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
```

```
_S

Ps
--
YZ

Z$

Z$

Z$


Z$


Z$





Z$

Z$

Z$

Z$

Z$

Z$


Z$
```

```
**FILE**ID**SWAPPER                                    K  2
```

```
  SSSSSSSS  WW      WW    AAAAAA    PPPPPPPP    PPPPPPPP    EEEEEEEEEE  RRRRRRRR
  SSSSSSSS  WW      WW    AAAAAA    PPPPPPPP    PPPPPPPP    EEEEEEEEEE  RRRRRRRR
SS          WW      WW  AA      AA  PP      PP  PP      PP  EE          RR      RR
SS          WW      WW  AA      AA  PP      PP  PP      PP  EE          RR      RR
SS          WW      WW  AA      AA  PP      PP  PP      PP  EE          RR      RR
SS          WW      WW  AA      AA  PP      PP  PP      PP  EE          RR      RR
  SSSSSS    WW      WW  AA      AA  PPPPPPPP    PPPPPPPP    EEEEEEEE    RRRRRRRR
  SSSSSS    WW      WW  AA      AA  PPPPPPPP    PPPPPPPP    EEEEEEEE    RRRRRRRR
        SS  WW  WW  WW  AAAAAAAAAA  PP          PP          EE          RR  RR
        SS  WW  WW  WW  AAAAAAAAAA  PP          PP          EE          RR  RR
        SS  WWWW  WWWW  AA      AA  PP          PP          EE          RR    RR
        SS  WWWW  WWWW  AA      AA  PP          PP          EE          RR    RR        ....
SSSSSSSS    WW      WW  AA      AA  PP          PP          EEEEEEEEEE  RR      RR      ....
SSSSSSSS    WW      WW  AA      AA  PP          PP          EEEEEEEEEE  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
```

SWA
V04

Table of contents

```
     (2)       231         DECLARATIONS
     (9)       885         EXE$SWAPINIT - INITIALIZATION AND STARTUP FOR SWAPPER
     (10)     1078         SWAPPER - MAIN LOOP
     (11)     1105         BALANCE FREE PAGE COUNT
     (12)     1152         SCHEDULE SWAP
     (13)     1222         OUTSWAP
     (16)     1479         RELPHD - RELEASE PROCESS HEADER
     (17)     1573         DELPHD - DELETE PROCESS HEADER FOR DELETED PROCESS
     (18)     1619         GBLTRANS/GBLVALID/GBLWRTVALID - HANDLE GLOBAL PAGES
     (19)     1699         PROCTRANS - PROCESS PAGE IN TRANSITION
     (20)     1751         PAGE TABLE WORKING SET LIST ENTRIES
     (21)     1767         INSWAP
     (24)     2257         FILLPHD - FILL SPT ENTRIES TO MAP PHD
     (25)     2313         RELINIT - INITIALIZE REGISTERS FOR PAGE RELEASE LOOP
     (26)     2342         OSINIT - OUTSWAP SCAN REGISTER INITIALIZATION
     (27)     2366         RELPAGE - RELEASE DUPLICATE PAGE
     (28)     2401         SWPREAD/SWPWRITE - SWAPPER I/O ROUTINES
```

```
0000            .TITLE  SWAPPER  WORKING SET SWAPPER
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, SWAPPER
0000    ;
0000    ; ABSTRACT: THE SWAPPER SCHEDULES AND EXECUTES SWAPPING OF PROCESS
0000    ;           WORKING SETS BETWEEN SWAP STORAGE AND MAIN MEMORY.
0000    ;
0000    ; ENVIRONMENT:
0000    ;       MODE = KERNEL , RESIDENT
0000    ;
0000    ; AUTHOR:  R. HUSTVEDT   CREATION DATE: 30-NOV-76
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-029 ACG0440         Andrew C. Goldstein,    24-Jul-1984  10:50
0000    ;               Add ref count field to ORB
0000    ;
0000    ;       V03-028 LMP0275         L. Mark Pilant,         12-Jul-1984  20:31
0000    ;               Initialize the ACL info in the ORB to be a null descriptor
0000    ;               list rather than an empty queue.  This avoids the overhead
0000    ;               of locking and unlocking the ACL mutex, only to find out
0000    ;               that the ACL was empty.
0000    ;
0000    ;       V03-027 TMK0011         Todd M. Katz            11-Apr-1984
0000    ;               The ACL mutexes within the Object Rights Blocks of the system
0000    ;               and system directory logical name tables are currently
0000    ;               incorrectly initialized to ^X00001111. Initialize them to
0000    ;               ^X0000FFFF.
0000    ;
0000    ;       V03-026 MSH0029         Michael S. Harvey       9-Apr-1984
0000    ;               The translation of LNM$TEMPORARY_MAILBOX will now be LNM$JOB
0000    ;               instead of LNM$GROUP. This is a part of an effort to close
```

```
0000    ;               some privilege related security holes involving logical names
0000    ;               and temporary mailbox creation.
0000    ;
0000    ;       V03-025 TMK0010         Todd M. Katz            26-Mar-1984
0000    ;               Modify the logical name system services to make use of the
0000    ;               updated internal protection checking mechanisms. What this
0000    ;               involves is replacing the system directory and system logical
0000    ;               name tables' CHIP protection templates with quad-word aligned
0000    ;               Object Rights Blocks.
0000    ;
0000    ;       V03-024 TMK0009         Todd M. Katz            07-Mar-1984
0000    ;               Add a hash code field, LNMX$W_HASH, to every translation block
0000    ;               of every logical name table template defined. This hash code
0000    ;               field will be used in an optimization of logical name table name
0000    ;               PROCESSing.
0000    ;
0000    ;       V03-023 LY00b7          Larry Yetto             16-FEB-1984 14:33
0000    ;               Fix alignment of logical name tables
0000    ;
0000    ;       V03-022 ROW62094        Ralph O. Weber          25-JAN-1984
0000    ;               Add PROCESSing for inswapped global page when there currently
0000    ;               exists a equivalent global page having a page read error.
0000    ;               This makes the list of ossible conditions to be handled for an
0000    ;               inswap of a global page: 1) no equivalent global page exsts,
0000    ;               2) an equivalent global page exists, 3) the equivalent page is
0000    ;               still being read (from a page fault read), 4) the equivalent
0000    ;               page was read but encountered a page read error.
0000    ;
0000    ;       V03-021 TMK0008         Todd M. Katz            06-Jan-1984
0000    ;               Never allow the system directory logical name table to be
0000    ;               deleted. This is done as follows:
0000    ;
0000    ;               1. Set the LNMB$V_NODELETE bit within the LNMB$B_FLAGS field
0000    ;                  of the system directory logical name table.
0000    ;               2. Check for this bit within the logical name system services
0000    ;                  whenever a LNMB is to be deleted.
0000    ;               3. If this bit is set, do not allow the LNMB to be deleted;
0000    ;                  otherwise, proceed with the deletion.
0000    ;
0000    ;               This mechanism will prevent the directories from ever being
0000    ;               explicitely or implicitely deleted which can cause all sorts
0000    ;               of problems.
0000    ;
0000    ;       V03-020 TMK0007         Todd M. Katz            25-Dec-1983
0000    ;               Make a small change to TMK0006. Setup the remaining quota
0000    ;               byte field of the system directory logical name table with a
0000    ;               value of positive infinity (i.e. - ^X7FFFFFFF) instead of a
0000    ;               value of positive infinity minus the size of the system table.
0000    ;               This is necessary because the routine which is used to
0000    ;               appropriately insert the system table performs the necessary
0000    ;               quota subtractions; thus, in TMK0006 quota for the system
0000    ;               table was being subtracted twice from the system directory
0000    ;               logical name table.
0000    ;
0000    ;       V03-019 TMK0006         Todd M. Katz            18-Dec-1983
0000    ;               Handcraft the system logical name table, LNM$SYSTEM_TABLE,
0000    ;               instead of using the system service ($CRELNT) to CREATE it.
```

```
0000  ;
0000  ;       V03-018 WMC0018         Wayne Cardoza           02-Dec-1983
0000  ;               PHD$W_WSLX, PHD$W_BAK have become longwords.
0000  ;
0000  ;       V03-017 TMK0005         Todd M. Katz            19-Oct-1983
0000  ;               Add the following kernel mode logical names which will be
0000  ;               used in order to optimize $TRNLOGs:
0000  ;
0000  ;               TRNLOG$_PROCESS_GROUP
0000  ;               TRNLOG$_PROCESS_SYSTEM
0000  ;               TRNLOG$_GROUP_SYSTEM
0000  ;               TRNLOG$_PROCESS_GROUP_SYSTEM
0000  ;
0000  ;               Also, CREATE LNM$TEMPORARY_MAILBOX with a translation of
0000  ;               LNM$GROUP instead of LNM$JOB.
0000  ;
0000  ;       V03-016 TMK0004         Todd M. Katz            11-Oct-1983
0000  ;               Make the following changes to the logical names and tables
0000  ;               that are CREATEd at system initialization time:
0000  ;
0000  ;               1. CREATE LNM$FILE_DEV with the translations
0000  ;                  LNM$PROCESS,LNM$JOB,LNM$GROUP,LNM$SYSTEM (the change is the
0000  ;                  addition of the LNM$JOB translation).
0000  ;               2. CREATE LNM$TEMPORARY_MAILBOX with the translation LNM$JOB.
0000  ;               3. Remove LNM$TRNLOG_PG, LNM$TRNLOG_PS, LNM$TRNLOG_GS,
0000  ;                  LNM$TRNLOG_PGS, and LNM$DEFAULT_SEARCH.
0000  ;               4. Add the following kernel mode logical names which will be
0000  ;                  used in order to provide compatibility between V3 and V4
0000  ;                  for all of the old logical name system services ($TRNLOG,
0000  ;                  $CRELOG, $DELLOG):
0000  ;
0000  ;                  LOG$PROCESS
0000  ;                  LOG$GROUP
0000  ;                  LOG$SYSTEM
0000  ;
0000  ;               5. Add to the PQB$AB_SYSPQL quota list a PQL$_JTQUOTA item.
0000  ;
0000  ;       V03-015 TMK0003         Todd M. Katz            09-Apr-1983
0000  ;               Statically define the CHIP protection structure of
0000  ;               LNM$SYSTEM_DIRECTORY, set the (internal) attribute bit
0000  ;               LNM$V_SYSTEM when creating LNM$SYSTEM_TABLE, and change the
0000  ;               CHIP protection of LNM$SYSTEM_DIRECTORY and LNM$SYSTEM_TABLE to
0000  ;               S:RWE O:RWE G:R W:R. Also, CREATE the supervisor mode logical
0000  ;               name LNM$FILE_DEV with the translations LNM$PROCESS,LNM$GROUP,
0000  ;               LNM$SYSTEM instead of LNM$DEFAULT_SEARCH, mark both
0000  ;               translations of LNM$DIRECTORIES with the TERMINAL attribute,
0000  ;               and CREATE the non-aliasable kernal mode logical names
0000  ;               LNM$TRNLOG_PG,LNM$TRNLOG_PS,LNM$TRNLOG_GS,LNM$TRNLOG_PGS.
0000  ;
0000  ;       V03-014 KDM0052         Kathleen D. Morse       11-Jul-1983
0000  ;               Replace references of PR$_TODR with EXE$GQ_SYSTIME+2.
0000  ;
0000  ;       V03-013 DMW4060         DMWalp                  23-Jun-1983
0000  ;               Change $xxLNM value parameters to be by reference
0000  ;
0000  ;       V03-012 DMW4054         DMWalp                  21-Jun-1983
0000  ;               Convert SYS$DISK and SYS$SYSDEVICE creation from $CRELOG
```

```
0000  ;               to $CRELNM
0000  ;
0000  ;       V03-011 RAS0158          Ron Schaefer              23-May-1983
0000  ;               Add CHIP protection structure to the logical name structures.
0000  ;               Protection stuff only supports SOGW checking for now.
0000  ;               Fix quota for LNM$SYSTEM_TABLE.
0000  ;
0000  ;       V03-010 TMK0002          Todd M. Katz              26-Apr-1983
0000  ;               CREATE the following logical name structures at system
0000  ;               initialization time:
0000  ;
0000  ;               1. LNM$SYSTEM_TABLE.
0000  ;               2. LNM$SYSTEM.
0000  ;               3. LNM$FILE_DEV (Executive Mode).
0000  ;               4. LNM$FILE_DEV (Supervisor Mode).
0000  ;               5. LNM$DEFAULT_SEARCH.
0000  ;               6. LNM$TEMPORARY_MAILBOX.
0000  ;               7. LNM$PERMANENT_MAILBOX.
0000  ;               8. LNM$DIRECTORIES.
0000  ;
0000  ;               Change the name of LNT$SYSTEM_DIRECTORY to LNM$SYSTEM_DIRECTORY.
0000  ;
0000  ;       V03-009 TMK0001          Todd M. Katz              14-Apr-1983
0000  ;               Make the following changes to the system directory logical
0000  ;               name table:
0000  ;
0000  ;               1. Make the table a kernel access mode table.
0000  ;               2. Make LNMB$L_TABLE point to the system directory table's
0000  ;                  table header.
0000  ;               3. Set the bits LNMTH$V_SHAREABLE and LNMTH$V_DIRECTORY within
0000  ;                  LNMTH$B_FLAGS.
0000  ;               4. Delete the field LNMTH$L_LOGNAM.
0000  ;
0000  ;       V03-008 HRJ0200          Herb Jacobs               05-Feb-1983
0000  ;               Add check to BALANCE to remove confusion as to why
0000  ;               swapper has woken up.  If there are FREELIM pages on
0000  ;               Freelist, then don't acquire FREEGOAL pages, but rather
0000  ;               perform requested function woken up for.
0000  ;
0000  ;       V03-007 DMW4020          DMWalp                    30-Dec-1982
0000  ;               Added creation system logical directory.
0000  ;
0000  ;       V03-006 DMW4019          DMWalp                    15-Dec-1982
0000  ;               Calculate LNM hash table parameters and CREATE hash table.
0000  ;
0000  ;       V03-005 DMW4006          DMWalp                    10-NOV-1982
0000  ;               Recode creation SYS$DISK and SYS$SYSDEVICE to use
0000  ;               external interface ( not internal ) of $CRELOG
0000  ;
0000  ;       V03-004 HRJ0101          Herb Jacobs               30-Jun-1982
0000  ;               Add perturbation to balance set slot scanner to try to
0000  ;               alleviate deadlocks caused there if seemingly the
0000  ;               best swapper action is to try to free PROCESS waited
0000  ;               for service from an outswapped PROCESS.
0000  ;
0000  ;
0000  ;--
```

```
0000            .SBTTL  DECLARATIONS
0000    ;
0000    ; INCLUDE FILES:
0000    ;
0000
0000            $ACBDEF                         ; DEFINE AST CONTROL BLOCK OFFSETS
0000            $DYNDEF                         ; DEFINE STRUCTURE TYPE CODES
0000            $IPLDEF                         ; DEFINE INTERRUPT PRIORITY LEVELS
0000            $IRPDEF                         ; DEFINE I/O REQUEST PACKET OFFSETS
0000            $LNMDEF                         ; DEFINE LOGICAL NAME OFFSETS
0000            $LNMSTRDEF                      ; DEFINE LOGICAL NAME STRUCTURE OFFSETS
0000            $OPDEF                          ; DEFINE OPCODE EQUIVALENT VALUES
0000            $ORBDEF                         ; DEFINE OBJECT RIGHTS BLOCK OFFSETS
0000            $PCBDEF                         ; DEFINE PCB OFFSETS
0000            $PFLDEF                         ; DEFINE SWAP FILE TABLE OFFSETS
0000            $PFNDEF                         ; DEFINE PFN VALUES
0000            $PHDDEF                         ; DEFINE PHD OFFSETS
0000            $PQLDEF                         ; DEFINE QUOTA SYMBOLS
0000            $PRDEF                          ; DEFINE PROCESSOR REGISTERS
0000            $PRCDEF                         ; CREATE PROCESS FLAGS
0000            $PSLDEF                         ; DEFINE PSL VALUES
0000            $PTEDEF                         ; DEFINE PAGE TABLE ENTRY
0000            $VADEF                          ; DEFINE VIRTUAL ADDRESS FIELDS
0000            $WSLDEF                         ; DEFINE WORKING SET LIST BITS
0000
0000    ;
0000    ; ASSUMPTIONS ABOUT THE STRUCTURE OF LOGICAL NAME AND OBJECT RIGHTS BLOCKS:
0000    ;
0000
0000            ASSUME  LNMB$L_FLINK,           EQ,     0
0000            ASSUME  LNMB$L_FLINK+4,         EQ,     LNMB$L_BLINK
0000            ASSUME  LNMB$L_BLINK+4,         EQ,     LNMB$W_SIZE
0000            ASSUME  LNMB$W_SIZE+2,          EQ,     LNMB$B_TYPE
0000            ASSUME  LNMB$B_TYPE+1,          EQ,     LNMB$B_ACMODE
0000            ASSUME  LNMB$B_ACMODE+1,        EQ,     LNMB$L_TABLE
0000            ASSUME  LNMB$L_TABLE+4,         EQ,     LNMB$B_FLAGS
0000            ASSUME  LNMB$B_FLAGS+1,         EQ,     LNMB$T_NAME
0000
0000            ASSUME  LNMX$B_FLAGS,           EQ,     0
0000            ASSUME  LNMX$B_FLAGS+1,         EQ,     LNMX$B_INDEX
0000            ASSUME  LNMX$B_INDEX+1,         EQ,     LNMX$W_HASH
0000            ASSUME  LNMX$W_HASH+2,          EQ,     LNMX$T_XLATION
0000
0000            ASSUME  LNMTH$B_FLAGS,          EQ,     0
0000            ASSUME  LNMTH$B_FLAGS+1,        EQ,     LNMTH$L_HASH
0000            ASSUME  LNMTH$L_HASH+4,         EQ,     LNMTH$L_ORB
0000            ASSUME  LNMTH$L_ORB+4,          EQ,     LNMTH$L_NAME
0000            ASSUME  LNMTH$L_NAME+4,         EQ,     LNMTH$L_PARENT
0000            ASSUME  LNMTH$L_PARENT+4,       EQ,     LNMTH$L_CHILD
0000            ASSUME  LNMTH$L_CHILD+4,        EQ,     LNMTH$L_SIBLING
0000            ASSUME  LNMTH$L_SIBLING+4,      EQ,     LNMTH$L_QTABLE
0000            ASSUME  LNMTH$L_QTABLE+4,       EQ,     LNMTH$L_BYTESLM
0000            ASSUME  LNMTH$L_BYTESLM+4,      EQ,     LNMTH$L_BYTES
0000
0000            ASSUME  ORB$L_OWNER,            EQ,     0
0000            ASSUME  ORB$L_OWNER+4,          EQ,     ORB$L_ACL_MUTEX
0000            ASSUME  ORB$L_ACL_MUTEX+4,      EQ,     ORB$W_SIZE
```

```
0000         ASSUME  ORB$W_SIZE+2,       EQ,  ORB$B_TYPE
0000         ASSUME  ORB$B_TYPE+1,       EQ,  ORB$B_FLAGS
0000         ASSUME  ORB$B_FLAGS+3,      EQ,  ORB$W_REFCOUNT
0000         ASSUME  ORB$W_REFCOUNT+2,   EQ,  ORB$Q_MODE_PROT
0000         ASSUME  ORB$Q_MODE_PROT+8,  EQ,  ORB$L_SYS_PROT
0000         ASSUME  ORB$L_SYS_PROT+4,   EQ,  ORB$L_OWN_PROT
0000         ASSUME  ORB$L_OWN_PROT+4,   EQ,  ORB$L_GRP_PROT
0000         ASSUME  ORB$L_GRP_PROT+4,   EQ,  ORB$L_WOR_PROT
0000         ASSUME  ORB$L_WOR_PROT+4,   EQ,  ORB$L_ACL_COUNT
0000         ASSUME  ORB$L_ACL_COUNT+4,  EQ,  ORB$L_ACL_DESC
0000         ASSUME  ORB$L_ACL_DESC+4,   EQ,  ORB$R_MIN_CLASS
0000         ASSUME  ORB$R_MIN_CLASS+ORB$S_MIN_CLASS,-
0000                                     EQ,  ORB$R_MAX_CLASS
0000         ASSUME  ORB$R_MAX_CLASS+ORB$S_MAX_CLASS,-
0000                                     EQ,  ORB$K_LENGTH
```

```
                                 0000
                                 0000  ;
                                 0000  ; OWN STORAGE:
                                 0000  ;
                                 0000
                             00000000          .PSECT  $$$220,LONG              ; SWAPPER/SCHEDULER WRITABLE DATA
                                 0000  IOROUTINE:                               ; ADDRESS OF PROPER BUILD PACKET ROUTINE
                     00000000    0000          .LONG   0                        ;
                     00000000    0004  IOEA:   .LONG   0                        ; I/O END ACTION RETURN
                     00000000    0008  RWSSWP: .LONG   0                        ; REMAINING WS SWP ADDRESS
                     00000000    000C  RSVAPTE:.LONG   0                        ; REMAINING SVA OF PTE
                         0000    0010  RPGCNT: .WORD   0                        ; REMAINING PAGE COUNT
                         0000    0012  OSWPPGS:.WORD   0                        ; OUTSWAP PAGE COUNT
                     00000000    0014  OSWPPCB:.LONG   0                        ; PCB ADDRESS OF OUTSWAP PROCESS
                                 0018  SWP$GW_BALCNT::                          ; COUNT OF PROCESSES IN BALANCE SET
                         FFFF    0018          .WORD   -1                       ; EXCLUDING NULL PROCESS AND SWAPPER
                                 001A  SCH$GW_SWPFCNT::                         ; COUNT OF SUCCESSIVE SWAP
                         0000    001A          .WORD   0                        ; SCHEDULE FAILURES.
                                 001C
                             00000000          .PSECT  $$$260,5                 ; WRITABLE, HIGH USE PSECT
                                 0000
                                 0000  ;
                                 0000  ; LNM$SYSTEM_DIRECTORY - THE SYSTEM DIRECTORY LOGICAL NAME TABLE.
                                 0000  ;
                                 0000
                                 0000  LNM$SYSTEM_DIRECTORY::
                     00000000    0000          .LONG   0                        ; FORWARD LINK
                     00000000    0004          .LONG   0                        ; BACK LINK
                        00C0'    0008          .WORD   LNM_SYS_DIR_SIZ          ; SIZE OF STRUCTURE
                           40    000A          .BYTE   DYN$C_LNM                ; TYPE OF STRUCTURE
                           00    000B          .BYTE   PSL$C_KERNEL             ; KERNEL ACCESS MODE
                    0000002B'    000C          .ADDRESS LNM_SYSTEM_DIR_LNMTH    ; DIRECTORY TABLE HEADER ADDRESS
                           19    0010          .BYTE   LNMB$M_NO_ALIAS!-        ; DIRECTORY TABLES CAN NOT BE ALIASED
                                 0011                  LNMB$M_TABLE!-           ; DIRECTORIES ARE TABLES
                                 0011                  LNMB$M_NODELETE          ; DIRECTORIES CAN NOT BE DELETED
5F 4D 45 54 53 59 53 24 4D 4E 4C 00'    0011          .ASCIC  "LNM$SYSTEM_DIRECTORY"   ; NAME OF DIRECTORY TABLE
         59 52 4F 54 43 45 52 49 44    001D
                           14    0011
                                 0026
                           02    0026          .BYTE   LNMX$M_TERMINAL          ; FLAGS BYTE. NO MORE TRANSLATIONS
                           82    0027          .BYTE   LNMX$C_TABLE             ; TRANSLATION INDEX ( SPECIAL TABLE )
                         0000    0028          .WORD   0                        ; TRANSLATION HASH CODE
                           25    002A          .BYTE   LNMTH$K_LENGTH           ; SIZE OF TABLE HEADER BLOCK
                                 002B
                                 002B  LNM_SYSTEM_DIR_LNMTH::
                           03    002B          .BYTE   LNMTH$M_SHAREABLE!-      ; DIRECTORY IS A SHAREABLE TABLE
                                 002C                  LNMTH$M_DIRECTORY        ; TABLE IS A DIRECTORY TABLE
                     00000000    002C          .LONG   0                        ; ADDRESS OF HASH TABLE
                    00000058'    0030          .ADDRESS LNM_SYSTEM_DIR_ORB      ; ADDRESS OF OBJECT RIGHTS BLOCK
                    00000000'    0034          .ADDRESS LNM$SYSTEM_DIRECTORY    ; ADDRESS OF CONTAINING LNMB BLOCK
                     00000000    0038          .LONG   0                        ; ADDRESS OF PARENT TABLE
                     00000000    003C          .LONG   0                        ; ADDRESS OF CHILD TABLE
                     00000000    0040          .LONG   0                        ; ADDRESS OF SIBLING TABLE
                    0000002B'    0044          .ADDRESS LNM_SYSTEM_DIR_LNMTH    ; ADDRESS OF TABLE HOLDING QUOTA
                     7FFFFFFF    0048          .LONG   ^X7FFFFFFF               ; INITIAL QUOTA   ( POSITIVE INFINITY )
                     7FFFFFFF    004C          .LONG   ^X7FFFFFFF               ; REMAINING QUOTA ( POSITIVE INFINITY )
                                 0050
```

```
                              04  0050   359             .BYTE   LNMX$M_XEND                  ; FLAGS BYTE. NO MORE TRANSLATIONS
                                  0051   360
                                  0051   361             .ALIGN  QUAD
                                  0058   362 LNM_SYSTEM_DIR_ORB:
                        00010004  0058   363             .LONG   ^X00010004                   ; SYSTEM DIRECTORY OWNER IS [1,4]
                       0000 FFFF  005C   364             .WORD   -1,0                         ; INITIALIZE ACL MUTEX
                           0068'  0060   365             .WORD   LNM_SYS_DIR_ORB_SIZ          ; SIZE OF OBJECT RIGHTS BLOCK
                              49  0062   366             .BYTE   DYN$C_ORB                    ; BLOCK TYPE
                              00  0063   367             .BYTE   0                            ; NO ACL AS YET
                        00000000  0064   368             .LONG   0                            ; ZERO RESERVED WORD & REF COUNT
               00000000 00000000  0068   369             .QUAD   0                            ; OBJECT DOES NOT HAVE AN ACCESS MODE
                        00000008  0070   370             .LONG   ^X00000008                   ; SYSTEM PROTECTION IS RWE
                        00000008  0074   371             .LONG   ^X00000008                   ; OWNER PROTECTION IS RWE
                        0000000E  0078   372             .LONG   ^X0000000E                   ; GROUP PROTECTION IS R
                        0000000E  007C   373             .LONG   ^X0000000E                   ; WORLD PROTECTION IS R
               00000000 00000000  0080   374             .LONG   0,0                          ; NULL INITIAL ACL
00'00'00'00'00'00'00'00'00'00'00' 0088   375             .BYTE   0[ORB$S_MIN_CLASS]           ; MINIMUM CLASSIFICATION MASK
         00'00'00'00'00'00'00'00' 0094
00'00'00'00'00'00'00'00'00'00'00' 009C   376             .BYTE   0[ORB$S_MAX_CLASS]           ; MAXIMUM CLASSIFICATION MASK
         00'00'00'00'00'00'00'00' 00A8
                                  00B0   377
                                  00B0   378             .ALIGN  5
                        00000068  00C0   379 LNM_SYS_DIR_ORB_SIZ = . - LNM_SYSTEM_DIR_ORB
                        000000C0  00C0   380 LNM_SYS_DIR_SIZ = . - LNM$SYSTEM_DIRECTORY
                                  00C0   381
                                  00C0   382 ;
                                  00C0   383 ; LNM$SYSTEM_TABLE - THE SYSTEM LOGICAL NAME TABLE.
                                  00C0   384 ;
                                  00C0   385
                                  00C0   386 SYSTEM_TABLE:
                        00000000  00C0   387             .LONG   0                            ; FORWARD LINK
                        00000000  00C4   388             .LONG   0                            ; BACK LINK
                           00C0'  00C8   389             .WORD   SYSTEM_TABLE_SIZE            ; SIZE OF STRUCTURE
                              40  00CA   390             .BYTE   DYN$C_LNM                    ; TYPE OF STRUCTURE
                              00  00CB   391             .BYTE   PSL$C_KERNEL                 ; KERNEL ACCESS MODE
                       0000002B'  00CC   392             .ADDRESS LNM_SYSTEM_DIR_LNMTH        ; DIRECTORY TABLE HEADER ADDRESS
                              09  00D0   393             .BYTE   LNMB$M_NO_ALIAS!-            ; TABLE CAN NOT BE ALIASED
                                  00D1   394                     LNMB$M_TABLE                 ; TABLE
5F 4D 45 54 53 59 53 24 4D 4E 4C 00'  00D1   395             .ASCIC  "LNM$SYSTEM_TABLE"       ; TABLE NAME
                     45 4C 42 41 54  00DD
                                 10  00D1
                                  00E2   396
                              02  00E2   397             .BYTE   LNMX$M_TERMINAL              ; FLAGS BYTE. NO MORE TRANSLATIONS
                              82  00E3   398             .BYTE   LNMX$C_TABLE                 ; TRANSLATION INDEX ( SPECIAL TABLE )
                            0000  00E4   399             .WORD   0                            ; TRANSLATION HASH CODE
                              25  00E6   400             .BYTE   LNMTH$K_LENGTH               ; SIZE OF TABLE HEADER BLOCK
                                  00E7   401
                                  00E7   402 SYSTEM_TABLE_LNMTH:
                              09  00E7   403             .BYTE   LNMTH$M_SHAREABLE!-          ; TABLE IS SHAREABLE
                                  00E8   404                     LNMTH$M_SYSTEM               ; THIS IS THE SYSTEM LOGICAL NAME TABLE
                        00000000  00E8   405             .LONG   0                            ; ADDRESS OF HASH TABLE
                       00000110'  00EC   406             .ADDRESS SYSTEM_TABLE_ORB            ; ADDRESS OF OBJECT RIGHTS BLOCK
                       000000C0'  00F0   407             .ADDRESS SYSTEM_TABLE                ; ADDRESS OF CONTAINING LNMB BLOCK
                       0000002B'  00F4   408             .ADDRESS LNM_SYSTEM_DIR_LNMTH        ; ADDRESS OF PARENT TABLE
                        00000000  00F8   409             .LONG   0                            ; ADDRESS OF CHILD TABLE
                        00000000  00FC   410             .LONG   0                            ; ADDRESS OF SIBLING TABLE
                       0000002B'  0100   411             .ADDRESS LNM_SYSTEM_DIR_LNMTH        ; ADDRESS OF TABLE HOLDING QUOTA
```

```
                          00000000  0104   412             .LONG   0                       ; INITIAL QUOTA   ( POOLED )
                          00000000  0108   413             .LONG   0                       ; REMAINING QUOTA ( POOLED )
                                    010C   414
                                04  010C   415             .BYTE   LNMX$M_XEND             ; FLAGS BYTE. NO MORE TRANSLATIONS
                                    010D   416
                                    010D   417             .ALIGN  QUAD
                                    0110   418 SYSTEM_TABLE_ORB:
                          00010004  0110   419             .LONG   ^X00010004              ; SYSTEM TABLE OWNER IS [1,4]
                         0000 FFFF  0114   420             .WORD   -1,0                    ; INITIALIZE ACL MUTEX
                             0070'  0118   421             .WORD   SYSTEM_TABLE_ORB_SIZ    ; SIZE OF OBJECT RIGHTS BLOCK
                                49  011A   422             .BYTE   DYN$C_ORB               ; BLOCK TYPE
                                00  011B   423             .BYTE   0                       ; NO ACL AS YET
                          00000000  011C   424             .LONG   0                       ; ZERO RESERVED WORD & REF COUNT
                 00000000 00000000  0120   425             .QUAD   0                       ; OBJECT DOES NOT HAVE AN ACCESS MODE
                          00000008  0128   426             .LONG   ^X00000008              ; SYSTEM PROTECTION IS RWE
                          00000008  012C   427             .LONG   ^X00000008              ; OWNER PROTECTION IS RWE
                          0000000E  0130   428             .LONG   ^X0000000E              ; GROUP PROTECTION IS R
                          0000000E  0134   429             .LONG   ^X0000000E              ; WORLD PROTECTION IS R
                 00000000 00000000  0138   430             .LONG   0,0                     ; NULL INITIAL ACL
00'00'00'00'00'00'00'00'00'00'00'00'  0140   431             .BYTE   0[ORB$S_MIN_CLASS]      ; MINIMUM CLASSIFICATION MASK
         00'00'00'00'00'00'00'00'00'  014C
00'00'00'00'00'00'00'00'00'00'00'00'  0154   432             .BYTE   0[ORB$S_MAX_CLASS]      ; MAXIMUM CLASSIFICATION MASK
         00'00'00'00'00'00'00'00'00'  0160
                                    0168   433
                                    0168   434             .ALIGN  5
                          00000070  0180   435 SYSTEM_TABLE_ORB_SIZ = . - SYSTEM_TABLE_ORB
                          000000C0  0180   436 SYSTEM_TABLE_SIZE = . - SYSTEM_TABLE
```

```
                                          0180   438
                                      00000000   439              .PSECT  YF$LOWUSE                  ; PAGED PSECT AT END OF SYS.EXE
                                          0000   440
49 4E 49 53 59 53 00000008'010E0000'     0000   441  IMGDESC:.ASCID  /SYSINIT.EXE/              ; SYSTEM INITIALIZATION PROCESS
                      45 58 45 2E 54     000E
   3A 30 41 50 4F 0000001B'010E0000'     0013   442  TTODESC:.ASCID  /OPA0:/
                                          0020   443
                                          0020   444  ;
                                          0020   445  ; DESCRIPTORS AND CHARACTER STRING BUFFERS FOR THE LOGICAL NAME TABLE NAMES,
                                          0020   446  ; LOGICAL NAMES, AND LOGICAL NAME EQUIVALENCE STRINGS THAT ARE CREATED AT
                                          0020   447  ; SYSTEM INITIALIZATION TIME.
                                          0020   448  ;
                                          0020   449
                                          0020   450  LNM_DIRECTORIES_DESC:
49 44 24 4D 4E 4C 00000028'010E0000'     0020   451          .ASCID  /LNM$DIRECTORIES/
         53 45 49 52 4F 54 43 45 52     002E
                                          0037   452
                                          0037   453  LNM_FILE_DEV_DESC:
49 46 24 4D 4E 4C 0000003F'010E0000'     0037   454          .ASCID  /LNM$FILE_DEV/
                   56 45 44 5F 45 4C     0045
                                          004B   455
                                          004B   456  LNM_PERMANENT_MAILBOX_DESC:
45 50 24 4D 4E 4C 00000053'010E0000'     004B   457          .ASCID  /LNM$PERMANENT_MAILBOX/
4C 49 41 4D 5F 54 4E 45 4E 41 4D 52     0059
                            58 4F 42     0065
                                          0068   458
                                          0068   459  LNM_SYSTEM_DESC:
                            0000000A'     0068   460          .LONG   LNM_SYSTEM_LENGTH
                            0000017D'     006C   461          .ADDRESS LNM_SYSTEM
                                          0070   462
                                          0070   463  LNM_SYSTEM_DIRECTORY_DESC:
                            00000014'     0070   464          .LONG   LNM_SYSTEM_DIRECTORY_LENGTH
                            00000187'     0074   465          .ADDRESS LNM_SYSTEM_DIRECTORY
                                          0078   466
                                          0078   467  LNM_TEMPORARY_MAILBOX_DESC:
45 54 24 4D 4E 4C 00000080'010E0000'     0078   468          .ASCID  /LNM$TEMPORARY_MAILBOX/
4C 49 41 4D 5F 59 52 41 52 4F 50 4D     0086
                            58 4F 42     0092
                                          0095   469
                                          0095   470  LOG_G_DESC:
                            00000009'     0095   471          .LONG   LOG_GROUP_LENGTH
                            000001AB'     0099   472          .ADDRESS LOG_GROUP
                                          009D   473
                                          009D   474  LOG_P_DESC:
                            0000000B'     009D   475          .LONG   LOG_PROCESS_LENGTH
                            000001B4'     00A1   476          .ADDRESS LOG_PROCESS
                                          00A5   477
                                          00A5   478  LOG_S_DESC:
                            0000000A'     00A5   479          .LONG   LOG_SYSTEM_LENGTH
                            000001BF'     00A9   480          .ADDRESS LOG_SYSTEM
                                          00AD   481
                                          00AD   482  SYS_DISK_DESC:
49 44 24 53 59 53 000000B5'010E0000'     00AD   483          .ASCID  /SYS$DISK/
                               4B 53     00BB
                                          00BD   484
                                          00BD   485  SYS_SYSDEVICE_DESC:
59 53 24 53 59 53 000000C5'010E0000'     00BD   486          .ASCID  /SYS$SYSDEVICE/
```

```
             45 43 49 56 45 44 53  00CB
                                   00D2
                                   00D2  TRNLOG_GS_DESC:
47 4F 4C 4E 52 54 000000DA'010E0000'  00D2          .ASCID  /TRNLOG$_GROUP_SYSTEM/
54 53 59 53 5F 50 55 4F 52 47 5F 24  00E0
                              4D 45  00EC
                                   00EE
                                   00EE  TRNLOG_PG_DESC:
47 4F 4C 4E 52 54 000000F6'010E0000'  00EE          .ASCID  /TRNLOG$_PROCESS_GROUP/
52 47 5F 53 53 45 43 4F 52 50 5F 24  00FC
                           50 55 4F  0108
                                   010B
                                   010B  TRNLOG_PS_DESC:
47 4F 4C 4E 52 54 00000113'010E0000'  010B          .ASCID  /TRNLOG$_PROCESS_SYSTEM/
59 53 5F 53 53 45 43 4F 52 50 5F 24  0119
                        4D 45 54 53  0125
                                   0129
                                   0129  TRNLOG_PGS_DESC:
47 4F 4C 4E 52 54 00000131'010E0000'  0129          .ASCID  /TRNLOG$_PROCESS_GROUP_SYSTEM/
52 47 5F 53 53 45 43 4F 52 50 5F 24  0137
      4D 45 54 53 59 53 5F 50 55 4F  0143
                                   014D
                                   014D  LNM_GROUP:
         50 55 4F 52 47 24 4D 4E 4C  014D          .ASCII  /LNM$GROUP/
                           00000009  0156  LNM_GROUP_LENGTH = . - LNM_GROUP
                                   0156
                                   0156  LNM_JOB:
               42 4F 4A 24 4D 4E 4C  0156          .ASCII  /LNM$JOB/
                           00000007  015D  LNM_JOB_LENGTH  = . - LNM_JOB
                                   015D
                                   015D  LNM_PROCESS:
   53 53 45 43 4F 52 50 24 4D 4E 4C  015D          .ASCII  /LNM$PROCESS/
                           0000000B  0168  LNM_PROCESS_LENGTH = . - LNM_PROCESS
                                   0168
                                   0168  LNM_PROCESS_DIRECTORY:
5F 53 53 45 43 4F 52 50 24 4D 4E 4C  0168          .ASCII  /LNM$PROCESS_DIRECTORY/
         59 52 4F 54 43 45 52 49 44  0174
                           00000015  017D  LNM_PROCESS_DIRECTORY_LENGTH = . - LNM_PROCESS_DIRECTORY
                                   017D
                                   017D  LNM_SYSTEM:
      4D 45 54 53 59 53 24 4D 4E 4C  017D          .ASCII  /LNM$SYSTEM/
                           0000000A  0187  LNM_SYSTEM_LENGTH = . - LNM_SYSTEM
                                   0187
                                   0187  LNM_SYSTEM_DIRECTORY:
44 5F 4D 45 54 53 59 53 24 4D 4E 4C  0187          .ASCII  /LNM$SYSTEM_DIRECTORY/
            59 52 4F 54 43 45 52 49  0193
                           00000014  019B  LNM_SYSTEM_DIRECTORY_LENGTH = . - LNM_SYSTEM_DIRECTORY
                                   019B
                                   019B  LNM_SYSTEM_TABLE:
54 5F 4D 45 54 53 59 53 24 4D 4E 4C  019B          .ASCII  /LNM$SYSTEM_TABLE/
                        45 4C 42 41  01A7
                           00000010  01AB  LNM_SYSTEM_TABLE_LENGTH = . - LNM_SYSTEM_TABLE
                                   01AB
                                   01AB  LOG_GROUP:
         50 55 4F 52 47 24 47 4F 4C  01AB          .ASCII  /LOG$GROUP/
                           00000009  01B4  LOG_GROUP_LENGTH = . - LOG_GROUP
                                   01B4
```

```
                                  01B4  LOG_PROCESS:
 53 53 45 43 4F 52 50 24 47 4F 4C 01B4          .ASCII  /LOG$PROCESS/
                         0000000B 01BF  LOG_PROCESS_LENGTH = . - LOG_PROCESS
                                  01BF
                                  01BF  LOG_SYSTEM:
    4D 45 54 53 59 53 24 47 4F 4C 01BF          .ASCII  /LOG$SYSTEM/
                         0000000A 01C9  LOG_SYSTEM_LENGTH = . - LOG_SYSTEM
                                  01C9
                                  01C9  ;
                                  01C9  ; ATTRIBUTE, ACCESS MODE AND ITEM BUFFERS WHICH ARE PASSED BY REFERENCE.
                                  01C9  ;
                                  01C9
                                  01C9  EXEC_MODE:                                    ; EXECUTIVE ACCESS MODE BUFFER
                         00000001 01C9          .LONG   PSL$C_EXEC
                                  01CD
                                  01CD  KERNEL_MODE:                                  ; KERNEL ACCESS MODE BUFFER
                         00000000 01CD          .LONG   PSL$C_KERNEL
                                  01D1
                                  01D1  SUPER_MODE:                                   ; SUPERVISOR ACCESS MODE BUFFER
                         00000002 01D1          .LONG   PSL$C_SUPER
                                  01D5
                                  01D5  LNM_NO_ALIAS:                                 ; NO_ALIAS ATTRIBUTE BUFFER
                         00000001 01D5          .LONG   LNM$M_NO_ALIAS
                                  01D9
                                  01D9  TERMINAL_BUFFER:                              ; TERMINAL ATTRIBUTES ITEM BUFFER
                         00000200 01D9          .LONG   LNM$M_TERMINAL
                                  01DD
                                  01DD  ;
                                  01DD  ; ITEM LISTS FOR THE CREATION OF THE LOGICAL NAMES SETUP AT SYSTEM
                                  01DD  ; INITIALIZATION TIME.
                                  01DD  ;
                                  01DD
                                  01DD  DIRECTORIES_LIST:                             ; ITEM LIST FOR LNM$DIRECTORIES
                             0004 01DD          .WORD   4                             ; TERMINAL ATTRIBUTES ITEM
                             0003 01DF          .WORD   LNM$_ATTRIBUTES
                        000001D9' 01E1          .ADDRESS TERMINAL_BUFFER
                         00000000 01E5          .LONG   0
                                  01E9
                             0015 01E9          .WORD   LNM_PROCESS_DIRECTORY_LENGTH ; LNM$PROCESS_DIRECTORY STRING ITEM
                             0002 01EB          .WORD   LNM$_STRING
                        00000168' 01ED          .ADDRESS LNM_PROCESS_DIRECTORY
                         00000000 01F1          .LONG   0
                                  01F5
                             0014 01F5          .WORD   LNM_SYSTEM_DIRECTORY_LENGTH ; LNM$SYSTEM_DIRECTORY STRING ITEM
                             0002 01F7          .WORD   LNM$_STRING
                        00000187' 01F9          .ADDRESS LNM_SYSTEM_DIRECTORY
                         00000000 01FD          .LONG   0
                                  0201
                         00000000 0201          .LONG   0                             ; END OF ITEM LIST
                                  0205
                                  0205  FILE_DEV_SUPER_LIST:                          ; ITEM LIST FOR SUPERVISOR LNM$FILE_DEV
                             000B 0205          .WORD   LNM_PROCESS_LENGTH            ; LNM$PROCESS STRING ITEM
                             0002 0207          .WORD   LNM$_STRING
                        0000015D' 0209          .ADDRESS LNM_PROCESS
                         00000000 020D          .LONG   0
                                  0211
                             0007 0211          .WORD   LNM_JOB_LENGTH                ; LNM$JOB STRING ITEM
```

```
     0002  0213                .WORD   LNM$_STRING
 00000156' 0215                .ADDRESS LNM_JOB
 00000000  0219                .LONG   0
           021D
     0009  021D                .WORD   LNM_GROUP_LENGTH        ; LNM$GROUP STRING ITEM
     0002  021F                .WORD   LNM$_STRING
 0000014D' 0221                .ADDRESS LNM_GROUP
 00000000  0225                .LONG   0
           0229
     000A  0229                .WORD   LNM_SYSTEM_LENGTH       ; LNM$SYSTEM STRING ITEM
     0002  022B                .WORD   LNM$_STRING
 0000017D' 022D                .ADDRESS LNM_SYSTEM
 00000000  0231                .LONG   0
           0235
 00000000  0235                .LONG   0                       ; END OF ITEM LIST
           0239
           0239        FILE_DEV_EXEC_LIST:                     ; ITEM LIST FOR EXECUTIVE LNM$FILE_DEV
           0239        PERMANENT_MAILBOX_LIST:                 ; ITEM LIST FOR LNM$PERMANENT_MAILBOX
     000A  0239                .WORD   LNM_SYSTEM_LENGTH       ; LNM$SYSTEM STRING ITEM
     0002  023B                .WORD   LNM$_STRING
 0000017D' 023D                .ADDRESS LNM_SYSTEM
 00000000  0241                .LONG   0
           0245
 00000000  0245                .LONG   0                       ; END OF ITEM LIST
           0249
           0249        LOG_G_LIST:                             ; ITEM LIST FOR LOG$GROUP
     0009  0249                .WORD   LNM_GROUP_LENGTH        ; LNM$GROUP STRING ITEM
     0002  024B                .WORD   LNM$_STRING
 0000014D' 024D                .ADDRESS LNM_GROUP
 00000000  0251                .LONG   0
           0255
 00000000  0255                .LONG   0                       ; END OF ITEM LIST
           0259
           0259        LOG_P_LIST:                             ; ITEM LIST FOR LOG$PROCESS
     000B  0259                .WORD   LNM_PROCESS_LENGTH      ; LNM$PROCESS STRING ITEM
     0002  025B                .WORD   LNM$_STRING
 0000015D' 025D                .ADDRESS LNM_PROCESS
 00000000  0261                .LONG   0
           0265
     0007  0265                .WORD   LNM_JOB_LENGTH          ; LNM$JOB STRING ITEM
     0002  0267                .WORD   LNM$_STRING
 00000156' 0269                .ADDRESS LNM_JOB
 00000000  026D                .LONG   0
           0271
 00000000  0271                .LONG   0                       ; END OF ITEM LIST
           0275
           0275        LOG_S_LIST:                             ; ITEM LIST FOR LOG$SYSTEM
     000A  0275                .WORD   LNM_SYSTEM_LENGTH       ; LNM$SYSTEM STRING ITEM
     0002  0277                .WORD   LNM$_STRING
 0000017D' 0279                .ADDRESS LNM_SYSTEM
 00000000  027D                .LONG   0
           0281
 00000000  0281                .LONG   0                       ; END OF ITEM LIST
           0285
           0285        SYSTEM_LIST:                            ; ITEM LIST FOR LNM$SYSTEM
     0004  0285                .WORD   4                       ; TERMINAL ATTRIBUTES ITEM
     0003  0287                .WORD   LNM$_ATTRIBUTES
```

```
000001D9'  0289              .ADDRESS TERMINAL_BUFFER
00000000   028D              .LONG   0
           0291
    0010   0291              .WORD   LNM_SYSTEM_TABLE_LENGTH ; LNM$SYSTEM_TABLE STRING ITEM
    0002   0293              .WORD   LNM$_STRING
0000019B'  0295              .ADDRESS LNM_SYSTEM_TABLE
00000000   0299              .LONG   0
           029D
00000000   029D              .LONG   0                       ; END OF ITEM LIST
           02A1
           02A1      TEMPORARY_MAILBOX_LIST:                 ; ITEM LIST FOR LNM$TEMPORARY_MAILBOX
    0007   02A1              .WORD   LNM_JOB_LENGTH          ; LNM$JOB STRING ITEM
    0002   02A3              .WORD   LNM$_STRING
00000156'  02A5              .ADDRESS LNM_JOB
00000000   02A9              .LONG   0
           02AD
00000000   02AD              .LONG   0                       ; END OF ITEM LIST
           02B1
           02B1      TRNLOG_GS_LIST:                         ; ITEM LIST FOR TRNLOG$_GROUP_SYSTEM
    0009   02B1              .WORD   LOG_GROUP_LENGTH        ; LOG$GROUP STRING ITEM
    0002   02B3              .WORD   LNM$_STRING
000001AB'  02B5              .ADDRESS LOG_GROUP
00000000   02B9              .LONG   0
           02BD
    000A   02BD              .WORD   LOG_SYSTEM_LENGTH       ; LOG$SYSTEM STRING ITEM
    0002   02BF              .WORD   LNM$_STRING
000001BF'  02C1              .ADDRESS LOG_SYSTEM
00000000   02C5              .LONG   0
           02C9
00000000   02C9              .LONG   0                       ; END OF ITEM LIST
           02CD
           02CD      TRNLOG_PG_LIST:                         ; ITEM LIST FOR TRNLOG$_PROCESS_GROUP
    000B   02CD              .WORD   LOG_PROCESS_LENGTH      ; LOG$PROCESS STRING ITEM
    0002   02CF              .WORD   LNM$_STRING
000001B4'  02D1              .ADDRESS LOG_PROCESS
00000000   02D5              .LONG   0
           02D9
    0009   02D9              .WORD   LOG_GROUP_LENGTH        ; LOG$GROUP STRING ITEM
    0002   02DB              .WORD   LNM$_STRING
000001AB'  02DD              .ADDRESS LOG_GROUP
00000000   02E1              .LONG   0
           02E5
00000000   02E5              .LONG   0                       ; END OF ITEM LIST
           02E9
           02E9      TRNLOG_PS_LIST:                         ; ITEM LIST FOR TRNLOG$_PROCESS_SYSTEM
    000B   02E9              .WORD   LOG_PROCESS_LENGTH      ; LOG$PROCESS STRING ITEM
    0002   02EB              .WORD   LNM$_STRING
000001B4'  02ED              .ADDRESS LOG_PROCESS
00000000   02F1              .LONG   0
           02F5
    000A   02F5              .WORD   LOG_SYSTEM_LENGTH       ; LOG$SYSTEM STRING ITEM
    0002   02F7              .WORD   LNM$_STRING
000001BF'  02F9              .ADDRESS LOG_SYSTEM
00000000   02FD              .LONG   0
           0301
00000000   0301              .LONG   0                       ; END OF ITEM LIST
           0305
```

```
          0305  TRNLOG_PGS_LIST:                            ; ITEM LIST FOR TRNLOG$_PROCESS_GROUP_SYSTEM
    000B  0305          .WORD   LOG_PROCESS_LENGTH          ; LOG$PROCESS STRING ITEM
    0002  0307          .WORD   LNM$_STRING
000001B4' 0309          .ADDRESS LOG_PROCESS
00000000  030D          .LONG   0
          0311
    0009  0311          .WORD   LOG_GROUP_LENGTH            ; LOG$GROUP STRING ITEM
    0002  0313          .WORD   LNM$_STRING
000001AB' 0315          .ADDRESS LOG_GROUP
00000000  0319          .LONG   0
          031D
    000A  031D          .WORD   LOG_SYSTEM_LENGTH           ; LOG$SYSTEM STRING ITEM
    0002  031F          .WORD   LNM$_STRING
000001BF' 0321          .ADDRESS LOG_SYSTEM
00000000  0325          .LONG   0
          0329
00000000  0329          .LONG   0                           ; END OF ITEM LIST
          032D
          032D  ;
          032D  ; ARGUMENT LISTS FOR THE $CRELNMS. THIS SYSTEM SERVICES CAN NOT BE DIRECTLY
          032D  ; ISSUED AT SYSTEM INITIALIZATION BECAUSE THE SWAPPER DOES NOT HAVE A P1 SPACE
          032D  ; WITH SYSTEM SERVICE VECTORS; HOWEVER, IT MAYBE CALLED DIRECTLY. SETUP AN
          032D  ; ARGUMENT LIST FOR EACH AND EVERY DIRECT CALL.
          032D  ;
          032D
          032D  DIRECTORIES_ARG:                            ; ARGUMENT LIST FOR LNM$DIRECTORIES
          032D          $CRELNM -
          032D                  ACMODE  = KERNEL_MODE, -
          032D                  ATTR    = LNM_NO_ALIAS, -
          032D                  ITMLST  = DIRECTORIES_LIST, -
          032D                  LOGNAM  = LNM_DIRECTORIES_DESC, -
          032D                  TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
          0345
          0345  FILE_DEV_EXEC_ARG:                          ; ARGUMENT LIST FOR EXECUTIVE LNM$FILE_DEV
          0345          $CRELNM -
          0345                  ACMODE  = EXEC_MODE, -
          0345                  ITMLST  = FILE_DEV_EXEC_LIST, -
          0345                  LOGNAM  = LNM_FILE_DEV_DESC, -
          0345                  TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
          035D
          035D  FILE_DEV_SUPER_ARG:                         ; ARGUMENT LIST FOR SUPERVISOR LNM$FILE_DEV
          035D          $CRELNM -
          035D                  ACMODE  = SUPER_MODE, -
          035D                  ITMLST  = FILE_DEV_SUPER_LIST, -
          035D                  LOGNAM  = LNM_FILE_DEV_DESC, -
          035D                  TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
          0375
          0375  LOG_G_ARG:                                  ; ARGUMENT LIST FOR LOG$GROUP
          0375          $CRELNM -
          0375                  ACMODE  = KERNEL_MODE, -
          0375                  ITMLST  = LOG_G_LIST, -
          0375                  LOGNAM  = LOG_G_DESC, -
          0375                  TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
          038D
          038D  LOG_P_ARG:                                  ; ARGUMENT LIST FOR LOG$PROCESS
          038D          $CRELNM -
          038D                  ACMODE  = KERNEL_MODE, -
```

```
038D                                ITMLST  = LOG_P_LIST, -
038D                                LOGNAM  = LOG_P_DESC, -
038D                                TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
03A5
03A5    LOG_S_ARG:                                          ; ARGUMENT LIST FOR LOG$SYSTEM
03A5                    $CRELNM -
03A5                                ACMODE  = KERNEL_MODE, -
03A5                                ITMLST  = LOG_S_LIST, -
03A5                                LOGNAM  = LOG_S_DESC, -
03A5                                TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
03BD
03BD    PERMANENT_MAILBOX_ARG:                              ; ARGUMENT LIST FOR LNM$PERMANENT_MAILBOX
03BD                    $CRELNM -
03BD                                ACMODE  = KERNEL_MODE, -
03BD                                ITMLST  = PERMANENT_MAILBOX_LIST, -
03BD                                LOGNAM  = LNM_PERMANENT_MAILBOX_DESC, -
03BD                                TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
03D5
03D5    SYSTEM_ARG:                                         ; ARGUMENT LIST FOR LNM$SYSTEM
03D5                    $CRELNM -
03D5                                ACMODE  = KERNEL_MODE, -
03D5                                ATTR    = LNM_NO_ALIAS, -
03D5                                ITMLST  = SYSTEM_LIST, -
03D5                                LOGNAM  = LNM_SYSTEM_DESC, -
03D5                                TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
03ED
03ED    TEMPORARY_MAILBOX_ARG:                              ; ARGUMENT LIST FOR LNM$TEMPORARY_MAILBOX
03ED                    $CRELNM -
03ED                                ACMODE  = KERNEL_MODE, -
03ED                                ITMLST  = TEMPORARY_MAILBOX_LIST, -
03ED                                LOGNAM  = LNM_TEMPORARY_MAILBOX_DESC, -
03ED                                TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
0405
0405    TRNLOG_GS_ARG:                                      ; ARGUMENT LIST FOR TRNLOG$_GROUP_SYSTEM
0405                    $CRELNM -
0405                                ACMODE  = KERNEL_MODE, -
0405                                ITMLST  = TRNLOG_GS_LIST, -
0405                                LOGNAM  = TRNLOG_GS_DESC, -
0405                                TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
041D
041D    TRNLOG_PG_ARG:                                      ; ARGUMENT LIST FOR TRNLOG$_PROCESS_GROUP
041D                    $CRELNM -
041D                                ACMODE  = KERNEL_MODE, -
041D                                ITMLST  = TRNLOG_PG_LIST, -
041D                                LOGNAM  = TRNLOG_PG_DESC, -
041D                                TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
0435
0435    TRNLOG_PS_ARG:                                      ; ARGUMENT LIST FOR TRNLOG$_PROCESS_SYSTEM
0435                    $CRELNM -
0435                                ACMODE  = KERNEL_MODE, -
0435                                ITMLST  = TRNLOG_PS_LIST, -
0435                                LOGNAM  = TRNLOG_PS_DESC, -
0435                                TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
044D
044D    TRNLOG_PGS_ARG:                                     ; ARGUMENT LIST FOR TRNLOG$_PROCESS_GROUP_SY
044D                    $CRELNM -
044D                                ACMODE  = KERNEL_MODE, -
```

```
    044D                        ITMLST  = TRNLOG_PGS_LIST, -
    044D                        LOGNAM  = TRNLOG_PGS_DESC, -
    044D                        TABNAM  = LNM_SYSTEM_DIRECTORY_DESC
    0465
00000180           .PSECT  $$$260                          ; WRITABLE PSECT
    0180                                                   ; ITMLST MUST BE FOLLOWING TWO CRELNM
    0180
    0130    SYS_DISK_ARG:                                  ; ARGUMENT LIST FOR SYS$DISK
    0180            $CRELNM -
    0180                        ACMODE  = EXEC_MODE, -
    0180                        LOGNAM  = SYS_DISK_DESC, -
    0180                        TABNAM  = LNM_SYSTEM_DESC
    0198
    0198    SYS_SYSDEVICE_ARG:                             ; ARGUMENT LIST FOR SYS$SYSDEVICE
    0198            $CRELNM -
    0198                        ACMODE  = EXEC_MODE, -
    0198                        LOGNAM  = SYS_SYSDEVICE_DESC, -
    0198                        TABNAM  = LNM_SYSTEM_DESC
```

```
              01B0
          00000465             .PSECT  YF$LOWUSE                  ; PAGED PSECT AT END OF SYS.EXE
              0465
              0465     ;
              0465     ; DEFINE A QUOTA LIST TO BE USED BY VARIOUS PIECES OF THE SYSTEM WHEN
              0465     ; CREATING A SPECIAL SYSTEM PROCESS, LIKE A FILES-11 ACP.  EVERY QUOTA
              0465     ; IS MENTIONED EXPLICITLY.  NOTE THAT THIS LIST CAN BE TAILORED BY
              0465     ; COPYING IT TO SOME TEMPORARY LOCATION AND APPENDING NEW QUOTA ITEMS
              0465     ; TO THE END OF THE LIST.  THE $CREPRC SYSTEM SERVICE USES THE LAST
              0465     ; VALUE OF A SPECIFIED QUOTA IN THE LIST WHEN IT CREATES A PROCESS.
              0465     ; NOTE THAT THE END OF THE LIST MUST BE TERMINATED BY A ZERO BYTE,
              0465     ; AND THAT THE LENGTH OF THE LIST, AS GIVEN BY PQL$C_SYSPQLLEN, DOES NOT
              0465     ; INCLUDE THE LIST TERMINATOR.
              0465     ;
              0465
              0465     PQL$AB_SYSPQL::                            ; SYSTEM PROCESS QUOTA LIST
        01    0465             .BYTE   PQL$_ASTLM                 ; PROCESS AST LIMIT
  0000000A    0466             .LONG   10
        02    046A             .BYTE   PQL$_BIOLM                 ; PROCESS BUFFERED I/O LIMIT
  0000000A    046B             .LONG   10
        03    046F             .BYTE   PQL$_BYTLM                 ; PROCESS BUFFERED I/O BYTE LIMIT
  00008000    0470             .LONG   32768
        04    0474             .BYTE   PQL$_CPULM                 ; PROCESS CPU TIME LIMIT
  00000000    0475             .LONG   0                          ; ZERO IMPLIES NO LIMIT
        05    0479             .BYTE   PQL$_DIOLM                 ; PROCESS DIRECT I/O LIMIT
  0000000A    047A             .LONG   10
        06    047E             .BYTE   PQL$_FILLM                 ; PROCESS OPEN FILE LIMIT
  0000003C    047F             .LONG   60
        07    0483             .BYTE   PQL$_PGFLQUOTA             ; PROCESS PAGE FILE QUOTA
  00004E20    0484             .LONG   20000
        08    0488             .BYTE   PQL$_PRCLM                 ; PROCESS SUBPROCESS CREATION LIMIT
  00000008    0489             .LONG   8
        09    048D             .BYTE   PQL$_TQELM                 ; PROCESS TIMER QUEUE ENTRY LIMIT
  00000008    048E             .LONG   8
        0B    0492             .BYTE   PQL$_WSDEFAULT             ; PROCESS DEFAULT WORKING SET SIZE
  00000064    0493             .LONG   100
        0A    0497             .BYTE   PQL$_WSQUOTA               ; PROCESS WORKING SET QUOTA
  000000C8    0498             .LONG   200
        0D    049C             .BYTE   PQL$_WSEXTENT              ; PROCESS WORKING SET EXTENT LIMIT
  000003E8    049D             .LONG   1000
        0C    04A1             .BYTE   PQL$_ENQLM                 ; PROCESS LOCK LIMIT
  00000064    04A2             .LONG   100
        0E    04A6             .BYTE   PQL$_JTQUOTA               ; JOB-WIDE LOGICAL NAME TABLE QUOTA
  00000400    04A7             .LONG   1024
        00    04AB     10$:    .BYTE   PQL$_LISTEND               ; END OF PROCESS QUOTA LIST
              04AC
  00000046    04AC     PQL$C_SYSPQLLEN == 10$ - PQL$AB_SYSPQL     ; LENGTH OF LIST (MINUS TERMINATOR)
```

```
                                          04AC              .SBTTL  EXE$SWAPINIT - INITIALIZATION AND STARTUP FOR SWAPPER
                                          04AC
                                          04AC      ;++
                                          04AC      ; FUNCTIONAL DESCRIPTION:
                                          04AC      ;       EXE$SWAPINIT IS ENTERED WHEN THE SWAPPER PROCESS IS FIRST
                                          04AC      ;       SCHEDULED AFTER A SYSTEM BOOT/STARTUP.  THIS TRANSFER OCCURS
                                          04AC      ;       VIA THE INITIAL PC VALUE BUILT INTO THE HARDWARE PCB FOR THE
                                          04AC      ;       SWAPPER PROCESS.  R4 CONTAINS THE ADDRESS OF THE SWAPPER PCB.
                                          04AC      ;
                                          04AC      ;--
                                          04AC
                                          04AC      EXE$SWAPINIT::                          ; SWAPPER INITIALIZATION
                                          04AC      ;
                                          04AC      ; INITIALIZE PAGED POOL.
                                          04AC      ;
             5B   00000000'GF         D0  04AC              MOVL    G^EXE$GL_PAGED,R11      ; POINT TO START OF PAGED POOL
                           8B         D4  04B3              CLRL    (R11)+                  ; ZAP FORWARD LINK
             6B   00000000'GF         D0  04B5              MOVL    G^SGN$GL_PAGEDYN,(R11)  ; AND SET SIZE
                                          04BC
                                          04BC      ;
                                          04BC      ; ALLOCATE LOGICAL NAME HASH TABLE.  THE NUMBER OF ENTRIES IN THE HASH TABLE
                                          04BC      ; MUST BE A POWER OF TWO.  SO THE ALLOCATED SIZE IS THE SMALLEST POWER OF
                                          04BC      ; TWO LARGER THAN THE SYSGEN PARAMETER.
                                          04BC      ;
                  00000000'FF         DD  04BC              PUSHL   @LNM$AL_HASHTBL         ; SAVE ADDR OF CRELNM ITMLST BLOCKS FOR
                                          04C2                                              ; "SYS$DISK" AND "SYS$SYSDEVICE"
                           58   01    D0  04C2              MOVL    #1, R8                  ; DO THIS TWICE
          57    00000000'GF48   01    C3  04C5      40$:    SUBL3   #1,G^LNM$GL_HTBLSIZS[R8],R7 ; PICK UP ONE LESS THAN SYSGEN PARM
                           57   57    4E  04CE              CVTLF   R7,R7                   ; CONVERT TO FLOATING
                 57   57   07   07    EF  04D1              EXTZV   #7,#7,R7,R7             ; PICK UP EXPONENT-NOW THE POWER OF 2
                                51    D4  04D6              CLRL    R1                      ; CLEAR A REGISTER
                        00 51   57    E2  04D8              BBSS    R7,R1,50$               ; THE SIZE OF THE TABLE ROUNDED UP
                0000000'GF48    51    D0  04DC      50$:    MOVL    R1,G^LNM$GL_HTBLSIZS[R8]; WRITE BACK THE CORRECT VALUE
                           DE   58    F4  04E4              SOBGEQ  R8,40$                  ; LOOP TWO TIMES
                                          04E7      ;
                                          04E7      ; INITIALIZE THE SYSTEM SPACE HASH TABLE.
                                          04E7      ;
             51   00000000'GF         D0  04E7              MOVL    G^LNM$GL_HTBLSIZS,R1    ; SIZE OF TABLE IN ENTRIES
                51   0000000C 9F41    DE  04EE              MOVAL   @#LNMHSH$K_BUCKET[R1],R1; MULT BY 4 AND ADD OVERHEAD
                  00000000'GF         16  04F6              JSB     G^EXE$ALOPAGED          ; ALLOCATE MEMORY
                                06    BB  04FC              PUSHR   #^M<R1,R2>              ; SAVE REGISTERS DESTORYED BY MOVC5
         62   51   00   00 8F   00    2C  04FE              MOVC5   #0,#0,#0,R1,(R2)        ; ZERO HASH TABLE
                                12    BA  0505              POPR    #^M<R1,R4>              ; RESTORE REGISTERS DESTORYED BY MOVC5
                                          0507                                              ;  NOTE: tHAT R2 COMES BACK AS R4
             50   00000000'GF   01    C3  0507              SUBL3   #1,G^LNM$GL_HTBLSIZS,R0 ; CALC UPPER BOUND OF HASH INDEX
                           64   50    D2  050F              MCOML   R0,LNMHSH$L_MASK(R4)    ; STORE HASH INDEX MASK
                        08 A4   51    B0  0512              MOVW    R1,LNMHSH$W_SIZE(R4)    ; STORE SIZE IN STRUCTURE HEADER
                        0A A4   38    90  0516              MOVB    #DYN$C_RSHT,LNMHSH$B_TYPE(R4)
                                          051A                                              ; STORE STRUCTURE TYPE
                  00000000'FF   64    9E  051A              MOVAB   (R4),@LNM$AL_HASHTBL    ; STUFF WAY POINTER TO TABLE
                                          0521                                              ; NOTE: THAT THE HASH TABLE HAS BEEN
                                          0521                                              ; INITIALIZED TO ZERO
                                          0521
                                          0521      ;
                                          0521      ; FIX UP THE SYSTEM LOGICAL NAME DIRECTORY, AND INSERT IT IN INTO THE
                                          0521      ; APPROPRIATE HASH BUCKET OF THE SHAREABLE LOGICAL NAME HASH TABLE.
                                          0521      ;
```

```
                                 0521
         53  00000000'EF    9E   0521          MOVAB   LNM$SYSTEM_DIRECTORY,R3
       0000002C'EF    64    DE   0528          MOVAL   (R4),LNM_SYSTEM_DIR_LNMTH+LNMTH$L_HASH
                                 052F                                                  ; HASH TABLE ADDRESS IN LNMTH
                50   11 A3  9A   052F          MOVZBL  LNMB$T_NAME(R3),R0              ; GET SIZE OF DIRECTORY NAME
                51   12 A3  9E   0533          MOVAB   LNMB$T_NAME+1(R3),R1            ; GET ADDRESS OF DIRECTORY NAME
             00000000'EF    16   0537          JSB     LNM$HASH                        ; HASH THE DIRECTORY NAME
                     50 64  CA   053D          BICL2   LNMHSH$L_MASK(R4),R0            ; MODIFY THE INDEX TO BE IN RANGE
            0C A440    63   DE   0540          MOVAL   (R3),LNMHSH$C_BUCKET(R4)[R0]
                                 0545                                                  ; INSERT DIRECTORY LNMB IN HASH TABLE
        04 A3    0C A440    DE   0545          MOVAL   LNMHSH$C_BUCKET(R4)[R0],LNMB$L_BLINK(R3)
                                 054B                                                  ; INSERT HASH TBL ENTRY IN DIRECT LNMB
                                 054B
                                 054B   ;
                                 054B   ; FIXUP THE SYSTEM LOGICAL NAME TABLE, LNM$SYSTEM_TABLE, AND INSERT IT INTO THE
                                 054B   ; APPROPRIATE HASH BUCKET OF THE SYSTEM LOGICAL NAME HASH TABLE.
                                 054B   ;
                                 054B
         51   000000C0'EF   9E   054B          MOVAB   SYSTEM_TABLE,R1                 ; RETRIEVE SYSTEM TABLE LNMB ADDRESS
              00000000'FF   D0   0552          MOVL    @LNM$AL_HASHTBL,-               ; MOVE THE ADDRESS OF THE SHAREABLE
              000000E8'EF        0558                  SYSTEM_TABLE_LNMTH+-            ; LOGICAL NAME HASH TABLE INTO THE
                                 055D                  LNMTH$L_HASH                    ; SYSTEM TABLE'S TABLE HEADER
                       52   D4   055D          CLRL    R2                              ; NO SPECIAL INSERTION ATTRIBUTES
              00000000'GF   16   055F          JSB     G^LNM$INSLOGTAB                 ; APPROPRIATELY INSERT LNM$SYSTEM_TABLE
                                 0565
                                 0565   ;
                                 0565   ; CREATE THE SYSTEM LOGICAL NAMES, CONTAINED WITHIN THE SYSTEM DIRECTORY TABLE,
                                 0565   ; - ALL OF WHICH MUST BE CREATED AT SYSTEM INITIALIZATION TIME.
                                 0565   ;
                                 0565
                                 0565          CALLG -                                 ; CREATE LNM$DIRECTORIES
  80000000'9F      FDC4 CF  FA   0565                  DIRECTORIES_ARG, -
                                 056E                  @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                 056E
                                 056E          CALLG -                                 ; CREATE EXECUTIVE LNM$FILE_DEV
  80000000'9F      FDD3 CF  FA   056E                  FILE_DEV_EXEC_ARG, -
                                 0577                  @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                 0577
                                 0577          CALLG -                                 ; CREATE SUPERVSIOR LNM$FILE_DEV
  80000000'9F      FDE2 CF  FA   0577                  FILE_DEV_SUPER_ARG, -
                                 0580                  @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                 0580
                                 0580          CALLG -                                 ; CREATE LOG$GROUP
  80000000'9F      FDF1 CF  FA   0580                  LOG_G_ARG, -
                                 0589                  @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                 0589
                                 0589          CALLG -                                 ; CREATE LOG$PROCESS
  80000000'9F      FE00 CF  FA   0589                  LOG_P_ARG, -
                                 0592                  @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                 0592
                                 0592          CALLG -                                 ; CREATE LOG$SYSTEM
  80000000'9F      FE0F CF  FA   0592                  LOG_S_ARG, -
                                 059B                  @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                 059B
                                 059B          CALLG -                                 ; CREATE LNM$PERMANENT_MAILBOX
  80000000'9F      FE1E CF  FA   059B                  PERMANENT_MAILBOX_ARG, -
                                 05A4                  @#SYS$CRELNM-P1SYSVECTORS+^X80000000
```

```
                                  05A4
                                  05A4        CALLG -                         ; CREATE LNM$SYSTEM
    80000000'9F    FE2D CF    FA  05A4                SYSTEM_ARG, -
                                  05AD                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                  05AD
                                  05AD        CALLG -                         ; CREATE LNM$TEMPORARY_MAILBOX
    80000000'9F    FE3C CF    FA  05AD                TEMPORARY_MAILBOX_ARG, -
                                  05B6                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                  05B6
                                  05B6        CALLG -                         ; CREATE TRNLOG$_GROUP_SYSTEM
    80000000'9F    FE4B CF    FA  05B6                TRNLOG_GS_ARG, -
                                  05BF                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                  05BF
                                  05BF        CALLG -                         ; CREATE TRNLOG$_PROCESS_GROUP
    80000000'9F    FE5A CF    FA  05BF                TRNLOG_PG_ARG, -
                                  05C8                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                  05C8
                                  05C8        CALLG -                         ; CREATE TRNLOG$_PROCESS_SYSTEM
    80000000'9F    FE69 CF    FA  05C8                TRNLOG_PS_ARG, -
                                  05D1                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                  05D1
                                  05D1        CALLG -                         ; CREATE TRNLOG$_PROCESS_GROUP_SYSTEM
    80000000'9F    FE78 CF    FA  05D1                TRNLOG_PGS_ARG, -
                                  05DA                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                  05DA
                                  05DA  ;
                                  05DA  ; CREATE TWO STARTUP LOGICAL NAMES.
                                  05DA  ;
                                  05DA
                          54 8ED0 05DA        POPL    R4                      ; RECOVER ADDR OF CRELOG BLOCK
          00000194'EF     64   DE 05DD        MOVAL   (R4),SYS_DISK_ARG+CRELNM$_ITMLST
                                  05E4                                        ; STUFF THE ADDRESS OF THE ITEM LIST
      000001AC'EF     0000'C4  DE 05E4        MOVAL   BDL$L_SYSDLOG(R4),SYS_SYSDEVICE_ARG+CRELNM$_ITMLST
                                  05ED                                        ; STUFF THE ADDRESS OF THE ITEM LIST
                                  05ED
                                  05ED        CALLG -                         ; CREATE SYS$DISK
  80000000'9F    00000180'EF  FA  05ED                SYS_DISK_ARG, -
                                  05F8                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                  05F8
                                  05F8        CALLG -                         ; CREATE SYS$SYSDEVICE
  80000000'9F    00000198'EF  FA  05F8                SYS_SYSDEVICE_ARG, -
                                  0603                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                  0603
                  51    00'8F  9A 0603        MOVZBL  #BDL$S_CRELNM_ITMLST,R1 ; GET THE SIZE OF THE STRUCTURE
                     50    54  D0 0607        MOVL    R4,R0                   ; MOVE STRUCTURE ADDR INTO CORRECT REG
                 00000000'EF   16 060A        JSB     EXE$DEANONPGDSIZ        ; RETURN THE MEMORY
                                  0610
                                  0610  ;
                                  0610  ; CREATE INITIAL PROCESSES
                                  0610  ;
                                  0610
                                  0610  ; THE $CREPRC_S MACRO CANNOT BE USED BECAUSE THAT MACRO GENERATES A
                                  0610  ; CALL THROUGH THE P1 SYSTEM SERVICE VECTOR PAGES AND THE SWAPPER DOES
                                  0610  ; NOT HAVE A P1 SPACE. tHE SENSE OF THE CREATE PROCESS CALL IS THE
                                  0610  ; FOLLOWING.
                                  0610  ;
                                  0610  ;     $CREPRC_S       INPUT=TTODESC,- ;
                     0610  ;                               OUTPUT=TTODESC,-;
                     0610  ;                               ERROR=TTODESC,-  ;
                     0610  ;                               IMAGE=IMGDESC,-  ;
                     0610  ;                               UIC=#^X80020,-   ;
                     0610  ;                               STSFLG=#<PRC$M_NOACNT!PRC$M_SSRWAIT>,-
                     0610  ;                               BASPRI=#2        ;
                     0610
             09  DD  0610          PUSHL   #<PRC$M_NOACNT!PRC$M_SSRWAIT>
             7E  D4  0612          CLRL    -(SP)
    00080020 8F  DD  0614          PUSHL   #^X80020
             02  DD  061A          PUSHL   #2
             7E  7C  061C          CLRQ    -(SP)
             00  DD  061E          PUSHL   #0
        F9EF CF  7F  0620          PUSHAQ  TTODESC
             6E  DD  0624          PUSHL   (SP)
             6E  DD  0626          PUSHL   (SP)
        F9D4 CF  7F  0628          PUSHAQ  IMGDESC
             00  DD  062C          PUSHL   #0
80000000'9F  0C  FB  062E          CALLS   #12,@#SYS$CREPRC-P1SYSVECTORS+^X80000000
 00000000'GF     17  0635          JMP     G^LOOP                  ; JUMP OFF TO THE MAIN LOOP
```

```
                      063B          .SBTTL  SWAPPER - MAIN LOOP
                      063B
                      063B     ;++
                      063B     ; FUNCTIONAL DESCRIPTION:
                      063B     ;        THE MAIN LOOP OF THE SWAPPER IS EXECUTED WHENEVER THE SWAPPER IS AWAKENED
                      063B     ;        FOR ANY REASON.  EACH OF THE FUNCTIONAL ROUTINES WILL CHECK TO SEE IF
                      063B     ;        THEY HAVE ANY ACTION TO PERFORM.
                      063B     ;--
                      063B
                  00000000          .PSECT  $AEXENONPAGED                 ; NON-PAGED PSECT
          2E   10     0000     LOOP:   BSBB    BALANCE                       ; BALANCE FREE PAGE COUNT
       FFFB'   30     0002          BSBW    MMG$WRTMFYPAG                 ; WRITE MODIFIED PAGES
        0088   30     0005          BSBW    SWAPSCHED                     ; SCHEDULE SWAP
     0000'CF   D5     0008          TSTL    W^EXE$GL_PFATIM               ; CHECK FOR POWER FAIL TIME
          06   13     000C          BEQL    15$                           ; BRANCH IF NO POWERFAIL
 00000000'EF   16     000E          JSB     EXE$POWERAST                  ; GIVE ANY REQUIRED POWER FAIL ASTS
 54  0000'CF   D0     0014     15$:    MOVL    W^SCH$GL_CURPCB,R4            ; GET PROPER PCB ADDRESS
 52  0000'CF   7E     0019          MOVAQ   W^SCH$GQ_HIBWQ,R2             ; AND ADDRESS OF WAIT QUEUE HEADER
                      001E          SETIPL  #IPL$_SYNCH                   ; BLOCK SYSTEM EVENTS WHILE CHECKING
05 24 A4   0C   E4    0021          BBSC    #PCB$V_WAKEPEN,PCB$L_STS(R4),20$ ; TEST AND CLEAR WAKE PENDING
          00   DD     0026          PUSHL   #0                            ; NULL PSL
       FFD5'   30     0028          BSBW    SCH$WAITK                     ; WAIT WITH STACK CLEAN
                      002B     20$:    SETIPL  #0                            ; DROP IPL
          D0   11     002E          BRB     LOOP                          ; CHECK FOR WORK TO DO
                      0030          .DISABLE LSB
```

```
                           0030             .SBTTL  BALANCE FREE PAGE COUNT
                           0030
                           0030   ;++
                           0030   ; FUNCTIONAL DESCRIPTION:
                           0030   ;         BALANCE WILL ENSURE THAT THE FREE PAGE LIST HAS AT LEAST THE NUMBER OF
                           0030   ;         PAGES SPECIFIED BY THE PARAMETER FREELIM.  IF NOT, PAGES WILL BE MADE
                           0030   ;         AVAILABLE BY EITHER WRITING MODIFIED PAGES OR OUTSWAPPING PROCESSES.
                           0030   ;         IF SUFFICIENT FREE PAGES ARE AVAILABLE, THEN A CHECK IS MADE FOR
                           0030   ;         DELETED PROCESS HEADERS IN NEED OF CLEANUP.
                           0030   ;--
                           0030
                           0030   BALANCE:                                              ; BALANCE FREE PAGE COUNT
   0000'CF    0000'CF  D1  0030           CMPL    W^SGN$GL_FREELIM,W^SCH$GL_FREECNT ; ARE WE HERE DUE TO FREELIM?
                   0A  15  0037           BLEQ    5$                                ; BRANCH IF NOT
53 0000'CF    0000'CF  C3  0039           SUBL3   W^SGN$GL_FREEGOAL,W^SCH$GL_FREECNT,R3 ; SUFFICIENT FREE PAGES?
                   0B  19  0041           BLSS    20$                               ; NO, MUST ACQUIRE SOME
              0000'CF  B5  0043   5$:     TSTW    W^SCH$GW_DELPHDCT                 ; CHECK FOR DELETED PROCESS HEADERS
                   04  13  0047           BEQL    10$                               ; NONE, EXIT
                   53  D4  0049           CLRL    R3                                ; INDICATE NO FREE PAGES NEEDED
                   1E  11  004B           BRB     25$                               ;
                       05  004D   10$:    RSB                                       ; IN BALANCE, RETURN
                           004E   20$:                                              ;
  17 0000'CF      00'  E0  004E           BBS     S^#SCH$V_MPW,W^SCH$GB_SIP,25$     ; MODIFIED PAGE WRITING ACTIVE
50 0000'CF    0000'CF  C3  0054           SUBL3   W^SCH$GL_MFYLOLIM,W^SCH$GL_MFYCNT,R0 ; HOW MUCH WILL WRITING PAGES
                   0D  15  005C           BLEQ    25$                               ; NONE, MUST OUTSWAP
              50   53  C0  005E           ADDL    R3,R0                             ; YIELD RELATIVE TO WHAT WE NEED?
                   08  19  0061           BLSS    25$                               ; NOT ENOUGH, MUST OUTSWAP
   0000'CF    0000'CF  D0  0063           MOVL    W^SCH$GL_MFYLOLIM,W^SCH$GL_MFYLIM ; TRIGGER MODIFIED PAGE WRITING
                       05  006A           RSB                                       ; AND EXIT TO LET IT HAPPEN
                           006B   ;
                           006B   ;       MUST OUTSWAP, FIRST CHECK FOR SWAP IN PROGRESS SINCE SWAPPER IS
                           006B   ;       NOT RE-ENTRANT.  IF PURGING DELETED HEADERS, THEN THE NUMBER OF
                           006B   ;       REQUIRED PAGES (IN R3) WILL BE SET TO ZERO.  AN INFINITE INSWAP PRIORITY
                           006B   ;       WILL BE ASSUMED WHILE BALANCING THE NUMBER FO FREE PAGES.
                           006B   ;
    DC 0000'CF    00'  E2  006B   25$:    BBSS    S^#SCH$V_SIP,W^SCH$GB_SIP,10$     ; EXIT IF SWAPPER ALREADY BUSY
                           0071           SETIPL  #IPL$_SYNCH                       ; BLOCK SYSTEM EVENTS
              3FC0 8F  BB  0074           PUSHR   #^M<R6,R7,R8,R9,R10,R11,AP,FP>    ; SAVE NON-STANDARD REGISTERS
              0000'CF  94  0078           CLRB    W^SWP$GB_ISWPRI                   ; SET PRIORITY FOR SWAP SCHEDULE
              5D   53  D0  007C           MOVL    R3,FP                             ; GET AND TEST FREE PAGE DEFICIT
                   08  18  007F           BGEQ    30$                               ; NONE, PURGING DELETED HEADERS
              0018'CF  B5  0081           TSTW    W^SWP$GW_BALCNT                   ; CHECK FOR SINGULAR BALANCE SET
                   02  12  0085           BNEQ    30$                               ; NO, CAN OUTSWAP
                   5D  D4  0087           CLRL    FP                                ; PREVENT OUTSWAP SCHEDULE
                 008A  31  0089   30$:    BRW     OUTSWAP                           ; TRY TO FORCE AN OUTSWAP
```

```
                              008C          .SBTTL  SCHEDULE SWAP
                              008C
                              008C  ;++
                              008C  ; FUNCTIONAL DESCRIPTION:
                              008C  ;       SWAPSCHED IS CALLED BY THE MAIN LOOP OF THE SWAPPER PROCESS TO CHECK
                              008C  ;       ELIGIBLE INSWAP CANDIDATES AND TO PROVIDE MEMORY NEEDED FOR THEIR
                              008C  ;       INSWAP.  A QUICK EXIT WILL BE TAKEN IF THE SWAPPER IS ALREADY BUSY.
                              008C  ;       NO OUTSWAP WILL BE NEEDED IF THE NUMBER OF REQUIRED PAGES CAN BE
                              008C  ;       TAKEN FROM THE FREE PAGE LIST LEAVING AT LEAST FREELIM STILL FREE.
                              008C  ;       OTHERWISE OUTSWAP WILL BE ENTERED TO MAKE PAGES AVAILABLE BY ANY
                              008C  ;       MEANS NECESSARY.
                              008C  ;--
                              008C
                              008C  QEMPTY: BUG_CHECK QUEUEMPTY,FATAL         ; EMPTY QUEUE OR NOT A PCB
                              0090
                              0090  SWAPSCHED:                                ;
                              0090          SETIPL  #IPL$_SYNCH               ; BLOCK SYSTEM EVENTS
    0F 0000'CF      00'  E2   0093          BBSS    S^#SCH$V_SIP,W^SCH$GB_SIP,5$    ; EXIT IF SWAP IN PROGRESS
52    0000'CF  20   00   EA   0099          FFS     #0,#32,W^SCH$GL_COMOQS,R2; FIND HIGHEST PRIORITY QUEUE
                    0A   12   00A0          BNEQ    10$                       ; FOUND ONE
    00 0000'CF      00'  E5   00A2          BBCC    S^#SCH$V_SIP,W^SCH$GB_SIP,5$    ; CLEAR SWAP IN PROGRESS
                              00A8  5$:     SETIPL  #0                        ; DROP IPL
                         05   00AB          RSB                               ; AND RETURN
                              00AC
                              00AC  10$:                                      ;
             3FC0 8F     BB   00AC          PUSHR   #^M<R6,R7,R8,R9,R10,R11,AP,FP>  ; SAVE REGS OTHER THAN R0-R5
      53    0000'CF42    7E   00B0          MOVAQ   W^SCH$AQ_COMQH[R2],R3     ; COMPUTE ADDRESS OF QUEUE HEADER
             54   63     D0   00B6          MOVL    (R3),R4                   ; GET PCB ADDRESS
                              00B9  ;
                              00B9  ; THE FOLLOWING CHECK IS NEEDED DUE TO THE ODIOUS MISLEADING SYMPTOMS THAT
                              00B9  ; MIGHT OTHERWISE RESULT.
                              00B9  ;
           0A A4    0C   91   00B9          CMPB    #DYN$C_PCB,PCB$B_TYPE(R4); IS THIS A GOOD PCB?
                    CD   12   00BD          BNEQ    QEMPTY                    ;  BUGCHECK IF NOT
                              00BF  ;
                              00BF  ; DETERMINE THE SIZE OF THE INSWAP CANDIDATE, TAKING INTO ACCOUNT THE FACT
                              00BF  ; THAT THE PROCESS HEADER MIGHT ALREADY BE RESIDENT.
                              00BF  ;
           50    36 A4   3C   00BF          MOVZWL  PCB$W_PPGCNT(R4),R0       ; COUNT OF PROCESS PAGES
           5A    34 A4   3C   00C3          MOVZWL  PCB$W_GPGCNT(R4),R10      ; COUNT OF GLOBAL PAGES
              5A    50   C0   00C7          ADDL    R0,R10                    ; SUM PAGE COUNTS
        07 24 A4    12   E1   00CA          BBC     #PCB$V_PHDRES,PCB$L_STS(R4),15$ ; CONTINUE IF HEADER NON-RESIDENT
           50    30 A4   3C   00CF          MOVZWL  PCB$W_APTCNT(R4),R0       ; GET ACTIVE PAGE TABLE COUNT
              5A    50   C2   00D3          SUBL    R0,R10                    ; SUBTRACT RESIDENT HEADER PAGES FROM REQUIR
                              00D6  15$:                                      ;
50    0000'CF   0000'CF  C3   00D6          SUBL3   W^SCH$GL_FREELIM,W^SCH$GL_FREECNT,R0    ; COMPUTE PAGES AVAILABLE
   0000'CF       0B A4   90   00DE          MOVB    PCB$B_PRI(R4),W^SWP$GB_ISQPRI   ; SAVE PRIORITY OF INSWAP
         5D    50   5A   C3   00E4          SUBL3   R10,R0,FP                       ; WILL PROCESS FIT?
                    0A   19   00E8          BLSS    20$                       ; NO, MUST OUTSWAP
      001A'CF   0000'CF  B0   00EA          MOVW    W^SCH$GW_SWPFAIL,W^SCH$GW_SWPFCNT       ; RESET FAILURE COUNTER
                  0441   31   00F1          BRW     INSWAP                    ; YES PERFORM SWAP
                              00F4  ;
                              00F4  ; IF INSWAPPING A NON-REAL TIME PROCESS, THEN CHECK TO SEE IF ITS CURRENT
                              00F4  ; PRIORITY IS THE DEFAULT BACKGROUND PRIORITY.  IF SO, THEN DELAY AT LEAST
                              00F4  ; SWAPRATE INTERVAL SINCE THE LAST INSWAP.  THE EFFECT WILL BE TO AVOID FILLING
                              00F4  ; THE BALANCE SET WITH CRUNCHING PROCESSES IMMEDIATELY.
                              00F4  ;
```

```
                                 00F4  1209 20$:                                          ; SCHEDULE OUTSWAP
          0B A4    10    91      00F4  1210         CMPB    #16,PCB$B_PRI(R4)             ; IS THIS A REAL TIME PROCESS?
                   1C    14      00F8  1211         BGTR    40$                           ; BR IF SO
  51   1F    0000'CF     83      00FA  1212         SUBB3   W^SYS$GB_DEFPRI,#31,R1        ; CONVERT PRIORITY TO INTERNAL FORM
               51  52    91      0100  1213         CMPB    R2,R1                         ; IS THIS A CRUNCHER OR LOW PRIORITY JOB?
                   11    19      0103  1214         BLSS    40$                           ; BR IF NOT
  51      00000000'EF    D0      0105  1215         MOVL    EXE$GQ_SYSTIME,R1             ; GET CURRENT TIME IN APPROX. 10MS UNITS
         0000'CF   51    D1      010C  1216         CMPL    R1,W^SWP$GL_SWTIME            ; HAS INTERVAL ELAPSED?
                   03    1A      0111  1217         BGTRU   40$                           ; BR IF YES
                 08CB    31      0113  1218         BRW     SWAPEXIT                      ; CAN'T DO SWAP NOW
                                 0116  1219 40$:                                          ;
```

```
                            0116                 .SBTTL  OUTSWAP
                            0116
                            0116         ;------------------------------------------------------------------------
                            0116         ;
                            0116         ;       SCHEDULE AND PERFORM OUTSWAPS IF POSSIBLE
                            0116         ;
                            0116         ;------------------------------------------------------------------------
                            0116         ; FUNCTIONAL DESCRIPTION:
                            0116         ;       THE OUTSWAP STRATEGY IS TO FREE PROCESS HEADERS FOR OUTSWAP PROCESSES,
                            0116         ;       USE AVAILABLE MODIFIED PAGES (AFTER WRITING THEM) AND FINALLY AS A LAST
                            0116         ;       RESORT OUTSWAP ANOTHER PROCESS.  ONLY ONE OF THESE ACTIONS WILL BE TAKEN
                            0116         ;       AT A TIME THEN THE SCHEDULING SITUATION WILL BE RE-EVALUATED.  THE VALUE
                            0116         ;       IN FP INDICATES THE SIZE OF THE PAGE DEFICIT AND WILL BE SET POSITIVE IF
                            0116         ;       ENTERED FROM BALANCE TO FREE DELETED PROCESS HEADERS.
                            0116         ;
                            0116         ; INPUT:        FP - NEGATIVE VALUE WILL PERMIT PROCESS OUTSWAP
                            0116         ;                    ZERO OR POSITIVE WILL PURGE HEADERS ONLY.
                            0116         ;
                            0116
                            0116         OUTSWAP:                                        ; TRY TO OUTSWAP
58   00000000'EF   01   C3  0116                 SUBL3   #1,SGN$GL_BALSETCT,R8   ; INIT INDEX FOR BALANCE SLOT SCAN
              59   00   D2  011E                 MCOML   #0,R9                   ; INDICATE NO FREE LIST PURGE CANDIDATE
            0000'DF48   B5  0121         10$:    TSTW    @W^PHV$GL_REFCBAS[R8]   ; IS SLOT IN NEED OF CLEANUP?
                   03   12  0126                 BNEQ    12$                     ; CONTINUE IF NOT RELEASABLE
                 00D1   31  0128                 BRW     60$                     ; GO RELEASE PAGE TABLES AND HEADER
      54    0000'DF48   32  012B         12$:    CVTWL   @W^PHV$GL_PIXBAS[R8],R4 ; GET PROCESS INDEX
                   0A   15  0131                 BLEQ    15$                     ; DELETED PROCESS OR VACANT SLOT
      54    0000'DF44   D0  0133                 MOVL    @W^SCH$GL_PCBVEC[R4],R4 ; GET PCB ADDRESS FOR PIX
             0F 24 A4   E8  0139                 BLBS    PCB$L_STS(R4),20$       ; SKIP IF PROCESS IS RESIDENT
                   0D   13  013D         15$:    BEQL    20$                     ; VACANT SLOT
              0000'CF   95  013F                 TSTB    W^EXE$GQ_SYSTIME        ; ADD 1 IN 8 RANDOMNESS TO DECISION
                   04   13  0143                 BEQL    17$                     ; BRANCH ON LOW PROBABILITY
                   59   D5  0145                 TSTL    R9                      ; CHECK FOR REMEMBERED INDEX
                   03   18  0147                 BGEQ    20$                     ; YES DONT OVERWRITE
              59   58   D0  0149         17$:    MOVL    R8,R9                   ; SAVE BALANCE SLOT NUMBER OF CANDIDATE
                D2 58   F4  014C         20$:    SOBGEQ  R8,10$                  ; TRY ALL SLOTS
              0001'CF   95  014F                 TSTB    W^EXE$GQ_SYSTIME+1      ; ADD 1 IN 256 RANDOMNESS TO DECISION
                   05   13  0153                 BEQL    21$                     ; BRANCH ON VERY LOW PROBABILITY
              58   59   D0  0155                 MOVL    R9,R8                   ; GET AND TEST SLOT INDEX FOR SECONDARY CAND
                   11   18  0158                 BGEQ    24$                     ; BR IF SLOT FOR CLEANUP
                   5D   D5  015A         21$:    TSTL    FP                      ; CHECK FOR HEADER PURGE
                   07   18  015C                 BGEQ    22$                     ; EXIT IF SO
                            015E         ;
                            015E         ; SINCE THERE WAS NO HEADER TO FREE, WE MUST NOW WRITE MODIFIED PAGES OR OUTSWAP
                            015E         ; SOME PROCESS.  ONLY IF MODIFIED PAGES (MFYCNT-LOLIM) WILL TOTALLY SATISFY OUR
                            015E         ; NEEDS WILL THEY BE WRITTEN.  OTHERWISE THE LEAST USEFUL (BY SOME OPINION) PROCESS
                            015E         ; WILL BE OUTSWAPPED AND THE SITUATION RECONSIDERED.
                            015E         ;
                FE9F'   30  015E                 BSBW    SCH$OSWPSCHED           ; SCHEDULE OUTSWAP
                   54   D5  0161                 TSTL    R4                      ; ANY CANDIDATE?
                   03   12  0163                 BNEQ    23$                     ; YES
                 0879   31  0165         22$:    BRW     SWAPEXIT                ; ELSE EXIT AND TRY LATER
                 0094   31  0168         23$:    BRW     70$                     ; GO DO OUTSWAP
                            016B         ;
                            016B         ; A HEADER SLOT IN NEED OF CLEANUP WAS FOUND, NOW SCAN THE FREELIST FOR ALL
                            016B         ; PAGES WHOSE PTE BACK POINTERS PLACE THEM WITHIN THIS HEADER.  DELETE THE
                            016B         ; CONTENT OF THOSE PAGES VIA MMG$DELCONPFN TO FINALLY REDUCE THE REFERENCE
```

```
                                  016B  1279 ; COUNT BINDING THE HEADER.
                                  016B  1280 ;
  57   0000'CF     02        9C   016B  1281 24$:     ROTL     #2,W^SWP$GL_BSLOTSZ,R7    ; GET SIZE OF BALANCE SLOT IN BYTES
       56    58    57        C5   0171  1282          MULL3    R7,R8,R6                  ; COMPUTE OFFSET TO BASE OF SLOT
             0000'DF46       9F   0175  1283          PUSHAB   @W^SWP$GL_BALSPT[R6]      ; ADD BASE TO GET ADDRESS
       52    56    07        9C   017A  1284          ROTL     #7,R6,R2                  ; FORM OFFSET TO PHD BASE
             0000'DF42       9F   017E  1285          PUSHAB   @W^SWP$GL_BALBASE[R2]     ; BASE ADDRESS FOR PHD
       51    57    07        9C   0183  1286          ROTL     #7,R7,R1                  ; MUL SPT SLOT SIZE BY 128
                04 BE47      9F   0187  1287          PUSHAB   @4(SP)[R7]                ; FORM HIGH LIMIT FOR PAGTBLPPTE
                04 BE41      9F   018B  1288          PUSHAB   @4(SP)[R1]                ; ANS SAVE PTE HIGH LIMIT
                                  018F  1289
                                  018F  1290 ;
                                  018F  1291 ;        AT THIS POINT:
                                  018F  1292 ;
                                  018F  1293 ;        00(SP) - HIGH LIMIT ADDRESS FOR PROCESS HEADER
                                  018F  1294 ;        04(SP) - HIGH LIMIT FOR PROCESS PAGE TABLE PTE
                                  018F  1295 ;        08(SP) - LOW LIMIT FOR PROCESS HEADER
                                  018F  1296 ;        12(SP) - LOW LIMIT  FOR PROCESS PAGE TABLE PTE
                                  018F  1297 ;
                                  018F  1298          ASSUME   PFN$C_FREPAGLST EQ 0      ;
                                  018F  1299          ASSUME   PFN$C_MFYPAGLST EQ 1      ;
                                  018F  1300          ASSUME   PFN$C_BADPAGLST EQ 2      ;
                       57    D4   018F  1301          CLRL     R7                        ; ASSUME ONLY FREELIST SCAN
             0000'DF48       B5   0191  1302          TSTW     @W^PHV$GL_PIXBAS[R8]      ; IS THIS A DELETED PROCESS HEADER?
                       03    18   0196  1303          BGEQ     25$                       ; BR IF NOT.  ONLY SCAN FREELIST
                 57    02    D0   0198  1304          MOVL     #PFN$C_BADPAGLST,R7       ; INITIALIZE LOOP SCAN TO BADPAGLST
       50    0000'CF47       D0   019B  1305 25$:     MOVL     W^PFN$AL_HEAD[R7],R0      ; GET HEAD OF LIST TO START SCAN
                       3E    13   01A1  1306          BEQL     45$                       ; NO PAGES, DONE
                                  01A3  1307 30$:              PFN_REFERENCE    -
                                  01A3  1308          MOVZWL   <@W^PFN$Ax_FLINK[R0],R9>,-       ; GET FORWARD LINK
                                  01A3  1309                   LONG_OPCODE=MOVL,-
                                  01A3  1310                   IMAGE=SYS_NONPAGED
       53    0000'DF40       D0   01A9  1311          MOVL     @W^PFN$AL_PTE[R0],R3      ; GET SVA OF PTE FOR PAGE
                                  01AF  1312          ASSUME   PFN$C_PPGTBL EQ 4
                                  01AF  1313          ASSUME   PFN$C_GPGTBL EQ 5
  51   0000'DF40   01    02  EF   01AF  1314          EXTZV    #2,#1,@W^PFN$AB_TYPE[R0],R1        ; GET PAGE TABLE BIT
             08 AE41   53    D1   01B7  1315          CMPL     R3,8(SP)[R1]              ; COMPARE WITH LOW LIMIT
                       1E    1F   01BC  1316          BLSSU    40$                       ; OUT OF RANGE
                6E41   53    D1   01BE  1317          CMPL     R3,(SP)[R1]               ; COMPARE WITH HIGH LIMIT
                       18    1E   01C2  1318          BGEQU    40$                       ; OUT OF RANGE
                 0C    57    E9   01C4  1319          BLBC     R7,35$                    ; BR IF FREE OR BAD LIST
       52  00000000'FF40     D0   01C7  1320          MOVL     @PFN$AL_BAK[R0],R2        ; GET BACKING STORE ADDRESS
              09 52    16    E0   01CF  1321          BBS      #PTE$V_TYPO,R2,40$        ; LEAVE MODIFIED SECTION PAGES
                 52    57    D0   01D3  1322 35$:     MOVL     R7,R2                     ; SET LIST NUMBER FOR DELETE
                    FE27'    30   01D6  1323          BSBW     MMG$REMPFN                ; REMOVE PAGE FROM FREE LIST
                     0895    30   01D9  1324          BSBW     RELDELPAGE                ; RELEASE PAGE DELETING CONTENT
                 50    59    D0   01DC  1325 40$:     MOVL     R9,R0                     ; FLINK TO NEXT PAGE
                       C2    12   01DF  1326          BNEQ     30$                       ; ANOTHER PAGE TO TRY
                 B7    57    F4   01E1  1327 45$:     SOBGEQ   R7,25$                    ; NEXT LIST
                 5E    10    C0   01E4  1328          ADDL     #16,SP                    ; CLEAN STACK OF LIMITS
             0000'DF48       B5   01E7  1329          TSTW     @W^PHV$GL_REFCBAS[R8]     ; DID WE FREE PROCESS HEADER
                       0E    13   01EC  1330          BEQL     60$                       ; YES, RELEASE IT
                                  01EE  1331 ;
                                  01EE  1332 ; THERE ARE TWO REASONS THAT MIGHT PREVENT THE HEADER FROM BEING RELEASED BY
                                  01EE  1333 ; THE FREELIST SCAN:
                                  01EE  1334 ;                           1. SOME OF THE TRANSITION PAGES ARE ON THE MODIFED LIST.
                                  01EE  1335 ;                           2. THERE IS I/O IN PROGRESS ON THE TRANSITION PAGES.
```

```
                         01EE   ; TO COVER THE FORMER CASE (SINCE WE CANT REALLY TELL), THE MODIFIED LIST MUST
                         01EE   ; BE TOTALLY FLUSHED.  HOWEVER THIS IS ACTUALLY QUITE RARE.
                         01EE   ;
         0000'CF    D4   01EE           CLRL    W^SCH$GL_MFYLOLIM       ; FORCE ENTIRE MODIFY LIST TO BE WRITTEN
         0000'CF    B4   01F2           CLRW    W^SCH$GL_MFYLIM         ; CLEAR PART OF HI LIMIT, NOT PART THAT
                         01F6                                           ; INDICATES MODIFIED WRITING IN PROGRESS
          59  00    D2   01F6           MCOML   #0,R9                   ; NO, TRY FOR ANOTHER
            FF50    31   01F9           BRW     20$                     ; NOW ATTEMPT CLEANUP
            0117    31   01FC   60$:    BRW     RELPHD                  ; GO RELEASE PROCESS HEADER
                         01FF   70$:                                    ;
                         01FF   ;
                         01FF   ;       R4 - PCB OF OUTSWAP CANDIDATE, ALREADY MARKED NON-RESIDENT
                         01FF   ;
                         01FF
      55   6C A4    D0   01FF           MOVL    PCB$L_PHD(R4),R5        ; GET PROCESS HEADER ADDRESS
                         0203
         0018'CF    B7   0203           DECW    W^SWP$GW_BALCNT         ; DECREASE NUMBER IN BALANCE SET
            0857    30   0207           BSBW    OSINIT                  ; INIT REGISTERS FOR SCAN
           30 A4    B4   020A           CLRW    PCB$W_APTCNT(R4)        ; INITIALIZE ACTIVE PAGE TABLE COUNT
      57   08 A5    3C   020D           MOVZWL  PHD$W_WSLIST(R5),R7     ; WS INDEX FOR PERM PAGES
      56   12 A5    3C   0211           MOVZWL  PHD$W_WSLAST(R5),R6     ; END OF WORKING SET LIST
                         0215   ;
                         0215   ;       REGISTER CONVENTIONS FOR OWSLOOP ARE:
                         0215   ;
                         0215   ;       R0 - PFN
                         0215   ;       R1 - SCRATCH, WSLX
                         0215   ;       R2 - WORKING SET LIST ENTRY (VIRTUAL ADDRESS+FLAGS)
                         0215   ;       R3 - SVA OF PTE FOR WORKING SET LIST ENTRY
                         0215   ;       R4 - PCB ADDRESS
                         0215   ;       R5 - PHD ADDRESS
                         0215   ;       R6 - END INDEX TO WORKING SET LIST
                         0215   ;       R7 - WSLX (WORKING SET LIST INDEX)
                         0215   ;       R8 - PTE CONTENT
                         0215   ;       R9 - WORKING POINTER TO SWP$AL_MAP
                         0215   ;       R10 - PTE$M_VALID!PTE$C_ERKW
                         0215   ;       R11 - BASE ADDRESS OF SWP$AL_MAP
                         0215   ;
                         0215   OWSLOOP:                                ; OUTSWAP WS LOOP
         52  6547   D0   0215           MOVL    (R5)[R7],R2             ; GET WORKING SET LIST ENTRY
           15 52    E9   0219           BLBC    R2,NOTVALID             ; SKIP IF NOT VALID
           FDE1'    30   021C           BSBW    MMG$SVAPTECHK           ; CONVERT VA TO SVA OF PTE
                         021F   ;
                         021F   ;       R3 <- SVA OF PTE FOR VA IN R2
                         021F   ;
          58  63    D0   021F           MOVL    (R3),R8                 ; GET CONTENT OF PTE
              02    19   0222           BLSS    10$                     ; CONTINUE IF VALID PAGE
              52    D7   0224           DECL    R2                      ; CLEAR VALID FLAG
        52  E0 8F   8A   0226   10$:    BICB    #^C<WSL$M_VALID!WSL$M_PAGTYP!WSL$M_PFNLOCK>,R2; ISOLATE INTERESTING
   50   58  15  00  EF   022A           EXTZV   #PTE$V_PFN,#PTE$S_PFN,R8,R0     ; GET PFN FROM PTE
              06    10   022F           BSBB    OSDISPATCH              ; DISPATCH ON PAGE TYPE
                         0231   NOTVALID:                               ;
        E0 57 56    F3   0231           AOBLEQ  R6,R7,OWSLOOP           ; PROCESS ENTIRE WORKING SET LIST
              21    11   0235           BRB     PROCWRT                 ; DONE WITH WORKING SET LIST, RESET HEADER
                         0237   OSDISPATCH:                             ;
                         0237           ASSUME  WSL$V_VALID   EQ  0
                         0237           ASSUME  WSL$V_PAGTYP  EQ  1
                         0237           ASSUME  WSL$V_PFNLOCK EQ  4
```

```
                      0237  1393            ASSUME  PFN$C_PROCESS EQ  0
                      0237  1394            ASSUME  PFN$C_SYSTEM  EQ  1
                      0237  1395            ASSUME  PFN$C_GLOBAL  EQ  2
                      0237  1396            ASSUME  PFN$C_GBLWRT  EQ  3
                      0237  1397            ASSUME  PFN$C_PPGTBL  EQ  4
                      0237  1398            ASSUME  PFN$C_GPGTBL  EQ  5
        5D   6547  DE 0237  1399            MOVAL   (R5)[R7],FP                  ; COMPUTE ADDRESS OF WSL ENTRY
                      023B  1400            CASE    R2,<-                        ; SWITCH ON WSL PAGE TYPE + PTE VALID BIT
                      023B  1401                    PROCTRANS,-                  ; 0 => PROCESS TRANSITION PAGE
                      023B  1402                    PROCVALID,-                  ; 1 => PROCESS VALID PAGE
                      023B  1403                    WSLERR,-                     ; 2 => ????  BUGCHECK
                      023B  1404                    WSLERR,-                     ; 3 => ????  BUGCHECK
                      023B  1405                    GBLTRANS,-                   ; 4 => GLOBAL TRANSITION
                      023B  1406                    GBLVALID,-                   ; 5 => GLOBAL VALID
                      023B  1407                    GBLWRTTRANS,-                ; 6 => GLOBAL WRITABLE TRANSITION
                      023B  1408                    GBLWRTVALID-                 ; 7 => GLOBAL WRITABLE VALID
                      023B  1409                    PPGTBLTRANS,-                ; 8 => PROCESS PAGE TABLE TRANSITION
                      023B  1410                    PPGTBLVALID,-                ; 9 => PROCESS PAGE TABLE VALID
                      023B  1411                    >,TYPE=B                     ;
                   05 0253  1412            RSB                                  ; SKIP PFN LOCK PAGES
                      0254  1413
                      0254  1414 SPACEFAIL:
                      0254  1415            BUG_CHECK INSSWPFIL,FATAL            ; INSUFFICIENT SWAP FILE SPACE
                      0258  1416
                      0258  1417 PROCWRT:                                        ; RESET PROCESS HEADER BASE REGISTERS
     52    20 A4   D0 0258  1418            MOVL    PCB$L_WSSWP(R4),R2           ; GET SWAP ADDRESS
              F6   15 025C  1419            BLEQ    SPACEFAIL                    ; BRANCH IF NO VBN AVAILABLE TO USE
     50    30 A4   3C 025E  1420            MOVZWL  PCB$W_APTCNT(R4),R0          ; GET COUNT OF ACTIVE PAGE TABLES
   0014'CF    54   D0 0262  1421            MOVL    R4,W^OSWPPCB                 ; SAVE ADDRESS OF OUTSWAP PROCESS
        59    5B   C2 0267  1422            SUBL    R11,R9                       ; COMPUTE NUMBER OF PAGES * 4
  54    59    1E   9C 026A  1423            ROTL    #<32-2>,R9,R4                ; DIVIDE COUNT BY 4
     52 A5    54   B1 026E  1424            CMPW    R4,PHD$W_SWAPSIZE(R5)        ; DO WE HAVE ENOUGH SPACE FOR SWAP
              E0   1A 0272  1425            BGTRU   SPACEFAIL                    ; BRANCH IF NOT, THIS IS FATAL
   0012'CF    54   B0 0274  1426            MOVW    R4,W^OSWPPGS                 ; SAVE COUNT OF OUTSWAP PAGES
        53    5B   D0 0279  1427            MOVL    R11,R3                       ; SVAPTE FOR OUTSWAP I/O
        52    50   C0 027C  1428            ADDL    R0,R2                        ; SKIP HEADER AND ACTIVE PAGE TABLES
```

```
                                 027F
                                 027F       ;----------------------------------------------------------------------
                                 027F       ;
                                 027F       ;       DO OUTSWAP I/O FOR PROCESS HEADER AND BODY
                                 027F       ;
                                 027F       ;----------------------------------------------------------------------
                                 027F
             0000'CF     D6      027F               INCL    W^SWP$GL_OSWPCNT                ; ACCOUNT FOR OUTSWAP
                0808     30      0283               BSBW    SWPWRITE                        ; WRITE HEADER AND BODY
               04 50     E8      0286               BLBS    R0,20$                          ; CONTINUE IF NO I/O ERROR
                                 0289               BUG_CHECK OUTSWPERR,FATAL               ; **** OUT SWAP I/O ERROR
                                 028D       20$:                                            ;
                                 028D
                07C7     30      028D               BSBW    RELINIT                         ; INIT REGISTERS FOR RELEASE LOOP
         55   6C A4      D0      0290               MOVL    PCB$L_PHD(R4),R5                ; GET POINTER TO PHD
         58   42 A5      3C      0294               MOVZWL  PHD$W_PHVINDEX(R5),R8           ; GET PROCESS HEADER SLOT INDEX
       50   89   5A      CB      0298       30$:    BICL3   R10,(R9)+,R0                    ; GET PAGE NUMBER TO RELEASE
 02   0000'DF40   03 00  ED      029C               CMPZV   #PFN$V_PAGTYP,#PFN$S_PAGTYP,@W^PFN$AB_TYPE[R0],#PFN$C_GLOBAL   ;
                  62     13      02A4               BEQL    80$                             ; PAGE IS GLOBAL, COMPLEX CLEANUP
        56   0000'DF40   D0      02A6               MOVL    @W^PFN$AL_PTE[R0],R6            ; GET POINTER TO PAGE TABLE FOR  PAGE
     66   84000000 8F    CA      02AC               BICL    #<PTE$M_VALID!PTE$M_MODIFY>,(R6); CLEAR VALID AND MODIFY
           0000'DF40     B5      02B3               TSTW    @W^PFN$AW_SWPVBN[R0]            ; WAS I/O IN PROGRESS?
                  25     13      02B8               BEQL    40$                             ; NO, DONT MARK PAGE MODIFIED
    0000'DF40   80 8F    88      02BA               BISB    #PFN$M_MODIFY,@W^PFN$AB_STATE[R0] ; MARK PAGE MODIFIED
               03 00     ED      02C1               CMPZV   #PFN$V_LOC,#PFN$S_LOC,-         ; IF THIS WAS READ IN PROGRESS
        04   0000'DF40           02C4                       @W^PFN$AB_STATE[R0],#PFN$C_RDERR ; AND IS NOW PAGE READ ERROR
                  14     12      02C9               BNEQ    40$                             ;
                                 02CB               DECREF                                  ; AND IF THIS IS THE LAST REFERENCE
                  2B     12      02D5               BNEQ    60$                             ;
             52   02     9A      02D7               MOVZBL  #PFN$C_BADPAGLST,R2             ; THEN DIVERT THE PAGE TO
               FD23'     30      02DA               BSBW    MMG$INSPFNT                     ; THE BAD PAGE LIST
                  23     11      02DD               BRB     60$                             ;
                                 02DF       40$:    DECREF                                  ; DECREMENT REFERENCE COUNT FOR PAGE
                  05     12      02E9               BNEQ    55$                             ; NOT RELEASABLE YET
               FD12'     30      02EB       50$:    BSBW    MMG$RELPFN                      ; RELEASE PFN AS APPROPRIATE
                  12     11      02EE               BRB     60$                             ; GO FOR NEXT PAGE
 07   0000'DF40   03 00  ED      02F0       55$:    CMPZV   #PFN$V_LOC,#PFN$S_LOC,@W^PFN$AB_STATE[R0],- ;
                                 02F8                       #PFN$C_ACTIVE                   ; IS STATE ACTIVE?
                  08     12      02F8               BNEQ    60$                             ; NO, THEN LEAVE UNCHANGED
                  03     F0      02FA               INSV    #PFN$C_RELPEND,-                ; MAKE STATE BE RELEASE PENDING
      0000'DF40   03 00          02FC                       #PFN$V_LOC,#PFN$S_LOC,@W^PFN$AB_STATE[R0]; IF SOME I/O OUTSTANDING
               93 57     F5      0302       60$:    SOBGTR  R7,30$                          ; NEXT PAGE IN LIST
                000E     31      0305               BRW     RELPHD                          ; RELEASE PROCESS HEADER IF POSSIBLE
                                 0308       80$:    DECSHR  GTR=60$,-                       ; DECREASE SHARE COUNT FOR PAGE
                                 0308                       IMAGE_FLAG=SYS_NONPAGED
                  C9     11      0314               BRB     40$                             ; RELEASE PAGE TO FREE LIST IF REFCNT=0
```

```
                                 0316          .SBTTL  RELPHD - RELEASE PROCESS HEADER
                                 0316
                                 0316  ;++
                                 0316  ; FUNCTIONAL DESCRIPTION:
                                 0316  ;       RELPHD CHECKS THE REFERENCE  COUNT ON THE PROCESS HEADER
                                 0316  ;       AND RELEASES THE PAGE TABLES FROM THE PROCESS HEADER WHEN ALL
                                 0316  ;       OF THEIR PAGES HAVE BEEN RELEASED.  THE PAGE TABLES ARE FIRST WRITTEN
                                 0316  ;       TO THE SWAP IMAGE IF THEY ARE MARKED AS UPDATED.
                                 0316  ;
                                 0316  ; CALLING SEQUENCE:
                                 0316  ;       BRW/JMP RELPHD
                                 0316  ;
                                 0316  ; INPUT PARAMETERS:
                                 0316  ;       R8 - BALANCE SLOT INDEX FOR HEADER TO BE RELEASED
                                 0316  ;
                                 0316  ; OUTPUT PARAMTERS:
                                 0316  ;       R0-R7,R9,R10 VOLATILE
                                 0316  ;
                                 0316  ; SIDE EFFECTS:
                                 0316  ;       THE PAGE TABLES FROM THE PROCESS HEADER MAY BE WRITTEN TO THE
                                 0316  ;       SWAP IMAGE FOR THE PROCESS IF THEY HAVE BEEN UPDATED.
                                 0316  ;
                                 0316  ;--
                                 0316
                                 0316
                                 0316  RELPHD:                                                  ;
             0000'DF48      B5   0316          TSTW    @W^PHV$GL_REFCBAS[R8]             ; SEE IF PROCESS HEADER IS RELEASABLE
                    03      13   031B          BEQL    5$                                ; YES, FREE ACTIVE PAGE TABLES
                  00E1      31   031D          BRW     OSWPEXIT                          ; NO, TRY LATER
        57   0000'CF        D0   0320  5$:     MOVL    W^SWP$GL_BSLOTSZ,R7               ; SET ITERATION COUNT TO WHOLE BALANCE SLOT
        51    58    57      C5   0325          MULL3   R7,R8,R1                          ; GET LONG WORD OFFSET TO SLOT
    56       0000'DF41      DE   0329          MOVAL   @W^SWP$GL_BALSPT[R1],R6           ; POINT TO BASE OF THIS SLOT
                  072F      30   032F          BSBW    OSINIT                            ; INIT REGISTERS FOR SCAN
    54       0000'DF48      32   0332          CVTWL   @W^PHV$GL_PIXBAS[R8],R4           ; GET INDEX TO PROCESS IN SLOT
                    43      19   0338          BLSS    12$                               ; BR IF DELETED PROCESS
    54       0000'DF44      D0   033A          MOVL    @W^SCH$GL_PCBVEC[R4],R4           ; AND TRANSLATE TO PCB ADDRESS
           55    6C A4      D0   0340          MOVL    PCB$L_PHD(R4),R5                  ; GET PROCESS HEADER ADDRESS
           6C A4    58      D0   0344          MOVL    R8,PCB$L_PHD(R4)                  ; INDICATE NO PHD FOR PROCESS
         00C8 C5    55      C2   0348          SUBL    R5,PHD$L_P0BR(R5)                 ; UNBIAS MEMORY MANAGEMENT BASE REGISTERS
         00D0 C5    55      C2   034D          SUBL    R5,PHD$L_P1BR(R5)                 ; FOR BOTH P0 AND P1 SPACE
        00 24 A4    12      E5   0352          BBCC    #PCB$V_PHDRES,PCB$L_STS(R4),7$  ; MARK PHD NON-RESIDENT
           5C    48 A5      D0   0357  7$:     MOVL    PHD$L_WSLX(R5),AP                 ; GET POINTER TO WSLX SAVE AREA
             5C   654C      DE   035B          MOVAL   (R5)[AP],AP                       ; AND CONVERT TO BYTE ADDRESS
           5D    44 A5      D0   035F          MOVL    PHD$L_BAK(R5),FP                  ; GET POINTER TO BACKING STORE VECTOR
             5D   654D      DE   0363          MOVAL   (R5)[FP],FP                       ; AND CONVERT TO BYTE ADDRESS
               00D8 C5      B4   0367          CLRW    PHD$W_EMPTPG(R5)                  ; CLEAR COUNT OF EMPTY WSL PAGES
                 8D 86      D0   036B  10$:    MOVL    (R6)+,(FP)+                       ; COPY ENTRY FROM SPT
                    10      19   036E          BLSS    15$                               ; BR IF VALID
                    04      12   0370          BNEQ    11$                               ; BR IF NOT EMPTY WSL PAGE
               00D8 C5      B6   0372          INCW    PHD$W_EMPTPG(R5)                  ; COUNT EMPTY WSL PAGES
                 FC A6      D4   0376  11$:    CLRL    -4(R6)                            ; ZAP INVALID ENTRY TO NO-ACCESS
                    8C      B4   0379          CLRW    (AP)+                             ; AND CLEAR WSLX VALUE FOR PAGE
                    23      11   037B          BRB     20$                               ;
                  0084      31   037D  12$:    BRW     DELPHD                            ; FINISH DELETE FOR PROCESS
   50    FC AD    15    00      EF   0380  15$:    EXTZV   #PTE$V_PFN,#PTE$S_PFN,-4(FP),R0 ; GET PFN FOR VALID ENTRY
                    EE      13   0386          BEQL    11$                               ; DEMAND ZERO OR NULL PTE
FC AD   16   00  0000'DF40      F0   0388          INSV    @W^PFN$AL_BAK[R0],#PTE$V_PGFLVB,#PTE$S_PGFLVB,-4(FP)     ; SAVE BACKU
```

```
                        0391   1536               PFN_REFERENCE    -
                        0391   1537         MOVW  <aW^PFN$Ax_WSLX[R0],(AP)+>,-     ; AND WORKING SET LIST INDEX
                        0391   1538               LONG_OPCODE=CVTLW,-
                        0391   1539               IMAGE=SYS_NONPAGED
  89    5A    50    C9  0397   1540         BISL3   R0,R10,(R9)+                  ; SET INTO SWAPPER MAP
 00 FC AD     1F    E2  039B   1541         BBSS    #PTE$V_VALID,-4(FP),20$       ; MARK PAGE VALID FOR INSWAP PURPOSES
        C8    57    F5  03A0   1542 20$:    SOBGTR  R7,10$                        ; SCAN ENTIRE BALANCE SLOT
        59    5B    C2  03A3   1543         SUBL    R11,R9                        ; COMPUTE NUMBER OF PAGES * 4
    52    20  A4    D0  03A6   1544         MOVL    PCB$L_WSSWP(R4),R2            ; WORKING SET SWAP SLOT
   0014'CF    54    D0  03AA   1545         MOVL    R4,W^OSWPPCB                  ; SAVE PCB ADDRESS FOR SLOT OWNER
  54    59    1E    9C  03AF   1546         ROTL    #<32-2>,R9,R4                 ; DIVIDE COUNT BY 4
   0012'CF    54    B0  03B3   1547         MOVW    R4,W^OSWPPGS                  ; SAVE COUNT OF OUTSWAP PAGES
        53    5B    D0  03B8   1548         MOVL    R11,R3                        ; SET SVA OF MAP FOR I/O
         0000'CF    D6  03BB   1549         INCL    W^SWP$GL_HOSWPCNT             ; ACCOUNT FOR HEADER OUTSWAP
            06CC    30  03BF   1550         BSBW    SWPWRITE                      ; WRITE ACTIVE PAGE TABLES
           04 50    E8  03C2   1551         BLBS    R0,30$                        ; CONTINUE IF NO ERROR
                        03C5   1552         BUG_CHECK APTWRTERR,FATAL             ; **** ACTIVE PAGE TABLE SWAP I/O ERROR
            068B    30  03C9   1553 30$:    BSBQ    RELINIT                       ; INIT REGISTERS FOR RELEASE LOOP
    58    6C  A4    D0  03CC   1554         MOVL    PCB$L_PHD(R4),R8              ; RESTORE BALANCE SLOT INDEX
  50    89    5A    CB  03D0   1555 40$:    BICL3   R10,(R9)+,R0                  ; ISOLATE PAGE FRAME NUMBER
56    0000'DF40     D0  03D4   1556         MOVL    aW^PFN$AL_PTE[R0],R6          ; GET PTE ADDRESS
        66    50    D0  03DA   1557         MOVL    R0,(R6)                       ; MAKE PTE CORRECT BUT INVALID
      0000'DF40     B7  03DD   1558         DECW    aW^PFN$AW_REFCNT[R0]          ; DROP REFERENCE COUNT
              04    13  03E2   1559         BEQL    50$                           ; MUST BE ZERO
                        03E4   1560         BUG_CHECK APTREFHIGH,FATAL            ; INCONSISTENT PAGE TABLE REFERENCE COUNT
        53    56    D0  03E8   1561 50$:    MOV[    R6,R3                         ; SVAPTE FOR DELCON
            0683    30  03EB   1562         BSBW    RELDELPAGE                    ; RELEASE PAGE THROUGH DELCONPFN
              66    D4  03EE   1563         CLRL    (R6)                          ; SET NO ACCESS ON PFN
           DD 57    F5  03F0   1564         SOBGTR  R7,40$                        ; CONTINUE FOR ALL ACTIVE PAGE TABLES
0000'DF48     01    AE  03F3   1565         MNEGW   #1,aW^PHV$GL_REFCBAS[R8]            ; MARK BALANCE SLOT AVAIL
      0000'DF48     B4  03F9   1566         CLRW    aW^PHV$GL_PIXBAS[R8]          ; AND SET PIX TO NULL
           6C A4    D4  03FE   1567         CLRL    PCB$L_PHD(R4)                 ; AND SEVER CONNECTION WITH PROCESS
                        0401   1568
                        0401   1569 OSWPEXIT:                                     ; OUTSWAP COMPLETE
            05D3    31  0401   1570         BRW     SWAPRETRY                     ; RETRY SWAP SCHEDULE AFTER OUTSWAP
```

```
                          0404          .SBTTL  DELPHD - DELETE PROCESS HEADER FOR DELETED PROCESS
                          0404
                          0404    ;
                          0404    ; FUNCTIONAL DESCRIPTION:
                          0404    ;         DELPHD IS ENTERED BY RELPHD IF THE PROCESS INDEX ASSOCIATED WITH
                          0404    ;         THE BALANCE SLOT IS NEGATIVE INDICATING THE PROCESS HAS BEEN DELETED.
                          0404    ;         NOW THAT THE REFERENCE COUNT FOR THE HEADER IS ZERO, ALL PAGES AND
                          0404    ;         BACKING STORE PAGES CAN BE RELEASED PERMITTING RELEASE OF THE BALANCE
                          0404    ;         SLOT.  AT THIS POINT THE SPT ENTRIES ARE VALID WITH A PFN, DEMAND ZERO,
                          0404    ;         OR BACKING STORE ADDRESS FORM.  THERE ARE NO REMAINING TRANSITION PAGES.
                          0404    ;
                          0404    ; INPUT PARAMETERS:
                          0404    ;         R1 - PRODUCT OF SGN$C_BSLOTSZ * BALANCE_SLOT_INDEX
                          0404    ;         R6 - ADDRESS OF FIRST SPT ENTRY FOR THIS BALANCE SLOT
                          0404    ;         R7 - SGN$C_BSLOTSZ
                          0404    ;         R8 - BALANCE_SLOT_INDEX
                          0404    ;         R10- MASK OF PTE$M_VALID!PTE$M_MODIFY!PTE$C_ERKW
                          0404    ;
                          0404
                          0404    DELPHD:                                               ;
     55   51    09   9C   0404            ROTL    #9,R1,R5                      ; COMPUTE OFFSET TO PHD FROM BASE
     55   0000'CF    C0   0408            ADDL    W^SWP$GL_BALBASE,R5           ; FORM PHD ADDRESS
      5B    1F  A5   9A   040D            MOVZBL  PHD$B_PAGFIL(R5),R11          ; GET PAGING FILE NUMBER
          50    86   D0   0411    10$:    MOVL    (R6)+,R0                      ; GET PTE FROM SPT
                2C   13   0414            BEQL    40$                           ; BR IF EMPTY
                04   19   0416            BLSS    20$                           ; BR IF VALID
       16 50    1A   E0   0418            BBS     #PTE$V_TYP1,R0,25$            ; BR IF TYPE 1 (BACKING STORE)
          50    5A   CA   041C    20$:    BICL    R10,R0                        ; ISOLATE PFN
                1E   13   041F            BEQL    30$                           ; SKIP DEMAND ZERO PTE
     59    0000'DF40 D0   0421            MOVL    @W^PFN$AL_BAK[R0],R9          ; GET BACKUP ADDRESS
   FF A6     84 8F   8A   0427            BICB    #<<PTE$M_VALID!PTE$M_MODIFY>@-24>,-1(R6) ; CLEAR VALID AND MODIFY
              0642   30   042C            BSBW    RELDELPAGE                    ; RELEASE PAGE
          50    59   D0   042F            MOVL    R9,R0                         ; GET BACKUP ADDRESS
 50   50    16  00   EF   0432    25$:    EXTZV   #PTE$V_PGFLVB,#PTE$S_PGFLVB,R0,R0       ; GET PAG FIL VB
                06   13   0437            BEQL    30$                           ; BR IF NONE
          53    5B   D0   0439            MOVL    R11,R3                        ; SET PAGING FILE NUMBER FOR RELEASE
             FBC1'   30   043C            BSBW    MMG$DALCPAGFIL                ; DEALLOCATE PAGING FILE PAGE
             FC A6   D4   043F    30$:    CLRL    -4(R6)                        ; ZAP SPT ENTRY
             CC 57   F5   0442    40$:    SOBGTR  R7,10$                        ; RELEASE ENTIRE HEADER
                          0445            INVALID                               ; INVALIDATE HEADER
  0000'DF48     01   AE   0448            MNEGW   #1,@W^PHV$GL_REFCBAS[R8]              ; MARK SLOT EMPTY
         0000'DF48   B4   044E            CLRW    @W^PHV$GL_PIXBAS[R8]          ; POINT OWNER PIX AT NULL PROCESS
           0000'CF   B7   0453            DECW    W^SCH$GW_DELPHDCT             ; ACCOUNT FOR DELETED HEADER
              057D   31   0457            BRW     SWAPRETRY                     ; AND RETRY SWAP ATTEMPT
```

```
                                     045A       .SBTTL  GBLTRANS/GBLVALID/GBLWRTVALID - HANDLE GLOBAL PAGES
                                     045A
                                     045A  ;
                                     045A  ;        GBLTRANS HANDLES THE CASE OF A GLOBAL PAGE IN TRANSITION,
                                     045A  ;        WHICH IMPLIES THAT THE PAGE HAS BEEN FAULTED BUT IS NOT YET
                                     045A  ;        RESIDENT.  THE WORKING SET LIST ENTRY FOR THIS PAGE WILL BE
                                     045A  ;        DELETED AND THE PAGE WILL HAVE TO BE FAULTED AGAIN.
                                     045A  ;
                                     045A
                                     045A  GBLTRANS:                                       ; TRANSITION GLOBAL PAGE
                                     045A  GBLWRTTRANS:                                    ; TRANSITION WRITABLE GLOBAL PAGE
 50   15   00   0000'DF40      F0    045A       INSV    aW^MMG$GL_GPTBASE[R0],#PTE$V_PFN,#PTE$S_PFN,R0  ; GET GLOBAL PFN FRO
                                     0462
                                     0462       .ENABL  LSB
                                     0462  GBLDROP:                                        ; DROP GLOBAL PAGE FROM WORKING SET
                     51   57       D0    0462       MOVL    R7,R1                           ; GET WSL INDEX FOR RELEASE
                          53       DD    0465       PUSHL   R3                              ; SAVE SVAPTE FOR FOLLOWING DECPTREF
                        FB96'      30    0467       BSBW    MMG$DELWSLEX                    ; DELETE WSL GIVEN INDEX
                          08       BA    046A       POPR    #^M<R3>                         ; RESTORE SVAPTE
                        FB91'      30    046C       BSBW    MMG$DECPTREF                    ; AND DROP PAGE TABLE REFERENCE
                                     046F       DECSHR  GTR=10$,-                       ; DECREASE SHARE COUNT
                                     046F               IMAGE_FLAG=SYS_NONPAGED
                                     047B  PROCDROP:
                                     047B       DECREF  GTR=20$                         ; AND REF COUNT IF LAST SHARER
                        FB76'      30    0487       BSBW    MMG$RELPFN                      ; RELEASE PAGE IF LAST REFERENCE
                                     048A  10$:                                            ;
                                   05    048A       RSB                                     ; RETURN FOR NEXT PAGE
 07   0000'DF40   03     00      ED    048B  20$:  CMPZV   #PFN$V_LOC,#PFN$S_LOC,aW^PFN$AB_STATE[R0],- ;
                                     0493               #PFN$C_ACTIVE                   ; CHECK FOR ACTIVE STATE
                          08       12    0493       BNEQ    30$                             ; NO, THEN LEAVE STATE UNCHANGED
                          03       F0    0495       INSV    #PFN$C_RELPEND,-                ; SET STATE TO RELEASE PENDING IF
      0000'DF40   03     00            0497               #PFN$V_LOC,#PFN$S_LOC,aW^PFN$AB_STATE[R0] ; I/O OUTSTANDING
                                   05    049D  30$:  RSB                                     ;
                                     049E       .DSABL  LSB
                                     049E
                                     049E  ;
                                     049E  ;        GBLVALID HANDLES A VALID, NON-WRITABLE, PAGE.
                                     049E  ;
                                     049E  GBLVALID:                                       ; VALID GLOBAL PAGE
                                     049E  ;
                                     049E  ;        IF THE GLOBAL PAGE IS BEING ACTIVELY SHARED THEN IT WILL BE DROPPED
                                     049E  ;        FROM THE WORKING SET AND REFAULTED LATER (PRESUMABLY WITHOUT I/O).
                                     049E  ;
               08 6D     05      E0    049E       BBS     #WSL$V_WSLOCK,(FP),10$  ; DON'T DROP PAGES LOCKED IN WORKING SET
                                     04A2               PFN_REFERENCE    -
                                     04A2       CMPW    <#1,aW^PFN$Ax_SHRCNT[R0]>,-      ; IS THERE ACTIVE SHARING OF THIS PA
                                     04A2               LONG_OPCODE=CMPL,-
                                     04A2               IMAGE=SYS_NONPAGED
                         25      12    04A8       BNEQ    GBLWRTVALID                     ; YES, DROP IT AND REFAULT LATER
                                     04AA  ;
                                     04AA  ;        OTHERWISE THE PAGE WILL BE WRITTEN TO THE SWAP IMAGE SINCE IT IS
                                     04AA  ;        UNLIKELY TO BE RESIDENT UPON INSWAP.
                                     04AA  ;
            89   5A      50      C9    04AA  10$:  BISL3   R0,R10,(R9)+                    ; SET IN SWAPPER MAP FOR OUT SWAP
                       FB4F'     30    04AE       BSBW    MMG$DECPTREF                    ; DROP PAGE TABLE REFERENCE FOR PAGE
                                     04B1  GBLRESET:                                       ; RESET SLAVE PTE TO GPTX FORMAT
 51   0000'DF40   0000'CF        C3    04B1       SUBL3   W^MMG$GL_GPTBASE,aW^PFN$AL_PTE[R0],R1   ; GET GPTX FOR PAGE
```

```
          51    51    1E   9C  04BA            ROTL    #<32-2>,R1,R1             ; AND CONVERT TO CORRECT SCALE
                               04BE            ASSUME  PTE$V_TYP0 EQ PTE$S_GPTX
             00 51    16   E2  04BE            BBSS    #PTE$V_TYP0,R1,10$        ; MARK AS GLOBAL
  52    63    845FFFFF 8F  CB  04C2    10$:    BICL3   #<PTE$M_VALID ! -         ; OBTAIN PERMANENT BITS FOR PTE
                               04CA                       PTE$M_TYP0  ! -        ; BY CLEARING ALL OTHERS
                               04CA                       PTE$M_TYP1  ! -        ;
                               04CA                       PTE$M_PFN>,(R3),R2     ; TO FORM TRANSITION GLOBAL PTE
          63    52    51   C9  04CA            BISL3   R1,R2,(R3)                ; MUST SET ENTIRE PTE AT ONE TIME
                               04CE                                              ; SO THAT I/O CAN SEE CONSISTENT PTE
                          05   04CE            RSB                               ; RETURN FOR NEXT PAGE
                               04CF
                               04CF    ;
                               04CF    ;       GBLWRTVALID HANDLES THE CASE OF A WRITABLE GLOBAL PAGE.
                               04CF    ;       SUCH PAGES ARE DROPPED FROM THE WORKING SET BEFORE OUTSWAPPING
                               04CF    ;       AND MUST BE SUBSEQUENTLY RE-FAULTED.
                               04CF    ;
                               04CF    GBLWRTVALID:                              ; VALID WRITABLE GLOBAL PAGE
             07 63    1A   E5  04CF            BBCC    #PTE$V_MODIFY,(R3),10$    ; TEST AND CLEAR MODIFY BIT IN SLAVE PTE
        0000'DF40  80 8F   88  04D3            BISB    #PFN$M_MODIFY,@W^PFN$AB_STATE[R0] ; AND SAVE MODIFY STATE
                      D5   10  04DA    10$:    BSBB    GBLRESET                  ; RESET PTE
                      84   11  04DC            BRB     GBLDROP                   ; DELETE WORKING SET LIST ENTRY
```

```
                          04DE        .SBTTL  PROCTRANS - PROCESS PAGE IN TRANSITION
                          04DE
                          04DE  ;
                          04DE  ;       THIS PAGE IS IN TRANSITION DUE TO THE FACT THAT THE PAGE FAULT
                          04DE  ;       READ OPERATION HAS NOT YET COMPLETED.  IT IS TREATED AS AN
                          04DE  ;       I/O IN PROGRESS PAGE.
                          04DE  ;
                          04DE
                          04DE  PROCTRANS:                                      ; PROCESS PAGE IN TRANSITION
          03   00     ED  04DE          CMPZV   #PFN$V_LOC,#PFN$S_LOC,-       ; IF THIS PAGE COULD NOT
   04    0000'DF40        04E1                  @W^PFN$AB_STATE[R0],#PFN$C_RDERR ; BE SUCCESSFULLY READ
               08     12  04E6          BNEQ    PROCVALID                     ;
          51   57     D0  04E8          MOVL    R7,R1                         ; DROP IT FROM THE WORKING SET
             FB12'    30  04EB          BSBW    MMG$DELWSLEX                  ; DELETE THE WSL ENTRY GIVEN WSL INDEX
               8B     11  04EE          BRB     PROCDROP                      ; AND RELEASE THE PFN IF LAST REF
                          04F0
                          04F0  ;
                          04F0  ;       PROCVALID HANDLES THE CASE OF A VALID PROCESS PAGE WHICH INCURS
                          04F0  ;       SOME SPECIAL PROCESSING IF THERE IS I/O IN PROGRESS.  AN I/O IN
                          04F0  ;       PROGRESS PAGE IS SWAPPED WITH THE BODY OF THE PROCESS TO RESERVE
                          04F0  ;       SPACE FOR IT IN THE SWAP IMAGE AND IS LATER WRITTEN WITH CORRECT
                          04F0  ;       CONTENT BY THE MODIFIED PAGE WRITER TO THIS RESERVED SPACE IN THE
                          04F0  ;       SWAP IMAGE.
                          04F0  ;
                          04F0  PROCVALID:                                      ; PROCESS VALID PAGE
                          04F0          .ENABL  LSB
                          04F0  10$:                                            ;
04 0000'DF40   07     E4  04F0          BBSC    #PFN$V_MODIFY,@W^PFN$AB_STATE[R0],20$    ; BR IF PAGE MODIFIED
       04 63   1A     E1  04F7          BBC     #PTE$V_MODIFY,(R3),30$        ; BR IF PAGE NOT MODIFIED
       00 6D   08     E2  04FB  20$:    BBSS    #WSL$V_MODIFY,(FP),30$        ; SET WORKING SET MODIFIED BIT
                          04FF  30$:                                            ;
   0000'DF40   01     B1  04FF          CMPW    #1,@W^PFN$AW_REFCNT[R0]       ; CHECK FOR I/O OUTSTANDING
               14     13  0505          BEQL    40$                           ; NO, NONE
          04   52     E9  0507          BLBC    R2,SETWRTBAK                  ; BRANCH IF TRANSITION PAGE
       0D 6D   08     E1  050A          BBC     #WSL$V_MODIFY,(FP),40$        ; DONT WRITE UNMODIFIED PAGES
                          050E  SETWRTBAK:                                      ; SET PAGE FOR WRITE BACK TO SWAP FILE
     51   59   5B     C3  050E          SUBL3   R11,R9,R1                     ; GET OFFSET TO PAGE IN SWAP MAP
          51   04     C6  0512          DIVL    #4,R1                         ; SCALE BACK TO PAGE NUMBER
   0000'DF40   51     B0  0515          MOVW    R1,@W^PFN$AW_SWPVBN[R0]       ; SET OFFSET INTO SWAP IMAGE LESS APTCNT
                          051B  40$:                                            ;
     89   5A   50     C9  051B          BISL3   R0,R10,(R9)+                  ; PUT PAGE IN SWAPPER MAP
                          051F  ;
                          051F  ;       SET DELETE CONTENT FLAG TO CAUSE PAGE TO BE PLACED AT HEAD
                          051F  ;       OF FREE PAGE LIST AND CONTENT FORGOTTEN.
                          051F  ;
   0000'DF40   10     88  051F  DELCON: BISB    #PFN$M_DELCON,@W^PFN$AB_STATE[R0] ; SET TO DELETE CONTENT
                      05  0525          RSB                                   ; RETURN FOR NEXT PAGE
                          0526          .DSABL  LSB                           ;
                          0526
                          0526  WSLERR: BUG_CHECK IVWSETLIST,FATAL            ; INVALID WORKING SET LIST ENTRY
```

```
                        052A   1751          .SBTTL  PAGE TABLE WORKING SET LIST ENTRIES
                        052A   1752
                        052A   1753 ;
                        052A   1754 ;        PAGE TABLE AND PROCESS HEADER ENTRIES IN THE WORKING SET LIST
                        052A   1755 ;        ARE IGNORED DURING THE PROCESS BODY OUTSWAP SCAN OF THE WORKING
                        052A   1756 ;        SET LIST.
                        052A   1757 ;
                        052A   1758
                        052A   1759 PPGTBLTRANS:                                   ; TRANSITION PAGE TABLE
                        052A   1760 PPGTBLVALID:                                   ; VALID PAGE TABLE
         30 A4     B6   052A   1761          INCW    PCB$W_APTCNT(R4)             ; ACCUMULATE ACTIVE PAGE TABLE COUNT
         6D  55    C2   052D   1762          SUBL    R5,(FP)                      ; UNBIAS WSL VA FOR PAGE TABLE
      00 6D  1F    E2   0530   1763          BBSS    #VA$V_SYSTEM,(FP),10$        ; BUT FORCE SYSTEM BIT ON IN VA
                   05   0534   1764 10$:     RSB                                  ; RETURN
```

```
                              0535  1767          .SBTTL  INSWAP
                              0535  1768
                              0535  1769  ;---------------------------------------------------------------------------
                              0535  1770  ;
                              0535  1771  ;       PERFORM REQUESTED INSWAP
                              0535  1772  ;
                              0535  1773  ;       INPUTS:
                              0535  1774  ;               R4 - PCB ADDRESS OF INSWAP CANDIDATE
                              0535  1775  ;---------------------------------------------------------------------------
                              0535  1776  INSWAP:                                         ; PERFORM INSWAP
        55    6C A4     D0    0535  1777          MOVL    PCB$L_PHD(R4),R5                ; GET CURRENT PROCESS HEADER SLOT
                 06     13    0539  1778          BEQL    10$                             ; NONE, MUST ALLOCATE ONE
        58    42 A5     3C    053B  1779          MOVZWL  PHD$W_PHVINDEX(R5),R8           ; GET BALANCE SLOT INDEX
                 2F     11    053F  1780          BRB     40$                             ; AND CONTINUE
                 58     D4    0541  1781  10$:    CLRL    R8                              ; INIT INDEX FOR BALANCE SLOT SEARCH
          0000'DF48     B5    0543  1782  20$:    TSTW    @W^PHV$GL_REFCBAS[R8]           ; CHECK FOR EMPTY
                 0F     19    0548  1783          BLSS    30$                             ; YES, GOT ONE
 F1 58    00000000'EF   F2    054A  1784          AOBLSS  SGN$GL_BALSETCT,R8,20$          ; TRY ALL BALANCE SET SLOTS
     5D     01   1F     9C    0552  1785          ROTL    #31,#1,FP                       ; SET FLAG TO PERMIT OUTSWAPPING
                              0556  1786                                                  ; OF PROCESSES
               FBBD     31    0556  1787          BRW     OUTSWAP                         ; OUTSWAP IF NECESSARY TO GET SLOT
  0000'DF48    60 A4    B0    0559  1788  30$:    MOVW    PCB$L_PID(R4),@W^PHV$GL_PIXBAS[R8]      ; SET PIX FOR BALANCE SET SL
          0000'DF48     B4    0560  1789          CLRW    @W^PHV$GL_REFCBAS[R8]           ; AND BUMP REFERENCE COUNT
  50    58    0000'CF   C5    0565  1790          MULL3   W^SWP$GL_BSLOTSZ,R8,R0          ; COMPUTE BALANCE SLOT OFFSET
     6C A4    50   09   9C    056B  1791          ROTL    #9,R0,PCB$L_PHD(R4)             ; MAKE BYTE OFFSET
                              0570  1792                                                  ; POSITIVE UNTIL I/O COMPLETE
                 59     D4    0570  1793  40$:    CLRL    R9                              ; INITIALIZE SWAPPER MAP INDEX
              FA8B'     30    0572  1794  50$:    BSBW    MMG$ALLOCPFN                    ; ALLOCATE A PAGE
                 50     D5    0575  1795          TSTL    R0                              ; MAKE SURE IT WAS ALLOCATED
                 04     18    0577  1796          BGEQ    60$                             ; YES, CONTINUE
                              0579  1797          BUG_CHECK INSNFREPAG,FATAL              ; INSUFFICIENT FREE PAGES
          0000'DF40     B6    057D  1798  60$:    INCW    @W^PFN$AW_REFCNT[R0]            ; REFERENCE PAGE
      0000'DF40   07    90    0582  1799          MOVB    #PFN$C_ACTIVE,@W^PFN$AB_STATE[R0]       ; AND MARK IT ACTIVE
0000'DF49   50   B0000000 8F  C9    0588  1800          BISL3   #<PTE$C_ERKW!PTE$M_VALID>,R0,@W^SWP$GL_MAP[R9] ; MARK VALID, WRITABL
           DB 59   5A   F2    0593  1801          AOBLSS  R10,R9,50$                      ; REPEAT FOR ALL REQUIRED PAGES
          0000'DF49     D4    0597  1802          CLRL    @W^SWP$GL_MAP[R9]               ; PUT STOPPER IN LIST
                              059C  1803  ;
                              059C  1804  ;       ALL PAGES HAVE NOW BEEN ACQUIRED AND A BALANCE SET SLOT
                              059C  1805  ;       ALLOCATED.  THE INSWAP I/O OPERATION CAN NOW BE PERFORMED.
                              059C  1806  ;
          0018'CF       B6    059C  1807          INCW    W^SWP$GW_BALCNT                 ; ADD ONE PROCESS TO BALANCE SET
  0000'CF      54       D0    05A0  1808          MOVL    R4,W^SWP$GL_INPCB               ; SAVE POINTER TO IN SWAP PCB
  0000'CF      5A       D0    05A5  1809          MOVL    R10,W^SWP$GL_ISPAGCNT           ; SAVE COUNT OF ALLOCATED PAGES
  0000'CF      58       B0    05AA  1810          MOVW    R8,W^SWP$GW_IBALSETX            ; AND BALANCE SET SLOT NUMBER
                              05AF  1811  ;---------------------------------------------------------------------------
                              05AF  1812  ;
                              05AF  1813  ;       PERFORM INSWAP I/O OPERATION
                              05AF  1814  ;
                              05AF  1815  ;---------------------------------------------------------------------------
                              05AF  1816
        52    20 A4     D0    05AF  1817          MOVL    PCB$L_WSSWP(R4),R2              ; GET SWAP IMAGE DISK ADDRESS
                 2A     15    05B3  1818          BLEQ    COPYSHELL                       ; BRANCH IF SHELL IN SWAP
     07 24 A4    12     E1    05B5  1819          BBC     #PCB$V_PHDRES,PCB$L_STS(R4),70$ ; SWAP EVERYTHING IF HEADER NON-RES
        50    30 A4     3C    05BA  1820          MOVZWL  PCB$W_APTCNT(R4),R0             ; GET ACTIVE PAGE TABLE COUNT
              52   50   C0    05BE  1821          ADDL    R0,R2                           ; ADD PAGE TABLE COUNT
     53    0000'DF      DE    05C1  1822  70$:    MOVAL   @W^SWP$GL_MAP,R3                ; SVA OF PAGE TABLE FOR I/O
              54   5A   D0    05C6  1823          MOVL    R10,R4                          ; NUMBER OF PAGES TO READ
        0000'CF    5A   C0  05C9          ADDL2   R10,W^SWP$GL_ISWPPAGES  ; UPDATE TOTAL PAGES INSWAPPED
          0000'CF       D6  05CE          INCL    W^SWP$GL_ISWPCNT        ; BUMP INSWAP COUNTER
             04B3       30  05D2          BSBW    SWPREAD                 ; PERFORM READ
            04 50       E8  05D5          BLBS    R0,80$                  ; BRANCH IF NO ERROR IN READ
                            05D8          BUG_CHECK       INSWAPERR,FATAL ; **** BUGCHECK ON I/O ERROR
             0089       31  05DC  80$:    BRW     SETUP                   ; SET UP PROCESS IN BALANCE SLOT
                            05DF
                            05DF  COPYSHELL:
     54   0000'CF       DE  05DF          MOVAL   W^MMG$AL_SYSPCB,R4      ; ADDRESS OF SYSTEM PCB
       55    6C A4      D0  05E4          MOVL    PCB$L_PHD(R4),R5        ; ADDRESS OF SYSTEM PROCESS HEADER
     56   0000'CF       D0  05E8          MOVL    W^SWP$GL_SHELIO,R6      ; GET I/O PAGE COUNT FOR SHELL
52    00000000'EF       9E  05ED          MOVAB   L^SWP$GL_SHELLBAS,R2    ; GET ADDRESS OF SHELL
            FA09'       30  05F4          BSBW    MMG$SVAPTECHK           ; GET ADDRESS OF PAGE TABLE ENTRY
          0048 8F       BB  05F7          PUSHR   #^M<R3,R6>              ; SAVE SVAPTE AND PAGE COUNT FOR LATER
               58       D4  05FB          CLRL    R8                      ; SET FLAG INDICATING NO I/O NEEDED
     57   0103 8F       3C  05FD          MOVZWL  #256+3,R7               ; SET FLAGS TO LOCK ONLY VALID OR
                            0602                                          ; TRANSITION PAGES AND CREATE OTHERS
                            0602                                          ; WITHOUT ZEROING THE PHYSICAL PAGE
          52     02     90  0602          MOVB    #WSL$C_SYSTEM,R2        ; SET PAGE TYPE IN LOW BITS
     0000'CF    00'     88  0605          BISB    S^#MMG$M_NOWAIT,W^MMG$GB_FREWFLGS ; PREVENT FREWLSE MWAIT
            F9F3'       30  060A  10$:    BSBW    MMG$IOLOCKPAG           ; LOCK THE PAGE INTO SYSTEM WORKING SET
            04 50       E8  060D          BLBS    R0,20$                  ; BRANCH IF SUCCEEDED
                            0610          BUG_CHECK INSNFREPAG,FATAL      ; INSUFFICIENT FREE PAGES
          58     50     88  0614  20$:    BISB    R0,R8                   ; SET FLAG (BIT 1) IF WE HAVE TO I/O IT
     52   0200 C2       9E  0617          MOVAB   512(R2),R2              ; BUMP VA TO NEXT PAGE
          53     04     C0  061C          ADDL    #4,R3                   ; BUMP PTE TO NEXT ENTRY
            E8 56       F5  061F          SOBGTR  R6,10$                  ; LOOP THROUGH THE PAGES
               18       BA  0622          POPR    #^M<R3,R4>              ; RECOVER SVAPTE AND PAGE COUNT
        16 58    01     E1  0624          BBC     #1,R8,40$               ; BRANCH IF ALL SHELL PAGES IN MEMORY
     0000'CF    00'     8A  0628          BICB    S^#MMG$M_NOWAIT,W^MMG$GB_FREWFLGS ; ALLOW FREWLSE MWAIT
               52       D4  062D          CLRL    R2                      ; SHELL IS PAGE FILE 0 AND VBN 0
             0456       30  062F          BSBW    SWPREAD                 ; PERFORM SHELL READ
            04 50       E8  0632          BLBS    R0,30$                  ; BRANCH IF NO ERROR IN READ
                            0635          BUG_CHECK INSWAPERR,FATAL       ; **** BUGCHECK ON I/O ERROR
     0000'CF    00'     88  0639  30$:    BISB    S^#MMG$M_NOWAIT,W^MMG$GB_FREWFLGS ; PREVENT FREWLSE MWAIT
                            063E  40$:    SETIPL  #IPL$_ASTDEL            ; ALLOW RESCHEDULE AND PAGEFAULTS WHILE
                            0641                                          ; COPYING SHELL BUT NOT COMPLETION ASTS
     56   0000'CF       D0  0641          MOVL    W^SWP$GL_SHELIO,R6      ; GET I/O PAGE COUNT FOR SHELL
57    00000000'EF       9E  0646          MOVAB   L^SWP$GL_SHELLBAS,R7    ; GET ADDRESS OF SHELL
     50    56    09     78  064D          ASHL    #9,R6,R0                ; GET BYTE COUNT
               51       D4  0651          CLRL    R1                      ; FORM DESTINATION VA
     61    67    50     28  0653          MOVC3   R0,(R7),(R1)            ; COPY THE SHELL TO LOCATION 0
                            0657          SETIPL  #IPL$_SYNCH             ; BACK TO BLOCKING IPL
     0000'CF    00'     8A  065A          BICB    S^#MMG$M_NOWAIT,W^MMG$GB_FREWFLGS ; ALLOW FREWLSE MWAIT
          51     56     7D  065F          MOVQ    R6,R1                   ; SET UP COUNT AND VA OF SHELL AGAIN
            F99B'       30  0662          BSBW    MMG$SVAPTECHK           ; GET ADDRESS OF PAGE TABLE ENTRY
            F998'       30  0665          BSBW    MMG$UNLOCK              ; DROP THE REFERENCE COUNTS
                            0668                                          ; CONTINUE PROCESS CREATION
```

```
                             0668
                             0668   ;-----------------------------------------------------------------
                             0668   ;
                             0668   ;       SET UP PROCESS IN BALANCE SET SLOT
                             0668   ;
                             0668   ;-----------------------------------------------------------------
                             0668
                             0668   SETUP:                                          ; SETUP INSWAP PROCESS
              03F6    30     0668           BSBW    OSINIT                          ; INIT REGISTERS
      54   0000'CF    D0     066B           MOVL    W^SWP$GL_INPCB,R4               ; GET PCB ADDRESS OF INSWAP PROCESS
             6C A4    D5     0670           TSTL    PCB$L_PHD(R4)                   ; CHECK FOR NEWLY ALLOCATED PHD
                06    19     0673           BLSS    10$                             ; BR IF NOT
   6C A4   0000'CF    C0     0675           ADDL    W^SWP$GL_BALBASE,PCB$L_PHD(R4)  ; AND SET ADDRESS IN PCB
             20 A4    D5     067B   10$:    TSTL    PCB$L_WSSWP(R4)                 ; CHECK FOR SHELL INSWAP
                09    14     067E           BGTR    NOTSHELL                        ; BR IF NOT
                             0680           INVALID                                 ; CLEAR TRANSLATION BUFFER
       00000000'9F    16     0683           JSB     @#SWP$SHELINIT                  ; CALL SHELL INITIALIZATION
                             0689                                                   ; WHICH RETURNS WITH A FULLY INITED PHD
                             0689   NOTSHELL:                                       ;
      58   0000'CF    3C     0689           MOVZWL  W^SWP$GW_IBALSETX,R8            ; AND BALANCE SET INDEX
   57 58   0000'CF    C5     068E           MULL3   W^SWP$GL_BSLOTSZ,R8,R7          ; COMPUTE OFFSET TO THIS SLOT
     57   0000'DF47   DE     0694           MOVAL   @W^SWP$GL_BALSPT[R7],R7         ; FORM BASE ADDRESS OF MAP FOR SLOT
             53   57  D0     069A           MOVL    R7,R3                           ; NOW POINT TO PROCESS HEADER
     1C 24 A4     12  E2     069D           BBSS    #PCB$V_PHDRES,PCB$L_STS(R4),5$  ; SKIP IF PROCESS HEADER STILL RESID
           0000'CF    D6     06A2           INCL    W^SWP$GL_HISWPCNT               ; COUNT SWAPS INCLUDING HEADER
              0346    30     06A6           BSBW    FILLPHD                         ; SET INTO SPT ENTRIES
                             06A9   ;
                             06A9   ;       FILLPHD RETURNS WITH R5 POINTING TO THE PROCESS HEADER POSITION
                             06A9   ;       WITHIN ITS P0 SPACE.
                             06A9   ;
        42 A5     58  B0     06A9           MOVW    R8,PHD$W_PHVINDEX(R5)           ; SET BALANCE SLOT INDEX
   00C8 C5     6C A4  C0     06AD           ADDL    PCB$L_PHD(R4),PHD$L_P0BR(R5)        ; RELOCATE P0 BASE REGISTER
   00D0 C5     6C A4  C0     06B3           ADDL    PCB$L_PHD(R4),PHD$L_P1BR(R5)        ; RELOCATE P1 BASE REGISTER
     00 36 A5     03  E2     06B9           BBSS    #PHD$V_NOACCVIO,PHD$W_FLAGS(R5),5$ ; INDICATE PHD INSWAP TO PAGER
 50    67   15    00  EF     06BE   5$:     EXTZV   #0,#PTE$S_PFN,(R7),R0           ; GET PHYSICAL ADDRESS OF PCB
       50   50    09  9C     06C3           ROTL    #9,R0,R0                        ; AND SET IN SOFTWARE PCB
     18 A4     78 A0  9E     06C7           MOVAB   PHD$L_PCB(R0),PCB$L_PHYPCB(R4)  ; ADD OFFSET TO HW PCB
                             06CC   ;
                             06CC   ;       NOW SET PAGES FROM WORKING SET LIST INTO PAGE TABLE ENTRIES
                             06CC   ;
        55     6C A4  D0     06CC           MOVL    PCB$L_PHD(R4),R5                ; GET PROCESS HEADER ADDRESS
                             06D0           INVALID                                 ; CLEAR TRANSLATION BUFFER TO SEE IT
                             06D3   ;
                             06D3   ;       A WINDOW IN P1 SPACE IS DOUBLE MAPPED TO ALL OF THE PROCESS
                             06D3   ;       HEADER EXCEPT FOR THE PAGE TABLES. THIS PERMITS REFERENCE TO
                             06D3   ;       MOST OF THE PROCESS HEADER WHILE RUNNING AT IPL LESS THAN THE
                             06D3   ;       SCHEDULER.  TO REFERENCE THE PROCESS HEADER IN SYSTEM SPACE
                             06D3   ;       A PROCESS(OTHER THAN THE SWAPPER) MUST RAISE TO IPL$_SYNCH.
                             06D3   ;
                             06D3
   52   00000000'EF   D0     06D3           MOVL    SWP$GL_PHDBASVA,R2              ; VIRTUAL ADDRESS OF PHD WINDOW
              F923'   30     06DA           BSBW    MMG$SVAPTECHK                   ; GET POINTER TO WINDOW PTE
      52   0000'CF    D0     06DD           MOVL    W^SGN$GL_PHDPAGCT,R2            ; SET COUNT OF PAGES FOR WINDOW
   51   F0000000 8F   D0     06E2           MOVL    #<PTE$C_URKW!PTE$M_VALID>,R1        ; SKELETON PTE
            50    87  D0     06E9   10$:    MOVL    (R7)+,R0                        ; GET SWAPPER PTE FOR PHD
                  04  19     06EC           BLSS    20$                             ; BR IF VALID PAGE
                  83  D4     06EE           CLRL    (R3)+                           ; NO, SET NO ACCESS
```

```
SWA
Sym

$$A
$$T
...
ACB
ACB
ACB
BAL
BDL
BDL
BUG
BUG
BUG
BUG
BUG
BUG
BUG
BUG
BUG
BUG
COP
CRE
CRE
CRE
CRE
CRE
CRE
DEL
DEL
DIR
DIR
DYN
DYN
DYN
DYN
EXE
EXE
EXE
EXE
EXE
EXE
EXE
EXE
EXE
EXE
EXE
FIL
FIL
FIL
FIL
FIL
GBL
GBL
GBL
GBL
GBL
```

```
                  08   11 06F0         BRB     30$                     ;
   51   15   00   50   F0 06F2 20$:    INSV    R0,#PTE$V_PFN,#PTE$S_PFN,R1     ; AND INSERT PFN INTO WINDOW PTE
             83   51   D0 06F7         MOVL    R1,(R3)+                ; STORE IN WINDOW AND ADVANCE TO NEX PTE
             EC   52   F5 06FA 30$:    SOBGTR  R2,10$                  ; MAP ENTIRE PHD WINDOW
                          06FD    ;
                          06FD    ;        THE REMAINING LIST OF PAGES READ BY THE SWAPPER ARE NOW PROCESSED
                          06FD    ;        ACCORDING TO THE CONTENT OF THE WORKING SET LIST IN THE HEADER OF
                          06FD    ;        THE INSWAP PROCESS.  THE DISPOSITION OF EACH INSWAP PAGE DEPENDS
                          06FD    ;        ON ITS TYPE AND WHETHER THE PAGE IS ALREADY PRESENT IN WHICH CASE
                          06FD    ;        THE NEW, REDUNDANT COPY IS DISCARDED.  SHARED PAGES READ FROM THE
                          06FD    ;        SWAP IMAGE WHICH ARE NOT ALREADY RESIDENT BECOME THE MASTER COPY
                          06FD    ;        AS WELL AS SATISFYING THE REQUIREMENT OF THE INSWAP PROCESS.
                          06FD    ;
        5B   01   1F   9C 06FD         ROTL    #PTE$V_VALID,#1,R11     ; FORM VALID MASK
         56   08 A5    3C 0701         MOVZWL  PHD$W_@SLIST(R5),R6     ; INDEX TO START OF PERM ENTRIES
         57   12 A5    3C 0705         MOVZWL  PHD$W_WSLAST(R5),R7     ; POINTER TO LAST WS ENTRY
                          0709         .ENABL  LSB                     ;
          52    6546   D0 0709 WSLOOP: MOVL    (R5)[R6],R2             ; GET A WORKING SET ENTRY
                  12   10 070D         BSBB    10$                     ; AND PROCESS IT
           F6 56  57   F3 070F         AOBLEQ  R7,R6,WSLOOP            ; SCAN ENTIRE WORKING SET LIST
        50   89   5A   CB 0713 5$:     BICL3   R10,(R9)+,R0            ; GET AND RELEASE EXCESS PAGES
                  05   13 0717         BEQL    7$                      ; BR IF NO MORE
                0358   30 0719         BSBW    RELPAGE                 ; RELEASE AN EXCESS PAGE
                  F5   11 071C         BRB     5$                      ; AND TRY FOR ANOTHER
                0261   31 071E 7$:     BRW     SETASTLVL               ; END OF WORKING SET LIST
                          0721         ASSUME  WSL$V_VALID EQ 0        ; FOR USE OF BLBS
             0A   52   E8 0721 10$:    BLBS    R2,20$                  ; CHECK FOR VALIDITY, BR IF VALID
                       05 0724 15$:    RSB                             ; GET NEXT WSL ENTRY IF NOT VALID
             52   55   C0 0725 17$:    ADDL    R5,R2                   ; REBIAS VA FOR WSL ENTRY
     6546    52   5B   C9 0728         BISL3   R11,R2,(R5)[R6]         ; AND SET SYSTEM BIT IN VA
                       05 072D 18$:    RSB                             ; NEXT WORKING SET LIST ENTRY
                          072E 20$:                                    ;
                  F5   19 072E         BLSS    17$                     ; SKIP PAGE TABLE ENTRIES
               F8CD'   30 0730         BSBW    MMG$SVAPTECHK           ; GET SVA OF PTE FOR PAGE
                          0733    ;
                          0733    ;        R0 - ALL BITS EXCEPT PFN FIELD ARE CLEAR
                          0733    ;        R2 - WS LIST ENTRY
                          0733    ;        R3 - SVA OF PTE
                          0733    ;        R4 - INSWAP PROCESS PCB
                          0733    ;        R5 - PHD ADDRESS FOR INSWAP PROCESS
                          0733    ;        R6 - WORKING SET INDEX
                          0733    ;        R7 - END INDEX TO WORKING SET
                          0733    ;        R8 - BALANCE SET SLOT INDEX
                          0733    ;        R9 - ADDRESS OF PHYSICAL PAGE POINTER IN SWP$AL_MAP
                          0733    ;        R10 - PTE$C_ERKW!PTE$M_VALID!PTE$M_MODIFY
                          0733    ;        R11 - CONSTANT PFN$M_VALID
                          0733    ;
        50   89   5A   CB 0733         BICL3   R10,(R9)+,R0            ; GET PFN FROM MAP
                  04   12 0737         BNEQ    30$                     ; GOT A GOOD PFN
                          0739         BUG_CHECK ZEROPAGE,FATAL        ; ZERO PAGE TABLE ENTRY FROM SWAP MAP
             5C   63   D0 073D 30$:    MOV[    (R3),AP                 ; GET CONTENT OF PTE
                  0B   18 0740         BGEQ    35$                     ; PTE VALID => PFN LOCK, NOT SWAPPED
                          0742    ;
                          0742    ; IF THE PAGE IS VALID, IT MUST HAVE BEEN LOCKED IN MEMORY AND WAS IGNORED AT
                          0742    ; OUTSWAP.  THE REDUNDANT PAGE ALLOCATED FOR THIS WORKING SET LIST ENTRY MUST
                          0742    ; BE RELEASED AFTER ALL WORKING SET LIST ENTRIES ARE PROCESSED.  NO OTHER
                          0742    ; ACTION IS NEEDED SINCE THE PTE FOR THE LOCKED PAGE IS ALREADY CORRECT.
```

```
                            0742  ; THE PFN THAT WOULD HAVE MATCHED THIS ENTRY REALLY BELONGS TO THE NEXT
                            0742  ; WSL WITHOUT PFNLOCK SET, SO THE MAP POINTER IN R9 MUST BE BACKED UP FOR
                            0742  ; ANOTHER TRY.
                            0742  ;
          03 52   04   E1   0742           BBC     #WSL$V_PFNLOCK,R2,32$    ; ERROR IF PAGE NOT LOCKED IN MEMORY
                  79   D5   0746           TSTL    -(R9)                    ; BACK UP IN SWAPPER MAP
                       05   0748           RSB                              ; AND CONTINUE WITH NEXT WSL
                            0749  32$:     BUG_CHECK       ICPAGELOC,FATAL ; INCONSISTENT PTE/WSL
          71 5C   1A   E1   074D  35$:     BBC     #PTE$V_TYP1,AP,NTYP1     ; BR IF NOT TYPE 1
      0000'DF40   53   D0   0751           MOVL    R3,@W^PFN$AL_PTE[R0]     ; NOTE LOCATION OF PTE
                            0757                   PFN_REFERENCE   -
                            0757           MOVW    <R6,@W^PFN$Ax_WSLX[R0]>,-        ; AND POSITION IN WORKING SET
                            0757                   LONG_OPCODE=MOVZWL,-
                            0757                   IMAGE=SYS_NONPAGED
   51    63    17   00   EF   075D           EXTZV   #PFN$V_BAK,#PFN$S_BAK,(R3),R1 ; GET BACKING ADR FROM PTE
          06 51   16   E0   0762           BBS     #PTE$V_TYP0,R1,40$       ; BR IF SECTION ADDRESS
 51   08   18   1F A5   F0   0766           INSV    PHD$B_PAGFIL(R5),#PFN$V_PGFLX,#PFN$S_PGFLX,R1
                            076C                                            ; SET PAGING FILE NUMBER
      0000'DF40   51   D0   076C  40$:     MOVL    R1,@W^PFN$AL_BAK[R0]     ; STORE BACKING ADDRESS
      0000'DF40   07   90   0772           MOVB    #PFN$C_ACTIVE,@W^PFN$AB_STATE[R0]; SET PAGE ACTIVE
                            0778  RECONNECT:                                ; RECONNECT TO PAGE
 51   63   867FFFFF 8F   CB   0778           BICL3   #^C<PTE$M_PROT!PTE$M_OWN>,(R3),R1       ; RETAIN PERMANENT BITS
             52   6546   DE   0780           MOVAL   (R5)[R6],R2              ; GET ADDRESS OF WORKING SET LIST ENTRY
          07 62   08   E5   0784           BBCC    #WSL$V_MODIFY,(R2),50$   ; CHECK FOR MODIFIED AND CLEAR
  0000'DF40   80 8F   88   0788           BISB    #PFN$M_MODIFY,@W^PFN$AB_STATE[R0] ; RECORD MODIFY STATE
             51   5B   C8   078F  50$:     BISL    R11,R1                   ; SET VALID BIT FOR PTE
       63    51   50   C9   0792           BISL3   R0,R1,(R3)               ; MERGE BITS WITH PFN  AND STORE IN PGTBL
   51    53    15   09   EF   0796           EXTZV   #VA$V_VPN,#VA$S_VPN,R3,R1        ; GET VPN OF PAGE TABLE
50   15   00   0000'DF41   F0   079B           INSV    @W^MMG$GL_SPTBASE[R1],#0,#PTE$S_PFN,R0  ; GET PT PFN
                            07A3                                            ; ASSUMES HIGH ORDER BITS OF R0 ARE CLEAR
                            07A3                   PFN_REFERENCE   -
                            07A3           TSTW    <@W^PFN$Ax_SHRCNT[R0]>,-         ; CHECK FOR FIRST ACTIVE PAGE
                            07A3                   LONG_OPCODE=TSTL,-
                            07A3                   IMAGE=SYS_NONPAGED
                  12   12   07A8           BNEQ    60$                      ; NO, JUST RAISE SHARE COUNT FOR PT
                            07AA                   PFN_REFERENCE   -
                            07AA           MOVZWL  <@W^PFN$Ax_WSLX[R0],R1>,-        ; GET INDEX TO WSL ENTRY FOR PAGE TA
                            07AA                   LONG_OPCODE=MOVL,-
                            07AA                   IMAGE=SYS_NONPAGED
           6541   20   C8   07B0           BISL    #WSL$M_WSLOCK,(R5)[R1]   ; AND MARK IT LOCKED IN WORKING SET
              70 A5   B6   07B4           INCW    PHD$W_PTCNTACT(R5)       ; COUNT ANOTHER ACTIVE PAGE TABLE
           0000'DF48   B6   07B7           INCW    @W^PHV$GL_REFCBAS[R8]    ; RAISE REFERENCE COUNT OF BALANCE SLOT
                            07BC  60$:                                      ;
                            07BC                   PFN_REFERENCE   -
                            07BC           INCW    <@W^PFN$Ax_SHRCNT[R0]>,-         ; INDICATE ANOTHER ACTIVE PAGE FOR P
                            07BC                   LONG_OPCODE=INCL,-
                            07BC                   IMAGE=SYS_NONPAGED
                       05   07C1           RSB                              ; RETURN TO GET NEXT WSL ENTRY
                            07C2           .DSABL  LSB                      ;
                            07C2  NTYP1:                                    ; GLOBAL OR TRANSITION
          2D 5C   16   E0   07C2           BBS     #PTE$V_TYP0,AP,11$       ; BR IF GLOBAL PAGE
   5D    5C    15   00   EF   07C6           EXTZV   #PTE$V_PFN,#PTE$S_PFN,AP,FP      ; GET OLD PFN IF ANY
                  29   13   07CB           BEQL    12$                      ; BR IF ZERO PAGE (BUG CHECK)
                            07CD  ;
                            07CD  ;        RELEASE PFN FOR PAGE ALREADY PRESENT
                            07CD  ;
                02A4   30   07CD           BSBW    RELPAGE                  ; RELEASE DUPLICATE PAGE
```

```
SWA
Sym

PCB
PCB
PCB
PCB
PCB
PCB
PCB
PCB
PCB
PER
PER
PFL
PFL
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PFN
PHD
PHD
PHD
PHD
PHD
PHD
PHD
```

```
                  50   5D   D0  07D0          MOVL    FP,R0                   ; GET SAVED PFN
  52  0000'DF40   03   00   EF  07D3          EXTZV   #PFN$V_LOC,#PFN$S_LOC,@W^PFN$AB_STATE[R0],R2
                                07DB          ASSUME  PFN$C_FREPAGLST EQ 0    ;
                                07DB          ASSUME  PFN$C_MFYPAGLST EQ 1    ;
                                07DB          ASSUME  PFN$C_BADPAGLST EQ 2    ;
                                07DB          ASSUME  PFN$C_RELPEND   EQ 3    ;
                                07DB          ASSUME  PFN$C_RDERR     EQ 4    ;
                                07DB          ASSUME  PFN$C_WRTINPROG EQ 5    ;
                                07DB          ASSUME  PFN$C_RDINPROG  EQ 6    ;
                                07DB          ASSUME  PFN$C_ACTIVE    EQ 7    ;
                                07DB          CASE    R2,<-                   ; DISPATCH ON PAGE LOCATION
                                07DB                  20$,-                   ; 0 => FREE PAGE LIST
                                07DB                  20$,-                   ; 1 => MODIFIED PAGE LIST
                                07DB                  60$,-                   ; 2 => BAD PAGE LIST, PAGE READ/WRITE ERR
                                07DB                  30$,-                   ; 3 => RELEASE PENDING
                                07DB                  10$,-                   ; 4 => PAGE READ ERROR
                                07DB                  30$,-                   ; 5 => WRITE IN PROGRESS
                                07DB                  40$,-                   ; 6 => READ IN PROGRESS
                                07DB                  30$>                    ; 7 => ACTIVE ( I/O NOT YET COMPLETE
                                07EF
                                07EF  10$:    BUG_CHECK ICPAGELOC,FATAL       ; INCONSISTENT PAGE LOCATION
                                07F3
                     0072   31  07F3  11$:    BRW     GLOBAL                  ; GLOBAL PAGE
                                07F6  12$:    BUG_CHECK ZEROPAGE,FATAL        ; ZERO PFN IN PTE
                                07FA
                       53   DD  07FA  20$:    PUSHL   R3                      ; SAVE SVAPTE
                     F801'  30  07FC          BSBW    MMG$REMPFN              ; UNLINK PFN FROM FREE OR MODIFY LIST
                       08   BA  07FF          POPR    #^M<R3>                 ; RESTORE SVAPTE
                  52 6546   DE  0801          MOVAL   (R5)[R6],R2             ; COMPUTE ADDRESS OF WSL ENTRY
                  63   5B   C8  0805  30$:    BISL    R11,(R3)                ; SET VALID BIT FOR PTE
                                0808          ASSUME  PFN$V_LOC EQ 0          ; TO USE BISB INSTEAD OF INSV
           0000'DF40   07   88  0808          BISB    #PFN$C_ACTIVE,@W^PFN$AB_STATE[R0]       ;
                            8A  080E  40$:    BICB    #<PFN$M_DELCON!-        ; CLEAR DELETE AND
                                080F                  PFN$M_MODIFY>,-         ; MODIFY
        0000'DF40   90 8F       080F                  @W^PFN$AB_STATE[R0]     ; FLAGS
                0000'DF40   B6  0815  45$:    INCW    @W^PFN$AW_REFCNT[R0]    ; RAISE REFERENCE COUNT
                0000'DF40   B4  081A          CLRW    @W^PFN$AW_SWPVBN[R0]    ; INDICATE NO ALTERNATE LOCATION
                  52 6546   DE  081F          MOVAL   (R5)[R6],R2             ; COMPUTE ADDRESS OF WSL ENTRY
               07 62   08   E5  0823          BBCC    #WSL$V_MODIFY,(R2),50$  ; CLEAR MODIFY BIT FOR WSL
        0000'DF40   80 8F   88  0827          BISB    #PFN$M_MODIFY,@W^PFN$AB_STATE[R0] ; RECORD PAGE AS MODIFIED
                                082E  50$:            PFN_REFERENCE   -
                                082E          MOVW    <R6,@W^PFN$Ax_WSLX[R0]>,-        ; SET WORKING SET LIST INDEX FOR PAG
                                082E                  LONG_OPCODE=MOVZWL,-
                                082E                  IMAGE=SYS_NONPAGED
                            05  0834          RSB                             ; AND RETURN FOR NEXT PAGE
                                0835  ;
                                0835  ; PAGE IS ON THE BAD PAGE LIST.  IT HAS THE FOLLOWING POSSIBLE STATES
                                0835  ;       1) BADPAG BIT SET IN PFN$AB_TYPE => BUG_CHECK
                                0835  ;       2) SWPVBN CLEAR => PAGE WRITE ERROR, CORRECT COPY OF MODIFY BIT
                                0835  ;           IS THE LOGICAL OR OF THE WSLE BIT AND THE PFN BIT
                                0835  ;       3) SWPVBN SET => PAGE READ ERROR, SET RDERR STATE.
                                0835  ;
        B3 0000'DF40   05   E0  0835  60$:    BBS     #PFN$V_BADPAG,@W^PFN$AB_TYPE[R0],10$ ; ERROR IF BADPAG
                       53   DD  083C          PUSHL   R3                      ; SAVE PTE ADDRESS
                     F7BF'  30  083E          BSBW    MMG$REMPFN              ; UNLINK PFN FROM THE BAD PAGE LIST
                       08   BA  0841          POPR    #^M<R3>                 ; RESTORE PTE ADDRESS
                  52 6546   DE  0843          MOVAL   (R5)[R6],R2             ; COMPUTE ADDRESS OF WSL ENTRY
```

```
          0000'DF40  B5  0847  2102          TSTW    @W^PFN$AW_SWPVBN[R0]      ; IF SWPVBN SET, THEN PAGE READ ERROR
                 0D  12  084C  2103          BNEQ    80$                       ; BRANCH IF PAGE READ ERROR
                         084E  2104 ;
                         084E  2105 ; PAGE WRITE ERROR
                         084E  2106 ;
                         084E  2107          ASSUME  PFN$V_MODIFY EQ 7
          0000'DF40  95  084E  2108          TSTB    @W^PFN$AB_STATE[R0]       ; IF PFN MODIFY BIT IS SET
                 B0  18  0853  2109          BGEQ    30$                       ;
           AC 62 08  E2  0855  2110          BBSS    #WSL$V_MODIFY,(R2),30$    ; THEN JAM THE WSL ENTRY MODIFY BIT
                 AA  11  0859  2111          BRB     30$                       ; AND CONNECT TO THE PAGE
                         085B  2112 ;
                         085B  2113 ; PAGE READ ERROR
                         085B  2114 ;
                 14  90  085B  2115 80$:     MOVB    #<PFN$M_DELCON ! PFN$C_RDERR>,- ; SET DELCON
          0000'DF40      085D  2116                  @W^PFN$AB_STATE[R0]       ; AND PAGE READ ERROR STATE
       62  0100 8F   AA  0861  2117          BICW    #<WSL$M_MODIFY>,(R2)      ; CLEAN UP WSLE
                 AD  11  0866  2118          BRB     45$                       ; AND LEAVE PTE IN TRANSITION STATE
                         0868  2119 ;
                         0868  2120 ;        INSWAP GLOBAL PAGE
                         0868  2121 ;
                         0868  2122 GLOBAL:                                    ; GLOBAL PAGE INSWAP
   51   5C   16  00  EF  0868  2123          EXTZV   #PTE$V_GPTX,#PTE$S_GPTX,AP,R1   ; GET GLOBAL PAGE TABLE INDEX
     51  0000'DF41   DE  086D  2124          MOVAL   @W^MMG$GL_GPTBASE[R1],R1        ; AND CONVERT TO ADDRESS OF GPTE
             52  61  D0  0873  2125          MOVL    (R1),R2                   ; PICK UP GLOBAL MASTER PTE
                 24  19  0876  2126          BLSS    10$                       ; BR IF VALID
        1D 52    16  E0  0878  2127          BBS     #PTE$V_TYPO,R2,5$         ; BR IF GLOBAL SECTION TYPE
   50   52   15  00  EF  087C  2128          EXTZV   #PTE$V_PFN,#PTE$S_PFN,R2,R0     ; GET PFN OF TRANSITION PAGE
                         0881  2129          ASSUME  PFN$C_FREPAGLST EQ 0
52  0000'DF40  03  00  EF  0881  2130          EXTZV   #PFN$V_LOC,#PFN$S_LOC,@W^PFN$AB_STATE[R0],R2      ; TEST FOR FREE PAGE
                 1B  13  0889  2131          BEQL    20$                       ; YES, REFAULT IT
                         088B  2132          ASSUME PFN$C_RDINPROG EQ <PFN$C_RDERR + 2>
                         088B  2133          CASE    R2, < -                   ; DISPATCH ON READ CASES:
                         088B  2134                  55$, -                    ;  READ ERROR
                         088B  2135                  4$, -                     ;  WRONG STATE
                         088B  2136                  60$ >, -                  ;  READ IN PROGRESS
                         088B  2137                  LIMIT=#PFN$C_RDERR
                         0895  2138 4$:      BUG_CHECK       ICPAGELOC, FATAL          ; WRONG STATE - CRASH SYSTEM
               0057  31  0899  2139 5$:      BRW     50$                       ; A BRANCH ASSIST
                         089C  2140
                         089C  2141 10$:                                       ; INSWAP WITH VALID GLOBAL PAGE
               01D5  30  089C  2142          BSBW    RELPAGE                   ; RELEASE REDUNDANT PAGE
   50   52   15  00  EF  089F  2143          EXTZV   #PTE$V_PFN,#PTE$S_PFN,R2,R0     ; GET PFN FROM MASTER
                 45  11  08A4  2144          BRB     40$                       ; AND GO SETUP SLAVE PTE
                         08A6  2145 20$:                                       ; GLOBAL ON FREE LIST
                 50  DD  08A6  2146          PUSHL   R0                        ; SAVE MASTER PFN
      50   FC A9 5A  CB  08A8  2147          BICL3   R10,-4(R9),R0             ; GET REDUNDANT PFN
               01C4  30  08AD  2148          BSBW    RELPAGE                   ; AND RELEASE IT (PRESERVING R1-R3)
                 01  BA  08B0  2149          POPR    #^M<R0>                   ; RESTORE MASTER PFN
             61  5B  C8  08B2  2150          BISL    R11,(R1)                  ; SET PAGE VALID
                 0A  BB  08B5  2151          PUSHR   #^M<R1,R3>                ; SAVE SVAGPTE, SVAPTE
              F746'  30  08B7  2152          BSBW    MMG$REMPFN                ; REMOVE PFN FROM FREELIST
                 0A  BA  08BA  2153          POPR    #^M<R1,R3>                ; RESTORE SVAGPTE, SVAPTE
0000'DF40  03   00   07  F0  08BC  2154          INSV    #PFN$C_ACTIVE,#PFN$V_LOC,#PFN$S_LOC,@W^PFN$AB_STATE[R0] ;
          0000'DF40  B6  08C4  2155          INCW    @W^PFN$AW_REFCNT[R0]      ; RAISE REFERENCE COUNT
                 20  11  08C9  2156          BRB     40$                       ;
   51   51   15  09  EF  08CB  2157 30$:     EXTZV   #VA$V_VPN,#VA$S_VPN,R1,R1       ; GET VPN OF PAGE TABLE
     51  0000'DF41   D0  08D0  2158          MOVL    @W^MMG$GL_SPTBASE[R1],R1        ; GET PAGE TABLE PTE
```

```
SWA
Sym

WSL
WSL
WSL
WSL

PSE
---
.
$AB
$$$
$$$
YF$
$AE
Z$I

Pha
---
Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
133
The
249
38

Mac
---
_$2
_$2
TOT

169

The

MAC
```

```
        51   51   15   00   EF  08D6        EXTZV   #PTE$V_PFN,#PTE$S_PFN,R1,R1        ; EXTRACT PFN
                                08DB                PFN_REFERENCE   -
                                08DB                TSTW    <@W^PFN$Ax_SHRCNT[R1]>,-   ; CHECK FOR FIRST REFERENCE TO PTABL
                                08DB                LONG_OPCODE=TSTL,-
                                08DB                IMAGE=SYS_NONPAGED
                       04   12  08E0        BNEQ    35$                       ; NO
                                08E2        BUG_CHECK GBLPAGSZRO,FATAL        ; GLOBAL PAGE SHARE COUNT ZERO
                                08E6  35$:          PFN_REFERENCE   -
                                08E6        INCW    <@W^PFN$Ax_SHRCNT[R1]>,-           ; RAISE GLOBAL PAGE TABLE SHARE COUN
                                08E6                LONG_OPCODE=INCL,-
                                08E6                IMAGE=SYS_NONPAGED
                                08EB  40$:          PFN_REFERENCE   -
                                08EB        INCW    <@W^PFN$Ax_SHRCNT[R0]>,-           ; RAISE SHARE COUNT FOR GLOBAL PAGE
                                08EB                LONG_OPCODE=INCL,-
                                08EB                IMAGE=SYS_NONPAGED
                     FE85   31  08F0        BRW     RECONNECT                 ; RECONNECT AND REFERENCE PAGE TABLE
  0000'DF40   52   17   00   EF  08F3  50$:  EXTZV   #PFN$V_BAK,#PFN$S_BAK,R2,@W^PFN$AL_BAK[R0] ; SAVE BACKING ADDR
         52   867FFFFF   8F   CA  08FB        BICL    #^C<PTE$M_PROT!PTE$M_OWN>,R2    ; SAVE PROTECTION AND OWNER FIELDS
                  52   5B   C8  0902        BISL    R11,R2                    ; SET PTE VALID
             61   52   50   C9  0905        BISL3   R0,R2,(R1)                ; AND STORE WITH PFN IN GPT
            0000'DF40   51   D0  0909        MOVL    R1,@W^PFN$AL_PTE[R0]      ; SET SVAGPTE IN PFN DATA BASE
            0000'DF40   07   90  090F        MOVB    #PFN$C_ACTIVE,@W^PFN$AB_STATE[R0] ; SET STATE TO ACTIVE
            0000'DF40   02   90  0915        MOVB    #PFN$C_GLOBAL,@W^PFN$AB_TYPE[R0]  ; AND TYPE TO GLOBAL
                       AE   11  091B        BRB     30$                       ; NOW GO SETUP SLAVE PTE
                                091D
                                091D  55$:  ; PAGE READ ERROR IN GPTE
                                091D        ;   THE PFN IN THE GPTE WILL BE DEALLOCATED
                                091D        ;   THE GPTE WILL BE ALTERED TO USE THE PFN FROM THE INSWAP IMAGE
                                091D        ;   THE DATA BASE WILL BE ADJUSTED AS APPROPRIATE
                                091D
          52   FC A9   5A   CB  091D        BICL3   R10, -4(R9), R2                           ; GET SWAP IMAGE PFN.
    0000'DF42   0000'DF40   D0  0922        MOVL    @W^PFN$AL_BAK[R0], @W^PFN$AL_BAK[R2]      ; COPY BACKING STORE.
            0000'DF42   07   90  092B        MOVB    #PFN$C_ACTIVE, @W^PFN$AB_STATE[R2]        ; SET STATE TO ACTIVE.
            0000'DF42   02   90  0931        MOVB    #PFN$C_GLOBAL, @W^PFN$AB_TYPE[R2]         ; SET TYPE TO GLOBAL.
                                0937        PFN_REFERENCE =                                   ; COPY SHARE COUNT.
                                0937        MOVQ    < @W^PFN$Ax_SHRCNT[R0], @W^PFN$Ax_SHRCNT[R2] >, -
                                0937                LONG_OPCODE = MOVL, -
                                0937                IMAGE=SYS_NONPAGED
                0000'DF40   D4  0940        CLRL    @W^PFN$AL_PTE[R0]                         ; SETUP FOR AND RELEASE
                     012C   30  0945        BSBW    RELPAGE                                   ; READ ERROR PFN.
                                0948                                                          ; NOBODY CAN USE IT.
                  50   52   D0  0948        MOVL    R2, R0                                    ; SETUP NEW MASTER PFN.
        61   15   00   50   F0  094B        INSV    R0, #PTE$V_PFN, #PTE$S_PFN, (R1)          ; PLANT PFN IN GPTE.
            0000'DF40   51   D0  0950        MOVL    R1, @W^PFN$AL_PTE[R0]                     ; PLANT PTE IN DATABASE.
                  61   5B   C8  0956        BISL    R11, (R1)                                 ; MAKE PTE VALID.
                       90   11  0959        BRB     40$                                       ; JOIN COMMON CODE.
                                095B
            0000'DF40   10   88  095B  60$:  BISB    #PFN$M_COLLISION,@W^PFN$AB_TYPE[R0]       ; FLAG COLLISION FOR PAGEREA
                       3C   BB  0961        PUSHR   #^M<R2,R3,R4,R5>          ; SAVE REGS OVER WAIT
             54   0000'CF   D0  0963        MOVL    W^SCH$GL_CURPCB,R4        ; AND SET PCB ADDRESS
                    F695'   30  0968        BSBW    SCH$NEWLVL                ; SET ASTLVL CORRECTLY
             52   0000'CF   7E  096B        MOVAQ   W^SCH$GQ_COLPGWQ,R2       ; GET ADDRESS OF WAIT QUEUE
                       00   DD  0970        PUSHL   #0                        ; NULL KERNEL MODE PSL
                    F68B'   30  0972        BSBW    SCH$WAITK                 ; WAIT WITH NO CALL FRAME
                                0975        SETIPL  #IPL$_SYNCH               ; BLOCK SYSTEM EVENTS
                       3C   BA  0978        POPR    #^M<R2,R3,R4,R5>          ; RESTORE REGS
          50   FC A9   5A   CB  097A        BICL3   R10,-4(R9),R0             ; RESTORE CURRENT PFN
```

```
                  FEE6   31  097F            BRW     GLOBAL                          ; AND ATTEMPT TO REASSOCIATE PAGE
                             0982
                             0982
                             0982     ;
                             0982     ;      SET PROPER AST LEVEL
                             0982     ;
                             0982     SETASTLVL:                                     ;
           53    10 A4   DE  0982            MOVAL   PCB$L_ASTQFL(R4),R3             ; GET POINTER TO HEAD OF AST QUEUE
                 52  63  D0  0986            MOVL    (R3),R2                         ; GET POINTER TO FIRST AST CONTROL BLOCK
                 52  53  D1  0989            CMPL    R3,R2                           ; IS LIST EMPTY?
                     1C  13  098C            BEQL    20$                             ; YES, DONE
                     50  D4  098E            CLRL    R0                              ; ASSUME KERNEL MODE
           53    0B A2   90  0990            MOVB    ACB$B_RMOD(R2),R3               ; GET ACTUAL MODE
                     0F  19  0994            BLSS    10$                             ; BR IF SPECIAL KERNEL AST
      50   53    02  00  EF  0996            EXTZV   #ACB$V_MODE,#ACB$S_MODE,R3,R0 ; GET ACCESS MODE
    51   0D A4    0C A4  8B  099B            BICB3   PCB$B_ASTACT(R4),PCB$B_ASTEN(R4),R1       ; CHECK FOR DELIVERABILITY
           05 51     50  E1  09A1            BBC     R0,R1,20$                       ; BR IF NOT PRESENTLY DELIVERABLE
         00CF C5     50  90  09A5   10$:     MOVB    R0,PHD$B_ASTLVL(R5)             ; SET AST LEVEL FOR PROCESS
           24 A4     09  C8  09AA   20$:     BISL    #<<1@PCB$V_RES>!<1@PCB$V_INQUAN>>,PCB$L_STS(R4) ; MARK PROCESS RESID
   3C A5   00000000'EF   B0  09AE            MOVW    SCH$GW_QUAN,PHD$W_QUANT(R5)      ; AND GIVE NEW QUANTUM
           50    0B A4   9A  09B6            MOVZBL  PCB$B_PRI(R4),R0                ; GET CURRENT PRIORITY OF PROCESS
           51    1F  50  83  09BA            SUBB3   R0,#31,R1                       ; COMPUTE EXTERNAL PRIORITY FOR COMPARE
         0000'CF     51  91  09BE            CMPB    R1,W^SYS$GB_DEFPRI              ; IS THIS A "CRUNCHER"?
                     0F  14  09C3            BGTR    30$                             ; NO, CONTINUE
      51   00000000'EF   D0  09C5            MOVL    EXE$GQ_SYSTIME,R1               ; GET CURRENT TIME IN APPROX. 10MS UNITS
 0000'CF   51    0000'CF C1  09CC            ADDL3   W^SCH$GL_SWPRATE,R1,W^SWP$GL_SWTIME ; SET NEW CRUNCHER INTERVAL
                  F629'  30  09D4   30$:     BSBW    SCH$CHSEP                       ; CHANGE TO RESIDENT COMPUTE
                             09D7   SWAPRETRY:                                       ; RETRY SWAP SCHEDULING
           54    0000'CF D0  09D7            MOVL    W^SCH$GL_CURPCB,R4              ; GET PCB ADDRESS
        00 24 A4     0C  E6  09DC            BBSSI   #PCB$V_WAKEPEN,PCB$L_STS(R4),20$          ; SET TO CANCEL HIBER
                             09E1   20$:
                             09E1            .DSABL  LSB
                             09E1   SWAPEXIT:                                        ; EXIT SWAPPER
      00 0000'CF     00' E5  09E1            BBCC    S^#SCH$V_SIP,W^SCH$GB_SIP,10$   ; CLEAR SWAP IN PROGRESS
                             09E7   10$:
                             09E7   SWAPEXITA:                                       ; ALTERNATE EXIT, LEAVING SIP SET
               3FC0 8F   BA  09E7            POPR    #^M<R6,R7,R8,R9,R10,R11,AP,FP>  ; RESTORE REGISTERS
                             09EB            SETIPL  #0                              ; DROP IPL
                         05  09EE            RSB                                     ;
```

```
                             09EF          .SBTTL  FILLPHD - FILL SPT ENTRIES TO MAP PHD
                             09EF
                             09EF  ;++
                             09EF  ; FUNCTIONAL DESCRIPTION:
                             09EF  ;       FILLPHD SETS THE PTE ENTRIES FOR THE PROCESS HEADER INTO THE
                             09EF  ;       SPT.
                             09EF  ;
                             09EF  ; CALLING SEQUENCE:
                             09EF  ;       BSB/JSB FILLPHD
                             09EF  ;
                             09EF  ; INPUT PARAMETERS:
                             09EF  ;       R3  -  POINTER TO FIRST SPT ENTRY FOR PHD
                             09EF  ;       R9  -  ADDRESS OF SWAPPER MAP ENTRY TO BE MOVED TO SPT
                             09EF  ;       R10 -  PTE$C_ERKW!PTE$M_VALID!PTE$M_MODIFY
                             09EF  ;
                             09EF  ; OUTPUT PARAMETERS:
                             09EF  ;       R5 - ZERO
                             09EF  ;       R6 - DESTROYED
                             09EF  ;       R9 - UPDATED
                             09EF  ;       R11 - DESTROYED
                             09EF  ;       AP - DESTROYED
                             09EF  ;       FP - DESTROYED
                             09EF  ;--
                             09EF
                             09EF  FILLPHD:                                  ;
                             09EF
                    55   D4  09EF          CLRL    R5                        ; SET PHD ADDRESS TO SWAPPER P0 SPACE
                             09F1          INVALID                           ; TO SEE CORRECT PROCESS HEADER IN SWAPPER P
                    5B   D4  09F4          CLRL    R11                       ; INIT HEADER PAGE INDEX
          56   00D8 C5   3C  09F6          MOVZWL  PHD$W_EMPTPG(R5),R6       ; GET COUNT OF EMPTY PAGES
          56   56   09   78  09FB          ASHL    #9,R6,R6                  ; CONVERT TO BYTE OFFSET
            5C   48 A5   D0  09FF          MOVL    PHD$L_WSLX(R5),AP         ; FORM BASE ADDRESS FOR WSLX
             5C   654C   DE  0A03          MOVAL   (R5)[AP],AP               ; SAVE VECTOR FOR PHD
            5D   44 A5   D0  0A07          MOVL    PHD$L_BAK(R5),FP          ; FORM BASE ADDRESS FOR BACKING STORE ADDRES
             5D   654D   DE  0A0B          MOVAL   (R5)[FP],FP               ; VECTOR
               5C   56   C2  0A0F          SUBL    R6,AP                     ; ACCOUNT FOR EMPTY PAGES
               5D   56   C2  0A12          SUBL    R6,FP                     ; BY SUBTRACTING THEIR SPACE
          56   0000'CF   D0  0A15          MOVL    W^SWP$GL_BSLOTSZ,R6       ; SET ITERATION COUNT FOR ENTIRE HEADER
               83   8D   D0  0A1A  10$:    MOVL    (FP)+,(R3)+               ; SET BACKUP FORM OF PTE IN SPT SLOT
                    33   18  0A1D          BGEQ    30$                       ; DONE IF NOT VALID
          50   89   5A   CB  0A1F          BICL3   R10,(R9)+,R0              ; GET PAGE FROM SWAPPER MAP
        0000'DF40   73   DE  0A23          MOVAL   -(R3),@W^PFN$AL_PTE[R0] ; SET PTE BACK POINTER
     51   63   17   00   EF  0A29          EXTZV   #PFN$V_BAK,#PFN$S_BAK,(R3),R1    ; ISOLATE BACKING STORE ADDRESS
  51   08   18   1F A5   F0  0A2E          INSV    PHD$B_PAGFIL(R5),#PFN$V_PGFLX,#PFN$S_PGFLX,R1    ; ADD FILE NUMBER
        0000'DF40   51   D0  0A34          MOVL    R1,@W^PFN$AL_BAK[R0]      ; SAVE IN PFN DATA BASE
                             0A3A                  PFN_REFERENCE   -
                             0A3A          MOVW    <(AP)[R11],@W^PFN$Ax_WSLX[R0]>,-          ; SAVE WORKING SET LIST INDE
                             0A3A                  LONG_OPCODE=MOVZWL,-
                             0A3A                  IMAGE=SYS_NONPAGED
          83   5A   50   C9  0A41          BISL3   R0,R10,(R3)+              ; SET VALID PTE FOR PAGE
     0000'DF40   87 8F   90  0A45          MOVB    #<PFN$C_ACTIVE!PFN$M_MODIFY>,@W^PFN$AB_STATE[R0] ; MARK PAGE ACTIVE
        0000'DF40   04   90  0A4C          MOVB    #PFN$C_PPGTBL,@W^PFN$AB_TYPE[R0]          ; STORE TYPE IN PFN DATA BAS
            C4 5B   56   F2  0A52  30$:    AOBLSS  R6,R11,10$                ; FILL ENTIRE PROCESS HEADER
                         05  0A56          RSB
```

```
                      0A57  2313                .SBTTL  RELINIT - INITIALIZE REGISTERS FOR PAGE RELEASE LOOP
                      0A57  2314
                      0A57  2315 ;++
                      0A57  2316 ; FUNCTIONAL DESCRIPTION:
                      0A57  2317 ;       RELINIT SETS UP REGISTERS FOR THE PAGE RELEASE LOOPS FOLLOWING
                      0A57  2318 ;       OUTSWAP I/O OPERATIONS.
                      0A57  2319 ;
                      0A57  2320 ; CALLING SEQUENCE:
                      0A57  2321 ;       BSB/JSB RELINIT
                      0A57  2322 ;
                      0A57  2323 ; INPUT PARAMETERS:
                      0A57  2324 ;       NONE
                      0A57  2325 ;
                      0A57  2326 ; OUTPUT PARAMETERS:
                      0A57  2327 ;       R0  -  0
                      0A57  2328 ;       R4  -  OUT SWAP PCB ADDRESS (OSWPPCB)
                      0A57  2329 ;       R7  -  PAGE COUNT TO RELEASE
                      0A57  2330 ;       R9  -  BASE ADDRESS FOR SWAPPER MAP (SWP$AL_MAP)
                      0A57  2331 ;       R10 -  PTE$C_ERKW!PTE$M_VALID!PTE$M_MODIFY
                      0A57  2332 ;       R11 -  BASE ADDRESS FOR SWAPPER MAP (SWP$AL_MAP)
                      0A57  2333 ;
                      0A57  2334 ;--
                      0A57  2335
                      0A57  2336 RELINIT:                                      ; RELEASE LOOP INITIALIZATION
54    0014'CF    D0   0A57  2337         MOVL    W^OSWPPCB,R4                  ; GET PCB ADDRESS OF OUT SWAP PROCESS
57    0012'CF    3C   0A5C  2338         MOVZWL  W^OSWPPGS,R7                  ; AND PAGE COUNT FOR RELEASE LOOP
                      0A61  2339 ;       BRB     OSINIT                        ; FALL INTO OSINIT
```

```
                        0A61          .SBTTL  OSINIT - OUTSWAP SCAN REGISTER INITIALIZATION
                        0A61
                        0A61    ;++
                        0A61    ; FUNCTIONAL DESCRIPTION:
                        0A61    ;       OSINIT SETS UP REGISTERS FOR PAGE TABLE SCANS REQUIRED DURING
                        0A61    ;       OUTSWAPPING.
                        0A61    ;
                        0A61    ; INPUT PARAMETERS:
                        0A61    ;       NONE
                        0A61    ;
                        0A61    ; OUTPUT PARAMETERS:
                        0A61    ;       R9 - BASE ADDRESS OF SWAPPER MAP (SWP$AL_MAP)
                        0A61    ;       R10 - PTE$C_ERKW!PTE$M_VALID
                        0A61    ;       R11 - BASE ADDRESS OF SWAPPER MAP (SWP$AL_MAP)
                        0A61    ;
                        0A61    ;--
                        0A61
                        0A61   OSINIT:                                          ;
      59   0000'DF   DE 0A61          MOVAL   @W^SWP$GL_MAP,R9                 ; SET BASE OF SWAPPER MAP
           5B   59   D0 0A66          MOVL    R9,R11                           ; AND MAKE REFERENCE COPY
 5A  B4000000 8F     D0 0A69          MOVL    #<PTE$C_ERKW!PTE$M_VALID!PTE$M_MODIFY>,R10 ; MASK TO VALIDATE SWAP P
                05      0A70          RSB                                      ; RETURN
```

```
                  0A71           .SBTTL  RELPAGE - RELEASE DUPLICATE PAGE
                  0A71
                  0A71      ;++
                  0A71      ; FUNCTIONAL DESCRIPTION:
                  0A71      ;        RELPAGE RELEASES A PHYSICAL PAGE WHICH DUPLICATES A PAGE ALREADY
                  0A71      ;        PRESENT FOR THE PROCESS.  THIS SITUATION CAN ARISE DUE TO A PARTIAL
                  0A71      ;        INSWAP OR A GLOBAL PAGE WHICH IS ALREADY PRESENT.
                  0A71      ;
                  0A71      ; CALLING SEQUENCE:
                  0A71      ;        BSB/JSB RELPAGE
                  0A71      ;
                  0A71      ; INPUT PARAMETERS:
                  0A71      ;        R0 - PFN TO RELEASE
                  0A71      ;        R3 - SVA OF PTE (RELDELPAGE ONLY)
                  0A71      ;
                  0A71      ; OUTPUT PARAMETERS:
                  0A71      ;        R1 - PRESERVED (RELPAGE ONLY)
                  0A71      ;        R2 - PRESERVED (RELPAGE ONLY)
                  0A71      ;        R3 - PRESERVED (RELPAGE ONLY)
                  0A71      ;
                  0A71      ;--
                  0A71
                  0A71      RELDELPAGE:                                   ; RELEASE PAGE THROUGH DELCONPFN
     F58C'   30   0A71              BSBW    MMG$DELCONPFN                ; DELETE PAGE CONTENT AND INIT PFN DATA
                  0A74      RELPAGE:                                      ; RELEASE PAGE
       0E    BB   0A74              PUSHR   #^M<R1,R2,R3>                ; PRESERVE REGISTERS
0000'DF40    94   0A76              CLRB    @W^PFN$AB_STATE[R0]          ; INIT PFN DATA FOR RELEASE
0000'DF40    B4   0A7B              CLRW    @W^PFN$AW_REFCNT[R0]         ; ZERO REFERENCE COUNT
                  0A80              ASSUME  PFN$C_FREPAGLST EQ 0         ;
       52    D4   0A80              CLRL    R2                           ; INDICATE FREELIST
     F57B'   30   0A82              BSBW    MMG$INSPFNH                  ; RELEASE PFN TO HEAD OF FREE LIST
       0E    BA   0A85              POPR    #^M<R1,R2,R3>                ; RESTORE REGISTERS
             05   0A87              RSB                                  ; AND RETURN TO CALLER
```

```
                         0A88          .SBTTL  SWPREAD/SWPWRITE - SWAPPER I/O ROUTINES
                         0A88
                         0A88    ;++
                         0A88    ; FUNCTIONAL DESCRIPTION:
                         0A88    ;         SWPREAD AND SWPWRITE PERFORM THE DETAIL WORK REQUIRED TO READ
                         0A88    ;         OR WRITE A SET OF CONTIGUOUS PAGES IN A WORKING SET SWAP IMAGE.
                         0A88    ;         THE CALL TO EITHER SWPREAD OR SWPWRITE IS ACTUALLY A CO-ROUTINE
                         0A88    ;         CALL WHICH RETURNS ONLY AFTER ALL SEGMENTS OF THE I/O OPERATION
                         0A88    ;         HAVE BEEN PERFORMED.  THIS RETURN IS EFFECTED BY A SPECIAL KERNEL
                         0A88    ;         AST.
                         0A88    ;
                         0A88    ; CALLING SEQUENCE:
                         0A88    ;         BSB/JSB SWPREAD/SWPWRITE
                         0A88    ;
                         0A88    ; INPUT PARAMETERS:
                         0A88    ;         R0 - SWAP FILE INDEX
                         0A88    ;         R2 - WSSWP FORM DISK ADDRESS
                         0A88    ;         R3 - SYSTEM VIRTUAL ADDRESS OF PTE
                         0A88    ;         R4 - PAGE COUNT
                         0A88    ;
                         0A88    ;         00(SP) - RETURN ADDRESS AFTER I/O COMPLETION
                         0A88    ;         04(SP) - SAVED R6
                         0A88    ;         08(SP) - SAVED R7
                         0A88    ;         12(SP) - SAVED R8
                         0A88    ;         16(SP) - SAVED R9
                         0A88    ;         20(SP) - SAVED R10
                         0A88    ;         24(SP) - SAVED R11
                         0A88    ;         28(SP) - SAVED AP
                         0A88    ;         32(SP) - SAVED FP
                         0A88    ;         36(SP) - SAVED IPL
                         0A88    ;         40(SP) - RETURN TO PREVIOUS THREAD
                         0A88    ;
                         0A88    ; IMPLICIT INPUTS:
                         0A88    ;         PAGE FILE TABLE ENTRY (PFL) SELECTED BY WSSWP INPUT
                         0A88    ;
                         0A88    ; OUTPUT PARAMETERS:
                         0A88    ;         R0 - COMPLETION STATUS OF I/O OPERATION
                         0A88    ;
                         0A88    ;--
                         0A88
                         0A88            .ENABL  LSB
                         0A88    SWPREAD:                                        ; SWAP READ INITIATION
         0000'CF    9F   0A88            PUSHAB  W^EXE$BLDPKTSWPR                ; SET ADDRESS OF BUILD PACKET ROUTINE
              04    11   0A8C            BRB     10$                             ;
                         0A8E    SWPWRITE:                                       ; SWAP WRITE INITIATION
         0000'CF    9F   0A8E            PUSHAB  W^EXE$BLDPKTSWPW                ; SET ADDRESS OF BUILD PACKET ROUTINE
      51 0000'CF    9E   0A92    10$:    MOVAB   W^IOROUTINE,R1                  ; ADDRESS OF I/O DATA
         81    8E   7D   0A97            MOVQ    (SP)+,(R1)+                     ; SAVE I/O END ACTION ADDRESS
         3FC0  8F   BA   0A9A            POPR    #^M<R6,R7,R8,R9,R10,R11,AP,FP>; RESTORE REGISTERS OTHER THAN STANDAR
   50  52  08  18   EF   0A9E    15$:    EXTZV   #24,#8,R2,R0                    ; GET SWAP FILE INDEX
     50  0000'DF40  D0   0AA3            MOVL    @W^MMG$GL_PAGSWPVC[R0],R0; GET BASE ADDRESS OF PAGE FILE TABLE
       55  7F  8F   9A   0AA9            MOVZBL  #127,R5                         ; SET I/O SIZE
           55  54   D1   0AAD            CMPL    R4,R5                           ; COMPARE REMAINING PGCNT WITH MAX TRANSFER
               03   14   0AB0            BGTR    20$                             ; USE MAXIMUM TRANSFER
           55  54   D0   0AB2            MOVL    R4,R5                           ; SET TRANSFER TO REMAINING PAGES
       81  52  55   C1   0AB5    20$:    ADDL3   R5,R2,(R1)+                     ; SAVE UPDATED DISK ADDRESS
         81  6345   DE   0AB9            MOVAL   (R3)[R5],(R1)+                  ; AND UPDATED SAVPTE
```

```
                                0ABD  2458          SETIPL  #0                          ; DROP IPL
           61   54    55  A3    0AC0  2459          SUBW3   R5,R4,(R1)                  ; SAVE REMAINING PAGE COUNT
                      53  DD    0AC4  2460          PUSHL   R3                          ; SAVE SVAPTE
                   0C A0  DD    0AC6  2461          PUSHL   PFL$L_WINDOW(R0)            ; GET WINDOW ADDRESS
                7E 55  09  9C   0AC9  2462          ROTL    #9,R5,-(SP)                 ; CONVERT PAGES TO BYTE COUNT
        7E      52 18  00  EF   0ACD  2463          EXTZV   #0,#24,R2,-(SP)             ; AND ISOLATE BLOCK NUMBER
                6E  10 A0  C0   0AD2  2464          ADDL    PFL$L_VBN(R0),(SP)          ; ADD BASE VBN
                54 0000'CF DO   0AD6  2465          MOVL    W^SCH$GL_CURPCB,R4          ; SET PCB ADDRESS
                55 0000'DF 0F   0ADB  2466          REMQUE  @W^IOC$GL_IRPFL,R5          ; GET A PACKET IF POSSIBLE
                       06  1C   0AE0  2467          BVC     30$                         ; BR IF ONE AVAILABLE
                    F51B'  30   0AE2  2468          BSBW    EXE$ALLOCIRP                ; ALLOCATE ONE THE LONG WAY
                55     52  D0   0AE5  2469          MOVL    R2,R5                       ; SET PACKET ADDRESS IN PROPER REGISTER
             14 A5  FB'AF  9E   0AE8  2470 30$:     MOVAB   B^IODONE,IRP$L_ASTPRM(R5); SET ADDRESS FOR COMPLETION
        23 A5   1F 0000'CF 83   0AED  2471          SUBB3   W^SWP$GB_PRIO,#31,IRP$B_PRI(R5) ; SET PRIORITY FOR TRANSFER
                       0F  BA   0AF4  2472          POPR    #^M<R0,RT,R2,R3>            ; RESTORE VBN,BYTECNT,WINDOW,SVAPTE
                  0000'DF  16   0AF6  2473          JSB     @W^IOROUTINE                ; CALL READ OR WRITE ROUTINE
                           05   0AFA  2474          RSB                                 ; AND RETURN TO ORIGINAL CALLER
                                0AFB  2475
                                0AFB  2476 IODONE:                                      ; CONTINUATION CALLED AS KERNEL AST
                    38 A5  DD   0AFB  2477          PUSHL   IRP$L_MEDIA(R5)             ; SAVE COMPLETION STATUS
                50     55  D0   0AFE  2478          MOVL    R5,R0                       ; SET PACKET ADDRESS FOR RELEASE
                    F4FC'  30   0B01  2479          BSBW    EXE$DEANONPAGED             ; AND RELEASE IT
                50     8E  D0   0B04  2480          MOVL    (SP)+,R0                    ; RESTORE STATUS
                                0B07  2481          SETIPL  #IPL$_SYNCH                 ; BLOCK SYSTEM EVENTS
                    11 50  E9   0B0A  2482          BLBC    R0,60$                      ; EXIT IF ERROR
           51     0008'CF  9E   0B0D  2483          MOVAB   W^RWSSWP,R1                 ; GET ADDRESS OF REMAINING TRANSFER PARAMS
                52     61  7D   0B12  2484          MOVQ    (R1),R2                     ; RESTORE WSSWP,SVAPTE TO R2,R3
              54    08'A1  3C   0B15  2485          MOVZWL  B^<RPGCNT-RWSSWP>(R1),R4          ; AND REMAINING PAGE COUNT
                       03  13   0B19  2486          BEQL    60$                         ; DONE IF NO MORE PAGES REMAIN
                     FF80  31   0B1B  2487          BRW     15$                         ; CONTINUE IF MORE PAGES REMAIN
                  3FC0 8F  BB   0B1E  2488 60$:     PUSHR   #^M<R6,R7,R8,R9,R10,R11,AP,FP>; SAVE NON-STANDARD REGISTERS
                  0004'DF  17   0B22  2489          JMP     @W^IOEA                     ; AND CONTINUE SWAP
                                0B26  2490
                                0B26  2491          .DSABL  LSB                         ;
                                0B26  2492          .END
```

```
$$ARGS                = 00000005                   GBLWRTVALID             000004CF R       05
$$T1                  = 00000018                   GLOBAL                  00000868 R       05
...PFN                = 00000A3A R       05        IMGDESC                 00000000 R       04
ACB$B_RMOD            = 0000000B                   INSWAP                  00000535 R       05
ACB$S_MODE            = 00000002                   IOC$GL_IRPFL            ********      X  05
ACB$V_MODE            = 00000000                   IODONE                  00000AFB R       05
BALANCE                 00000030 R       05        IOEA                    00000004 R       02
BDL$L_SYSDLOG           ********      X  04        IOROUTINE               00000000 R       02
BDL$S_CRELNM_ITMLST     ********      X  04        IPL$_ASTDEL           = 00000002
BUG$_APTREFHIGH         ********      X  05        IPL$_SYNCH            = 00000008
BUG$_APTWRTERR          ********      X  05        IRP$B_PRI             = 00000023
BUG$_GBLPAGSZRO         ********      X  05        IRP$L_ASTPRM          = 00000014
BUG$_ICPAGELOC          ********      X  05        IRP$L_MEDIA           = 00000038
BUG$_INSNFREPAG         ********      X  05        KERNEL_MODE             000001CD R       04
BUG$_INSSWPFIL          ********      X  05        LNM$AL_HASHTBL          ********      X  04
BUG$_INSWAPERR          ********      X  05        LNM$GL_HTBLSIZS         ********      X  04
BUG$_IVWSETLIST         ********      X  05        LNM$HASH                ********      X  04
BUG$_OUTSWPERR          ********      X  05        LNM$INSLOGTAB           ********      X  04
BUG$_QUEUEEMPTY         ********      X  05        LNM$M_NO_ALIAS        = 00000001
BUG$_ZEROPAGE           ********      X  05        LNM$M_TERMINAL        = 00000200
COPYSHELL               000005DF R       05        LNM$SYSTEM_DIRECTORY    00000000 RG      03
CRELNM$_ACMODE        = 00000010                   LNM$_ATTRIBUTES       = 00000003
CRELNM$_ATTR          = 00000004                   LNM$_STRING           = 00000002
CRELNM$_ITMLST        = 00000014                   LNMB$B_ACMODE         = 0000000B
CRELNM$_LOGNAM        = 0000000C                   LNMB$B_FLAGS          = 00000010
CRELNM$_NARGS         = 00000005                   LNMB$B_TYPE           = 0000000A
CRELNM$_TABNAM        = 00000008                   LNMB$L_BLINK          = 00000004
DELCON                  0000051F R       05        LNMB$L_FLINK          = 00000000
DELPHD                  00000404 R       05        LNMB$L_TABLE          = 0000000C
DIRECTORIES_ARG         0000032D R       04        LNMB$M_NODELETE       = 00000010
DIRECTORIES_LIST        000001DD R       04        LNMB$M_NO_ALIAS       = 00000001
DYN$C_LNM             = 00000040                   LNMB$M_TABLE          = 00000008
DYN$C_ORB             = 00000049                   LNMB$T_NAME           = 00000011
DYN$C_PCB             = 0000000C                   LNMB$W_SIZE           = 00000008
DYN$C_RSHT            = 00000038                   LNMHSH$B_TYPE         = 0000000A
EXE$ALLOCIRP            ********      X  05        LNMHSH$C_BUCKET       = 0000000C
EXE$ALOPAGED            ********      X  04        LNMHSH$K_BUCKET       = 0000000C
EXE$BLDPKTSWPR          ********      X  05        LNMHSH$L_MASK         = 00000000
EXE$BLDPKTSWPW          ********      X  05        LNMHSH$W_SIZE         = 00000008
EXE$DEANONPAGED         ********      X  05        LNMTH$B_FLAGS         = 00000000
EXE$DEANONPGDSIZ        ********      X  04        LNMTH$K_LENGTH        = 00000025
EXE$GL_PAGED            ********      X  04        LNMTH$L_BYTES         = 00000021
EXE$GL_PFATIM           ********      X  05        LNMTH$L_BYTESLM       = 0000001D
EXE$GQ_SYSTIME          ********      X  05        LNMTH$L_CHILD         = 00000011
EXE$POWERAST            ********      X  05        LNMTH$L_HASH          = 00000001
EXE$SWAPINIT            000004AC RG      04        LNMTH$L_NAME          = 00000009
EXEC_MODE               000001C9 R       04        LNMTH$L_ORB           = 00000005
FILE_DEV_EXEC_ARG       00000345 R       04        LNMTH$L_PARENT        = 0000000D
FILE_DEV_EXEC_LIST      00000239 R       04        LNMTH$L_QTABLE        = 00000019
FILE_DEV_SUPER_ARG      0000035D R       04        LNMTH$L_SIBLING       = 00000015
FILE_DEV_SUPER_LIST     00000205 R       04        LNMTH$M_DIRECTORY     = 00000002
FILLPHD                 000009EF R       05        LNMTH$M_SHAREABLE     = 00000001
GBLDROP                 00000462 R       05        LNMTH$M_SYSTEM        = 00000008
GBLRESET                000004B1 R       05        LNMX$B_FLAGS          = 00000000
GBLTRANS                0000045A R       05        LNMX$B_INDEX          = 00000001
GBLVALID                0000049E R       05        LNMX$C_TABLE          = FFFFFF82
GBLWRTTRANS             0000045A R       05        LNMX$M_TERMINAL       = 00000002
```

```
LNMX$M_XEND                     = 00000004              MMG$M_NOWAIT                      ********    X  05
LNMX$T_XLATION                  = 00000004              MMG$REFCNTNEG                     ********    X  05
LNMX$W_HASH                     = 00000002              MMG$RELPFN                        ********    X  05
LNM_DIRECTORIES_DESC              00000020 R     04     MMG$REMPFN                        ********    X  05
LNM_FILE_DEV_DESC                 00000037 R     04     MMG$SHRCNTNEG                     ********    X  05
LNM_GROUP                         0000014D R     04     MMG$SVAPTECHK                     ********    X  05
LNM_GROUP_LENGTH                = 00000009              MMG$UNLOCK                        ********    X  05
LNM_JOB                           00000156 R     04     MMG$WRTMFYPAG                     ********    X  05
LNM_JOB_LENGTH                  = 00000007              NOTSHELL                          00000689 R     05
LNM_NO_ALIAS                      000001D5 R     04     NOTVALID                          00000231 R     05
LNM_PERMANENT_MAILBOX_DESC        0000004B R     04     NTYP1                             000007C2 R     05
LNM_PROCESS                       0000015D R     04     OP$_CMPL                        = 000000D1
LNM_PROCESS_DIRECTORY             00000168 R     04     OP$_CMPW                        = 000000B1
LNM_PROCESS_DIRECTORY_LENGTH    = 00000015              OP$_CVTLW                       = 000000F7
LNM_PROCESS_LENGTH              = 0000000B              OP$_DECL                        = 000000D7
LNM_SYSTEM                        0000017D R     04     OP$_DECW                        = 000000B7
LNM_SYSTEM_DESC                   00000068 R     04     OP$_INCL                        = 000000D6
LNM_SYSTEM_DIRECTORY              00000187 R     04     OP$_INCW                        = 000000B6
LNM_SYSTEM_DIRECTORY_DESC         00000070 R     04     OP$_MOVL                        = 000000D0
LNM_SYSTEM_DIRECTORY_LENGTH     = 00000014              OP$_MOVW                        = 000000B0
LNM_SYSTEM_DIR_LNMTH              0000002B RG    03     OP$_MOVZWL                      = 0000003C
LNM_SYSTEM_DIR_ORB                00000058 R     03     OP$_TSTL                        = 000000D5
LNM_SYSTEM_LENGTH               = 0000000A              OP$_TSTW                        = 000000B5
LNM_SYSTEM_TABLE                  0000019B R     04     ORB$B_FLAGS                     = 0000000B
LNM_SYSTEM_TABLE_LENGTH         = 00000010              ORB$B_TYPE                      = 0000000A
LNM_SYS_DIR_ORB_SIZ             = 00000068              ORB$K_LENGTH                    = 00000058
LNM_SYS_DIR_SIZ                 = 000000C0              ORB$L_ACL_COUNT                 = 00000028
LNM_TEMPORARY_MAILBOX_DESC        00000078 R     04     ORB$L_ACL_DESC                  = 0000002C
LOG_GROUP                         000001AB R     04     ORB$L_ACL_MUTEX                 = 00000004
LOG_GROUP_LENGTH                = 00000009              ORB$L_GRP_PROT                  = 00000020
LOG_G_ARG                         00000375 R     04     ORB$L_OWNER                     = 00000000
LOG_G_DESC                        00000095 R     04     ORB$L_OWN_PROT                  = 0000001C
LOG_G_LIST                        00000249 R     04     ORB$L_SYS_PROT                  = 00000018
LOG_PROCESS                       000001B4 R     04     ORB$L_WOR_PROT                  = 00000024
LOG_PROCESS_LENGTH              = 0000000B              ORB$Q_MODE_PROT                 = 00000010
LOG_P_ARG                         0000038D R     04     ORB$R_MAX_CLASS                 = 00000044
LOG_P_DESC                        0000009D R     04     ORB$R_MIN_CLASS                 = 00000030
LOG_P_LIST                        00000259 R     04     ORB$S_MAX_CLASS                 = 00000014
LOG_SYSTEM                        000001BF R     04     ORB$S_MIN_CLASS                 = 00000014
LOG_SYSTEM_LENGTH               = 0000000A              ORB$W_REFCOUNT                  = 0000000E
LOG_S_ARG                         000003A5 R     04     ORB$W_SIZE                      = 00000008
LOG_S_DESC                        000000A5 R     04     OSDISPATCH                        00000237 R     05
LOG_S_LIST                        00000275 R     04     OSINIT                            00000A61 R     05
LOOP                              00000000 R     05     OSWPEXIT                          00000401 R     05
MMG$ALLOCPFN                      ********    X  05     OSWPPCB                           00000014 R     02
MMG$AL_SYSPCB                     ********    X  05     OSWPPGS                           00000012 R     02
MMG$DALCPAGFIL                    ********    X  05     OUTSWAP                           00000116 R     05
MMG$DECPTREF                      ********    X  05     OWSLOOP                           00000215 R     05
MMG$DELCONPFN                     ********    X  05     P1SYSVECTORS                      ********    X  04
MMG$DELWSLEX                      ********    X  05     PCB$B_ASTACT                    = 0000000C
MMG$GB_FREWFLGS                   ********    X  05     PCB$B_ASTEN                     = 0000000D
MMG$GL_GPTBASE                    ********    X  05     PCB$B_PRI                       = 0000000B
MMG$GL_PAGSWPVC                   ********    X  05     PCB$B_TYPE                      = 0000000A
MMG$GL_SPTBASE                    ********    X  05     PCB$L_ASTQFL                    = 00000010
MMG$INSPFNH                       ********    X  05     PCB$L_PHD                       = 0000006C
MMG$INSPFNT                       ********    X  05     PCB$L_PHYPCB                    = 00000018
MMG$IOLOCKPAG                     ********    X  05     PCB$L_PID                       = 00000060
```

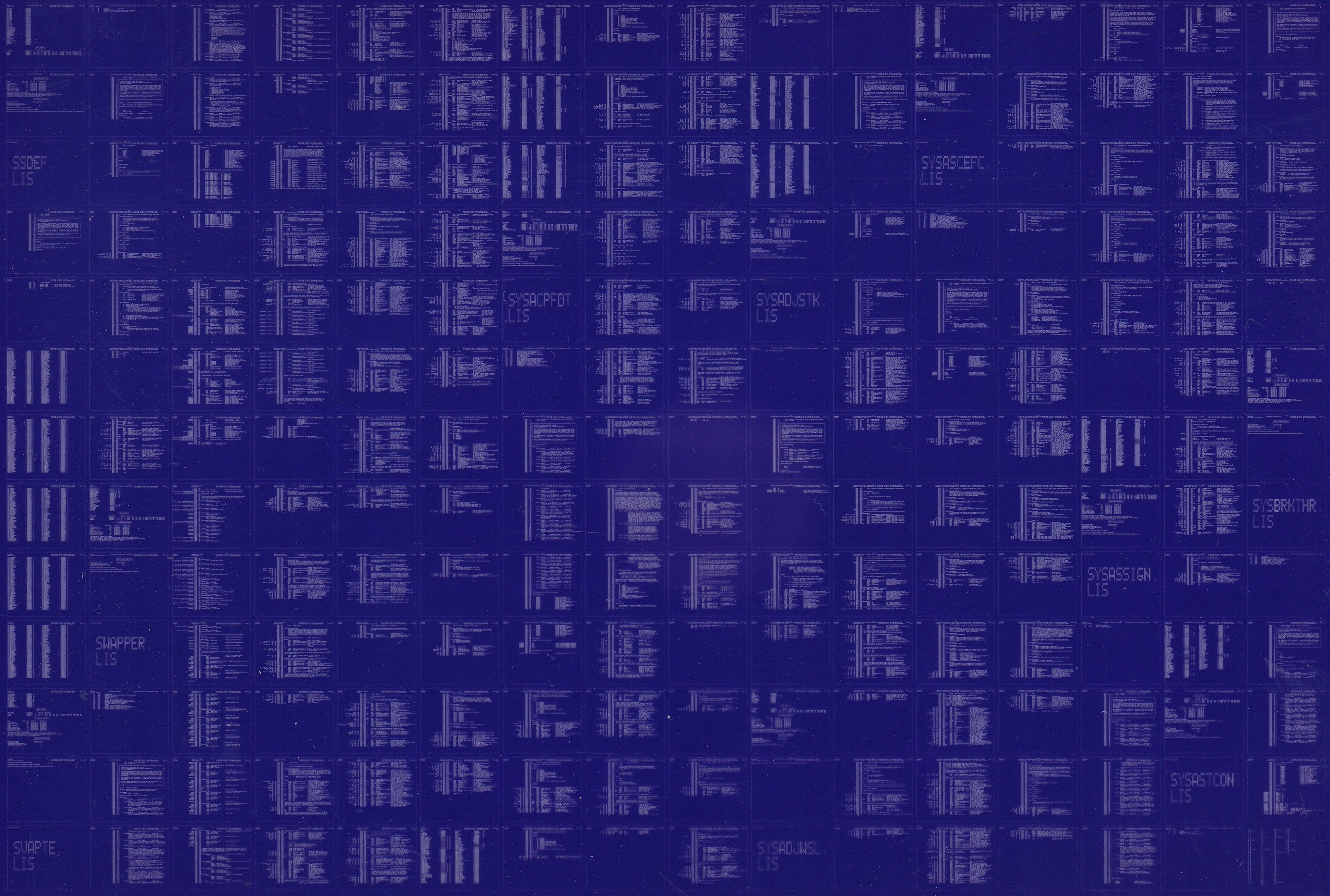
```
PCB$L_STS                      = 00000024                PHD$V_NOACCVIO                 = 00000003
PCB$L_WSSWP                    = 00000020                PHD$W_EMPTPG                   = 000000D8
PCB$V_INQUAN                   = 00000003                PHD$W_FLAGS                    = 00000036
PCB$V_PHDRES                   = 00000012                PHD$W_PHVINDEX                 = 00000042
PCB$V_RES                      = 00000000                PHD$W_PTCNTACT                 = 00000070
PCB$V_WAKEPEN                  = 0000000C                PHD$W_QUANT                    = 0000003C
PCB$W_APTCNT                   = 00000030                PHD$W_SWAPSIZE                 = 00000052
PCB$W_GPGCNT                   = 00000034                PHD$W_WSLAST                   = 00000012
PCB$W_PPGCNT                   = 00000036                PHD$W_WSLIST                   = 00000008
PERMANENT_MAILBOX_ARG            000003BD R       04     PHV$GL_PIXBAS                    ********   X    05
PERMANENT_MAILBOX_LIST           00000239 R       04     PHV$GL_REFCBAS                   ********   X    05
PFL$L_VBN                      = 00000010                PPGTBLTRANS                      0000052A R      05
PFL$L_WINDOW                   = 0000000C                PPGTBLVALID                      0000052A R      05
PFN$AB_STATE                     ********   X     05     PQL$AB_SYSPQL                    00000465 RG     04
PFN$AB_TYPE                      ********   X     05     PQL$C_SYSPQLLEN                = 00000046  G
PFN$AL_BAK                       ********   X     05     PQL$_ASTLM                     = 00000001
PFN$AL_HEAD                      ********   X     05     PQL$_BIOLM                     = 00000002
PFN$AL_PTE                       ********   X     05     PQL$_BYTLM                     = 00000003
PFN$AW_REFCNT                    ********   X     05     PQL$_CPULM                     = 00000004
PFN$AW_SWPVBN                    ********   X     05     PQL$_DIOLM                     = 00000005
PFN$AX_FLINK                     ********   X     05     PQL$_ENQLM                     = 0000000C
PFN$AX_SHRCNT                    ********   X     05     PQL$_FILLM                     = 00000006
PFN$AX_WSLX                      ********   X     05     PQL$_JTQUOTA                   = 0000000E
PFN$C_ACTIVE                   = 00000007                PQL$_LISTEND                   = 00000000
PFN$C_BADPAGLST                = 00000002                PQL$_PGFLQUOTA                 = 00000007
PFN$C_FREPAGLST                = 00000000                PQL$_PRCLM                     = 00000008
PFN$C_GBLWRT                   = 00000003                PQL$_TQELM                     = 00000009
PFN$C_GLOBAL                   = 00000002                PQL$_WSDEFAULT                 = 0000000B
PFN$C_GPGTBL                   = 00000005                PQL$_WSEXTENT                  = 0000000D
PFN$C_MFYPAGLST                = 00000001                PQL$_WSQUOTA                   = 0000000A
PFN$C_PPGTBL                   = 00000004                PR$_IPL                        = 00000012
PFN$C_PROCESS                  = 00000000                PR$_TBIA                       = 00000039
PFN$C_RDERR                    = 00000004                PRC$M_NOACNT                   = 00000008
PFN$C_RDINPROG                 = 00000006                PRC$M_SSRWAIT                  = 00000001
PFN$C_RELPEND                  = 00000003                PROCDROP                         0000047B R      05
PFN$C_SYSTEM                   = 00000001                PROCTRANS                        000004DE R      05
PFN$C_WRTINPROG                = 00000005                PROCVALID                        000004F0 R      05
PFN$M_COLLISION                = 00000010                PROCWRT                          00000258 R      05
PFN$M_DELCON                   = 00000010                PSL$C_EXEC                     = 00000001
PFN$M_MODIFY                   = 00000080                PSL$C_KERNEL                   = 00000000
PFN$S_BAK                      = 00000017                PSL$C_SUPER                    = 00000002
PFN$S_LOC                      = 00000003                PTE$C_ERKW                     = 30000000
PFN$S_PAGTYP                   = 00000003                PTE$C_URKW                     = 70000000
PFN$S_PGFLX                    = 00000008                PTE$M_MODIFY                   = 04000000
PFN$V_BADPAG                   = 00000005                PTE$M_OWN                      = 01800000
PFN$V_BAK                      = 00000000                PTE$M_PFN                      = 001FFFFF
PFN$V_LOC                      = 00000000                PTE$M_PROT                     = 78000000
PFN$V_MODIFY                   = 00000007                PTE$M_TYP0                     = 00400000
PFN$V_PAGTYP                   = 00000000                PTE$M_TYP1                     = 04000000
PFN$V_PGFLX                    = 00000018                PTE$M_VALID                    = 80000000
PHD$B_ASTLVL                   = 000000CF                PTE$S_GPTX                     = 00000016
PHD$B_PAGFIL                   = 0000001F                PTE$S_PFN                      = 00000015
PHD$L_BAK                      = 00000044                PTE$S_PGFLVB                   = 00000016
PHD$L_P0BR                     = 000000C8                PTE$V_GPTX                     = 00000000
PHD$L_P1BR                     = 000000D0                PTE$V_MODIFY                   = 0000001A
PHD$L_PCB                      = 00000078                PTE$V_PFN                      = 00000000
PHD$L_WSLX                     = 00000048                PTE$V_PGFLVB                   = 00000000
```

SYS
V04

```
PTE$V_TYP0                      = 00000016                 SWP$GL_ISPAGCNT                 ********   X   05
PTE$V_TYP1                      = 0000001A                 SWP$GL_ISWPCNT                  ********   X   05
PTE$V_VALID                     = 0000001F                 SWP$GL_ISWPPAGES                ********   X   05
QEMPTY                            0000008C R     05        SWP$GL_MAP                      ********   X   05
RECONNECT                         00000778 R     05        SWP$GL_OSWPCNT                  ********   X   05
RELDELPAGE                        00000A71 R     05        SWP$GL_PHDBASVA                 ********   X   05
RELINIT                           00000A57 R     05        SWP$GL_SHELIO                   ********   X   05
RELPAGE                           00000A74 R     05        SWP$GL_SHELLBAS                 ********   X   05
RELPHD                            00000316 R     05        SWP$GL_SWTIME                   ********   X   05
RPGCNT                            00000010 R     02        SWP$GW_BALCNT                   00000018 RG    02
RSVAPTE                           0000000C R     02        SWP$GW_IBALSETX                 ********   X   05
RWSSWP                            00000008 R     02        SWP$SHELINIT                    ********   X   05
SCH$AQ_COMOH                      ********   X   05        SWPREAD                         00000A88 R     05
SCH$CHSEP                         ********   X   05        SWPWRITE                        00000A8E R     05
SCH$GB_SIP                        ********   X   05        SYS$CRELNM                      ********   X   04
SCH$GL_COMOQS                     ********   X   05        SYS$CREPRC                      ********   X   04
SCH$GL_CURPCB                     ********   X   05        SYS$GB_DEFPRI                   ********   X   05
SCH$GL_FREECNT                    ********   X   05        SYSTEM_ARG                      000003D5 R     04
SCH$GL_FREELIM                    ********   X   05        SYSTEM_LIST                     00000285 R     04
SCH$GL_MFYCNT                     ********   X   05        SYSTEM_TABLE                    000000C0 R     03
SCH$GL_MFYLIM                     ********   X   05        SYSTEM_TABLE_LNMTH              000000E7 R     03
SCH$GL_MFYLOLIM                   ********   X   05        SYSTEM_TABLE_ORB                00000110 R     03
SCH$GL_PCBVEC                     ********   X   05        SYSTEM_TABLE_ORB_SIZ          = 00000070
SCH$GL_SWPRATE                    ********   X   05        SYSTEM_TABLE_SIZE             = 000000C0
SCH$GQ_COLPGWQ                    ********   X   05        SYS_DISK_ARG                    00000180 R     03
SCH$GQ_HIBWQ                      ********   X   05        SYS_DISK_DESC                   000000AD R     04
SCH$GW_DELPHDCT                   ********   X   05        SYS_SYSDEVICE_ARG               00000198 R     03
SCH$GW_QUAN                       ********   X   05        SYS_SYSDEVICE_DESC              000000BD R     04
SCH$GW_SWPFAIL                    ********   X   05        TEMPORARY_MAILBOX_ARG           000003ED R     04
SCH$GW_SWPFCNT                    0000001A RG    02        TEMPORARY_MAILBOX_LIST          000002A1 R     04
SCH$NEWLVL                        ********   X   05        TERMINAL_BUFFER                 000001D9 R     04
SCH$OSWPSCHED                     ********   X   05        TMP...                        = 00000001
SCH$V_MPW                         ********   X   05        TRNLOG_GS_ARG                   00000405 R     04
SCH$V_SIP                         ********   X   05        TRNLOG_GS_DESC                  000000D2 R     04
SCH$WAITK                         ********   X   05        TRNLOG_GS_LIST                  000002B1 R     04
SETASTLVL                         00000982 R     05        TRNLOG_PGS_ARG                  0000044D R     04
SETUP                             00000668 R     05        TRNLOG_PGS_DESC                 00000129 R     04
SETWRTBAK                         0000050E R     05        TRNLOG_PGS_LIST                 00000305 R     04
SGN$GL_BALSETCT                   ********   X   05        TRNLOG_PG_ARG                   0000041D R     04
SGN$GL_FREEGOAL                   ********   X   05        TRNLOG_PG_DESC                  000000EE R     04
SGN$GL_FREELIM                    ********   X   05        TRNLOG_PG_LIST                  000002CD R     04
SGN$GL_PAGEDYN                    ********   X   04        TRNLOG_PS_ARG                   00000435 R     04
SGN$GL_PHDPAGCT                   ********   X   05        TRNLOG_PS_DESC                  0000010B R     04
SPACEFAIL                         00000254 R     05        TRNLOG_PS_LIST                  000002E9 R     04
SUPER_MODE                        000001D1 R     04        TTODESC                         00000013 R     04
SWAPEXIT                          000009E1 R     05        VA$S_VPN                      = 00000015
SWAPEXITA                         000009E7 R     05        VA$V_SYSTEM                   = 0000001F
SWAPRETRY                         000009D7 R     05        VA$V_VPN                      = 00000009
SWAPSCHED                         00000090 R     05        WSL$C_SYSTEM                  = 00000002
SWP$GB_ISWPRI                     ********   X   05        WSL$M_MODIFY                  = 00000100
SWP$GB_PRIO                       ********   X   05        WSL$M_PAGTYP                  = 0000000E
SWP$GL_BALBASE                    ********   X   05        WSL$M_PFNLOCK                 = 00000010
SWP$GL_BALSPT                     ********   X   05        WSL$M_VALID                   = 00000001
SWP$GL_BSLOTSZ                    ********   X   05        WSL$M_WSLOCK                  = 00000020
SWP$GL_HISWPCNT                   ********   X   05        WSL$V_MODIFY                  = 00000008
SWP$GL_HOSWPCNT                   ********   X   05        WSL$V_PAGTYP                  = 00000001
SWP$GL_INPCB                      ********   X   05        WSL$V_PFNLOCK                 = 00000004
```

```
WSL$V_VALID                     = 00000000
WSL$V_WSLOCK                    = 00000005
WSLERR                            00000526 R     05
WSLOOP                            00000709 R     05

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

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (   0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                           00000000  (     0.)  01 (   1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$$$220                          0000001C  (    28.)  02 (   2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC LONG
$$$260                          000001B0  (   432.)  03 (   3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC 21
YF$LOWUSE                       0000063B  (  1595.)  04 (   4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$AEXENONPAGED                   00000B26  (  2854.)  05 (   5.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
Z$INIT$PFN_FIXUP_TABLE          0000005A  (    90.)  06 (   6.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE

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

Phase                       Page faults    CPU Time       Elapsed Time
-----                       -----------    --------       ------------
Initialization                       35    00:00:00.07    00:00:01.61
Command processing                  127    00:00:00.50    00:00:04.66
Pass 1                              515    00:00:22.43    00:01:06.36
Symbol table sort                     0    00:00:02.75    00:00:05.98
Pass 2                              423    00:00:06.86    00:00:20.84
Symbol table output                   1    00:00:00.33    00:00:01.01
Psect synopsis output                 0    00:00:00.04    00:00:00.04
Cross-reference output                0    00:00:00.00    00:00:00.00
Assembler run totals               1103    00:00:32.98    00:01:40.51

The working set limit was 2250 pages.
133641 bytes (262 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1677 non-local and 138 local symbols.
2492 source lines were read in Pass 1, producing 45 object records in Pass 2.
38 pages of virtual memory were used to define 36 macros.

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

Macro library name                         Macros defined
------------------                         --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                   20
_$255$DUA28:[SYSLIB]STARLET.MLB;2                13
TOTALS (all libraries)                           33

1690 GETS were required to define 33 macros.

There were no errors, warnings or information messages.

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

0381 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY



SSDEF
LIS

SYSASCEFC
LIS

SYSACPFDT
LIS

SYSADJSTK
LIS

SYSBRKTHR
LIS

SYSASSIGN
LIS

SWAPPER
LIS

SYSASTCON
LIS

SVAPTE
LIS

SYSADJWSL
LIS