


```

SSSSSSSS YY YY SSSSSSSS WW WW AAAAAA IIIIII TTTTTTTTTT
SSSSSSSS YY YY SSSSSSSS WW WW AAAAAA IIIIII TTTTTTTTTT
SS SS YY YY SS SS WW WW AA AA II II TT
SS SS YY YY SS SS WW WW AA AA II II TT
SS SS YY YY SS SS WW WW AA AA II II TT
SSSSSSS YY YY SSSSSS SS WW WW AA AA II II TT
SSSSSSS YY YY SSSSSS SS WW WW AA AA II II TT
SS SS YY YY SS SS WW WW AA AA II II TT
SSSSSSSS YY YY SSSSSSSS WW WW AA AA IIIIII TTTTTTTTTT
SSSSSSSS YY YY SSSSSSSS WW WW AA AA IIIIII TTTTTTTTTT

LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```

SYS
Pse

PSE

\$AE
AE)

Pha

In1
Con
Pas
Syn
Pas
Syn
Pse
Crc
Ass

The
657
The
725
22

Mac

\$2
-52
T01

119

The
MAC

(1)	83	DECLARATIONS
(1)	114	EXESWFLAND - WAIT FOR LOGICAL AND OF EVENT FLAGS
(1)	180	EXESWFLOR - WAIT FOR LOGICAL OR OF EVENTS
(1)	250	EXESWAITFR - WAIT FOR SINGLE EVENT
(1)	319	EXESWAIT - WAIT COMMON CODE
(1)	562	SCHSWAIT - PLACE PROCESS IN SELECTED WAIT QUEUE

```

0000 1      .NLIST  CND
0000 3      :TITLE  SYSWAIT EVENT FLAG WAIT SERVICES
0000 7      :IDENT  'V04-000'
0000 8      :*****
0000 9      :*
0000 10     :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 11     :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 12     :*  ALL RIGHTS RESERVED.
0000 13     :*
0000 14     :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 15     :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 16     :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 17     :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 18     :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 19     :*  TRANSFERRED.
0000 20     :*
0000 21     :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 22     :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 23     :*  CORPORATION.
0000 24     :*
0000 25     :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 26     :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 27     :*
0000 28     :*
0000 29     :*****
0000 30     :++
0000 32     : FACILITY: EXECUTIVE, EVENT FLAG SERVICES
0000 33     :
0000 34     : ABSTRACT: WAIT CONTAINS THE THREE FORMS OF EVENT FLAG WAIT
0000 35     : SYSTEM SERVICES WHICH PROVIDE FOR SINGLE EVENT WAIT AS
0000 36     : WELL AS COMBINATIONS OF MULTIPLE EVENTS.
0000 43     :
0000 44     :--
0000 45     :
0000 46     : AUTHOR:
0000 47     :   R.HUSTVEDT : VERSION
0000 48     :
0000 49     : MODIFIED BY:
0000 50     :
0000 51     :   V03-007 SSA0016      Stan Amway      8-Mar-1984
0000 52     :   Subtract IOTA from automatic working set adjustment
0000 53     :   time reference in PHD on any entry to SCHSWAITx code.
0000 54     :   (Acknowledgements go to Wayne Cardoza and Larry Kenah,
0000 55     :   who both collaborated on this change.)
0000 56     :
0000 57     :   V03-006 WMC0001      Wayne Cardoza    22-Feb-1984
0000 58     :   No reason to try to wake up swapper on every EFN wait.
0000 59     :
0000 60     :   V03-005 SSA0003      Stan Amway      5-Dec-1983
0000 61     :   Added support for outswap scheduling changes.
0000 62     :   Changed process wait code to store wait time in PCB
0000 63     :   as longword system absolute time.
0000 64     :
0000 65     :   V03-004 KDM003,      Kathleen D. Morse  14-Dec-1982
0000 66     :   Fix assembly switch for performance collection for
0000 67     :   kernel mode services executed on secondary processor.
0000 68     :

```

0000	69	:	V03-003	KDM0034	Kathleen D. Morse	13-Dec-1982
0000	70	:			Correct logic for secondary continuing execution of	
0000	71	:			a process after a WAITCHK request is done by the primary.	
0000	72	:				
0000	73	:	V03-002	KDM0030	Kathleen D. Morse	18-Nov-1982
0000	74	:			Add IFPRIMARY logic that allows primary to execute	
0000	75	:			secondary-specific code without turning into a secondary.	
0000	76	:				
0000	77	:	V03-001	KDM0018	Kathleen D. Morse	13-Oct-1982
0000	78	:			Add multi-processing switch, which generates these	
0000	79	:			system services for the secondary processor.	
0000	80	:				
0000	81	:				

```

0000 83      .SBTTL  DECLARATIONS
0000 84
0000 85      :
0000 86      : INCLUDE FILES:
0000 87      :
0000 88
0000 89      $CEBDEF      ;COMMON EVENT BLOCK DEFS
0000 90      $DYNDEF      ;DYNAMIC STRUCTURE TYPES
0000 91      $IPLDEF      ;IPL DEFINITIONS
0000 97      $PCBDEF      ;PCB DEFINITIONS
0000 98      $PHDDEF      ;PHD DEFINITIONS
0000 99      $PRDEF       ;PROCESSOR REGISTER DEFS
0000 100     $PSLDEF      ;PSL DEFINITIONS
0000 101     $$SHBDEF     ;SHARED MEMORY CONTROL BLK DEFS
0000 102     $$SHDDEF     ;SHARED MEMORY COMMON DATA PAGE
0000 103     $$$SDEF      ;STATUS CODE DEFINITIONS
0000 104     $$STATEDEF   ;STATE DEFINITIONS
0000 105     $WQHDEF      ;WAIT QUEUE HEADER DEFS
0000 106     :
0000 107     : EQUATES:
0000 108     :
00000004 0000 109     $FN=4      ;EVENT FLAG NUMBER
00000008 0000 110     MASK=8    ;WAIT MASK
00000000 111     .PSECT  AEXENONPAGED ;NON-PAGED

```

```

0000 114 .SBTTL EXESWFLAND - WAIT FOR LOGICAL AND OF EVENT FLAGS
0000 118 :++
0000 119 : FUNCTIONAL DESCRIPTION:
0000 120 :
0000 122 : EXESWFLAND RETURNS TO THE CALLER WHEN THE SET OF EVENT
0000 126 : FLAGS SELECTED BY THE MASK ARE ALL SET AND RETURNS THE
0000 127 : STATE OF ALL EVENT FLAGS IN THE SPECIFIED CLUSTER.
0000 128 :
0000 129 : CALLING SEQUENCE:
0000 130 :
0000 132 : CALLG  ARGLIST,EXESWFLAND
0000 136 :
0000 137 : INPUT PARAMETERS:
0000 138 :
0000 139 : 04(AP) - EVENT FLAG NUMBER SELECTING CLUSTER
0000 140 : 08(AP) - MASK SELECTING COMBINATION OF EVENTS
0000 141 : R4 - PCB ADDRESS OF CURRENT PROCESS
0000 142 :
0000 143 : OUTPUT PARAMETERS:
0000 144 :
0000 145 : R0 - COMPLETION STATUS CODE
0000 146 : CONDITION IS SATISFIED.
0000 147 :
0000 148 : SIDE EFFECTS:
0000 149 :
0000 150 : IF THE SET OF EVENT FLAGS SELECTED BY THE MASK ARE NOT
0000 151 : ALL SET, THEN THE PROCESS ISSUING THE WAIT SERVICE CALL WILL
0000 152 : BE PLACED IN A WAIT STATE.
0000 153 :
0000 154 : COMPLETION CODES:
0000 155 :
0000 156 : $$$_NORMAL - NORMAL SUCCESSFUL COMPLETION
0000 157 : $$$_ILLEFC - ILLEGAL EVENT FLAG CLUSTER NUMBER.  EVENT NUMBER
0000 158 : NOT IN THE RANGE 0-127.
0000 159 : $$$_UNASEFC - UNASSIGNED EVENT FLAG CLUSTER.
0000 167 :
0000 168 :--
0000 169 :
51 01 003C 0000 171 EXESWFLAND:: ;WAIT FOR LOGICAL AND
04 04 11 0002 175 .WORD ^M<R2,R3,R4,R5> ;REGISTER SAVE MASK FOR R2-R5
0000 176 MOVL #1,R1 ;SET MODE TO WAITALL
0000 177 BRB WFR1 ;AND MERGE WITH COMMON CODE

```

```

0007 180 .SBTTL EXESWFLO - WAIT FOR LOGICAL OR OF EVENTS
0007 184 :++
0007 185 : FUNCTIONAL DESCRIPTION:
0007 186 :
0007 188 : EXESWFLO RETURNS TO THE CALLER WHEN ANY OF THE
0007 192 : EVENTS SELECTED BY THE MASK WITHIN THE SPECIFIED CLUSTER
0007 193 : ARE SET AND RETURNS THE STATE OF ALL 32 EVENT FLAGS IN THE
0007 194 : CLUSTER.
0007 195 :
0007 196 : CALLING SEQUENCE:
0007 197 :
0007 199 : CALLG  ARGLIST,EXESWFLO
0007 203 :
0007 204 : INPUT PARAMETERS:
0007 205 :
0007 206 : 04(AP) - EVENT FLAG NUMBER TO SELECT CLUSTER
0007 207 : 08(AP) - MASK SELECTING DESIRED COMBINATION OF EVENTS
0007 208 : R4 - PCB ADDRESS OF CURRENT PROCESS
0007 209 :
0007 210 : OUTPUT PARAMETERS:
0007 211 :
0007 212 : R0 - COMPLETION STATUS CODE
0007 213 : IS SATISFIED.
0007 214 :
0007 215 : COMPLETION CODES:
0007 216 :
0007 217 : $$$_NORMAL - NORMAL SUCCESSFUL COMPLETION
0007 218 : $$$_ILLEFC - ILLEGAL EVENT FLAG NUMBER NOT IN THE RANGE 0-127.
0007 219 : $$$_UNASEFC - UNASSIGNED EVENT FLAG CLUSTER.
0007 220 :
0007 221 : SIDE EFFECTS:
0007 222 :
0007 223 : THE PROCESS ISSUING THE SERVICE CALL IS BE PLACED IN A
0007 224 : WAIT STATE IF NONE OF THE SPECIFIED EVENTS ARE SET.
0007 232 :
0007 233 :--
0007 234 :
0007 236 EXESWFLO::
0007 240 .WORD ^M<R2,R3,R4,R5> ;WAIT FOR LOGICAL OR
0009 241 CLRL R1 ;REGISTER SAVE MASK FOR R2-R5
0008 242 WFRL: MOVL MASK(AP),R0 ;SET MODE TO WAIT ANY
000F 244 BRB EXESWAIT ;GET WAIT MASK
;MERGE WITH COMMON CODE

```

```

50 08 AC D0
    09 11
    003C

```



```

0011 250 .SBTTL EXESWAITFR - WAIT FOR SINGLE EVENT
0011 254 :++
0011 255 : FUNCTIONAL DESCRIPTION:
0011 256 :
0011 258 : EXESWAITFR RETURNS TO THE CALLER WHEN THE SPECIFIED SINGLE
0011 262 : EVENT FLAG IS SET. UPON RETURN THE STATE OF ALL 32 EVENT FLAGS
0011 263 : WITHIN THE CLUSTER CONTAINING THE SPECIFIED EVENT ARE RETURN.
0011 264 :
0011 265 : CALLING SEQUENCE:
0011 266 :
0011 268 : CALLG  ARGLIST,EXESWAITFR
0011 272 :
0011 273 : INPUT PARAMETERS:
0011 274 :
0011 275 : 04(AP)=EVENT FLAG NUMBER
0011 276 : R4 - PCB ADDRESS OF CURRENT PROCESS
0011 277 :
0011 278 : OUTPUT PARAMETERS:
0011 279 :
0011 280 : R0 - COMPLETION STATUS CODE
0011 281 : SATISFIED.
0011 282 :
0011 283 : SIDE EFFECTS:
0011 284 :
0011 285 : IF THE SPECIFIED EVENT FLAG IS NOT SET, THE PROCESS ISSUING THE
0011 286 : WAIT SYSTEM SERVICE WILL BE PLACED IN THE APPROPRIATE WAIT
0011 287 : STATE.
0011 288 :
0011 289 : COMPLETION CODES:
0011 290 :
0011 291 : $$$_NORMAL - NORMAL SUCCESSFUL COMPLETION
0011 292 : $$$_ILLEFC - ILLEGAL EVENT FLAG NUMBER NOT IN THE RANGE 0-127.
0011 293 : $$$_UNASEFC - UNASSIGNED EVENT FLAG CLUSTER.
0011 294 :
0011 295 : ENVIRONMENT:
0011 296 :
0011 298 : MODE=KERNEL
0011 302 :--
0011 303 :
0011 305 EXESWAITFR:: ;WAIT FOR SINGLE EVENT
0011 309 .WORD ^M<R2,R3,R4,R5> ;SAVE REGISTERS R2,R3,R4,R5
50 01 04 AC 51 003C 0013 310 CLRL R1 ;SET MODE
0015 311 ROTL EFN(AP),#1,R0 ;INIT MASK
001A 313 BRB EXESWAIT ;AND MERGE WITH COMMON CODE

```

```

001A 319 .SBTTL EXESWAIT - WAIT COMMON CODE
001A 323 :++
001A 324 : FUNCTIONAL DESCRIPTION:
001A 325 :
001A 326 : THIS IS THE COMMON WAIT CODE FOR ALL THE EVENT FLAG WAIT
001A 327 : SYSTEM SERVICES.
001A 328 :
001A 329 : INPUT PARAMETERS:
001A 330 :
001A 331 : O4(AP) = EVENT FLAG NUMBER
001A 332 : R0 = MASK SELECTING EVENTS OF INTEREST
001A 333 : R1 = ANY/ALL MODE SELECTOR
001A 334 : 0 => ANY
001A 335 : 1 => ALL
001A 336 : R4 = PCB ADDRESS OF CURRENT PROCESS
001A 337 :
001A 338 : IMPLICIT INPUTS:
001A 339 :
001A 340 : CEB IF NON-LOCAL CLUSTER.
001A 341 :
001A 342 : OUTPUT PARAMETERS:
001A 343 :
001A 344 : R0 - COMPLETION STATUS CODE
001A 345 : SATISFIED.
001A 353 :
001A 354 : --
001A 355 :
001A 357 EXESWAIT:
52 04 AC 98 001A 361 CVTBL EFN(AP),R2 ;WAIT COMMON CODE
1A 19 001E 362 BLSS 10$ ;GET CLUSTER NUMBER
52 52 FB 8F 78 0020 363 ASHL #-5,R2,R2 ;ILLEGAL IF NOT (0,1,2,3)
53 50 A442 DE 0025 364 MOVAL PCB$EFC(R4)[R2],R3 ;RIGHT ALIGN CLUSTER NUMBER
2E A4 52 90 002A 365 MOVB R2,PCB$B_WQFC(R4) ;POINTER TO PCB EVENT CLUSTER
15 52 F5 002E 366 SOBGTR R2,30$ ;SAVE WAIT CLUSTER NUMBER
52 0000000'GF 7E 0031 367 MOVAQ G$SCH$GQ_LEFWQ,R2 ;BR IF COMMON CLUSTER R2 = (2,3)
23 11 0038 368 BRB WAITCK ;SET WAIT QUEUE POINTER
50 00EC 8F 3C 003A 369 10$: MOVZWL #SS$_ILLEFC,R0 ;SET ERROR CODE FOR ILLEGAL CLUSTER
04 003F 370 RET ;AND EXIT
50 0234 8F 3C 0040 371 20$: MOVZWL #SS$_UNASEFC,R0 ;SET ERROR CODE FOR UNASSIGNED
04 0045 372 RET ;AND EXIT
52 63 10 C1 0046 373 30$: ADDL3 #CEB$L_EFC,(R3),R2 ;GET CEB ADDRESS FOR EVENT FLAGS
F4 18 004A 374 BGEQ 20$ ;CEB ASSIGNED (SYSTEM SPACE ADDRESS)
FA A2 2D 91 004C 375 CMPB #DYN$C_SLAVCEB,<CEB$B_TYPE-CEB$L_EFC>(R2) ;IS THIS IN SH MEM?
08 12 0050 376 BNEQ 40$ ;BR IF IN LOCAL MEMORY
53 30 A2 DO 0052 377 MOVL <CEB$L_MASTER-CEB$L_EFC>(R2),R3 ;GET ADR OF MASTER CEB
62 10 A3 DO 0056 378 MOVL CEB$L_EFC(R3),(R2) ;COPY EFC FROM MASTER TO SLAVE CEB
005A 379 ASSUME <CEB$L_EFC+4> EQ CEB$L_WQFL
53 82 DE 005A 380 40$: MOVAL (R2)+,R3 ;GET EVENT POINTER AND WAIT QUEUE ADDR
005D 381 ; R3=CEB$L_EFC, R2=CEB$L_WQFL
005D 382 :
005D 383 : R0 - MASK SELECTING EVENTS OF INTEREST
005D 384 : R1 - ANY/ALL MODE SELECTOR
005D 385 : R2 - ADDRESS OF WAIT QUEUE HEADER
005D 386 : R3 - ADDRESS OF EVENT FLAG VECTOR
005D 387 : R4 - PCB ADDRESS
005D 388 :
005D 389 WAITCK: ;CHECK FOR WAIT SATISFIED

```

				005D	390	SETIPL	#IPL\$ SYNCH		;BLOCK SCHEDULING ACTIVITY
	50	63	D3	0060	391	BITL	(R3),R0		;WAIT FOR LOGICAL OR MAY BE SATISFIED
		13	13	0063	392	BEQL	WAIT		;NO, MUST WAIT
		OB	51	E8	0065	BLBS	R1,WAITALL		; 1 => WAIT FOR ALL IN MASK
				0068	393				
	5D	OC	AD	D0	0068	NOWAIT:			
		SE	00	C0	006C	MOVL	12(FP),FP		;GET SAVED FRAME POINTER
		50	01	9A	006F	ADDL	S^#EXE\$C CMSTKSZ,SP		;CLEAN STACK TO PC,PSL
				02	0072	MOVZBL	#SS\$ _NORMAL,R0		;RETURN SUCCESS CODE
					0072	REI			;RETURN TO CALLER
					0073				
					0073				
					0073	WAITALL:			; WAIT FOR ALL SELECTED EVENTS
	50	63	CA	0073	411	BICL2	(R3),R0		; CLEAR BITS FOR ALREADY SET FLAGS
			FO	13	0076	BEQL	NOWAIT		; YES, DONT WAIT
24	A4	01	OD	51	FO	WAIT:	INSV	R1,#PCBSV_WALL,#1,PCBSL_	STS(R4) ;SET WAIT ALL FLAG
		4C	A4	50	D2		MCOML	R0,PCBSL_EFWM(R4)	;SAVE INVERTED WAIT MASK
					007E				

```

0082 562 .SBTTL SCH$WAIT - PLACE PROCESS IN SELECTED WAIT QUEUE
0082 563 :++
0082 568 : FUNCTIONAL DESCRIPTION:
0082 569 :
0082 571 : SCH$WAIT PLACES THE CURRENT PROCESS IN A WAIT QUEUE
0082 575 : SELECTED BY A WAIT QUEUE HEADER ADDRESS SUPPLIED IN A REGISTER
0082 576 : A NEW PROCESS IS THEN SELECTED FOR EXECUTION.
0082 577 :
0082 578 : CALLING SEQUENCE:
0082 579 :
0082 581 : JMP/BR SCH$WAIT
0082 585 :
0082 586 : INPUT PARAMETERS:
0082 587 :
0082 592 : EXESC_CMSTKSZ+00(SP) - PC AT WHICH TO RESTART PROCESS AFTER EF WAIT
0082 593 : EXESC_CMSTKSZ+04(SP) - PSL WITH WHICH TO RESTART PROCESS
0082 617 :
0082 618 : IMPLICIT INPUTS:
0082 619 :
0082 621 : NONE
0082 622 :
0082 639 : SIDE EFFECTS:
0082 640 :
0082 641 : THE PROECESS SPECIFIED BY THE PCB ADDRESS IN R4 IS PLACED
0082 643 : IN THE WAIT QUEUE LOCATED BY R2, ITS CONTEXT SAVED,
0082 644 : AND A NEW PROCESS SCHEDULED.
0082 653 :
0082 654 :--
0082 655 :
0082 657 SCH$WAIT:: :PLACE PROCESS IN WAIT STATE
0082 661 MOVL 12(SP),FP :RESTORE FP
0082 662 ADDL2 S^#EXESC_CMSTKSZ,SP :CLEAN UP KERNEL STACK
0082 663 SUBL2 #4,(SP) :BACK UP SAVED PC
0082 664 SCH$WAITK:: :WAIT WITH STACK ALREADY CLEAN
0082 670 INCW WQHSW_WQCNT(R2) :INCREMENT COUNT FOR QUEUE
0082 671 INSQUE (R4),WQHSW_WQFL(R2) :INSERT IN QUEUE
0082 672 MOVW WQHSW_WQSTATE(R2),PCBSW_STATE(R4) ;SET STATE FOR PROCESS
0097 673 :
0097 674 : THE STATE NUMBER IS CONTAINED
0097 675 : IN THE QUEUE HEADER
0097 676 :
0097 678 SCH$WAITL:: :WAIT WITH STATE SET, STACK CLEANED
0097 679 SVPCTX :SAVE PROCESS CONTEXT
0098 680 SCH$WAITM:: :ENTRY FOR MEMORY MANAGEMENT WAIT CODE
0098 681 MOVL PCBSL_PHD(R4),R5 :GET PROCESS HEADER ADDRESS
0098 683 MOVZWL G^SCH$GW_IOTA,R0 :Get a longword copy of IOTA
0098 684 ADDW R0,PHD$W_QUANT(R5) :CHARGE QUOTA FOR VOLUNTARY WAIT
0098 685 SUBL2 R0,PHD$W_TIMREF(R5) :Adjust AWSA time reference
0098 686 MOVL G^EXESGL_ABSTIM,PCBSL_WAITIME(R4) ;RECORD TIME AT WAIT START
0098 687 CMPB #4,PHD$B_ASTLVL(R5) :NULL ASTLVL?
0098 688 BNEQ 20$ :NO, DO LONG CHECK
0098 690 10$: BRW W^SCH$SCHED ;GO SCHEDULE NEXT PROCESS
0098 696 :
0098 697 20$: MOVZBL PHD$B_ASTLVL(R5),R0 ;FETCH AND ZERO EXTEND PENDING ASTLVL
0098 698 CMPZV #PSL$V_CURMOD,#PSL$S_CURMOD,PHD$W_PSL(R5),R0 ;COMPARE WITH WAIT
0098 699 :ACCESS MODE
0098 700 BLSS 10$ ;BRANCH IF AST NOT DELIVERABLE

```

```

00CD 701
00CD 702 :
00CD 703 : Test for assumptions that are being made about the layout of the
00CD 704 : PSL that enables the next instruction to work correctly.
00CD 705 :
00CD 706 : o IPL field begins on a byte boundary
00CD 707 : o IPL field fits into a single byte
00CD 708
00CD 709
00CD 710 ASSUME <<<PSL$V_IPL/8>*8> - PSL$V_IPL> EQ 0 ; IPL must be byte aligned
00CD 711 ASSUME PSL$$_IPC LE 8 ; IPL field must fit into byte
00CD 712
00C6 1F 93 00CD 713 BITB #<PSL$M_IPL@-PSL$V_IPL>,-
C5 00CF 714 <PSL$V_IPL/8>+PHD$[_PSL(R5) ;MUST BE AT IPL 0 FOR DELIVERY
E8 12 00D2 715 BNEQ 10$ ;BRANCH IF AST NOT DELIVERABLE
52 D4 00D4 716 CLRL R2 ;SET NULL PRIORITY INCREMENT
00D6 718 RPTEVT AST ;REPORT AST EVENT
E0 11 00DA 722 BRB 10$ ;GO SCHEDULE NEXT PROCESS
00DC 724
00DC 725 .END

```

SYSWAIT
Symbol table

EVENT FLAG WAIT SERVICES

L 5

16-SEP-1984 02:36:57 VAX/VMS Macro V04-00
5-SEP-1984 03:58:01 [SYS.SRC]SYSWAIT.MAR;1

Page 11
(1)

TI
VO

CEBSB_TYPE	=	0000000A		
CEBSL_EFC	=	00000010		
CEBSL_MASTER	=	00000040		
CEBSL_WQFL	=	00000014		
DYN\$C_SLAVCEB	=	0000002D		
EFN	=	00000004		
EVTS_AST		*****	X	02
EXESC_CMSTKSZ		*****	X	02
EXESGC_ABSTIM		*****	X	02
EXESWAIT		0000001A	R	02
EXESWAITFR		00000011	RG	02
EXESWFLAND		00000000	RG	02
EXESWFLOR		00000007	RG	02
IPLS_SYNCH	=	00000008		
MASK	=	00C00008		
NOWAIT		00000068	R	02
PCBSB_WEFC	=	0000002E		
PCBSL_EFCS	=	00000050		
PCBSL_EFWM	=	0000004C		
PCBSL_PHD	=	0000006C		
PCBSL_STS	=	00000024		
PCBSL_WAITIME	=	00000118		
PCBSV_WALL	=	0000000D		
PCBSW_STATE	=	0000002C		
PHDSB_ASTLVL	=	000000CF		
PHDSL_PSL	=	000000C4		
PHDSL_TIMREF	=	00000100		
PHDSW_QUANT	=	0000003C		
PRS_IPL	=	00000012		
PSLSM_IPL	=	001F0000		
PSLSS_CURMOD	=	00000002		
PSLSS_IPL	=	00000005		
PSLSV_CURMOD	=	00000018		
PSLSV_IPL	=	00000010		
SCHSGO_LEFWQ		*****	X	02
SCHSGW_IOTA		*****	X	02
SCHSRSE		*****	X	02
SCHSSCHED		*****	X	02
SCHSWAIT		00000082	RG	02
SCHSWAITK		0000008C	RG	02
SCHSWAITL		00000097	RG	02
SCHSWAITM		00000098	RG	02
SSS_ILLEFC	=	000000EC		
SSS_NORMAL	=	00000001		
SSS_UNASEFC	=	00000234		
WAIT		00000078	R	02
WAITALL		00000073	R	02
WAITCK		0000005D	R	02
WFRL		0000000B	R	02
WQMSL_WQFL	=	00000000		
WQMSW_WQCNT	=	00000008		
WQMSW_WQSTATE	=	0000000A		

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
AEXENONPAGED	000000DC (220.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.09	00:00:00.27
Command processing	133	00:00:00.66	00:00:02.13
Pass 1	333	00:00:10.15	00:00:12.00
Symbol table sort	0	00:00:01.67	00:00:01.77
Pass 2	79	00:00:02.19	00:00:02.45
Symbol table output	7	00:00:00.09	00:00:00.10
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	594	00:00:14.88	00:00:18.76

The working set limit was 1500 pages.
65723 bytes (129 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1087 non-local and 6 local symbols.
725 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	11
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	18

1192 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

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

