L 12

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
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
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

```
**FILE**ID**USRVECTOR                                    C 9
```

SYS
V04

```
UU      UU    SSSSSSSS  RRRRRRRR    VV      VV  EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT    OOOOOO    RRRRRRRR
UU      UU    SSSSSSSS  RRRRRRRR    VV      VV  EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT    OOOOOO    RRRRRRRR
UU      UU  SS          RR      RR  VV      VV  EE          CC              TT      OO      OO  RR      RR
UU      UU  SS          RR      RR  VV      VV  EE          CC              TT      OO      OO  RR      RR
UU      UU  SS          RR      RR  VV      VV  EE          CC              TT      OO      OO  RR      RR
UU      UU  SS          RR      RR  VV      VV  EE          CC              TT      OO      OO  RR      RR
UU      UU    SSSSSS    RRRRRRRR    VV      VV  EEEEEEEE    CC              TT      OO      OO  RRRRRRRR
UU      UU    SSSSSS    RRRRRRRR    VV      VV  EEEEEEEE    CC              TT      OO      OO  RRRRRRRR
UU      UU          SS  RR  RR      VV      VV  EE          CC              TT      OO      OO  RR  RR
UU      UU          SS  RR  RR      VV      VV  EE          CC              TT      OO      OO  RR  RR
UU      UU          SS  RR    RR      VV  VV    EE          CC              TT      OO      OO  RR    RR        ....
UU      UU          SS  RR    RR      VV  VV    EE          CC              TT      OO      OO  RR    RR        ....
UUUUUUUUUU  SSSSSSSS    RR      RR      VV      EEEEEEEEEE    CCCCCCCC      TT        OOOOOO    RR      RR      ....
UUUUUUUUUU  SSSSSSSS    RR      RR      VV      EEEEEEEEEE    CCCCCCCC      TT        OOOOOO    RR      RR      ....


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

```
00000001  0000   LIBSWITCH=1                              ;GENERATE LIBRARY FORM OF SERVICE VECTOR
          0000          .IF     NDF,LIBSWITCH
          0000          .TITLE  SHELL PROCESS DEFINITION
          0000          .IFF
          0000          .TITLE  SYS$USRVECTOR - USER ACCESSIBLE P1 CELLS
          0000          .ENDC
          0000          .IDENT  'V04-000'
          0000   ;
          0000   ;****************************************************************************
          0000   ;*                                                                          *
          0000   ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
          0000   ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
          0000   ;*  ALL RIGHTS RESERVED.                                                    *
          0000   ;*                                                                          *
          0000   ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
          0000   ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
          0000   ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
          0000   ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
          0000   ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
          0000   ;*  TRANSFERRED.                                                            *
          0000   ;*                                                                          *
          0000   ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
          0000   ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
          0000   ;*  CORPORATION.                                                            *
          0000   ;*                                                                          *
          0000   ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
          0000   ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
          0000   ;*                                                                          *
          0000   ;*                                                                          *
          0000   ;****************************************************************************
          0000
          0000   ;++
          0000   ; FACILITY:     EXECUTIVE, CREATE PROCESS/SWAPPER DATA BASE
          0000   ;
          0000   ; ABSTRACT:     SHELL DEFINES THE SWAP IMAGE FOR THE INITIAL PROCESS
          0000   ;               STARTED BY THE CREATE PROCESS SYSTEM SERVICE.
          0000   ;
          0000   ; ENVIRONMENT:
          0000   ;
          0000   ; AUTHOR: RICHARD I. HUSTVEDT, CREATION DATE: 30-NOV-76
          0000   ;
          0000   ; MODIFIED BY:
          0000   ;
          0000   ;       V03-039 RAS0319         Ron Schaefer            29-Jun-1984
          0000   ;               Add queue header cells for logical name table name
          0000   ;               translation cache.
          0000   ;
          0000   ;       V03-038 LJK0287         Lawrence J. Kenah       25-Jun-1984
          0000   ;               Add three longword array to contain the counts of termination
          0000   ;               handlers declared in each of the three outer access modes.
          0000   ;               Add sequence number cell for process private logical name
          0000   ;               directory.
          0000   ;
          0000   ;       V03-037 MSH0054         Michael S. Harvey       30-May-1984
          0000   ;               Remove some obsolete symbols.
          0000   ;
          0000   ;       V03-036 LJK0282         Lawrence J. Kenah       9-May-1984
```

```
0000  ;                 Add demand zero page after PIO page to accommodate long
0000  ;                 directory names that straddle page boundary.
0000  ;
0000  ;         V03-035 TMK0001         Todd M. Katz            27-Apr-1984
0000  ;                 Remove CTL$GL_LOGTBL which was required only under the old
0000  ;                 logical name design.
0000  ;
0000  ;         V03-034 LJK0272         Lawrence J. Kenah       10-Apr-1984
0000  ;                 Add image activator context cells privileged vectors
0000  ;                 and shareable image initialization.
0000  ;
0000  ;         V03-033 MHB0133         Mark Bramhall            9-Apr-1984
0000  ;                 Remove preset values for CTL$T_USERNAME and CTL$T_ACCOUNT.
0000  ;                 Validate CLI information ordering.
0000  ;
0000  ;         V03-032 RAS0281         Ron Schaefer            28-Mar-1984
0000  ;                 Add PIO$GB_DFNBC process-specific RMS parameter for
0000  ;                 network block count transfer size.
0000  ;                 Re-use the cell once used from RMS Compatibility.
0000  ;                 Add CTL$GT_SPAWNCLI and CTL$GT_SPAWNTABLE as well.
0000  ;
0000  ;         V03-031 WMC0006         Wayne Cardoza           21-Mar-1984
0000  ;                 Go back to 3 pages of kernel stack.
0000  ;                 Add cells for default image I/O area.
0000  ;
0000  ;         V03-030 MSH0006         Michael S. Harvey       3-Feb-1984
0000  ;                 Protect image header buffer page against USER WRITE.
0000  ;
0000  ;                 Protect image activator scratch page from user by changing
0000  ;                 its owner to EXEC.
0000  ;
0000  ;                 Replace obsolete image rundown control flags with an image
0000  ;                 list pointer for use by the debugger. This fixed cell pointer
0000  ;                 allows the debugger to avoid linking against the executive.
0000  ;
0000  ;         V03-029 RSH0091         R. Scott Hanna          31-Jan-1984
0000  ;                 Add dedicated P1 demand zero pages for the security auditing
0000  ;                 Impure Data Table (IDT). Remove the old vector page pointer
0000  ;                 to the IDT.
0000  ;
0000  ;         V03-028 LJK0258         Lawrence J. Kenah       18-Jan-1984
0000  ;                 Correct problems introduced by LJK0257. In SHELINIT, only
0000  ;                 reorder a number of pagesequal to the  pages in this module
0000  ;                 that are NOT kernel stack pages. Convert a PIO page from
0000  ;                 a page file page to demand zero.
0000  ;
0000  ;         V03-027 BLS0262         Benn Schreiber          16-Jan-1984
0000  ;                 Correct .ADDRESS to be .LONG in P1 lookaside list links
0000  ;
0000  ;         V03-026 LJK0257         Lawrence J. Kenah       4-Jan-1984
0000  ;                 A variety of changes.
0000  ;
0000  ;                     Add CTL$GQ_TERMCHAR and CREPRC_FLAGS to P1 vector page.
0000  ;
0000  ;                     Add UAF_FLAGS to CLI data page.
0000  ;
0000  ;                     Change PQB$C_MAXDIRLEN references to PQB$S_DDSTRING.
```

```
0000 ;
0000 ;                   Add secor  page for compatibility mode context.
0000 ;
0000 ;                   Reorder SHELL pages to achieve demand zero compaction.
0000 ;
0000 ;                   Add listhead and demand zero pages for P1 pool lookaside
0000 ;                   list.
0000 ;
0000 ;                   Move page file index and characteristics from PQB to PCB.
0000 ;
0000 ;       V03-025 WMC0005          Wayne Cardoza            02-Dec-1983
0000 ;               PHD$W_BAK, PHD$W_WSLX are now longwords.
0000 ;
0000 ;       V03-024 WMC0004          Wayne Cardoza            28-Nov-1983
0000 ;               Move the kstack expansion WSL slots.
0000 ;
0000 ;       V03-023 WMC0003          Wayne Cardoza            13-Oct-1983
0000 ;               Reserve space for 4 extra kstack pages.
0000 ;
0000 ;       V03-022 LJK0251          Lawrence J. Kenah       7-Sep-1983
0000 ;               Add CLI name counted string to CLI data page.
0000 ;
0000 ;       V03-021 LJK0246          Lawrence J. Kenah       24-Aug-1983
0000 ;               Add cell to P1 pointer page that records the size of the
0000 ;               user stack.
0000 ;
0000 ;       V03-020 WMC0002          Wayne Cardoza            22-Aug-1983
0000 ;               Temporarily increase kernel stack to 4 pages.
0000 ;
0000 ;       V03-019 LJK0239          Lawrence J. Kenah       1-Aug-1983
0000 ;               Increase size of exec stack to 16 pages.
0000 ;
0000 ;       V03-018 RAS0159          Ron Schaefer             22-Jul-1983
0000 ;               Add symbol PIO$S_EODSTR to specify the length of the
0000 ;               $EOD string for SYS$INPUT.
0000 ;
0000 ;       V03-017 LJK0208          Lawrence J. Kenah       26-May-1983
0000 ;               Add listheads for image control block lists to image
0000 ;               activator context page. Change name of image activator
0000 ;               work area.
0000 ;
0000 ;       V03-016 RSH0017          R. Scott Hanna           21-May-1983
0000 ;               Change CTL$GL_AUDITCHAN to CTL$GL_NSA_IDTPTR. This cell
0000 ;               contains a pointer to the security auditing impure data
0000 ;               table. The audit channel number is now in the IDT.
0000 ;
0000 ;       V03-015 PCA1016          Paul C. Anagnostopoulos 28-Apr-1983
0000 ;               Add CTL$AG_CLITABLE, two longwords which specify the
0000 ;               virtual address range into which the CLI table is mapped.
0000 ;
0000 ;       V03-014 GAS0122          Gerry Smith              14-Apr-1983
0000 ;               Add CTL$GQ_HELPFLAGS, two longwords of information on
0000 ;               current help settings. Also double the size of the
0000 ;               supervisor stack, to 32 pages.
0000 ;
0000 ;       V03-013 RSH0012          R. Scott Hanna           13-Mar-1983
0000 ;               Add the cell CTL$GL_AUDITCHAN to store the security
```

```
0000   ;                         audit trail journal channel number.
0000   ;
0000   ;          V03-012 RSH0002          R. Scott Hanna          9-Feb-1983
0000   ;                  Add CTL$GL_RDIPTR to support rights database system
0000   ;                  services.
0000   ;
0000   ;          V03-011 KBT0454          Keith B. Thompson       7-Jan-1983
0000   ;                  Make directory cache 2 pages due to long directory names.
0000   ;
0000   ;          V03-010 JWH0138          Jeffrey W. Horn         29-Nov-1982
0000   ;                  Fix problems related with movement within the
0000   ;                  vector page introduced by JWH0118 and DMW4008.
0000   ;
0000   ;          V03-009 JWH0118          Jeffrey W. Horn         19-Nov-1982
0000   ;                  Re-organize RMS Impure Areas.   Add cells needed to
0000   ;                  support P1 allocation routines.  Eliminate the
0000   ;                  RMS Process IO segment and the Process  Allocation
0000   ;                  Region from SHELL, now created in PROCSTRT.
0000   ;
0000   ;          V03-008 DMW4008          DMWalp                  12-Nov-1982
0000   ;                  Add definations for new logical name structures.
0000   ;
0000   ;          V03-007 WMC0001          Wayne Cardoza           19-Oct-1982
0000   ;                  Add support for specifying desired page file.
0000   ;
0000   ;          V03-006 JWH0001          Jeffrey W. Horn         27-Aug-1982
0000   ;                  Add cells needed to support RMS Recovery Units.
0000   ;                  PIO$GL_RULOCK and PIO$GL_NXTIRBSEQ.
0000   ;
0000   ;          V03-005 CDS0001          C Saether               13-Aug-1982
0000   ;                  Define CTL$GL_F11BXQP cell to dispatch to XQP.
0000   ;
0000   ;          V03-004 LJK47269         Lawrence J. Kenah       10-Aug-1982
0000   ;                  Change protection on image activator scratch pages to
0000   ;                  UREW. Remove $PRDEF.
0000   ;
0000   ;          V03-004 KDM46395         Kathleen D. Morse       28-Jun-1982
0000   ;                  Increment the PTWSLELCK byte array for the P1 system
0000   ;                  service vector pages, which are WINDOW PTEs.
0000   ;
0000   ;          V03-002 KDM0002          Kathleen D. Morse       28-Jun-1982
0000   ;                  Added $PRDEF.
0000   ;
0000   ;
```

```
                  0000   216          .SBTTL  DECLARATIONS
                  0000   217
                  0000   218 ;
                  0000   219 ; INCLUDE FILES:
                  0000   220 ;
                  0000   221          $DYNDEF                   ;DYNAMIC DATA STRUCTURE TYPE DEFINITIONS
                  0000   222          $FWADEF                   ;RMS FILE WORK AREA CONSTANTS
                  0000   223          $IAFDEF                   ;IMAGE ACTIVATOR FIXUP VECTOR OFFSETS
                  0000   224          $IMPDEF                   ;RMS INTERNAL STRUCTURE DEFINITION
                  0000   225          $IPLDEF                   ;DEFINE INTERRUPT LEVELS
                  0000   226          $JIBDEF                   ;JOB INFORMATION BLOCK DEFINITIONS
                  0000   227          $NSAIDTDEF                ;SECURITY AUDITING IMPURE DATA TABLE OFFSETS
                  0000   228          $PCBDEF                   ;PROCESS CONTROL BLOCK DEFINITIONS
                  0000   229          $PFLDEF                   ;PAGE FILE DEFINITIONS
                  0000   230          $PFNDEF                   ;PFN DATA BASE DEFINITICNS
                  0000   231          $PHDDEF                   ;PROCESS HEADER DEFINITIONS
                  0000   232          $PQBDEF                   ;PROCESS QUOTA BLOCK DEFINITIONS
                  0000   233          $PRTDEF                   ;DEFINE PROTECTION CODES
                  0000   234          $PSLDEF                   ;DEFINE PSL FIELDS
                  0000   235          $PTEDEF                   ;PAGE TABLE ENTRY DEFINTIONS
                  0000   236          $SECDEF                   ;SECTION DEFINITIONS (GSTE/PSTE)
                  0000   237          $SGNDEF                   ;SYSGEN VALUE DEFINITIONS
                  0000   238          $VADEF                    ;DEFINE VIRTUAL ADDRESS FIELDS
                  0000   239          $WSLDEF                   ;WORKING SET LIST DEFINITIONS
                  0000   240
                  0000   241 ;
                  0000   242 ; EQUATES:
                  0000   243 ;
                  0000   244
        00000000  0000   245 PFILPGCNT=0                            ; INITIALIZE COUNT OF PAGE FILE PAGES
        00000003  0000   246 KSTACK=3                               ; THREE PAGES OF KERNEL STACK
        00000004  0000   247 KSTACK_EX=4                            ; KERNEL STACK EXPANSION PAGES
        00000010  0000   248 ESTACK=16                              ; SIXTEEN PAGES OF EXEC STACK
        00000020  0000   249 SSTACK=32                              ; THIRTY TWO PAGES OF SUPER STACK
        00000008  0000   250 IMGACTBUF=8                            ; EIGHT PAGES FOR IMAGE ACTIVATE BUFFER
                  0000   251                                        ; FOUR FOR TOP LEVEL CALL, FOUR FOR
                  0000   252                                        ; ONE RECURSIVE CALL.
        00000004  0000   253 KRP_COUNT=4                            ; SIZE IN PAGES OF P1 LOOKASIDE LIST
                  0000   254
        00000001  0000   255 DBGPTCNT=1                             ; NUMBER OF DEBUG AREA PAGE TABLES
                  0000   256
                  0000   257          .IF     NDF,LIBSWITCH         ; IF EXECUTIVE, DEFINE THESE GLOBALLY
                  0000   258 SWP$C_DBGPTCNT==DBGPTCNT
                  0000   259 SWP$C_KSTACK==KSTACK
                  0000   260 SWP$C_KSTACK_EX==KSTACK_EX
                  0000   261 CTL$C_KRP_COUNT==KRP_COUNT
                  0000   262 CTL$C_KRP_SIZE==512                    ; EACH KRP PACKET IS 512 BYTES LONG
                  0000   263          .ENDC
                  0000   264
        00000004  0000   265 COMMON=4                               ; FOUR PAGES FOR USER COMMON
                  0000   266                                        ; FOUR PAGES FOR DEC COMMON
                  0000   267
                  0000   268 ;
                  0000   269 ; MACROS:
                  0000   270 ;
                  0000   271
                  0000   272          .MACRO  WSL     SYM,TYPE      ; MAKE VALID WS LIST ENTRY
```

```
0000          .LONG   SYM!WSL$M_VALID!WSL$C_'TYPE      ; SET VALID BIT
0000          .ENDM   WSL
0000
0000          .MACRO  CTL     SYM
0000          .IF     NDF,LIBSWITCH
0000  CTL$'SYM==.-CTLVECPAG+VECTORS
0000          .IFF
0000  CTL$'SYM=.-CTLVECPAG+VECTORS
0000          .ENDC
0000          .ENDM   CTL
0000
0000          .MACRO  PHD     SYM
0000          .=PHD...+PHD$'SYM
0000          .ENDM   PHD
0000
0000
0000          .MACRO  PIO     SYM             ; DEFINE PROCESS I/O SEGMENT SYMBOL
0000  PIO$'SYM==PIOBASE+<.-PIO>               ;
0000          .ENDM   PIO
0000
0000          .MACRO  PIOL    SYM             ; DEFINE PROCESS I/O LOCAL SYMBOL
0000  PIOL'SYM=PIOBASE+<.-PIO>                ;
0000          .ENDM   PIOL                    ;
0000
0000          .MACRO  PIO2    SYM             ; DEFINE PROCESS I/O LOCAL SYMBOL
0000  PIO2'SYM=PIOTBL+<.-TBL>
0000          .ENDM   PIO2
0000
0000  ;
0000  ; P1PTE - GENERATE PTES FOR CONTROL REGION PAGE TABLE
0000  ;
0000
0000          .MACRO  P1PTE   NUM,ACCESS=NA,OWNER=K,STATE,SYM
0000          .IF     NB,SYM
0000          SYM=VA...
0000          .ENDC
0000
0000          .REPT   NUM
0000          .=.-4
0000          .IF     IDN,<STATE>,<VALID>
0000          TMP...=<1@31>
0000          .IFF
0000          TMP...=0
0000          .ENDC
0000
0000          .IF     IDN,<STATE>,<DZRO>
0000          PFILPGCNT=PFILPGCNT+1
0000          .ENDC
0000
0000          .IF     IDN,<STATE>,<PFIL>
0000          TMP...=PTE$M_TYP1
0000          PFILPGCNT=PFILPGCNT+1
0000          .ENDC
0000          .LONG   PTE$C_'ACCESS!PTE$C_'OWNER'OWN!TMP...
0000
0000          .=.-4
0000          VA...=VA...-^X200
```

```
   0000     330          .ENDR
   0000     331          .ENDM    P1PTE
   0000     332
```

```
          0000
          0000         .IF      DF,LIBSWITCH               ; FOR USRVECTOR MODULE,
          0000         .PSECT   $ABS$,ABS                  ; DO NOT GENERATE ANY STORAGE
00000000  0000         .=0
          0000         .IFF                                ; ELSE,
          0000
          0000         .PSECT   AEXENONPAGED
          0000  SWP$GL_SHELIO::                            ; I/O PAGE COUNT FOR SHELL
          0000         .LONG    SWP$C_SHELLSIZ             ;
          0000  ;
          0000  ;      WARNING: SWP$C_SHLP1PT MUST BE MANUALLY COMPUTED SINCE IT IS USED
          0000  ;               TO ALLOCATE THE SPACE FOR P1 PAGE TABLE ENTRIES.  IF UPDATES
          0000  ;               ARE MADE THAT INTRODUCE MORE P1 PAGE TABLE ENTRIES, THEN
          0000  ;               VERIFY THAT THEY STILL FIT IN THE NUMBER OF PAGES PROVIDED
          0000  ;
          0000  SWP$C_SHLP1PT==2                           ; COUNT OF BASIC P1 PAGE TABLES FOR SHELL
          0000         .PSECT   YYSHELLPAGED,PAGE          ; SHELL AT END OF PAGEABLE EXEC
          0000  SHELL:                                     ; BASE OF SHELL SWAP IMAGE
          0000  SWP$GL_SHELLBAS::                          ;
          0000         .PAGE
          0000         .SBTTL   SHELL PROCESS HEADER
          0000  ;
          0000  ;      PROCESS HEADER FOR SHELL PROCESS
          0000  ;
          0000
          0000  PHD...=.                                   ; BASE OF PROCESS HEADER
          0000         .BLKB    PHD$C_LENGTH               ; RESERVE SPACE FOR HEADER
          0000
          0000  ;
          0000  ;      WORKING SET LIST
          0000  ;
          0000  ; PAGES DESCRIBED IN THE WORKING SET LIST MUST OCCUR IN THE SAME ORDER
          0000  ; WITHIN THE PROCESS BODY.
          0000  ;
          0000         .ALIGN  LONG                        ; AT LONGWORD RESOLUTION
          0000  SWP$C_KSTACK_EX_WSL==<.-PHD...>@-2
          0000         .BLKL    KSTACK_EX                  ; EXTRA SLOTS FOR KERNEL STACK EXPANSION
          0000  WSL...=.                                   ; BASE OF WORKING SET LIST
          0000  SWP$C_KSTACK_WSL==<.-PHD...>@-2
          0000         WSL      <<KSPINI-<1*512>>!WSL$M_WSLOCK>,PROCESS; KERNEL STACK PAGE 1
          0000         WSL      <<KSPINI-<2*512>>!WSL$M_WSLOCK>,PROCESS; KERNEL STACK PAGE 2
          0000         WSL      <<KSPINI-<3*512>>!WSL$M_WSLOCK>,PROCESS; KERNEL STACK PAGE 3
          0000  NOT_KSTACK_WSL=.                           ; REST OF WSL IS NOT KERNEL STACK
          0000         WSL      <VECTORS!WSL$M_WSLOCK>,PROCESS  ; VECTOR PAGE
          0000  DYNWSL=.                                   ; START OF DYNAMIC WORKING SET
          0000         WSL      <PIOBASE!WSL$M_MODIFY>,PROCESS  ; PROCESS I/O SEGMENT BASE PAGE 1
          0000
          0000  FWSL...=.                                  ; FREE WORKING SET LIST
          0000  NDYN=<FWSL...-DYNWSL>@-2                   ; NUMBER OF DYNAMIC ENTRIES IN SHELL
          0000  SHUFFLE_COUNT=<FWSL...-NOT_KSTACK_WSL>@-2 ; AMOUNT OF SHUFFLING BY SHELINIT
          0000  SWP$C_NDYN==NDYN                           ; GLOBAL NAME FOR NUMBER OF DYNAMIC PAGES
          0000         .ALIGN  PAGE                        ; GET TO END OF PAGE
          0000  SAV...=.                                   ;
          0000
          0000  TMP...=<DYNWSL-PHD...>@-2                  ; BASE OF FLUID WORKING SET LIST
          0000
          0000         PHD      W_WSLOCK                   ; POINTER TO START OF LOCKED PAGES
```

```
     0000    391          .WORD   TMP...                       ;
     0000    392
     0000    393          PHD     W_WSDYN                      ; POINTER TO START OF DYNAMIC PAGES
     0000    394          .WORD   TMP...                       ;
     0000    395
     0000    396          PHD     W_WSLIST                     ;
     0000    397          .WORD   <WSL...-PHD...>@-2           ; START OF WORKING SET LIST
     0000    398          PHD     W_WSNEXT                     ; NEXT WORKING SET ENTRY
     0000    399          .WORD   <TMP...+NDYN-1>              ;
     0000    400
     0000    401          PHD     L_FREP1VA                    ; VA OF FIRST FREE PAGE IN P1 SPACE
     0000    402          .LONG   VA...-512                    ;
     0000    403
     0000    404          PHD     Q_PRIVMSK                    ; ENABLE ALL PRIVILEGES
     0000    405          .LONG   -1,-1                        ;
     0000    406
     0000    407          PHD     W_QUANT                      ; QUANTUM OF ONE SECOND
     0000    408          .WORD   -100                         ;
     0000    409
     0000    410          PHD     L_KSP                        ; STACK POINTERS
     0000    411          .LONG   KSPINI                       ; KERNEL STACK POINTER
     0000    412          .LONG   ESPINI                       ; EXECUTIVE STACK POINTER
     0000    413          .LONG   SSPINI                       ; SUPER STACK POINTER
     0000    414                                               ; NO INITIAL USER STACK
     0000    415
     0000    416          PHD     L_PC                         ; INITIAL PROGRAM COUNTER
     0000    417          .LONG   EXE$PROCSTRT                 ; START PROCESS
     0000    418
     0000    419          PHD     L_PSL                        ; PROGRAM STATUS LONGWORD
     0000    420          .LONG   IPL$_ASTDEL@PSL$V_IPL        ; MUST RUN AT ASTDEL
     0000    421          PHD     L_P0BR                       ; P0 BASE REGISTER
     0000    422          .LONG   0                            ; RELATIVE OFFSET
     0000    423
     0000    424          PHD     L_P0LRASTL                   ; NO AST PENDING, P0LR = 0
     0000    425          .LONG   <4@PHD$V_ASTLVL>             ;
     0000    426
     0000    427          PHD     L_P1BR                       ; SVA OF P1 PAGE TABLE BASE
     0000    428          .LONG   -<1@23>                      ;
     0000    429
     0000    430          PHD     L_P1LR                       ; P1 LENGTH REGISTER
     0000    431          .LONG   <1@21>-P1PTLEN               ;
     0000    432
     0000    433          PHD     L_FREPTECNT                  ; FREE PTE'S BETWEEN P0 AND P1 PT
     0000    434          .LONG   -<1+<<1@31-VA...>@-9>>       ; (NUMBER OF AVAILABLE LONGWORDS
     0000    435
     0000    436          PHD     W_PRCLM                      ; SUB PROCESS LIMIT
     0000    437          .WORD   4                            ; ALLOW FOUR
     0000    438
     0000    439          PHD     W_WSFLUID                    ; GUARANTEED FLUID PAGE COUNT
     0000    440          .WORD   10                           ; GUESS AT TEN FOR NOW (WAG)
     0000    441
     0000    442          PHD     B_DFPFC                      ; DEFAULT PAGE FAULT CLUSTER
     0000    443          .BYTE   16                           ;
     0000    444
     0000    445          PHD     B_CPUMODE                    ; ACCESS MODE FIELD FOR CPU TIME LIMIT
     0000    446          .BYTE   PSL$C_USER                   ;  EXPIRATION AST STARTS WITH USER MODE
     0000    447
```

```
                  0000           .=SAV...                         ; RESTORE LOCATION COUNTER
                  0000           .PAGE
                  0000           .SBTTL  SHELL PROCESS PAGE TABLES
                  0000   ;
                  0000   ;
                  0000   ;       SHELL PROCESS PAGE TABLES
                  0000   ;
                  0000   ;
                  0000
                  0000   P1PTBAS=.                                ;
                  0000
                  0000           TMP=<SWP$C_SHLP1PT*512>          ; ALLOCATE SPACE FOR PAGE TABLES
                  0000           .BLKB   TMP                      ; ALLOCATE SPACE FOR PAGE TABLES
                  0000   SAV...=.                                 ; REMEMBER BASE OF PAGE TABLES
                  0000
                  0000           .IFTF                            ; ASSEMBLE FOLLOWING FOR BOTH
                  0000                                            ; SHELL AND USRVECTOR MODULES:
                  0000
                  0000   ;
                  0000   ;       INITIALIZE VIRTUAL ADDRESS BASE
                  0000   ;
        80000000  0000           VA...=1@31                       ;
                  0000
                  0000   ;       P1PTE   128*DBGPTCNT             ; DEBUGGER CONTEXT AREA
        7FFF0000  0000           VA...=VA...-<DBGPTCNT*128*512>   ; ACCOUNT FOR IT IN VA...
        7FFF0000  0000   DBGAREA=VA...                            ; ADDRESS OF DEBUG AREA
                  0000           P1PTE   1,URKW,K,PFIL            ; VECTOR PAGE
        7FFEFE00  FFFC   VECTORS=VA...                            ;
                  FFFC
                  FFFC           .IFF                             ; ASSEMBLE REST ONLY FOR SHELL MODULE:
                  FFFC
                  FFFC   CTL$GL_VECTORS==VECTORS                  ; BASE OF VECTOR PAGE
                  FFFC           P1PTE   16                       ; PAGES FOR SYSTEM SERVICE VECTORS
                  FFFC   P1SYSVECTORS==VA...
                  FFFC           P1PTE   SSTACK,URSW,S,DZRO,SSPINI        ; SUPERVISOR STACK
                  FFFC           P1PTE   ESTACK,SREW,E,DZRO,ESPINI        ; EXECUTIVE STACK
                  FFFC           P1PTE   KSTACK,SRKW,K,PFIL,KSPINI        ; KERNEL STACK
                  FFFC   CTL$GL_KSTKBAS==VA...                    ; MAXIMUM TOP OF KERNEL STACK
                  FFFC           P1PTE   KSTACK_EX,NA,K,PFIL              ; EXPANSION FOR  KERNEL STACK
                  FFFC   CTL$GL_KSTKBASEXP==VA...                 ; MAXIMUM TOP OF EXPANS!ON KERNEL STACK
                  FFFC   CTL$GL_KSPINI==KSPINI                    ; INITIAL KERNEL STACK
                  FFFC           P1PTE   1                        ; NULL STOPPER PAGE
                  FFFC           P1PTE   KRP_COUNT,URKW,K,DZRO    ; P1 LOOKASIDE LIST FOR KERNEL MODE
                  FFFC   CTL$GL_KRP==VA...                        ; ADDRESS USED BY PROCSTRT TO FIND IT
                  FFFC           P1PTE   1,URSW,E,DZRO            ; IMAGE HEADER BUFFER
                  FFFC   MMG$IMGHDRBUF==VA...                     ; ADDRESS OF IMAGE HEADER BUFFER
                  FFFC           P1PTE   2,UREW,K,DZRO            ; VECTORS FOR USER SYS SRV & ERR MSGS
                  FFFC   CTL$A_DISPVEC==VA...                     ; ADR OF VECTOR PAGES FOR SYS SRV/ERRMSG
                  FFFC           P1PTE   4,UW,U,DZRO              ; DEBUGGER CONTEXT
                  FFFC           P1PTE   IMGACTBUF,UREW,E,DZRO    ; IMAGE ACTIVATOR SCRATCH PAGES (8)
                  FFFC   IAC$AL_IMGACTBUF==VA...                  ; ADR OF IMAGE ACTIVATOR SCRATCH PAGES
                  FFFC   CLIDATAEND=VA...
                  FFFC           P1PTE   12,URSW,S,DZRO           ; COMMMAND INTERPRETER DATA PAGES
                  FFFC   CLIDATAPAG=VA...                         ; BASE ADDRESS OF CLI DATA PAGE
                  FFFC           P1PTE   1,UREW,E,DZRO            ; IMAGE ACTIVATOR CONTEXT
                  FFFC   IMGACTCTX=VA...
                  FFFC           P1PTE   NSA$K_IDT_PAGES,KW,K,DZRO ; SECURITY AUDITING IMPURE DATA TABLE
```

```
FFFC     NSA$T_ID1==VA...
FFFC             P1PTE   2                           ; UNUSED PAGES
FFFC     ;
FFFC     ;       THE FOLLOWING PAGE IS A USER-MODE WRITABLE CONTEXT PAGE
FFFC     ;       FOR MISC. USER-MODE PACKAGES THAT ARE MAINTAINED BY VMS.
FFFC     ;       (SEE DETAILED ALLOCATION OF PAGE BELOW)
FFFC     ;
FFFC             P1PTE   1,UW,K,DZRO                 ; GLOBAL USER-MODE OWN STORAGE PAGE
FFFC     UWVECPAG=VA...
FFFC     ;
FFFC     ;       THE FOLLOWING PAGE IS A WRITABLE CONTEXT PAGE FOR
FFFC     ;       THE USE OF THE COMPATIBILITY MODE EMULATOR AND EXCEPTION
FFFC     ;
FFFC             P1PTE   2,UW,K,DZRO                 ; WRITABLE PAGES FOR COMPATIBILITY MODE
FFFC     CTL$AG_CMEDATA==VA...                       ;
FFFC     ;
FFFC     ;       THE FOLLOWING PAGES ARE FOR USE BY BASIC/BASIC-PLUS TO PROVIDE
FFFC     ;       THE ''CORE COMMON'' REQUIRED TO PASS DATA WHEN CHAINING FROM IMAGE
FFFC     ;       TO IMAGE.  AN IDENTICAL AREA, NEGATIVELY DISPLACED HAS BEEN ALLOCATED
FFFC     ;       FOR USERS AND CSS.
FFFC     ;
FFFC             P1PTE   COMMON,UW,K,DZRO            ; DEC ''CORE COMMON'' PAGES
FFFC     CTL$A_COMMON==VA...                         ; BASE ADDRESS
FFFC             P1PTE   COMMON,UW,K,DZRO            ; USER ''CORE COMMON'' PAGES
FFFC             P1PTE   1,UREW,E,DZRO               ; PROCESS IFB/IRB TABLES
FFFC     PIOTBL=VA...
FFFC             P1PTE   2,UREW,E,DZRO               ; RMS DIRECTORY CACHE PAGES
FFFC     PIO$A_DIRCACHE==VA...
FFFC             P1PTE   1,UREW,E,DZRO               ; RMS TRACEPOINT PAGE
FFFC     PIO$A_TRACE==VA...
FFFC             P1PTE   1,UREW,E,DZRO               ; EXTENSION TO RMS POINTER PAGE
FFFC             P1PTE   1,UREW,E,PFIL               ; RMS POINTER PAGE
FFFC     PIOBASE=VA...                               ;
FFFC     USPINI=VA...                                ; USER STACK AREA
FFFC
FFFC     END...=.                                    ; END OF CONTROL REGION
FFFC
FFFC     P1PTLEN=<1@31-VA...>@-9                     ; ENTRIES IN P1PT
FFFC
FFFC     SWP$C_SHLFPTE==<<.-P1PTBAS>@-2> ; FREE P1PT IN BASIC P1 PAGE TABLES
FFFC     ;
FFFC     ; RESTORE LOCATION POINTER
FFFC     ;
FFFC             .=SAV...                            ; FINISHED WITH PAGE TABLES
FFFC             PFILPGCNT=PFILPGCNT-KSTACK-1        ; NO PAGE FILE FOR KERNEL STACK OR VECTORS
FFFC             SWP$C_SHELLPFIL==PFILPGCNT          ; DEFINE GLOBAL VALUE FOR SHELL PAGE FILE
FFFC                                                 ; REQUIREMENT
FFFC             .PAGE
FFFC             .SBTTL  BODY OF SHELL PROCESS
FFFC     ;
FFFC     ;       BODY OF SHELL PROCESS
FFFC     ;
FFFC             .=SAV...                            ; POSITION TO END OF HEADER
FFFC     ;----------------------------------------------------------------------
FFFC     ;
FFFC     ;       VECTOR PAGE
FFFC     ;               ***** NOTE: The cells in this page must not move.  There are
```

```
              FFFC  ;                 *****      facilities which locate cells via offsets from
              FFFC  ;                 *****      the top of this page.  If you delete a cell
              FFFC  ;                 *****      subsitute a .LONG 0 and comment it as SPARE.  If
              FFFC  ;                 *****      you add a cell either replace a SPARE cell or
              FFFC  ;                 *****      add it to the end.
              FFFC  ;
              FFFC  ;-------------------------------------------------------------------------------
              FFFC  SWP$AL_PTRPAG==.-SHELL                          ; ADDRESS IN SWAPPER MAP
              FFFC
              FFFC          .IFT                                    ; IF USRVECTOR MODULE,
    7FFEFE00  FFFC          .=VECTORS                               ; SHOW ACTUAL ADDRESSES IN LISTING
              FE00
              FE00          .IFTF                                   ; ASSEMBLE FOLLOWING FOR BOTH
              FE00                                                  ; SHELL AND USRVECTOR MODULES:
              FE00
    7FFEFE00  FE00  CTLVECPAG=.                                     ; BASE OF VECTOR PAGE
              FE00          CTL     GW_NMIOCH                       ; NUMBER OF CHANNELS
        0000  FE00          .WORD   0                               ; FILLED IN IN PROCSTRT
              FE02
              FE02          CTL     GW_CHINDX                       ; MAXIMUM CHANNEL INDEX
        0000  FE02          .WORD   0                               ; FILLED IN IN PROCSTRT
              FE04
              FE04          CTL     GL_LNMHASH                      ; process Logical NaMe HASH table
    00000000  FE04          .LONG   0                               ;  pointer
              FE08          CTL     GL_LNMDIRECT                    ; process Logical NaMe DIRECTory
    00000000  FE08          .LONG   0                               ;  pointer
              FE0C
              FE0C  ;
              FE0C  ;       ARRAY OF STACK POINTERS
              FE0C  ;
              FE0C  ;       THE -1 ENTRY OF CTL$AL_STACK IS USED TO DETERMINE THE MAXIMUM
              FE0C  ;       EXTENT OF THE KERNEL STACK.
              FE0C  ;
   00000000'  FE0C          .LONG   CTL$GL_KSTKBAS                  ; MAXIMUM TOP OF KERNEL STACK
              FE10          CTL     AL_STACK                        ; STACK POINTER RESTART ARRAY
   00000000'  FE10          .LONG   KSPINI                          ; KERNEL STACK POINTER
   00000000'  FE14          .LONG   ESPINI                          ; EXEC STACK POINTER
   00000000'  FE18          .LONG   SSPINI                          ; SUPERVISOR STACK POINTER
   00000000'  FE1C          .LONG   USPINI                          ; USER STACK POINTER
              FE20
              FE20  ;
              FE20  ; LOGICAL NAME TABLE TRANSLATION CACHE QUEUE HEADER
              FE20  ;
              FE20
              FE20          CTL     GQ_LNMTBLCACHE                  ; QUEUE HEADER FOR LOGICAL NAME TABLE
    7FFEFE20  FE20          .LONG   CTL$GQ_LNMTBLCACHE              ;  TRANSLATION CACHE (FORWARD LINK)
    7FFEFE20  FE24          .LONG   CTL$GQ_LNMTBLCACHE              ;                    (BACK LINK)
              FE28
              FF28  ;
              FE28  ; THE FOLLOWING THREE VECTORS MUST BE ADJACENT
              FE28  ;
              FE28
              FE28          CTL     GL_CMSUPP                       ; SUPERVISOR CHANGE MODE DISPATCHER
    00000000  FE28          .LONG   0                               ; NULL
              FE2C
              FE2C          CTL     GL_CMUSER                       ; USER CHANGE MODE DISPATCHER
    00000000  FE2C          .LONG   0                               ; NULL
```

```
                           FE30   619
                           FE30   620           CTL     GL_CMHANDLR             ; COMPATIBILITY MODE HANDLER
                  00000000 FE30   621           .LONG   0
                           FE34   622
                           FE34   623           CTL     AQ_EXCVEC               ; PRIMARY/SECONDARY EXCEPTION VECTORS
                  00000000 FE34   624           .LONG   0                       ; KERNEL MODE PRIMARY
                  00000000 FE38   625           .LONG   0                       ; KERNEL MODE SECONDARY
                  00000000 FE3C   626           .LONG   0                       ; EXEC MODE PRIMARY
                  00000000 FE40   627           .LONG   0                       ; EXEC MODE SECONDARY
                  00000000 FE44   628           .LONG   0                       ; SUPERVISOR MODE PRIMARY
                  00000000 FE48   629           .LONG   0                       ; SUPERVISOR MODE SECONDARY
                  00000000 FE4C   630           .LONG   0                       ; USER MODE PRIMARY
                  00000000 FE50   631           .LONG   0                       ; USER MODE SECONDARY
                           FE54   632
                           FE54   633           CTL     GL_THEXEC               ; EXECUTIVE TERMINATION HANDLER
                  00000000 FE54   634           .LONG   0                       ; NULL
                           FE58   635
                           FE58   636           CTL     GL_THSUPR               ; SUPERVISOR TERMINATION HANDLER
                  00000000 FE58   637           .LONG   0                       ; NULL
                           FE5C   638
                           FE5C   639 ;         CTL     GL_THUSER               ; USER TERMINATION HANDLER
                  00000000 FE5C   640           .LONG   0                       ; NULL
                           FE60   641
                           FE60   642 ;         FORMER LOCATION OF COMPATIBILITY MODE CONTEXT
                           FE60   643           CTL     GQ_COMMON               ; CORE COMMON DESCRIPTOR
                  00000800 FE60   644           .LONG   512*COMMON              ; SIZE IN BYTES
                  00000000' FE64  645           .LONG   CTL$A_COMMON            ; AND ADDRESS
                           FE68   646
                           FE68   647           CTL     GL_GETMSG               ; PER-PROCESS VECTOR TO USER MESSAGE
                  00000000 FE68   648           .LONG   0                       ; DISPATCHER
                           FE6C   649
                           FE6C   650           CTL     AL_STACKLIM             ; STACK LIMIT ARRAY (INDEXED BY MODE)
                  00000000' FE6C  651           .LONG   CTL$GL_KSTKBAS          ; KERNEL STACK LO (TOP) LIMIT
                  00000000' FE70  652           .LONG   KSPINI                  ; EXEC   STACK LO (TOP) LIMIT
                  00000000' FE74  653           .LONG   ESPINI                  ; SUPER  STACK LO (TOP) LIMIT
                  00000000 FE78   654           .LONG   0                       ; USER STACK (NOT CHECKED)
                           FE7C   655
                           FE7C   656           CTL     GL_CTLBASVA             ; BASE CONTROL REGION ADDRESS
                  00000000 FE7C   657           .LONG   0                       ; FILLED IN BY PROCSTRT
                           FE80   658
                           FE80   659           CTL     GL_IMGHDRBF             ; ADDRESS OF IMAGE ACTIVATOR'S IMAGE
                           FE80   660                                           ; HEADER BUFFER, IF IMAGE IS ACTIVE;
                  00000000 FE80   661           .LONG   0                       ; 0 IF NO IMAGE ACTIVE
                           FE84   662
                           FE84   663           CTL     GL_IMGLSTPTR            ; ADDRESS OF ICB LIST (FOR DEBUGGER)
                  00000000' FE84  664           .LONG   IAC$GL_IMAGE_LIST
                           FE88   665
                           FE88   666           CTL     GL_PHD                  ; ADDRESS OF PHD WINDOW
                  00000000 FE88   667           .LONG   0                       ; INIT BY INSWAP
                           FE8C   668
                           FE8C   669           CTL     GQ_ALLOCREG             ; HEAD OF PROCESS ALLOCATION REGION POOL
         00000000 00000000 FE8C   670           .LONG   0,0                     ;       (Filled in by PROCSTRT)
                           FE94   671
                           FE94   672           CTL     GQ_MOUNTLST             ; MOUNTED DEVICE LIST
                  7FFEFE94 FE94   673           .LONG   CTL$GQ_MOUNTLST         ; FORWARD LINK
                  7FFEFE94 FE98   674           .LONG   CTL$GQ_MOUNTLST         ; BACKWARD LINK ( LIST EMPTY )
                           FE9C   675
```

```
                   FE9C  ;**********************************************************************
                   FE9C  ; NOTE: The order of the accounting data between CTL$T_USERNAME and
                   FE9C  ;       CTL$T_REMOTEID should be preserved!
                   FE9C  ;**********************************************************************
                   FE9C
                   FE9C             CTL     T_USERNAME              ; USERNAME
          7FFEFEA8 FE9C             .BLKB   JIB$S_USERNAME          ;  Filled in by PROCSTRT
                   FEA8
                   FEA8             CTL     T_ACCOUNT               ; ACCOUNT NAME
          7FFEFEB0 FEA8             .BLKB   JIB$S_ACCOUNT           ;  Filled in by PROCSTRT
                   FEB0
                   FEB0  ASSUME  <CTL$T_ACCOUNT - CTL$T_USERNAME> EQ <JIB$T_ACCOUNT - JIB$T_USERNAME>
                   FEB0
                   FEB0             CTL     GQ_LOGIN                ; SYSTEM TIME AT PROCESS CREATION
 00000000 00000000 FEB0             .LONG   0,0                     ;
                   FEB8
                   FEB8             CTL     GL_FINALSTS             ; FINAL EXIT STATUS FOR PROCESS
          00000000 FEB8             .LONG   0                       ;
                   FEBC
                   FEBC             CTL     GL_WSPEAK               ; PEAK WORKING SET SIZE
          00000000 FEBC             .LONG   0                       ;
                   FEC0
                   FEC0             CTL     GL_VIRTPEAK             ; PEAK VIRTUAL SIZE
          00000000 FEC0             .LONG   0                       ;
                   FEC4
                   FEC4             CTL     GL_VOLUMES              ; COUNT OF VOLUMES MOUNTED
          00000000 FEC4             .LONG   0                       ;
                   FEC8
                   FEC8             CTL     GQ_ISTART               ; IMAGE ACTIVATION TIME
 00000000 00000000 FEC8             .LONG   0,0
                   FED0
                   FED0             CTL     GL_ICPUTIM              ; INITIAL IMAGE CPU TIME
          00000000 FED0             .LONG   0
                   FED4
                   FED4             CTL     GL_IFAULTS              ; INITIAL IMAGE FAULT COUNT
          00000000 FED4             .LONG   0
                   FED8
                   FED8             CTL     GL_IFAULTIO             ; INITIAL IMAGE FAULT I/O COUNT
          00000000 FED8             .LONG   0
                   FEDC
                   FEDC             CTL     GL_IWSPEAK              ; IMAGE WORKING SET PEAK
          00000000 FEDC             .LONG   0
                   FEE0
                   FEE0             CTL     GL_IPAGEFL              ; IMAGE PAGE FILE PEAK USAGE
          00000000 FEE0             .LONG   0
                   FEE4
                   FEE4             CTL     GL_IDIOCNT              ; INITIAL IMAGE DIRECT I/O COUNT
          00000000 FEE4             .LONG   0
                   FEE8
                   FEE8             CTL     GL_IBIOCNT              ; INITIAL IMAGE BUFFERED I/O COUNT
          00000000 FEE8             .LONG   0
                   FEEC
                   FEEC             CTL     GL_IVOLUMES             ; INITIAL IMAGE VOLUME MOUNT COUNT
          00000000 FEEC             .LONG   0
                   FEF0
                   FEF0             CTL     T_NODEADDR              ; REMOTE NODE ADDRESS (BINARY)
                00 FEF0             .BYTE   0                               ; BYTE COUNT BYTE
```

```
         7FFEFEF7  FEF1         .BLKB   6                       ; 6 BYTES MAX
                   FEF7
                   FEF7         CTL     T_NODENAME              ; REMOTE NODE NAME (ASCII)
               00  FEF7         .BYTE   0                               ; BYTE COUNT BYTE
         7FFEFEFE  FEF8         .BLKB   6                               ; 6 CHARACTERS MAX
                   FEFE
                   FEFE         CTL     T_REMOTEID              ; REMOTE ID
               00  FEFE         .BYTE   0                               ; BYTE COUNT BYTE
         7FFEFF0F  FEFF         .BLKB   16                              ; 16 CHARACTERS MAX
                   FF0F
         7FFEFF10  FF0F         .BLKB   1                       ; LONGWORD ALIGN SECTION
                   FF10
                   FF10   ;*********************************
                   FF10   ; End of adjacent accounting data
                   FF10   ;*********************************
                   FF10
                   FF10         CTL     GQ_PROCPRIV             ; PROCESS PRIVILEGE MASK
FFFFFFFF FFFFFFFF  FF10         .LONG   -1,-1                   ;  ALL PRIVILEGES PERMITTED
                   FF18
                   FF18         CTL     GL_USRCHMK              ; PER-PROCESS VECTOR TO USER CHANGE MODE
         00000000  FF18         .LONG   0                       ; TO KERNEL HANDLER
                   FF1C
                   FF1C         CTL     GL_USRCHME              ; PER-PROCESS VECTOR TO USER CHANGE MODE
         00000000  FF1C         .LONG   0                       ; TO EXECUTIVE HANDLER
                   FF20
                   FF20         CTL     GL_POWERAST             ; POWER FAIL AST ADDRESS
         00000000  FF20         .LONG   0                       ;
                   FF24         CTL     GB_PWRMODE              ; ACCESS MODE FOR POWER FAIL AST
               00  FF24         .BYTE   0                       ;
                   FF25
                   FF25         CTL     GB_SSFILTER             ; SYS SERV INHIBIT FILTER MASK
               00  FF25         .BYTE   0
         7FFEFF28  FF26         .BLKB   2                       ; SPARE
                   FF28
```

```
                  FF28
                  FF28  ;
                  FF28  ;    ARRAY OF FINAL EXCEPTION VECTORS - ONE PER MODE
                  FF28  ;
                  FF28  ;    THESE VECTORS ARE USED TO LOCATE AN EXCEPTION HANDLER WHEN ALL
                  FF28  ;    OTHER METHODS FAIL.  PERMITS DEBUGGERS TO RECEIVE CONTROL EVEN
                  FF28  ;    WHEN THE STACK IS CLOBBERED.
                  FF28  ;
                  FF28       CTL  AL_FINALEXC    ; FINAL EXCEPTION HANDLER ARRAY
        00000000' FF28       .LONG EXE$EXCPTN    ; KERNEL MODE
        00000000' FF2C       .LONG EXE$EXCPTNE   ; EXECUTIVE MODE
         00000000 FF30       .LONG 0             ; SUPERVISOR MODE
         00000000 FF34       .LONG 0             ; USER MODE
                  FF38  ;
                  FF38  ;    POINTER TO BASE OF CHANNELS
                  FF38  ;
                  FF38       CTL  GL_CCBBASE     ; BASE OF I/O CHANNELS
         00000000 FF38       .LONG 0             ; FILLED IN IN PROCSTRT
                  FF3C       CTL  GQ_DBGAREA     ; DEBUG AREA
         00010000 FF3C       .LONG 64*1024       ; SIZE
         7FFF0000 FF40       .LONG DBGAREA       ; ADDRESS
                  FF44  ;
                  FF44  ; POINTER TO RMS IMAGE
                  FF44  ;
                  FF44       CTL  GL_RMSBASE
         00000000 FF44       .LONG 0             ; POINT TO RMS IN SYSTEM SPACE
                  FF48  ;
                  FF48  ;    PROCESS PERMANENT MESSAGE SECTION POINTER AND DEFAULT DISPLAY FLAGS
                  FF48  ;
                  FF48       CTL  GL_PPMSG       ; ADDRESS OF PROCESS PERM. MSG SECTION
00000000 00000000 FF48       .QUAD 0             ; STARTING/ENDING ADDRESS OF SECTION
                  FF50
                  FF50       CTL  GB_MSGMASK     ; DEFAULT MESSAGE DISPLAY FLAGS
               0F FF50       .BYTE 15            ; DEFAULT: FAC,SEV,IDENT AND TEXT (ALL)
                  FF51
                  FF51       CTL  GB_DEFLANG     ; DEFAULT MESSAGE LANGUAGE
               00 FF51       .BYTE 0             ; (CURRENTLY UNUSED)
                  FF52
                  FF52       CTL  GW_PPMSGCHN    ; CHANNEL TO PROCESS PERM. MESSAGE
             0000 FF52       .WORD 0             ; SECTION (MAPPED IN CTL$GL_PPMSG)
                  FF54
                  FF54       CTL  GL_USRUNDWN    ; PER-PROCESS VECTOR TO USER RUNDOWN
         00000000 FF54       .LONG 0             ;  SERVICE
                  FF58
                  FF58       CTL  GL_PCB         ; ADDRESS OF PROCESS CONTROL BLOCK
         00000000 FF58       .LONG 0             ; INIT BY PROCSTRT
                  FF5C
                  FF5C       CTL  GL_RUF         ; POINTER TO RECOVERY UNIT BLOCKS
         00000000 FF5C       .LONG 0
                  FF60
                  FF60       CTL  GL_SITESPEC    ; SITE-SPECIFIC PER-PROCESS CELL
         00000000 FF60       .LONG 0
                  FF64
                  FF64       CTL  GL_KNOWNFIL    ; PROCESS KNOWN FILE LIST POINTER
         00000000 FF64       .LONG 0
                  FF68
                  FF68       CTL  AL_IPASTVEC    ; VECTOR OF IPAST ADDRESSES
```

```
00000000 00000000 00000000 00000000  FF68           .LONG   0,0,0,0,0,0,0,0
00000000 00000000 00000000 00000000  FF78
                                     FF88
                                     FF88           CTL     GL_CMCNTX               ; ADDRESS OF AME CONTEXT PAGE
                           00000000' FF88           .LONG   CTL$AL_CMCNTX
                                     FF8C
                                     FF8C           CTL     GL_IAFLNKPTR            ; ADDRESS OF IAF LIST (FOR DEBUGGER)
                           00000000' FF8C           .LONG   CTL$GL_IAFLINK
                                     FF90
                                     FF90           CTL     GL_F11BXQP              ; ADDRESS OF F11B XQP QUEUE AND
                           00000000  FF90           .LONG   0                       ; DISPATCH VECTORS
                                     FF94
                                     FF94           CTL     GQ_P0ALLOC              ; HEADER OF P0 EXTENTION TO PROCESS
                  00000000 00000000  FF94           .LONG   0,0                     ; ALLOCATION REGION
                                     FF9C
                                     FF9C           CTL     GL_PRCALLCNT            ; COUNT OF BYTES OF PROCESS ALLOCATION
                           00000000  FF9C           .LONG   0                       ; REGION USABLE BY IMAGE REQUESTS.
                                     FFA0
                                     FFA0           CTL     GL_RDIPTR               ; POINTER TO RIGHTS DATABASE IDENTIFIER (RDI
                           00000000  FFA0           .LONG   0
                                     FFA4
                                     FFA4           CTL     GL_LNMDIRSEQ            ; SEQUENCE NUMBER FOR CACHE OF LOGICAL
                           00000000  FFA4           .LONG   0                       ;  NAME TABLE TRANSLATIONS
                                     FFA8
                                     FFA8           CTL     GQ_HELPFLAGS            ; HELP FLAGS, ONE LONGWORD FOR USE
                  00000000 00000000  FFA8           .LONG   0,0                     ; NOW, ONE FOR LATER
                                     FFB0
                                     FFB0           CTL     GQ_TERMCHAR             ; HOME FOR TERMINAL CHARACTERISTICS
                  00000000 00000000  FFB0           .LONG   0 , 0                   ; NEED A QUADWORD
                                     FFB8
                                     FFB8           CTL             GL_KRPFL        ; P1 POOL LOOKASIDE LIST FORWARD LINK
                           7FFEFFB8  FFB8           .LONG           CTL$GL_KRPFL
                                     FFBC           CTL             GL_KRPBL        ; P1 POOL LOOKASIDE LIST BACKWARD LINK
                           7FFEFFB8  FFBC           .LONG           CTL$GL_KRPFL
                                     FFC0
                                     FFC0           CTL             GL_CREPRC_FLAGS ; $CREPRC FLAGS
                           00000000  FFC0           .LONG   0                       ;
                                     FFC4
                                     FFC4           CTL     GL_THCOUNT              ; COUNT OF TERMINATION HANDLERS FOR
                           00000000  FFC4           .LONG   0                       ;  EXEC MODE,
                           00000000  FFC8           .LONG   0                       ;  SUPERVISOR MODE, AND
                           00000000  FFCC           .LONG   0                       ;  USER MODE
                                     FFD0
                                     FFD0   ;*************************************************************************
                                     FFD0   ;
                                     FFD0   ;       END OF VECTOR PAGE
                                     FFD0   ;
                                     FFD0   ;*************************************************************************
                           7FFEFFD0  FFD0   CTLVECEND=.
                           FFFFFFD0  FFD0           .IF GREATER     <CTLVECEND-CTLVECPAG>-512
                                     FFD0           .ERROR          ; *** VECTOR PAGE NOW LARGER THAN A PAGE ***
                                     FFD0           .ENDC
                                     FFD0
                                     FFD0           .IFF                            ; ASSEMBLE REST ONLY FOR SHELL MODULE:
                                     FFD0
                                     FFD0           .PAGE
                                     FFD0           .SBTTL  PROCESS I/O SEGMENT
```

```
FFD0  ;--------------------------------------------------------------------------------
FFD0  ;
FFD0  ;       PROCESS I/O SEGMENT INITIAL CONTENT
FFD0  ;
FFD0  ;--------------------------------------------------------------------------------
FFD0
FFD0          .ALIGN  PAGE                          ; PAGE BOUNDARY
FFD0  PIO:                                          ; BASE OF PROCESS I/O SEGMENT
FFD0
FFD0          PIO     GL_FMLH                       ; FREE MEMORY LIST HEAD
FFD0          .LONG   PIO$GL_FMLH,PIO$GL_FMLH ;           (Empty List)
FFD0
FFD0          PIO     GL_IIOFSPLH                   ; FREE LIST HEADER FOR IMAGE I/O SEG
FFD0          .LONG   PIO$GL_IIOFSPLH               ;
FFD0          .LONG   PIO$GL_IIOFSPLH               ;
FFD0
FFD0          PIO     GW_STATUS                     ; RMS OVERALL STATUS
FFD0          .WORD   0                             ;
FFD0
FFD0
FFD0  PIO$S_EODSTR==16                              ; SIZE OF STRING + COUNT BYTE
FFD0          PIO     GT_ENDSTR                     ; END OF DATA STRING
FFD0          .BLKB   PIO$S_EODSTR                  ; (COUNTED STRING)
FFD0
FFD0          PIO     GW_DFPROT                     ; DEFAULT FILE PROTECTION
FFD0          .WORD   ^XFA00                        ; SYS:RWED, OWN:RWED, GROUP: RE, WORLD:N
FFD0
FFD0          PIO     GB_DFMBC                      ; DEFAULT MULTI-BLOCK COUNT
FFD0          .BLKB   1
FFD0
FFD0          PIO     GB_DFMBFSDK                   ; DEFAULT MULTI-BUFFER COUNT SEQ. DISK
FFD0          .BLKB   1
FFD0
FFD0          PIO     GB_DFMBFSMT                   ; DEFAULT MULTI-BUFFER COUNT MAGTAPE
FFD0          .BLKB   1
FFD0
FFD0          PIO     GB_DFMBFSUR                   ; DEFAULT MULTI-BUFFER COUNT UNIT REC.
FFD0          .BLKB   1
FFD0
FFD0          PIO     GB_DFMBFREL                   ; DEFAULT MULTI-BUFFER COUNT RELATIVE
FFD0          .BLKB   1
FFD0
FFD0          PIO     GB_DFMBFIDX                   ; DEFAULT MULTI-BUFFER COUNT INDEXED
FFD0          .BLKB   1
FFD0
FFD0          PIO     GB_DFMBFHSH                   ; DEFAULT MULTI-BUFFER COUNT HASHED
FFD0          .BLKB   1
FFD0
FFD0          PIO     GB_DFNBC                      ; Network block count transfer size
FFD0          .BLKB   1
FFD0
FFD0          PIO     GB_RMSPROLOG                  ; Structure level for RMS files
FFD0          .BLKB   1
FFD0
FFD0          PIO     GW_RMSEXTEND                  ; Extend quantity for RMS files
FFD0          .BLKW   1
FFD0
```

```
FFD0            .ALIGN  LONG
FFD0
FFD0            PIO     GL_DIRCACHE             ; DIRECTORY CACHE LIST HEAD
FFD0            .LONG   PIO$GL_DIRCACHE         ; EMPTY LIST
FFD0            .LONG   PIO$GL_DIRCACHE
FFD0
FFD0            PIO     GL_DIRCFRLH             ; FREE LIST FOR DIRECTORY CACHE NODES
FFD0            .LONG   0                       ;   (NOTE: SINGLY LINKED)
FFD0
FFD0            PIO     GL_RULOCK               ; List of locks held for Recovery Units
FFD0            .LONG   0                       ;   (Note: Singly Linked)
FFD0
FFD0            PIO     GL_NXTIRBSEQ            ; Next sequence number for IRB$L_IDENT
FFD0            .LONG   0
FFD0
FFD0            .PAGE
FFD0
FFD0            .ALIGN  QUAD
FFD0
FFD0   ;
FFD0   ;        PROCESS I/O SEGMENT CONTEXT AREA
FFD0   ;
FFD0            PIO     GW_PIOIMPA              ;
FFD0            .WORD   0                       ; FLAGS
FFD0            .WORD   PRT$C_UREW              ; I/O BUFFER PROTECTION
FFD0            .LONG   0,0                     ; PIO SEGMENT, SET UP BY PROCSTRT
FFD0
FFD0            PIOL    A1                      ; FREE PAGE LIST HEAD
FFD0            .LONG   PIOLA1,PIOLA1           ; FREE LIST HEADER
FFD0            .BLKL   1                       ; SP SAVED LONGWORD
FFD0            .LONG   PIO2A10                 ; IFAB TABLE ADDRESS
FFD0            .LONG   PIO2A20                 ; IRAB TABLE ADDRESS
FFD0            .LONG   IMP$C_NPIOFILES         ; # OF SLOTS PER TABLE
FFD0
FFD0            .PAGE
FFD0   ;
FFD0   ;        IMAGE I/O SEGMENT CONTEXT AREA
FFD0   ;
FFD0            .ALIGN  QUAD
FFD0
FFD0            PIO     GW_IIOIMPA              ; IMAGE I/O IMPURE AREA
FFD0            .WORD   1                       ; FLAGS WITH IIO SEGMENT SET
FFD0            .WORD   PRT$C_UREW              ; PROTECTION STO SET ON PAGES
FFD0
FFD0            .LONG   0                       ; MUST BE INITIALIZED BY THE IMAGE
FFD0                                            ; ACTIVATOR TO THE ADDRESS OF THE FIRST
FFD0                                            ; PAGE IN THE IMAGE I/O SEGMENT.
FFD0                                            ; (NOTE: THIS LONGWORD IS REFERENCED BY:
FFD0                                            ; PIO$GW_IIOIMPA+IMP$L_IOSEGADDR)
FFD0            .LONG   0                       ; LENGTH OF IMAGE I/O SEGMENT IN BYTES
FFD0                                            ; MUST BE INITIALIZED BY THE IMAGE ACTIVATOR
FFD0                                            ; (NOTE: THIS LONGWORD IS REFERENCED BY:
FFD0                                            ; PIO$GW_IIOMPA+IMP$L_IOSEGLEN)
FFD0
FFD0            PIOL    B1                      ;
FFD0            .LONG   PIOLB1,PIOLB1           ; FREEPAGE LIST HEAD
FFD0
```

```
FFD0
FFD0            PIOL    B2                          ;
FFD0            .BLKL   1                           ; SP SAVE LONGWORD
FFD0
FFD0            .LONG   PIOLB10                     ; IFAB TABLE ADDRESS
FFD0            .LONG   PIOLB20                     ; IRAB TABLE ADDRESS
FFD0            .LONG   IMP$C_ENTPERSEG             ; # OF SLOTS PER TABLE
FFD0
FFD0            PIOL    B10                         ;
FFD0            .LONG   0                           ; LINK TO NEXT TABLE SEGMENT
FFD0            .BLKL   IMP$C_ENTPERSEG             ; IFAB TABLE SLOTS
FFD0
FFD0            PIOL    B20                         ;
FFD0            .LONG   0                           ; LINK TO NEXT TABLE SEGMENT
FFD0            .BLKL   IMP$C_ENTPERSEG             ; IRAB TABLE SLOTS
FFD0
FFD0            PIO     AL_RMSEXH                   ; EXIT HANDLER CONTROL BLOCK
FFD0            .BLKL   2                           ;
FFD0            .LONG   1                           ; ARGUMENT COUNT
FFD0            .LONG   PIO$AL_RMSEXH+8             ; STORE EXIT CODE OVER ARGUMENT COUNT
FFD0
FFD0            PIO     GQ_IIODEFAULT               ; DEFAULT IMAGE I/O AREA
FFD0            .LONG   0,0
FFD0
FFD0            .ALIGN  LONG
FFD0
FFD0    ;
FFD0    ;       DEFAULT DIRECTORY INFORMATION
FFD0    ;
FFD0            PIO     GT_DDSTRING                 ; DEFAULT DIRECTORY STRING
FFD0
FFD0    FIL$GT_DDSTRING==PIO$GT_DDSTRING            ; FILE READ DEFAULT DIRECTORY STRING
FFD0
FFD0    ; THE DEFAULT DIRECTORY STRING IN THE PROCESS QUOTA BLOCK MOVED FROM A
FFD0    ; CREATOR PROCESS TO A NEWLY CREATED PROCESS MUST BE AT LEAST AS LARGE
FFD0    ; AS THE LARGEST DIRECTORY STRING ALLOWED BY RMS.
FFD0
FFD0            ASSUME  PQB$S_DDSTRING GE FWA$C_MAXDIRLEN
FFD0
FFD0    TEMP$...=.
FFD0            .ASCIC  \[SYSEXE]\                  ; DEFAULT DIRECTORY STRING
FFD0
FFD0    ; ****************************** NOTE WELL ******************************
FFD0    ;
FFD0    ; THE DEFAULT DIRECTORY STRING MUST BE THE LAST ELEMENT TO APPEAR IN THE
FFD0    ; RMS CONTEXT AREA. IN FACT, IT HANGS OFF THE END OF THE PAGE. IF STORAGE
FFD0    ; WERE ALLOCATED, THE SHELL WOULD GROW IN SIZE BY ONE PAGE, AN UNNECESSARY
FFD0    ; WASTE OF SPACE. THE .BLKB DIRECTIVE HERE IS SYMBOLIC, TO EXPRESS THE SIZE
FFD0    ; OF THE DIRECTORY STRING IF SPACE WERE REALLY ALLOCATED.
FFD0    ;
FFD0    ;       .BLKB   PQB$S_DDSTRING-<.-TEMP$...>         ; FILL TO MAXIMUM POSSIBLE SIZE
FFD0    ;
FFD0    ; ***************************** END OF NOTE *****************************
FFD0            .PAGE
FFD0            .SBTTL  INITIALZE SHELL WITH SYSGEN PARAMETERS
FFD0    ;++
FFD0    ; FUNCTIONAL DESCRIPTION:
```

```
FFD0      ;       SWP$SHELINIT IS CALLED BY THE SWAPPER FOLLOWING THE INSWAP OF A SHELL
FFD0      ;       PROCESS TO APPLY THE SYSGEN PARAMETERS AND CONFIGURE THE PROCESS
FFD0      ;       FOR THE PROPER WORKING SET SIZE AND VIRTUAL ADDRESS SPACE.
FFD0      ;       THIS CODE IS CONTAINED IN PAGES THAT ARE TO BECOME THE KERNEL
FFD0      ;       STACK FOR THE NEW PROCESS.
FFD0      ;
FFD0      ; CALLING SEQUENCE:
FFD0      ;       JSB     @#SWP$SHELINIT
FFD0      ;
FFD0      ; INPUT PARAMETERS:
FFD0      ;       R4 - PCB ADDRESS
FFD0      ;       R9 - SWAPPER MAP POINTER
FFD0      ;       R10 - PTC$M_VALID!PTE$C_ERKW!PTE$M_MODIFY
FFD0      ;       R11 - SWAPPER END MAP POINTER
FFD0      ;
FFD0      ; OUTPUT PARAMETERS:
FFD0      ;       PHD AND CONTROL REGION FOR THE NEW PROCESS
FFD0      ;
FFD0      ;--
FFD0
FFD0              .ALIGN  PAGE                        ; START OUT ON PAGE BOUNDARY
FFD0
FFD0      SWP$SHELINIT==.-SHELL                       ;
FFD0              MOVL    PCB$L_PHD(R4),R5            ; GET PROCESS HEADER BASE ADDRESS
FFD0              MOVL    @#SWP$GL_SHELLSIZ,R0        ; GET PAGES ALLOCATED FOR SHELL
FFD0              MOVAL   (R9)[R0],R11                ; COMPUTE END ADDRESS IN MAP
FFD0              ASHL    #2,@#SWP$GL_SHELIO,R0       ; GET I/O SIZE OF SHELL
FFD0              SUBL3   R9,R11,R1                   ; COMPUTE SIZE OF EXTENSION PAGES
FFD0              SUBL    R0,R1                       ; LESS I/O SIZE
FFD0              ROTL    #3,R1,R1                    ; CONVERT TO DOUBLE QUAD COUNT
FFD0              ROTL    #9,@#SWP$GL_SHELIO,R0       ; COMPUTE ADDRESS AT END OF I/O TRANSFER
FFD0      10$:    CLRQ    (R0)+                       ; CLEAR PAGES
FFD0              CLRQ    (R0)+                       ; NOT READ FROM SHELL
FFD0              SOBGTR  R1,10$                      ;
FFD0              MOVL    R5,R2                       ; VA OF PHD
FFD0              JSB     @#MMG$SVAPTECHK             ; GET SVA OF FIRST PHD PTE
FFD0              MOVL    (R9)+,(R3)+                 ; MAP PROCESS HEADER FIXED PAGE
FFD0              MOVZWL  @#SWP$GW_WSLPTE,R1          ; GET COUNT OF ADDED PAGES FOR WSL+PST
FFD0              BEQL    30$                         ; BR IF NONE
FFD0              DECL    R1                          ; LESS FIXED HEADER PAGE
FFD0      20$:    MOVL    -(R11),(R3)+                ; MAP A WSL PAGE
FFD0              SOBGTR  R1,20$                      ; DO THEM ALL
FFD0      30$:    MOVZWL  @#SWP$GW_EMPTPTE,R0         ; GET COUNT OF EMPTY PAGES
FFD0              MOVAL   (R3)[R0],R3                 ; UPDATE MAP POINTER
FFD0              MOVZWL  @#SWP$GW_BAKPTE,R0          ; GET COUNT OF BAK/WSL/VAL/LCK PTE
FFD0      40$:    MOVL    -(R11),(R3)+                ; MAP BAK/WSL/VAL/LCK PAGES
FFD0              SOBGTR  R0,40$                      ;
FFD0              INVALID                             ; INVALIDATE TRANSLATION BUFFER
FFD0      ;
FFD0      ;       ALL OF THE HEADER PAGES LESS PAGE TABLES HAVE NOW BEEN MAPPED
FFD0      ;
FFD0              MOVZWL  @#SWP$GW_IBALSETX,R8        ; GET BALANCE SLOT INDEX
FFD0              MOVW    R8,PHD$W_PHVINDEX(R5)       ; SET INTO PROCESS HEADER
FFD0              CVTWB   @#SGN$GW_DFPFC,PHD$B_DFPFC(R5)  ; SET DEFAULT PAGE FAULT CLUSTER
FFD0              CLRL    R0                          ; COUNT MAXIMUM FREE SPACE TILL NOW
FFD0              MOVZWL  @#MMG$GW_MINPFIDX,R3        ; START SCAN AT FIRST PAGE FILE
FFD0              MOVL    R3,R1                       ; SET DEFAULT PAGING FILE TO CHOOSE
```

```
FFD0             MOVL    @#MMG$GL_PAGSWPVC,R6     ; GET ADDRESS OF PAGE FILES VECTOR
FFD0        ;
FFD0        ;    CHECK FOR SPECIFICATION OF A PAGE FILE
FFD0        ;    THIS IS TAKEN AS ADVICE ONLY, NORMAL ALGORITHM USED IF SELECTION IMPOSSIBLE
FFD0        ;
FFD0             MOVZWL  PCB$B_PGFLINDEX(R4),R2   ; GET THE REQUESTED PAGE FILE INDEX
FFD0             BEQL    44$                      ; NONE SPECIFIED
FFD0             CMPL    R2,R1                    ; CHECK FOR LEGAL INDEX
FFD0             BLSSU   44$                      ; A SWAP FILE WAS SPECIFIED - GIVE UP
FFD0             CMPL    R2,@#MMG$GL_MAXPFIDX
FFD0             BGTRU   44$                      ; TOO HIGH - GIVE UP
FFD0             MOVL    (R6)[R2],R8              ; GET PFL BLOCK ADDRESS
FFD0             BBC     #PFL$V_INITED,PFL$B_FLAGS(R8),44$ ; BRANCH IF NOT USABLE
FFD0             MOVL    R2,R1
FFD0             BRB     47$                      ; WE HAVE A GOOD SELECTION
FFD0     44$:    MOVL    (R6)[R3],R2              ; GET ADDRESS OF NEXT PAGE FILE BLOCK
FFD0             BISB    #PFL$M_INITED,PCB$W_PGFLCHAR(R4); ADD INITED TO THE REQUESTED FLAGS
FFD0             BISB3   PCB$W_PGFLCHAR(R4),PFL$B_FLAGS(R2),R8 ; OR THE FLAGS TOGETHER
FFD0             CMPB    R8,PFL$B_FLAGS(R2)       ; DID THEY CHANGE
FFD0             BNEQ    46$                      ; YES - NO MATCH - TRY AGAIN
FFD0             CMPL    PFL$L_FREPAGCNT(R2),R0   ; CHOOSE PAGING FILE WITH MOST FREE PAGES
FFD0             BLEQ    46$                      ; BRANCH IF WE HAVE BETTER CANDIDATE
FFD0             MOVL    PFL$L_FREPAGCNT(R2),R0   ; SAVE NEW FREE PAGE COUNT
FFD0             MOVL    R3,R1                    ; SAVE NEW INDEX
FFD0     46$:    AOBLEQ  @#MMG$GL_MAXPFIDX,R3,44$; LOOP THROUGH ALL PAGE FILES
FFD0     47$:    MOVB    R1,PHD$B_PAGFIL(R5)      ; SET PAGING FILE INDEX
FFD0             MOVB    @#SGN$GB_PGTBPFC,PHD$B_PGTBPFC(R5); SET SYSTEM DEFAULT PT CLUSTER
FFD0             MOVW    #SWP$C_KSTACK_WSL,PHD$W_WSLIST(R5); INIT POINTER TO LIST BASE
FFD0             MOVL    @#SGN$GL_PHDPAGCT,R7     ; GET TOTAL COUNT OF HEADER PAGES
FFD0             MOVZWL  @#SWP$GW_WSLPTE,R6       ; GET COUNT OF WSL HEADER PAGES
FFD0             ASHL    #9,R6,PHD$L_PSTBASOFF(R5)        ; SET END AS BASE FOR PST
FFD0             ADDW    @#SWP$GW_EMPTPTE,R6      ; ASSUMES NO OVERFLOW POSSIBLE
FFD0             ASHL    #7,R6,R6                 ; CONVERT TO LONGWORD COUNT
FFD0             MOVL    R6,PHD$L_WSLX(R5)        ; SET BASE OFFSET TO WSLX AREA
FFD0             ADDL    @#SGN$GL_PTPAGCNT,R7     ; ADD PAGE TABLES TO COUNT
FFD0             ADDL3   #1,R7,R0                 ; ROUND TO LONGWORD SIZE
FFD0             DIVL    #2,R0                    ;
FFD0             ADDL    R0,R6                    ; ALLOCATE SPACE IN WHOLE LONGWORDS
FFD0             MOVL    R6,PHD$L_BAK(R5)         ; SET BASE OF BACKING STORE VECTOR
FFD0             ADDL    R7,R6                    ; ALLOCATE SPACE FOR BACKING STORE VECTOR
FFD0             MULL    #4,R6                    ; CONVERT TO BYTE OFFSET
FFD0             MOVL    R6,PHD$L_PTWSLELCK(R5)   ; SET BASE OF LOCKED COUNT VECTOR
FFD0             MOVL    @#SGN$GL_PTPAGCNT,R7     ; GET COUNT OF PAGE TABLES
FFD0             MOVAB   (R6)[R7],PHD$L_PTWSLEVAL(R5)     ; ALLOCATE AND SET BASE OF VAL CNT
FFD0             MOVAB   @PHD$L_PTWSLEVAL(R5)[R5],R0      ; SET BASE FOR VALID COUNT VEC
FFD0             MNEGL   #1,R2                            ; MINUS ONE FOR BACKGROUND
FFD0             MOVAB   @PHD$L_PTWSLELCK(R5)[R5],R1      ; AND BASE FOR LOCKED COUNT
FFD0     50$:    MOVB    R2,(R0)+                 ; INIT BOTH COUNT VECTORS TO MINUS ONE
FFD0             MOVB    R2,(R1)+                 ;
FFD0             SOBGTR  R7,50$                   ; FOR ALL PAGE TABLE SLOTS
FFD0             ADDB    #KSTACK+1,-2(R1)         ; COUNT OF LOCK PAGES FOR SHELL
FFD0             ADDB    #KSTACK+3,-2(R0)         ; COUNT OF VALID PAGES
FFD0             MOVL    @#SGN$GL_PHDAPCNT,R6     ; GET ACTUAL HEADER SIZE
FFD0             MOVZBL  @#SWP$GB_SHLP1PT,R0      ; NUMBER OF PERM PAGE TABLES
FFD0             MOVW    R0,PHD$W_PTCNTLCK(R5)    ; COUNT OF PT CONTAINING LOCKED PAGES
FFD0             MOVW    R0,PHD$W_PTCNTVAL(R5)    ; COUNT OF PT CONTAINING VALID PAGES
FFD0             MOVW    R0,PHD$W_PTCNTACT(R5)    ; COUNT OF ACTIVE PAGE TABLES
```

```
FFD0            MOVW    R0,PHD$W_PTCNTMAX(R5)   ; COUNT OF PT WITH NON-ZERO ENTRIES
FFD0            ADDL    R0,R6                   ; MAKE ROOM IN WSLIST FOR HDR+PGTBLS
FFD0            MOVZWL  PHD$W_WSDYN(R5),R7      ; GET INDEX TO CURRENT DYNAMIC ENTRIES
FFD0            PUSHL   R7                      ; SAVE POINTER FOR LATER USE
FFD0            ADDW    R6,PHD$W_WSLOCK(R5)     ; CORRECT LOCK WSL INDEX
FFD0            ADDW    R6,PHD$W_WSDYN(R5)      ; AND DYNAMIC INDEX
FFD0            ADDW    R6,PHD$W_WSNEXT(R5)     ; AND NEXT POINTER
FFD0            ADDL3   R6,R7,R1                ; SET DESTINATION INDEX
FFD0            MOVZBL  #SWP$C_NDYN,R0          ; SET COUNT OF DYNAMIC PAGES
FFD0      60$:  MOVL    (R5)[R7],(R5)[R1]       ; MOVE A DYNAMIC WORKING SET PAGE
FFD0            INCL    R1                      ; NEXT DESTINATION
FFD0            INCL    R7                      ; NEXT WSL ENTRY
FFD0            SOBGTR  R0,60$                  ; MOVE ALL DYNAMIC PAGES
FFD0            MOVL    #<WSL$M_VALID-          ; WSL ENTRIES FOR HEADER ARE VALID
FFD0                    !WSL$M_WSLOCK-          ; LOCKED,
FFD0                    !WSL$C_PPGTBL-          ; PROCESS PAGE TABLE,
FFD0                    !VA$M_SYSTEM>,R8        ; AND SYSTEM SPACE
FFD0            MOVL    R5,R2                   ; SET PHD ADDRESS TO GET SVAPTE
FFD0            JSB     @#MMG$SVAPTECHK         ; SVAPTE TO R3
FFD0            MOVL    R8,R0                   ; MAKE WORKING COPY
FFD0            MOVZWL  @#SWP$GW_WSLPTE,R1      ; COUNT OF WSLPTE
FFD0            MOVL    (SP)+,R2                ; RESTORE WORKING SET INDEX TO LOCKED
FFD0            BSBW    FILLPHD                 ; SET UP FOR VALID PHD PAGES
FFD0            MOVZWL  @#SWP$GW_EMPTPTE,R1     ; GET COUNT OF EMPTY PTE AGAIN
FFD0            MOVAL   (R3)[R1],R3             ; ADVANCE SPT ADDRESS
FFD0            ASHL    #9,R1,R1                ; CONVERT TO BYTE COUNT
FFD0            ADDL    R1,R0                   ; AND UPDATE CURRENT VA
FFD0            MOVZWL  @#SWP$GW_BAKPTE,R1      ; GET COUNT OF REMAINING HEADER PAGES
FFD0            BSBW    FILLPHD                 ; SET UP FOR VALID PHD PAGES
FFD0            MOVL    @#SGN$GL_PTPAGCNT,R1    ; GET COUNT OF PAGE TABLES
FFD0      65$:  MOVL    #PTE$C_ERKW,-4(R3)[R1]  ; SET A DEMAND ZERO PAGE FOR PAGE TABLES
FFD0            SOBGTR  R1,65$                  ; INIT ENTIRE PAGE TABLE AREA
FFD0            MOVZBL  @#SWP$GB_SHLP1PT,R1     ; GET COUNT OF P1 PAGE TABLES
FFD0            MNEGL   R1,R6                   ; NEGATE COUNT
FFD0            DECL    R6                      ; LESS ONE MORE FOR DEBUGGER PAGE
FFD0            ADDL    @#SGN$GL_PTPAGCNT,R6    ; PAGE OFFSET TO PAGE TABLE
FFD0            MOVAL   (R3)[R6],R3             ; COMPUTE SVAPTE
FFD0            ASHL    #9,R6,R6                ; CONVERT TO BYTE OFFSET
FFD0            ADDL    R6,R0                   ; ADD TO CURRENT BASE WSL
FFD0            PUSHR   #^M<R1,R3>              ; SAVE KEY REGS
FFD0            SUBL    S^#SWP$C_SHLP1PT,R1     ; LESS STANDARD P1 PAGE TABLES
FFD0            BLEQ    80$                     ; BR IF NO EXTENSIONS
FFD0      70$:  MOVL    -(R11),(R3)+            ; STORE A PTE FOR EXTENSION
FFD0            SOBGTR  R1,70$                  ; DO ALL EXTENSION PAGE TABLES
FFD0      80$:  MOVL    S^#SWP$C_SHLP1PT,R1     ; SET COUNT OF STANDARD
FFD0      90$:  MOVL    (R9)+,(R3)+             ; STORE PTE FOR STANDARD
FFD0            SOBGTR  R1,90$                  ; DO ALL STANDARD P1PT
FFD0            POPR    #^M<R1,R3>              ; RESTORE KEY REGISTERS
FFD0            BSBW    FILLPHD                 ; SET UP FOR VALID PHD PAGES
FFD0            ROTL    #9,@#SGN$GL_PHDPAGCT,R0 ; OFFSET FOR P0BR
FFD0            ADDL    R0,PHD$L_P0BR(R5)       ; FORM P0BR BASE VALUE
FFD0            ROTL    #9,@#SGN$GL_PTPAGCNT,R1 ; OFFSET TO P1BR
FFD0            ADDL    R0,R1
FFD0            ADDL    R1,PHD$L_P1BR(R5)       ; FORM P1BR BASE VALUE
FFD0            SUBL    R0,PHD$L_FREP1VA(R5)    ; COMPUTE CORRECT FREP1VA
FFD0            SUBL    R0,R1                   ; EXTRACT PT SIZE TOTAL
FFD0            ASHL    #-9,R0,R0               ; BACK TO PAGE COUNT
```

```
FFD0            SUBL    R0,PHD$L_P1LR(R5)       ; NOW CORRECT P1 LENGTH REGISTER
FFD0            ASHL    #-2,R1,R1               ; COUNT OF AVAILABLE PTES
FFD0            SUBL    R0,R1                   ; DIMINISH BY PHD WINDOW SIZE
FFD0            ADDL    R1,PHD$L_FREPTECNT(R5)  ; AND FORM COUNT OF FREE PTES
FFD0
FFD0      ; THE FOLLOWING SHUFFLE OF THE WORKING SET LIST REARRANGES ALL WORKING SET
FFD0      ; LIST ENTRIES THAT ARE NOT KERNEL STACK PAGES. THE SWAPPER MAP IS BEING
FFD0      ; RECRDERED TO AGREE WITH THE WORKING SET LIST DEFINED IN THE SHELL.
FFD0
FFD0            .REPEAT SHUFFLE_COUNT           ; COUNT FROM WSL TEMPLATE
FFD0            MOVL    (R9)+,(R11)+            ; REARRANGE SWAP PAGE LIST
FFD0            .ENDR
FFD0
FFD0            CLRL    (R11)+                  ; SET STOPPER IN MAP
FFD0            ADDL3   #512,PHD$L_FREP1VA(R5),-        ; SET ADDRESS OF WINDOW
FFD0                    @#<SWP$AL_PTRPAG+-      ; INTO POINTER PAGE
FFD0                    <CTL$GL_PHD-CTL$GL_VECTORS>>    ; THROUGH SWAPPER MAP
FFD0            BBSS    #PCB$V_PHDRES,PCB$L_STS(R4),100$; MARK PHD RESIDENT
FFD0      100$:                                 ;
FFD0            MOVZWL  PHD$W_WSLIST(R5),R0     ; GET INDEX TO WS BASE
FFD0            DECL    R0                      ;
FFD0            ADDW3   R0,@#SGN$GL_MAXWSCNT,R1                   ; CALC MAX AUTHORIZED
FFD0            MOVW    R1,PHD$W_WSAUTH(R5)                       ; SET MAX AUTHORIZED
FFD0            MOVW    R1,PHD$W_WSQUOTA(R5)                      ; AND QUOTA
FFD0            MOVW    R1,PHD$W_WSEXTENT(R5)                     ; AND EXTENT
FFD0            MOVW    R1,PHD$W_WSAUTHEXT(R5)                    ; AND AUTHORIZED EXTENT
FFD0            MOVW    @#SGN$GW_MINWSCNT,PHD$W_WSFLUID(R5)       ; SET FLUID REQUIREMENT
FFD0            MOVW    @#SGN$GW_DFWSCNT,R1                       ; GFT DEFUALT WS SIZE
FFD0            ADDW    R1,R0                                     ; CALC LAST
FFD0            MOVW    R0,PHD$W_WSLAST(R5)                       ; SET LAST
FFD0            MOVW    R0,PHD$W_DFWSCNT(R5)                      ; AND DEFAULT COUNT
FFD0            MOVW    R1,PHD$W_WSSIZE(R5)                       ; SET WS SIZE
FFD0            ADDL    R5,PHD$L_P0BR(R5)       ; BIAS P0 BASE REGISTER
FFD0            ADDL    R5,PHD$L_P1BR(R5)       ; AND P1 BASE REGISTER ALSO
FFD0      ;
FFD0      ;     REMAP SYSTEM SERVICE VECTORS FOR THIS PROCESS
FFD0      ;
FFD0            MOVL    #P1SYSVECTORS,R2        ; PUT THEM HERE
FFD0            JSB     @#MMG$SVAPTECHK         ; GET ADDRESS OF PTE FOR VECTORS
FFD0            MOVL    @#MMG$GL_SPTBASE,R0     ; SYSTEM PAGE TABLE
FFD0            MOVZBL  #SGN$C_SYSVECPGS,R1     ; NUMBER OF PAGES
FFD0      110$: BISL3   #PTE$M_WINDOW,(R0)+,(R3)+
FFD0            SOBGTR  R1,110$
FFD0      ;
FFD0      ;     THE PAGE TABLE ARRAY FOR LOCKED WSLE'S MUST BE INCREMENTED ONE
FFD0      ;     FOR EACH WINDOW PTE.  THIS BALANCES THE LOGIC IN $CREPAG/$DELPAG
FFD0      ;     THAT PLACES THE PAGE TABLE PAGE IN THE LOCKED PORTION OF THE
FFD0      ;     WORKING SET FOR USER PFNMAPPED PAGES.
FFD0      ;
FFD0      ;     THIS LOGIC ASSUMES THAT THE VECTOR PAGES ALL LIVE IN THE SAME
FFD0      ;     PAGE TABLE PAGE.
FFD0      ;
FFD0            SUBL3   PHD$L_P0BR(R5),R3,R0    ;BYTE OFFSET OF PTE
FFD0            ASHL    #-9,R0,R0               ;BYTE INDEX OF CONTAINING PAGE TABLE
FFD0            ADDL    PHD$L_PTWSLELCK(R5),R0 ;COMPUTE THE ADDRESS OF THE LOCKED
FFD0            ADDL    R5,R0                   ; WORKING SET LIST PAGE TABLE BYTE
FFD0            ADDB    #SGN$C_SYSVECPGS,(R0)   ; ARRAY AND INCR BY NUMBER OF VEC PAGES
```

```
FFD0
FFD0 ;
FFD0 ;          FINISH UP INITIALIZATION, SET DATA NEEDED BY SWAPPER
FFD0 ;
FFD0            MNEGL   PCB$L_WSSWP(R4),PCB$L_WSSWP(R4)
FFD0                                            ; INDICATE SHELL FINISHED TO SWAPPER
FFD0            CVTLW   PCB$L_SWAPSIZE(R4),PHD$W_SWAPSIZE(R5) ; SET INITIAL SWAP ALLOCATION
FFD0
FFD0 ; CLEAR THE PCB FIELDS THAT WERE USED TO CARRY PAGE FILE AND SWAP FILE
FFD0 ; INFORMATION TO THE NEW PROCESS. THESE FIELDS WILL BECOME THE COMMON EVENT
FFD0 ; FLAG CLUSTER POINTERS.
FFD0
FFD0            ASSUME PCB$L_EFC3P EQ <PCB$L_EFC2P + 4>
FFD0            ASSUME PCB$W_PGFLCHAR EQ PCB$L_EFC2P
FFD0            ASSUME PCB$B_PGFLINDEX EQ <PCB$L_EFC2P + 2>
FFD0            ASSUME PCB$L_SWAPSIZE EQ <PCB$L_EFC2P + 4>
FFD0
FFD0            CLRQ    PCB$L_EFC2P(R4)         ; START WITH NO COMMON EF CLUSTERS
FFD0
FFD0            RSB                             ; RETURN
FFD0
FFD0            .PAGE
FFD0            .SBTTL  FILLPHD - SETUP A VALID PHD PTE
FFD0 ;
FFD0 ;          R0 - WORKING SET LIST ENTRY, ADDRESS AND FLAGS (UPDATED)
FFD0 ;          R1 - COUNT OF PAGES TO FILL (UPDATED)
FFD0 ;          R2 - WORKING SET LIST INDEX (UPDATED)
FFD0 ;          R3 - SVAPTE FOR PHD PAGE (UPDATED)
FFD0 ;          R4 - PCB ADDRESS
FFD0 ;          R5 - PHD ADDRESS
FFD0 ;          R7 - SCRATCH
FFD0 ;          R8 - SCRATCH
FFD0 ;          R10 - PFN$C_ERKW!PFN$M_MODIFY!PFN$M_VALID
FFD0 ;
FFD0 FILLPHD:                                   ; SETUP VALID PHD PTES
FFD0 10$:       MOVL    R0,(R5)[R2]             ; STORE WORKING SET LIST ENTRY
FFD0            BICL3   R10,(R3)+,R7            ; GET PFN FROM SPT
FFD0            PUSHL   @#PFN$AB_STATE          ; STACK ADDRESS BASE FOR PFN STATE
FFD0            MOVB    #<PFN$C_ACTIVE!PFN$M_MODIFY>,@(SP)+[R7] ; MARK PAGE ACTIVE
FFD0            PUSHL   @#PFN$AB_TYPE           ; STACK ADDRESS BASE FOR PFN TYPE
FFD0            MOVB    #PFN$C_PPGTBL,@(SP)+[R7]; SET PAGE TYPE TO PAGE TABLE
FFD0            PUSHL   @#PFN$AL_BAK            ; STACK ADDRESS BASE FOR PFN BAK
FFD0            MOVZBL  PHD$B_PAGFIL(R5),R8     ; GET PAGE FILE NUMBER
FFD0            ROTL    #PFN$V_PGFLX,R8,@(SP)+[R7]     ; SET BACKING STORE ADDRESS
FFD0            PUSHL   @#PFN$AL_PTE            ; STACK ADDRESS OF PFN PTE BACK POINTER
FFD0            MOVAL   -4(R3),@(SP)+[R7]       ; SET PTE BACK POINTER FOR PAGE
FFD0 ;
FFD0 ;  The following use of the PFN_REFERENCE macro must force absolute addressing
FFD0 ;  because of the peculiar method in which this code executes. In addition,
FFD0 ;  this code is not a part of the nonpaged executive and cannot have its opcode
FFD0 ;  automatically fixed up by INIT.
FFD0 ;
FFD0            PUSHL   @#PFN$Ax_WSLX           ; STACK BASE ADDRESS OF WSLX VECTOR
FFD0                    PFN_REFERENCE   -
FFD0            MOVW    <R2,@(SP)+[R7]>,-       ; SET WSLX FOR PAGE
FFD0                    LONG_OPCODE=MOVZWL,-
FFD0                    IMAGE=SHELL,-
```

```
FFD0                      MODE=@#                 ; FORCE POSITION INDEPENDENCE
FFD0              INCL    R2                      ; NEXT WORKING SET LIST ENTRY
FFD0              MOVAL   512(R0),R0              ; NEXT VA
FFD0              SOBGTR  R1,10$                  ; FILL ALL REQUESTED PAGES
FFD0              RSB                             ; AND RETURN
FFD0
FFD0              .ALIGN  PAGE                    ; PAGE ALIGN
FFD0      SWP$C_SHELLSIZ=<.-SHELL>@-9             ; SIZE OF SHELL PROCESS IN PAGES
FFD0              .IF GT  SWP$C_SHELLSIZ-8
FFD0              .ERROR  SWP$C_SHELLSIZ; Shell size changed, update swapper and FREELIM
FFD0              .ENDC
FFD0
FFD0              .PAGE
FFD0
FFD0      ;
FFD0      ;       Process IFI/ISI Tables
FFD0      ;
FFD0
FFD0      ; The rest of this module merely defines global symbols and offsets into
FFD0      ; various P1 pages. No more storage is added to the SHELL module.
FFD0
FFD0              .PSECT  $ABS$,ABS
FFD0
FFD0              .=PIOTBL
FFD0      TBL:
FFD0              PIO2    A10                     ; LINK TO NEXT TABLE SEGMENT
FFD0              .BLKL   1                       ; INITIALLY ZERO
FFD0              .BLKL   IMP$C_NPIOFILES         ; IFAB TABLE SLOTS
FFD0
FFD0              PIO2    A20                     ; LINK TO NEXT TABLE SEGMENT
FFD0              .BLKL   1                       ; INITIALLY ZERO
FFD0              .BLKL   IMP$C_NPIOFILES         ; IRAB TABLE SLOTS
FFD0
FFD0              .PAGE
FFD0              .SBTTL  COMMAND LANGUAGE INTERPRETER DATA AREA
FFD0      ;-----------------------------------------------------------------------
FFD0      ;               GENERIC COMMAND LANGUAGE INTERPRETER DATA
FFD0      ;-----------------------------------------------------------------------
FFD0
FFD0              .=CLIDATAPAG                    ; SET BASE VALUE FOR REGION
FFD0      CTL$AL_CLICALBK::                       ; CALL BACK VECTOR FOR CLI
FFD0              .BLKL   1                       ;
FFD0              .BLKL   1                       ; ALLOW FOR EXPANSION
FFD0      CTL$AG_CLIMAGE::
FFD0              .BLKL   2                       ; VA RANGE INTO WHICH CLI IS MAPPED
FFD0      CTL$AG_CLITABLE::                       ; VA RANGE INTO WHICH CLI TABLE IS
FFD0              .BLKL   2                       ; MAPPED.
FFD0
FFD0      CTL$GL_UAF_FLAGS::
FFD0              .BLKL   1                       ; FLAGS FROM AUTHORIZATION RECORD
FFD0
FFD0      CTL$GT_CLINAME::                        ; CLI NAME (FILE NAME ONLY)
FFD0              .BLKB   PQB$S_CLI_NAME          ; STORED AS COUNTED ASCII
FFD0      CTL$GT_TABLENAME::                      ; CLI TABLE NAME (FULL FILE SPEC)
FFD0              .BLKB   PQB$S_CLI_TABLE         ; STORED AS COUNTED ASCII
FFD0      CTL$GT_SPAWNCLI::                       ; SPAWN CLI NAME (FILE NAME ONLY)
FFD0              .BLKB   PQB$S_SPAWN_CLI         ; STORED AS COUNTED ASCII
```

```
FFD0  CTL$GT_SPAWNTABLE::                       ; SPAWN CLI TABLE NAME (FULL FILE SPEC)
FFD0          .BLKB   PQB$S_SPAWN_TABLE         ; STORED AS COUNTED ASCII
FFD0
FFD0  ASSUME <CTL$GT_TABLENAME -CTL$GT_CLINAME> EQ <PQB$T_CLI_TABLE  -PQB$T_CLI_NAME>
FFD0  ASSUME <CTL$GT_SPAWNCLI  -CTL$GT_CLINAME> EQ <PQB$T_SPAWN_CLI  -PQB$T_CLI_NAME>
FFD0  ASSUME <CTL$GT_SPAWNTABLE-CTL$GT_CLINAME> EQ <PQB$T_SPAWN_TABLE-PQB$T_CLI_NAME>
FFD0
FFD0  CTL$AG_CLIDATA::                          ; START OF DATA AREA
FFD0  CTL$C_CLIDATASZ==CLIDATAEND-CTL$AG_CLIDATA      ; SIZE OF DATA AREA
FFD0
FFD0          .PAGE
FFD0          .SBTTL  COMPATIBILITY MODE EMULATOR CONTEXT PAGE
FFD0  ;--------------------------------------------------------------------
FFD0  ;       COMPATIBILITY MODE EMULATOR CONTEXT PAGE
FFD0  ;--------------------------------------------------------------------
FFD0          .=CTL$AG_CMEDATA                  ; POINT TO START OF AREA
FFD0  CTL$AL_CMCNTX::                           ; COMPATIBILITY MODE REGISTER CONTEXT
FFD0          .BLKL   10                        ; SAVED BY EXCEPTION
FFD0
FFD0          .PAGE
FFD0          .SBTTL  GLOBAL USER-MODE CONTEXT PAGE
FFD0  ;--------------------------------------------------------------------
FFD0  ;       GLOBAL USER-MODE-ONLY CONTEXT PAGE
FFD0  ;
FFD0  ;       This page may ONLY be used by user-mode code, since the data
FFD0  ;       has absolutely no protection from errant user-mode programs.
FFD0  ;       Do NOT use this area for cells which cannot be wiped out by
FFD0  ;       any user-mode program at any time during image execution,
FFD0  ;       except when it only prevents that program from running.
FFD0  ;--------------------------------------------------------------------
FFD0          .=UWVECPAG                        ; POINT TO START OF AREA
FFD0
FFD0  CTL$GL_DCLPRSOWN::                        ; ADDRESS OF DCL WRK AREA CREATED
FFD0          .BLKL   1                         ; BY DCL OR CLI$DCL_PARSE
FFD0  CTL$GL_CLINTOWN::                         ; ADDRESS OF CONTEXT BLOCK USED BY
FFD0          .BLKL   1                         ; CLI$INTERFACE PACKAGE IN DCL/MCR
FFD0
FFD0  ; THE REST OF THIS PAGE IS UNUSED
FFD0
FFD0          .PAGE
FFD0          .SBTTL  IMAGE ACTIVATOR CONTEXT PAGE
FFD0  ;--------------------------------------------------------------------
FFD0  ;       IMAGE ACTIVATOR CONTEXT PAGE
FFD0  ;
FFD0  ;       The following page contains image activator context that must
FFD0  ;       remain behind after an image is activated. The IAFLIST survives
FFD0  ;       across successive activations. The IAFEXE and IAFMERGE lists are
FFD0  ;       cleared when a new image activation is initiated. The FIXUP link
FFD0  ;       is cleared as soon as a given fixup pass completes.
FFD0  ;
FFD0  ;--------------------------------------------------------------------
FFD0          .=IMGACTCTX                       ; Point to start of area
FFD0
FFD0  CTL$GL_IAFLINK::                          ; Listhead of linked list of fixup vectors
FFD0          .BLKL   1                         ;  (containing shareable image lists)
FFD0
FFD0  CTL$GL_IAFLAST::                          ; Address of last fixup vector in list
```

```
FFD0          .BLKL   1                   ;  (used in normal and merged image activati
FFD0
FFD0 CTL$GL_FIXUPLNK::                    ; Listhead of linked list of fixup
FFD0          .BLKL   1                   ;  vectors used during fixup pass
FFD0
FFD0 CTL$GL_P1MERGE::                     ; Listhead of linked list of fixup vectors
FFD0          .BLKL   1                   ;  for images merged into P1 space
FFD0
FFD0 ; The following data structure is an empty fixup vector that facilitates
FFD0 ; the fixup vector list manipulation. It is plugged into the fixup vector
FFD0 ; lists by PROCSTRT when a process is first created.
FFD0
FFD0 CTL$GL_IAFPERM::
FFD0          .BLKB   IAF$K_LENGTH
FFD0
FFD0 ; The following label locates the link field in the just allocated structure.
FFD0 ; While an image is active, it locates the fixup vector for an executable image.
FFD0 ; It is cleared as part of the image activation initialization code path.
FFD0
FFD0 CTL$GL_IAFEXE == CTL$GL_IAFPERM + IAF$L_IAFLINK
FFD0
FFD0 IAC$GL_IMAGCTX::
FFD0          .BLKL   1                   ; Context that exists for life of image
FFD0
FFD0 IAC$GL_PROCCTX::
FFD0          .BLKL   1                   ; Context that exists beyond image exit
FFD0
FFD0 IAC$AL_VECADDR::
FFD0          .BLKL   4                   ; Array of altered opcode addresses
FFD0
FFD0 IAC$AL_VECOPCOD::
FFD0          .BLKB   4                   ; Array of saved opcodes
FFD0
FFD0 IAC$AW_VECRESET::
FFD0          .BLKW   4                   ; Array of offsets used to reset vectors
FFD0
FFD0 IAC$AW_VECSET::
FFD0          .BLKW   4                   ; Array of offsets used to locate vectors
FFD0
FFD0 ; The following linked list contains image control blocks for all of the
FFD0 ; images currently mapped into a process' address space.
FFD0
FFD0 IAC$GL_IMAGE_LIST::
FFD0          .BLKL   2                   ; Room for forward and backward links
FFD0
FFD0 ; The following linked list is used by the image activator to record work
FFD0 ; in progress. It is empty while an image is executing.
FFD0
FFD0 IAC$GL_WORK_LIST::
FFD0          .BLKL   2                   ; Room for forward and backward links
FFD0
FFD0 ; The following list is a potential source of unused image control blocks.
FFD0 ; Although empty when a process is created, it grows to reflect the largest
FFD0 ; number of images activated at the same time.
FFD0
FFD0 IAC$GL_ICBFL::
FFD0          .BLKL   2                   ; Room for forward and backward links
```

```
FFD0
FFD0   ; The following two cells locate the ICB for the main image and the ICB for
FFD0   ; the image most recently merged into the address space.
FFD0
FFD0   IAC$GL_MAIN_ICB::
FFD0           .BLKL   1                               ; ICB of main image
FFD0
FFD0   IAC$GL_FIRST_ICB::
FFD0           .BLKL   1                               ; ICB of image just merged
FFD0
FFD0   IAC$GL_STACK_SIZE::
FFD0           .BLKL   1                               ; Amount by which to expand user stack
FFD0           .PAGE
FFD0
FFD0           .IFT                                    ; IF USRVECTOR MODULE:
FFD0
FFD0   ;
FFD0   ; These symbols (SYS$...) specify which cells in the P1 pointer page may
FFD0   ; be used by any program, regardless of system version, without having
FFD0   ; to link with SYS.STB (and thus, make it system dependant).
FFD0   ;
FFD0   ; These symbols will remain constant forever and will continue to be supported
FFD0   ; from release to release.
FFD0   ;
FFD0
FFD0           .MACRO  USRSYM  SYM,VALCHECK,CTLSYM
FFD0           .IF     B,CTLSYM
FFD0   SYS$'SYM == CTL$'SYM
FFD0           .IF     NB,VALCHECK
FFD0           .IF     NE,CTL$'SYM'-^X'VALCHECK'
FFD0           .ERROR  ; *** CTL$'SYM' MUST EQ ^X'VALCHECK' TO REMAIN COMPATIBLE WITH PREVI
FFD0           .ENDC
FFD0           .ENDC
FFD0           .IFF
FFD0   SYS$'SYM == CTL$'CTLSYM
FFD0           .IF     NB,VALCHECK
FFD0           .IF     NE,CTL$'CTLSYM'-^X'VALCHECK'
FFD0           .ERROR  ; *** CTL$'CTLSYM' MUST EQ ^X'VALCHECK' TO REMAIN COMPATIBLE WITH PR
FFD0           .ENDC
FFD0           .ENDC
FFD0           .ENDC
FFD0           .ENDM
FFD0
FFD0           .DISABLE TRACEBACK
FFD0
FFD0           USRSYM  GL_CMCNTX,7FFEFF88      ; Address of 2 pages of AME storage
FFD0           USRSYM  GL_IAFLINK,7FFEFF8C,-   ; Points to IMGACT fixup listhead
FFD0                           GL_IAFLNKPTR
FFD0           USRSYM  GL_IMGLSTPTR,7FFEFE84   ; Points to ICM list (for debugger)
FFD0
FFD0           .ENDC                           ; END OF SHELL / USRVECTOR CONDITIONAL
FFD0
FFD0           .END
```

```
COMMON                                  = 00000004                 CTL$GL_VOLUMES                    = 7FFEFEC4
CTL$AL_CMCNTX                             ********   X    01       CTL$GL_WSPEAK                     = 7FFEFEBC
CTL$AL_FINALEXC                         = 7FFEFF28                 CTL$GQ_ALLOCREG                   = 7FFEFE8C
CTL$AL_IPASTVEC                         = 7FFEFF68                 CTL$GQ_COMMON                     = 7FFEFE60
CTL$AL_STACK                            = 7FFEFE10                 CTL$GQ_DBGAREA                    = 7FFEFF3C
CTL$AL_STACKLIM                         = 7FFEFE6C                 CTL$GQ_HELPFLAGS                  = 7FFEFFA8
CTL$AQ_EXCVEC                           = 7FFEFE34                 CTL$GQ_ISTART                     = 7FFEFEC8
CTL$A_COMMON                              ********   X    01       CTL$GQ_LNMTBLCACHE                = 7FFEFE20
CTL$GB_DEFLANG                          = 7FFEFF51                 CTL$GQ_LOGIN                      = 7FFEFEB0
CTL$GB_MSGMASK                          = 7FFEFF50                 CTL$GQ_MOUNTLST                   = 7FFEFE94
CTL$GB_PWRMODE                          = 7FFEFF24                 CTL$GQ_POALLOC                    = 7FFEFF94
CTL$GB_SSFILTER                         = 7FFEFF25                 CTL$GQ_PROCPRIV                   = 7FFEFF10
CTL$GL_CCBBASE                          = 7FFEFF38                 CTL$GQ_TERMCHAR                   = 7FFEFFB0
CTL$GL_CMCNTX                           = 7FFEFF88                 CTL$GW_CHINDX                     = 7FFEFE02
CTL$GL_CMHANDLR                         = 7FFEFE30                 CTL$GW_NMIOCH                     = 7FFEFE00
CTL$GL_CMSUPR                           = 7FFEFE28                 CTL$GW_PPMSGCHN                   = 7FFEFF52
CTL$GL_CMUSER                           = 7FFEFE2C                 CTL$T_ACCOUNT                     = 7FFEFEA8
CTL$GL_CREPRC_FLAGS                     = 7FFEFFC0                 CTL$T_NODEADDR                    = 7FFEFEF0
CTL$GL_CTLBASVA                         = 7FFEFE7C                 CTL$T_NODENAME                    = 7FFEFEF7
CTL$GL_F11BXQP                          = 7FFEFF90                 CTL$T_REMOTEID                    = 7FFEFEFE
CTL$GL_FINALSTS                         = 7FFEFEB8                 CTL$T_USERNAME                    = 7FFEFE9C
CTL$GL_GETMSG                           = 7FFEFE68                 CTLVECEND                         = 7FFEFFD0
CTL$GL_IAFLINK                            ********   X    01       CTLVECPAG                         = 7FFEFE00
CTL$GL_IAFLNKPTR                        = 7FFEFF8C                 DBGAREA                           = 7FFF0000
CTL$GL_IBIOCNT                          = 7FFEFEE8                 DBGPTCNT                          = 00000001
CTL$GL_ICPUTIM                          = 7FFEFED0                 ESPINI                              ********   X    01
CTL$GL_IDIOCNT                          = 7FFEFEE4                 ESTACK                            = 00000010
CTL$GL_IFAULTIO                         = 7FFEFED8                 EXE$EXCPTN                          ********   X    01
CTL$GL_IFAULTS                          = 7FFEFED4                 EXE$EXCPTNE                         ********   X    01
CTL$GL_IMGHDRBF                         = 7FFEFE80                 IAC$GL_IMAGE_LIST                   ********   X    01
CTL$GL_IMGLSTPTR                        = 7FFEFE84                 IMGACTBUF                         = 00000008
CTL$GL_IPAGEFL                          = 7FFEFEE0                 JIB$S_ACCOUNT                     = 00000008
CTL$GL_IVOLUMES                         = 7FFEFEEC                 JIB$S_USERNAME                    = 0000000C
CTL$GL_IWSPEAK                          = 7FFEFEDC                 JIB$T_ACCOUNT                     = 00000018
CTL$GL_KNOWNFIL                         = 7FFEFF64                 JIB$T_USERNAME                    = 0000000C
CTL$GL_KRPBL                            = 7FFEFFBC                 KRP_COUNT                         = 00000004
CTL$GL_KRPFL                            = 7FFEFFB8                 KSPINI                              ********   X    01
CTL$GL_KSTKBAS                            ********   X    01       KSTACK                            = 00000003
CTL$GL_LNMDIRECT                        = 7FFEFE08                 KSTACK_EX                         = 00000004
CTL$GL_LNMDIRSEQ                        = 7FFEFFA4                 LIBSWITCH                         = 00000001
CTL$GL_LNMHASH                          = 7FFEFE04                 PFILPGCNT                         = 00000001
CTL$GL_PCB                              = 7FFEFF58                 PTE$C_KOWN                        = 00000000
CTL$GL_PHD                              = 7FFEFE88                 PTE$C_URKW                        = 70000000
CTL$GL_POWERAST                         = 7FFEFF20                 PTE$M_TYP1                        = 04000000
CTL$GL_PPMSG                            = 7FFEFF48                 SSPINI                              ********   X    01
CTL$GL_PRCALLCNT                        = 7FFEFF9C                 SSTACK                            = 00000020
CTL$GL_RDIPTR                           = 7FFEFFA0                 SYS$GL_CMCNTX                     = 7FFEFF88   G
CTL$GL_RMSBASE                          = 7FFEFF44                 SYS$GL_IAFLINK                    = 7FFEFF8C   G
CTL$GL_RUF                              = 7FFEFF5C                 SYS$GL_IMGLSTPTR                  = 7FFEFE84   G
CTL$GL_SITESPEC                         = 7FFEFF60                 TMP...                              04000000
CTL$GL_THCOUNT                          = 7FFEFFC4                 USPINI                              ********   X    01
CTL$GL_THEXEC                           = 7FFEFE54                 VA...                             = 7FFEFE00
CTL$GL_THSUPR                           = 7FFEFE58                 VECTORS                           = 7FFEFE00
CTL$GL_USRCHME                          = 7FFEFF1C
CTL$GL_USRCHMK                          = 7FFEFF18
CTL$GL_USRUNDWN                         = 7FFEFF54
CTL$GL_VIRTPEAK                         = 7FFEFEC0
```

```
                                            +----------------+
                                            ! Psect synopsis !
                                            +----------------+

PSECT name                      Allocation          PSECT No.  Attr
----------                      ----------          ---------  --------
.  ABS  .                       00000000  (    0.)  00 (  0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                           FFFFFFFC  (    0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE

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

Phase                        Page faults   CPU Time       Elapsed Tim
-----                        -----------   --------       ------------
Initialization                        35   00:00:00.04    00:00:01.61
Command processing                   133   00:00:00.73    00:00:08.92
Pass 1                               374   00:00:13.74    00:00:46.85
Symbol table sort                      0   00:00:01.85    00:00:05.48
Pass 2                               266   00:00:03.96    00:00:14.94
Symbol table output                   15   00:00:00.09    00:00:00.38
Psect synopsis output                  2   00:00:00.03    00:00:00.02
Cross-reference output                 0   00:00:00.00    00:00:00.00
Assembler run totals                 827   00:00:20.44    00:01:18.21

The working set limit was 1950 pages.
73627 bytes (144 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1278 non-local and 0 local symbols.
1561 source lines were read in Pass 1, producing 8 object records in Pass 2.
35 pages of virtual memory were used to define 34 macros.

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

Macro library name                           Macros defined
------------------                           --------------
_$255$DUA28:[SHRLIB]RMS.MLB;1                         1
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                       15
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     7
TOTALS (all libraries)                               23

1313 GETS were required to define 23 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:USRVECTOR/OBJ=OBJ$:USRVECTOR MSRC$:LBSW/UPDATE=(ENH$:LBSW)+MSRC$:SHELL/UPDATE=(ENH$:SHELL)+EXECML$/LIB+SHRLIB$:RMS.ML
```

0389 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY