


```

PPPPPPPP      HH      HH  DDDDDDDD      UU      UU  TTTTTTTTTT  LL
PPPPPPPP      HH      HH  DDDDDDDD      UU      UU  TTTTTTTTTT  LL
PP      PP     HH      HH  DD      DD     UU      UU  TT      LL
PP      PP     HH      HH  DD      DD     UU      UU  TT      LL
PP      PP     HH      HH  DD      DD     UU      UU  TT      LL
PP      PP     HH      HH  DD      DD     UU      UU  TT      LL
PPPPPPPP      HHHHHHHHHH DD      DD     UU      UU  TT      LL
PPPPPPPP      HHHHHHHHHH DD      DD     UU      UU  TT      LL
PP      HH      HH  DD      DD     UU      UU  TT      LL
PP      HH      HH  DD      DD     UU      UU  TT      LL
PP      HH      HH  DD      DD     UU      UU  TT      LL
PP      HH      HH  DD      DD     UU      UU  TT      LL
PP      HH      HH  DDDDDDDD UUUUUUUUUU  TT      LLLLLLLLLL  ....
PP      HH      HH  DDDDDDDD UUUUUUUUUU  TT      LLLLLLLLLL  ....

```

```

LL      I I I I I      SSSSSSSS
LL      I I I I I      SSSSSSSS
LL      I I          SS
LL      I I          SS
LL      I I          SS
LL      I I          SS
LL      I I          SSSSSS
LL      I I          SSSSSS
LL      I I          SS
LL      I I          SS
LL      I I          SS
LL      I I          SS
LLLLLLLLLL  I I I I I  SSSSSSSS
LLLLLLLLLL  I I I I I  SSSSSSSS

```

(1)	35	HISTORY	: DETAILED
(2)	62	DECLARATIONS	
(3)	96	IMGRESET - IMAGE RESET, DELETE IMAGE PAGES	
(4)	301	CHK DBG SYMTAB - ANY DEBUG SYMBOL TABLE TO DELETE?	
(5)	338	DECSECF - DECREMENT SECTION TABLE REFERENCE COUNT	
(6)	407	DALCSTXSCN - DEALLOCATE SECTION INDEX SCANNER	
(7)	523	DALCSTX - DEALLOCATE SECTION TABLE INDEX	
(8)	578	ALCSTX - ALLOCATE SECTION TABLE INDEX	
(9)	651	ALCPHD - ALLOCATE PROCESS HEADER	
(10)	757	SECTBLRST - RESET SECTION TABLE	
(11)	835	EXPANDPHD - EXPAND THE PROCESS HEADER	

```

0000 1 .TITLE PHDUTL - PROCESS HEADER UTILITIES
0000 2 .IDENT 'V04-000'
0000 3
0000 4 *****
0000 5 *
0000 6 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 * ALL RIGHTS RESERVED.
0000 9 *
0000 10 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 * TRANSFERRED.
0000 16 *
0000 17 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 * CORPORATION.
0000 20 *
0000 21 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 *
0000 24 *
0000 25 *****
0000 26 *****
0000 27 :++
0000 28 : FACILITY: EXECUTIVE, MEMORY MANAGEMENT UTILITIY ROUTINES
0000 29
0000 30 : ABSTRACT:
0000 31
0000 32 : ENVIRONMENT: KERNEL MODE, RUNNING MOSTLY AT IPL ASTDEL WITH THE
0000 33 : PROCESS HEADER LOCKED
0000 34
0000 35 : .SBTTL HISTORY ; DETAILED
0000 36
0000 37 : AUTHOR: PETER H. LIPMAN , CREATION DATE: 23-SEP-76
0000 38
0000 39 : MODIFIED BY:
0000 40
0000 41 : V03-007 WMC0003 Wayne Cardoza 30-Jul-1984
0000 42 : Limit process header expansion to quota/2.
0000 43
0000 44 : V03-006 WMC0002 Wayne Cardoza 02-Dec-1983
0000 45 : PSTBASMAX is now longword.
0000 46
0000 47 : V03-005 WMC0001 Wayne Cardoza 07-Sep-1983
0000 48 : Fix the test for EXPANDPHD bugcheck.
0000 49
0000 50 : V03-004 LJK0210 Lawrence J. Kenah 21-Jun-1983
0000 51 : Add code to deallocate image control blocks that were allocated
0000 52 : as part of image activation. Eliminate call to IAC$CLRIACLOCK.
0000 53
0000 54 : V03-003 TCM0001 Trudy C. Matthews 1-Apr-1983
0000 55 : Change references to working set fields in PHD so that
0000 56 : they are used as unsigned words.
0000 57

```

PHDUTL
V04-000

- PROCESS HEADER UTILITIES
HISTORY ; DETAILED

K 3

16-SEP-1984 00:55:46 VAX/VMS Macro V04-00
5-SEP-1984 03:46:09 [SYS.SRC]PHDUTL.MAR;1

Page 2
(1)

0000 58 :
0000 59 :
0000 60 :--

V03-002 RNG0002

Rod N. Gamache

02-Feb-1983

Use longword displacement for call to MMG\$COMPRESS subroutine.

PHI
VOI

```
0000 62          .SBTTL  DECLARATIONS
0000 63
0000 64 :
0000 65 : INCLUDE FILES:
0000 66 :
0000 67          $ARBDEF          ;ACCESS RIGHTS BLOCK DEFINITIONS
0000 68          $CCBDEF          ;CHANNEL CONTROL BLOCK DEFINITIONS
0000 69          $IPLDEF          ;PROCESSOR PRIORITY LEVELS
0000 70          $OPDEF           ; DEFINE OPCODE EQUIVALENT VALUES
0000 71          $PCBDEF          ;PROCESS CONTROL BLOCK DEFINITIONS
0000 72          $PHDDEF          ;PROCESS HEADER DEFINITIONS
0000 73          $PRDEF           ;PROCESSOR REGISTER DEFINITIONS
0000 74          $PRTDEF          ;PROTECTION CODE DEFINITIONS
0000 75          $PSLDEF          ;PROCESSOR STATUS LONGWORD DEFINITIONS
0000 76          $PTEDEF          ;PAGE TABLE ENTRY DEFINITIONS
0000 77          $$SECDEF         ;SECTION TABLE ENTRY DEFINITIONS
0000 78          $$SSDEF         ;SYSTEM STATUS DEFINITIONS
0000 79          $VADEF           ;VIRTUAL ADDRESS YIELD DEFINITIONS
0000 80          $WCBDEF          ;WINDOW CONTROL BLOCK DEFINITIONS
0000 81          $WSLDEF          ;WORKING SET LIST DEFINITIONS
0000 82 :
0000 83 : EXTERNAL SYMBOLS:
0000 84 :
0000 85 :
0000 86 :
0000 87 : MACROS:
0000 88 :
0000 89 :
0000 90 : EQUATED SYMBOLS:
0000 91 :
0000 92 :
0000 93 : OWN STORAGE:
0000 94 :
```

```

0000 96          .SBTTL  IMGRESET - IMAGE RESET, DELETE IMAGE PAGES
0000 97
0000 98 :++
0000 99 : FUNCTIONAL DESCRIPTION:
0000 100 :
0000 101 :
0000 102 : CALLING SEQUENCE:
0000 103 :
0000 104 :         BSBW      MMG$IMGRESET
0000 105 :         MUST BE CALLED FROM KERNEL MODE AT IPL = 0
0000 106 :
0000 107 :
0000 108 : INPUT PARAMETERS:
0000 109 :
0000 110 :         R4 = PROCESS CONTROL BLOCK ADDRESS
0000 111 :
0000 112 : IMPLICIT INPUTS:
0000 113 :         NONE
0000 114 :
0000 115 : OUTPUT PARAMETERS:
0000 116 :
0000 117 :         DEALLOCATES ALL IMAGE CONTROL BLOCKS
0000 118 :         DELETES ALL PAGES IN P0 SPACE
0000 119 :         DELETES ALL PAGES IN P1 SPACE EXCEPT THE FIXED PORTION
0000 120 :         RESETS THE PROCESS SECTION TABLE
0000 121 :         RESETS THE WORKING SET SIZE TO THE DEFAULT
0000 122 :
0000 123 :         R0 = ERROR STATUS CODE
0000 124 :
0000 125 : IMPLICIT OUTPUTS:
0000 126 :         NONE
0000 127 :
0000 128 : COMPLETION CODES:
0000 129 :         NONE
0000 130 :
0000 131 : SIDE EFFECTS:
0000 132 :         NONE
0000 133 :
0000 134 : --
0000 135 :
0000 136 :
0000 137 : *****
0000 138 : *****
0000 139 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0000 140 :
0000 141 :         .PSECT  Y$EXEPAGED
0000 142 :
0000 143 : *****
0000 144 :
0000 145 MMG$IMGRESET:
0000 146         PUSHL   R5          ;PRESERVE R5
125 00000000'GF DD 0002 147         MOVL    G^CTL$GL PHD,R5 ;R5=PHD ADDRESS
00000000'EF  16 0009 148         JSB    MMG$EXTRADYNWS ;FORCE EXTRA DYNAMIC WS COMPUTATION
FFEE'       30 000F 149         BSBW   RMS$RESET ;RESET RMS CONTEXT
FFEB'       30 0012 150         BSBW   MMG$WSPEAKCHK ;ENABLE OR DISABLE WORKING SET PEAK CHECK
0015 151
0015 152 ; DEALLOCATE ALL IMAGE CONTROL BLOCKS THAT DESCRIBE CURRENTLY ACTIVATED IMAGES

```

```

52 00000000'GF 7E 0015 153
    00000000'GF 7E 0015 154
    50 00 B1 0F 001C 155
    04 B2 06 1D 0023 156
    60 0E 0027 157 5$:
    F4 11 0029 158
    002D 159
    002F 160
    002F 161
    002F 162 ; LOOK AT THE WORK LIST IN CASE ANYTHING GOT LEFT THERE
51 00000000'GF 7E 002F 163
    50 00 B1 0F 0036 164 10$:
    04 B2 06 1D 003A 165 15$:
    F4 11 003C 166
    0040 167
    0042 168
    0042 169
    0042 170
    0042 171 ; DELETE P0 SPACE
    0042 172
62 3FFFFFFF 7E 7C 0042 173 20$:
    52 5E D0 0044 174
    8F D0 0047 175
    28 A5 D5 004E 176
    10 13 0051 177
    6F 50 E9 0053 178
    0060 179
    0063 180
    0063 181
    0063 182 ; DELETE P1 SPACE
    0063 183
    0063 184 25$:
00000000'EF 00000200 62 D6 0063 185
    04 A2 8F C3 0065 186
    0070 187
    0072 188
    04 A2 30 A5 D1 0072 189
    37 18 0077 190
    49 50 E9 0079 191
    0000006A'EF 16 0086 192
    50 D5 0089 193
    1D 13 008F 194
    04 A2 7FFFFFFF 8F D0 0091 195
    02 A2 06 A2 B0 0093 196
    00A0 197
    22 50 E9 009B 198
    50 01 3C 00AD 199
    51 08 A5 A3 00B0 200 30$:
    51 51 B6 00B3 201
    7E 50 A5 3C 00B9 202
    51 8E C2 00B8 203
    0B 13 00B8 204
    51 76 A5 3C 00BE 205
    00C2 206
    00C5 207 35$:
    00C7 208 40$:
    00D2 209
    MOVAQ G^IAC$GL_ICBFL,R2 ;R2 = ADDRESS OF FREE LIST
    MOVAQ G^IAC$GL_IMAGE_LIST,R1 ;R1 = LISTHEAD OF ICBS
    REMQUE @ (R1),R0 ;REMOVE NEXT ENTRY
    BVS 10$ ;LIST EMPTY - ALL DONE
    INSQUE (R0),@4(R2) ;INSERT AT END OF FREE LIST
    BRB 5$ ;GO BACK FOR MORE
    MOVAQ G^IAC$GL_WORK_LIST,R1 ;R1 = LISTHEAD FOR WORK IN PROGRESS
    REMQUE @ (R1),R0 ;REMOVE NEXT ENTRY
    BVS 20$ ;LIST EMPTY - ALL DONE
    INSQUE (R0),@4(R2) ;INSERT AT END OF FREE LIST
    BRB 15$ ;GO BACK FOR MORE
    CLRQ -(SP) ;2 LONG WORDS FOR RANGE TO DELETE
    MOVL SP,R2 ;ADDRESSED BY R2
    MOVL #<<1@30>-1>,(R2) ;STARTVA = TOP OF P0 SPACE
    ;ENDVA = 0
    TSTL PHD$S_FREPOVA(R5) ;IF NOTHING TO DELETE
    BEQL 25$ ;DON'T CALL THE SERVICE
    $DELTVA_S INADR=(R2),ACMODE=#PSL$C KERNEL ;DELETE P0 SPACE
    BLBC -R0,40$ ;BRANCH IF ERROR
    INCL (R2) ;STARTVA = LOWEST P1 SPACE ADR
    SUBL3 #512,CTL$GL_CTLBASVA,4(R2)
    ;ENDVA = HIGHEST P1 SPACE ADR
    ;BELOW THE FIXED REGION
    Cmpl PHD$S_FREP1VA(R5),4(R2) ;IF P1 SPACE IS EMPTY
    BGEQ 30$ ;SKIP THE DELETE SERVICE
    $DELTVA_S INADR=(R2),ACMODE=#PSL$C KERNEL ;DELETE P1 SPACE
    BLBC -R0,40$ ;BRANCH IF ERROR
    JSB CHK_DBG_SYMTAB ;SEE IF ANY SYMBOL TABLE TO DELETE
    TSTL R0 ;IF ZERO, NONE TO DELETE
    BEQL 30$ ;BRANCH IF NONE
    MOVL #^X7FFFFFFF,4(R2) ;DELETE DEBUGGER SYMBOL TABLE
    MOVW 6(R2),2(R2) ;TOP 65K BYTES OF P1 SPACE
    $DELTVA_S INADR=(R2),ACMODE=#PSL$C KERNEL ;DELETE DEBUG SYMBOL TABLE
    BLBC -R0,40$ ;BRANCH IF ERROR
    MOVZWL #$$$ NORMAL,R0 ;SUCCESSFUL EXIT
    SUBW3 PHD$W_WSLIST(R5),PHD$W_DFWSCNT(R5),R1 ;SIZE OF DEFAULT WS-1
    INCW R1 ;REAL SIZE OF DEF WSL
    MOVZWL R1,R1 ;GET IT IN A LONGWORD
    MOVZWL PHD$W_WSSIZE(R5),-(SP) ;GET NUMBER OF PAGES
    SUBL (SP)+,R1 ;TO CHANGE WORKING SET SIZE BY
    BEQL 40$ ;BRANCH IF NO
    $ADJWSL_S R1 ;ADJUST THE WORKING SET LIMIT
    MOVZWL -PHD$W_EXTDYNWS(R5),R1 ;ENOUGH EXTRA DYNAMIC WORKING SET

```



```

51 02 51 C3 00D6 209 ;TO RUN A REASONABLE IMAGE?
;MAKE SURE WE CAN HAVE AT LEAST 2
;MORE PAGE TABLES THEN CURRENTLY HAVE
;BRANCH IF GOT ENOUGH
1E 15 00DA 211
00DA 212 BLEQ 50$
00DC 213
00DC 214 : R1 IS THE NUMBER OF ADDITIONAL WORKING SET ENTRIES NEEDED
00DC 215
52 08 A5 50 A5 A1 00DC 216 ADDW3 PHD$W_WSSIZE(R5),PHD$W_WSLIST(R5),R2 ;GET END OF WS POINTER
18 A5 52 B1 00E2 217 CMPW R2,PHD$W_WSQUOTA(R5) ;IF WE ARE AT QUOTA
DF 1B 00E6 218 BLEQU 35$ ;BRANCH IF QUOTA SUPPORTS NEEDED
;INCREASE IN WSSIZE
18 A5 51 A0 00E8 220 ADDW R1,PHD$W_WSQUOTA(R5) ;QUOTA IS ARTIFICIALLY LOW, RAISE IT
16 A5 18 A5 B1 00EC 221 CMPW PHD$W_WSQUOTA(R5),PHD$W_WSEXTENT(R5) ;EXTENT MUST ALWAYS >= QUOTA
D4 1B 00F1 222 BLEQU 35$ ;BRANCH IF OK
16 A5 18 A5 B0 00F3 223 MOVW PHD$W_WSQUOTA(R5),PHD$W_WSEXTENT(R5) ;BUMP EXTENT TO QUOTA
CD 11 00F8 224 BRB 35$ ;AND ADD MORE PAGES TO WORKING SET
00FA 225
0172 30 00FA 226 50$: SETIPL #IPL$ ASTDEL ;RAISE IPL AND
00FD 227 BSBW MMG$SECTBLRST ;RESET THE SECTION TABLE
0100 228 SETIPL #0 ;LOWER PRIORITY LEVEL
0103 229 ;EMPTY PST FREE LIST
65 00000000'GF 7D 0103 230 MOVQ G^CTL$GQ_PROCPRIV,(R5) ;SET DEFAULT PRIVILEGES
50 00BC C4 D0 010A 231 MOVL PCB$SL_ARB(R4),R0 ;GET ACCESS RIGHTS BLOCK ADDRESS
60 65 7D 010F 232 MOVQ (R5),ARB$Q_PRIV(R0) ;AND RESET COPY OF PRIVILEGE MASK
00000000'GF D4 0112 233 CLRL G^CTL$GL_IMGHDRBF ;NO CURRENTLY ACTIVE IMAGE
0118 234
0118 235 : SEE IF ANY GLOBAL SECTIONS TO BE DELETED AS A RESULT OF RELEASING THE
0118 236 : PROCESS' ADDRESS SPACE
0118 237
55 00000000'FF DE 0118 238 MOVAL @MMG$GL_SYSPHD,R5 ;SYSTEM PROCESS HEADER ADDRESS
12 36 A5 01 E1 011F 239 BBC #PHD$V_DALCSTX,PHD$W_FLAGS(R5),60$ ;BRANCH IF NO GBL SECS TO DELETE?
50 00000000'EF DE 0124 240 MOVAL EXE$GL_GSDMTX,R0 ;ADDRESS OF GLOBAL SECTION MUTEX
FED2' 30 012B 241 BSBW SCH$LOCKW ;LOCK THE MUTEX FOR WRITING
012E 242
012E 243 : RETURNS AT IPL = ASTDEL MITH MUTEX LOCKED FOR WRITING
012E 244
1D 10 012E 245 BSBW MMG$DALCSTX$N ;SCAN AND DELETE GLOBAL SECTIONS
FEC D' 30 0130 246 BSBW MMG$GSDMTXULK ;RELEASE THE GLOBAL SECTION MUTEX
;STAYS AT IPL = ASTDEL
FECA' 30 0133 248 BSBW MMG$DELGBLWCB ;DELETE ANY WINDOWS RELEASED
55 00000000'GF D0 0136 249 60$: MOVL G^CTL$GL_PHD,R5 ;R5=PHD ADDRESS
00000000'GF 16 013D 250 JSB G^MMG$COMPRESS ;SHRINK WSLAST BY WHATEVER WE CAN
0143 251 ;RETURNS AT IPL 0
2C BA 0143 252 POPR #*M<R2,R3,R5> ;CLEAN OFF 8 BYTES INTO R2, R3
0145 253 ;AND RESTORE R5
0145 254 RSB
0146 255
0146 256 : ***** THE FOLLOWING CODE MUST BE RESIDENT *****
0146 257 :
00000000 258 .PSECT $MMG$COD
0000 259
0000 260 : *****
0000 261
0000 262 MMG$COMPRESS:
0000 263 SETIPL #IPL$ SYNCH ;LOCK PAGING DATABASE
51 08 A5 50 A5 3C 0003 264 MOVZWL PHD$W_WSLAST(R5),R0 ;GET POINTER TO END OF ACTIVE WS LIST
A1 0007 265 ADDW3 PHD$W_WSSIZE(R5),PHD$W_WSLIST(R5),R1 ;SET MINIMUM SIZE OF WSL

```

```

51 50 B1 000D 266 CMPW R0,R1 ;ARE WE ALREADY AT MINIMUM?
      1C 1F 0010 267 BLSSU 100$ ;BRANCH IF SO
52 01 CE 0012 268 MNEGL #1,R2 ;SET UP FOR DECREMENT LOOP
      6540 D5 0015 269 70$: TSTL (R5)[R0] ;IS ENTRY EMPTY?
      06 12 0018 270 BNEQ 80$ ;BRANCH IF NOT
FFFS 50 52 51 F1 001A 271 ACBL R1,R2,R0,70$ ;LOOP THRU MAXIMUM POSSIBLE REDUCTION
      50 10 A5 B1 0020 272 80$: CMPW PHD$W_WSNEXT(R5),R0 ;IS WSNEXT STILL IN RANGE
      04 1B 0024 273 BLEQU 90$ ;BRANCH IF YES
      10 A5 50 B0 0026 274 MOVW R0,PHD$W_WSNEXT(R5) ;UPDATE TO SAFE PLACE
      12 A5 50 B0 002A 275 90$: MOVW R0,PHD$W_WSLAST(R5) ;RESET TO NEW MINIMUM END
      002E 276
      24 10 002E 277 100$: BSBB MMG$CALCSWAPSIZE ;FORM THE SIZE IN R2 OF MINIMUM SWAP AREA
50 20 A4 D0 0030 278 MOVL PCB$W_WSSWP(R4),R0 ;GET SWAP TYPE VBN OF ALLOCATION
      16 13 0034 279 BEQL 120$ ;BRANCH IF NON-SWAPPABLE PROCESS
51 52 A5 3C 0036 280 MOVZWL PHD$W_SWAPSIZE(R5),R1 ;SET SIZE OF CURRENT ALLOCATION
      52 51 D1 003A 281 CMPL R1,R2 ;IS CURRENT HOLDING CORRECT?
      0D 13 003D 282 BEQL 120$ ;BRANCH IF SO
      FFBE' 30 003F 283 BSBW MMG$ALLOCSWAPAREA ;ALLOCATE A SWAP AREA, R0-R3 CHANGED
      0C 15 0042 284 BLEQ 130$ ;BRANCH IF THIS FAILED, IT CAN'T
      20 A4 50 D0 0044 285 MOVL R0,PCB$W_WSSWP(R4) ;UPDATE SWAP FILE VBN
      52 A5 52 B0 0048 286 MOVW R2,PHD$W_SWAPSIZE(R5) ;AND SIZE OF AREA
      004C 287 120$: SETIPL #0 ;DROP BACK TO DESIRED IPL TO EXIT
      05 004F 288 RSB ;RETURN
      0050 289
      0050 290 130$: BUG_CHECK INSSWPFIL,FATAL ;BITMAP CONSISTENCY PROBLEM
      0054 291
      0054 292 MMG$CALCSWAPSIZE::
50 34 A4 36 A4 A1 0054 293 ADDW3 PCB$W_PPGCNT(R4),PCB$W_GPGCNT(R4),R0 ;GET CURRENT PAGE COUNT
      51 0000' CF 3C 005A 294 MOVZWL W*SWP$GW_SWPINC,R1 ;ROUND UP TO NEXT GROUP
      52 52 D4 005F 295 CLRL R2 ;INIT COUNT FOR WHAT WE NEED BACK
      52 51 C0 0061 296 110$: ADDL R1,R2 ;BUMP OUR ALLOCATION AMOUNT
      52 50 B1 0064 297 CMPW R0,R2 ;IS OUR PHYSICAL SIZE BIGGER THAN THIS?
      F8 1A 0067 298 BGTRU 110$ ;HANDLE CORRECT MULTIPLE OF 8 CASE
      05 0069 299 RSB

```

```

006A 301      .SBTTL  CHK_DBG_SYMTAB - ANY DEBUG SYMBOL TABLE TO DELETE?
006A 302      :
006A 303      : INPUTS:
006A 304      :
006A 305      :     R5 = PROCESS HEADER ADDRESS (P1 SPACE IS OK)
006A 306      :
006A 307      : OUTPUTS:
006A 308      :
006A 309      :     R0 = 0 IF DEBUG SYMBOL TABLE PAGE TABLE IS ALREADY EMPTY
006A 310      :     = NON ZERO IF DEBUG SYMBOL TABLE PAGE TABLE NEEDS TO BE DELETED
006A 311      :     R1,R2,R3 PRESERVED
006A 312      :
006A 313      : *****
006A 314      : ***** THE FOLLOWING CODE MUST BE RESIDENT *****
006A 315      : *****
0000006A 316      :
006A 317      :     .PSECT  $MMGCD
006A 318      :
006A 319      : *****
006A 320      :
006A 321      : CHK_DBG_SYMTAB:
006A 322      :     SETIPL #IPL$ SYNCH          ;NO SWAPPING WHILE LOOKING AT SPT
50 15 09 EF 006D 323      :     EXTZV  #VASV_VPN,#VASS_VPN,- ;SPT INDEX FOR FIRST
50 00C8 C5 0070 324      :     PHD$L-POBR(R5),R0          ;PAGE TABLE PAGE
50 0000 CF 50 0074 325      :     ADDL   W^SGN$GL_PTPAGCNT,R0 ;
0079 326      :     DECL   R0                  ;INDEX TO LAST PAGE TABLE
007B 327      :     BICL3 #^C<PTESM_VALID !-   ;GET VALID BIT
007C 328      :     PTESM_TYPT ! PTESM_TYPO !- ;PAGE TYPE BITS
007C 329      :     PTESM-PGFLVB>,-          ;AND BACKING STORE ADR OR PFN
50 0000 DF40 7B800000 8F 007C 330      :     @W^MMG$GL_SPTBASE[R0],R0  ;FROM SPT ENTRY OF LAST PAGE TABLE
0086 331      :
0086 332      :
0086 333      : WILL BE ZERO IF THE PAGE TABLE IS DEMAND ZERO
0086 334      :
0086 335      :     SETIPL #0                 ;RESET IPL
05 0089 336      :     RSB                      ;AND RETURN

```

```

008A 338 .SBTTL DECSECF - DECREMENT SECTION TABLE REFERENCE COUNT
008A 339 :++
008A 340 : FUNCTIONAL DESCRIPTION:
008A 341 :
008A 342 : THIS ROUTINE DECREASES THE REFERENCE COUNT ON THE SECTION TABLE
008A 343 : SPECIFIED. IF THE REFERENCE COUNT GOES TO ZERO AND THE SECTION IS NOT
008A 344 : PERMANENT, THEN THE SECTION DELETION LOGIC IS TRIGGERED.
008A 345 :
008A 346 : CALLING SEQUENCE:
008A 347 :
008A 348 : BSBW MMG$DECSECF
008A 349 : BSBW MMG$SUBSECF ;SUBTRACT R0 FROM SECTION REFERENCE COUNT
008A 350 :
008A 351 : INPUT PARAMETERS:
008A 352 :
008A 353 : R0 = NUMBER OF REFERENCES TO SUBTRACT IF ENTERED AT SUBSECF
008A 354 : R1 = SECTION TABLE INDEX
008A 355 : R5 = PROCESS HEADER ADDRESS
008A 356 :
008A 357 : IMPLICIT INPUTS:
008A 358 :
008A 359 : NONE
008A 360 :
008A 361 : OUTPUT PARAMETERS:
008A 362 :
008A 363 : R0,R3 PRESERVED
008A 364 :
008A 365 : IMPLICIT OUTPUTS:
008A 366 :
008A 367 : NONE
008A 368 :
008A 369 : COMPLETION CODES:
008A 370 :
008A 371 : NONE
008A 372 :
008A 373 : SIDE EFFECTS:
008A 374 :
008A 375 : NONE
008A 376 :
008A 377 : --
008A 378 :
008A 379 : *****
008A 380 : *****
008A 381 : ***** THE FOLLOWING CODE MUST BE RESIDENT *****
008A 382 :
0000008A 383 : .PSECT $MMGCD
008A 384 :
008A 385 : *****
008A 386 :
008A 387 : .ENABL LSB
008A 388 MMG$SUBSECF::
008A 389 ADDL3 PHD$L PSTBASOFF(R5),R5,R2 ;BASE ADDRESS OF SECTION TABLE
008A 390 SUBL R0,SECF$_REFCNT(R2)(R1) ;REDUCE THE REFERENCE COUNT
008A 391 BLEQ 10$ ;BRANCH IF NO MORE REFERENCES
008A 392 RSB
008A 393
008A 394 MMG$DECSECF::

```

52 55 20 A5
18 A241 50
OC

C1
C2
15
05

52	55	20 A5	C1	0097	395	ADDL3	PHD\$V_PSTBASOFF(R5),R5,R2 ;BASE ADDRESS OF SECTION TABLE
		18 A241	D7	009C	396	DECL	SEC\$V_REFcnt(R2)[R1] ;COUNT ONE LESS SECTION REFERENCE
		OF	14	00A0	397	BGTR	20\$;BRANCH IF SECTION IS STILL IN USE
		OE	19	00A2	398	BLSS	40\$;BRANCH IF COUNT WENT NEGATIVE
52	14	A241	11	9C	00A4	ROTL	#<31-SEC\$V_PERM>,SEC\$W_FLAGS(R2)[R1],R2 ;PERMANENT BIT TO SIGN BIT
				19	00AA	BLSS	20\$;BRANCH IF SECTION PERMANENT
	00	36 A5	01	E6	00AC	BBSS1	#PHD\$V_DALCSTX,PHD\$W_FLAGS(R5),20\$;NOTE SECTION TO BE DEALLOCATED
				05	00B1	RSB	
					00B2	BUG_CHECK	SECFNEG,FATAL ;SECTION REFERENCE COUNT WENT NEGATIVE
					00B6		
					00B6	.DSABL	LSB


```

53 55 20 A5 C1 0152 454 ADDL3 PHD$PSTBASOFF(R5),R5,R3 ;BASE OF SECTION TABLE
    51 24 A5 32 0157 465 CVTWL PHD$W_PSTLAST(R5),R1 ;INDEX TO LAST SECTION ALLOCATED
      18 A341 D5 0158 466 20$: TSTL SEC$L_REFCNT(R3)[R1] ;DEALLOCATE THIS SECTION?
        06 13 015F 467 BEQL 50$ ;BRANCH IF NEED TO DEALLOCATE
        51 08 C0 0161 468 30$: ADDL #SEC$C_LENGTH@-2,R1 ;NEXT SECTION INDEX
          F5 19 0164 469 BLSS 20$ ;BRANCH IF NOT AT END OF TABLE
            05 0166 470 40$: RSB
              0167 471 :
              0167 472 : IF THIS IS NOT A PERMANENT SECTION, THEN NEED TO DEALLOCATE IT
              0167 473 :
52 14 A341 11 9C 0167 474 50$: ROTL #<32-<SEC$V_PERM+1>>,SEC$W_FLAGS(R3)[R1],R2 ;PERMANENT BIT TO SIGN B
          F2 19 016D 475 BLSS 30$ ;BRANCH IF PERMANENT SECTION
          02 10 016F 476 BSBB 60$ ;DO THE DEALLOCATE
          EE 11 0171 477 BRB 30$ ;AND CONTINUE THE SCAN
            0173 478 :
            0173 479 : NEED TO DEALLOCATE PROCESS OR GLOBAL SECTION, RSB FROM HERE WHEN DONE
            0173 480 : MUST PRESERVE R1 AND R3
            0173 481 :
          40 52 11 E0 0173 482 60$: BBS #<SEC$V_GBL+<32-<SEC$V_PERM+1>>>,R2,200$ ;BRANCH IF GLOBAL SECTION
            51 DD 0177 483 PUSHL R1 ;SAVE SECTION INDEX
          52 6341 D0 0179 484 MOVL SEC$L_CCB(R3)[R1],R2 ;CHANNEL CONTROL BLOCK ADDRESS
50 04 A341 D0 017D 485 MOVL SEC$W_SECXFL(R3)[R1],R0 ;GET FORWARD AND BACKWARD SECTION INDEX LINK
      2F 13 0182 486 BEQL 130$ ;BRANCH IF UCB TYPE SECTION
        50 51 B1 0184 487 CMPW R1,R0 ;IF CURRENT = FORWARD, THIS IS THE LAST
          08 12 0187 488 BNEQ 110$ ;SECTION IN THE CHAIN
04 A2 0C A341 D0 0189 489 MOVL SEC$L_WINDOW(R3)[R1],CCB$L_WIND(R2) ;PUT WINDOW ADR BACK IN CHANNEL
      22 11 018F 490 BRB 130$ ;AND DEALLOCATE THE SECTION
        51 04 A2 B1 0191 491 110$: CMPW CCB$L_WIND(R2),R1 ;IF SECTION INDEX IN CCB
          04 12 0195 492 BNEQ 120$ ;POINTS TO SECTION BEING DELETED
52 04 A2 50 B0 0197 493 MOVW R0,CCB$L_WIND(R2) ;THEN POINT IT AT THE FORWARD LINK
      50 F0 8F 78 019B 494 120$: ASHL #-16,R0,R2 ;BACKWARD SECTION INDEX
        51 6342 DE 01A0 495 MOVAL (R3)[R2],R1 ;ADR OF BACKWARD SECTION TABLE ENTRY
          04 A1 50 B0 01A4 496 MOVW R0,SEC$W_SECXFL(R1) ;BAKSEC(FL) = CURSEC(FL)
            50 32 01A8 497 CVTWL R0,R0 ;FORWARD SECTION INDEX
            51 6340 DE 01AB 498 MOVAL (R3)[R0],R1 ;ADR OF FORWARD SECTION TABLE ENTRY
            06 A1 52 B0 01AF 499 MOVW R2,SEC$W_SECXBL(R1) ;FORSEC(BL) = CURSEC(BL)
              02 BA 01B3 500 130$: POPR #*M<R1> ;RECOVER SECTION INDEX TO BE DEALLOCATED
              19 11 01B5 501 BRB MMG$DALCSTX ;DEALLOCATE THE SECTION TABLE ENTRY
                01B7 502 :
                01B7 503 :
                01B7 504 : DEALLOCATE GLOBAL SECTION
                01B7 505 :
          0A BB 01B7 506 200$: PUSHR #*M<R1,R3> ;SAVE SECTION TABLE BASE, SECTION INDEX
            53 6341 DE 01B9 507 MOVAL (R3)[R1],R3 ;ADDRESS OF SECTION TABLE ENTRY
          09 14 A3 04 E0 01BD 508 BBS #SEC$V_SHMGS,SEC$W_FLAGS(R3),SHM_GS ;BR IF GS IS IN SHM
            00000000'EF 16 01C2 509 JSB MMG$DE[GBLSEC] ;DELETE THE GLOBAL SECTION
              0A BA 01C8 510 210$: POPR #*M<R1,R3> ;RESTORE SECTION TABLE BASE AND INDEX
                05 01CA 511 RSB
                  01CB 512 :
                  01CB 513 : THIS SECTION IS IN SHARED MEMORY. CHECK IF IT IS READY TO BE DELETED.
                  01CB 514 : NOTE THAT ALL SHARED MEMORY GLOBAL SECTIONS MUST BE PERMANENT. SINCE THIS
                  01CB 515 : ONE IS NOT PERMANENT, A $DGBLSC REQUEST MUST HAVE BEEN ISSUED FOR IT.
                  01CB 516 :
                  01CB 517 :
                FE32' 30 01CB 518 SHM_GS: BSBW MMG$DELSHMGS ;CHECK IF SHM GS CAN BE DELETED
                  F8 11 01CE 519 BRB 210$ ;CONTINUE SCAN
                    01D0 520

```

PH
Ps

PS
--
:SA
YS
\$M
ZS

Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As

Th
85
Th
98
29

Ma
--
-S
-S
TC

16
Th
MA

PHDUTL
V04-000

I 4
- PROCESS HEADER UTILITIES
DALCSTXSCN - DEALLOCATE SECTION INDEX SC 16-SEP-1984 00:55:46 VAX/VMS Macro V04-00
5-SEP-1984 03:46:09 [SYS.SRC]PHDUTL.MAR;1
01D0 521 .DSABL LSB


```

01D0 523 .SBTTL DALCSTX - DEALLOCATE SECTION TABLE INDEX
01D0 524 :++
01D0 525 : FUNCTIONAL DESCRIPTION:
01D0 526 :
01D0 527 : THIS ROUTINE PLACES THE GIVEN SECTION TABLE ENTRY ON THE THE
01D0 528 : SECTION TABLE FREE LIST, TURNING ON THE SIGN BIT OF THE REFCNT LONG WORD
01D0 529 : AS AN INDICATION THAT IT IS NOT IN USE.
01D0 530 :
01D0 531 : CALLING SEQUENCE:
01D0 532 :
01D0 533 : BSBW MMGSDALCSTX
01D0 534 :
01D0 535 : INPUT PARAMETERS:
01D0 536 :
01D0 537 : R1 = SECTION TABLE INDEX
01D0 538 : R5 = PROCESS HEADER ADDRESS
01D0 539 :
01D0 540 : IMPLICIT INPUTS:
01D0 541 :
01D0 542 : NONE
01D0 543 :
01D0 544 : OUTPUT PARAMETERS:
01D0 545 :
01D0 546 : R0,R1,R2,R3 PRESERVED
01D0 547 :
01D0 548 : IMPLICIT OUTPUTS:
01D0 549 :
01D0 550 : NONE
01D0 551 :
01D0 552 : COMPLETION CODES:
01D0 553 :
01D0 554 : NONE
01D0 555 :
01D0 556 : SIDE EFFECTS:
01D0 557 :
01D0 558 : NONE
01D0 559 :
01D0 560 : --
01D0 561 :
01D0 562 : *****
01D0 563 :
01D0 564 : ***** THE FOLLOWING CODE MAY BE PAGED *****
01D0 565 :
000001D0 566 : .PSECT YSEXEPAGED
01D0 567 :
01D0 568 : *****
01D0 569 :
01D0 570 MMGSDALCSTX::
7E 55 20 A5 C1 01D0 571 ADDL3 PHD$ PSTBASOFF(R5),R5,-(SP) ;BASE ADDRESS OF SECTION TABLE
7E 7E 6E 18 C1 01D5 572 ADDL3 #SEC$ REFCNT,(SP),-(SP) ;OFFSET TO REFERENCE COUNT
9E41 9E41 01 CE 01D9 573 MNEGL #1,@(SP)+[R1] ;NEG REF COUNT MEANS NOT IN USE
9E41 26 A5 32 01DD 574 CVTWL PHD$W PSTFREE(R5),@(SP)+[R1] ;PUT FREE POINTER INTO THIS SECTION ENT
26 A5 51 B0 01E2 575 MOVW R1,PHD$W_PSTFREE(R5) ;AND THIS SECTION INDEX IN FREE POINTER
05 01E6 576 RSB

```

```

01E7 578 .SBTTL ALCSTX - ALOCATE SECTION TABLE INDEX
01E7 579 :++
01E7 580 : FUNCTIONAL DESCRIPTION:
01E7 581 :
01E7 582 : THIS ROUTINE ALLOCATES A SECTION TABLE INDEX FROM THE
01E7 583 : SECTION TABLE BELONGING TO THE SPECIFIED PROCESS HEADER
01E7 584 :
01E7 585 : CALLING SEQUENCE:
01E7 586 :
01E7 587 : BSBW MMG$ALCSTX
01E7 588 :
01E7 589 : INPUT PARAMETERS:
01E7 590 :
01E7 591 : R4 = PROCESS CONTROL BLOCK ADDRESS
01E7 592 : R5 = PROCESS HEADER ADDRESS
01E7 593 : IPL = ASTDEL
01E7 594 : ASSUMPTION: ONLY GST IS ALLOCATED FROM THE SYSTEM HEADER
01E7 595 : AND THE GLOBAL SECTION MUTEX IS LOCKED FOR WRITING WHEN
01E7 596 : THIS IS DONE. OTHERWISE, THE SYSTEM HEADER WOULD REQUIRE
01E7 597 : IPL = SYNCH
01E7 598 :
01E7 599 : IMPLICIT INPUTS:
01E7 600 :
01E7 601 : NONE
01E7 602 :
01E7 603 : OUTPUT PARAMETERS:
01E7 604 :
01E7 605 : R0 = SYSTEM STATUS CODE
01E7 606 : R1 = SECTION TABLE INDEX IF SUCCESSFUL
01E7 607 : R2,R3 PRESERVED
01E7 608 :
01E7 609 : IMPLICIT OUTPUTS:
01E7 610 :
01E7 611 : NONE
01E7 612 :
01E7 613 : COMPLETION CODES:
01E7 614 :
01E7 615 : $$$_SECTBLFUL ;SECTION TABLE FULL
01E7 616 :
01E7 617 : SIDE EFFECTS:
01E7 618 :
01E7 619 : NONE
01E7 620 :
01E7 621 : --
01E7 622 :
01E7 623 :
01E7 624 : *****
01E7 625 : *****
01E7 626 : ***** THE FOLLOWING CODE MAY BE PAGED *****
01E7 627 :
0000 01E7 628 : .PSECT Y$EXEPAGED
01E7 629 :
01E7 630 : *****
01E7 631 :
01E7 632 : MMG$ALCSTX:
51 26 A5 32 01E7 633 : CVTWL PHD$W_PSTFREE(R5),R1 ;SEE IF ANY SECTIONS ON FREE LIST
OE 13 01E8 634 : BEQL 20$ ;BRANCH IF NONE

```

50	55	20	A5	C1	01ED	635	ADDL3	PHD\$ _L PSTBASOFF(R5),R5,R0	;BASE ADDRESS OF SECTION TABLE
	26	A5	6041	F7	01F2	636	CVTLW	(R0)[R1],PHD\$ _W PSTFREE(R5)	;REMOVE THIS SECTION FROM FREE LIST
		50	01	3C	01F7	637	MOVZWL	#SS\$ _{-NORMAL} ,R0	;SUCCESSFUL COMPLETION
				05	01FA	638	RSB		;RETURN TO CALLER
					01FB	639			
					01FB	640			
					01FB	641			
					01FB	642			
	51	08		9A	01FB	642	20\$: MOVZBL	#SEC\$C_LENGTH@-2,R1	;SPACE REQUIRED IN LONG WORDS
		0C		BB	01FE	643	PUSHR	#*M<R2,R3>	;PRESERVE THESE REGISTERS
		0E		10	0200	644	BSBB	MMG\$ALCPHD	;TRY TO ALLOCATE PROCESS HEADER SPACE
		0C		BA	0202	645	POPR	#*M<R2,R3>	;RESTORE SAVED REGISTERS
		08	50	E9	0204	646	BLBC	R0,40\$;BRANCH IF NOT ENOUGH SPACE AVAILABLE
24	A5	51		A2	0207	647	SUBW	R1,PHD\$ _W PSTLAST(R5)	;UPDATE LAST ALLOCATED PST INDEX
51	24	A5		32	020B	648	CVTLW	PHD\$ _W PSTLAST(R5),R1	;SECTION INDEX
				05	020F	649	RSB		

```

0210 651 .SBTTL ALCPHD - ALLOCATE PROCESS HEADER
0210 652 :++
0210 653 : FUNCTIONAL DESCRIPTION:
0210 654 :
0210 655 : THIS ROUTINE IS CALLED TO ALLOCATE SOME SPACE IN THE PROCESS HEADER
0210 656 : FOR EITHER A NEW SECTION TABLE ENTRY OR SOME NUMBER OF WORKING
0210 657 : SET LIST ENTRIES. THE WORKING SET LIST ENTRIES GROW TOWARD LARGER
0210 658 : ADDRESSES FROM PHD$W WSLAST. THE SECTION TABLE GROWS FROM THE
0210 659 : HIGHEST ADDRESS TOWARD THE END OF THE WORKING SET. WHEN THEY
0210 660 : MEET, IT IS POSSIBLE THAT THERE IS STILL MORE SPACE AVAILABLE IF
0210 661 : THE PROCESS SECTION TABLE BASE ADDRESS IS NOT EQUAL TO THE PO PAGE
0210 662 : TABLE ADDRESS. IN ORDER TO ALLOW THE SECTION TABLE TO BE EASILY
0210 663 : RELOCATED TO A HIGHER ADDRESS, ALL SECTION INDICES ARE NEGATIVE
0210 664 : LONG WORD INDICES FROM A SECTION BASE ADDRESS. THIS ADDRESS IS
0210 665 : COMPUTED AS THE SUM OF THE PROCESS HEADER ADDRESS AND THE CONTENTS
0210 666 : OF PSTBASOFF.
0210 667 :
0210 668 : CALLING SEQUENCE:
0210 669 :
0210 670 : BSBW MMG$ALCPHD
0210 671 :
0210 672 : INPUT PARAMETERS:
0210 673 :
0210 674 : R1 = NUMBER OF LONG WORDS REQUIRED
0210 675 : R4 = PROCESS CONTROL BLOCK ADDRESS
0210 676 : R5 = PROCESS HEADER ADDRESS (P1 SPACE UNLESS SYSTEM PCB)
0210 677 : IPL = ASTDEL
0210 678 :
0210 679 : IMPLICIT INPUTS:
0210 680 :
0210 681 : NONE
0210 682 :
0210 683 : OUTPUT PARAMETERS:
0210 684 :
0210 685 : R0 = SYSTEM STATUS CODE
0210 686 : R1 = PRESERVED IF SUCCESSFUL
0210 687 : = NUMBER OF LONG WORDS AVAILABLE IF FAILED
0210 688 :
0210 689 : IMPLICIT OUTPUTS:
0210 690 :
0210 691 : NONE
0210 692 :
0210 693 : COMPLETION CODES:
0210 694 :
0210 695 : $$$_NORMAL ;SUCCESSFUL COMPLETION
0210 696 : $$$_SECTBLFUL ;SECTION TABLE FULL
0210 697 :
0210 698 : SIDE EFFECTS:
0210 699 :
0210 700 : NONE
0210 701 :
0210 702 :--

```

```

0210 704 :
0210 705 : *****
0210 706 :
0210 707 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0210 708 :
0000 0210 709 .PSECT YSEXEPAGED
0210 710 :
0210 711 : *****
0210 712 :
0210 713 MMG$ALCPHD::
0210 714 :
0210 715 : ESTABLISH HOW MUCH ROOM IS AVAILABLE
0210 716 :
52 50 24 A5 32 0210 717 10$: CVTWL PHD$W PSTLAST(R5),R0 ;SECTION INDEX TO LOWEST ADDRESSED SECTION
20 A5 55 C1 0214 718 ADDL3 R5,PHD$L PSTBASOFF(R5),R2 ;P1 ADR OF END OF PROC SEC TAB
50 6240 DE 0219 719 MOVAL (R2)[R0],R0 ;LOWEST ADDRESS IN SECTION TABLES
52 12 A5 3C 021D 720 MOVZWL PHD$W WSLAST(R5),R2 ;WSLX FOR LAST (INCLUSIVE) ENTRY
52 04 A542 DE 0221 721 MOVAL 4(R5)[R2],R2 ;ADR OF FIRST AVAILABLE BYTE BEYOND WSL
52 50 52 C3 0226 722 SUBL3 R2,R0,R2 ;NO. OF BYTES AVAILABLE
52 42 19 022A 723 BLSS ALCPHDBUG ;BRANCH IF OVERLAP, ERROR
52 04 C6 022C 724 DIVL #4,R2 ;CONVERT TO LONG WORDS
52 51 D1 022F 725 CMPL R1,R2 ;ENOUGH SPACE AVAILABLE
50 04 14 0232 726 BGTR 20$ ;BRANCH IF NOT, TRY TO GET MORE
50 01 3C 0234 727 MOVZWL #SS$_NORMAL,R0 ;INDICATE SUCCESSFUL COMPLETION
05 0237 728 RSB ;AND RETURN
0238 729 :
0238 730 : NEED TO EXPAND PROCESS HEADER IF POSSIBLE
0238 731 : R0 = P1 SPACE ADDRESS OF LOWEST ADDRESSED SECTION TABLE ENTRY
0238 732 : R1 = DESIRED LONG WORD COUNT
0238 733 : R2 = NO. OF LONG WORDS AVAILABLE WITHOUT EXPANSION
0238 734 :
50 24 07 BB 0238 735 20$: PUSHR #*M<R0,R1,R2> ;SAVE AVAIL, DESIRED, & LOWEST P1 ADRS
24 A5 32 023A 736 CVTWL PHD$W PSTLAST(R5),R0 ;GET OFFSET TO LAST SEC TBL ENTRY
32 10 023E 737 BSBB MMG$SECTBLRST ;TRY TO COMPRESS SECTION TABLE
50 24 A5 B1 0240 738 CMPW PHD$W PSTLAST(R5),R0 ;LAST SEC TBL ENTRY SAME AS BEFORE?
04 13 0244 739 BEQL 25$ ;BR IF LAST ENTRY IS SAME AS BEFORE
07 BA 0246 740 POPR #*M<R0,R1,R2> ;RESTORE AVAIL, DESIRED, $ LOWEST P1
C6 11 0248 741 BRB 10$ ;TRY TO ALLOCATE SECTION NOW
50 8ED0 024A 742 25$: POPL R0 ;RESTORE P1 ADR OF LOWEST SEC TBL ENTRY
20 A5 1E 9C 024D 743 ROTL #<32-2>,PHD$L PSTBASOFF(R5),R1 ;LONG WORD OFFSET TO BASE OF PST
48 A5 51 B1 0252 744 CMPW R1,PHD$L PSTBASMAX(R5) ;AT MAX PST BASE?
OF 1E 0256 745 BGEQU 40$ ;BRANCH IF YES, CANNOT EXPAND
000000B6'EF 16 0258 746 JSB MMG$EXPANDPHD ;EXPAND PROCESS HEADER IF POSSIBLE
06 BA 025E 747 30$: POPR #*M<R1,R2> ;R1=DESIRED SPACE, R2=AVAILABLE SPACE
AD 50 E8 0260 748 BLBS R0,10$ ;IF EXPANDED OK, TRY TO ALLOCATE AGAIN
51 52 D0 0263 749 MOVL R2,R1 ;OTHERWISE RETURN WHAT IS AVAILABLE
50 021C 8F 05 0266 750 RSB ;ALONG WITH ERRCR CONDITION IN R0
FO 11 0267 751 40$: MOVZWL #SS$_SECTBLFUL,R0 ;SECTION TABLE FULL
026C 752 BRB 30$
026E 753
026E 754 ALCPHDBUG:
026E 755 BUG_CHECK ALCPHD,FATAL

```

```

0272 757 .SBTTL SECTBLRST - RESET SECTION TABLE
0272 758 :++
0272 759 : FUNCTIONAL DESCRIPTION:
0272 760 :
0272 761 : THIS ROUTINE IS CALLED TO COMPRESS THE SECTION TABLE PORTION OF THE
0272 762 : PROCESS HEADER. IT WILL RESET THE POINTER TO THE LAST SECTION TABLE
0272 763 : USED IF SOME ENTRIES PREVIOUSLY IN USE ARE NO LONGER ALLOCATED. THIS
0272 764 : ROUTINE IS CALLED AT IMAGE RESET TIME AND ALSO WHEN TRYING TO
0272 765 : EXPAND THE WORKING SET OR THE PROCESS SECTION TABLE.
0272 766 :
0272 767 : CALLING SEQUENCE:
0272 768 :
0272 769 : BSBW MMG$SECTBLRST
0272 770 :
0272 771 : INPUT PARAMETERS:
0272 772 :
0272 773 : R5 - PROCESS HEADER ADDRESS
0272 774 :
0272 775 : IMPLICIT INPUTS:
0272 776 :
0272 777 : IPL = ASTDEL
0272 778 :
0272 779 : OUTPUT PARAMETERS:
0272 780 :
0272 781 : R0,R3 PRESERVED
0272 782 :
0272 783 : IMPLICIT OUTPUTS:
0272 784 :
0272 785 : NONE
0272 786 :
0272 787 : COMPLETION CODES:
0272 788 :
0272 789 : NONE
0272 790 :
0272 791 : SIDE EFFECTS:
0272 792 :
0272 793 : NONE
0272 794 :
0272 795 : --
0272 796 :
0272 797 : *****
0272 798 :
0272 799 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0272 800 :
0000272 801 : .PSECT Y$EXEPAGED
0272 802 :
0272 803 : *****
0272 804 :
0272 805 MMG$SECTBLRST:
0272 806 :
52 55 26 A5 B4 0272 807 CLRW PHD$W_PSTFREE(R5) ;ASSUME ALL SECTIONS AVAILABLE
51 20 A5 C1 0275 808 ADDL3 PHD$L_PSTBASOFF(R5),R5,R2 ;BASE ADDRESS OF SECTION TABLE
51 24 A5 32 027A 809 CVTWL PHD$W_PSTLAST(R5),R1 ;GET OFFSET TO LAST SEC TBL ENTRY USED
09 11 027E 810 BRB 30$ ;ENTER SEARCH LOOP
18 A241 D5 0280 811 ;
09 18 0284 812 20$: TSTL SECL_REFcnt(R2)[R1] ;ARE THERE PAGES STILL MAPPED TO SEC?
813 BGEQ 40$ ;BR ON YES, PAGE MAPPED OR BACKING STORE

```

```

51 08 C0 0286 814 ADDL2 #SEC$C_LENGTH@-2,R1 ;GET OFFSET TO NEXT SECTION TABLE ENTRY
    F5 19 0289 815 30$: BLSS 20$ ;KEEP SEARCHING FOR SECTION IN USE
24 A5 B4 028B 816 CLRW PHD$W_PSTLAST(R5) ;NO SECTION STILL IN USE
    05 028E 817 RSB
      028F 818
24 A5 51 B0 028F 819 40$: MOVW R1,PHD$W_PSTLAST(R5) ;FOUND FIRST SECTION STILL IN USE
    OF 11 0293 820 BRB 80$ ;CONTINUE SEARCHING FOR MORE SECTIONS
      0295 821
      0295 822
      0295 823 : THE REMAINING SECTIONS ARE EITHER IN USE (REFCNT GEQ 0) OR DELETED
      0295 824 : (REFCNT LSS 0). THE DELETED ONE MUST BE CHAINED ONTO THE FREE SECTION LIST.
      0295 825 :
18 A241 D5 0295 826 60$: TSTL SEC$L_REFCNT(R2)[R1] ;IS THE SECTION STILL IN USE?
    09 18 0299 827 BGEQ 80$ ;BR IF ALLOCATED FOR USE
6241 26 A5 32 029B 828 CVTWL PHD$W_PSTFREE(R5),SEC$L_CCB(R2)[R1] ;CHAIN THIS ENTRY
26 A5 51 B0 02A0 829 MOVW R1,PHD$W_PSTFREE(R5) ;ONTO THE TREE LIST
      02A4 830
51 08 C0 02A4 831 80$: ADDL #SEC$C_LENGTH@-2,R1 ;GET OFFSET TO NEXT SECTION TABLE ENTRY
    EC 19 02A7 832 BLSS 60$ ;BR IF ANOTHER ENTRY
      05 02A9 833 RSB

```

```
02AA 835 .SBTTL EXPANDPHD - EXPAND THE PROCESS HEADER
02AA 836 :++
02AA 837 : FUNCTIONAL DESCRIPTION:
02AA 838 :
02AA 839 : THIS ROUTINE ATTEMPTS TO ADD A NEW PAGE OF ZEROES AT THE CURRENT
02AA 840 : END OF THE SECTION TABLE. IF THERE WAS SOME ROOM BEFORE THE PO PAGE
02AA 841 : TABLE, THEN THIS WILL SUCCEED AND THE SECTION TABLE IS MOVED INTO
02AA 842 : THE NEW AREA.
02AA 843 :
02AA 844 : CALLING SEQUENCE:
02AA 845 :
02AA 846 : BSBW MMG$EXPANDPHD
02AA 847 :
02AA 848 : INPUT PARAMETERS:
02AA 849 :
02AA 850 : R0 = ADDRESS OF LOWEST ADDRESSED SECTION TABLE ENTRY
02AA 851 : P1 SPACE ADDRESS UNLESS SYSTEM PROCESS HEADER
02AA 852 : R4 = PROCESS CONTROL BLOCK ADDRESS
02AA 853 : R5 = PROCESS HEADER ADDRESS (P1 SPACE UNLESS SYSTEM HEADER)
02AA 854 :
02AA 855 : IMPLICIT INPUTS:
02AA 856 :
02AA 857 : NONE
02AA 858 :
02AA 859 : OUTPUT PARAMETERS:
02AA 860 :
02AA 861 : R0 = SYSTEM STATUS CODE
02AA 862 :
02AA 863 : IMPLICIT OUTPUTS:
02AA 864 :
02AA 865 : NONE
02AA 866 :
02AA 867 : COMPLETION CODES:
02AA 868 :
02AA 869 : SSS_SECTBLFUL ;SECTION TABLE FULL
02AA 870 :
02AA 871 : SIDE EFFECTS:
02AA 872 :
02AA 873 : NONE
02AA 874 :
02AA 875 :--
```



```

02AA 877 :
02AA 878 : *****
02AA 879 :
02AA 880 : ***** THE FOLLOWING CODE MUST BE RESIDENT *****
02AA 881 :
000000B6 882 : .PSECT $MMGCOD
00B6 883 :
00B6 884 : *****
00B6 885 :
MMG$EXPANDPHD:
00B6 886
00B6 887 SETIPL #IPL$ SYNCH ;PROTECT PROCESS HEADER DATA BASE
56 0071 8F BB 00B9 888 PUSHR #*M<R0,R4,R5,R6> ;
52 55 20 A5 DO 00BD 889 MOVL PHD$L_PSTBASOFF(R5),R6 ;BYTE OFFSET TO END OF SECTION TABLE
52 55 56 C1 00C1 890 ADDL3 R6,R5,R2 ;P1 SPACE ADDRESS FOR THE NEW PAGE
FF38' 30 00C5 891 BSBW MMG$PTEADRCHK ;GET ITS PTE ADDRESS
63 D5 00C8 892 TSTL (R3) ;P1 PTE SHOULD BE ZERO
OD 13 00CA 893 BEQL 5$ ;BRANCH IF OK
00CC 894 3$: SETIPL #IPL$ ASTDEL ;BACK TO AST LEVEL
50 0071 8F BA 00CF 895 POPR #*M<R0,R4,R5,R6> ;FIX STACK
021C 8F 3C 00D3 896 MOVZWL #SS$_SECTBLFUL,R0 ;SECTION TABLE FULL
05 00D8 897 RSB
00D9 898
00D9 899 :
00D9 900 : The process header will be limited to <working set quota>/2 pages.
00D9 901 : This is necessary to prevent it from occupying the whole working set
00D9 902 : when the extent:quota ratio is extremely large.
00D9 903 :
52 18 A5 3C 00D9 904 5$: MOVZWL PHD$W_WSQUOTA(R5),R2
52 08 A5 A2 00DD 905 SUBW PHD$W_WSLIST(R5),R2 ;QUOTA
52 52 FF 8F 78 00E1 906 ASHL #-1,R2,R2 ;DIVIDE IT BY 2
53 56 F7 8F 78 00E6 907 ASHL #-9,R6,R3 ;PAGES OF PROCESS HEADER - ARRAYS
53 00000000'EF A0 00EB 908 ADDW SWP$GW_BAKPTE,R3 ;ADD IN THE ARRAYS
52 52 53 B1 00F2 909 CMPW R3,R2
6C A4 56 C1 00F5 910 BGTRU 3$ ;TOO MANY PAGES
FF01' 30 00FC 911 ADDL3 R6,PCBSL_PHD(R4),R2 ;SYSTEM ADR OF NEXT
63 63 D5 00FF 912 ;AVAILABLE PAGE OF PROCESS HEADER
66 12 0101 913 BSBW MMG$SVAPTECHK ;GET SYS VA OF PTE FOR SYSTEM PTE
63 06 18 9C 0103 914 TSTL (R3) ;MUST BE 0 OR DATA BASE ERROR
0107 915 BNEQ 30$ ;BRANCH IF ERROR
010A 916 ROTL #PTESV_PROT,#PRTSC_ERKW,(R3) ;MAKE INTO DEMAND ZERO PAGE
010A 917 10$: SETIPL #IPL$_ASTDEL ;BACK TO PAGE FAULTABLE STATE
010A 918 :
010A 919 : This code must run at ASTDEL to both allow the pagefault and prevent a growth
010A 920 : AST. This code is allowed to loop, since the swapper could break in here and
010A 921 : remove this page from the working set as the result of a shrink, although
010A 922 : unlikely- and is never expected to do it continuously causing a code loop.
010A 923 :
6C B446 95 010A 924 TSTB @PCBSL_PHD(R4)[R6] ;FAULT THE PROCESS HEADER PAGE
010E 925 ;THROUGH ITS SYSTEM ADDRESS
52 6C B446 9E 0111 926 SETIPL #IPL$ SYNCH ;PROTECT PROCESS HEADER DATA BASE AGAIN
FEE7' 30 0116 927 MOVAB @PCBSL_PHD(R4)[R6],R2 ;ADDRESS OF NEW PROCESS HEADER PAGE
50 63 DO 0119 928 BSBW MMG$SVAPTECHK ;GET SYS VIRTUAL ADR OF SPTE
E9 14 011C 930 MOVL (R3),R0 ;GET THE SPT ENTRY
73 13 011E 931 BGTR 10$ ;LOOP IF ENTRY EXISTS, BUT IS NOT VALID
50 DD 0120 932 BEQL EXPANDPHDBUG ;BRANCH IF NOT VALID, ERROR
50 50 15 00 EF 0122 933 PUSHL R0 ;SAVE THE SPTE
EXTZV #PTESV_PFN,#PTES_PFN,R0,R0 ;GET THE PFN

```

```

0127 934
0127 935 MOVZWL <@W*PFNSAx WSLX[R0],R1>,- ;GET WORKING SET LIST INDEX
0127 936 LONG_OPCODE=MOVL,-
0127 937 IMAGE=SYS NONPAGED
6541 20 C8 012D 938 BISL #WSLSM_WSCLOCK,(R5)[R1] ;SET THE WORKING SET LOCK
0131 939 ;PROCESS HDR PAGE IS NOT A CANDIDATE
0131 940 ;FOR WORKING SET LIST REPLACEMENT
0131 941 PUSHR #^M<R2,R3> ;SAVE SVAPTE AND VA FROM SWAPWSLE
52 OE AS BB 0133 942 MOVZWL PHDSW_WSDYN(R5),R2 ;WSLX TO SWAP TO
FEC6 30 0137 943 BSBW MMGSSQAPWSLE ;SWAP INTO LOCKED PORTION
OE AS B6 013A 944 INCW PHDSW_WSDYN(R5) ;DYNAMIC PORTION STARTS ONE HIGHER
OE AS 10 AS B1 013D 945 CMPW PHDSW_WSNEXT(R5),PHDSW_WSDYN(R5) ;MAKE SURE WSNEXT IS OK
10 AS 12 AS 1E 0142 946 BGEQU 20$ ;BRANCH IF IT IS
0144 947 MOVW PHDSW_WSLAST(R5),PHDSW_WSNEXT(R5) ;MUST RESET IT
0149 948 20$: POPR #^M<R2,R3> ;RECOVER VA, SVAPTE
FEB? 30 014B 949 BSBW MMGSEXTRADYNWS ;RECOMPUTE EXTRA DYNAMIC WSLE COUNT
52 55 56 C1 014E 950 ADDL3 R6,R5,R2 ;P1 SPACE ADDRESS FOR THE NEW PAGE
FEAB 30 0152 951 BSBW MMGSPTEADRCHK ;GET ITS PTE ADDRESS
51 8E D0 0155 952 MOVL (SP)+,R1 ;LOAD SVAPTE
63 D5 0158 953 TSTL (R3) ;P1 PTE TOO SHOULD HAVE BEEN ZERO
7E 51 15 00 EF 015C 954 BEQL 40$ ;BRANCH IF OK
7E 63 15 00 EF 0161 955 EXTZV #PTESV_PFN,#PTESS_PFN,R1,-(SP) ;GET THE PFN
8E 8E D1 0166 956 EXTZV #PTESV_PFN,#PTESS_PFN,(R3),-(SP) ;GET THE PFN
0169 957 CML (SP)+,(R3)+ ;STILL OK IF SAME, SWAP AT IPL 2
63 28 12 0169 959 30$: BNEQ EXPANDPHDBUG ;COULD CAUSE THIS TO BE FILLED
63 51 D0 016B 960 40$: MOVL R1,(R3) ;BRANCH IF IT WASN'T, ERROR
016E 961 ;FILL IN P1 SPACE PTE
016E 962 ;
016E 963 ; NOW MOVE THE SECTION TABLE PAGE(S) TO OPEN UP SOME SPACE BETWEEN
016E 964 ; THE WORKING SET LIST AND THE SECTION TABLE.
016E 965 ; MUST NOT LOWER IPL AT THIS POINT BECAUSE MODIFIED PAGE WRITER MIGHT
016E 966 ; TRY TO ACCESS THE SECTION TABLE.
016E 967 ; R2 = P1 VIRTUAL ADDRESS OF 1ST BYTE OF NEW PAGE
016E 968 ; 0(SP) = ADDRESS OF LOWEST ADDRESSED SECTION TABLE ENTRY
20 AS 0200 C6 DE 016E 969 MOVAL ^X200(R6),PHDSL_PSTBASOFF(R5) ;PST BASE IS 1 PAGE HIGHER
51 6E D0 0174 970 MOVL (SP),R1 ;ADDRESS OF 1ST SECTION TABLE ENTRY
52 51 C2 0177 971 SUBL R1,R2 ;NO. OF BYTES OF SECTION TABLE TO MOVE
9E 0200 C1 61 52 28 017A 972 MOVCS R2,(R1),^X200(R1) ;MOVE SECTION TABLE 1 PAGE
8F 00 61 00 2C 0180 973 MOVCS #0,(R1),#0,#^X200,@(SP)+ ;ZERO 1 PAGE FROM 1ST SEC TBL ENTH;
50 01 3C 0188 974 MOVZWL #SS$ NORMAL,R0 ;SET SUCCESS CODE
018B 975 SETIPL #IPL$ ASTDEL ;BACK TO AST LEVEL
0070 8F BA 018E 976 POPR #^M<R4,R5,R6> ;RESTORE REGISTERS
05 0192 977 RSB ;AND RETURN
0193 978 ;
0193 979 ; DATA BASE INCONSISTENCY
0193 980 ;
0193 981 EXPANDPHDBUG:
0193 982 BUG_CHECK EXPANDPHD,FATAL
0197 983
0197 984 .END

```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
YSEXEPAGED	000002AA (682.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$MMGCOD	00000197 (407.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
Z\$INIT\$PFN_FIXUP_TABLE	00000006 (6.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.03	00:00:02.30
Command processing	106	00:00:00.58	00:00:05.59
Pass 1	403	00:00:14.52	00:00:43.21
Symbol table sort	0	00:00:02.29	00:00:07.72
Pass 2	178	00:00:03.45	00:00:13.96
Symbol table output	14	00:00:00.09	00:00:00.47
Psect synopsis output	1	00:00:00.04	00:00:00.23
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	733	00:00:21.01	00:01:13.54

The working set limit was 1800 pages.
85759 bytes (168 pages) of virtual memory were used to buffer the intermediate code.
There were 80 pages of symbol table space allocated to hold 1477 non-local and 48 local symbols.
984 source lines were read in Pass 1, producing 21 object records in Pass 2.
29 pages of virtual memory were used to define 27 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	12
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	12
TOTALS (all libraries)	24

1613 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:PHDUTL/OBJ=OBJ\$:PHDUTL MSRCS:PHDUTL/UPDATE=(ENH\$:PHDUTL)+EXECMLS/LIB

0379 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

A grid of 16 columns and 16 rows of technical data sheets. Each sheet contains various diagrams, tables, and text. The sheets are arranged in a regular grid pattern. Some sheets have larger, bolded titles:

- RELOCDRU LIS (top right)
- RMSRESET LIS (second row, right side)
- PROCSTR LIS (middle row, center)
- PHOUTL LIS (middle row, left side)
- PMSDAT LIS (bottom row, left side)
- POSTEF LIS (bottom row, middle-left)
- POWERFAL LIS (bottom row, middle-right)
- PRDEF LIS (bottom row, center)
- PTEDUMP LIS (bottom row, right side)
- PDAT LIS (bottom left)
- RMSVECTOR LIS (bottom right)