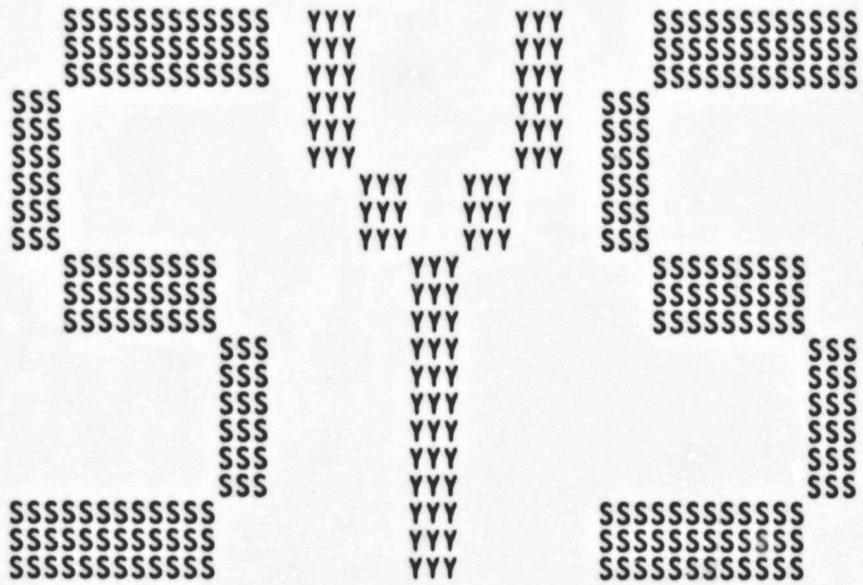
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS

```
  SSSSSSSS  YY      YY   SSSSSSSS  PPPPPPPP     CCCCCCCC  NN      NN  TTTTTTTTTT  RRRRRRRR    LL
  SSSSSSSS  YY      YY   SSSSSSSS  PPPPPPPP    CCCCCCCC  NN      NN  TTTTTTTTTT  RRRRRRRR    LL
SS          YY      YY  SS        PP      PP  CC        NN      NN      TT      RR      RR  LL
SS          YY      YY  SS        PP      PP  CC        NN      NN      TT      RR      RR  LL
SS            YY  YY    SS        PP      PP  CC        NNNN    NN      TT      RR      RR  LL
SS            YY  YY    SS        PP      PP  CC        NNNN    NN      TT      RR      RR  LL
  SSSSSS        YY      SSSSSS    PPPPPPPP    CC        NN  NN  NN      TT      RRRRRRRR    LL
  SSSSSS        YY      SSSSSS    PPPPPPPP    CC        NN  NN  NN      TT      RRRRRRRR    LL
        SS      YY            SS  PP          CC        NN    NNNN      TT      RR  RR      LL
        SS      YY            SS  PP          CC        NN    NNNN      TT      RR  RR      LL
        SS      YY            SS  PP          CC        NN      NN      TT      RR    RR    LL                  ....
      SS        YY            SS  PP          CC        NN      NN      TT      RR    RR    LL                  ....
SSSSSSSS        YY      SSSSSSSS  PP            CCCCCCCC  NN      NN      TT      RR      RR  LLLLLLLLLL      ....
SSSSSSSS        YY      SSSSSSSS  PP            CCCCCCCC  NN      NN      TT      RR      RR  LLLLLLLLLL      ....
```

```
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  SYSPCNTRL PROCESS CONTROL SERVICES
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, PROCESS CONTROL SYSTEM SERVICES
0000    30  ;
0000    31  ; ABSTRACT:
0000    32  ;       THIS MODULE CONTAINS THE ROUTINES WHICH IMPLEMENT THE PROCESS
0000    33  ;       CONTROL SERVICES, SUSPEND, RESUME, HIBERNATE AND WAKE.
0000    34  ;
0000    35  ; AUTHOR:
0000    36  ;       R. HUSTVEDT
0000    37  ;
0000    38  ; MODIFIED BY:
0000    39  ;
0000    40  ;       V03-013 LJK0256         Lawrence J. Kenah       7-Dec-1983
0000    41  ;               Only allow ASTs if XQP thread is active. Clear SUSPEN bit
0000    42  ;               if pool allocation fails.
0000    43  ;
0000    44  ;       V03-012 CWH3012         CW Hobbs                27-Sep-1983
0000    45  ;               In EXE$IPID_TO_EPID treat a null IPID as a special case,
0000    46  ;               and return the null.
0000    47  ;
0000    48  ;       V03-011 LJK0250         Lawrence J. Kenah       31-Aug-1983
0000    49  ;               Set the SUSPEN bit before lowering IPL to zero to insure
0000    50  ;               that the PCB of the target process has not disappeared.
0000    51  ;
0000    52  ;               Make the SUSPND AST a regular kernel AST so that it properly
0000    53  ;               interlocks with the XQP. Include the interlocking code.
0000    54  ;
0000    55  ;       V03-010 CWH1007         CW Hobbs                14-May-1983
0000    56  ;               Enable the storing of the actual cluster node info in the
0000    57  ;               high bits of the EPID.
```

```
0000     58 ;
0000     59 ;       V03-009 CWH1003         CW Hobbs                27-Apr-1983
0000     60 ;               Change pid conversion routines to do more checking
0000     61 ;               of pid against pids stored in PCB.  Make sure that
0000     62 ;               condition codes reflect final value in R0.
0000     63 ;
0000     64 ;       V03-008 LJK0197         Lawrence J. Kenah       25-Mar-1983
0000     65 ;               Insure that all success paths raise IPL to SYNCH.
0000     66 ;
0000     67 ;       V03-007 ACG0321         Andrew C. Goldstein,    24-Mar-1983  0:19
0000     68 ;               Allow non-privileged control over processes of equal UIC
0000     69 ;
0000     70 ;       V03-006 ROW0168         Ralph O. Weber           3-MAR-1983
0000     71 ;               Change W^ references to G^.
0000     72 ;
0000     73 ;       V03-005 CWH1002         CW Hobbs                19-Feb-1982
0000     74 ;               Modify EXE$NAMPID to use extended PIDs, add PID conversion
0000     75 ;               routines:
0000     76 ;                       EXE$IPID_TO_PCB         - internal pid to pcb address
0000     77 ;                       EXE$EPID_TO_PCB         - extended pid to pcb address
0000     78 ;                       EXE$IPID_TO_EPID        - internal pid to extended pid
0000     79 ;                       EXE$EPID_TO_IPID        - extended pid to internal pid
0000     80 ;
0000     81 ;       V03-004 LJK0188         Lawrence J. Kenah       22-Oct-1982
0000     82 ;               Do not allow processes that are being deleted to also
0000     83 ;               be suspended.
0000     84 ;
0000     85 ;       V03-003 KDM46395        Kathleen D. Morse       28-Jun-1982
0000     86 ;               Change word displacement to longword.
0000     87 ;
0000     88 ;---
```

```
                     0000     90          .SBTTL  DECLARATIONS
                     0000     91
                     0000     92 ;
                     0000     93 ; INCLUDE FILES:
                     0000     94 ;
                     0000     95
                     0000     96          $ACBDEF                         ; DEFINE AST CONTROL BLOCK
                     0000     97          $IPLDEF                         ; IPL DEFINITIONS
                     0000     98          $PCBDEF                         ; PCB OFFSET DEFINITIONS
                     0000     99          $PRDEF                          ; PROCESSOR REGISTER DEFS
                     0000    100          $PRIDEF                         ; PRIORITY INCREMENT DEFINITIONS
                     0000    101          $PRVDEF                         ; PRIVILEGE BIT DEFINITIONS
                     0000    102          $RSNDEF                         ; RESOURCE NUMBER DEFINITIONS
                     0000    103          $SSDEF                          ; STATUS DEFINITIONS
                     0000    104          $STATEDEF                       ; SCHEDULER STATE DEFINITIONS
                     0000    105 ;
                     0000    106 ; EQUATED SYMBOLS:
                     0000    107 ;
00000004             0000    108 PID=4                                    ; DISPLACEMENT TO PID ARGUMENT
00000008             0000    109 PRCNAM=8                                 ; DISPLACEMENT TO PROCESS NAME
                     0000    110
                00000000     111          .PSECT  AEXENONPAGED,BYTE       ; NONPAGED EXEC
                     0000    112
```

```
                     0000   114          .SBTTL  EXE$SUSPND - SUSPEND SYSTEM SERVICE
                     0000   115 ;++
                     0000   116 ;       EXE$SUSPND - SUSPEND SYSTEM SERVICE
                     0000   117 ;
                     0000   118 ; FUNCTIONAL DESCRIPTION:
                     0000   119 ;       EXE$SUSPND IMPLEMENTS THE SUSPEND PROCESS SYSTEM SERVICE.
                     0000   120 ;       THIS SERVICE CAUSES THE SPECIFIED PROCESS TO BE SUSPENDED
                     0000   121 ;       BY INITIATING A KERNEL MODE AST IF NOT THE CURRENT PROCESS.
                     0000   122 ;       A SUSPENDED PROCESS CANNOT RECEIVE ASTS AND WILL ONLY BE
                     0000   123 ;       RESUMED AS A RESULT OF THE RESUME SYSTEM SERVICE OR A
                     0000   124 ;       DELETE PROCESS REQUEST.
                     0000   125 ;
                     0000   126 ;
                     0000   127 ; CALLING SEQUENCE:
                     0000   128 ;       CALLG   ARGLIST,EXE$SUSPND
                     0000   129 ;
                     0000   130 ;
                     0000   131 ; INPUT PARAMETERS:
                     0000   132 ;       04(AP) - PROCESS IDENTIFICATION POINTER (PID)
                     0000   133 ;       08(AP) - PROCESS NAME DESCRIPTOR POINTER
                     0000   134 ;       R4 - PCB ADDRESS OF CURRENT PROCESS
                     0000   135 ;
                     0000   136 ; IMPLICIT INPUTS:
                     0000   137 ;       PCB OF CURRENT PROCESS
                     0000   138 ;       PCB OF TARGET PROCESS
                     0000   139 ;
                     0000   140 ;
                     0000   141 ; OUTPUT PARAMETERS:
                     0000   142 ;       R0 - COMPLETION STATUS
                     0000   143 ;       @PID(AP) - PROCESS IDENTIFICATION OF TARGET PROCESS
                     0000   144 ;
                     0000   145 ; COMPLETION CODES:
                     0000   146 ;       SS$_NORMAL  -  NORMAL SUCCESSFUL COMPLETION
                     0000   147 ;       SS$_NOPRIV  -  INSUFFICIENT PRIVILEGE FOR REQUESTED OPERATION
                     0000   148 ;       SS$_NONEXPR -  NON-EXISTENT PROCESS
                     0000   149 ;       SS$_ACCVIO  -  ACCESS VIOLATION ON WRITE DESTINATION
                     0000   150 ;       SS$_INSFMEM -  INSUFFICIENT DYNAMIC MEMORY FOR REQUEST
                     0000   151 ;                       ( ONLY RETURNED IF NO RESOURCE WAIT ENABLE )
                     0000   152 ;
                     0000   153 ; SIDE EFFECTS:
                     0000   154 ;       NONE
                     0000   155 ;
                     0000   156 ;--
                     0000   157
                     0000   158 EXE$SUSPND::                                ; SUSPEND SYSTEM SERVICE
                     003C   0000   159          .WORD   ^M<R2,R3,R4,R5>     ; REGISTER SAVE MASK FOR R2-R5
                   00AD     30   0002   160          BSBW    EXE$NAMPID          ; TRANSLATE AND VERIFY ARGS
                 30 50      E9   0005   161          BLBC    R0,30$              ; CONTINUE IF NO ERROR
                    51      DD   0008   162          PUSHL   R1                  ; SAVE PID
          24 24 A4   01      E0   000A   163          BBS     #PCB$V_DELPEN,PCB$L_STS(R4),20$ ; EXIT IF BEING DELETED
          1D 24 A4   0B      E2   000F   164          BBSS    #PCB$V_SUSPEN,PCB$L_STS(R4),10$ ; ... OR IF ALREADY SUSPENDED
                     0014   165          SETIPL  #IPL$_ASTDEL                ; ENABLE
                  FFE6'     30   0017   166          BSBW    EXE$ALLOCIRP        ; ALLOCATE I/O PACKET FOR AST
                 30 50      E9   001A   167          BLBC    R0,EXIT_NO_POOL     ; IF LBC THEN NO PACKET ALLOCATED
               55   52      D0   001D   168          MOVL    R2,R5               ; SETUP POINTER TO AST CONTROL BLK
          10 A5   3A'AF      DE   0020   169          MOVAL   B^SUSPND,ACB$L_AST(R5) ; SET FOR KERNEL AST OM PROCESS
                  0B A5      94   0025   170          CLRB    ACB$B_RMOD(R5)      ; SET ACCESS MODE FOR AST
```

```
                OC A5    8E   D0  0028   171          MOVL    (SP)+,ACB$L_PID(R5)     ; SET PID FOR AST
                         52   D4  002C   172          CLRL    R2                      ; SET NULL PRIORITY INCREMENT
                      FFCF' 30   002E   173          BSBW    SCH$QAST                ; QUEUE KERNEL AST
                         78   11  0031   174 10$:     BRB     EXITN                   ; EXIT WITH NORMAL STATUS
                                  0033   175
                50   08E8 8F   3C  0033   176 20$:     MOVZWL  #SS$_NONEXPR,R0         ; RETURN "NO SUCH PROCESS" IF DELPEN
                         74   11  0038   177 30$:     BRB     EXIT                    ; ERROR RETURN
```

```
                          003A        .SUBTITLE       KERNEL AST THAT SUSPENDS PROCESS
                          003A   ;+
                          003A   ;
                          003A   ; KERNEL AST ROUTINE TO SUSPEND PROCESS
                          003A   ;
                          003A   ; CALLING SEQUENCE:
                          003A   ;       (SAME EFFECT AS) DCLAST ASTADR=DELETE MODE=KERNEL
                          003A   ;
                          003A   ; INPUT PARAMETERS:
                          003A   ;       NONE
                          003A   ;
                          003A   ; OUTPUT PARAMETERS:
                          003A   ;       NONE
                          003A   ;
                          003A   ; IMPLICIT INPUTS:
                          003A   ;       PCB OF CURRENT PROCESS LOCATED VIA SCH$GL_CURPCB
                          003A   ;
                          003A   ; IMPLICIT OUTPUTS:
                          003A   ;       PCB$V_SUSPEN - CLEARED
                          003A   ;                               > WHEN PROCESS IS RESUMED
                          003A   ;       PCB$V_RESPEN - CLEARED
                          003A   ;-
                          003A
                          003A        .ENABLE         LOCAL_BLOCK
                          003A
                          003A   SUSPND:                                        ; SUSPEND KERNEL AST ROUTINE
                   003C   003A        .WORD   ^M<R2,R3,R4,R5>                  ; SAVE SOME REGISTERS
54   00000000'GF     D0   003C        MOVL    G^SCH$GL_CURPCB,R4               ; GET PCB ADDRESS
                          0043
                7E   DC   0043   10$: MOVPSL  -(SP)                            ; SAVE PSL ON STACK
                          0045        SETIPL  #IPL$_SYNCH                      ; DISABLE SYSTEM EVENTS
   09 24 A4     05   E5   0048        BBCC    #PCB$V_RESPEN,PCB$L_STS(R4),30$ ; BR IF NO PENDING RESUME
                          004D   EXIT_NO_POOL:
   00 24 A4     0B   E5   004D        BBCC    #PCB$V_SUSPEN,PCB$L_STS(R4),20$ ; CLEAR SUSPEND PENDING
                          0052   20$: SETIPL  #0                               ; DROP IPL TO ZERO
                     04   0055        RET                                      ; AND EXIT
                          0056
         2A A4       95   0056   30$: TSTB    PCB$B_DPC(R4)                    ; TEST FOR OUTSTANDING XQP ACTIVITY
                0F   13   0059        BEQL    40$                              ; BRANCH IF NONE (ALLOW SUSPENSION)
      0C A4     01   8A   005B        BICB2   #1,PCB$B_ASTACT(R4)              ; CLEAR KERNEL AST ACTIVE
             FF9E'   30   005F        BSBW    SCH$NEWLVL                       ; COMPUTE NEW AST LEVEL
         50     01   D0   0062        MOVL    #RSN$_ASTWAIT,R0                 ; NOTE AST RESOURCE
             FF98'   30   0065        BSBW    SCH$RWAIT                        ; WAIT FOR AST
                D9   11   0068        BRB     10$                              ; MAKE THE TEST AGAIN
                          006A
52   00000000'GF     DE   006A   40$: MOVAL   G^SCH$GQ_SUSP,R2                 ; GET QUEUE HEADER ADDRESS
             FF8C'   30   0071        BSBW    SCH$WAITR                        ; WAIT WITH CLEAN STACK
                CD   11   0074        BRB     10$                              ; AND CLEAR RESUME PENDING FLAG
                          0076
                          0076        .DISABLE        LOCAL_BLOCK
                          0076
```

```
                     0076   231          .SBTTL  EXE$RESUME - RESUME SYSTEM SERVICE
                     0076   232 ;++
                     0076   233 ;       EXE$RESUME - RESUME SYSTEM SERVICE
                     0076   234 ;
                     0076   235 ; FUNCTIONAL DESCRIPTION:
                     0076   236 ;       EXE$RESUME IMPLEMENTS THE RESUME SYSTEM SERVICE WHICH RESTARTS
                     0076   237 ;       A SUSPENDED PROCESS.
                     0076   238 ;
                     0076   239 ; INPUT PARAMETERS:
                     0076   240 ;       04(AP) - PROCESS IDENTIFICATION POINTER (PID)
                     0076   241 ;       08(AP) - PROCESS NAME DESCRIPTOR POINTER
                     0076   242 ;       R4 - PCB ADDRESS OF CURRENT PROCESS
                     0076   243 ;
                     0076   244 ; IMPLICIT INPUTS:
                     0076   245 ;       PCB OF CURRENT PROCESS
                     0076   246 ;       PCB OF TARGET PROCESS
                     0076   247 ;       PROCESS HEADER OF CURRENT PROCESS
                     0076   248 ;
                     0076   249 ; OUTPUT PARAMETERS:
                     0076   250 ;       R0 - COMPLETION STATUS
                     0076   251 ;       @PID - PROCESS IDENTIFICATION OF TARGET PROCESS
                     0076   252 ;
                     0076   253 ; IMPLICIT OUTPUTS:
                     0076   254 ;       NONE
                     0076   255 ;
                     0076   256 ; COMPLETION CODES:
                     0076   257 ;       SS$_NORMAL  -  NORMAL SUCCESSFUL COMPLETION
                     0076   258 ;       SS$_NOPRIV  -  INSUFFICIENT PRIVILEGE FOR REQUESTED OPERATION
                     0076   259 ;       SS$_NONEXPR -  NON-EXISTENT PROCESS
                     0076   260 ;       SS$_ACCVIO  -  ACCESS VIOLATION ON WRITE DESTINATION
                     0076   261 ;
                     0076   262 ; SIDE EFFECTS:
                     0076   263 ;       NONE
                     0076   264 ;--
                     0076   265
                     0076   266 EXE$RESUME::                                ; RESUME SYSTEM SERVICE
                 001C 0076   267          .WORD   ^M<R2,R3,R4>              ; REGISTER SAVE MASK FOR R2-R4
              38   10 0078   268          BSBB    EXE$NAMPID                ; CONVERT AND VALIDATE
           31 50   E9 007A   269          BLBC    R0,EXIT                   ; EXIT IF ERROR OCCURRED
        52    02   9A 007D   270          MOVZBL  #PRI$_RESAVL,R2           ; SET PRIORITY INCREMENT CLASS
00 24 A4    05   E2 0080   271          BBSS    #PCB$V_RESPEN,PCB$L_STS(R4),10$ ; SET RESUME PENDING
                     0085   272 10$:     RPTEVT  RESUME                    ; REPORT RESUME EVENT
              20   11 0089   273          BRB     EXITN                     ; AND TAKE NORMAL EXIT
                     008B   274
```

```
                     008B   276          .SBTTL  EXE$HIBER - HIBERNATE SYSTEM SERVICE
                     008B   277 ;++
                     008B   278 ;       EXE$HIBER - HIBERNATE SYSTEM SERVICE
                     008B   279 ;
                     008B   280 ; FUNCTIONAL DESCRIPTION:
                     008B   281 ;       EXE$HIBER IMPLEMENTS THE HIBERNATE SYSTEM SERVICE WHICH
                     008B   282 ;       PLACES THE PROCESS IN A WAIT STATE, HIB , UNTIL IT
                     008B   283 ;       IS RE-AWAKENED BY A WAKE SYSTEM SERVICE.  ASTS MAY BE DELIVERED
                     008B   284 ;       WHILE THE PROCESS IS IN A HIBERNATE STATE.
                     008B   285 ;
                     008B   286 ;
                     008B   287 ;
                     008B   288 ; CALLING SEQUENCE:
                     008B   289 ;       CALLG   ARGLIST,EXE$HIBER
                     008B   290 ;
                     008B   291 ;
                     008B   292 ; INPUT PARAMETERS:
                     008B   293 ;       R4 - PCB ADDRESS OF CURRENT PROCESS
                     008B   294 ;
                     008B   295 ; IMPLICIT INPUTS:
                     008B   296 ;       PROCESS CONTROL BLOCK(PCB) OF THE PROCESS ISSUING THE HIBERNATE
                     008B   297 ;       SYSTEM SERVICE.
                     008B   298 ;
                     008B   299 ;
                     008B   300 ; OUTPUT PARAMETERS:
                     008B   301 ;       R0 - COMPLETION STATUS CODE
                     008B   302 ;
                     008B   303 ; IMPLICIT OUTPUTS:
                     008B   304 ;       NONE
                     008B   305 ;
                     008B   306 ; COMPLETION CODES:
                     008B   307 ;       SS$_NORMAL - NORMAL SUCCESSFUL COMPLETION
                     008B   308 ;
                     008B   309 ; SIDE EFFECTS:
                     008B   310 ;       THE PROCESS WILL BE PLACED IN A WAIT STATE UNTIL EITHER
                     008B   311 ;       AN AST IS DELIVERED OR A WAKE REQUEST IS MADE.
                     008B   312 ;
                     008B   313 ;--
                     008B   314
                     008B   315
                     008B   316 EXE$HIBER::                                  ; HIBERNATE SYSTEM SERVICE
             001C    008B   317          .WORD   ^M<R2,R3,R4>                ; REGISTER SAVE MASK FOR R2-R4
                     008D   318          SETIPL  #IPL$_SYNCH                 ; BLOCK SCHEDULING EVENTS
 02 24 A4    0C  E7  0090   319          BBCCI   #PCB$V_WAKEPEN,PCB$L_STS(R4),10$ ; CHECK FOR PENDING WAKE
               14  11  0095   320          BRB     EXITN                       ; AND RETURN TO CALLER
                     0097   321
                     0097   322 10$:                                         ; MUST HIBERNATE
 52  00000000'GF    DE  0097   323          MOVAL   G^SCH$GQ_HIBWQ,R2          ; SET ADDRESS OF WAIT QUEUE HDR
             FF5F'   31  009E   324          BRW     SCH$WAIT                    ; AND WAIT
```

```
                        .SBTTL  EXE$WAKE - WAKE SYSTEM SERVICE
                ;++
                ;       EXE$WAKE - WAKE SYSTEM SERVICE
                ;
                ; FUNCTIONAL DESCRIPTION:
                ;       THE WAKE SYSTEM SERVICE CAUSES A PROCESS IN A HIBERNATE STATE
                ;       TO BE CHANGED TO AN EXECUTABLE STATE AND RE-EXECUTED.
                ;       IF THE TARGET OF A WAKE SERVICE IS NOT CURRENTLY HIBERNATING,
                ;       THEN A BIT IS POSTED WHICH WILL CAUSE A SUBSEQUENT HIBERNATE
                ;       CALL BY THAT PROCESS TO RETURN IMMEDIATELY.
                ;
                ; CALLING SEQUENCE:
                ;       CALLG   ARGLIST,EXE$WAKE
                ;
                ; INPUT PARAMETERS:
                ;       04(AP) = PROCESS IDENTIFICATION (PID) OF PROCESS TO WAKE
                ;       08(AP) = ADDRESS OF PROCESS NAME DESCRIPTOR
                ;       R4 - PCB ADDRESS
                ;
                ; IMPLICIT INPUTS:
                ;       PCB OF CURRENT PROCESS
                ;       ALL PCBS LOCATED BY THE VECTOR @SCH$GL_PCBVEC
                ;
                ; OUTPUT PARAMETERS:
                ;       R0 - COMPLETION STATUS CODE
                ;       @PID(AP) - PROCESS IDENTIFICATION (PID) OF PROCESS AWAKENED
                ;
                ; IMPLICIT OUTPUTS:
                ;       PCB$V_WAKEPEN BIT IN PCB$L_STS OF TARGET PROCESS WILL BE
                ;       SET IF PROCESS IS NOT HIBERNATING.
                ;
                ; COMPLETION CODES:
                ;       SS$_NORMAL - NORMAL SUCCESSFUL COMPLETION
                ;       SS$_NONEXPR - NON-EXISTENT PROCESS
                ;       SS$_NOPRIV - NO PRIVILEGE FOR ATTEMPTED OPERATION
                ;       SS$_ACCVIO - ACCESS VIOLATION ON WRITE DESTINATION
                ;
                ; SIDE EFFECTS:
                ;       THE TARGET PROCESS WILL BE CHANGED TO AN EXECUTABLE STATE,
                ;       COM OR COM0, IF IT IS IN A HIBERNATE STATE AND
                ;       RESCHEDULING WILL BE INITIATED IF NECESSARY.
                ;
                ;--
                EXE$WAKE::                              ; WAKE SYSTEM SERVICE
001C            .WORD   ^M<R2,R3,R4>                    ; SAVE MASK FOR R2-R4
0D   10         BSBB    EXE$NAMPID                      ; CONVERT NAME TO PID
                ;
                ;       R0 - SUCCESS INDICATOR
                ;       R1 - PID CORRESPONDING TO NAME STRING
                ;       R4 - PCB ADDRESS IF NAME WAS FOUND
                ;
06 50   E9      BLBC    R0,EXIT                         ; CONTINUE IF PROCESS LOCATED
FF55'   30      BSBW    SCH$WAKE                        ; WAKE PROCESS BY PID
                EXITN:                                  ; EXIT HIBERNATE SERVICE
50   01   3C    MOVZWL  #SS$_NORMAL,R0                  ; SET NORMAL COMPLETION
                EXIT:                                   ; RETURN WITH R0 SET
                SETIPL  #0                              ; ENABLE
```

```
04  00B1   383          RET                     ; AND RETURN TO CALLER
    00B2   384
    00B2   385
```

```
                     00B2   387          .SBTTL  EXE$NAMPID - CONVERT PROCESS NAME TO PID
                     00B2   388 ;++
                     00B2   389 ;       EXE$NAMPID  -  CONVERT PROCESS NAME TO PID
                     00B2   390 ;
                     00B2   391 ; FUNCTIONAL DESCRIPTION:
                     00B2   392 ;       EXE$NAMPID OBTAINS THE PROPER PID AND PCB ADDRESS FOR A
                     00B2   393 ;       STANDARD PROCESS CONTROL SERVICE ARGUMENT LIST CONSISTING
                     00B2   394 ;       OF A PID/PROCESS-NAME PAIR.  THE ABSENCE OF BOTH SELECTS THE
                     00B2   395 ;       CURRENT PROCESS.  AFTER ANY NECESSARY NAME TRANSLATION AND
                     00B2   396 ;       PID VALIDATION, GROUP AND WORLD PROCESS CONTROL PRIVILEGES
                     00B2   397 ;       ARE CHECKED.
                     00B2   398 ;
                     00B2   399 ;
                     00B2   400 ; CALLING SEQUENCE:
                     00B2   401 ;       JSB/BSB EXE$NAMPID
                     00B2   402 ;
                     00B2   403 ; INPUT PARAMETERS:
                     00B2   404 ;       PID(AP) - ADDRESS OF PID SOURCE/DESTINATION (EXTENDED PID)
                     00B2   405 ;       PRCNAM(AP) - POINTER TO PROCESS DESCRIPTOR TO CONVERT TO PID
                     00B2   406 ;       R4 - PCB ADDRESS
                     00B2   407 ;
                     00B2   408 ; IMPLICIT INPUTS:
                     00B2   409 ;       @SCH$GL_PCBVEC - VECTOR OF PCB ADDRESSES
                     00B2   410 ;       PHD$L_PRIV - PRIVILEGE BIT VECTOR IN PROCESS HEADER
                     00B2   411 ;
                     00B2   412 ; OUTPUT PARAMETERS:
                     00B2   413 ;       R0 - COMPLETION STATUS
                     00B2   414 ;       R1 - INTERNAL PROCESS IDENTIFICATION (PID) OF NAMED PROCESS.
                     00B2   415 ;            ZERO IF NO MATCH IS FOUND.
                     00B2   416 ;       R4 - PCB ADDRESS OF PROCESS IF MATCH IS FOUND.
                     00B2   417 ;       @PID(AP) - EXTENDED PROCESS IDENTIFICATION (EPID) OF SELECTED PROCESS
                     00B2   418 ;       IPL - IPL$_SYNCH (IPL UNCHANGED IF SS$_ACCVIO OR SS$_IVLOGNAM)
                     00B2   419 ;
                     00B2   420 ; COMPLETION CODES:
                     00B2   421 ;       SS$_NORMAL   -  NORMAL SUCCESSFUL COMPLETION
                     00B2   422 ;       SS$_IVLOGNAM  -  INVALID LOGICAL NAME STRING
                     00B2   423 ;       SS$_NONEXPR -  NONEXISTENT PROCESS OR INVALID PID
                     00B2   424 ;       SS$_NOPRIV  -  NO PRIVILEGE FOR SPECIFIED OPERATION.
                     00B2   425 ;       SS$_ACCVIO  -  ACCESS VIOLATION FOR WRITE DESTINATION
                     00B2   426 ;
                     00B2   427 ; SIDE EFFECTS:
                     00B2   428 ;       NONE
                     00B2   429 ;
                     00B2   430 ;--
                     00B2   431 EXE$NAMPID::                            ; TRANSLATE PNAME TO PID
50   04 AC    D0     00B2   432          MOVL    PID(AP),R0             ; GET PID ADDRESS
        18    13     00B6   433          BEQL    10$                    ; NO PID ADDRESS
                     00B8   434          IFNOWRT #4,(R0),ACCVIO         ; ERROR IF ACCESS VIOLATION
   51   60    D0     00BE   435          MOVL    (R0),R1                ; NOW FETCH (EXTENDED) PID
        0D    13     00C1   436          BEQL    10$                    ; BRANCH IF NO PID FOUND
   50   51    D0     00C3   437          MOVL    R1,R0                  ; PASS EPID TO ROUTINE IN R0
     00F9    30     00C6   438          BSBW    EXE$EPID_TO_IPID       ; CONVERT TO IPID
   51   50    D0     00C9   439          MOVL    R0,R1                  ; NOW R1 HAS THE USEFUL IPID
        50    D4     00CC   440          CLRL    R0                     ; CLEAR PID ADDRESS, DON'T NEED TO REWRITE S
        65    11     00CE   441          BRB     GOTPID                 ; YES,
51   60 A4    D0     00D0   442 10$:     MOVL    PCB$L_PID(R4),R1       ; ASSUME CALLERS PID
53   08 AC    D0     00D4   443          MOVL    PRCNAM(AP),R3          ; GET PNAME ADDRESS IF SPECIFIED
```

```
                    5B   13   00D8   444          BEQL    GOTPID                          ; NONE SPECIFIED, USE COMMON EXIT
                                00DA   445 20$:                                          ; MUST LOOK UP PROCESS NAME
                                00DA   446          IFNORD  #8,(R3),ACCVIO              ; CHECK DESCRIPTOR FOR READABLITY
                52  63   7D     00E0   447          MOVQ    (R3),R2                     ; GET DESCRIPTOR
                    52   B5     00E3   448          TSTW    R2                          ; AND CHECK FOR ZERO LENGTH
                    41   13     00E5   449          BEQL    IVLNAM                      ; NOT A VALID NAME STRING
                52  0F   B1     00E7   450          CMPW    #15,R2                      ; CHECK FOR MAXIMUM LENGTH
                    3C   1F     00EA   451          BLSSU   IVLNAM                      ; NOT A VALID NAME STRING
                                00EC   452          IFNORD  R2,(R3),ACCVIO              ; ACCESS VIOLATION IF STRING NOT READABLE
                    50   DD     00F2   453          PUSHL   R0                          ; SAVE PID ADDRESS
          50  00000000'EF   D0     00F4   454          MOVL    SCH$GL_MAXPIX,R0            ; INITIALIZE PROCESS INDEX
                                00FB   455 PIXLOOP:                                     ; LOOP FOR EACH PROCESS INDEX
        51  00000000'FF40   D0     00FB   456          MOVL    @L^SCH$GL_PCBVEC[R0],R1     ; GET PCB ADDRESS FROM VECTOR
     00BE C4    00BE C1   B1     0103   457          CMPW    PCB$W_GRP(R1),PCB$W_GRP(R4) ; COMPARE GROUP NUMBERS
                    11   12     010A   458          BNEQ    NEXTPIX                     ; NOT SAME GROUP, NEXT PIX
            70 A1   52   91     010C   459          CMPB    R2,PCB$T_LNAME(R1)          ; COMPARE NAME LENGTH
                    0B   12     0110   460          BNEQ    NEXTPIX                     ; DIFFERENT LENGTH
                    0F   BB     0112   461          PUSHR   #^M<R0,R1,R2,R3>            ; SAVE REGISTERS FOR CMPC3
         71 A1   63   52   29     0114   462          CMPC3   R2,(R3),PCB$T_LNAME+1(R1)    ; COMPARE TEXT OF NAME
                    0F   BA     0119   463          POPR    #^M<R0,R1,R2,R3>            ; RESTORE REGISTERS
                    11   13     011B   464          BEQL    GOTNAM                      ; FOUND A MATCHING PROCESS NAME
                                011D   465 NEXTPIX:                                     ; STEP TO NEXT PROCESS
                DB  50   F4     011D   466          SOBGEQ  R0,PIXLOOP                  ; UPDATE INDEX AND TRYA AGAIN
                    8E   D5     0120   467          TSTL    (SP)+                       ; CLEAN PID ADDRESS FROM STACK
                    2E   11     0122   468          BRB     NONEX                       ; EXIT WITH NONEXISTENT PROCESS STATUS
                                0124   469
                                0124   470 ACCVIO:                                     ; ACCESS VIOLATION
                50  0C   3C     0124   471          MOVZWL  #SS$_ACCVIO,R0              ; SET ERROR CODE
                    05   0127   472          RSB                                 ; AND EXIT
                                0128   473
                                0128   474 IVLNAM:                                     ; INVALID NAME
           50   0154 8F   3C     0128   475          MOVZWL  #SS$_IVLOGNAM,R0            ; SET ERROR CODE
                    05   012D   476          RSB                                 ; AND RETURN
                                012E   477
             51   60 A1   D0     012E   478 GOTNAM: MOVL    PCB$L_PID(R1),R1            ; GET FULL PID FOR NAME
                    50 8ED0   0132   479          POPL    R0                          ; RESTORE PID ADDRESS
                                0135   480 GOTPID:                                     ; VERIFY PID AND CHECK PRIV
                                0135   481          SETIPL  #IPL$_SYNCH                 ; BLOCK SYSTEM EVENTS
                52   51   3C     0138   482          MOVZWL  R1,R2                       ; EXTRACT PROCESS INDEX
      00000000'EF    52   D1     013B   483          CMPL    R2,SCH$GL_MAXPIX            ; TEST AGAINST MAXIMUM VALUE
                    0E   1A     0142   484          BGTRU   NONEX                       ; NONEXISTENT IF GTRU THAN MAXPIX
    52  00000000'FF42   D0     0144   485          MOVL    @L^SCH$GL_PCBVEC[R2],R2     ; GET PCB ADDRESS
             60 A2   51   D1     014C   486          CMPL    R1,PCB$L_PID(R2)            ; CHECK FOR VALID PID
                    06   13     0150   487          BEQL    VALPID                      ; YES,
                                0152   488 NONEX:                                     ; PROCESS NON-EXISTENT
           50   08E8 8F   3C     0152   489          MOVZWL  #SS$_NONEXPR,R0             ; SET ERROR STATUS
                    05   0157   490          RSB                                 ; AND RETURN TO CALLER
                                0158   491 VALPID:                                     ; PID IS VALID, CHECK PRIV
     0080 C4    0080 C2   D1     0158   492          CMPL    PCB$L_JIB(R2),PCB$L_JIB(R4) ; IS IT IN OUR JOB (TREE)?
                    1E   13     015F   493          BEQL    RETURN                      ; IF SO, ALLOW IT WITHOUT PRIVILEGES
     00BC C4    00BC C2   D1     0161   494          CMPL    PCB$L_UIC(R2),PCB$L_UIC(R4) ; DOES PROCESS HAVE SAME UIC?
                    15   13     0168   495          BEQL    RETURN                      ; IF SO, ALLOW IT WITHOUT PRIVILEGES
                                016A   496          IFPRIV  WORLD,RETURN,R4             ; SUCCESS IF WORLD PRIVILEGE
     00BE C4    00BE C2   B1     0170   497          CMPW    PCB$W_GRP(R2),PCB$W_GRP(R4) ; ARE GROUP NUMBERS EQUAL
                    1C   12     0177   498          BNEQ    NOPRIV                      ; IF NOT, NO PRIVILEGE
                                0179   499          IFNPRIV GROUP,NOPRIV,R4             ; ERROR IF NOT GROUP PRIV
                                017F   500 RETURN:                                      ; SUCCESSFUL EXIT
```

```
                54   52   D0  017F   501          MOVL    R2,R4                   ; MOVE PCB ADDRESS OF TARGET
                                0182   502                                          ; NORMAL STATUS EXIT
                     50   D5  0182   503          TSTL    R0                      ; WAS PID ADDRESS SPECIFIED
                     0B   13  0184   504          BEQL    10$                     ; NO, SKIP STORE OF PID
                                0186   505          SETIPL  #IPL$_ASTDEL            ; ALLOW PAGE FAULTS
              60    64   A4  D0  0189   506          MOVL    PCB$L_EPID(R4),(R0)     ; STORE EXTENDED PID IN DESTINATION
                     50   D4  018D   507          CLRL    R0                      ; DO NOT WRITE PID A SECOND TIME
                     A4   11  018F   508          BRB     GOTPID                  ; MAKE SURE THAT PID IS STILL VALID
                                0191   509
                50   01   3C  0191   510 10$:     MOVZWL  #SS$_NORMAL,R0          ; SET SUCCESS STATUS
                                05  0194   511          RSB                     ; AND RETURN TO CALLER
                50   24   3C  0195   512 NOPRIV:  MOVZWL  #SS$_NOPRIV,R0         ; SET ERROR STATUS
                                05  0198   513          RSB                     ; AND RETURN TO CALLER
                                0199   514
```

```
                0199   516          .SBTTL  EXE$xPID_TO_xxx - CONVERT PID TO OTHER PID OR PCB ADDRESS
                0199   517 ;++
                0199   518 ; FUNCTIONAL DESCRIPTIONS:
                0199   519 ;
                0199   520 ;       EXE$IPID_TO_PCB         - convert internal pid to pcb address
                0199   521 ;       EXE$EPID_TO_PCB         - convert extended pid to pcb address
                0199   522 ;       EXE$IPID_TO_EPID        - convert internal pid to extended pid
                0199   523 ;       EXE$EPID_TO_IPID        - convert extended pid to internal pid
                0199   524 ;
                0199   525 ; CALLING SEQUENCE:
                0199   526 ;       JSB/BSB EXE$xPID_TO_xxx
                0199   527 ;
                0199   528 ; INPUT PARAMETERS:
                0199   529 ;       R0      - input pid
                0199   530 ;
                0199   531 ; IMPLICIT INPUTS:
                0199   532 ;       @SCH$GL_PCBVEC - VECTOR OF PCB ADDRESSES
                0199   533 ;       SCH$GL_PIXWIDTH - WIDTH OF PIX FIELD IN EXTENDED PID
                0199   534 ;
                0199   535 ; OUTPUT PARAMETERS:
                0199   536 ;       R0      - output pid or pcb address, 0 if any problems
                0199   537 ;       CONDITION CODES - set according to the value in R0, so that any call
                0199   538 ;                         can be followed by a BEQL without another test
                0199   539 ;
                0199   540 ; COMPLETION CODES:
                0199   541 ;       NONE
                0199   542 ;
                0199   543 ; SIDE EFFECTS:
                0199   544 ;
                0199   545 ;       Non-paged code and data, no page faults possible.
                0199   546 ;
                0199   547 ;       Callers of these routines must be prepared for the routines to save
                0199   548 ;       registers R1 through R5 to allow for future additions.  For example,
                0199   549 ;       a BLISS linkage declaration of
                0199   550 ;
                0199   551 ;               LINKAGE
                0199   552 ;                   pid_call = JSB (REGISTER=0) : PRESERVE (1,2,3,4,5)
                0199   553 ;                                                 NOTUSED (6,7,8,9,10,11);
                0199   554 ;
                0199   555 ;       will force the enclosing procedure to save R2-R5 in the procedure
                0199   556 ;       entry mask.
                0199   557 ;--
                0199   558
                0199   559 ;+
                0199   560 ; Convert an extended PID to a PCB address.  We will first convert the EPID to an
                0199   561 ; IPID, then convert the IPID to the PCB address.  The condition codes will be set
                0199   562 ; according to the value in R0.
                0199   563 ;-
                0199   564 EXE$EPID_TO_PCB::                ; CONVERT EXTENDED PID TO PCB ADDRESS
 27   10        0199   565          BSBB    EXE$EPID_TO_IPID        ; GET THE IPID IN R0
 02   13        019B   566          BEQL    10$                     ; COULDN'T CONVERT THE EPID
 01   10        019D   567          BSBB    EXE$IPID_TO_PCB         ; CONVERT THE IPID TO THE PCB ADDR
      05        019F   568 10$:     RSB
                01A0   569
                01A0   570 ;+
                01A0   571 ; Convert internal PID to PCB address.  Return 0 if the input IPID does not match
                01A0   572 ; the IPID stored in the corresponding PCB.  Set the condition codes according to
```

```
                                01A0    573 ; the presence of a returned address in R0, so that the BSBx can be followed by a
                                01A0    574 ; BEQL or BNEQ
                                01A0    575 ;-
                                01A0    576 EXE$IPID_TO_PCB::                  ; CONVERT INTERNAL PID TO PCB ADDRESS
    00000000'EF    50    B1     01A0    577         CMPW    R0,SCH$GL_MAXPIX       ; TEST AGAINST MAXIMUM VALUE
                  16    1A     01A7    578         BGTRU   10$                    ; NONEXISTENT IF GTRU THAN MAXPIX
                  50    DD     01A9    579         PUSHL   R0                     ; SAVE A COPY OF THE IPID
                50      50     3C     01AB    580         MOVZWL  R0,R0          ; EXTRACT PROCESS INDEX FIELD
          50    00000000'FF40    D0     01AE    581         MOVL    @SCH$GL_PCBVEC[R0],R0  ; MOVE PCB ADDRESS TO R0
              8E    60  A0     D1     01B6    582         CMPL    PCB$L_PID(R0),(SP)+    ; DOES THE PID IN THE PCB MATCH?
                  03    12     01BA    583         BNEQ    10$                    ; NO MATCH, RETURN 0 ADDRESS
                  50    D5     01BC    584         TSTL    R0                     ; SET THE CONDITION CODES
                        05     01BE    585         RSB
                  50    D4     01BF    586 10$:    CLRL    R0                     ; NONEXISTENT PID, RETURN ZERO
                        05     01C1    587         RSB
                               01C2    588
                               01C2    589 ;+
                               01C2    590 ; Convert an extended PID to the internal PID.  Return 0 if the EPID refers to
                               01C2    591 ; another node.  Do not check that either the EPID or IPID are valid.
                               01C2    592 ;-
                               01C2    593 EXE$EPID_TO_IPID::                 ; CONVERT EXTENDED PID TO INTERNAL PID
                  06    BB     01C2    594         PUSHR   #^M<R1,R2>             ; SAVE SOME WORKING REGISTERS
                               01C4    595
                               01C4    596 ; WE WILL EXTRACT THE NODE FIELD FROM THE EPID TO SEE IF THIS IS FOR THE LOCAL
                               01C4    597 ; NODE.  WE WILL INCLUDE THE WILDCARD BIT IN THIS TEST.  VERIFY SOME ASSUMPTIONS
                               01C4    598 ; ABOUT THE LOCATIONS OF THESE FIELDS.
                               01C4    599
             0000000A          01C4    600 NODE_WIDTH = PCB$S_EPID_NODE_IDX+PCB$S_EPID_NODE_SEQ
                               01C4    601
                               01C4    602         ASSUME  PCB$V_EPID_WILD EQ -           ; CHECK THAT WILD BIT IS RIGHT
                               01C4    603                 <PCB$V_EPID_NODE_IDX + NODE_WIDTH> ;  AFTER NODE FIELDS
                               01C4    604         ASSUME  PCB$V_EPID_NODE_SEQ EQ -       ;  AND SEQ IS RIGHT AFTER IDX
                               01C4    605                 <PCB$V_EPID_NODE_IDX + PCB$S_EPID_NODE_IDX>
                               01C4    606
  51    50    0B    15    EF   01C4    607         EXTZV   #PCB$V_EPID_NODE_IDX, - ; MOVE NODE + WILD TO R1
                               01C9    608                 #<NODE_WIDTH+1>,R0,R1
                  09    13     01C9    609         BEQL    10$                    ; TREAT NODE ZERO AS LOCAL NODE ??
   51    00000000'EF    B1     01CB    610         CMPW    SCH$GW_LOCALNODE,R1    ; IS IT THE LOCAL NODE?
                  1D    12     01D2    611         BNEQ    30$                    ; NOT LOCAL, CAN'T MAKE AN IPID
                               01D4    612
                               01D4    613 ; EPID IN R0 IS FOR LOCAL NODE, EXTRACT THE PIX AND SEQUENCE NUMBER TO FORM IPID
                               01D4    614
   51    00000000'EF    D0     01D4    615 10$:    MOVL    SCH$GL_PIXWIDTH,R1     ; LOAD WIDTH OF EXTENDED PIX FIELD
      52    15    51    C3     01DB    616         SUBL3   R1,#PCB$S_EPID_PROC,R2 ;  AND WIDTH OF THE SEQ NUM FIELD
  52  50    52    51    EF     01DF    617         EXTZV   R1,R2,R0,R2            ; R2 IS LONGWORD SEQ NUM
  50  50    51    00    EF     01E4    618         EXTZV   #0,R1,R0,R0            ; R0 IS LONGWORD PIX
  50  0F    10    52    F0     01E9    619         INSV    R2,#16,#15,R0          ; INSERT SEQ NUM IN HIGH WORD
                               01EE    620                                        ;  WHICH MAKES AN IPID IN R0
                  06    BA     01EE    621 20$:    POPR    #^M<R1,R2>             ; RESTORE REGISTERS
                        05     01F0    622         RSB                            ; CONDITION CODES SET FOR VALUE OF R0
                               01F1    623
                               01F1    624 ; COULD NOT TURN EPID INTO AN IPID, RETURN AN IPID OF 0
                               01F1    625
                  50    D4     01F1    626 30$:    CLRL    R0                     ; RETURN ZERO PID (& COND CODE = 0)
                  F9    11     01F3    627         BRB     20$                    ; RESTORE REGISTERS AND RETURN
                               01F5    628
                               01F5    629 ;+
```

```
                                01F5   630 ; Convert an IPID to an EPID.  We do not check that the IPID is valid.  The local
                                01F5   631 ; node is moved into the node field of the EPID, the seq number and pix of the IPID
                                01F5   632 ; are moved into the EPID.  The condition codes reflect the final value of R0.
                                01F5   633 ;-
                                01F5   634 EXE$IPID_TO_EPID::                   ; INTERNAL PID TO EXTENDED PID
                          50    D5 01F5   635         TSTL    R0                ; TREAT A ZERO PID AS A SPECIAL CASE
                          28    13 01F7   636         BEQL    10$               ; ZERO, WE DON'T TOUCH IT
                          0E    BB 01F9   637         PUSHR   #^M<R1,R2,R3>     ; SAVE SOME WORKING REGISTERS
                       53       50 01FB   638         MOVZWL  R0,R3             ; R3 IS LONGWORD PIX
                    50    50    F0 8F 78 01FE   639         ASHL    #-16,R0,R0        ; R0 IS LONGWORD SEQ NUM
                 51    00000000'EF    D0 0203   640         MOVL    SCH$GL_PIXWIDTH,R1        ; LOAD WIDTH OF EXTENDED PIX FIELD
              52    15    51    C3 020A   641         SUBL3   R1,#PCB$S_EPID_PROC,R2    ; AND THE WIDTH OF THE SEQ NUM FIELD
           53    52    51    50    F0 020E   642         INSV    R0,R1,R2,R3       ; INSERT SEQ NUM BESIDE PIX
53    0A    15    00000000'EF    F0 0213   643         INSV    SCH$GW_LOCALNODE, -       ; INSERT LOCAL NODE INTO THE EPID
                                021C   644                 #PCB$V_EPID_NODE_IDX, #NODE_WIDTH, R3
                       50    53    D0 021C   645         MOVL    R3,R0             ; RETURN THE EPID IN R0
                          0E    BA 021F   646         POPR    #^M<R1,R2,R3>     ; RESTORE REGISTERS
                          05    0221   647 10$:     RSB                       ; N.B. COND CODES SET ON VALUE OF R0
                                0222   648
```

```
                     0222   650          .SBTTL  EXE$SETPRN - SET PROCESS NAME
                     0222   651
                     0222   652 ;++
                     0222   653 ; FUNCTIONAL DESCRIPTION:
                     0222   654 ;       EXE$SETPRN IMPLEMENTS THE SET PROCESS NAME SYSTEM
                     0222   655 ;       SERVICE WHICH ALLOWS A PROCESS TO ESTABLISH A LOGICAL NAME
                     0222   656 ;       FOR ITSELF.  ALL SUCH LOGICAL NAMES ARE IMPLICITLY QUALIFIED
                     0222   657 ;       BY THE GROUP NUMBER OF THE PROCESS THEREBY ALLOWING THE SAME
                     0222   658 ;       LOGICAL NAME TO BE USED BY PROCESSES IN DIFFERENT GROUPS.
                     0222   659 ;
                     0222   660 ; CALLING SEQUENCE:
                     0222   661 ;       CALLG   ARGLIST,EXE$SETPRN
                     0222   662 ;
                     0222   663 ; INPUT PARAMETERS:
                     0222   664 ;       04(AP) - ADDRESS OF PROCESS NAME STRING DESCRIPTOR
           00000004  0222   665 PRCNAM=4
                     0222   666 ;       R4 - PCB ADDRESS OF CURRENT PROCESS
                     0222   667 ;
                     0222   668 ; IMPLICIT INPUTS:
                     0222   669 ;       SCH$GL_CURPCB - POINTER TO PCB OF CURRENT PROCESS
                     0222   670 ;       @SCH$GL_PCBVEC - VECTOR OF ALL PCB ADDRESSES
                     0222   671 ;
                     0222   672 ; OUTPUT PARAMETERS:
                     0222   673 ;       NONE
                     0222   674 ;
                     0222   675 ; IMPLICIT OUTPUTS:
                     0222   676 ;       PCB$T_NAME IN CURRENT PCB IS FILLED WITH THE SPECIFIED NAME
                     0222   677 ;       PROVIDED NO ERROR HAS OCURRED.
                     0222   678 ;
                     0222   679 ; SIDE EFFECTS:
                     0222   680 ;       NONE
                     0222   681 ;
                     0222   682 ; COMPLETION CODES:
                     0222   683 ;       SS$_NORMAL    -  NORMAL SUCCESSFUL COMPLETION STATUS
                     0222   684 ;       SS$_ACCVIO    -  ALL OR PART OF NAME STRING IS INACCESSIBLE FOR READ
                     0222   685 ;       SS$_IVLOGNAM  -  ILLEGAL LOGICAL NAME STRING LENGTH (>15)
                     0222   686 ;       SS$_DUPLNAM   -  DUPLICATE PROCESS NAME WITHIN GROUP
                     0222   687 ;
                     0222   688 ;--
                     0222   689
                     0222   690 EXE$SETPRN::                                ; SET PROCESS NAME
                 00FC 0222   691          .WORD   ^M<R2,R3,R4,R5,R6,R7>    ; SAVE REGISTERS R2-R7
          55  04 AC   D0    0224   692          MOVL    PRCNAM(AP),R5        ; GET ADDRESS OF PROCESS NAME
                05   12    0228   693          BNEQ    5$                   ; WAS SPECIFIED
             70 A4   D4    022A   694          CLRL    PCB$T_LNAME(R4)      ; CLEAR NAME FIELD OF PCB
                59   11    022D   695          BRB     65$                  ; AND EXIT WITH NORMAL STATUS
                      022F   696 5$:     IFNORD  #8,(R5),80$          ; CHECK ACCESS FOR DESCRIPTOR
             7E 65   7D    0235   697          MOVQ    (R5),-(SP)           ; PUSH DESCRIPTOR ON STACK
                6E   B5    0238   698          TSTW    (SP)                 ; CHECK FOR ZERO LENGTH STRING
                0C   13    023A   699          BEQL    10$                  ; INVALID NAME
                      023C   700          IFNORD  (SP),@4(SP),80$      ; PROBE ENDS OF STRING
             0F 6E   B1    0243   701          CMPW    (SP),#15             ; CHECK FOR MAXIMUM LENGTH
                06   1B    0246   702          BLEQU   20$                  ; IF LEQU, WITHIN LIMIT
          50  0154 8F   3C    0248   703 10$:     MOVZWL  #SS$_IVLOGNAM,R0     ; INVALID PROCESS NAME STATUS
                04   024D   704          RET                          ; AND RETURN
       56  00000000'EF   D0    024E   705 20$:     MOVL    SCH$GL_MAXPIX,R6     ; SET MAXIMUM PROCESS INDEX
   57  00000000'FF46   D0    0255   706 30$:     MOVL    @L^SCH$GL_PCBVEC[R6],R7 ; GET PCB ADDRESS
```

```
                00BE C7    00BE C4    B1   025D   707            CMPW    PCB$W_GRP(R4),PCB$W_GRP(R7)  ; CHECK FOR SAME GROUP
                                      0E   12   0264   708            BNEQ    40$                          ; NO, SKIP IT
                      70 A7    6E     91   0266   709            CMPB    (SP),PCB$T_LNAME(R7)         ; COMPARE LENGTHS
                                      08   12   026A   710            BNEQ    40$                          ; NOT EQUAL, TRY ANOTHER
                71 A7    04 BE    6E     29   026C   711            CMPC3   (SP),@4(SP),PCB$T_LNAME+1(R7)    ; COMPARE NAMES WITH COUNTS
                                      05   13   0272   712            BEQL    50$                          ; MATCH
                         DE       56     F4   0274   713 40$:       SOBGEQ  R6,30$                       ; CONTINUE FOR ALL PCBS
                                      05   11   0277   714            BRB     60$                          ; NOT FOUND
                            57     54     D1   0279   715 50$:       CMPL    R4,R7                        ; SAME PROCESS?
                                      0E   12   027C   716            BNEQ    70$                          ; DUPLICATE NAME ERROR
                      70 A4    6E     90   027E   717 60$:       MOVB    (SP),PCB$T_LNAME(R4)         ; SAVE NAME LENGTH
                71 A4    04 BE    6E     28   0282   718            MOVC3   (SP),@4(SP),PCB$T_LNAME+1(R4)    ; MOVE NAME TO PCB
                            50     01     3C   0288   719 65$:       MOVZWL  #SS$_NORMAL,R0               ; SUCCESSFUL STATUS
                                      04   028B   720            RET                                  ; AND RETURN
                   50    0094 8F    3C   028C   721 70$:       MOVZWL  #SS$_DUPLNAM,R0              ; DUPLICATE NAME WITHIN GROUP
                                      04   0291   722            RET                                  ; AND RETURN
                                           0292   723
                            50     0C     3C   0292   724 80$:       MOVZWL  #SS$_ACCVIO,R0               ; ACCESS VIOLATION
                                      04   0295   725            RET                                  ; RETURN WITH ERROR STATUS
                                           0296   726            .END
```

```
ACB$B_RMOD                      =  0000000B
ACB$L_AST                       =  00000010
ACB$L_PID                       =  0000000C
ACCVIO                             00000124  R       02
EVT$_RESUME                        ********    X     02
EXE$ALLOCIRP                       ********    X     02
EXE$EPID_TO_IPID                   000001C2  RG      02
EXE$EPID_TO_PCB                    00000199  RG      02
EXE$HIBER                          0000008B  RG      02
EXE$IPID_TO_EPID                   000001F5  RG      02
EXE$IPID_TO_PCB                    000001A0  RG      02
EXE$NAMPID                         000000B2  RG      02
EXE$RESUME                         00000076  RG      02
EXE$SETPRN                         00000222  RG      02
EXE$SUSPND                         00000000  RG      02
EXE$WAKE                           000000A1  RG      02
EXIT                               000000AE  R       02
EXITN                              000000AB  R       02
EXIT_NO_POOL                       0000004D  R       02
GOTNAM                             0000012E  R       02
GOTPID                             00000135  R       02
IPL$_ASTDEL                     =  00000002
IPL$_SYNCH                      =  00000008
IVLNAM                             00000128  R       02
NEXTPIX                            0000011D  R       02
NODE_WIDTH                      =  0000000A
NONEX                              00000152  R       02
NOPRIV                             00000195  R       02
PCB$B_ASTACT                    =  0000000C
PCB$B_DPC                       =  0000002A
PCB$L_EPID                      =  00000064
PCB$L_JIB                       =  00000080
PCB$L_PID                       =  00000060
PCB$L_STS                       =  00000024
PCB$L_UIC                       =  000000BC
PCB$Q_PRIV                      =  00000084
PCB$S_EPID_NODE_IDX             =  00000008
PCB$S_EPID_NODE_SEQ             =  00000002
PCB$S_EPID_PROC                 =  00000015
PCB$T_LNAME                     =  00000070
PCB$V_DELPEN                    =  00000001
PCB$V_EPID_NODE_IDX             =  00000015
PCB$V_EPID_NODE_SEQ             =  0000001D
PCB$V_EPID_WILD                 =  0000001F
PCB$V_RESPEN                    =  00000005
PCB$V_SUSPEN                    =  0000000B
PCB$V_WAKEPEN                   =  0000000C
PCB$W_GRP                       =  000000BE
PID                             =  00000004
PIXLOOP                            000000FB  R       02
PR$_IPL                         =  00000012
PRCNAM                          =  00000004
PRI$_RESAVL                     =  00000002
PRV$V_GROUP                     =  00000008
PRV$V_WORLD                     =  00000010
RETURN                             0000017F  R       02
RSN$_ASTWAIT                    =  00000001
SCH$GL_CURPCB                      ********    X     02
SCH$GL_MAXPIX                      ********    X     02
SCH$GL_PCBVEC                      ********    X     02
SCH$GL_PIXWIDTH                    ********    X     02
SCH$GQ_HIBWQ                       ********    X     02
SCH$GQ_SUSP                        ********    X     02
SCH$GW_LOCALNODE                   ********    X     02
SCH$NEQLVL                         ********    X     02
SCH$QAST                           ********    X     02
SCH$RSE                            ********    X     02
SCH$RWAIT                          ********    X     02
SCH$WAIT                           ********    X     02
SCH$WAITK                          ********    X     02
SCH$WAKE                           ********    X     02
SS$_ACCVIO                      =  0000000C
SS$_DUPLNAM                     =  00000094
SS$_IVLOGNAM                    =  00000154
SS$_NONEXPR                     =  000008E8
SS$_NOPRIV                      =  00000024
SS$_NORMAL                      =  00000001
SUSPND                             0000003A  R       02
VALPID                             00000158  R       02
```

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
AEXENONPAGED	00000296 (662.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.29
Command processing	105	00:00:00.56	00:00:01.82
Pass 1	288	00:00:08.52	00:00:16.46
Symbol table sort	0	00:00:01.25	00:00:02.88
Pass 2	141	00:00:02.21	00:00:05.49
Symbol table output	11	00:00:00.07	00:00:00.14
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	578	00:00:12.70	00:00:27.10

The working set limit was 1500 pages.
49110 bytes (96 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 807 non-local and 28 local symbols.
726 source lines were read in Pass 1, producing 16 object records in Pass 2.
23 pages of virtual memory were used to define 22 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	12
_255DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	19

909 GETS were required to define 19 macros.

There were no errors, warnings or information messages.

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

0387 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SYSPURGWS
LIS

SYSPUTMSG
LIS

SYSPCNTRL
LIS

SYSQIOFDT
LIS

SYSQIOREQ
LIS

SYSRUNDWN
LIS

SYSRDBRES
LIS

SYSRTSLST
LIS