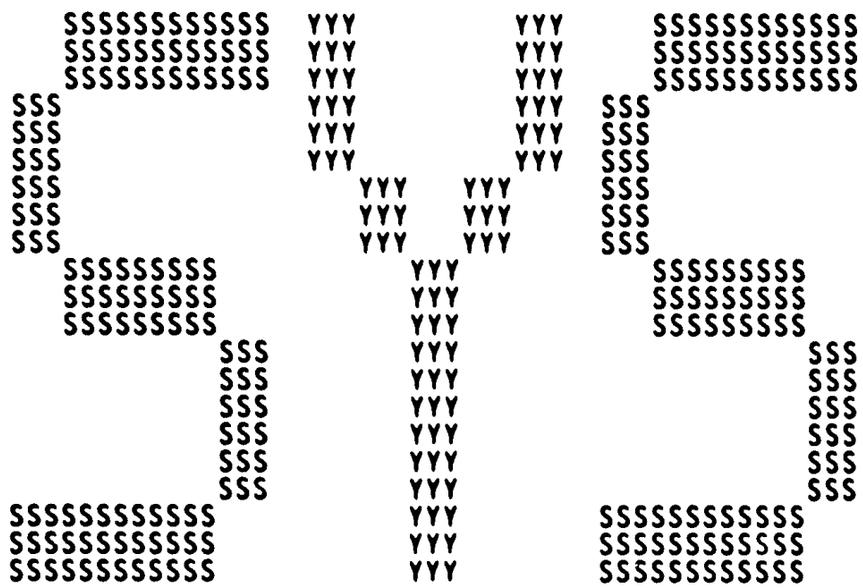

```
RRRRRRRR      SSSSSSSS  EEEEEEEEEE
RRRRRRRR      SSSSSSSS  EEEEEEEEEE
RR      RR  SS          EE
RR      RR  SS          EE
RR      RR  SS          EE
RR      RR  SS          EE
RRRRRRRR      SSSSSS    EEEEEEEE
RRRRRRRR      SSSSSS    EEEEEEEE
RR  RR              SS  EE
RR  RR              SS  EE
RR    RR            SS  EE              ....
RR    RR            SS  EE              ....
RR      RR  SSSSSSSS    EEEEEEEEEE      ....
RR      RR  SSSSSSSS    EEEEEEEEEE      ....


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

Table of contents

```
0000     1          .TITLE  RSE - REPORT SYSTEM EVENT
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE            *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER       *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY         *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE      *
0000    19  ;*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT         *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS      *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27
0000    28  ;++
0000    29  ; FACILITY: EXECUTIVE, SCHEDULER
0000    30  ;
0000    31  ; ABSTRACT:
0000    32  ;       THIS MODULE CONTAINS THE SYSTEM EVENT REPORTING ROUTINES AND
0000    33  ;       THEIR SUPPORTING SUBROUTINES.
0000    34  ;
0000    35  ; ENVIRONMENT:
0000    36  ;       MODE = KERNEL
0000    37  ;--
0000    38  ;
```

```
0000     40          .SBTTL  HISTORY                 ; DETAILED
0000     41 ;
0000     42 ; AUTHOR:        R. HUSTVEDT      CREATION DATE: 6-SEP-76
0000     43 ;
0000     44 ;       V03-008 SSA0015         Stan Amway              8-Mar-1984
0000     45 ;               Allow expansion of working set if PFRATH exceeded and
0000     46 ;               number of active pages (PPG + GPG) exceeds 75% of WSSIZE.
0000     47 ;               Previously, expansion was done only if the number of active
0000     48 ;               pages was equal to WSSIZE.
0000     49 ;               (Acknowledgements go to Wayne Cardoza and Larry Kenah,
0000     50 ;                who both collaborated on this change.)
0000     51 ;
0000     52 ;       V03-007 WMC0002         Wayne Cardoza           28-Feb-1984
0000     53 ;               Fix checks for waking swapper.
0000     54 ;
0000     55 ;       V03-006 LY00B4          Larry Yetto             10-FEB-1984 10:29
0000     56 ;               Fix truncation errors
0000     57 ;
0000     58 ;       V03-005 TMK0002         Todd M. Katz            27-Dec-1983
0000     59 ;               Fix broken branches.
0000     60 ;
0000     61 ;       V03-004 SSA0003         Stan Amway               5-Dec-1983
0000     62 ;               Added support for outswap scheduling changes.
0000     63 ;               Record event time for process unwait and quantum end
0000     64 ;               event.
0000     65 ;
0000     66 ;       V03-003 TMK0001         Todd M. Katz            13-Nov-1983
0000     67 ;               Fix broken branches.
0000     68 ;
0000     69 ;       V03-002 TCM0001         Trudy C. Matthews       4-Apr-1983
0000     70 ;               Change references to working set fields in PHD so that
0000     71 ;               they are used as unsigned words.
0000     72 ;
0000     73 ;       V03-001 WMC0001         Wayne Cardoza           11-Mar-1983
0000     74 ;               Bad comparison against BORROWLIM.
0000     75 ;
0000     76
```

```
                        .SBTTL  DECLARATIONS

                ;
                ; INCLUDE FILES:
                ;
                        $ACBDEF                         ; DEFINE AST CONTROL BLOCK
                        $DYNDEF                         ; DEFINE STRUCTURE TYPE CODES
                        $CEBDEF                         ; DEFINE COMMON EVENT BLOCK
                        $IPLDEF                         ; IPL DEFINITIONS
                        $PCBDEF                         ; PCB DEFINITIONS
                        $PHDDEF                         ; PROCESS HEADER DEFINITIONS
                        $PRDEF                          ; PROCESSOR REGISTER DEFS
                        $PRIDEF                         ; PRIORITY INCREMENT CLASSES
                        $SSDEF                          ; DEFINE STATUS CODES
                        $STATEDEF                       ; STATE DEFINITIONS
                        $WQHDEF                         ; WAIT QUEUE HEADER DEFINITIONS
                ;
                ; MACROS:
                ;
                        .MACRO  EVENT,EVTN,STATLIST,EACTION,CONT=0
                        .IF     NB,EVTN
                EVT$_'EVTN==EVTCTR
                        .ENDC
                EVTCTR=EVTCTR+1
                        .WORD   EACTION-STACT
                RSE...=.
                        .PSECT  AES2,BYTE
                STMSK=CONT
                        .IRP    ST,<STATLIST>
                STMSK=STMSK+<1@SCH$C_'ST>
                        .ENDR
                        .LONG   STMSK
                        .PSECT  AES1,BYTE
                .=RSE...
                        .ENDM   EVENT

                ;
                ;       GENERATE MASK FOR WAIT STATES
                ;
                ;       GMASK   STATENAME
                ;

                        .MACRO  GMASK,STATE
                ST=SCH$C_'STATE
                WAITST=WAITST+<1@ST>
                        .ENDM   GMASK
                ;
                ; EQUATED SYMBOLS:
                ;
00000000        EVTCTR=0                                ; INITIALIZE EVENT COUNTER TO 0
00000000        WAITST=0                                ; INITIALIZE WAIT STATE MASK
00000000        ASTEXIT=0                               ; AST EXIT CHANGE MODE CODE
                ;
                ; OWN STORAGE:
                ;
                        .PSECT  AES2,BYTE               ; STATE EVENT MASK PSECT
00000000        STET=   .                               ; BASE OF STATE EVENT TABLE
```

```
                00000000    135         .PSECT  AES1,BYTE
```

```
                                                .SBTTL  SCH$RSE - REPORT SYSTEM EVENT

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       SCH$RSE RECEIVES SYSTEM EVENT REPORTS FROM VARIOUS SOURCES
                                        ;       AND PERFORMS THE APPROPRIATE ACTION FOR THE SPECIFIED PROCESS.
                                        ;       EVENT REPORTING MUST BE PERFORMED WITH IPL=IPL$_SYNCH.
                                        ;       AS A SIDE EFFECT OF AN EVENT REPORT, THE RESCHEDULING INTERRUPT
                                        ;       MAY BE TRIGGERED IF APPROPRIATE.
                                        ;
                                        ; CALLING SEQUENCE:
                                        ;       BSB/JSB SCH$RSE
                                        ;       .BYTE   EVT$_EVENTNAME
                                        ;
                                        ;       THIS CALLING SEQUENCE IS GENERATED BY THE RPTEVT SYSTEM MACRO
                                        ;
                                        ;       REPEVT  EVENTNAME
                                        ;
                                        ; INPUT PARAMETERS:
                                        ;       R2 - SITUATIONAL PRIORITY INCREMENT CLASS NUMBER
                                        ;       R4 - PCB ADDRESS OF PROCESS FOR WHICH EVENT IS REPORTED
                                        ;
                                        ;       EVENT NUMBER CONTAINED IN BYTE LOCATED BY ADDRESS AT TOP
                                        ;       OF STACK.  @(SP)
                                        ;
                                        ; IMPLICIT INPUTS:
                                        ;       SCHEDULER DATA BASE
                                        ;
                                        ; OUTPUT PARAMETERS:
                                        ;       NONE
                                        ;
                                        ; IMPLICIT OUTPUTS:
                                        ;       NONE
                                        ;
                                        ; COMPLETION CODES:
                                        ;       NONE
                                        ;
                                        ; SIDE EFFECTS:
                                        ;       A RESECHEDULING INTERRUPT MAY BE REQUESTED IF THE SPECIFIED
                                        ;       PROCESS IS HIGHER IN PRIORITY THAN THE CURRENT PROCESS.
                                        ;
                                        ;--

                                SCH$RSE::                               ; REPORT SYSTEM EVENT
    53    00 BE    9A                   MOVZBL  @(SP),R3                ; GET EVENT NUMBER
             6E    D6                   INCL    (SP)                    ; UPDATE RETURN ADDRESS
    51    2C A4    3C                   MOVZWL  PCB$W_STATE(R4),R1      ; GET CURRENT STATE NUMBER
50   0000'CF43    D0           10$:     MOVL    W^STET[R3],R0           ; GET STATE MASK FOR EVENT
    06 50    51   E0                   BBS     R1,R0,ACTION            ; DO ACTION IF STATE BIT SET
             53   D6                   INCL    R3                      ; CHECK NEXT ACTION
          F1 50   E8                   BLBS    R0,10$                  ; IF CONTINUATION
                  05                   RSB                             ; OTHERWISE IGNORE EVENT

  0A'  00   53    CF           ACTION: CASEL   R3,#0,S^#MAXEVT         ; SWITCH ON EVENT NUMBER(UPDATED)
                               STACT:                                  ; BASE OF ACTION TABLE
                                       EVENT   AST,<-                  ; AST EVENT
                                               CEF,-                   ; COMMON EVENT FLAG WAIT
```

```
                     001E   194                 COLPG,-                 ; COLLIDED PAGE WAIT
                     001E   195                 FPG,-                   ; FREE PAGE WAIT
                     001E   196                 HIB,-                   ; RESIDENT HIBERNATE
                     001E   197                 HIBO,-                  ; NON-RESIDENT HIBERNATE
                     001E   198                 LEF,-                   ; LOCAL EVENT FLAG WAIT
                     001E   199                 LEFO,-                  ; LOCAL EVENT FLAG WAIT (NON-RES)
                     001E   200                 MWAIT,-                 ; MUTEX WAIT
                     001E   201                 PFW,-                   ; PAGE FAULT WAIT
                     001E   202                 >,EVENTE                ; AST EXECUTABLE STATE CHANGE
00000000             0020   203 EVT$_COLPGA==EVT$_AST                   ; USE SAME EVENT FOR COLLIDED PAGE AVAIL
                     0020   204
                     0020   205
                     0020   206         EVENT   EVENT,<LEF>,LEFEVT,CONT=1       ; EVENT FLAG SETTING
                     0022   207         EVENT   ,<CEF>,CEFEVT,CONT=1    ; COMMON EVENT FLAG SET
                     0024   208         EVENT   ,<LEFO>,EVENTE          ; NON-RESIDENT LOCAL EVENT
                     0026   209
                     0026   210         EVENT   FPGA,<-                 ; FREE PAGE AVAILABLE
                     0026   211                 FPG,-                   ; FREE PAGE WAIT STATE
                     0026   212                 >,EVENTF                ; EXECUTABLE STATE CHANGE
                     0028   213         EVENT   WAKE,<-                 ; WAKE EVENT
                     0028   214                 HIB,-                   ; RESIDENT HIBERNATE
                     0028   215                 HIBO,-                  ; NON-RESIDENT HIBERNATE
                     0028   216                 >,EVENTE                ;
                     002A   217
                     002A   218
                     002A   219         EVENT   RESUME,<-               ; RESUME EVENT
                     002A   220                 SUSP,-                  ; RESIDENT SUSPENDED
                     002A   221                 SUSPO,-                 ; NON-RESIDENT SUSPENDED
                     002A   222                 >,EVENTE                ; EXECUTABLE
                     002C   223
                     002C   224
                     002C   225         EVENT   PFCOM,<-                ; PAGE FAULT COMPLETE EVENT
                     002C   226                 PFW,-                   ; PAGE FAULT WAIT
                     002C   227                 >,EVENTE                ; EXECUTABLE
                     002E   228
                     002E   229         EVENT   SETPRI,<-               ; SET PRIORITY EVENT
                     002E   230                 COM,-                   ; RESIDENT COMPUTE
                     002E   231                 COMO,-                  ; NON-RESIDENT COMPUTE
                     002E   232                 >,EVENTF                ; EXECUTABLE
                     0030   233
                     0030   234         EVENT   SWPOUT,<-               ; SWAP OUT EVENT
                     0030   235                 HIB,-                   ; RESIDENT HIBERNATE
                     0030   236                 LEF,-                   ; RESIDENT LOCAL EVENT FLAG WAIT
                     0030   237                 SUSP,-                  ; RESIDENT SUSPENDED
                     0030   238                 >,SWPO,CONT=1           ; SIMPLE SWAP OUT
                     0032   239         EVENT   ,<-                     ; SWAP OUT EVENT CONINUATION
                     0032   240                 COM,-                   ; RESIDENT COMPUTE
                     0032   241                 >,SWPOE                 ; EXECUTABLE OUTSWAP
                     0034   242
                     0034   243 ;
                     0034   244 ;       IF CASE FALLS THROUGH, THEN BUGCHECK WITH ILLEGAL EVENT
                     0034   245 ;       NUMBER.
                     0034   246 ;
0000000A             0034   247 MAXEVT=EVTCTR-1                          ; MAXIMUM EVENT NUMBER
                     0034   248         BUG_CHECK ILLEVTNUM,FATAL       ; ILLEGAL EVENT NUMBER
                     0038   249
```

```
                     0038   251 ;
                     0038   252 ;       COMMON EVENT FLAG SET
                     0038   253 ;
                     0038   254 CEFEVT:                                          ;
          0E 24 A4   E9  0038   255          BLBC    PCB$L_STS(R4),EVENTE        ; BR IF NOT RESIDENT
                     003C   256 LEFEVT:                                          ; LOCAL EVENT FLAG SET FOR RESIDENT PROCESS
       53   6C A4   D0  003C   257          MOVL    PCB$L_PHD(R4),R3            ; POINT TO PHD
  00C0 C3    04   C0  0040   258          ADDL    #4,PHD$L_PC(R3)             ; SKIP PC OVER CHMK INSTRUCTION
  0088 C3    01   3C  0045   259          MOVZWL  #SS$_NORMAL,PHD$L_R0(R3)    ; SET NORMAL COMPLETION FOR WAIT
                     004A   260 ;
                     004A   261 ;       EVENT EXECUTABLE ACTION ROUTINE
                     004A   262 ;
                     004A   263
                     004A   264 EVENTE:                                         ; EVENT EXECUTABLE STATE CHANGE
                     004A   265 EVENTF:                                         ; ACTIVATE WITH NO WAIT TIME ACCOUNTING
        88'AF   9F   004A   266          PUSHAB  B^SCH$CHSE                    ; MAKE UNWAIT EXIT THROUGH CHSE
                     004D   267 ;        BRB     SCH$UNWAIT                    ; AND FALL INTO UNWAIT
                     004D   268
```

```
                               004D          .SBTTL  SCH$UNWAIT - DECREMENT COUNT IN WAIT QUEUE
                               004D  ;++
                               004D  ;
                               004D  ; FUNCTIONAL DESCRIPTION:
                               004D  ;       SCH$UNWAIT DECREMENTS THE NUMBER OF PROCESSES IN THE WAIT
                               004D  ;       QUEUE SELECTED BY THE SPECIFIED PCB AND STATE VALUE.
                               004D  ;
                               004D  ; CALLING SEQUENCE:
                               004D  ;       BSB/JSB SCH$UNWAIT
                               004D  ;
                               004D  ; INPUT PARAMETERS:
                               004D  ;       R1 - STATE NUMBER (PRESERVED)
                               004D  ;       R2 - UNUSED       (PRESERVED)
                               004D  ;       R4 - PCB ADDRESS  (PRESERVED)
                               004D  ;
                               004D  ; IMPLICIT INPUTS:
                               004D  ;       PCB LOCATED BY ADDRESS IN R4
                               004D  ;
                               004D  ; IMPLICIT OUTPUTS:
                               004D  ;       COUNT IN WAIT QUEUE HEADER IS DECREMENTED IF STATE IS A WAIT
                               004D  ;       STATE.
                               004D  ;
                               004D  ;--
                               004D
                               004D SCH$UNWAIT::                                 ; DECREMENT PROPER WAIT COUNT
     1A 7B'AF     51     E1    004D         BBC     R1,B^WAITMSK,20$            ; SKIP OUT IF NOT WAIT STATE
            51    03     B1    0052         CMPW    #SCH$C_CEF,R1               ; CHECK FOR COMMON EVENT FLAG WAIT
                  16     13    0055         BEQL    30$                         ; CEF WAIT
            51    0C     C4    0057         MULL    #WQH$C_LENGTH,R1            ; COMPUTE BYTE INDEX TO WQ HDR
 50    00000000'EF41     9E    005A         MOVAB   L^SCH$AQ_WQHDR[R1],R0       ; COMPUTE ADDRESS OF WAIT Q HEADER
               08 A0     B7    0062 10$:    DECW    WQH$W_WQCNT(R0)             ; DECREMENT WAIT QUEUE COUNT
 0118 C4    0000'CF      D0    0065         MOVL    W^EXE$GL_ABSTIM,PCB$L_WAITIME(R4) ; Record event time
                         05    006C 20$:    RSB                                 ; RETURN
                               006D
      50    2E A4        9A    006D 30$:    MOVZBL  PCB$B_WEFC(R4),R0           ; WAIT CLUSTER NUMBER
    50    50 A440        D0    0071         MOVL    PCB$L_EFCS(R4)[R0],R0       ; GET CLUSTER ADDRESS
         50    14        C0    0076         ADDL    #CEB$C_WQFL,R0              ; POINT TO WAIT QUEUE HEADER
               E7        11    0079         BRB     10$                         ; GO DECREMENT WAIT COUNT
                               007B
                               007B         GMASK   CEF                         ; COMMON EVENT FLAG
                               007B         GMASK   LEF                         ; LOCAL EVENT FLAG WAIT
                               007B         GMASK   LEFO                        ; LOCAL EVENT FLAG WAIT
                               007B         GMASK   HIB                         ; HIBERNAT WAIT
                               007B         GMASK   HIBO                        ; HIBERNATE WAIT
                               007B         GMASK   FPG                         ; FREE PAGE WAIT
                               007B         GMASK   COLPG                       ; COLLISION PAGE WAIT
                               007B         GMASK   PFW                         ; PAGE FAULT WAIT
                               007B         GMASK   SUSP                        ; SUSPENDED WAIT
                               007B         GMASK   SUSPO                       ; SUSPENDED WAIT
                               007B         GMASK   MWAIT                       ; MUTEX WAIT
                 00000FFE      007B WAITMSK:.LONG  WAITST                       ; MASK OF WAIT STATES
                               007F
```

```
                     007F    323          .SBTTL  SITUATIONAL PRIORITY INCREMENT TABLE
                     007F    324 ;
                     007F    325 ; FIXED DATA:
                     007F    326 ;                SITUATIONAL PRIORITY INCREMENT TABLE
                     007F    327 ;                    (INDEXED BY PRIORITY INCREMENT CLASS)
                     007F    328 ;
                     007F    329
                     007F    330 B_PINC:                                          ;
          00         007F    331          .BYTE   0                              ; CLASS 0 - NONE
          02         0080    332          .BYTE   2                              ; CLASS 1 - I/O COMPLETE
          03         0081    333          .BYTE   3                              ; CLASS 2 - RESOURCE AVAIL
          04         0082    334          .BYTE   4                              ; CLASS 3 - TERM OUTPUT COMP
          06         0083    335          .BYTE   6                              ; CLASS 4 - TERM INPUT COMP
                     0084    336
                     0084    337
                     0084    338 EXESTATE:                                        ; EXECUTABLE STATE MASK
00003000             0084    339          .LONG   <1@SCH$C_COM>!<1@SCH$C_COMO>
                     0088    340
```

```
                     0088    342          .SBTTL  SCH$CHSE - CHANGE STATE TO EXECUTABLE
                     0088    343 ;++
                     0088    344 ; FUNCTIONAL DESCRIPTION:
                     0088    345 ;       SCH$CHSE CHANGES THE STATE OF A PROCESS, AS REPRESENTED BY
                     0088    346 ;       ITS PCB, TO AN EXECUTABLE STATE.  THE RESCHEDULING INTERRUPT
                     0088    347 ;       WILL BE TRIGGERED IF THE PROCESS IS RESIDENT AND HAS A PRIOITY
                     0088    348 ;       GREATER THAN THAT OF THE CURRENTLY EXECUTING PROCESS.  A
                     0088    349 ;       PRIORITY INCREMENT CLASS NUMBER SUPPLIED AS A REGISTER CONTAINED
                     0088    350 ;       ARGUMENT IS USED TO COMPUTE THE NEW PROCESS PRIORITY FROM ITS
                     0088    351 ;       BASE PRIORITY.
                     0088    352 ;
                     0088    353 ; CALLING SEQUENCE:
                     0088    354 ;       BSB/JSB SCH$CHSE
                     0088    355 ;
                     0088    356 ; INPUT PARAMETERS:
                     0088    357 ;       R0 - NEW PRIORITY       (SCH$CHSEP ONLY)
                     0088    358 ;       R2 - PRIORITY INCREMENT CLASS NUMBER   (SCH$CHSE ONLY)
                     0088    359 ;               0 => NO INCREMENT (PAGEFAULT I/O COMPLETION)
                     0088    360 ;               1 => NON-TERMINAL I/O COMPLETION
                     0088    361 ;               2 => RESOURCE AVAILABILITY
                     0088    362 ;               3 => TERMINAL OUTPUT COMPLETION
                     0088    363 ;               4 => TERMINAL INPUT COMPLETION
                     0088    364 ;       R4 - PCB ADDRESS
                     0088    365 ;
                     0088    366 ; IMPLICIT INPUTS:
                     0088    367 ;       SCH$AQ_COMT - COMPUTE QUEUE HEADERS FOR COM,COMO STATES
                     0088    368 ;       SCH$GB_PRI  - CURRENT PROCESS PRIORITY.
                     0088    369 ;
                     0088    370 ;
                     0088    371 ; OUTPUT PARAMETERS:
                     0088    372 ;       R2 - R2 , PRIORITY INCREMENT CLASS NUMBER IF SCH$CHSE. (PRESERVED)
                     0088    373 ;       R3 - R3 (PRESERVED)
                     0088    374 ;
                     0088    375 ; IMPLICIT OUTPUTS:
                     0088    376 ;       SCH$AQ_COMH  -  VECTOR OF COMPUTE QUEUE HEADERS.
                     0088    377 ;       SCH$GL_COMQS -  COMPUTE QUEUE SUMMARY BIT VECTOR.
                     0088    378 ;
                     0088    379 ; COMPLETION CODES:
                     0088    380 ;       NONE
                     0088    381 ;
                     0088    382 ; SIDE EFFECTS:
                     0088    383 ;       THE PCB SPECIFIED IS REMOVED FROM ITS PRESENT STATE QUEUE
                     0088    384 ;       AND INSERTED IN THE APPROPRIATE COMPUTE QUEUE, COM OR COMO,
                     0088    385 ;       AT THE PRIORITY COMPUTED FOR THE SPECIFIED SITUATION CLASS.
                     0088    386 ;       THE SUMMARY BIT FOR THE DESTINATION STATE QUEUE IS SET TO
                     0088    387 ;       NOTE THAT IT IS OCCUPIED.
                     0088    388 ;       IF THE NEW PRIORITY FOR THE PROCESS IS GREATER THAN THAT OF
                     0088    389 ;       CURRENT PROCESS AND IT IS RESIDENT, THE RESCHEDULING INTERRUPT
                     0088    390 ;       WILL BE TRIGGERED.
                     0088    391 ;
                     0088    392 ;--
                     0088    393 SCH$CHSE::                                  ;CHANGE TO EXECUTABLE STATE
                 50  D4  0088    394          CLRL    R0                     ; CLEAR HIGH SUM BITS FOR ADDB
50  2F A4  F1 AF 42  83  008A    395          SUBB3   B_PINC[R2],PCB$B_PRIB(R4),R0    ; ADD PRIORITY INCR
        0B A4   50  91  0091    396          CMPB    R0,PCB$B_PRI(R4)       ; CHECK FOR > CURRENT PRI
                04  15  0095    397          BLEQ    10$                    ; NO
            50  0B A4  90  0097    398          MOVB    PCB$B_PRI(R4),R0       ; KEEP CURRENT PRIORITY INSTEAD
```

```
                10   50   91   009B   399 10$:    CMPB    R0,#16                  ; CHECK FOR RESULT >15
                     04   18   009E   400         BGEQ    SCH$CHSEP               ; YES, USE COMPUTED VALUE
              50   2F A4   90   00A0   401         MOVB    PCB$B_PRIB(R4),R0       ; KEEP AT BASE IF LESS
                                00A4   402
                                00A4   403 ;
                                00A4   404 ;       SCH$CHSEP - SUB-ENTRY POINT WITH PRIORITY PRECOMPUTED IN R0
                                00A4   405 ;
                                00A4   406
                                00A4   407 SCH$CHSEP::                          ; ENTRY WITH PRIO IN R0
                51   12   DB   00A4   408         MFPR    #PR$_IPL,R1             ; GET IPL
                08   51   D1   00A7   409         CMPL    R1,#IPL$_SYNCH          ; MUST BE AT SYNCH OR GREATER
                     62   19   00AA   410         BLSS    BADIPL                  ; NO, FATAL ERROR
                51   64   0F   00AC   411         REMQUE  (R4),R1                 ; REMOVE FROM CURREN QUEUE
                     1C   12   00AF   412         BNEQ    10$                     ; CONTINUE IF QUEUE NOTEMPTY
             51   2C A4   3C   00B1   413         MOVZWL  PCB$W_STATE(R4),R1      ; GET OLD STATE
          13 CB AF   51   E1   00B5   414         BBC     R1,EXESTATE,10$         ; NO SUMMARY BITS
             51   0B A4   9A   00BA   415         MOVZBL  PCB$B_PRI(R4),R1        ; GET CURRENT PRI
               03 2C A4   E9   00BE   416         BLBC    PCB$W_STATE(R4),5$      ; SKIP IF RESIDENT
                51   20   C0   00C2   417         ADDL    #32,R1                  ; MAKE NONRES PRIO
    00 00000000'EF   51   E5   00C5   418 5$:     BBCC    R1,L^SCH$GL_COMQS,10$   ; CLEAR PRESENCE BIT FOR STATE
             0B A4   50   90   00CD   419 10$:    MOVB    R0,PCB$B_PRI(R4)        ; SAVE NEW PRIO
                51   0C   D0   00D1   420         MOVL    #SCH$C_COM,R1           ; ASSUME COM STATE
              12 24 A4   E8   00D4   421         BLBS    PCB$L_STS(R4),20$       ; CHECK FOR RESIDENCE
                51   D6   00D8   422         INCL    R1                      ; COMQ=COM+1
                50   20   C0   00DA   423         ADDL2   #32,R0                  ; COMQ HEADERS FOLLOW COM
    00 00000000'EF   50   E2   00DD   424         BBSS    R0,L^SCH$GL_COMQS,15$   ; SET SUMMARY BIT FOR NEW QUEUE
                01CD   30   00E5   425 15$:     BSBW    SCH$SWPWAKE             ; WAKE SWAPPER
                14   11   00E8   426         BRB     35$                     ; COMPLETE STATE CHANGE
       50   00000000'EF   91   00EA   427 20$:     CMPB    L^SCH$GB_PRI,R0         ; IS PRIO GREATER THAN CURRENT PROCESS
                03   19   00F1   428         BLSS    30$                     ; NO, DONT RESCHEDULE
                          00F3   429         SOFTINT #IPL$_SCHED             ; TRIGGER RESCHEDULE INTERRUPT
    00 00000000'EF   50   E2   00F6   430 30$:     BBSS    R0,L^SCH$GL_COMQS,35$   ; SET SUMMARY BIT FOR NEW QUEUE
             2C A4   51   B0   00FE   431 35$:     MOVW    R1,PCB$W_STATE(R4)      ; SET NEW STATE
     51   00000000'EF40   7E   0102   432         MOVAQ   L^SCH$AQ_COMT[R0],R1    ; COMPUTE HDR ADDR
                91   64   0E   010A   433         INSQUE  (R4),@(R1)+             ; IN  RT IN NEW QUEUE
                     05   010D   434         RSB                             ; RETURN
                          010E   435
                          010E   436 BADIPL: BUG_CHECK BADRSEIPL,FATAL       ; BAD IPL AT ENTRANCE TO RSE
                          0112   437
```

```
                     0112   439          .SBTTL  SWPO - SWAP OUT SIMPLE NON-EXECUTABLE
                     0112   440 ;
                     0112   441 ;       SWPO - SWAP OUT ACTION ROUTINE FOR SIMPLE NON-EXECUTABLE STATES
                     0112   442 ;
                     0112   443 SWPO:                                        ; NON-EXECUTABLE OUTSWAP
        FF38   30    0112   444          BSBW    SCH$UNWAIT                 ; REMOVE FROM WAIT QUEUE
      2C A4    B6    0115   445          INCW    PCB$W_STATE(R4)            ; UPDATE STATE NUMBER
   51   64    0F    0118   446          REMQUE  (R4),R1                    ; REMOVE FROM WAIT QUEUE
10 B0   64    0E    011B   447          INSQUE  (R4),@WQH$L_WQBL+WQH$C_LENGTH(R0)       ; INSERT AT TAIL OF QUEUE
     14 A0    B6    011F   448          INCW    WQH$W_WQCNT+WQH$C_LENGTH(R0)    ; NOTE COUNT IN WAIT QUEUE
              05    0122   449          RSB                                ; EXIT
                     0123   450
                     0123   451 ;
                     0123   452 ;       SWPOE - SWAP OUT EXECUTABLE ACTION ROUTINE
                     0123   453 ;
50   0B A4    9A    0123   454 SWPOE:  MOVZBL  PCB$B_PRI(R4),R0            ; GET PRIORITY
       FF7A   31    0127   455          BRW     SCH$CHSEP                  ; AND CHANGE TO COMO
```

```
                     012A   457          .SBTTL  SCH$QEND - QUANTUM END ROUTINE
                     012A   458
                     012A   459 ;++
                     012A   460 ;
                     012A   461 ; FUNCTIONAL DESCRIPTION:
                     012A   462 ;       SCH$QEND IS CALLED BY THE TIMER WHEN THE QUANTUM FOR THE CURRENT
                     012A   463 ;       PROCESS HAS BEEN EXHAUSTED.  A NEW QUANTUM IS INITIALIZED
                     012A   464 ;       THE PROCESS PLACED AT ITS BASE PRIORITY AND THE RESCHEDULING
                     012A   465 ;       INTERRUPT TRIGGERED.  A CHECK IS MADE FOR CPU TIME LIMIT EXPIRATION
                     012A   466 ;       AND APPROPRIATE EXIT ASTS GENERATED WHEN THE LIMIT IS REACHED.
                     012A   467 ;       THE AUTOMATIC WORKING SET SIZE LOGIC IS INVOKED IF ENABLED TO
                     012A   468 ;       TRADEOFF WORKING SET SIZE AGAINST PAGEFAULT RATE.
                     012A   469 ;
                     012A   470 ; CALLING SEQUENCE:
                     012A   471 ;       BSB/JSB SCH$QEND
                     012A   472 ;
                     012A   473 ; INPUT PARAMETERS:
                     012A   474 ;       R4 - PCB ADDRESS OF CURRENT PROCESS
                     012A   475 ;       R5 - PROCESS HEADER ADDRESS
                     012A   476 ;
                     012A   477 ; IMPLICIT INPUTS:
                     012A   478 ;       PCB OF CURRENT PROCESS
                     012A   479 ;       PROCESS HEADER OF CURRENT PROCESS
                     012A   480 ;
                     012A   481 ; IMPLICIT OUTPUTS:
                     012A   482 ;       PHD$W_QUANT - INITIALIZED TO A NEW QUANTUM
                     012A   483 ;       PCB$V_INQUAN - INITIAL QUANTUM FLAG CLEARED
                     012A   484 ;
                     012A   485 ;--
                     012A   486
                     012A   487 SCH$QEND::                              ; QUANTUM END ROUTINE
          00 24 A4    03   E5  012A   488          BBCC    #PCB$V_INQUAN,PCB$L_STS(R4),10$ ; CLEAR INITIAL QUAN FLAG
    3C A5   00000000'EF   B0  012F   489 10$:     MOVW    SCH$GW_QUAN,PHD$W_QUANT(R5)    ; SET NEW QUANTUM
      0118 C4    0000'CF   D0  0137   490          MOVL    W^EXE$GL_ABSTIM,PCB$L_WAITIME(R4) ; Record event time
             10    0B A4   91  013E   491          CMPB    PCB$B_PRI(R4),#16               ; CHECK FOR REAL-TIME
                   26   19  0142   492          BLSS    50$                             ; YES
                        0144   493 ;
                        0144   494 ;
                        0144   495 ;       CHECK FOR CPU TIME LIMIT EXPIRATION
                        0144   496 ;
                 5C A5   D5  0144   497          TSTL    PHD$L_CPULIM(R5)                ; IS THERE ANY LIMIT?
                   22   12  0147   498          BNEQ    60$                             ; YES, GO CHECK IT OUT
          09 24 A4    18   E0  0149   499 40$:     BBS     #PCB$V_DISAWS,PCB$L_STS(R4),45$ ; BRANCH IF ADJUSTMENT DISABLED
           53   0000'CF   D0  014E   500          MOVL    W^SCH$GL_WSINC,R3               ; ASSUME INCREMENT
                   02   13  0153   501          BEQL    45$                             ; BR IF NO AUTO WS ADJUSTMENT
                   49   10  0155   502          BSBB    WSADJUST                        ; ELSE GO DO IT
        00000000'EF   D5  0157   503 45$:     TSTL    L^SCH$GL_COMQS                  ; IS THERE ANY INSWAP PENDING?
                   08   13  015D   504          BEQL    47$                             ; NO
          0B A4    2F A4   90  015F   505          MOVB    PCB$B_PRIB(R4),PCB$B_PRI(R4)    ; YES, FORCE TO BASE PRIORITY
                  014E   30  0164   506          BSBW    SCH$SQPWAKE                     ; AND WAKE SWAPPER
                         0167   507 47$:     SOFTINT #IPL$_SCHED                     ; TRIGGER RESCHEDULING INT
                      05  016A   508 50$:     RSB                                     ; AND RETURN
                         016B   509
                         016B   510 ;
                         016B   511 ;       A non-zero limit exists, check for processor time expiration
                         016B   512 ;
                         016B   513 ;       If CPU time limit is exceeded then an additional amount of time will
```

```
                                016B   514 ;       be allowed for each access mode.  An AST will be issued to cause an
                                016B   515 ;       exit for each of the access modes.  The additional time allowance will
                                016B   516 ;       be provided for each access mode.
                                016B   517 ;
                                016B   518
     50   38 A5   5C A5   C3    016B   519 60$:    SUBL3   PHD$L_CPULIM(R5),PHD$L_CPUTIM(R5),R0 ; HAS LIMIT BEEN REACHED
                      D6   1F   0171   520         BLSSU   40$                     ; NO, CONTINUE NORMALLY
                                0173   521 ;
                                0173   522 ;       CPU LIMIT HAS EXPIRED, AN AST WILL BE SENT TO NOTIFY THE PROCESS
                                0173   523 ;
          50   0000'CF   C0    0173   524         ADDL2   W^SGN$GL_EXTRACPU,R0    ; COMPUTE TOTAL AMOUNT OF EXTRA TIME
            5C A5    50   C0    0178   525         ADDL2   R0,PHD$L_CPULIM(R5)     ; GIVE EXTRA TIME FOR CLEANUP
          0110 C5    50   C0    017C   526         ADDL2   R0,PHD$L_EXTRACPU(R5)   ;  AND RECORD AMOUNT OF EXTRA TIME
            50    60 A5   9E    0181   527         MOVAB   PHD$B_CPUMODE(R5),R0    ; GET ADDRESS OF AST ACCESS MODE
                  C1 AF   9F    0185   528         PUSHAB  40$                     ; SET RETURN ADDRESS
          53   20AC 8F   3C    0188   529         MOVZWL  #SS$_EXCPUTIM,R3        ; PASS EXIT STATUS TO SENDAST
                                018D   530 SCH$FORCEDEXIT::
                     00BD   30  018D   531         BSBW    SENDAST                 ; SEND AST TO PROCESS
                                0190   532 ;
                                0190   533 ;       CPU TIME EXPIRATION AST HANDLER
                                0190   534 ;
                          0000  0190   535 CPUABRT:.WORD   0                       ; NULL ENTRY MASK
                       00   BC  0192   536         CHMK    S^#ASTEXIT              ; EXIT FROM AST ROUTINE (CLEAR AST)
                                0194   537 10$:    $EXIT_S 4(AP)                   ; EXIT TO INVOKE EXIT HANDLERS
                       F4   11  019E   538         BRB     10$                     ; JUST IN CASE
                                01A0   539
                                01A0   540 ;
                                01A0   541 ;       Adjust working set size automatically to achieve desired tradeoff
                                01A0   542 ;       between page fault rate and working set size.  There are two page
                                01A0   543 ;       fault rate thresholds: SCH$GL_PFRATL, the lower threshold and
                                01A0   544 ;       SCH$GL_PFRATH, the higher threshold.  Each time SCH$QEND is invoked,
                                01A0   545 ;       the page fault rate is computed and compared with these thresholds.
                                01A0   546 ;       If it is above the high threshold the working set size is increased
                                01A0   547 ;       by SCH$GW_WSINC and if the rate is below the lower threshold, the
                                01A0   548 ;       working set size is decreased by SCH$GW_WSDEC.  The actual adjustment
                                01A0   549 ;       is performed by a normal kernel mode AST.
                                01A0   550 ;
                                01A0   551 ;       Automatic adjustment of working set size is constrained by the values:
                                01A0   552 ;       SCH$GW_AWSMIN and WSEXTENT per process that establish upper and lower
                                01A0   553 ;       values for automatic working set size adjustment.  Working set size
                                01A0   554 ;       adjustment is further constrained by the process quota.
                                01A0   555 ;
                                01A0   556 ;
                                01A0   557 ;       R3 - Working set increment
                                01A0   558 ;
                                01A0   559 WSADJUST:                            ; AUTO-ADJUST WORKING SET SIZE
          0100 C5    C3   C3    01A0   560         SUBL3   PHD$L_TIMREF(R5),-      ; COMPUTE DELTA-T
            50   38 A5           01A4   561                 PHD$L_CPUTIM(R5),R0  ;
                      02   12    01A7   562         BNEQ    10$                     ; BR IF NON-ZERO
                      50   D6    01A9   563         INCL    R0                      ; ELSE FORCE TO ONE FOR DIVIDE
                                01AB   564 10$:
          0000'CF     50   D1    01AB   565         CMPL    R0,W^SCH$GL_AWSTIME     ; IS THIS A MEANINGFUL INTERVAL?
                      3F   19    01B0   566         BLSS    NOADJUST                ; NO, TRY AGAIN LATER
          00FC C5    C3   C3    01B2   567         SUBL3   PHD$L_PFLREF(R5),-      ; COMPUTE DELTA-PGFLT
            51   4C A5           01B6   568                 PHD$L_PAGEFLTS(R5),R1 ;
       00FC C5   4C A5    D0    01B9   569         MOVL    PHD$L_PAGEFLTS(R5),PHD$L_PFLREF(R5) ; SAVE NEW PAGE FAULT REF
       0100 C5   38 A5    D0    01BF   570         MOVL    PHD$L_CPUTIM(R5),PHD$L_TIMREF(R5)   ; AND SAVE CPUTIME REF
```

```
                51    000003E8 8F    C4   01C5    571          MULL    #1000,R1                    ; MULTIPLY BY SCALE FACTOR
                      51    50 C6    C6   01CC    572          DIVL    R0,R1                       ; AND COMPUTE PAGEFLTS/10SEC
                   00F8 C5    51    D0   01CF    573          MOVL    R1,PHD$L_PFLTRATE(R5)       ; SAVE CURRENT RATE
                   0000'CF    51    D1   01D4    574          CMPL    R1,W^SCH$GL_PFRATH          ; ARE WE ABOVE HIGH THRESHOLD?
                              17    18   01D9    575          BGEQ    ADJUSTUP                    ; YES,
                   53   0000'CF    CE   01DB    576          MNEGL   W^SCH$GL_WSDEC,R3           ; NO, GET DECREMENT VALUE
                   0000'CF    51    D1   01E0    577          CMPL    R1,W^SCH$GL_PFRATL          ; ARE WE BELOW LOW THRESHOLD?
                              0A    18   01E5    578          BGEQ    NOADJUST                    ; NO, IN DEAD BAND -- NOTHING TO DO
                0000'CF    36 A4    B1   01E7    579          CMPW    PCB$W_PPGCNT(R4),W^SCH$GW_AWSMIN ; ARE WE AT LOWER WS LIMIT?
                              02    1B   01ED    580          BLEQU   NOADJUST                    ; YES, NOTHING TO DO
                              39    11   01EF    581          BRB     ADJUST                      ;
                                    01F1    582 NOADJUST:                                       ;
                              05    01F1    583          RSB                                     ;
                                    01F2    584 ADJUSTUP:                                       ;
            51    18 A5    08 A5    A3   01F2    585          SUBW3   PHD$W_WSLIST(R5),PHD$W_WSQUOTA(R5),R1
                                    01F8    586                                                  ; ASSUME HIGH LIMIT WILL BE QUOTA
                      50    50 A5    3C   01F8    587          MOVZWL  PHD$W_WSSIZE(R5),R0         ; GET CURRENT WORKING SET SIZE
         00000000'EF    0000'CF    D1   01FC    588          CMPL    W^SCH$GL_BORROWLIM,L^SCH$GL_FREECNT ; ARE THERE LOTS OF FREE PAGES?
                              06    1A   0205    589          BGTRU   10$                         ; BRANCH IF MEMORY IS AT A PREMIUM
            51    16 A5    08 A5    A3   0207    590          SUBW3   PHD$W_WSLIST(R5),PHD$W_WSEXTENT(R5),R1
                                    020D    591                                                  ; ALLOW LARGER GROWTH SIZE
                         51    50 B1    020D    592 10$:     CMPW    R0,R1                       ; ARE WE AT MAXIMUM SIZE?
                              DF    1A   0210    593          BGTRU   NOADJUST                    ; YES, CAN'T GO ANY LARGER
            51    36 A4    34 A4    A1   0212    594          ADDW3   PCB$W_GPGCNT(R4),PCB$W_PPGCNT(R4),R1    ; GET CURRENT PHYSICAL SIZE
                         51    50 B1    0218    595          CMPW    R0,R1                       ; Be sure that pages in use don't exceed WS
                              2C    1F   021B    596          BLSSU   WSERR                       ; BRANCH IF WS SMALLER THAN PAGES IN USE
               52    50    FE 8F    78   021D    597          ASHL    #-2,R0,R2                   ; Compute 75% of WSSIZE as page threshold
                      50    52 A2    0222    598          SUBW2   R2,R0
                      50    51 B1    0225    599          CMPW    R1,R0                       ; If threshold not exceeded,
                              C7    1F   0228    600          BLSSU   NOADJUST                    ; skip WS adjustment
                   50    61 A5    9E   022A    601 ADJUST:  MOVAB   PHD$B_AWSMODE(R5),R0        ; GET ADDRESS OF AST ACCESS MODE
                              1D    10   022E    602          BSBB    SENDAST                     ; SEND AST TO PROCESS
                                    0230    603
                         0000       0230    604 ADJWS:   .WORD   0                           ;
         51    00000000'9F    D0   0232    605          MOVL    @#CTL$GL_PHD,R1             ; GET PHD ADDRESS SO
                  61 A1    94   0239    606          CLRB    PHD$B_AWSMODE(R1)           ;  ACCESS MODE FLAG CAN BE RESET
                                    023C    607          $ADJWSL_S       4(AP)               ; ADJUST BY PARAMETER IN AST ARGLIST
                              04    0248    608          RET                                 ; AND RETURN
                                    0249    609
                                    0249    610 WSERR:   BUG_CHECK WSSIZEERR,FATAL           ; WORKING SET SIZE CALC IN ERROR
```

```
                                024D   612          .SBTTL  SENDAST - Send AST to process
                                024D   613 ;++
                                024D   614 ; FUNCTIONAL DESCRIPTION: SENDAST IS CALLED BY SCH$QEND TO SEND ASTS TO THE
                                024D   615 ;       PROCESS THAT INVOKE FUNCTIONS UNAVAILABLE TO THE ENVIRONMENT OF SCH$QEND.
                                024D   616 ;       THESE INCLUDE ADJUSTING THE WORKING SET AND EXITTING.
                                024D   617 ;
                                024D   618 ; INPUT PARAMETERS:
                                024D   619 ;
                                024D   620 ;       R0      - ADDRESS OF ACCESS MODE FOR AST
                                024D   621 ;                 (NEGATIVE CONTENTS PREVENT SENDING AST)
                                024D   622 ;       R3      - AST PARAMETER
                                024D   623 ;       R4      - PCB ADDRESS
                                024D   624 ;       (SP)    - AST ADDRESS
                                024D   625 ;       4(SP)   - RETURN ADDRESS FOR THIS SUBROUTINE
                                024D   626 ;--
                                024D   627 SENDAST:                                     ;
                     50   DD   024D   628          PUSHL   R0                         ; SAVE ADDRESS OF ACCESS MODE
                     53   DD   024F   629          PUSHL   R3                         ; AND AST PARAMETER
                     60   95   0251   630          TSTB    (R0)                       ; CHECK VALUE OF ACCESS MODE
                     37   19   0253   631          BLSS    10$                        ; DO NOT QUEUE AST IF NEGATIVE
32 24 A4             01   E0   0255   632          BBS     #PCB$V_DELPEN,PCB$L_STS(R4),10$ ;  NOR IF MARKED FOR DELETE
                     51   1C   025A   633          MOVZWL  #ACB$C_LENGTH,R1           ; SET SIZE REQUIRED
                  FDA0'   30   025D   634          BSBW    EXE$ALONONPAGED            ; ALLOCATE A BLOCK
                     29   50   E9   0260   635          BLBC    R0,10$                     ; NONE, TRY LATER
                  0A A2   02   90   0263   636          MOVB    #DYN$C_ACB,ACB$B_TYPE(R2); SET TYPE OF STRUCTURE
                  08 A2   51   B0   0267   637          MOVW    R1,ACB$W_SIZE(R2)          ; AND SIZE OF STRUCTURE
                  14 A2   8E   D0   026B   638          MOVL    (SP)+,ACB$L_ASTPRM(R2)     ; AND AST PARAMETER VALUE
               0B A2   00   BE   90   026F   639          MOVB    @(SP),ACB$B_RMOD(R2)       ; SET ACCESS MODE FOR AST
                        9E   97   0274   640          DECB    @(SP)+                     ; INDICATE SUCCESS FOR THIS ACCESS MODE
                  10 A2   8E   D0   0276   641          MOVL    (SP)+,ACB$L_AST(R2)        ; SET AST ADDRESS
               0C A2   60 A4   D0   027A   642          MOVL    PCB$L_PID(R4),ACB$L_PID(R2)     ; SET PID FOR AST
                        30   BB   027F   643          PUSHR   #^M<R4,R5>                 ; SAVE REGS FOR QAST
                     55   52   D0   0281   644          MOVL    R2,R5                      ; SET ADDRESS OF ACB
                        52   D4   0284   645          CLRL    R2                         ; NULL PRIORITY INCREMENT
                  FD77'   30   0286   646          BSBW    SCH$QAST                   ; QUEUE AST FOR PROCESS
                        30   BA   0289   647          POPR    #^M<R4,R5>                 ; RESTORE PCB,PHD ADDRESSES
                             05   028B   648          RSB                                ; EXIT
                                  028C   649
                                  028C   650 ;  Error path if nonpaged pool allocation fails or if AST access mode is
                                  028C   651 ;  negative, indicating either an AST in progress (for automatic working
                                  028C   652 ;  set adjustment) or all access modes are done (for CPU time limit expiration)
                                  028C   653
                     5E   0C   C0   028C   654 10$:       ADDL    #12,SP                     ; CLEAN PARAMETERS FROM STACK
                             05   028F   655          RSB                                ; AND EXIT
                                  0290   656
```

```
                     0290   658
                     0290   659          .SBTTL  SCH$WAKE - WAKE PROCESS INTERNAL
                     0290   660 ;++
                     0290   661 ; FUNCTIONAL DESCRIPTION:
                     0290   662 ;       SCH$WAKE WAKES THE PROCESS SPECIFIED BY THE PID SUPPLIED.
                     0290   663 ;
                     0290   664 ; CALLING SEQUENCE:
                     0290   665 ;       BSB/JSB SCH$WAKE
                     0290   666 ;
                     0290   667 ; INPUT PARAMETERS:
                     0290   668 ;       R1 - PID OF PROCESS TO WAKE
                     0290   669 ;
                     0290   670 ; OUTPUT PARAMETERS:
                     0290   671 ;       R0 - COMPLETION STATUS CODE
                     0290   672 ;       R4 - PCB ADDRESS OF PROCESS AWAKENED
                     0290   673 ;
                     0290   674 ; COMPLETION CODES:
                     0290   675 ;       SS$_NORMAL  -  NORMAL SUCCESSFUL COMPLETION STATUS
                     0290   676 ;       SS$_NONEXPR -  NONEXISTENT PROCESS (INVALID PID)
                     0290   677 ;
                     0290   678 ; ENVIRONMENT:
                     0290   679 ;       IPL = IPL$_SYNCH
                     0290   680 ;
                     0290   681 ;--
                     0290   682 SCH$WAKE::                                ; WAKE PROCESS INTERNAL
        54    51   3C 0290   683          MOVZWL  R1,R4                   ; GET PROCESS INDEX (PIX)
54   0000'DF44   D0 0293   684          MOVL    @W^SCH$GL_PCBVEC[R4],R4 ; LOOK UP PCB ADDRESS
    60 A4    51   D1 0299   685          CMPL    R1,PCB$L_PID(R4)        ; VERIFY PID
             10   12 029D   686          BNEQ    30$                     ; REPORT ERROR
 00 24 A4    0C   E2 029F   687          BBSS    #PCB$V_WAKEPEN,PCB$L_STS(R4),10$   ; SET WAKE PENDING
                  02A4   688 10$:                                       ;
        52   02   9A 02A4   689          MOVZBL  #PRI$_RESAVL,R2         ; SET PRIORITY INCREMENT CLASS
                  02A7   690          RPTEVT  WAKE                    ; REPORT WAKE EVENT
        50   01   3C 02AB   691          MOVZWL  #SS$_NORMAL,R0          ; SET SUCCESS CODE
             05   02AE   692 20$:       RSB                             ; RETURN
                  02AF   693
 50  08E8 8F   3C 02AF   694 30$:       MOVZWL  #SS$_NONEXPR,R0         ; SET NONEXISTENT PROCESS STATUS
             05   02B4   695          RSB                             ;
```

```
                        02B5    697          .SBTTL  SCH$SWPWAKE - WAKE SWAPPER PROCESS
                        02B5    698 ;++
                        02B5    699 ; FUNCTIONAL DESCRIPTION:
                        02B5    700 ;       SCH$SWPWAKE AWAKENS THE SWAPPER PROCESS TO PERFORM SOME OPERATION.
                        02B5    701 ;
                        02B5    702 ; CALLING SEQUENCE:
                        02B5    703 ;       BSB/JSB SCH$SWPWAKE
                        02B5    704 ;
                        02B5    705 ; INPUT PARAMETERS:
                        02B5    706 ;       NONE
                        02B5    707 ;
                        02B5    708 ; OUTPUT PARAMETERS:
                        02B5    709 ;       R0-R4 PRESERVED
                        02B5    710 ;
                        02B5    711 ; SIDE EFFECTS:
                        02B5    712 ;       A WAKE EVENT IS GENERATED FOR THE SWAPPER PROCESS WHICH CAN
                        02B5    713 ;       CAUSE THE PROCESSOR TO BE RESCHEDULED.
                        02B5    714 ;
                        02B5    715 ;--
                        02B5    716
                        02B5    717 SCH$SWPWAKE::                               ; WAKE SWAPPER PROCESS
            0C000000'EF 05  02B5    718          TSTL    L^SCH$GL_COMQQS        ; ANY INSWAP CANDIDATES?
                     28 12  02B8    719          BNEQ    10$                    ; YES, MUST WAKE SWAPPER THEN
00000000'EF  C0000000'EF D1 02BD    720          CMPL    L^SCH$GL_MFYCNT,L^SCH$GL_MFYLIM ; ARE THERE MODIFIED PAGES TO WRITE?
                     1B 18  02C8    721          BGEQ    10$                    ; YES, MUST WAKE SWAPPER THEN
00000000'EF  00000000'EF D1 02CA    722          CMPL    L^SCH$GL_FREECNT,L^SCH$GL_FREELIM ; DO WE NEED FREE PAGES?
                     0E 19  02D5    723          BLSS    10$                    ; YES, MUST WAKE SWAPPER THEN
            00000000'EF B5  02D7    724          TSTW    L^SCH$GW_DELPHDCT      ; ARE THERE DELETED HEADERS TO PURGE?
                     06 12  02DD    725          BNEQ    10$                    ; YES, MUST WAKE SWAPPER THEN
                0000'CF D5  02DF    726          TSTL    W^EXE$GL_PFATIM        ; WAS THERE A POWER FAIL RECOVERY?
                     13 13  02E3    727          BEQL    20$                    ; BR IF NONE
             00000000'EF 95  02E5    728 10$:     TSTB    L^SCH$GB_SIP           ; SWAPPER ALREADY BUSY?
                     0B 12  02EB    729          BNEQ    20$                    ; BR IF YES
                     1F BB  02ED    730          PUSHR   #^M<R0,R1,R2,R3,R4>    ; SAVE R0-R4
51              0000'CF D0  02EF    731          MOVL    W^SCH$GL_SWPPID,R1     ; GET PID OF SWAPPER
                     9A 10  02F4    732          BSBB    SCH$WAKE               ; AND AWAKEN IT
                     1F BA  02F6    733          POPR    #^M<R0,R1,R2,R3,R4>    ; RESTORE R0-R4
                        05  02F8    734 20$:     RSB                            ; AND RETURN TO CALLER
                            02F9    735
                            02F9    736          .END
```

```
ACB$B_RMOD              =  0000000B
ACB$B_TYPE              =  0000000A
ACB$C_LENGTH            =  0000001C
ACB$L_AST               =  00000010
ACB$L_ASTPRM            =  00000014
ACB$L_PID               =  0000000C
ACB$W_SIZE              =  00000008
ACTION                     0000001A R       03
ADJUST                     0000022A R       03
ADJUSTUP                   000001F2 R       03
ADJWS                      00000230 R       03
ASTEXIT                 =  00000000
BADIPL                     0000010E R       03
BUG$_BADRSEIPL             ********   X     03
BUG$_ILLEVTNUM             ********   X     03
BUG$_WSSIZEERR             ********   X     03
B_PINC                     0000007F R       03
CEB$L_WQFL              =  00000014
CEFEVT                     00000038 R       03
CPUABRT                    00000190 R       03
CTL$GL_PHD                 ********   X     03
DYN$C_ACB               =  00000002
EVENTE                     0000004A R       03
EVENTF                     0000004A R       03
EVT$_AST                =  00000000  G
EVT$_COLPGA             =  00000000  G
EVT$_EVENT              =  00000001  G
EVT$_FPGA               =  00000004  G
EVT$_PFCOM              =  00000007  G
EVT$_RESUME             =  00000006  G
EVT$_SETPRI             =  00000008  G
EVT$_SWPOUT             =  00000009  G
EVT$_WAKE               =  00000005  G
EVTCTR                  =  0000000B
EXE$ALONONPAGED            ********   X     03
EXE$GL_ABSTIM              ********   X     03
EXE$GL_PFATIM              ********   X     03
EXESTATE                   00000084 R       03
IPL$_SCHED              =  00000003
IPL$_SYNCH              =  00000008
LEFEVT                     0000003C R       03
MAXEVT                  =  0000000A
NOADJUST                   000001F1 R       03
PCB$B_PRI               =  0000000B
PCB$B_PRIB              =  0000002F
PCB$B_WEFC              =  0000002E
PCB$L_EFCS              =  00000050
PCB$L_PHD               =  0000006C
PCB$L_PID               =  00000060
PCB$L_STS               =  0000C024
PCB$L_WAITIME           =  00000118
PCB$V_DELPEN            =  00000001
PCB$V_DISAWS            =  00000018
PCB$V_INQUAN            =  00000003
PCB$V_WAKEPEN           =  0000000C
PCB$W_GPGCNT            =  00000034
PCB$W_PPGCNT            =  00000036
PCB$W_STATE             =  0000002C
PHD$B_AWSMODE           =  00000061
PHD$B_CPUMODE           =  00000060
PHD$L_CPULIM            =  0000005C
PHD$L_CPUTIM            =  00000038
PHD$L_EXTRACPU          =  00000110
PHD$L_PAGEFLTS          =  0000004C
PHD$L_PC                =  000000C0
PHD$L_PFLREF            =  000000FC
PHD$L_PFLTRATE          =  000000F8
PHD$L_RO                =  00000088
PHD$L_TIMREF            =  00000100
PHD$W_QUANT             =  0000003C
PHD$W_WSEXTENT          =  00000016
PHD$W_WSLIST            =  00000008
PHD$W_WSQUOTA           =  00000018
PHD$W_WSSIZE            =  00000050
PR$_IPL                 =  00000012
PR$_SIRR                =  00000014
PRI$_RESAVL             =  00000002
RSE...                  =  00000034 R       03
SCH$AQ_COMT                ********   X     03
SCH$AQ_WQHDR               ********   X     03
SCH$CHSE                   00000088 RG      03
SCH$CHSEP                  000000A4 RG      03
SCH$C_CEF               =  00000003
SCH$C_COLPG             =  00000001
SCH$C_COM               =  0000000C
SCH$C_COMO              =  0000000D
SCH$C_FPG               =  0000000B
SCH$C_HIB               =  00000007
SCH$C_HIBO              =  00000008
SCH$C_LEF               =  00000005
SCH$C_LEFO              =  00000006
SCH$C_MWAIT             =  00000002
SCH$C_PFW               =  00000004
SCH$C_SUSP              =  00000009
SCH$C_SUSPO             =  0000000A
SCH$FORCEDEXIT             0000018D RG      03
SCH$GB_PRI                 ********   X     03
SCH$GB_SIP                 ********   X     03
SCH$GL_AWSTIME             ********   X     03
SCH$GL_BORROWLIM           ********   X     03
SCH$GL_COMOQS              ********   X     03
SCH$GL_COMQS               ********   X     03
SCH$GL_FREECNT             ********   X     03
SCH$GL_FREELIM             ********   X     03
SCH$GL_MFYCNT              ********   X     03
SCH$GL_MFYLIM              ********   X     03
SCH$GL_PCBVEC              ********   X     03
SCH$GL_PFRATH              ********   X     03
SCH$GL_PFRATL              ********   X     03
SCH$GL_SWPPID              ********   X     03
SCH$GL_WSDEC               ********   X     03
SCH$GL_WSINC               ********   X     03
SCH$GW_AWSMIN              ********   X     03
SCH$GW_DELPHDCT            ********   X     03
```

```
SCH$GW_QUAN                     ********    X   03
SCH$QAST                        ********    X   03
SCH$QEND                        0000012A RG     03
SCH$RSE                         00000000 RG     03
SCH$SWPWAKE                     000002B5 RG     03
SCH$UNWAIT                      0000004D RG     03
SCH$WAKE                        00000290 RG     03
SENDAST                         0000024D R      03
SGN$GL_EXTRACPU                 ********    X   03
SS$_EXCPUTIM                =   000020AC
SS$_NONEXPR                 =   000008E8
SS$_NORMAL                  =   00000001
ST                          =   00000002
STACT                           0000001E R      03
STET                        =   00000000 R      02
STMSK                       =   00001000
SWPO                            00000112 R      03
SWPOE                           00000123 R      03
SYS$ADJWSL                      ********  GX    03
SYS$EXIT                        ********  GX    03
WAITMSK                         0000007B R      03
WAITST                      =   00000FFE
WQH$C_LENGTH                =   0000000C
WQH$L_WQBL                  =   00000004
WQH$W_WQCNT                 =   00000008
WSADJUST                        000001A0 R      03
WSERR                           00000249 R      03
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
AES2	0000002C (44.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
AES1	000002F9 (761.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.09	00:00:01.96
Command processing	123	00:00:00.48	00:00:05.87
Pass 1	331	00:00:10.62	00:00:34.14
Symbol table sort	0	00:00:01.68	00:00:04.35
Pass 2	140	00:00:02.57	00:00:09.68
Symbol table output	18	00:00:00.13	00:00:00.13
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	652	00:00:15.60	00:00:56.16

The working set limit was 1650 pages.
62765 bytes (123 pages) of virtual memory were used to buffer the intermediate code.

There were 60 pages of symbol table space allocated to hold 1060 non-local and 26 local symbols.
736 source lines were read in Pass 1, producing 19 object records in Pass 2.
26 pages of virtual memory were used to define 25 macros.

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

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	12
_255DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	20

1119 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:RSE/OBJ=OBJ$:RSE MSRC$:RSE/UPDATE=(ENH$:RSE)+EXECML$/LIB
```

SHELL
LIS

RSE
LIS

SCBVECTOR
LIS

SDAT
LIS

SHMGSDRTN
LIS

SCSVEC
LIS

SCHED
LIS

SECAUDIT
LIS

SPTSKEL
LIS

RUFSYSVEC
LIS