

A large, stylized letter 'W' is formed using the letters 'S' and 'Y'. The 'S' characters are arranged in a grid-like pattern to form the outer and inner curves of the 'W'. The 'Y' characters are used to form the vertical stems and the central connecting parts of the letter. The overall shape is a bold, blocky 'W' that fills most of the page.

[illegible][illegible]

(1)	43	HISTORY	; DETAILED
(1)	61	DECLARATIONS	
(1)	83	SCH\$RWAIT - RESOURCE WAIT	
(1)	121	SCH\$LOCKNOWAIT - LOCK MUTEX FOR WRITE WITHOUT WAITING	
(1)	169	SCH\$IOLOCKW - LOCK I/O DATA BASE MUTEX FOR WRITE	
(1)	205	SCH\$LOCKW - LOCK MUTEX FOR WRITE	
(1)	252	SCH\$IOLOCKR - LOCK I/O DATABASE MUTEX FOR READ	
(1)	288	SCH\$LOCKR - LOCK MUTEX FOR READ	
(1)	355	SCH\$RAVAIL - DECLARE RESOURCE AVAILABILITY	
(1)	381	SCH\$IOUNLOCK - UNLOCK I/O DATABASE MUTEX	
(1)	410	SCH\$UNLOCK - UNLOCK MUTEX	


```
0000 1
0000 2      .TITLE MUTEX - MUTEX WAIT ROUTINES
0000 3      .IDENT 'V04-000'
0000 4
0000 5
0000 6 :*****
0000 7 :*
0000 8 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 :* ALL RIGHTS RESERVED.
0000 11 :*
0000 12 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 :* TRANSFERRED.
0000 18 :*
0000 19 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 :* CORPORATION.
0000 22 :*
0000 23 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 :*
0000 26 :*
0000 27 :*****
0000 28
0000 29 :++
0000 30 : FACILITY: EXECUTIVE, SCHEDULER
0000 31
0000 32 : ABSTRACT:
0000 33 : THIS MODULE CONTAINS THE ROUTINES WHICH IMPLEMENT THE MUTEX
0000 34 : LOCK AND UNLOCK SERVICES FOR INTERNAL EXECUTIVE USE.
0000 35
0000 36
0000 37 : ENVIRONMENT:
0000 38 : MODE = KERNEL
0000 39
0000 40 :--
0000 41
0000 42 :
0000 43 : .PAGE
0000 44 : .SBTTL HISTORY ; DETAILED
0000 45 :
0000 46 : AUTHOR: R. HUSTVEDT CREATION DATE: 25-AUG-76
0000 47 :
0000 48 : MODIFIED BY:
0000 49 :
0000 50 : V03-003 SSA0022 Stan Amway 2-Apr-1984
0000 51 : Backed out SSA0005. It was temporary.
0000 52 :
0000 53 : V03-002 SSA0005 Stan Amway 10-Jan-1984
0000 54 : Added code to maintain PMS MWAIT transition counters.
0000 55 : The counters (in MDAT) and supporting code will be removed
0000 56 : before V4 release.
0000 57 :
0000 : V03-001 RQW0168 Ralph O. Weber 3-MAR-1983
```


0000	58	:
0000	59	:

Change $W^{\wedge}$ references to $G^{\wedge}$.

MU
Sy

BU
DY
EX
EX
IO
IP
IP
IP
LK
MT
MT
NO
PC
PC
PC
PC
PC
PC
PC
PR
PR
PR
PS
PS
RD
SC
SC
SC
SC
SC
SC
SC
SC
SC
SC
SC
SC
SC
SC
SC
SS
UN
WA
WA
WQ

PS
--
.
SA
AE


```

0000 61 .SBTTL DECLARATIONS
0000 62
0000 63 :
0000 64 : INCLUDE FILES:
0000 65 :
0000 66
0000 67 $DYNDEF ; STRUCTURE TYPE DEFINITIONS
0000 68 $IPLDEF ; IPL DEFINITIONS
0000 69 $MTXDEF ; MUTEX DEFINITIONS
0000 70 $PCBDEF ; PCB DEFINITIONS
0000 71 $PRDEF ; PROCESSOR REGISTER DEFINITIONS
0000 72 $PRIDEF ; PRIORITY INCR CLASS DEFS
0000 73 $PSLDEF ; PSL DEFINITIONS
0000 74 $SSDEF ; SYSTEM STATUS CODES
0000 75 $STATEDEF ; SCHEDULER STATE DEFS
0000 76 $WQHDEF ; WAIT QUEUE HEADER DEFS
0000 77 :
0000 78 : EQUATED SYMBOLS
0000 79 :
0000 80
00000000 81 .PSECT AEXENONPAGED,BYTE ; NONPAGED EXEC

```



```
0000 83 .SBTTL SCH$RWAIT - RESOURCE WAIT
0000 84
0000 85 :++
0000 86 : FUNCTIONAL DESCRIPTION:
0000 87 : SCH$RWAIT SUSPENDS THE EXECUTION OF A PROCESS UNTIL REQUIRED
0000 88 : RESOURCES ARE AVAILABLE.
0000 89 :
0000 90 : CALLING SEQUENCE:
0000 91 : SETIPL/DSBINT #IPL$_SYNCH
0000 92 : PUSHL <PSL>
0000 93 : BSB/JSB SCH$RWAIT
0000 94 :
0000 95 : INPUT PARAMETERS:
0000 96 : R0 - RESOURCE NUMBER FOR WHICH TO WAIT
0000 97 : R4 - PCB ADDRESS
0000 98 : 00(SP) - PC AT WHICH TO RESUME
0000 99 : 04(SP) - PSL WITH WHICH TO RESUME
0000 100 :
0000 101 : IMPLICIT INPUTS:
0000 102 : SCH$GQ_MWAIT - MUTEX WAIT QUEUE HEADER
0000 103 : PCB OF CURRENT PROCESS
0000 104 :
0000 105 : OUTPUTS:
0000 106 : R0-R3 PRESERVED
0000 107 :
0000 108 : IMPLICIT OUTPUTS:
0000 109 : *** TBS ***
0000 110 :
0000 111 : SIDE EFFECTS:
0000 112 : *** TBS ***
0000 113 :
0000 114 : --
0000 115 :
0000 116 SCH$RWAIT::
0000 117 BBSSI R0,G^SCH$GL_RESMASK,10$ :;; RESOURCE WAIT ENTRY POINT
0008 118 10$: BRB WAITR :;; SET WAITING FLAG
000A 119 :;; AND ENTER WAIT STATE
```

00 00000000'GF 50 E6 0000
7E 11 0008
000A


```
000A 121 .SBTTL SCH$LOCKKWAIT - LOCK MUTEX FOR WRITE WITHOUT WAITING
000A 122
000A 123 :++
000A 124 : FUNCTIONAL DESCRIPTION:
000A 125 : SCH$LOCKKWAIT LOCKS THE SPECIFIED MUTEX FOR EXCLUSIVE WRITE ACCESS
000A 126 : TO THE PROTECTED STRUCTURE. IF ANOTHER PROCESS HAS ALREADY CLAIMED
000A 127 : THE MUTEX, THEN THIS ROUTINE RETURNS A FAILURE INDICATION.
000A 128 :
000A 129 :
000A 130 :
000A 131 : CALLING SEQUENCE:
000A 132 : BSB/JSB SCH$LOCKKWAIT
000A 133 :
000A 134 :
000A 135 : INPUT PARAMETERS:
000A 136 : R0 - ADDRESS OF MUTEX
000A 137 : R4 - PCB ADDRESS OF CURRENT PROCESS
000A 138 :
000A 139 : IMPLICIT INPUTS:
000A 140 : SCH$GQ_MWAIT - MUTEX WAIT QUEUE HEADER
000A 141 : PCB OF CURRENT PROCESS
000A 142 : MUTEX LOCATED BY R0
000A 143 :
000A 144 : OUTPUTS:
000A 145 : R0 LOW BIT SET IF LOCKED SUCCESSFULLY
000A 146 : LOW BIT CLEAR IF MUTEX IN USE
000A 147 : R1-R3 PRESERVED
000A 148 : IPL = ASTDEL
000A 149 :
000A 150 : IMPLICIT OUTPUTS:
000A 151 : *** TBS ***
000A 152 :
000A 153 : SIDE EFFECTS:
000A 154 : *** TBS ***
000A 155 :
000A 156 : --
000A 157 SCH$LOCKKWAIT::
000A 158 SETIPL #IPL$ SYNCH
000A 159 BBSSI #MTX$V WRT,(R0),20$
000A 160 INCW MTX$W_OWNCNT(R0)
000A 161 BNEQ 10$
000A 162 MOVZWL #SS$_NORMAL,R0
000A 163 BRB LKEX
000A 164 10$: DECW MTX$W_OWNCNT(R0)
000A 165 20$: CLRL R0
000A 166 SETIPL #IPL$_ASTDEL
000A 167 RSB
000A 158 : RAISE TO SYNCH IPL
000A 159 : SET WRITE PENDING
000A 160 : RAISE OWNER COUNT
000A 161 : RETURN FAILURE IF BUSY
000A 162 : INDICATE SUCCESSFUL COMPLETION
000A 163 : AND MERGE WITH COMMON EXIT CODE
000A 164 : CORRECT COUNT
000A 165 : SET FAILURE RETURN INDICATION
000A 166 : LOWER TO ASTDEL
000A 167 : AND RETURN

OB 60 10 E6 000D 159
60 B6 0011 160
05 12 0013 161
50 01 3C 0015 162
32 11 0018 163
60 B7 001A 164 10$:
50 D4 001C 165 20$:
001E 166
05 0021 167
```



```
0022 169 .SBTTL SCH$IOLOCKW - LOCK I/O DATA BASE MUTEX FOR WRITE
0022 170 :++
0022 171 : FUNCTIONAL DESCRIPTION:
0022 172 : SCH$IOLOCKW RETURNS TO THE CALLER WHEN THE I/O DATABASE MUTEX
0022 173 : HAS BEEN LOCKED FOR WRITE ASSURING EXCLUSIVE ACCESS.
0022 174 :
0022 175 :
0022 176 :
0022 177 : CALLING SEQUENCE:
0022 178 : BSB/JSB SCH$IOLOCKW
0022 179 :
0022 180 :
0022 181 : INPUT PARAMETERS:
0022 182 : R4 - PCB ADDRESS OF CURRENT PROCESS
0022 183 :
0022 184 : IMPLICIT INPUTS:
0022 185 : SCH$GQ_MWAIT - MUTEX WAIT QUEUE HEADER
0022 186 : PCB OF CURRENT PROCESS
0022 187 : I/O DATABASE MUTEX
0022 188 :
0022 189 : OUTPUTS:
0022 190 : R0 = ADDRESS OF I/O DATABASE MUTEX
0022 191 : R1-R3 PRESERVED
0022 192 : IPL = ASTDEL
0022 193 :
0022 194 : IMPLICIT OUTPUTS:
0022 195 : *** TBS ***
0022 196 :
0022 197 : SIDE EFFECTS:
0022 198 : *** TBS ***
0022 199 :
0022 200 :--
0022 201 :
0022 202 SCH$IOLOCKW::
50 00000000'EF 9E 0022 203 MOVAB IOC$GL_MUTEX,R0 ; LOCK I/O DATA BASE FOR WRITE ACCESS
; GET ADDRESS OF I/O DATABASE MUTEX
```



```
0029 205 .SBTTL SCH$LOCKW - LOCK MUTEX FOR WRITE
0029 206 :++
0029 207 : FUNCTIONAL DESCRIPTION:
0029 208 : SCH$LOCKW RETURNS TO THE CALLER WHEN THE SPECIFIED MUTEX
0029 209 : HAS BEEN LOCKED FOR WRITE ASSURING EXCLUSIVE ACCESS TO THE
0029 210 : PROTECTED STRUCTURE.
0029 211 :
0029 212 :
0029 213 :
0029 214 : CALLING SEQUENCE:
0029 215 : BSB/JSB SCH$LOCKW
0029 216 :
0029 217 :
0029 218 : INPUT PARAMETERS:
0029 219 : R0 - ADDRESS OF MUTEX
0029 220 : R4 - PCB ADDRESS OF CURRENT PROCESS
0029 221 :
0029 222 : IMPLICIT INPUTS:
0029 223 : SCH$GQ_MWAIT - MUTEX WAIT QUEUE HEADER
0029 224 : PCB OF CURRENT PROCESS
0029 225 : MUTEX LOCATED BY R0
0029 226 :
0029 227 : OUTPUTS:
0029 228 : R0-R3 PRESERVED
0029 229 : IPL = ASTDEL
0029 230 :
0029 231 : IMPLICIT OUTPUTS:
0029 232 : *** TBS ***
0029 233 :
0029 234 : SIDE EFFECTS:
0029 235 : *** TBS ***
0029 236 :
0029 237 : --
0029 238 :
0029 239 SCH$LOCKW::
0029 240 10$: SETIPL #IPL$ SYNCH ; LOCK MUTEX FOR WRITE
0029 241 BBSSI #MTX$W_WRT,(R0),30$ ; RAISE TO SYNCH IPL
0029 242 INCW MTX$W_OWN CNT(R0) ; SET WRITE PENDING
0029 243 BNEQ 20$ ; RAISE OWNER COUNT
0029 244 BRB LKEX ; WAIT IF BUSY
0029 245 ; MERGE WITH COMMON EXIT CODE
0029 246 20$: ; MUST WAIT FOR EXCLUSIVE USE
0029 247 DECW MTX$W_OWN CNT(R0) ; CORRECT COUNT
0029 248 30$: BSBB WAITM ; AND WAIT FOR MUTEX
0029 249 BRB 10$ ; REPEAT LOCK ATTEMPT WHEN
0029 250 ; RESCHEDULED
```

08 60 10 E6 002C 241
60 B6 0030 242
02 12 0032 243
16 11 0034 244
0036 245
60 B7 0036 246
43 10 0038 247
ED 11 003A 248
003C 249
003C 250


```
003C 252 .SBTTL SCH$IOLOCKR - LOCK I/O DATABASE MUTEX FOR READ
003C 253 :++
003C 254 : FUNCTIONAL DESCRIPTION:
003C 255 : SCH$IOLOCKR RETURNS TO THE CALLER WHEN NO WRITERS OWN THE I/O
003C 256 : DATABASE MUTEX THUS ASSURING THE I/O DATABASE WILL REMAIN UN-
003C 257 : CHANGED UNTIL THE MUTEX IS RELEASED. IPL IS RAISED TO PREVENT
003C 258 : AST DELIVERY WHILE THE MUTEX IS OWNED AND THE PROCESS WILL NOT
003C 259 : BE OUTSWAPPED.
003C 260 :
003C 261 : CALLING SEQUENCE:
003C 262 : BSB/JSB SCH$IOLOCKR
003C 263 :
003C 264 : INPUT PARAMETERS:
003C 265 : R4 - CURRENT PROCESS PCB ADDRESS
003C 266 :
003C 267 : IMPLICIT INPUTS:
003C 268 : SCH$GQ_MWAIT - MUTEX WAIT QUEUE HEADER
003C 269 : PCB OF CURRENT PROCESS
003C 270 : I/O DATABASE MUTEX
003C 271 :
003C 272 : OUTPUTS:
003C 273 : R0 = ADDRESS OF I/O DATABASE MUTEX
003C 274 : R1-R3 PRESERVED
003C 275 : IPL = ASTDEL
003C 276 :
003C 277 : IMPLICIT OUTPUTS:
003C 278 : *** TBS ***
003C 279 :
003C 280 : SIDE EFFECTS:
003C 281 : *** TBS ***
003C 282 :
003C 283 : --
003C 284 :
003C 285 SCH$IOLOCKR::
50 00000000'EF 9E 003C 286 MOVAB IOC$GL_MUTEX,R0 ; LOCK I/O DATABASE FOR READ ACCESS
; GET ADDRESS OF I/O DATA BASE MUTEX
```



```
0043 288 .SBTTL SCH$LOCKR - LOCK MUTEX FOR READ
0043 289 :++
0043 290 : FUNCTIONAL DESCRIPTION:
0043 291 : SCH$LOCKR RETURNS TO THE CALLER WHEN NO WRITERS OWN THE
0043 292 : SPECIFIED MUTEX. THUS THE STRUCTURE PROTECTED BY THE MUTEX
0043 293 : WILL REMAIN UNCHANGED UNTIL THE MUTEX IS RELEASED. IPL IS
0043 294 : RAISED TO PREVENT AST DELIVERY WHILE THE MUTEX IS OWNED AND
0043 295 : THE PROCESS WILL NOT BE OUTSWAPPED.
0043 296 :
0043 297 : CALLING SEQUENCE:
0043 298 : BSB/JSB SCH$LOCKR
0043 299 :
0043 300 : INPUT PARAMETERS:
0043 301 : R0 - ADDRESS OF MUTEX
0043 302 : R4 - CURRENT PROCESS PCB ADDRESS
0043 303 :
0043 304 : IMPLICIT INPUTS:
0043 305 : SCH$GQ_MWAIT - MUTEX WAIT QUEUE HEADER
0043 306 : PCB OF CURRENT PROCESS
0043 307 : MUTEX
0043 308 :
0043 309 : OUTPUTS:
0043 310 : R0-R3 PRESERVED
0043 311 : IPL = ASTDEL
0043 312 :
0043 313 : IMPLICIT OUTPUTS:
0043 314 : *** TBS ***
0043 315 :
0043 316 : SIDE EFFECTS:
0043 317 : *** TBS ***
0043 318 :
0043 319 :--
0043 320
0043 321 SCH$LOCKR::
0043 322 SETIPL #IPL$ SYNCH ; LOCK MUTEX FOR READ
30 60 10 E0 0046 323 BBS #MTX$V_WRT,(R0),RDWAIT ; RAISE TO SYNCH IPL
004A 324 ; WAIT IF WRITE PENDING OR
004A 325 INCW MTX$W_OWN CNT(R0) ; IN PROGRESS
0A A4 0C 91 004C 326 LKEX: CMPB #DYN$C_PCB,PCB$B_TYPE(R4) ; INCREASE OWNER COUNT
0050 327 BNEQ 20$ ; CHECK FOR PCB
0052 328 INCW PCB$W_MTXCNT(R4) ; BUG CHECK IF NOT PCB
01 0E A4 B1 0055 329 CMPW PCB$W_MTXCNT(R4),#1 ; NOTE IN PCB ALSO
0059 330 BNEQ 10$ ; IS THIS THE FIRST MUTEX IT OWNS?
28 A4 0B A4 90 005B 331 MOVB PCB$B_PRI(R4),PCB$B_PRISAV(R4) ; BR IF OWNS MORE THAN 1 MUTEX
29 A4 2F A4 90 0060 332 MOVB PCB$B_PRI(R4),PCB$B_PRIBSAV(R4) ; SAVE CURRENT PRIORITY
0B A4 10 91 0065 333 CMPB #16,PCB$B_PRI(R4) ; IS THIS A REAL TIME PROCESS?
0069 334 BGTRU 10$ ; BR IF SO
0B A4 0F 90 006B 335 MOVB #15,PCB$B_PRI(R4) ; ELSE FORCE TO LOWEST RT PRIORITY
2F A4 0F 90 006F 336 MOVB #15,PCB$B_PRIB(R4) ; AND SET PRIORITY BASE TO RT
0073 337 10$: SETIPL #IPL$ ASTDEL ; DROP TO ASTDEL IPL
0076 338 RSB ; AND RETURN
00AC 05 0077 339 20$: BRW NOTPCB
007A 340
007A 341 RDWAIT: ; MUST WAIT FOR READ
C6 AF DF 007A 342 PUSHAL SCH$LOCKR ; RETRY AFTER WAIT
007D 343
007D 344 WAITM: ; WAIT FOR MUTEX TO FREE
```


04	AE	05	10	04	AE	DD	007D	345	PUSHL	(SP)	::: FORM PC, PSL ON STACK
					DC	007F	346	MOVPSL	4(SP)	::: BUILD PSL	
					FO	0082	347	INSV	#IPL\$ ASTDEL, #PSL\$V_IPL, #PSL\$S IPL, 4(SP)	::: SET IPL TO ASTDEL	
					DO	0088	348	MOVL	RO, PCB\$E EFWM(R4)	::: SAVE ADDRESS OF MUTEX	
					OE	008C	349	INSQUE	(R4), G^SCH\$GQ MWAIT	::: INSERT AT HEAD OF WAIT QUEUE	
					B6	0093	350	INCW	G^SCH\$GQ MWAIT+WQH\$W WQCNT	::: INCREMENT COUNT IN QUEUE	
					B0	0099	351	MOVW	#SCH\$C MWAIT, PCB\$W_STATE(R4)	::: SET STATE	
					31	009D	352	BRW	SCH\$WAITL	::: WAIT WITH STACK CLEAN, STATE SET	
					00A0		353				

```

00A0 355      .SBTTL  SCH$RAVAIL - DECLARE RESOURCE AVAILABILITY
00A0 356
00A0 357      :++
00A0 358      : FUNCTIONAL DESCRIPTION:
00A0 359      :   SCH$RAVAIL IS CALLED TO SIGNAL THE AVAILABILITY OF THE SPECIFIED
00A0 360      :   RESOURCE AND RELEASE ANY WAITING PROCESSES.
00A0 361      :
00A0 362      : CALLING SEQUENCE:
00A0 363      :   BSB/JSB SCH$RAVAIL
00A0 364      :
00A0 365      : INPUT PARAMETERS:
00A0 366      :   RO - RESOURCE NUMBER
00A0 367      :
00A0 368      : IMPLICIT OUTPUTS:
00A0 369      :   *** TBS ***
00A0 370      :
00A0 371      : SIDE EFFECTS:
00A0 372      :   *** TBS ***
00A0 373      :
00A0 374      :--
00A0 375
00A0 376 SCH$RAVAIL::      : DECLARE RESOURCE AVAILABILITY
7D 00000000'GF 50 E7 00A0 377      BBCCI  RO,G^SCH$GL_RESMASK,EXIT      : ; CLEAR AND TEST WAITING FLAG
45 11 00A8 378      DSBINT #IPL$ SYNCH      : ; BLOCK SYSTEM EVENTS
00AE 379      BRB  UNLOCK      : ; MERGE WITH COMMON CODE

```


MUTEX
V04-000

```

- MUTEX WAIT ROUTINES 16-SEP-1984 00:37:32 VAX/VMS Macro V04-00
SCH$IOUNLOCK - UNLOCK I/O DATABASE MUTEX 5-SEP-1984 03:45:26 [SYS.SRC]MUTEX.MAR:1

```

Page 12
(1)

```

00B0 381 .SBTTL SCH$IOUNLOCK - UNLOCK I/O DATABASE MUTEX
00B0 382 :++
00B0 383 : FUNCTIONAL DESCRIPTION:
00B0 384 : SCH$IOUNLOCK RELEASES OWNERSHIP OF THE I/O DATABASE MUTEX AND
00B0 385 : RE-ACTIVATES ANY WAITING PROCESSES IF THE MUTEX HAS BECOME
00B0 386 : AVAILABLE AS A CONSEQUENCE OF THIS UNLOCK REQUEST.
00B0 387 :
00B0 388 : CALLING SEQUENCE:
00B0 389 : BSB/JSB SCH$IOUNLOCK
00B0 390 :
00B0 391 : INPUT PARAMETERS:
00B0 392 : R4 - PCB ADDRESS OF CURRENT PROCESS
00B0 393 :
00B0 394 : IMPLICIT INPUTS:
00B0 395 : SCH$GQ_MWAIT - MUTEX WAIT QUEUE HEADER
00B0 396 : PCB OF CURRENT PROCESS
00B0 397 : I/O DATABASE MUTEX
00B0 398 :
00B0 399 : IMPLICIT OUTPUTS:
00B0 400 : *** TBS ***
00B0 401 :
00B0 402 : SIDE EFFECTS:
00B0 403 : *** TBS ***
00B0 404 :
00B0 405 :--
00B0 406 :
00B0 407 SCH$IOUNLOCK:: : UNLOCK I/O DATABASE MUTEX
50 00000000'EF 9E 00B0 408 MOVAB IOC$GL_MUTEX,R0 : GET ADDRESS OF I/O DATABASE MUTEX

```

```
00B7 410 .SBTTL SCH$UNLOCK - UNLOCK MUTEX
00B7 411 :++
00B7 412 : FUNCTIONAL DESCRIPTION:
00B7 413 : SCH$UNLOCK RELEASES OWNERSHIP OF THE SPECIFIED MUTEX AND
00B7 414 : RE-ACTIVATES ANY WAITING PROCESSES IF THE MUTEX HAS BECOME
00B7 415 : AVAILABLE AS A CONSEQUENCE OF THIS UNLOCK REQUEST.
00B7 416 :
00B7 417 : CALLING SEQUENCE:
00B7 418 : BSB/JSB SCH$UNLOCK
00B7 419 :
00B7 420 : INPUT PARAMETERS:
00B7 421 : R0 - MUTEX ADDRESS
00B7 422 : R4 - PCB ADDRESS OF CURRENT PROCESS
00B7 423 :
00B7 424 : IMPLICIT INPUTS:
00B7 425 : SCH$GQ_MWAIT - MUTEX WAIT QUEUE HEADER
00B7 426 : PCB OF CURRENT PROCESS
00B7 427 : MUTEX
00B7 428 :
00B7 429 : IMPLICIT OUTPUTS:
00B7 430 : *** TBS ***
00B7 431 :
00B7 432 : SIDE EFFECTS:
00B7 433 : *** TBS ***
00B7 434 :
00B7 435 : --
00B7 436 :
00B7 437 SCH$UNLOCK::
00B7 438 DSBINT #IPL$ SYNCH : UNLOCK MUTEX
00BD 439 CMPB #DYN$C_PCB,PCBSB_TYPE(R4) : RAISE TO SYNCH IPL
00C1 440 BNEQ NOTPCB : STRUCTURE MUST BE PCB
00C3 441 DECB PCBSW_MTXCNT(R4) : NOTE UNLOCK IN PCB
00C6 442 BNEQ 10$ : MORE STILL OWNED
00C8 443 MOVB PCBSB_PRI$SAV(R4),PCBSB_PRI$ (R4) : RESTORE SAVED BASE PRIORITY
00CD 444 MOVB PCBSB_PRI$SAV(R4),R1 : GET ORIGINAL PRIORITY
00D1 445 MOVB R1,PCBSB_PRI(R4) : RESTORE IT
00D5 446 MOVB R1,G^SCH$GB PRI : AND ANNOUNCE IT
00DC 447 FFS #0,#32,G^SCH$GL_COMQS,R2 : FIND PRIORITY OF NEXT COMPUTABLE PROCESS
00E5 448 CMPB R1,R2 : CHECK FOR DELAYED PREEMPTION
00E8 449 BLEQU 10$ : NO, CONTINUE
00EA 450 SOFTINT #IPL$ SCHED : ELSE RESCHEDULE WHEN IPL DROPS
00ED 451 10$: DECB MTX$W_OWCNT(R0) : DECREMENT OWNERSHIP COUNT
00EF 452 BGEQ EXITN : EXIT IF NOT LAST
00F1 453 BBCCI #MTX$V_WRT,(R0),EXITN : EXIT IF NO WRITE IN PROGRESS
00F5 454 : OR PENDING
00F5 455 UNLOCK: PUSHR #^M<R0,R4> : SAVE PCB ADDRESS
00F7 456 MOVAL G^SCH$GQ_MWAIT,R3 : GET ADDRESS OF WAIT QUEUE
00FE 457 MOVL (R3),R4 : AND HEAD PCB
0101 458 MOVZBL #PRI$_RESAVL,R2 : SET PRIORITY INCREMENT CLASS
0104 459 10$: CMPL R3,R4 : CHECK FOR END OF QUEUE
0107 460 BEQL 30$ : YES, DONE
0109 461 CMPL (SP),PCBSL_EFWM(R4) : IS PROCESS WAITING FOR THIS MUTEX
010D 462 BNEQ 20$ : NO, SKIP IT
010F 463 PUSHL (R4) : SAVE FLINK
0111 464 BSBW SCH$CHSE : CHANGE TO EXECUTABLE STATE
0114 465 DECB WQH$W_WQCNTR(R3) : DECREASE QUEUE LENGTH
0117 466 POPR #^M<R4> : RESTORE FLINK
```

52 00000000'GF 20 00 52 03 1B 00E8 449 BLEQU 10\$

53 00000000'GF 54 63 D0 00FE 457 MOVL (R3),R4

4C A4 6E D1 0109 461 CMPL (SP),PCBSL_EFWM(R4)

FE EC 30 0111 464 BSBW SCH\$CHSE

08 A3 B7 0114 465 DECB WQH\$W_WQCNTR(R3)

10 BA 0117 466 POPR #^M<R4>


```

54  E9 11 0119 467 BRB 10$
    64 D0 011B 468 20$: MOVL (R4),R4
    E4 11 011E 469 BRB 10$
    11 BA 0120 470 30$: POPR #^M<R0,R4>
      0122 471 EXITN: ENBINT
      05 0125 472 EXIT: RSB
        0126 473
        0126 474 NOTPCB: BUG CHECK NOTPCB,FATAL
        012A 475 .END

```

::: AND CONTINUE
 ::: FLINK ON TO NEXT PCB
 ::: AND CONTINUE
 ::: RESTORE REGISTERS
 ::: ENABLE INTERRUPTS
 ; AND RETURN
 ; STRUCTURE NOT PCB

MUTEX
Symbol table

- MUTEX WAIT ROUTINES

C 15

16-SEP-1984 00:37:32 VAX/VMS Macro V04-00
5-SEP-1984 03:45:26 [SYS.SRC]MUTEX.MAR;1

Page 15
(1)

BUG\$ NOTPCB	*****	X	02
DYN\$C_PCB	= 0000000C		
EXIT	00000125	R	02
EXITN	00000122	R	02
IOCSGL_MUTEX	*****	X	02
IPL\$ _ASTDEL	= 00000002		
IPL\$ _SCHED	= 00000003		
IPL\$ _SYNCH	= 00000008		
LKEX	0000004C	R	02
MTX\$V_WRT	= 00000010		
MTX\$W_OWNCNT	= 00000000		
NOTPCB	00000126	R	02
PCBSB_PRI	= 0000000B		
PCBSB_PRI8	= 0000002F		
PCBSB_PRI8AV	= 00000029		
PCBSB_PRI8AV	= 00000028		
PCBSB_TYPE	= 0000000A		
PCBSL_EFWM	= 0000004C		
PCBSW_MTXCNT	= 0000000E		
PCBSW_STATE	= 0000002C		
PR\$ _IPL	= 00000012		
PR\$ _SIRR	= 00000014		
PR\$ _RESAVL	= 00000002		
PSL\$S_IPL	= 00000005		
PSL\$V_IPL	= 00000010		
RDWAIT	0000007A	R	02
SCH\$SCHSE	*****	X	02
SCH\$C_MWAIT	= 00000002		
SCH\$GB_PRI	*****	X	02
SCH\$GL_COMQS	*****	X	02
SCH\$GL_RESMASK	*****	X	02
SCH\$GQ_MWAIT	*****	X	02
SCH\$IOLOCKR	0000003C	RG	02
SCH\$IOLOCKW	00000022	RG	02
SCH\$IOUNLOCK	000000B0	RG	02
SCH\$LOCKR	00000043	RG	02
SCH\$LOCKW	00000029	RG	02
SCH\$LOCKWNOWAIT	0000000A	RG	02
SCH\$RAVAIL	000000A0	RG	02
SCH\$RWAIT	00000000	RG	02
SCH\$UNLOCK	000000B7	RG	02
SCH\$WAITL	*****	X	02
SS\$ NORMAL	= 00000001		
UNLOCK	000000F5	R	02
WAITM	0000007D	R	02
WAITR	00000088	R	02
WQH\$W_WQCNT	= 00000008		

+-----+
! 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				
AEXENONPAGED	0000012A (298.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				

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

Phase	Page faults	CPU Time	Elapsed Time
-----	-----	-----	-----
Initialization	29	00:00:00.06	00:00:01.92
Command processing	120	00:00:00.49	00:00:05.19
Pass 1	285	00:00:08.42	00:00:27.52
Symbol table sort	0	00:00:01.22	00:00:04.36
Pass 2	92	00:00:01.82	00:00:05.21
Symbol table output	6	00:00:00.08	00:00:00.57
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	536	00:00:12.13	00:00:44.79

The working set limit was 1350 pages.

48294 bytes (95 pages) of virtual memory were used to buffer the intermediate code.

There were 50 pages of symbol table space allocated to hold 872 non-local and 12 local symbols.

475 source lines were read in Pass 1, producing 13 object records in Pass 2.

22 pages of virtual memory were used to define 21 macros.

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

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	12
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	18

979 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

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

0377 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

