH=1 ;GENERATE P1 SPACE VECTORS
0000 1 .NLIST CND
0000 10 .TITLE SYSSP1 VECTOR - P1 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 $MTACCESS installation specific accessibility
0000 59 routine
0000 60
0000 61 V03-038 DAS0001 David Solomon 20-Feb-1984
0000 62 Implement new design for RMS echo SYSS$INPUT to SYSS$OUTPUT
0000 63 (vs V03-019). Echo is now performed by a caller's mode AST
0000 64 routine declared in RMS\RM$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 : Integrate 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 | : |                                                                     |
| 0000 | 131 | : | V03-024 RAS0147 Ron Schaefer 28-APR-1983                            |
| 0000 | 132 | : | Add \$FILESCAN. Add R8 and R9 to \$SETPRN register mask.            |
| 0000 | 133 | : |                                                                     |
| 0000 | 134 | : | V03-023 JLV0244 Jake VanNoy 27-APR-1983                             |
| 0000 | 135 | : | Add \$BRKTHRUW. Change \$BRDCST to all mode service.                |
| 0000 | 136 | : | \$BRDCST now uses \$BRKTHRU to do real work.                        |
| 0000 | 137 | : |                                                                     |
| 0000 | 138 | : | V03-022 LMP0099 L. Mark Pilant, 13-Apr-1983 19:15                   |
| 0000 | 139 | : | Add the \$CHKPRO system service.                                    |
| 0000 | 140 | : |                                                                     |
| 0000 | 141 | : | V03-021 ACG0319 Andrew C. Goldstein, 21-Mar-1983 13:51              |
| 0000 | 142 | : | Add \$GRANTID and \$REVOKID services                                |
| 0000 | 143 | : |                                                                     |
| 0000 | 144 | : | V03-020 JLV0234 Jake VanNoy 1-MAR-1983                              |
| 0000 | 145 | : | Add \$BRKTHRU service.                                              |
| 0000 | 146 | : |                                                                     |
| 0000 | 147 | : | V03-019 RAS0120 Ron Schaefer 25-Feb-1983                            |
| 0000 | 148 | : | Add support to echo SYSS\$INPUT to SYSS\$OUTPUT.                    |
| 0000 | 149 | : | This involves examining the return code from RMS for \$GET;         |
| 0000 | 150 | : | if the special status RMS\$ ECHO (not returned to users)            |
| 0000 | 151 | : | is found, then create a RAB on the caller's stack and               |
| 0000 | 152 | : | execute a \$PUT operation to echo the line.                         |
| 0000 | 153 | : | A certain amount of RMS synchronization code was                    |
| 0000 | 154 | : | shuffled around in order to make room for this.                     |
| 0000 | 155 | : |                                                                     |
| 0000 | 156 | : | V03-018 ACG0317 Andrew C. Goldstein, 22-Feb-1983 15:16              |
| 0000 | 157 | : | Fix off-by-one in kernel arg vector                                 |
| 0000 | 158 | : |                                                                     |
| 0000 | 159 | : | V03-017 RSH0004 R. Scott Hanna 10-Feb-1983                          |
| 0000 | 160 | : | Added \$ASCTOID, \$FINISH_RDB, and \$IDTOASC to system service list |
| 0000 | 161 | : |                                                                     |
| 0000 | 162 | : | V03-016 RNG0016 Rod N. Gamache 1-Feb-1983                           |
| 0000 | 163 | : | Added \$GETLKI to system service list                               |
| 0000 | 164 | : |                                                                     |
| 0000 | 165 | : | V03-015 WMC0015 Wayne Cardoza 12-Jan-1983                           |
| 0000 | 166 | : | Put back accidentally deleted space holder for RMS synchronization. |
| 0000 | 167 | : |                                                                     |
| 0000 | 168 | : | V03-014 DMW4023 DMWalp 7-Jan-1983                                   |
| 0000 | 169 | : | Added \$CRELNT, \$CRELNM, \$DELLNM and \$TRNLNM                     |
| 0000 | 170 | : |                                                                     |
| 0000 | 171 | : | V03-013 KDM0033 Kathleen D. Morse 13-Dec-1982                       |
| 0000 | 172 | : | Correct usage of an interlocked instruction to flush                |
| 0000 | 173 | : | the hardware cache queue.                                           |
| 0000 | 174 | : |                                                                     |
| 0000 | 175 | : | V03-012 ROW0146 Ralph O. Weber 6-DEC-1982                           |
| 0000 | 176 | : | Insert routine header comments for INHXP, CHECKARGLIST,             |
| 0000 | 177 | : | and EXE\$CMODKRNLY (MPSS\$CMODKRNLY). Move things around so         |
| 0000 | 178 | : | that EXE\$CMODKRNLY (MPSS\$CMODKRNLY) header comments are near      |
| 0000 | 179 | : | EXE\$CMODKRNLY (MPSS\$CMODKRNLY) and ASTEXIT comments are near      |
| 0000 | 180 | : | ASTEXIT. Make basic kernel-mode .PSECT definition for Y\$CMODK      |
| 0000 | 181 | : | or MP\$CMOD1 immediately after executive mode code so that new      |
| 0000 | 182 | : | code can be inserted in a way that preserves routine headers,       |
| 0000 | 183 | : | conditional assembly, and .PSECT definitions. Backout ROW145,       |
| 0000 | 184 | : | and in its place, correct conditional assembly of BGEQU 10\$        |
| 0000 | 185 | : | after ACCVIO_RET so that it is assembled only for MPCMOD and        |
|      |     | : | 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$EXCPTN) to before ASTEXIT (or
0000 191 : MP$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 $$$_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 $SCHDEF ;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 = 120
00000080 0000 270 CAT7 = 127
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 (SYSS,EXES,RMSS)
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 $$$0000,QUAD
0000 293 .IFF
0000 294 .PSECT $$$0000,QUAD
0000 295 .ENDC
0000 296 .ALIGN QUAD
0000 297 .IF DF LIBSWITCH
0000 298 SYSS'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 :
0000 320 :
0000 321 :
0000 322 :
0000 323 :
0000 324 :
0000 325 :
0000 326 :
0000 327 :
0000 328 :
0000 329 :
0000 330 .MACRO GCOMPSRVB,SRVNAME,REGMSK,PREFIX=SYSS
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 COMPSTR=
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 :
0000 354 :
0000 355 :
0000 356 :
0000 357 :
0000 358 :
0000 359 :
```

GCOMPSRVB - GENERATE COMPOSITE SYSTEM SERVICE ENTRY VECTOR BEGIN

GCOMPSRVB SRVNAME,REGISTER\_MASK[,PREFIX]

WHERE:

SRVNAME - SERVICE NAME LESS ANY PREFIX (SYSS, EXES)

REGISTER\_MASK - SYMBOLIC REGISTER MASK, E.G QIO MASK

PREFIX - IF SUPPLIED, THE PREFIX FOR THE SERVICE NAME.  
IF OMITTED, "SYSS" IS ASSUMED.

GCOMPSRVB,SRVNAME,REGMSK,PREFIX=SYSS

NDF,MPSWITCH

NDF,RMSSWITCH

DF,LIBSWITCH

\$\$\$0000,QUAD

\$\$\$000,QUAD

ALIGN QUAD

IF DF LIBSWITCH

IIF NOT\_BLANK, <SRVNAME>,-

'PREFIX' SRVNAME::

IFF

ENABL LSB

COMPSTR=

IIF NOT\_BLANK, <REGMSK>,-

WORD <REGMSK>

ENDC

ENDC

ENDC ;MPSWITCH

ENDM GCOMPSRVB

GCOMPSRVE - GENERATE COMPOSITE SYSTEM SERVICE ENTRY VECTOR END

GCOMPSRVE QUADWORDS

WHERE:

QUADWORDS - NUMBER OF QUADWORDS TO RESERVE FOR VECTOR



```
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 CMKSC_'SRVNAME==KCASCTR
0000 390 .IFF ;MPSWITCH DEFINED
0000 391 CMKSC_'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 EXES'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 CMESC_ 'SRVNAME=ECASCTR
0000 418 CHME #SRVNAME
0000 419 RET
0000 420 .PSECT Y$MODEN,BYTE
0000 421 .=ECASCTR
0000 422 ASSUME NARG LE 127
0000 423 .BYTE NARG
0000 424 .PSECT Y$MODEX,BYTE
0000 425 .=ECASCTR
0000 426 .BYTE MASK
0000 427 .PSECT Y$MODE,BYTE
0000 428 .SIGNED_WORD EXES' SRVNAME-ECASE+2
0000 429 .ENDC
0000 430 SRVNAME=ECASCTR
0000 431 ECASCTR=ECASCTR+1
0000 432 .ENDC :MPSWITCH
0000 433 .ENDM SRVE
0000 434 :
0000 435 :
0000 436 : MACROS FOR GENERATING RMS SYSTEM VECTORS
0000 437 :
0000 438 .MACRO RMSSRV SRVNAME NARG=1,REGS=<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>,-
0000 439 MASK,NOSYNC=0
0000 440 GSYSSRV SRVNAME,R,NARG,<REGS>,MASK,NOSYNC
0000 441 .ENDM RMSSRV
0000 442 :
0000 443 : SRVR - GENERATE ENTRY FOR RMS SERVICE (EXEC MODE)
0000 444 :
0000 445 .MACRO SRVR SRVNAME,NARG,MASK,NOSYNC
0000 446 .IF NDF,MPSWITCH
0000 447 .IF NDF,RMSSWITCH
0000 448 CMESC_ 'SRVNAME=RCASCTR
0000 449 CHME #SRVNAME
0000 450 .IF EQ NOSYNC
0000 451 .IIF GT <.+2-RMSSYNC>-127,-
0000 452 RMSSYNC=RMSWBR ;RESET BRANCH DESTINATION
0000 453 RMSWBR=.
0000 454 BRB RMSSYNC
0000 455 .IFF
0000 456 RET
0000 457 .ENDC
0000 458 .PSECT Y$MODEN,BYTE
0000 459 .=RCASCTR
0000 460 ASSUME NARG LE 127
0000 461 .BYTE NARG
0000 462 .PSECT Y$MODEX,BYTE
0000 463 .=RCASCTR
0000 464 .BYTE MASK
0000 465 .IFF
0000 466 .PSECT $$$RMSVEC,BYTE,NOWRT
0000 467 .SIGNED_WORD RMSS' 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 :
0000 476 :
0000 477 :
0000 478
0000 479
0000 480
0000 481
0000 482
0000 483
0000 484
0000 485

SRVALL - GENERATE ENTRY FOR ALL MODE SERVICE

.MACRO SRVALL,SRVNAME,NARG,MASK
.IF NDF,MPSWITCH
.IF NDF,RMSSWITCH
JMP @#EXES$SRVNAME+2
.ENDC
.ENDC ;MPSWITCH
.ENDM SRVALL

```

```
0000 487 .SBTTL Macros for Loadable Services
0000 488
0000 489 :
0000 490 :
0000 491 :
0000 492 :
0000 493 :
0000 494 :
0000 495 :
0000 496 :
0000 497 :
0000 498 :
0000 499 :
0000 500 :
0000 501 :
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 :
0000 529 :
0000 530 :
0000 531 :
0000 532 :
0000 533 .MACRO LVEC_K,PREFIX,SERVICE,EFN,IOSB,ALT_CHMK
0000 534 .IF BLANK ALT_CHMK
0000 535 CMK$_SERVICE = PREFIX'KCASCTR
0000 536 .IFF
0000 537 CMK$_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 @#EXESLDB_SYNC
```



```
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 CMESC_'SERVICE = PREFIX'ECASCTR
0000 564 .IFF
0000 565 CMESC_'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 @#EXESLDB_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 @#EXES'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 :
0000 1116 :
0000 1117 :
0000 1118 :
0000 1122 .PSECT $$$0000,QUAD,ABS
7FFEDE00 0000 1126 .:=^X7FFEDE00 ;BIASED IN P1 SPACE
DE00 1132 VECBASE: ;VECTOR AREA BASE
DE00 1133 :
DE00 1134 :
DE00 1135 :
DE00 1136 :
DE00 1137 :
DE00 1138 :
DE00 1139 :
DE00 1140 :
DE00 1141 :
DE00 1142 :
DE00 1143 :
DE00 1144 :
DE00 1145 :
DE00 1146 :
DE00 1147 :
DE00 1154 :
DE10 1158 :
DE10 1159 :
DE10 1160 :
DE10 1161 :
DE10 1162 :
DE10 1163 :
DE10 1164 :
DE10 1165 :
DE10 1166 :
DE10 1167 :
DE10 1168 :
DE10 1169 :
DE10 1172 :
DE10 1185 :
DE18 1190 :
DE18 1191 :
DE18 1192 :
DE18 1193 :
DE18 1194 :
DE18 1195 :
DE18 1196 :
DE18 1197 :
DE18 1198 :
DE18 1199 :
DE18 1201 :
DE18 1206 :
```

QIO AND WAIT COMPOSITE SERVICE

THE QIO AND WAITFR COMPOSITE SERVICE OCCUPIES THE FIRST TWO SYSTEM SERVICE VECTOR POSITIONS. IT IS CONSTRUCTED BY FROM TWO DISCRETE CHMK INSTRUCTIONS, ONE PERFORMING THE QIO AND THE OTHER PERFORMING THE WAITFR, WHICH RELY UPON THE COMPATIBLE ARGUMENT LISTS OF THESE TWO SERVICES. WAITFR HAS A SINGLE ARGUMENT, THE EVENT FLAG, WHICH IS THE FIRST ARGUMENT IN THE QIO ARGUMENT LIST.

GCOMPSRVB QIOW,- ;QIO AND WAIT  
                  <QIO\_MASK ! WAITFR\_MASK ! CLREF\_MASK ! SETEF\_MASK>  
GCOMPSRVE 2 ;RESERVE 2 QUADWORDS FOR VECTOR

CONDITION HANDLER DISPATCH VECTOR

THE FOLLOWING VECTOR IS INCLUDED IN THE SYSTEM VECTOR SPACE SO THAT BOTH HARDWARE-DETECTED (EXCEPTIONS) AND SOFTWARE-DETECTED (SIGNALS) CONDITIONS CAN BE DISPATCHED FROM THE SAME CALL INSTRUCTION. THIS IS NECESSARY SO THAT THE STACK SEARCH ALGORITHM AND THE UNWIND SYSTEM SERVICE CAN DETECT AND PROPERLY PROCESS MULTIPLE ACTIVE SIGNALS AND/OR EXCEPTIONS.

80000010 7FFEDE18 .ALIGN QUAD  
SYSCALL\_HANDL == . - ^X7FFEDE00 + ^X80000000  
                  .BLKQ 1 ;RESERVE SPACE

COMMAND INTERPRETER DISPATCH VECTOR

THE FOLLOWING VECTOR IS INCLUDED IN THE SYSTEM VECTOR SPACE SO THAT DIRECT CALLS CAN BE MADE TO THE CURRENT COMMAND INTERPRETER WITHOUT HAVING TO KNOW THE ADDRESS OF ITS SERVICE ROUTINE.

7FFEDE20 .ALIGN QUAD  
SYSCLI: .BLKQ 1 ;COMMAND INTERPRETER DISPATCH  
                  ;RESERVE SPACE



```
DE20 1213 :  
DE20 1214 :  
DE20 1215 :  
DE20 1216 :  
DE20 1217 :  
DE20 1218 :  
DE20 1219 :  
DE28 1220 :  
DE28 1221 :  
DE30 1222 :  
DE30 1223 :  
DE38 1224 :  
DE38 1225 :  
DE40 1226 :  
DE40 1227 :  
DE48 1228 :  
DE48 1229 :  
DE50 1230 :  
DE50 1231 :  
DE58 1232 :  
DE58 1233 :  
DE60 1234 :  
DE60 1235 :  
DE68 1236 :  
DE68 1237 :  
DE70 1238 :  
DE70 1239 :  
DE78 1240 :  
DE78 1241 :  
DE80 1242 :  
DE80 1243 :  
DE88 1244 :  
DE88 1245 :  
DE90 1246 :  
DE90 1247 :  
DE98 1248 :  
DE98 1249 :  
DEA0 1250 :  
DEA0 1251 :  
DEA8 1252 :  
DEA8 1253 :  
DEB0 1254 :  
DEB0 1255 :  
DEB8 1256 :  
DEB8 1257 :  
DECO 1258 :  
DECO 1259 :  
DEC8 1260 :  
DEC8 1261 :  
DEC8 1262 :  
DED0 1263 :  
DED0 1264 :  
DED8 1265 :  
DED8 1266 :  
DEEO 1267 :  
DEEO 1268 :  
DEE8 1269 :  
  
DEFINE REMAINING SERVICES  
  
GSYSSRV ADJSTK,K,3,- :ADJUST OUTER MODE STACK POINTER  
      <R2,R3,R4,R5,R6>,- :REGISTERS R2-R6  
      EXC MASK :EXCEPTION MASK  
GSYSSRV ADJWSL,K,2,- :ADJUST WORKING SET LIMIT  
      <R2,R3,R4,R5> :REGISTERS R2-R5  
GSYSSRV ALCDNP,K,4,- :ALLOCATE DIAGNOSTIC PAGE  
      <R2,R3,R4,R5,R6,R7> :REGISTERS R2-R7  
GSYSSRV ALLOC,K,4,- :ALLOCATE DEVICE  
      <R2,R3,R4,R5,R6> :REGISTERS R2-R6  
GSYSSRV ASCFC,K,4,- :ASSOCIATE COMMON EVENT FLAG CLUSTER  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV ASCTIM,ALL,3,- :CONVERT TO ASCII TIME  
      <R2,R3,R4,R5,R6> :REGISTERS R2-R6  
GSYSSRV ASSIGN,K,4,- :ASSIGN I/O CHANNEL  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV BINTIM,ALL,2,- :CONVERT TO BINARY TIME  
      <R2,R3,R4,R5,R6,R7,R8> :REGISTERS R2-R8  
GSYSSRV CANCEL,K,1,- :CANCEL I/O ON CHANNEL  
      <R2,R3,R4,R5,R6,R7,R8> :REGISTERS R2-R8  
GSYSSRV CANTIM,K,2,- :CANCEL TIMER REQUEST  
      <R2,R3,R4,R5> :REGISTERS R2-R5  
GSYSSRV CANWAK,K,2,- :CANCEL WAKE UP REQUESTS  
      <R2,R3,R4,R5> :REGISTERS R2-R5  
GSYSSRV CRMPSC,K,12,- :CREATE AND MAP SECTION  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV CLRPAR,K,2,- :CLEAR HARD PARITY ERROR  
      <R2,R3,R4,R5> :REGISTERS R2-R5  
GSYSSRV CMEXEC,E,2,- :CHANGE MODE TO EXECUTIVE  
      <R4> :REGISTER R4  
GSYSSRV CMKRNK,K,2,- :CHANGE MODE TO KERNEL  
      <R4> :REGISTER R4  
GSYSSRV CLREF,K,1,- :CLEAR EVENT FLAG  
      <R2,R3,R4,R5> :REGISTERS R2-R5. SEE WAITFR COMMENTS.  
GSYSSRV CNTREG,K,4,- :CONTRACT REGION  
      <R2,R3,R4,R5,R6,R7> :REGISTERS R2-R7  
GSYSSRV GETPTI,K,5,- :GET PAGE TABLE INFORMATION  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10> :REGISTERS R2-R10  
GSYSSRV CRELOG,ALL,4,- :CREATE LOGICAL NAME  
      <R2,R3,R4,R5,R6,R7,R8> :REGISTERS R2-R8  
GSYSSRV CREMBX,K,7,- :CREATE MAILBOX  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV CREPRC,K,12,- :CREATE PROCESS  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV CREIVA,K,3,- :CREATE VIRTUAL ADDRESS  
      <R2,R3,R4,R5,R6,R7,R8>,- :REGISTERS R2-R8  
      EXC MASK :EXCEPTION MASK  
GSYSSRV DACEFC,K,1,- :DISASSOCIATE EVENT FLAG CLUSTER  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV DALLOC,K,2,- :DEALLOCATE DEVICE  
      <R2,R3,R4,R5,R8> :REGISTERS R2-R5,R8  
GSYSSRV DASSGN,K,1,- :DEASSIGN I/O CHANNEL  
      <R2,R3,R4,R5,R6,R7,R8> :REGISTERS R2-R8  
GSYSSRV DCLAST,K,3,- :DECLARE AST SYSTEM SERVICE
```



|      |      |           |                                   |                                      |
|------|------|-----------|-----------------------------------|--------------------------------------|
| DEE8 | 1270 |           | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                     |
| DEFO | 1271 | GSYSSRV   | DCLXHM,K,1,-                      | :DECLARE EXIT HANDLER                |
| DEFO | 1272 |           | <R2,R3,R4>                        | :REGISTERS R2-R4                     |
| DEF8 | 1273 | GSYSSRV   | DELLOG,ALL,3,-                    | :DELETE LOGICAL NAME                 |
| DEF8 | 1274 |           | <R2,R3,R4,R5,R6,R7,R8>            | :REGISTERS R2-R8                     |
| DF00 | 1275 | GSYSSRV   | DELMBOX,K,1,-                     | :DELETE MAILBOX                      |
| DF00 | 1276 |           | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                     |
| DF08 | 1277 | GSYSSRV   | DELPRC,K,2,-                      | :DELETE PROCESS                      |
| DF08 | 1278 |           | <R2,R3,R4,R5,R6,R7>               | :REGISTERS R2-R5                     |
| DF10 | 1279 | GSYSSRV   | DELIVA,K,3,-                      | :DELETE VIRTUAL ADDRESS              |
| DF10 | 1280 |           | <R2,R3,R4,R5,R6,R7>,-             | :REGISTERS R2-R7                     |
| DF10 | 1281 |           | EXC MASK                          | :EXCEPTION MASK                      |
| DF18 | 1282 | GSYSSRV   | DGBLSC,K,3,-                      | :DELETE GLOBAL SECTION               |
| DF18 | 1283 |           | <R2,R3,R4,R5,R6,R7,R8,R9,R10>     | :REGISTERS R2-R10                    |
| DF20 | 1284 | GSYSSRV   | DLCDNP,K,2,-                      | :DEALLOCATE DIAGNOSTIC PAGE          |
| DF20 | 1285 |           | <R2,R3,R4,R5,R6,R7>               | :REGISTERS R2-R7                     |
| DF28 | 1286 | GSYSSRV   | DLCEFC,K,1,-                      | :DELETE COMMON EVENT CLUSTER         |
| DF28 | 1287 |           | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | :REGISTERS R2-R11                    |
| DF30 | 1288 | GSYSSRV   | UPDSEC,K,8,-                      | :UPDATE SECTION FILE                 |
| DF30 | 1289 |           | <R2,R3,R4,R5,R6,R7,R8>            | :R2-R8                               |
| DF38 | 1290 | GSYSSRV   | SNDERR,K,1,-                      | :SEND MSG TO ERROR LOGGER            |
| DF38 | 1291 |           | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                     |
| DF40 | 1292 | GSYSSRV   | EXIT,K,1,-                        | :IMAGE EXIT                          |
| DF40 | 1293 |           | <R4>,0                            | :REGISTER R4, ALWAYS ALLOWED!        |
| DF48 | 1294 | GSYSSRV   | EXPREG,K,4,-                      | :EXPAND PROGRAM REGION               |
| DF48 | 1295 |           | <R2,R3,R4,R5,R6,R7,R8>            | :REGISTERS R2-R8                     |
| DF50 | 1296 | GSYSSRV   | FAO,ALL,0,-                       | :FORMAT ASCII OUTPUT                 |
| DF50 | 1297 |           | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | :REGISTERS R2-R11                    |
| DF58 | 1298 | GSYSSRV   | FAOL,ALL,0,-                      | :FORMAT ASCII OUTPUT WITH VALUE LIST |
| DF58 | 1299 |           | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | :REGISTERS R2-R11                    |
| DF60 | 1300 | GSYSSRV   | FORCEX,K,3,-                      | :FORCE EXIT                          |
| DF60 | 1301 |           | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                     |
| DF68 | 1302 | GSYSSRV   | IMGSTA,ALL,6,-                    | :IMAGE STARTUP                       |
| DF68 | 1303 |           | <>                                | :REGISTERS NONE                      |
| DF70 | 1304 | GSYSSRV   | SNDJBC,E,7,-                      | :SEND TO JOB CONTROLLER              |
| DF70 | 1305 |           | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | :REGISTERS R2-R11                    |
| DF78 | 1306 | GSYSSRV   | GETTIM,E,1,-                      | :GET TIME                            |
| DF78 | 1307 |           | <>                                | :NO REGISTERS                        |
| DF80 | 1308 | GCOMPSRV  | UPDSECW,-                         | :UPDATE SECTION AND WAIT             |
| DF80 | 1309 |           | <UPDSEC_MASK ! GETJPI_SYNCH_MASK> |                                      |
| DF80 | 1317 | GCOMPSRVE | 1                                 |                                      |
| DF88 | 1318 | GSYSSRV   | HIBER,K,0,-                       | :HIBERNATE                           |
| DF88 | 1319 |           | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                     |
| DF90 | 1320 | GSYSSRV   | IMGACT,E,8,-                      | :IMAGE ACTIVATION                    |
| DF90 | 1321 |           | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | :REGISTERS R2-R11                    |
| DF98 | 1322 | GSYSSRV   | LCKPAG,K,3,-                      | :LOCK PAGE IN MEMORY                 |
| DF98 | 1323 |           | <R2,R3,R4,R5,R6,R7,R8>            | :REGISTERS R2-R8                     |
| DFA0 | 1324 | GSYSSRV   | LKWSET,K,3,-                      | :LOCK PAGES IN WORKING SET           |
| DFA0 | 1325 |           | <R2,R3,R4,R5,R6,R7,R8>            | :REGISTERS R2-R8                     |
| DFA8 | 1326 | GSYSSRV   | MGBLSC,K,7,-                      | :MAP GLOBAL SECTION                  |
| DFA8 | 1327 |           | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | :REGISTERS R2-R11                    |
| DFB0 | 1328 | GSYSSRV   | PURGWS,K,1,-                      | :PURGE WORKING SET                   |
| DFB0 | 1329 |           | <R2,R3,R4,R5,R6,R7,R8>            | :R2-R8                               |
| DFB8 | 1330 | GSYSSRV   | NUMTIM,E,2,-                      | :CONVERT TIME TO NUMERIC             |
| DFB8 | 1331 |           | <R2,R3,R4,R5,R6,R7>               | :REGISTERS R2-R7                     |
| DFC0 | 1332 | GSYSSRV   | SNDOPR,E,2,-                      | :SEND MSG TO OPERATOR                |
| DFC0 | 1333 |           | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | :REGISTERS R2-R11                    |



|      |      |         |                                   |                                        |
|------|------|---------|-----------------------------------|----------------------------------------|
| DFC8 | 1334 | GSYSSRV | QIO,K,12,-                        | :QUEUE I/O REQUEST                     |
| DFC8 | 1335 |         | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | :REGISTERS R2-R11                      |
| DFD0 | 1336 | GSYSSRV | READEP,K,2,-                      | :READ EVENT FLAG                       |
| DFD0 | 1337 |         | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                       |
| DFD8 | 1338 | GSYSSRV | RESUME,K,2,-                      | :RESUME PROCESS                        |
| DFD8 | 1339 |         | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                       |
| DFE0 | 1340 | GSYSSRV | RUNDWN,K,1,-                      | :RUNDOWN                               |
| DFE0 | 1341 |         | <R2,R3,R4,R5,R6,R7>               | :REGISTERS R2-R7                       |
| DFE8 | 1342 | GSYSSRV | 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 | GSYSSRV | SCHDWK,K,4,-                      | :SCHEDULE WAKEUP                       |
| DFF0 | 1345 |         | <R2,R3,R4,R5,R6,R7,R8,R9>         | :REGISTERS R2-R9                       |
| DFF8 | 1346 | GSYSSRV | SETAST,K,1,-                      | :SET AST ENABLE SERVICE                |
| DFF8 | 1347 |         | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                       |
| E000 | 1348 | GSYSSRV | SETEF,K,1,-                       | :SET EVENT FLAG                        |
| E000 | 1349 |         | <R2,R3,R4,R5>                     | :REGISTERS R2-R5. SEE WAITFR COMMENTS. |
| E008 | 1350 | GSYSSRV | SETXV,K,4,-                       | :SET EXCEPTION VECTOR                  |
| E008 | 1351 |         | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                       |
| E010 | 1352 | GSYSSRV | SETPRN,K,1,-                      | :SET PROCESS NAME                      |
| E010 | 1353 |         | <R2,R3,R4,R5,R6,R7,R8,R9>         | :REGISTERS R2-R9                       |
| E018 | 1354 | GSYSSRV | SETPRA,K,2,-                      | :SET POWER RECOVERY AST                |
| E018 | 1355 |         | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                       |
| E020 | 1356 | GSYSSRV | SETIMR,K,4,-                      | :SET TIMER                             |
| E020 | 1357 |         | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | :REGISTERS R2-R11                      |
| E028 | 1358 | GSYSSRV | SETPRI,K,4,-                      | :SET PROCESS PRIORITY                  |
| E028 | 1359 |         | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                       |
| E030 | 1360 | GSYSSRV | SETPRT,K,5,-                      | :SET PAGE PROTECTION                   |
| E030 | 1361 |         | <R2,R3,R4,R5,R6,R7,R8,R9>         | :REGISTERS R2-R9                       |
| E038 | 1362 | GSYSSRV | SETRWM,K,1,-                      | :SET RESOURCE WAIT MODE                |
| E038 | 1363 |         | <R4>                              | :REGISTER R4                           |
| E040 | 1364 | GSYSSRV | SETSFM,K,1,-                      | :SET SYSTEM SERVICE FAILURE MODE       |
| E040 | 1365 |         | <R4>,EXC_MASK                     | :REGISTER R4, AND EXCEPTION MASK       |
| E048 | 1366 | GSYSSRV | SETSWM,K,1,-                      | :SET PROCESS SWAP MODE                 |
| E048 | 1367 |         | <R4>                              | :REGISTER R4                           |
| E050 | 1368 | GSYSSRV | SUSPND,K,2,-                      | :SUSPEND PROCESS                       |
| E050 | 1369 |         | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                       |
| E058 | 1370 | GSYSSRV | TRNLOG,ALL,6,-                    | :TRANSLATE LOGICAL NAME                |
| E058 | 1371 |         | <R2,R3,R4,R5,R6,R7,R8>            | :REGISTERS R2-R8                       |
| E060 | 1372 | GSYSSRV | ULKPAK,K,3,-                      | :UNLOCK PAGE FROM MEMORY               |
| E060 | 1373 |         | <R2,R3,R4,R5,R6,R7,R8>            | :REGISTERS R2-R8                       |
| E068 | 1374 | GSYSSRV | ULWSET,K,3,-                      | :UNLOCK PAGES FROM WORKING SET         |
| E068 | 1375 |         | <R2,R3,R4,R5,R6,R7,R8>            | :REGISTERS R2-R8                       |
| E070 | 1376 | GSYSSRV | UNWIND,ALL,2,-                    | :UNWIND PROCEDURE CALL STACK           |
| E070 | 1377 |         | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                       |
| E078 | 1378 | GSYSSRV | 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 SYNCHRONIZATION CODE MUST BE  |
| E080 | 1381 |         |                                   | :MODIFIED TO SAVE IT ALSO.             |
| E080 | 1382 | GSYSSRV | WAKE,K,2,-                        | :WAKE PROCESS                          |
| E080 | 1383 |         | <R2,R3,R4,R5>                     | :REGISTERS R2-R5                       |
| E088 | 1384 | GSYSSRV | WFLAND,K,2,-                      | :WAIT FOR LOGICAL AND OF EVENT FLAGS   |
| E088 | 1385 |         | <R2,R3,R4,R5,R6>                  | :REGISTERS R2-R6                       |
| E090 | 1386 | GSYSSRV | WFLOR,K,2,-                       | :WAIT FOR LOGICAL OR OF EVENT FLAGS    |
| E090 | 1387 |         | <R2,R3,R4,R5,R6>                  | :REGISTERS R2-R5                       |
| E098 | 1388 | GSYSSRV | BRDCST,ALL,2,-                    | :BROADCAST TO TERMINALS                |
| E098 | 1389 |         | <R2,R3,R4,R5,R6>                  | :REGISTERS R2-R6                       |
| EOA0 | 1390 | GSYSSRV | DCLCMH,K,3,-                      | :DECLARE CHANGE MODE HANDLER           |



|      |      |         |                                   |                                   |
|------|------|---------|-----------------------------------|-----------------------------------|
| EOA0 | 1391 |         | <R4>                              | ;SAVE R4                          |
| EOA8 | 1392 | GSYSSRV | SETPFM,K,4,-                      | ;SET PAGE FAULT MONITORING        |
| EOA8 | 1393 |         | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | ;REGISTERS R2-R11                 |
| EOB0 | 1394 | GSYSSRV | GETMSG,ALL,5,-                    | ;GET MESSAGE                      |
| EOB0 | 1395 |         | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | ;REGISTERS R2-R11                 |
| EOB8 | 1396 | GSYSSRV | DERLMB,K,1,-                      | ;DECLARE ERROR LOG MAILBOX        |
| EOB8 | 1397 |         | <R2,R3,R4,R5>                     | ;REGISTERS R2-R5                  |
| EOC0 | 1398 | GSYSSRV | CANEXH,K,1,-                      | ;CANCEL EXIT HANDLER              |
| EOC0 | 1399 |         | <R2,R3,R4,R5>                     | ;REGISTERS R2-R5                  |
| EOC8 | 1400 | GSYSSRV | GETCHN,K,5,-                      | ;GET CHANNEL INFORMATION          |
| EOC8 | 1401 |         | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | ;REGISTERS R2-R11                 |
| EOD0 | 1402 | GSYSSRV | GETDEV,K,5,-                      | ;GET DEVICE INFORMATION           |
| EOD0 | 1403 |         | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | ;REGISTERS R2-R11                 |
| EOD8 | 1404 | GSYSSRV | GETJPI,K,7,-                      | ;GET JOB PROCESS INFORMATION      |
| EOD8 | 1405 |         | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | ;REGISTERS R2-R11                 |
| EOE0 | 1406 | GSYSSRV | PUTMSG,ALL,3,-                    | ;PUT FORMATTED ERROR MESSAGE      |
| EOE0 | 1407 |         | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | ;REGISTERS R2-R11                 |
| EOE8 | 1408 | GSYSSRV | EXCMMSG,ALL,2,-                   | ;OUTPUT EXCEPTION SUMMARY MESSAGE |
| EOE8 | 1409 |         | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | ;REGISTERS R2-R11                 |
| EOF0 | 1410 | GSYSSRV | SNDACC,E,2,-                      | ;SEND MSG TO ACCOUNTING MANAGER   |
| EOF0 | 1411 |         | <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> | ;REGISTERS R2-R11                 |
| EOF8 | 1412 | GSYSSRV | SETIME,K,1,-                      | ;SET SYSTEM TIME                  |
| EOF8 | 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 1417 :  
E108 1418 : SPECIAL VECTORS FOR AST DELIVERY AND CLEARING  
E108 1419 :  
E108 1420 : SYSSCLRAST CLEARS THE CURRENTLY ACTIVE AST STATUS  
E108 1421 :  
E108 1422 : SYSSGL_ASTRET CONTAINS THE VALUE OF THE RETURN ADDRESS FROM  
E108 1423 : THE CALL INSTRUCTION USED TO DISPATCH AN AST. THIS VALUE CAN  
E108 1424 : BE USED WHEN SEARCHING UP THE STACK FOR THE AST CALL FRAME.  
E108 1425 :  
E108 1429 : .PSECT $$$0000,QUAD  
E108 1433 : .ALIGN QUAD  
E108 1435 SYSSCLRAST:: : CLEAR ACTIVE AST  
7FFEE110 E108 1436 : .BLKL 2  
E110 1443 : .ALIGN QUAD  
E110 1445 SYSSGL_ASTRET:: :  
7FFEE114 E110 1446 : .BLKL 1  
E114 1447 SYSSGL_COMMON:: : ADDRESS OF CORE COMMON DESCRIPTOR  
7FFEE118 E114 1448 : .BLKL 1  
E118 1454 :  
E118 1455 :  
E118 1456 : ENTRY VECTOR FOR CONDITION HANDLER SEARCH. LIB$SIGNAL USES THIS VECTOR  
E118 1457 : TO SHARE EXCEPTION'S CODE TO SEARCH FOR AND CALL CONDITION HANDLERS.  
E118 1458 : THIS ENTRY IS NOT CALLED; RATHER, IT IS JUMPED TO. NO RETURN IS MADE.  
E118 1459 :  
E118 1460 :  
E118 1461 : .ALIGN QUAD  
E118 1463 SYSSSRCHANDLER:: :  
7FFEE120 E118 1467 : .BLKQ 1 : RESERVE SPACE  
E120 1469 :  
E120 1471 :  
E120 1472 : NOTE THAT THE CODE IN PSECT $$$0000 AT THIS POINT CANNOT EXCEED 320 (HEX)  
E120 1473 : WITHOUT MODIFYING THE RMS SYNCHRONIZATION CODE WHICH PRECEDES THE RMS  
E120 1474 : VECTORS WHICH CANNOT BE MOVED.  
E120 1475 :  
E120 1476 :
```



SYSSP1 VECTOR  
V04-000

- P1 SYSTEM SERVICE VECTOR DEFINITIONS<sup>C 4</sup>  
SYSTEM SERVICE VECTOR DEFINITION

16-SEP-1984 00:40:54 VAX/VMS Macro V04-00  
5-SEP-1984 03:40:37 [SYS.SRC]CMODSSDSP.MAR;1

Page 18  
(1)

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 1507 :++
E120 1508 :
E120 1509 : RMS SERVICES
E120 1510 :
E120 1511 :
E120 1512 : RMS SYNCHRONIZATION ROUTINE
E120 1513 :
E120 1514 : THE FOLLOWING ROUTINE IS USED BY THE VARIOUS RMS SERVICES IN ORDER
E120 1515 : TO AWAIT I/O COMPLETION. THE ROUTINE IS IN THE VECTOR AREA IN ORDER
E120 1516 : TO WAIT AT THE CALLER'S MODE, THUS ALLOWING AST ACTIVITY FOR EITHER
E120 1517 : USER OR SUPERVISOR MODE, OR BOTH.
E120 1518 :
E120 1519 : THE FAB/RAB IS CHECKED FOR A LEGAL BLOCK ID, I.E., A 1 OR 3, AND
E120 1520 : AN ERROR RETURNED IF INVALID. THE STRUCTURE IS NOT REPROBED.
E120 1521 :
E120 1522 : NOTE THAT EACH RMS SERVICE VECTOR TERMINATES WITH A BRANCH TO THIS
E120 1523 : ROUTINE.
E120 1524 :
E120 1525 : THIS ROUTINE ASSUMES THAT THE FOLLOWING REGISTERS HAVE BEEN SET BY THE
E120 1526 : EXITING RMS EXEC-LEVEL CODE WHENEVER A STALL IS REQUIRED:
E120 1527 :
E120 1528 : R3      EFN TO WAIT ON
E120 1529 : R8      RAB/FAB ADDRESS TO WAIT ON
E120 1530 : R4      (RMSWAIT BR ENTRY POINT ONLY, $WAIT SERVICE) FLAG FOR WAIT TYPE
E120 1531 :          (0 = SAME RAB, 1 = DIFFERENT RABS)
E120 1532 :
E120 1533 :--
E120 1537 :
7FFEE120 E120 1539 .PSECT $$$0000,QUAD
7FFEE168 E120 1616 .BLKB ^X320-<.-VECBASE>
E168 1617 .BLKB ^X48

```

```

;THIS TAKES THE SPACE OF THE CODE
;WHEN GENERATING THE GLOBAL SYMBOLS

```



```
E168 1621 :  
E168 1622 :  
E168 1623 : DEFINE RMS SERVICES  
E168 1624 :  
E168 1629 :  
E168 1630 : HIGH USE RECORD OPERATIONS  
E168 1631 :  
E168 1632 : RMSSRV DELETE ;DELETE A RECORD  
E170 1633 : .NLIST CND  
E170 1634 : RMSSRV FIND ;FIND RECORD  
E178 1635 : RMSSRV FREE ;RELEASE LOCK ON ALL RECORDS  
E180 1636 : RMSSRV GET ;GET A RECORD  
E188 1637 : RMSSRV PUT ;PUT A RECORD  
E190 1638 : RMSSRV READ ;READ A BLOCK  
E198 1639 : RMSSRV RELEASE ;RELEASE LOCK ON NAMED RECORD  
E1A0 1640 : RMSSRV UPDATE ;REWRITE EXISTING RECORD  
E1A8 1646 : RMSSRV WAIT ;STALL FOR RECORD OPERATION COMPLETE  
E1B0 1652 : RMSSRV WRITE ;WRITE BLOCK  
E1B8 1653 :  
E1B8 1654 : LOWER USAGE OPERATIONS  
E1B8 1655 :  
E1B8 1656 : RMSSRV CLOSE ;CLOSE FILE  
E1C0 1657 : RMSSRV CONNECT ;CONNECT RAB  
E1C8 1658 : RMSSRV CREATE ;CREATE FILE  
E1D0 1659 : RMSSRV DISCONNECT ;DISCONNECT RAB  
E1D8 1660 : RMSSRV DISPLAY ;DISPLAY FILE INFORMATION  
E1E0 1661 : RMSSRV ERASE ;ERASE (DELETE) FILE  
E1E8 1662 : RMSSRV EXTEND ;EXTEND FILE ALLOCATION  
E1F0 1663 : RMSSRV FLUSH ;FINISH I/O ACTIVITY FOR STREAM  
E1F8 1664 : RMSSRV MODIFY ;MODIFY FILE ATTRIBUTES  
E200 1665 : RMSSRV NXTVOL ;NEXT VOLUME  
E208 1666 : RMSSRV OPEN ;OPEN FILE  
E210 1667 : RMSSRV REWIND ;REWIND FILE  
E218 1668 : RMSSRV SPACE ;POSITION FOR TRANSFER  
E220 1669 : RMSSRV TRUNCATE ;TRUNCATE FILE  
E228 1670 : RMSSRV ENTER ;ENTER FILENAME INTO DIRECTORY  
E230 1671 : RMSSRV PARSE ;PARSE FILENAME SPECIFICATION  
E238 1672 : RMSSRV REMOVE ;REMOVE FILENAME FROM DIRECTORY  
E240 1673 : RMSSRV RENAME,NARG=4 ;RENAME A FILE  
E248 1674 : RMSSRV SEARCH ;SEARCH A FILE DIRECTORY  
E250 1675 : RMSSRV SETDDIR,NARG=3,NOSYNC=1  
E258 1676 : ;SET DEFAULT DIRECTORY STRING  
E258 1677 : RMSSRV SETDFPROT,REGS=<R2,R3>,NARG=2,NOSYNC=1  
E260 1678 : ;SET DEFAULT FILE PROTECTION MASK  
E260 1679 : RMSSRV SSVEXC,REGS=<>,NOSYNC=1  
E268 1680 : ;GENERATE SYS SERV EXCEPTION  
E268 1681 : RMSSRV RMSRUNDN,NARG=2,NOSYNC=1  
E270 1682 : ;PERFORM RUNDOWN ON RMS FILES  
E270 1683 : RMSSRV RMSRUHNDLR,NARG=5,NOSYNC=1  
E278 1684 : ;RMS Recovery Unit Handler  
E278 1685 : RMSSRV FILESCAN,NARG=3,NOSYNC=1  
E280 1686 : ;Perform syntax check for file specs  
E280 1687 :  
E280 1688 : ADD NEW RMS SERVICES IN FRONT OF THIS CODE!  
E280 1689 :  
E280 1690 : Now we add special non-vector code. Because of the CASE instruction  
E280 1691 : 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 ; FOR 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

```



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

E3C0 1734
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
E3C0 1744 GSYSSRV ENQ,K,11,- ; ENQUEUE
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
```



```
00004028 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 :
E510 1929 :
E510 1930 :
E510 1931 :
E518 1932 LDBSRV CJFS, ALLJDR, K, <R4>
E520 1933 LDBSRV CJFS, ASSJNL, K, <R4>
E528 1934 LDBSRV CJFS, CONUIC, K, <R4>
E530 1935 LDBSRV CJFS, CREJNL, K, <R4>
E538 1936 LDBSRV CJFS, DEALJDR, K, <R4>
E540 1937 LDBSRV CJFS, DEASJNL, ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
E548 1938 LDBSRV CJFS, DEASJNL_INT, K, <R4>
E550 1939 LDBSRV CJFS, DELJNL, K, <R4>
E558 1940 LDBSRV CJFS, DMTJMD, K, <R4>
E560 1941 LDBSRV CJFS, DSPJNL, K, <R4>
E568 1942 LDBSRV CJFS, GETJNL, K, <R4>
E570 1943 LDBSRV CJFS, GETRUI, K, <R4>
E578 1944 LDBSRV CJFS, MODFLT, K, <R4>
E580 1945 LDBSRV CJFS, POSJNL, K, <R4>
E588 1946 LDBSRV CJFS, READJNL, K, <R4>
E590 1947 LDBSRV CJFS, RECOVER, K, <R4>
E598 1948 LDBSRV CJFS, MNTJMD, K, <R4>
E5A0 1949 LDBSRV CJFS, CRENWV, K, <R4>
E5A8 1950 LDBSRV CJFS, CONJNLF, K, <R4>
E5B0 1951 LDBSRV CJFS, DCNJNLF, K, <R4>
E5B8 1952 LDBSRV CJFS, FORCEJNL, ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
E5C0 1953 LDBSRV CJFS, FORCEJNLW, ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
E5C8 1954 LDBSRV CJFS, WRITEJNL, ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
E5D0 1955 LDBSRV CJFS, WRITEJNLW, ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
E5D8 1956 LDBSRV CJFS, GETCJI, K, <R4>
E5E8 1957 LDBSRV CJFS, DMTJMDW, K, <R4>, 4, 5, DMTJMD
E5F8 1958 LDBSRV CJFS, MODFLTW, K, <R4>, 4, 5, MODFLT
LDBSRV CJFS, POSJNLW, K, <R4>, 4, 5, POSJNL
```



```

00004010 E608 1959      LDBSRV CJF$, READJNLW,    K,  <R4>, 4, 5, READJNL
          E618 1960      LDBSRV CJF$, RECOVERW,   K,  <R4>, 5, 6, RECOVER
          E628 1961
          E628 1962 ;
          E628 1963      RUF$KASCTR = 16400
          E628 1964 ;
          E628 1965      LDBSRV RUF$, 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
          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

```

SYSSP1 VECTOR  
V04-000

- P1 SYSTEM SERVICE VECTOR DEFINITIONS 16-SEP-1984 00:40:54 VAX/VMS Macro V04-00  
REGION 2 OF SYS. SERV. VECTOR DEFINITION 5-SEP-1984 03:40:37 [SYS.SRC]CMODSSDSP.MAR;1  
E690 2269

Page 25  
(1)

PAG  
V04



SYSSP1 VECTOR  
V04-000

- P1 SYSTEM SERVICE VECTOR DEFINITIONS<sup>K 4</sup> 16-SEP-1984 00:40:54 VAX/VMS Macro V04-00  
REGION 2 OF SYS. SERV. VECTOR DEFINITION 5-SEP-1984 03:40:37 [SYS.SRC]CMODSSDSP.MAR;1

Page 26  
(2)

E690 2345 .END

PAG  
V04



|                  |            |                  |            |
|------------------|------------|------------------|------------|
| \$\$ARGS         | = 00000008 | GETDVIS_NULLARG  | = 00000020 |
| \$\$T1           | = 00000024 | GETJPIS_ASTADR   | = 00000018 |
| CATO             | = 00000001 | GETJPIS_ASTPRM   | = 0000001C |
| CAT7             | = 00000080 | GETJPIS_EFN      | = 00000004 |
| CJFSALLJDR       | 7FFEE510 G | GETJPIS_IOSB     | = 00000014 |
| CJFSASSJNL       | 7FFEE518 G | GETJPIS_ITMLST   | = 00000010 |
| CJF\$CONJNLF     | 7FFEE5A0 G | GETJPIS_NARGS    | = 00000007 |
| CJF\$CONUIC      | 7FFEE520 G | GETJPIS_PIDADR   | = 00000008 |
| CJF\$CREJNL      | 7FFEE528 G | GETJPIS_PRCNAM   | = 0000000C |
| CJF\$SCRENWV     | 7FFEE598 G | GETLKIS_ASTADR   | = 00000014 |
| CJF\$DCNJNLF     | 7FFEE5A8 G | GETLKIS_ASTPRM   | = 00000018 |
| CJF\$DEALJDR     | 7FFEE530 G | GETLKIS_EFN      | = 00000004 |
| CJF\$DEASJNL     | 7FFEE538 G | GETLKIS_IOSB     | = 00000010 |
| CJF\$DEASJNL_INT | 7FFEE540 G | GETLKIS_ITMLST   | = 0000000C |
| CJF\$DELJNL      | 7FFEE548 G | GETLKIS_LKIDADR  | = 00000008 |
| CJF\$DMTJMD      | 7FFEE550 G | GETLKIS_NARGS    | = 00000007 |
| CJF\$DMTJMDW     | 7FFEE5D8 G | GETLKIS_RESERVED | = 0000001C |
| CJF\$DSPJNL      | 7FFEE558 G | GETSYIS_ASTADR   | = 00000018 |
| CJF\$FORCEJNL    | 7FFEE5B0 G | GETSYIS_ASTPRM   | = 0000001C |
| CJF\$FORCEJNLW   | 7FFEE5B8 G | GETSYIS_CSIDADR  | = 00000008 |
| CJF\$GETCJI      | 7FFEE5D0 G | GETSYIS_EFN      | = 00000004 |
| CJF\$GETJNL      | 7FFEE560 G | GETSYIS_IOSB     | = 00000014 |
| CJF\$GETRUI      | 7FFEE568 G | GETSYIS_ITMLST   | = 00000010 |
| CJF\$KASCTR      | = 00004028 | GETSYIS_NARGS    | = 00000007 |
| CJF\$MNTJMD      | 7FFEE590 G | GETSYIS_NODENAME | = 0000000C |
| CJF\$MODFLT      | 7FFEE570 G | LIBSWITCH        | = 00000001 |
| CJF\$MODFLTW     | 7FFEE5E8 G | P1VSWITCH        | = 00000001 |
| CJF\$POSJNL      | 7FFEE578 G | QIOS_ASTADR      | = 00000014 |
| CJF\$POSJNLW     | 7FFEE5F8 G | QIOS_ASTPRM      | = 00000018 |
| CJF\$READJNL     | 7FFEE580 G | QIOS_CHAN        | = 00000008 |
| CJF\$READJNLW    | 7FFEE608 G | QIOS_EFN         | = 00000004 |
| CJF\$RECOVER     | 7FFEE588 G | QIOS_FUNC        | = 0000000C |
| CJF\$RECOVERW    | 7FFEE618 G | QIOS_IOSB        | = 00000010 |
| CJF\$WRITEJNL    | 7FFEE5C0 G | QIOS_NARGS       | = 0000000C |
| CJF\$WRITEJNLW   | 7FFEE5C8 G | QIOS_P1          | = 0000001C |
| DEF MASK         | = 00000081 | QIOS_P2          | = 00000020 |
| ENQS_ACMODE      | = 00000028 | QIOS_P3          | = 00000024 |
| ENQS_ASTADR      | = 0000001C | QIOS_P4          | = 00000028 |
| ENQS_ASTPRM      | = 00000020 | QIOS_P5          | = 0000002C |
| ENQS_BLKAST      | = 00000024 | QIOS_P6          | = 00000030 |
| ENQS_EFN         | = 00000004 | RMSVECEND        | 7FFEE288   |
| ENQS_FLAGS       | = 00000010 | RUF\$CANCELRU    | 7FFEE648 G |
| ENQS_LKMODE      | = 00000008 | RUF\$CANRUH      | 7FFEE668 G |
| ENQS_LKSB        | = 0000000C | RUF\$DCLRUH      | 7FFEE660 G |
| ENQS_NARGS       | = 0000000B | RUF\$ENDRU       | 7FFEE678 G |
| ENQS_PARID       | = 00000018 | RUF\$KASCTR      | = 00004010 |
| ENQS_PROT        | = 0000002C | RUF\$MARKPOINTRU | 7FFEE650 G |
| ENQS_RESNAM      | = 00000014 | RUF\$PHASE1      | 7FFEE638 G |
| EXC MASK         | = 00000080 | RUF\$PHASE2      | 7FFEE640 G |
| GETDVIS_ASTADR   | = 00000018 | RUF\$REENTERRU   | 7FFEE628 G |
| GETDVIS_ASTPRM   | = 0000001C | RUF\$RESETRU     | 7FFEE658 G |
| GETDVIS_CHAN     | = 00000008 | RUF\$RUSTATUS    | 7FFEE670 G |
| GETDVIS_DEVNAM   | = 0000000C | RUF\$STARTRU     | 7FFEE630 G |
| GETDVIS_EFN      | = 00000004 | SNDJBCS_ASTADR   | = 00000018 |
| GETDVIS_IOSB     | = 00000014 | SNDJBCS_ASTPRM   | = 0000001C |
| GETDVIS_ITMLST   | = 00000010 | SNDJBCS_EFN      | = 00000004 |
| GETDVIS_NARGS    | = 00000008 | SNDJBCS_FUNC     | = 00000008 |



SYSSP1\_VECTOR  
Symbol table

M 4  
- P1 SYSTEM SERVICE VECTOR DEFINITIONS

16-SEP-1984 00:40:54 VAX/VMS Macro V04-00  
5-SEP-1984 03:40:37 [SYS.SRC]CMODSSDSP.MAR;1

Page 28  
(2)

|                  |            |   |
|------------------|------------|---|
| SNDJBC\$ IOSB    | = 00000014 |   |
| SNDJBC\$ ITMLST  | = 00000010 |   |
| SNDJBC\$ NARGS   | = 00000007 |   |
| SNDJBC\$ NULLARG | = 0000000C |   |
| SSVECREG2        | 7FFEE3C0   |   |
| SYNCH\$ EFN      | = 00000004 |   |
| SYNCH\$ IOSB     | = 00000008 |   |
| SYNCH\$ NARGS    | = 00000002 |   |
| SYSSADJSTK       | 7FFEDE20   | G |
| SYSSADJWSL       | 7FFEDE28   | G |
| SYSSALCDNP       | 7FFEDE30   | G |
| SYSSALLOC        | 7FFEDE38   | G |
| SYSSASCEFC       | 7FFEDE40   | G |
| SYSSASCTIM       | 7FFEDE48   | G |
| SYSSASCTOID      | 7FFEE4B0   | G |
| SYSSASSIGN       | 7FFEDE50   | G |
| SYSSBINTIM       | 7FFEDE58   | G |
| SYSSBRDCST       | 7FFEE098   | G |
| SYSSBRKTHRU      | 7FFEE4C8   | G |
| SYSSBRKTHRUW     | 7FFEE4E8   | G |
| SYSSCALL HANDL   | = 80000010 | G |
| SYSSCANCEL       | 7FFEDE60   | G |
| SYSSCANEXH       | 7FFEE0C0   | G |
| SYSSCANTIM       | 7FFEDE68   | G |
| SYSSCANWAK       | 7FFEDE70   | G |
| SYSSCHKPRO       | 7FFEE4E0   | G |
| SYSSCLI          | 7FFEDE18   | G |
| SYSSCLOSE        | 7FFEE1B8   | G |
| SYSSCLRAST       | 7FFEE108   | G |
| SYSSCLREF        | 7FFEDE98   | G |
| SYSSCLRPAR       | 7FFEDE80   | G |
| SYSSCMEXEC       | 7FFEDE88   | G |
| SYSSCMKRNL       | 7FFEDE90   | G |
| SYSSCNTREG       | 7FFEDEA0   | G |
| SYSSCONNECT      | 7FFEE1C0   | G |
| SYSSCREATE       | 7FFEE1C8   | G |
| SYSSCRELNM       | 7FFEE480   | G |
| SYSSCRELNT       | 7FFEE478   | G |
| SYSSCRELOG       | 7FFEDEB0   | G |
| SYSSCREMBX       | 7FFEDEB8   | G |
| SYSSCREPRC       | 7FFEDECO   | G |
| SYSSCRETVA       | 7FFEDEC8   | G |
| SYSSCRMPSC       | 7FFEDE78   | G |
| SYSSDACEFC       | 7FFEDED0   | G |
| SYSSDALLOC       | 7FFEDED8   | G |
| SYSSDASSGN       | 7FFEDEE0   | G |
| SYSSDCLAST       | 7FFEDEE8   | G |
| SYSSDCLCMH       | 7FFEE0A0   | G |
| SYSSDCLEXH       | 7FFEDEF0   | G |
| SYSSDELETE       | 7FFEE168   | G |
| SYSSDELLNM       | 7FFEE488   | G |
| SYSSDELLOG       | 7FFEDEF8   | G |
| SYSSDELMBX       | 7FFEDF00   | G |
| SYSSDELPRC       | 7FFEDF08   | G |
| SYSSDELTVA       | 7FFEDF10   | G |
| SYSSDEQ          | 7FFEE3C8   | G |
| SYSSDERLMB       | 7FFEE0B8   | G |

|                  |          |   |
|------------------|----------|---|
| SYSSDGBLSC       | 7FFEDF18 | G |
| SYSSDISCONNECT   | 7FFEE1D0 | G |
| SYSSDISPLAY      | 7FFEE1D8 | G |
| SYSSDLCDNP       | 7FFEDF20 | G |
| SYSSDLCEFC       | 7FFEDF28 | G |
| SYSS\$ENQ        | 7FFEE3C0 | G |
| SYSS\$ENQW       | 7FFEE3D0 | G |
| SYSS\$ENTER      | 7FFEE228 | G |
| SYSS\$ERAPAT     | 7FFEE470 | G |
| SYSS\$ERASE      | 7FFEE1E0 | G |
| SYSS\$EXCMG      | 7FFEE0E8 | G |
| SYSS\$EXIT       | 7FFEDF40 | G |
| SYSS\$EXPREG     | 7FFEDF48 | G |
| SYSS\$EXTEND     | 7FFEE1E8 | G |
| SYSS\$FAO        | 7FFEDF50 | G |
| SYSS\$FAOL       | 7FFEDF58 | G |
| SYSS\$FILESCAN   | 7FFEE278 | G |
| SYSS\$FIND       | 7FFEE170 | G |
| SYSS\$FINISH_RDB | 7FFEE4B8 | G |
| SYSS\$FLUSH      | 7FFEE1F0 | G |
| SYSS\$FORCEX     | 7FFEDF60 | G |
| SYSS\$FREE       | 7FFEE178 | G |
| SYSS\$GET        | 7FFEE180 | G |
| SYSS\$GETCHN     | 7FFEE0C8 | G |
| SYSS\$GETDEV     | 7FFEE0D0 | G |
| SYSS\$GETDVI     | 7FFEE410 | G |
| SYSS\$GETDVIW    | 7FFEE418 | G |
| SYSS\$GETJPI     | 7FFEE0D8 | G |
| SYSS\$GETJPIW    | 7FFEE420 | G |
| SYSS\$GETLKI     | 7FFEE498 | G |
| SYSS\$GETLKIW    | 7FFEE4A0 | G |
| SYSS\$GETMSG     | 7FFEE0B0 | G |
| SYSS\$GETPTI     | 7FFEDEA8 | G |
| SYSS\$GETQUI     | 7FFEE4F8 | G |
| SYSS\$GETQUIW    | 7FFEE500 | G |
| SYSS\$GETSYI     | 7FFEE3F8 | G |
| SYSS\$GETSYIW    | 7FFEE430 | G |
| SYSS\$GETTIM     | 7FFEDF78 | G |
| SYSS\$GL_ASTRET  | 7FFEE110 | G |
| SYSS\$GL_COMMON  | 7FFEE114 | G |
| SYSS\$GRANTID    | 7FFEE4D0 | G |
| SYSS\$HIBER      | 7FFEDF88 | G |
| SYSS\$IDTOASC    | 7FFEE4C0 | G |
| SYSS\$IMGACT     | 7FFEDF90 | G |
| SYSS\$IMGFIX     | 7FFEE400 | G |
| SYSS\$IMGFIX_2   | 7FFEE408 | G |
| SYSS\$IMGSTA     | 7FFEDF68 | G |
| SYSS\$LCKPAG     | 7FFEDF98 | G |
| SYSS\$LKWSET     | 7FFEDFA0 | G |
| SYSS\$MGBLSC     | 7FFEDFA8 | G |
| SYSS\$MODIFY     | 7FFEE1F8 | G |
| SYSS\$MTACCESS   | 7FFEE688 | G |
| SYSS\$NUMTIM     | 7FFEDFB8 | G |
| SYSS\$NXTVOL     | 7FFEE200 | G |
| SYSS\$OPEN       | 7FFEE208 | G |
| SYSS\$PARSE      | 7FFEE230 | G |
| SYSS\$PURGWS     | 7FFEDFB0 | G |

PAG  
V04



SYSSP1\_VECTOR  
Symbol table

N 4  
- P1 SYSTEM SERVICE VECTOR DEFINITIONS

16-SEP-1984 00:40:54 VAX/VMS Macro V04-00  
5-SEP-1984 03:40:37 [SYS.SRC]CMODSSDSP.MAR;1

Page 29  
(2)

|                |          |   |
|----------------|----------|---|
| SYSSPUT        | 7FFEE188 | G |
| SYSSPUTMSG     | 7FFEE0E0 | G |
| SYSSQIO        | 7FFEDFC8 | G |
| SYSSQIOW       | 7FFEDE00 | G |
| SYSSREAD       | 7FFEE190 | G |
| SYSSREADEF     | 7FFEDFD0 | G |
| SYSSRELEASE    | 7FFEE198 | G |
| SYSSREMOVE     | 7FFEE238 | G |
| SYSSRENAME     | 7FFEE240 | G |
| SYSSRESUME     | 7FFEDFD8 | G |
| SYSSREVOKID    | 7FFEE4D8 | G |
| SYSSREWIND     | 7FFEE210 | G |
| SYSSRMSRUHNDLR | 7FFEE270 | G |
| SYSSRMSRUNDWN  | 7FFEE268 | G |
| SYSSRUNDWN     | 7FFEDFE0 | G |
| SYSSSCHDWK     | 7FFEDFF0 | G |
| SYSSSEARCH     | 7FFEE248 | G |
| SYSSSETAST     | 7FFEDFF8 | G |
| SYSSSETDDIR    | 7FFEE250 | G |
| SYSSSETDFPROT  | 7FFEE258 | G |
| SYSSSETEF      | 7FFEE000 | G |
| SYSSSETEXV     | 7FFEE008 | G |
| SYSSSETIME     | 7FFEE0F8 | G |
| SYSSSETIMR     | 7FFEE020 | G |
| SYSSSETPFM     | 7FFEE0A8 | G |
| SYSSSETPRA     | 7FFEE018 | G |
| SYSSSETPRI     | 7FFEE028 | G |
| SYSSSETPRN     | 7FFEE010 | G |
| SYSSSETPRT     | 7FFEE030 | G |
| SYSSSETPRV     | 7FFEE100 | G |
| SYSSSETRWM     | 7FFEE038 | G |
| SYSSSETSFM     | 7FFEE040 | G |
| SYSSSETSSF     | 7FFEE3E8 | G |
| SYSSSETSTK     | 7FFEE3F0 | G |
| SYSSSETSUM     | 7FFEE048 | G |
| SYSSSNDACC     | 7FFEE0F0 | G |
| SYSSSNDERR     | 7FFEDF38 | G |
| SYSSSNDJBC     | 7FFEDF70 | G |
| SYSSSNDJBCW    | 7FFEE438 | G |
| SYSSSNDOPR     | 7FFEDFC0 | G |
| SYSSSND SMB    | 7FFEDFE8 | G |
| SYSSSPACE      | 7FFEE218 | G |
| SYSSSRCHANDLER | 7FFEE118 | G |
| SYSSSSVEXC     | 7FFEE260 | G |
| SYSSSUSPND     | 7FFEE050 | G |
| SYSSSYNCH      | 7FFEE440 | G |
| SYSSSTRNLNM    | 7FFEE490 | G |
| SYSSSTRNLOG    | 7FFEE058 | G |
| SYSSSTRUNCATE  | 7FFEE220 | G |
| SYSSSULKPAG    | 7FFEE060 | G |
| SYSSSULWSET    | 7FFEE068 | G |
| SYSSUNWIND     | 7FFEE070 | G |
| SYSSUPDATE     | 7FFEE1A0 | G |
| SYSSUPDSEC     | 7FFEDF30 | G |
| SYSSUPDSECW    | 7FFEDF80 | G |
| SYSSWAIT       | 7FFEE1A8 | G |
| SYSSWAITFR     | 7FFEE078 | G |

|                |            |   |
|----------------|------------|---|
| SYSSWAKE       | 7FFEE080   | G |
| SYSSWFLAND     | 7FFEE088   | G |
| SYSSWFLOR      | 7FFEE090   | G |
| SYSSWRITE      | 7FFEE1B0   | G |
| UPDSECS_ACMODE | = 0000000C |   |
| UPDSECS_ASTADR | = 0000001C |   |
| UPDSECS_ASTPRM | = 00000020 |   |
| UPDSECS_EFN    | = 00000014 |   |
| UPDSECS_INADR  | = 00000004 |   |
| UPDSECS_IOSB   | = 00000018 |   |
| UPDSECS_NARGS  | = 00000008 |   |
| UPDSECS_RETADR | = 00000008 |   |
| UPDSECS_UPDFLG | = 00000010 |   |
| VECBASE        | 7FFEDE00   |   |

PAG  
V04



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

| PSECT name | Allocation        | PSECT No. | Attributes                                              |
|------------|-------------------|-----------|---------------------------------------------------------|
| . ABS .    | 00000000 ( 0.)    | 00 ( 0.)  | NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE |
| \$AB\$\$   | 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:[SYS.LIB]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

**DIGITAL  
CONFIDE**

EQUIPMENT  
NTIAL AND

CORPORATION  
PROPRIETARY

This image shows a blank, aged, cream-colored page, likely an endpaper or flyleaf of a book. The paper has a slightly textured appearance with some faint smudges and discoloration, characteristic of old paper. The left edge of the page shows the binding of the book.