

A large, stylized letter 'A' is formed using the characters 'S' and 'Y'. The left and right vertical strokes are composed of 'S' characters, while the central vertical stroke and the two diagonal strokes are composed of 'Y' characters. The 'A' is symmetrical and has a bold, blocky appearance.

• • • •

• • • •

• • • •

• • • •

SSSSSSSS
SSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSS
SSSSSSSS

IOLOCK
Table of contents

- LOCK AND UNLOCK PAGES FOR I/O H 16

16-SEP-1984 00:19:38 VAX/VMS Macro V04-00

Page 0

(1) 38
(1) 96
(4) 304
(6) 520

HISTORY ; DETAILED
DECLARATIONS
IOLOCKPAG - LOCK INDIVIDUAL PAGE FOR I/O
UNLOCK PAGES AT COMPLETION OF I/O


```
0000 1      .TITLE IOLOCK - LOCK AND UNLOCK PAGES FOR I/O
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 SUBROUTINES
0000 29
0000 30     ABSTRACT: THIS MODULE PROVIDES THE ROUTINE FOR QI/O TO LOCK DOWN
0000 31     THE PAGES BEING TRANSFERRED AND THE ROUTINE REQUIRED BY THE I/O
0000 32     COMPLETION LOGIC TO UNLOCK THOSE PAGES.
0000 33
0000 34     ENVIRONMENT: KERNEL MODE
0000 35
0000 36     --
0000 37
0000 38     .SBTTL HISTORY ; DETAILED
0000 39
0000 40     AUTHOR: PETER H. LIPMAN , CREATION DATE: 17-SEP-76
0000 41
0000 42     MODIFIED BY:
0000 43
0000 44     V03-003 TCM0001 Trudy C. Matthews 31-Mar-1983
0000 45     Change references to working set fields in PHD so that
0000 46     they are used as unsigned words.
0000 47
0000 48     V03-002 WMC0001 Wayne Cardoza 17-May-1982
0000 49     Fix error recovery in the DEMANDZERO path to readjust the
0000 50     section reference count.
0000 51     When making a working set entry for a transition page, revalidate
0000 52     the PTE.
0000 53
0000 54     V03-001 HRJ0060 Herb Jacobs 20-Mar-1982
0000 55     Add support to allow swapper to lock shell pages into system
0000 56     working set, and tell whether any need to be read.
0000 57
```


0000	58	:	V0212	PHL0010	Peter H. Lipman	15-May-1981	
0000	59	:		Do not try to adjust working set if System Space transfer.			
0000	60	:		Fix bug in zeroing system space page.			
0000	61	:		Shut off DZRO section bit if zeroing section page.			
0000	62	:					
0000	63	:	V0211	HRJ0020	Herb Jacobs	27-Apr-1981	
0000	64	:		Change references to PHDSW_WSSIZE to get working set size.			
0000	65	:					
0000	66	:	V0210	RIH0060	RICHARD I. HUSTVEDT	3-MAR-1980	
0000	67	:		ATTEMPT TO INCREASE WORKING SET BEFORE GIVING SSS_INSFWSL ERROR			
0000	68	:		RETURN.			
0000	69	:					
0000	70	:	V0209	KDM0023	KATHLEEN D. MORSE	10-APR-1979	09:20
0000	71	:		COMBINE LOGIC AT 'FAULTREQUIRED' WITH 'MMGSUNLOCK1'.			
0000	72	:					
0000	73	:	V0208	KDM0022	KATHLEEN D. MORSE	04-APR-1979	15:15
0000	74	:		FIX BUG IN COMPUTING NUMBER OF PAGE TABLES TO LOCK.			
0000	75	:					
0000	76	:					
0000	77	:	V0207	SRB0001	STEVE BECKHARDT	30-MAR-1979	15:00
0000	78	:		CHANGED INTERFACE TO MMGSIOLOCK AND MMGSIOLOCKPAG			
0000	79	:		TO ALLOW FOR I/O DEVICES THAT READ AND WRITE MEMORY. THE			
0000	80	:		CHANGE ALLOWS THE CALLER TO INHIBIT AN OPTIMIZATION IN			
0000	81	:		MMGSIOLOCKPAG. NOTE THAT THIS CHANGE DOES NOT CHANGE ANY			
0000	82	:		CODE, ONLY THE INTERFACES. THE CODE WORKS AS IS.			
0000	83	:					
0000	84	:	V0206	KDM0001	KATHLEEN D. MORSE	14-NOV-1978	09:15
0000	85	:		REMOVE RESTRICTION OF 128 PAGES FROM MMGSIOLOCK AND			
0000	86	:		MMGSUNLOCK.			
0000	87	:					
0000	88	:	V0210	KDM0037	KATHLEEN D. MORSE	12-JUN-1979	14:10
0000	89	:		DECREMENT SECTION REFERENCE COUNT FOR CRF PAGES THAT ARE FAULTED			
0000	90	:		IN FOR AN I/O TRANSACTION.			
0000	91	:					
0000	92	:					
0000	93	:					
0000	94	:					

: VERSION

01


```
0000 96      .SBTTL  DECLARATIONS
0000 97
0000 98 :
0000 99 : INCLUDE FILES:
0000 100 :
0000 101      $IPLDEF      ;PRIORITY LEVEL DEFINITIONS
0000 102      $PCBDEF      ;PROCESS CONTROL BLOCK DEFINITIONS
0000 103      $PFNDEF      ;PFN DATA BASE DEFINITIONS
0000 104      $PHDDEF      ;PROCESS HEADER DEFINITIONS
0000 105      $PRDEF       ;PROCESSOR REGISTER DEFINITIONS
0000 106      $PTEDEF      ;PAGE TABLE ENTRY DEFINITIONS
0000 107      $SSDEF       ;SYSTEM STATUS DEFINITIONS
0000 108      $VADEF       ;VIRTUAL ADDRESS VIELDS
0000 109      $WSLDEF      ;WORKING SET LIST DEFINITIONS
0000 110 :
0000 111 : EXTERNAL SYMBOLS:
0000 112 :
0000 113 :
0000 114 :
0000 115 : MACROS:
0000 116 :
0000 117 :
0000 118 :
0000 119 : EQUATED SYMBOLS:
0000 120 :
0000 121 :
0000 122 :
0000 123 : OWN STORAGE:
0000 124 :
0000 125 :
0000 126 : *****
0000 127 : *****
0000 128 : ***** THIS ENTIRE MODULE MUST BE RESIDENT *****
0000 129 :
00000000 130      .PSECT  $MMGCODE
0000 131 :
0000 132 : *****
0000 133 :
```



```
0000 135 :++
0000 136 : FUNCTIONAL DESCRIPTION:
0000 137 :
0000 138 :     IOLOCK ACCEPTS A VIRTUAL ADDRESS AND BYTE COUNT OF AN I/O BUFFER
0000 139 :     AND A DIRECTION OF TRANSFER INDICATOR.  THE SPECIFIED PAGES
0000 140 :     ARE MADE RESIDENT AND MARKED MODIFIED IF APPROPRIATE, AND THEN
0000 141 :     THEY ARE LOCKED INTO MEMORY BY INCREMENTING THEIR REFERENCE COUNTS.
0000 142 :
0000 143 : CALLING SEQUENCE:
0000 144 :
0000 145 :     BSBW    MMG$IOLOCK
0000 146 :
0000 147 :
0000 148 : INPUT PARAMETERS:
0000 149 :
0000 150 :     R0 = VIRTUAL ADDRESS OF BEGINNING OF BUFFER
0000 151 :     R1 = SIZE OF BUFFER IN BYTES
0000 152 :     R2 = TRANSFER DIRECTION INDICATOR
0000 153 :         0 = WRITE OUT OF MEMORY TO I/O DEVICE
0000 154 :         1 = READ INTO MEMORY FROM I/O DEVICE
0000 155 :         5 = I/O DEVICE READS AND WRITES MEMORY
0000 156 :     R4 = CURRENT PROCESS PCB ADDRESS
0000 157 :
0000 158 :     IPL MUST BE AT ASTDEL
0000 159 :
0000 160 : IMPLICIT INPUTS:
0000 161 :
0000 162 :     NONE
0000 163 :
0000 164 : OUTPUT PARAMETERS:
0000 165 :
0000 166 :     IF SUCCESSFUL
0000 167 :         R0 = #SS$ NORMAL (LOW BIT SET)
0000 168 :         R1 = SYSTEM VIRTUAL ADDRESS OF FIRST PAGE TABLE ENTRY
0000 169 :
0000 170 :     IF NOT SUCCESSFUL
0000 171 :         R0 = 0
0000 172 :         R1 = VIRTUAL ADDRESS TO FAULT AND TRY AGAIN
0000 173 :     OR
0000 174 :         R0 = SYSTEM STATUS CODE (LOW BIT CLEAR)
0000 175 :
0000 176 : IMPLICIT OUTPUTS:
0000 177 :
0000 178 :     NONE
0000 179 :
0000 180 : COMPLETION CODES:
0000 181 :
0000 182 :     SS$_INSFWSL                                ;INSUFICIENT WORKING SET LIMIT
0000 183 :
0000 184 : SIDE EFFECTS:
0000 185 :
0000 186 :     NONE
0000 187 :
0000 188 :--
```



```
0000 190 MMG$IOLOCK::
0000 191 .ENABL LSB
0000 192 PUSHHR #M<R4,R5,R6,R7,R8>
56 50 09 00 EF 0007 193 MOVL R2,R7 ;DIRECTION INDICATOR
57 03 12 000C 194 EXTZV #VASV_BYTE,#VASS_BYTE,R0 ;R6 ;STARTING BYTE IN PAGE
57 02 C8 000E 195 BNEQ 10$ ;BRANCH IF PARTIAL PAGE
0011 196 BISL #2,R7 ;FIRST PAGE IS FULL PAGE
55 00000000'9F D0 0011 197 10$: MOVL @#CTL$GL_PHD,R5 ;UNLESS IT IS ALSO THE LAST
53 50 56 C3 0018 198 SUBL3 R6,R0,R3 ;P1 SPACE ADDRESS OF PROCESS HEADER
52 7041 9E 001C 199 MOVAB -(R0)[R1],R2 ;R3 = PAGE ALIGNED VA
1A 18 0020 200 BGEQ 20$ ;R2=LAST BYTE INCLUSIVE
50 0C 3C 0022 201 MOVZWL #SS$ ACCVIO,R0 ;BRANCH IF PROCESS VA
61 53 1F E1 0025 202 BBC #31,R3,IOLOCKACCVIO ;STATUS CODE FOR ACCESS VIOLATION
0000'CF 52 D1 0029 203 CMLP R2,W*SWP$GL_BALBASE ;ENTIRE RANGE MUST BE SYSTEM SPACE
5A 1E 002E 204 BGEQU IOLOCKACCVIO ;AND BELOW BALANCE SET SLOTS
54 0000'CF DE 0030 205 MOVAL W*MMG$AL_SYSPCB,R4 ;BRANCH IF NOT
55 6C A4 D0 0035 206 MOVL PCB$P_PHD(R4),R5 ;USE SYSTEM PCB
53 02 90 0039 207 MOVB #WSL$C_SYSTEM,R3 ;AND SYSTEM PHD
51 56 C0 003C 208 ADDL R6,R1 ;SET SYSTEM PAGE TYPE
56 51 F7 8F 78 003F 209 20$: ASHL #-9,R1,R6 ;BYTE COUNT + OFFSET
0044 210 BITW #VASM_BYTE,R1 ;PAGE COUNT NOT INCLUDING LAST
51 01FF 8F B3 0044 211 BEQL 30$ ;PARTIAL PAGE IF ANY
06 13 0049 212 INCL R6 ;IS LAST PAGE A PARTIAL PAGE?
56 D6 004B 213 BBSS #31,R7,30$ ;BRANCH IF NOT
00 57 1F E2 004D 214 30$: MOVZWL PHD$W_WSSIZE(R5),-(SP) ;COUNT ONE MORE PAGE
7E 50 A5 3C 0051 215 CLRL R0 ;SET 'LAST PARTIAL PAGE' INDICATOR
50 0E A5 08 A5 A3 0057 216 SUBW3 PHD$W_WSLIST(R5),PHD$W_WSDYN(R5),R0 ;RETRY AFTER AUGMENTING WS SIZE
50 8E 50 C3 005D 217 SUBL3 R0,(SP)+,R0 ;GET CURRENT SIZE
50 56 B1 0061 218 CMPW R6,R0 ;CLEAR UPPER HALF OF R0
28 1F 0064 219 BLSSU 32$ ;GET FIXED PORTION OF WS
0066 220 ;FIND DYNAMIC PORTION OF WS
0066 221 ;MAKE SURE THERE IS ENOUGH FOR REQUEST
0066 222 ;BRANCH IF ENOUGH
0066 223 : NUMBER OF PAGES REQUIRED EXCEEDS THE DYNAMIC SIZE OF THE WORKING SET
0066 224 :
0066 225 :
1B 53 1F E0 0066 226 BBS #31,R3,31$ ;NO WSL ADJUST IF SYSTEM SPACE BUFFER
51 DD 006A 227 PUSHL R1 ;SAVE VOLATILE REGISTER
7E 50 A5 3C 006C 228 MOVZWL PHD$W_WSSIZE(R5),-(SP) ;SAVE CURRENT WS SIZE
00 DD 0070 229 PUSHL #0 ;NO ADDRESS FOR PREVIOUS VALUE
7E 7F 8F 9A 0072 230 MOVZBL #127, -(SP) ;AUGMENT WS BY 127 PAGES
00000000'EF 02 FB 0076 231 CALLS #2,EXESADJWSL ;CALL INTERNAL ADJUST ROUTINE
03 BA 007D 232 POPR #M<R0,R1> ;RESTORE SAVED WSSIZE, R1
50 50 A5 B1 007F 233 CMPW PHD$W_WSSIZE(R5),R0 ;ANY CHANGE?
CC 12 0083 234 BNEQ 30$ ;IF YES, THEN TRY AGAIN
50 011C 8F 3C 0085 235 31$: MOVZWL #SS$_INSFWSL,R0 ;INSUFFICIENT WORKING SET LIMIT
008A 236 :
008A 237 : ACCESS VIOLATION - ATTEMPTED I/O REQUEST TO BALANCE SET SLOTS
008A 238 :
008A 239 IOLOCKACCVIO:
50 DD 008A 240 PUSHL R0 ;SO THAT THE POPR WILL BE RIGHT
53 11 008C 241 BRB IOLOCKEXIT
008E 242
7E 53 F0 8F 78 008E 243 32$: ASHL #-16,R3, -(SP) ;GET STARTING VA SHIFTED
50 52 F0 8F 78 0093 244 ASHL #-16,R2,R0 ;GET ENDING VA SHIFTED (ELIM BYTE OFF)
50 8E C2 0098 245 SUBL (SP)+,R0 ;GET # OF PAGE TABLE PAGES BETWEEN VA'S
50 D6 009B 246 35$: INCL R0 ;R0=COUNT OF PAGE TABLE PAGES TO LOCK
```



```

      09 BB 009D 247
      6E 01 D1 009F 248 38$: PUSHR #^M<R0,R3> ;SAVE START VA AND PUSH LOOP COUNTER
      04 12 00A2 249 CMPL S^#1,(SP) ;IS THIS THE FIRST PAGE TBL PAG TO LOCK?
      52 04 AE D0 00A4 250 BNEQ 40$ ;BR IF NOT FIRST PAGE
      FF52' 30 00AB 251 40$: MOVL 4(SP),R2 ;USE STARTING VA
      00AE 252 BSWW MMG$LOCK?GTB ;BACK TO ASTDEL IN ORDER TO FAULT PT
      00AE 253 ;REFERENCE AND LOCK PAGE TABLE PAGE
      00AE 254 SUBL2 #^X10000,R2 ;RETURNS SVAPTE IN R3, IPL=SYNCH
      E7 6E F5 00B5 255 SOBGTR (SP),38$ ;GET ADR IN NEXT PAGE TABLE PAGE
      05 BA 00B8 256 POPR #^M<R0,R2> ;LOCK ALL PAGE TABLE PAGES
      55 6C A4 D0 00BA 257 MOVL PCBSL_PHI(R4),R5 ;CLEAN STACK, R2=STARTING VIRTUAL ADR
      53 DD 00BE 258 PUSHL R3 ;NO LONGER SWAPPABLE, USE SYSTEM ADR OF
      03 11 00C0 259 BRB 55$ ;PHD AND SAVE IT FOR RETURN TO CALLER
      00C2 260 ;START AFTER THE 'INC' SVAPTE
      00C2 261 ; 0(SP) = STARTING SYS VIRTUAL ADDRESS OF PAGE TABLE ENTRY
      00C2 262 ; R6 = PAGE COUNT TO LOOP THROUGH
      00C2 263 ; R7<0> - SET IF READING INTO MEMORY
      00C2 264 ; R7<1> - SET IF TRANSFERRING THE ENTIRE PAGE
      00C2 265 ; R7<2> - SET IF I/O DEVICE IS READING AND WRITING MEMORY
      00C2 266 ; R7<31> - SET IF LAST PAGE IS PARTIAL PAGE
      00C2 267
      08 53 04 C0 00C2 268 50$: ADDL #4,R3 ;NEXT PAGE TABLE ENTRY ADDRESS
      57 1F E1 00C5 269 55$: BBC #31,R7,60$ ;BRANCH IF LAST PAGE IS A FULL PAGE
      01 56 D1 00C9 270 CMPL R6,#1 ;IS THIS THE LAST PAGE?
      03 14 00CC 271 BGTR 60$ ;BRANCH IF NOT
      57 02 CA 00CE 272 BICL #2,R7 ;INDICATE PARTIAL PAGE
      13 32 10 00D1 273 60$: BSBB MMG$IOLOCKPAG ;LOCK THIS PAGE FOR I/O
      57 50 E9 00D3 274 BLBC R0,FAULTREQUIRED ;BRANCH IF MUST PAGE FAULT
      52 57 02 C8 00D6 275 BISL #2,R7 ;AFTER FIRST PAGE, FULL PAGES
      0200 C2 DE 00D9 276 MOVAL ^X200(R2),R2 ;NEXT VIRTUAL PAGE
      E1 56 F5 00DE 277 SOBGTR R6,50$ ;ONCE FOR EACH PAGE
      00E1 278 .DSABL LSB
      00E1 279
      00E1 280 ; R0 = STATUS CODE
      00E1 281
      00E1 282 IOLOCKEXIT:
      01F2 8F BA 00E1 283 POPR #^M<R1,R4,R5,R6,R7,R8> ;RESTORE REGISTERS AND RETURN R1
      00E5 284 ;WITH SVAPTE IF SUCCESSFUL
      00E5 285 ;OR VA TO FAULT IF NOT
      00E5 286 SETIPL #IPL$ASTDEL ;RETURN WITH IPL AS CALLED
      05 00E8 287 RSB
      00E9 288
      00E9 289 ; FAILED TO LOCK THE PAGE FOR I/O, IT MUST BE PAGE FAULTED FIRST
      00E9 290 ; R3 = SVAPTE OF FIRST PAGE NOT LOCKED
      00E9 291
      00E9 292 FAULTREQUIRED:
      51 53 D0 00E9 293 MOVL R3,R1 ;R1=SVAPTE OF PAGE NOT LOCKED
      53 8E D0 00EC 294 MOVL (SP)+,R3 ;R3=SVAPTE OF FIRST PAGE IN TRANSFER
      50 FC A146 DE 00EF 295 MOVAL -4(R1)[R6],R0 ;R0=LAST SVAPTE FOR PG TBL PG INCLUSIVE
      51 53 C2 00F4 296 SUBL2 R3,R1 ;# OF BYTES OF PTE LOCKED
      51 51 FE 8F 78 00F7 297 ASHL #-2,R1,R1 ;R1=# PAGES IN TRANSFER LOCKED
      52 DD 00FC 298 PUSHL R2 ;SAVE VIRTUAL ADDRESS OF BE FAULTED
      012D 30 00FE 299 BSWW MMG$UNLOCK1 ;UNLOCK EVERYTHING THAT WAS LOCKED
      50 D4 0101 300 CLRL R0 ;RETURN STATUS INDICATING
      DC 11 0103 301 BRB IOLOCKEXIT ;"PAGE FAULT AND TRY AGAIN"
      0105 302
```



```
0105 304 .SBTTL IOLOCKPAG - LOCK INDIVIDUAL PAGE FOR I/O
0105 305 :++
0105 306 : FUNCTIONAL DESCRIPTION:
0105 307 :
0105 308 : THIS ROUTINE LOCKS THE SPECIFIED PAGE FOR I/O.
0105 309 : TO ACCOMPLISH THIS IT MUST MAKE THE PAGE VALID, EFFECTIVELY BY
0105 310 : FAULTING THE PAGE. BUT SINCE THIS ROUTINE DOES NOT WANT TO WAIT
0105 311 : FOR A PAGE TO FAULT, IT WILL ONLY MAKE THE PAGE VALID IF IT CAN
0105 312 : DO SO WITHOUT WAITING. OTHERWISE IT PASSES BACK THE CONDITION
0105 313 : THAT A FULL FLEDGED PAGE FAULT IS REQUIRED TO GET THE PAGE.
0105 314 :
0105 315 : IOLOCKPAG RECOGNIZES THE SPECIAL CASE OF A NON-RESIDENT
0105 316 : PAGE THAT IS GOING TO BE TOTALLY OVERWRITTEN BY THE I/O TRANSFER.
0105 317 : IN THIS CASE IT DEMAND ZEROES A PAGE INSTEAD OF READING IT IN.
0105 318 : HOWEVER, THIS OPTIMIZATION CAN BE INHIBITED IF THE PAGE IS ACTUALLY
0105 319 : GOING TO BE READ AND WRITTEN BY THE I/O DEVICE.
0105 320 :
0105 321 : CALLING SEQUENCE:
0105 322 :
0105 323 : BSBW MMG$IOLOCKPAG
0105 324 :
0105 325 : INPUT PARAMETERS:
0105 326 :
0105 327 : R2 = VIRTUAL ADDRESS (LOW 9 BITS = PAGTYP)
0105 328 : ONLY PROCESS OR SYSTEM PAGES, NO GLOBALS YET.
0105 329 : R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
0105 330 : R4 = PROCESS CONTROL BLOCK ADDRESS
0105 331 : R5 = PROCESS HEADER ADDRESS (SYSTEM SPACE)
0105 332 : R7 = <0> - SET IF MODIFYING THE PAGE (READING INTO MEMORY)
0105 333 : <1> - SET IF TRANSFERRING THE ENTIRE PAGE
0105 334 : <2> - SET TO INHIBIT NON-RESIDENT PAGE OPTIMIZATION (SEE ABOVE)
0105 335 : <3:7> = 0
0105 336 : IPL = SYNCH
0105 337 :
0105 338 : IMPLICIT INPUTS:
0105 339 :
0105 340 : NONE
0105 341 :
0105 342 : OUTPUT PARAMETERS:
0105 343 :
0105 344 : R0 = #SS$ NORMAL IF LOCKED OK
0105 345 : = 0 IF PAGE FAULT REQUIRED
0105 346 : R2,R3 PRESERVED
0105 347 :
0105 348 : IMPLICIT OUTPUTS:
0105 349 :
0105 350 : NONE
0105 351 :
0105 352 : COMPLETION CODES:
0105 353 :
0105 354 : NONE
0105 355 :
0105 356 : SIDE EFFECTS:
0105 357 :
0105 358 : NONE
0105 359 :
0105 360 :--
```



```
50 63 7B800000 8F CB 0105 362 MMG$IOLOCKPAG::
0105 363 .ENABL LSB
0105 364 BICL3 #^C<PTESM_VALID -;CHECK VALID BIT
010D 365 ! PTESM_TYP1 ! PTESM_TYPO -;GET PTE TYPE BITS
010D 366 ! PTESM_PGFLVB>,(R3),RO ;AND PFN/PAGE FILE VBN BITS TO RO
010D 367 BGEQ NOTVALID ;BRANCH IF PAGE NOT VALID
010F 368 :
010F 369 : PAGE IS VALID, LOCK IT FOR I/O
010F 370 :
50 50 15 00 EF 010F 371 EXTZV #PTESV_PFN,#PTESV_PFN,RO,RO ;RO=PFN
0000'CF 50 D1 0114 372 CMPL RO,W^MMG$GL_MAXPFN ;DOES THIS PAGE HAVE PFN DATABASE?
OC 14 0119 373 BGTR 10$ ;NO, SKIP LOCK FOR PAGE
011B 374 LOCKPAGE:
011B 375 INCW @W^PFNSAW_REFCNT[RO] ;I/O REFERENCE
0120 376 BLBC R7,10$ ;BRANCH IF NOT MODIFYING PAGE
0123 377 BISB #PTESM_MODIFY@-24,3(R3) ;SET MODIFY BIT IN PTE
0127 378 10$: MOVZWL #SS$_NORMAL,RO ;RETURN SUCCESS INDICATION
012A 379 RSB
012B 380 .DSABL LSB
012B 381 :
012B 382 : PAGE NOT VALID
012B 383 :
012B 384 NOTVALID:
51 50 05 6D 13 012B 385 BEQL DEMANDZERO ;BRANCH IF DEMAND ZERO PAGE
16 EE 012D 386 EXTV #PTESV_TYPO,#<PTESV_TYP1+1-PTESV_TYPO>,RO,R1
0132 387 :
0132 388 : R1 = SIGN EXTENDED VIELD OF BITS BETWEEN TYPO AND TYP1
0132 389 : IF R1 = 0 THEN BOTH BITS WERE ZERO
0132 390 : R1 NEGATIVE INDICATES TYP1 BIT SET
0132 391 : R1<0> IS THE TYPO BIT
0132 392 :
5F 12 0132 393 BNEQ NOTTRANSITION ;BRANCH IF NOT TRANSITION PAGE
0134 394 :
0134 395 : PAGE IS IN TRANSITION, RO = PFN
0134 396 : IN GETTING WORKING SET ENTRY, WE MUST BE CAREFUL THE PAGE IS NOT DISCARDED
0134 397 : BY THE DEAD PAGE TABLE SCAN
0134 398 :
0134 399 ASSUME PFNSC_ACTIVE EQ 7 ;ALL BITS SET
51 07 0000'DF40 8B 0134 400 BICB3 @W^PFNSAB_STATE[RO],#PFNSC_ACTIVE,R1 ;IS PAGE ACTIVE?
49 13 013B 401 BEQL ACTIVE ;BRANCH IF IT IS
OD BB 013D 402 PUSHR #^M<RO,R2,R3> ;SAVE SVAPTE, VA, PFN
FEFE' 30 013F 403 BSBW MMG$FREWSLE ;GET A WORKING SET LIST ENTRY
29 50 E9 0142 404 BLBC RO,30$ ;BRANCH IF MUST WAIT FOR ONE
08 BE 84400000 8F D3 0145 405 BITL #PTESM_VALID ! PTESM_TYPO ! PTESM_TYP1,@8(SP)
1F 12 014D 406 BNEQ 30$ ;NO LONGER TRANSITION - FAULT IT
01 BA 014F 407 POPR #^M<RO> ;RECOVER PFN
52 0000'DF40 03 00 EF 0151 408 EXTZV #PFNSV_LOC,#PFNS$LOC,@W^PFNSAB_STATE[RO],R2 ;LOCATION
0159 409 CASE R2,<-
0159 410 PFNLIST,- ;FREE PAGE LIST
0159 411 PFNLIST,- ;MODIFIED PAGE LIST
0159 412 20$,- ;BAD PAGE LIST
0159 413 RELEASEPEND,- ;PAGE WAITING TO BE RELEASED BY I/O DONE
0159 414 20$,- ;READ ERROR
0159 415 WRITEINPROG- ;WRITE IN PROGRESS
0159 416 >
52 8E 7D 0169 417 20$: MOVQ (SP)+,R2 ;RESTORE VA AND SVAPTE
22 11 016C 418 BRB MUSTFAULT ;RETURN 'MUST FAULT' STATUS
```



```
01  BA 016E 419 30$: POPR #^M<R0> ;CLEAN OFF SAVED PFN
F7  11 0170 420 BRB 20$ ;AND JOIN MUST FAULT CODE
    0172 421 :
    0172 422 : PAGE IS ON FREE OR MODIFIED PAGE LIST, R2 = LIST ID
    0172 423 :
    0172 424 PFNLIST:
FE8B' 30 0172 425 BSBW MMG$REMPFN ;REMOVE PAGE FROM SPECIFIED LIST
    0175 426 WRITEINPROG:
    0175 427 RELEASEPEND:
52  6E 7D 0175 428 MOVQ (SP),R2 ;RESTORE VA AND SVAPTE
FE85' 30 0178 429 BSBW MMG$MAKEWSLE ;MAKE NEW WORKING SET LIST ENTRY
52  8E 7D 017B 430 MOVQ (SP)+,R2 ;RESTORE VA AND SVAPTE
    017E 431 ASSUME PFN$V_DELCON EQ PFN$V_LOC+PFN$$LOC+1 ;DELCON IS 2ND BIT TO LEFT OF
    017E 432 ;BIT IN BETWEEN IS FOR LOC EXPANSION
05  00 07 F0 017E 433 INSV #PFN$C_ACTIVE,#PFN$V_LOC,#PFN$$LOC+2,- ;SET PAGE ACTIVE
    0182 434 @W^PFN$AB_STATE[R0] ;AND CLEAR DELCON
    0186 435 ACTIVE:
91  63 1F E3 0186 436 BBBS #PTESV_VALID,(R3),LOCKPAGE ;SET VALID BIT
    8F  11 018A 437 BRB LOCKPAGE ;FINISH UP BY LOCKING THE PAGE
    018C 438 :
    018C 439 : RESTORE VA AND SVAPTE AND FALL THROUGH TO MUSTFAULT1
    018C 440 :
    018C 441 MUSTFAULT3:
OC  BA 018C 442 POPR #^M<R2,R3> ;RESTORE VA AND SVAPTE
    018E 443 :
    018E 444 : POP 1 LONG WORD AND RETURN 'MUST FAULT' INDICATION
    018E 445 :
    018E 446 MUSTFAULT1:
8E  D5 018E 447 TSTL (SP)+ ;CLEAN UP STACK
    0190 448 :
    0190 449 : RETURN 'MUST FAULT' INDICATION
    0190 450 :
    0190 451 MUSTFAULT:
50  D4 0190 452 CLRL R0 ;INDICATE FAILURE
    05  05 0192 453 RSB ;
    0193 454 :
    0193 455 : MUST BE SECTION, PAGE FILE, OR GLOBAL PAGE
    0193 456 :
    0193 457 NOTRANSITION:
FB  14 0193 458 BGTR MUSTFAULT ;FAULT GLOBALS FOR NOW
    0195 459 :
    0195 460 : PAGE IS A SECTION OR PAGE FILE PAGE. IF TOTALLY OVERWRITING IT
    0195 461 : THEN SUPPLY A ZEROED PAGE INSTEAD OF FAULTING IT IN.
    0195 462 : (ASSUMING THIS OPTIMIZATION IS NOT INHIBITED BY BIT 2 IN R7 BEING SET)
    0195 463 :
57  03 91 0195 464 CMPB #3,R7 ;TOTALLY OVERWRITING THE PAGE?
    F6  12 0198 465 BNEQ MUSTFAULT ;BRANCH IF NOT, NEED TO READ IT
    019A 466 :
    019A 467 : DEMAND ZERO THIS PAGE, R0 IS BACKING STORE ADDRESS.
    019A 468 :
    019A 469 DEMANDZERO:
50  50 17 00 EF 019A 470 EXTZV #PFN$V_BAK,#PFN$$BAK,R0,R0 ;CALCULATE BACKING STORE ADR
    58  D4 019F 471 CLRL R8 ;INDICATE NOT CRF
    OD 50 16 E1 01A1 472 BBC #PTESV_TYPO,R0,20$ ;BRANCH IF PAGING FILE
    00 50 11 E4 01A5 473 BBSC #PTESV_DZRO,R0,10$ ;CLEAR DZRO BIT IF IT WAS SET
    0B 50 10 E1 01A9 474 10$:
    475 BBC #PTESV_CRF,R0,30$ ;BRANCH IF NOT CRF SECTION
```



```
50 08 18 1F A5 F0 01B2 478 20$: CRTL R0 ;GET SECTION TABLE INDEX
50 08 18 1F A5 F0 01B2 477 CLRL R0 ;NULL PAGING FILE FOR CRF SECTION
50 08 18 1F A5 F0 01B2 478 20$: INSV PHDSB_PAGFIL(R5),#PFNSV_PGFLX,#PFNS PGFLX,R0 ;PAGE FILE INDEX
50 08 18 1F A5 F0 01B2 479 30$: PUSH R0 ;SAVE BACKING STORE ADDRESS
50 08 18 1F A5 F0 01B2 480 MOVQ R2,-(SP) ;SAVE SVAPTE AND VA
50 08 18 1F A5 F0 01B2 481 BSBW MMGSFREWSLE ;GET A WORKING SET LIST ENTRY
50 08 18 1F A5 F0 01B2 482 MOVQ (SP),R2 ;RESTORE VA AND SVAPTE
50 08 18 1F A5 F0 01B2 483 BLBC R0,MUSTFAULT3 ;BRANCH IF NEED TO WAIT FOR WSLE
50 08 18 1F A5 F0 01B2 484 BSBW MMGSININEWPFN ;ALLOCATE AND INIT A PFN
50 08 18 1F A5 F0 01B2 485 MOVQ (SP)+,R2 ;RESTORE VA AND SVAPTE
50 08 18 1F A5 F0 01B2 486 BBS #31,R0,MUSTFAULT1 ;BRANCH IF MUST WAIT FOR ONE
50 08 18 1F A5 F0 01B2 487 MOVL R8,R1 ;SETION TABLE INDEX
50 08 18 1F A5 F0 01B2 488 BEQL 35$ ;NOT CRF
50 08 18 1F A5 F0 01B2 489 PUSH R2 ;SAVE VA
50 08 18 1F A5 F0 01B2 490 BSBW MMGSDECSECREP ;ONE LESS CRF PTE POINTING TO SECTION
50 08 18 1F A5 F0 01B2 491 POPL R2 ;RESTORE VA
50 08 18 1F A5 F0 01B2 492 35$: INCW @W^PFNSAW REFCNT[R0] ;2ND REFERENCE FOR I/O LOCK DOWN
50 08 18 1F A5 F0 01B2 493 BISB #PFNSC ACTIVE,@W^PFNSAB_STATE[R0] ;PAGE IS ACTIVE
50 08 18 1F A5 F0 01B2 494 MOVL (SP)+,@W^PFNSAL_BAK[R0] ;SET BACKING STORE ADDRESS
50 08 18 1F A5 F0 01B2 495 BICL3 #^C<PTESM PROT T PTESM_OWN>,(R3),R1 ;GET PROT AND OWNER FROM PTE
50 08 18 1F A5 F0 01B2 496 BBBS #PTESV_VALID,R1,40$ ;SET THE VALID BIT
50 08 18 1F A5 F0 01B2 497 40$: BBC #8,R7,45$ ;BRANCH IF NORMAL I/O OPERATION
50 08 18 1F A5 F0 01B2 498 BISL3 R0,R1,(R3) ;STORE THE VALID SHELL PFN
50 08 18 1F A5 F0 01B2 499 MOVL #3,R0 ;SET ALTERNATE SUCCESS
50 08 18 1F A5 F0 01B2 500 RSB ;RETURN
50 08 18 1F A5 F0 01B2 501
50 08 18 1F A5 F0 01B2 502 45$: BBBS #PTESV_MODIFY,R1,46$ ;SET THE MODIFY BIT
50 08 18 1F A5 F0 01B2 503 46$: BISL3 R0,R1,(R3) ;AND STORE IT WITH THE PFN
50 08 18 1F A5 F0 01B2 504
50 08 18 1F A5 F0 01B2 505 : AT THIS POINT DROP IPL TO ASTDEL SO SOMEONE ELSE MAY RUN IF NECESSARY
50 08 18 1F A5 F0 01B2 506 : THE PROCESS HEADER HAS BEEN NAILED DOWN BY THE I/O REF COUNT ON THIS
50 08 18 1F A5 F0 01B2 507 : PAGE SO IT CAN'T SLIP AWAY.
50 08 18 1F A5 F0 01B2 508 :
50 08 18 1F A5 F0 01B2 509 :
50 08 18 1F A5 F0 01B2 510 PUSH R #^M<R2,R3,R4,R5> ;PRESERVE THESE FROM MOVCS
50 08 18 1F A5 F0 01B2 511 BICW #VASM_BYTE,R2 ;SHUT OFF PAGE TYPE BITS IN VA
50 08 18 1F A5 F0 01B2 512 CMPL R2,(SF) ;IF NOT PROCESS PAGE
50 08 18 1F A5 F0 01B2 513 BNEQ 50$ ;THEN DON'T LOWER IPL
50 08 18 1F A5 F0 01B2 514 SETIPL #IPL$ ASTDEL ;MAKE PROCESS SCHEDULABLE AGAIN
50 08 18 1F A5 F0 01B2 515 MOVCS #0,(R2),#0,#^X200,(R2) ;ZERO THE PAGE
50 08 18 1F A5 F0 01B2 516 POPR #^M<R2,R3,R4,R5> ;RESTORE SAVED REGISTERS
50 08 18 1F A5 F0 01B2 517 SETIPL #IPL$ SYNCH ;BACK TO CALLED IPL
50 08 18 1F A5 F0 01B2 518 MOVZWL #SS$_NORMAL,R0 ;PAGE SUCCESSFULLY LOCKED FOR I/O
50 08 18 1F A5 F0 01B2 518 RSB ;RETURN TO CALLER
```



```
022E 520 .SBTTL UNLOCK PAGES AT COMPLETION OF I/O
022E 521 :++
022E 522 : FUNCTIONAL DESCRIPTION:
022E 523 :
022E 524 : UNLOCK PAGES THAT WERE PREVIOUSLY LOCKED FOR DIRECT I/O
022E 525 :
022E 526 : CALLING SEQUENCE:
022E 527 :
022E 528 : BSBW MMG$UNLOCK
022E 529 :
022E 530 :
022E 531 : INPUT PARAMETERS:
022E 532 :
022E 533 : R1 = PAGE COUNT
022E 534 : R3 = STARTING SVAPTE
022E 535 :
022E 536 : IMPLICIT INPUTS:
022E 537 : NONE
022E 538 :
022E 539 : OUTPUT PARAMETERS:
022E 540 : NONE
022E 541 :
022E 542 : IMPLICIT OUTPUTS:
022E 543 : NONE
022E 544 :
022E 545 : COMPLETION CODES:
022E 546 : NONE
022E 547 :
022E 548 : SIDE EFFECTS:
022E 549 : NONE
022E 550 :
022E 551 :--
022E 552 : .ENABL LSB
022E 553 :
022E 554 :
022E 555 : R0 = LAST SVAPTE INCLUSIVE FOR PAGE TABLE UNLOCKING (MMG$UNLOCK1 ONLY)
022E 556 : R1 = COUNT OF DATA PAGES TO UNLOCK
022E 557 : R3 = STARTING SVAPTE FOR UNLOCKING DATA PAGES AND PAGE TABLES
022E 558 :
022E 559 MMG$UNLOCK1:
022E 560 SAVIPL ;CALLER'S IPL TO STACK
022E 561 BRB 3$ ;JOIN COMMON CODE
022E 562
022E 563 MMG$UNLOCK::
022E 564 DSBINT #IPL$ SYNCH ;SAVE IPL AND RAISE TO SYNCH
022E 565 MOVAL -4(R3)(R1),R0 ;SVAPTE OF LAST PAGE OF XFER INCLUSIVE
022E 566 :
022E 567 : 0(SP) = CALLED IPL, CURRENTLY AT SYNCH
022E 568 : R0 = LAST SVAPTE INCLUSIVE
022E 569 : R1 = COUNT OF PAGES
022E 570 : R3 = FIRST SVAPTE
022E 571 :
022E 572 3$: PUSHR #M<R1,R3> ;PUSH START SVAPTE, PAGE COUNT
022E 573 BISW2 #X1FC,R0 ;ROUND UP TO END OF PAGE TABLE PAGE
022E 574 6$: BSBW MMG$DECPTRF ;UNDO PAGE TABLE REFERENCE
022E 575 MOVAL ^X200(R3),R3 ;GET NEXT PAGE TABLE PAGE ADDRESS
022E 576 CMPL R0,R3 ;PAST LAST PAGE OF XFER?
```

OB 11

50 FC A341 DE

50 01FC 8F BB 023E
FDB8' A8 0240
53 0200 C3 DE 0248
53 50 D1 024D


```
50      53      15      FDA2'      00      EF      D1      0263      584      20$:      CMPL      R0,W*MMG$GL_MAXPFN      30$      BGTR      30$      DECF      GTR=30$      MOVQ      R1,-(SP)      BSBW      MMG$RELPFN      25$:      MOVQ      (SP)+,R1      30$:      SOBGTR      R1,10$      ENBINT      RSB      05      0285      592      0286      593      0286      594      0286      595      0286      596      0286      597      0286      598      0286      599      0286      600      0286      601      0286      602      028A      603      028A      604      028A      605      028E      606      028E      607      577      578      579      580      581      582      583      584      585      586      587      588      589      590      591      592      593      594      595      596      597      598      599      600      601      602      603      604      605      606      607      BGEQU      6$      INCL      (SP)      BRB      25$      MOVL      (R2)+,R3      BLSS      20$      BSBW      IOC$PTETOPFN      EXTZV      #PTESV PFN,#PTES PFN,R3,R0 :PFN TO R0      CMPL      R0,W*MMG$GL_MAXPFN      BGTR      30$      DECF      GTR=30$      MOVQ      R1,-(SP)      BSBW      MMG$RELPFN      MOVQ      (SP)+,R1      SOBGTR      R1,10$      ENBINT      RSB      :PAGE REFERENCE OR SHARE COUNT WENT NEGATIVE, FATAL ERROR      :CALLING SEQUENCE:      BSBW      MMG$REFCNTNEG      BSBW      MMG$SHRCNTNEG      MMG$REFCNTNEG::      BUG_CHECK REFCNTNEG,FATAL      MMG$SHRCNTNEG::      BUG_CHECK SHRCNTNEG,FATAL      .END      :BRANCH ON NO      :INCREMENT PAGE COUNT, GO CHECK IF      :NO PAGES TO UNLOCK      :FETCH PAGE TABLE ENTRY      :BRANCH IF VALID      :RETURN R3<PFN> = PFN      :R0 :PFN TO R0      :DOES THIS PAGE HAVE PFN DATABASE?      :NO, NEXT PAGE      :ONE LESS REASON FOR THIS PAGE'S RESIDENCY      :SAVE REGISTERS      :RELEASE THE PAGE      :RESTORE PAGE COUNT AND STARTING SVAPTE      :ONCE AROUND FOR EACH PAGE      :RESTORE SAVED IPL      :R0 = PAGE FRAME NUMBER      :R0 = PAGE FRAME NUMBER      :PFN REFERENCE COUNT NEGATIVE      :PFN SHARE COUNT NEGATIVE
```


IOLOCK
Symbol table

- LOCK AND UNLOCK PAGES FOR I/O

I 1

16-SEP-1984 00:19:38 VAX/VMS Macro V04-00
5-SEP-1984 03:43:17 [SYS.SRC]IOLOCK.MAR;1Page 13
(6)

ACTIVE	00000186	R	02
BUGS_REFcntNEG	*****	X	02
BUGS_SHRCntNEG	*****	X	02
CTL\$GL_PHD	*****	X	02
DEMANDZERO	0000019A	R	02
EXESADJWSL	*****	X	02
FAULTREQUIRED	000000E9	R	02
IOCSPTETOPFN	*****	X	02
IOLOCKACCVIO	0000008A	R	02
IOLOCKEXIT	000000E1	R	02
IPL\$ASTDEL	= 00000002		
IPL\$SYNCH	= 00000008		
LOCKPAGE	0000011B	R	02
MMGSAL_SYSPCB	*****	X	02
MMGSDECPTRF	*****	X	02
MMGSDECSECF	*****	X	02
MMGSFREWSLE	*****	X	02
MMGSGL_MAXPFN	*****	X	02
MMGSININWPFN	*****	X	02
MMGSIOLOCK	00000000	RG	02
MMGSIOLOCKPAG	00000105	RG	02
MMGSLOCKPGTB	*****	X	02
MMGSMAKEWSLE	*****	X	02
MMGSREFcntNEG	00000286	RG	02
MMGSRELPFN	*****	X	02
MMGSREMPFN	*****	X	02
MMGS\$SHRCntNEG	0000028A	RG	02
MMGSUNLOCK	00000233	RG	02
MMGSUNLOCK1	0000022E	R	02
MUSTFAULT	00000190	R	02
MUSTFAULT1	0000018E	R	02
MUSTFAULT3	0000018C	R	02
NOTTRANSITION	00000193	R	02
NOTVALID	0000012B	R	02
PCBSL_PHD	= 0000006C		
PFNSAB_STATE	*****	X	02
PFNSAL_BAK	*****	X	02
PFNSAW_REFcnt	*****	X	02
PFNSC_ACTIVE	= 00000007		
PFNSS_BAK	= 00000017		
PFNSS_LOC	= 00000003		
PFNSS_PGFLX	= 00000008		
PFNSV_BAK	= 00000000		
PFNSV_DELCON	= 00000004		
PFNSV_LOC	= 00000000		
PFNSV_PGFLX	= 00000018		
PFNLIST	00000172	R	02
PHDSB_PAGFIL	= 0000001F		
PHDSW_WSDYN	= 0000000E		
PHDSW_WSLIST	= 00000008		
PHDSW_WSSIZE	= 00000050		
PRS_IPL	= 00000012		
PTESM_MODIFY	= 04000000		
PTESM_OWN	= 01800000		
PTESM_PGFLVB	= 003FFFFFF		
PTESM_PROT	= 78000000		
PTESM_TYPO	= 00400000		

PTESM_TYPI	= 04000000		
PTESM_VALID	= 80000000		
PTESM_PFN	= 00000015		
PTESV_CRF	= 00000010		
PTESV_DZRO	= 00000011		
PTESV_MODIFY	= 0000001A		
PTESV_PFN	= 00000000		
PTESV_TYPO	= 00000016		
PTESV_TYPI	= 0000001A		
PTESV_VALID	= 0000001F		
RELEASEPEND	00000175	R	02
SS\$ACCVIO	= 0000000C		
SS\$INSFWSL	= 0000011C		
SS\$NORMAL	= 00000001		
SWP\$GL_BALBASE	*****	X	02
TMP...	= 00000001		
VASM_BYTE	= 000001FF		
VASS_BYTE	= 00000009		
VASV_BYTE	= 00000000		
WRITEINPROG	00000175	R	02
WSL\$C_SYSTEM	= 00000002		

+-----+
! 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
\$MMGCOD	0000028E (654.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:01.45
Command processing	129	00:00:00.55	00:00:05.53
Pass 1	298	00:00:09.22	00:00:29.84
Symbol table sort	0	00:00:01.48	00:00:04.93
Pass 2	117	00:00:02.09	00:00:10.11
Symbol table output	11	00:00:00.07	00:00:00.16
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	588	00:00:13.49	00:00:52.04

The working set limit was 1350 pages.
53364 bytes (105 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 919 non-local and 31 local symbols.
607 source lines were read in Pass 1, producing 15 object records in Pass 2.
24 pages of virtual memory were used to define 23 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	14
-\$255\$DUA28:[SYS.LIB]STARLET.MLB;2	6
TOTALS (all libraries)	20

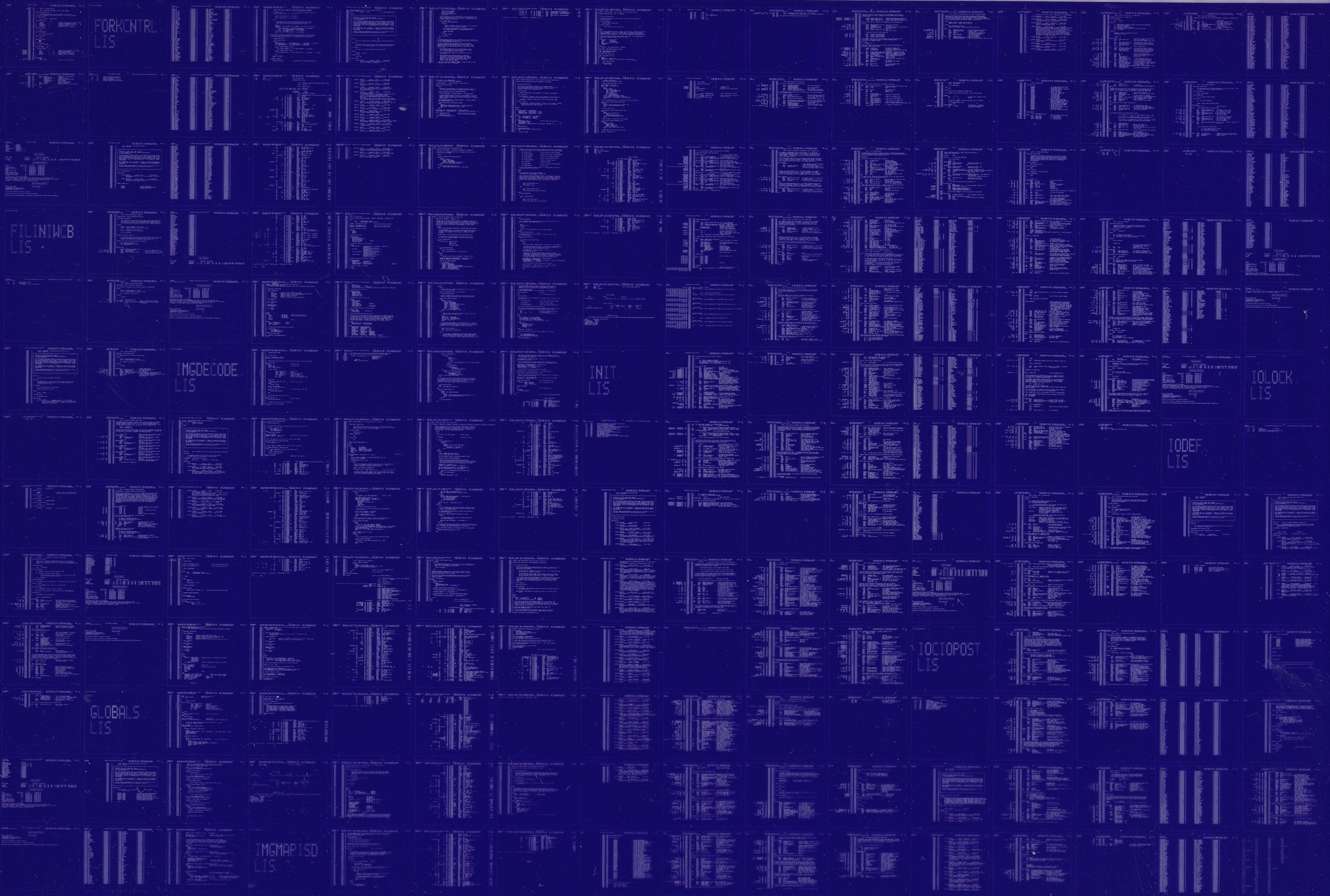
1044 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:IOLOCK/OBJ=OBJ\$:IOLOCK MSRC\$:IOLOCK/UPDATE=(ENH\$:IOLOCK)+EXECML\$/LIB

0375 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0376 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

