



```

SSSSSSSS  WW      WW      AAAAAA  PPPPPPPP  PPPPPPPP  EEEEEEEEE  RRRRRRRR
SSSSSSSS  WW      WW      AAAAAA  PPPPPPPP  PPPPPPPP  EEEEEEEEE  RRRRRRRR
SS        WW      WW      AA        AA  PP        PP  PP        PP  EEE         RR         RR
SS        WW      WW      AA        AA  PP        PP  PP        PP  EEE         RR         RR
SS        WW      WW      AA        AA  PP        PP  PP        PP  EEE         RR         RR
SS        WW      WW      AA        AA  PPPPPPPP  PPPPPPPP  EEEEEEEEE  RRRRRRRR
SSSSSSSS  WW      WW      AA        AA  PPPPPPPP  PPPPPPPP  EEEEEEEEE  RRRRRRRR
SS        WW      WW      AAAAAAAAAA  PP        PP  PP        PP  EEE         RR         RR
SS        WW      WW      AAAAAAAAAA  PP        PP  PP        PP  EEE         RR         RR
SS        WWWW    WWWW    AA        AA  PP        PP  PP        PP  EEE         RR         RR
SS        WWWW    WWWW    AA        AA  PP        PP  PP        PP  EEE         RR         RR
SSSSSSSS  WW      WW      AA        AA  PP        PP  PP        PP  EEEEEEEEE  RR         RR
SSSSSSSS  WW      WW      AA        AA  PP        PP  PP        PP  EEEEEEEEE  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

```

(2)	231	DECLARATIONS
(9)	885	EX\$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 1 .TITLE SWAPPER WORKING SET SWAPPER
0000 2 .IDENT 'V04-000'
0000 3
0000 4 *****
0000 5
0000 6 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 7 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 8 * ALL RIGHTS RESERVED. *
0000 9
0000 10 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 11 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 12 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 13 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 14 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 15 * TRANSFERRED. *
0000 16
0000 17 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 18 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 19 * CORPORATION. *
0000 20
0000 21 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 22 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 23
0000 24
0000 25 *****
0000 26
0000 27 **
0000 28 FACILITY: EXECUTIVE, SWAPPER
0000 29
0000 30 ABSTRACT: THE SWAPPER SCHEDULES AND EXECUTES SWAPPING OF PROCESS
0000 31 WORKING SETS BETWEEN SWAP STORAGE AND MAIN MEMORY.
0000 32
0000 33 ENVIRONMENT:
0000 34 MODE = KERNEL , RESIDENT
0000 35
0000 36 AUTHOR: R. HUSTVEDT CREATION DATE: 30-NOV-76
0000 37
0000 38 MODIFIED BY:
0000 39
0000 40 V03-029 ACG0440 Andrew C. Goldstein, 24-Jul-1984 10:50
0000 41 Add ref count field to ORB
0000 42
0000 43 V03-028 LMP0275 L. Mark Pilant, 12-Jul-1984 20:31
0000 44 Initialize the ACL info in the ORB to be a null descriptor
0000 45 list rather than an empty queue. This avoids the overhead
0000 46 of locking and unlocking the ACL mutex, only to find out
0000 47 that the ACL was empty.
0000 48
0000 49 V03-027 TMK0011 Todd M. Katz 11-Apr-1984
0000 50 The ACL mutexes within the Object Rights Blocks of the system
0000 51 and system directory logical name tables are currently
0000 52 incorrectly initialized to ^X00001111. Initialize them to
0000 53 ^X0000FFFF.
0000 54
0000 55 V03-026 MSH0029 Michael S. Harvey 9-Apr-1984
0000 56 The translation of LNM$TEMPORARY_MAILBOX will now be LNM$JOB
0000 57 instead of LNM$GROUP. This is a part of an effort to close

```

```
0000 58 : some privilege related security holes involving logical names
0000 59 : and temporary mailbox creation.
0000 60 :
0000 61 : V03-025 TMK0010 Todd M. Katz 26-Mar-1984
0000 62 : Modify the logical name system services to make use of the
0000 63 : updated internal protection checking mechanisms. What this
0000 64 : involves is replacing the system directory and system logical
0000 65 : name tables' CHIP protection templates with quad-word aligned
0000 66 : Object Rights Blocks.
0000 67 :
0000 68 : V03-024 TMK0009 Todd M. Katz 07-Mar-1984
0000 69 : Add a hash code field, LNM$W_HASH, to every translation block
0000 70 : of every logical name table template defined. This hash code
0000 71 : field will be used in an optimization of logical name table name
0000 72 : PROCESSING.
0000 73 :
0000 74 : V03-023 LY00b7 Larry Yetto 16-FEB-1984 14:33
0000 75 : Fix alignment of logical name tables
0000 76 :
0000 77 : V03-022 ROW62094 Ralph O. Weber 25-JAN-1984
0000 78 : Add PROCESSING for inswapped global page when there currently
0000 79 : exists a equivalent global page having a page read error.
0000 80 : This makes the list of ossible conditions to be handled for an
0000 81 : inswap of a global page: 1) no equivalent global page exsts,
0000 82 : 2) an equivalent global page exists, 3) the equivalent page is
0000 83 : still being read (from a page fault read), 4) the equivalent
0000 84 : page was read but encountered a page read error.
0000 85 :
0000 86 : V03-021 TMK0008 Todd M. Katz 06-Jan-1984
0000 87 : Never allow the system directory logical name table to be
0000 88 : deleted. This is done as follows:
0000 89 :
0000 90 : 1. Set the LNMB$V NODELETE bit within the LNMB$B_FLAGS field
0000 91 : of the system directory logical name table.
0000 92 : 2. Check for this bit within the logical name system services
0000 93 : whenever a LNMB is to be deleted.
0000 94 : 3. If this bit is set, do not allow the LNMB to be deleted;
0000 95 : otherwise, proceed with the deletion.
0000 96 :
0000 97 : This mechanism will prevent the directories from ever being
0000 98 : explicitly or implicitly deleted which can cause all sorts
0000 99 : of problems.
0000 100 :
0000 101 : V03-020 TMK0007 Todd M. Katz 25-Dec-1983
0000 102 : Make a small change to TMK0006. Setup the remaining quota
0000 103 : byte field of the system directory logical name table with a
0000 104 : value of positive infinity (i.e. - ^X7FFFFFFF) instead of a
0000 105 : value of positive infinity minus the size of the system table.
0000 106 : This is necessary because the routine which is used to
0000 107 : appropriately insert the system table performs the necessary
0000 108 : quota subtractions; thus, in TMK0006 quota for the system
0000 109 : table was being subtracted twice from the system directory
0000 110 : logical name table.
0000 111 :
0000 112 : V03-019 TMK0006 Todd M. Katz 18-Dec-1983
0000 113 : Handcraft the system logical name table, LNM$SYSTEM TABLE,
0000 114 : instead of using the system service ($CRELNT) to CREATE it.
```

```

0000 115 :
0000 116 :
0000 117 :
0000 118 :
0000 119 :
0000 120 :
0000 121 :
0000 122 :
0000 123 :
0000 124 :
0000 125 :
0000 126 :
0000 127 :
0000 128 :
0000 129 :
0000 130 :
0000 131 :
0000 132 :
0000 133 :
0000 134 :
0000 135 :
0000 136 :
0000 137 :
0000 138 :
0000 139 :
0000 140 :
0000 141 :
0000 142 :
0000 143 :
0000 144 :
0000 145 :
0000 146 :
0000 147 :
0000 148 :
0000 149 :
0000 150 :
0000 151 :
0000 152 :
0000 153 :
0000 154 :
0000 155 :
0000 156 :
0000 157 :
0000 158 :
0000 159 :
0000 160 :
0000 161 :
0000 162 :
0000 163 :
0000 164 :
0000 165 :
0000 166 :
0000 167 :
0000 168 :
0000 169 :
0000 170 :
0000 171 :

```

V03-018 WMC0018 Wayne Cardoza 02-Dec-1983  
PHDSW\_WSLX, PHDSW\_BAK have become longwords.

V03-017 TMK0005 Todd M. Katz 19-Oct-1983  
Add the following kernel mode logical names which will be used in order to optimize \$TRNLOGs:

```

TRNLOG$_PROCESS_GROUP
TRNLOG$_PROCESS_SYSTEM
TRNLOG$_GROUP_SYSTEM
TRNLOG$_PROCESS_GROUP_SYSTEM

```

Also, CREATE LNM\$TEMPORARY\_MAILBOX with a translation of LNM\$GROUP instead of LNM\$JOB.

V03-016 TMK0004 Todd M. Katz 11-Oct-1983  
Make the following changes to the logical names and tables that are CREATED at system initialization time:

1. CREATE LNM\$FILE\_DEV with the translations LNM\$PROCESS, LNM\$JOB, LNM\$GROUP, LNM\$SYSTEM (the change is the addition of the LNM\$JOB translation).
2. CREATE LNM\$TEMPORARY\_MAILBOX with the translation LNM\$JOB.
3. Remove LNM\$TRNLOG\_PG, LNM\$TRNLOG\_PS, LNM\$TRNLOG\_GS, LNM\$TRNLOG\_PGS, and LNM\$DEFAULT\_SEARCH.
4. Add the following kernel mode logical names which will be used in order to provide compatibility between V3 and V4 for all of the old logical name system services (\$TRNLOG, \$CRELOG, \$DELLOG):

```

LOG$PROCESS
LOG$GROUP
LOG$SYSTEM

```

5. Add to the PQB\$AB\_SYSPQL quota list a PQL\$\_JTQUOTA item.

V03-015 TMK0003 Todd M. Katz 09-Apr-1983  
Statically define the CHIP protection structure of LNM\$SYSTEM DIRECTORY, set the (internal) attribute bit LNM\$V\_SYSTEM when creating LNM\$SYSTEM TABLE, and change the CHIP protection of LNM\$SYSTEM DIRECTORY and LNM\$SYSTEM TABLE to S:RWE O:RWE G:R W:R. Also, CREATE the supervisor mode logical name LNM\$FILE\_DEV with the translations LNM\$PROCESS, LNM\$GROUP, LNM\$SYSTEM instead of LNM\$DEFAULT\_SEARCH, mark both translations of LNM\$DIRECTORIES with the TERMINAL attribute, and CREATE the non-aliasable kernel mode logical names LNM\$TRNLOG\_PG, LNM\$TRNLOG\_PS, LNM\$TRNLOG\_GS, LNM\$TRNLOG\_PGS.

V03-014 KDM0052 Kathleen D. Morse 11-Jul-1983  
Replace references of PR\$\_TODR with EXE\$GQ\_SYSTIME+2.

V03-013 DMW4060 DMWalp 23-Jun-1983  
Change \$xxLNM value parameters to be by reference

V03-012 DMW4054 DMWalp 21-Jun-1983  
Convert SYS\$DISK and SYS\$SYSDEVICE creation from \$CRELOG

0000 172 :  
0000 173 :  
0000 174 :  
0000 175 :  
0000 176 :  
0000 177 :  
0000 178 :  
0000 179 :  
0000 180 :  
0000 181 :  
0000 182 :  
0000 183 :  
0000 184 :  
0000 185 :  
0000 186 :  
0000 187 :  
0000 188 :  
0000 189 :  
0000 190 :  
0000 191 :  
0000 192 :  
0000 193 :  
0000 194 :  
0000 195 :  
0000 196 :  
0000 197 :  
0000 198 :  
0000 199 :  
0000 200 :  
0000 201 :  
0000 202 :  
0000 203 :  
0000 204 :  
0000 205 :  
0000 206 :  
0000 207 :  
0000 208 :  
0000 209 :  
0000 210 :  
0000 211 :  
0000 212 :  
0000 213 :  
0000 214 :  
0000 215 :  
0000 216 :  
0000 217 :  
0000 218 :  
0000 219 :  
0000 220 :  
0000 221 :  
0000 222 :  
0000 223 :  
0000 224 :  
0000 225 :  
0000 226 :  
0000 227 :  
0000 228 :--

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

```

0000 231      .SBTTL  DECLARATIONS
0000 232      :
0000 233      : INCLUDE FILES:
0000 234      :
0000 235      :
0000 236      $ACBDEF      ; DEFINE AST CONTROL BLOCK OFFSETS
0000 237      $DYNDEF      ; DEFINE STRUCTURE TYPE CODES
0000 238      $IPLDEF      ; DEFINE INTERRUPT PRIORITY LEVELS
0000 239      $IRPDEF      ; DEFINE I/O REQUEST PACKET OFFSETS
0000 240      $LNMDEF      ; DEFINE LOGICAL NAME OFFSETS
0000 241      $LNMSTRDEF   ; DEFINE LOGICAL NAME STRUCTURE OFFSETS
0000 242      $OPDEF       ; DEFINE OPCODE EQUIVALENT VALUES
0000 243      $ORBDEF      ; DEFINE OBJECT RIGHTS BLOCK OFFSETS
0000 244      $PCBDEF      ; DEFINE PCB OFFSETS
0000 245      $PFLDEF      ; DEFINE SWAP FILE TABLE OFFSETS
0000 246      $PFNDEF      ; DEFINE PFN VALUES
0000 247      $PHDDEF      ; DEFINE PHD OFFSETS
0000 248      $PQLDEF      ; DEFINE QUOTA SYMBOLS
0000 249      $PRDEF       ; DEFINE PROCESSOR REGISTERS
0000 250      $PRCDEF      ; CREATE PROCESS FLAGS
0000 251      $PSLDEF      ; DEFINE PSL VALUES
0000 252      $PTEDEF      ; DEFINE PAGE TABLE ENTRY
0000 253      $VADEF       ; DEFINE VIRTUAL ADDRESS FIELDS
0000 254      $WSLDEF      ; DEFINE WORKING SET LIST BITS
0000 255      :
0000 256      :
0000 257      : ASSUMPTIONS ABOUT THE STRUCTURE OF LOGICAL NAME AND OBJECT RIGHTS BLOCKS:
0000 258      :
0000 259      :
0000 260      ASSUME  LNMB$$_FLINK,      EQ,  0
0000 261      ASSUME  LNMB$$_FLINK+4,    EQ,  LNMB$$_BLINK
0000 262      ASSUME  LNMB$$_BLINK+4,    EQ,  LNMB$$_SIZE
0000 263      ASSUME  LNMB$$_SIZE+2,     EQ,  LNMB$$_TYPE
0000 264      ASSUME  LNMB$$_TYPE+1,     EQ,  LNMB$$_ACMODE
0000 265      ASSUME  LNMB$$_ACMODE+1,   EQ,  LNMB$$_TABLE
0000 266      ASSUME  LNMB$$_TABLE+4,    EQ,  LNMB$$_FLAGS
0000 267      ASSUME  LNMB$$_FLAGS+1,    EQ,  LNMB$$_NAME
0000 268      :
0000 269      ASSUME  LNMX$$_FLAGS,      EQ,  0
0000 270      ASSUME  LNMX$$_FLAGS+1,    EQ,  LNMX$$_INDEX
0000 271      ASSUME  LNMX$$_INDEX+1,    EQ,  LNMX$$_HASH
0000 272      ASSUME  LNMX$$_HASH+2,     EQ,  LNMX$$_XLATION
0000 273      :
0000 274      ASSUME  LNMT$$_FLAGS,      EQ,  0
0000 275      ASSUME  LNMT$$_FLAGS+1,    EQ,  LNMT$$_HASH
0000 276      ASSUME  LNMT$$_HASH+4,     EQ,  LNMT$$_ORB
0000 277      ASSUME  LNMT$$_ORB+4,      EQ,  LNMT$$_NAME
0000 278      ASSUME  LNMT$$_NAME+4,     EQ,  LNMT$$_PARENT
0000 279      ASSUME  LNMT$$_PARENT+4,   EQ,  LNMT$$_CHILD
0000 280      ASSUME  LNMT$$_CHILD+4,    EQ,  LNMT$$_SIBLING
0000 281      ASSUME  LNMT$$_SIBLING+4,  EQ,  LNMT$$_QTABLE
0000 282      ASSUME  LNMT$$_QTABLE+4,   EQ,  LNMT$$_BYTESLM
0000 283      ASSUME  LNMT$$_BYTESLM+4,  EQ,  LNMT$$_BYTES
0000 284      :
0000 285      ASSUME  ORB$$_OWNER,        EQ,  0
0000 286      ASSUME  ORB$$_OWNER+4,     EQ,  ORB$$_ACL_MUTEX
0000 287      ASSUME  ORB$$_ACL_MUTEX+4,  EQ,  ORB$$_SIZE

```



```
0000 288 ASSUME ORBSW_SIZE+2, EQ, ORBSB_TYPE
0000 289 ASSUME ORBSB_TYPE+1, EQ, ORBSB_FLAGS
0000 290 ASSUME ORBSB_FLAGS+3, EQ, ORBSW_REFCOUNT
0000 291 ASSUME ORBSW_REFCOUNT+2, EQ, ORBSQ_MODE_PROT
0000 292 ASSUME ORBSQ_MODE_PROT+8, EQ, ORBSL_SYS_PROT
0000 293 ASSUME ORBSL_SYS_PROT+4, EQ, ORBSL_OWN_PROT
0000 294 ASSUME ORBSL_OWN_PROT+4, EQ, ORBSL_GRP_PROT
0000 295 ASSUME ORBSL_GRP_PROT+4, EQ, ORBSL_WOR_PROT
0000 296 ASSUME ORBSL_WOR_PROT+4, EQ, ORBSL_ACL_COUNT
0000 297 ASSUME ORBSL_ACL_COUNT+4, EQ, ORBSL_ACL_DESC
0000 298 ASSUME ORBSL_ACL_DESC+4, EQ, ORBSR_MIN_CLASS
0000 299 ASSUME ORBSR_MIN_CLASS+ORBSB_MIN_CLASS, -
0000 300 EQ, ORBSR_MAX_CLASS
0000 301 ASSUME ORBSR_MAX_CLASS+ORBSB_MAX_CLASS, -
0000 302 EQ, ORBSK_LENGTH
```

```

0000 304
0000 305 :
0000 306 : OWN STORAGE:
0000 307 :
0000 308 :
00000000 309 .PSECT $$$220, LONG ; SWAPPER/SCHEDULER WRITABLE DATA
0000 310 IOROUTINE: ; ADDRESS OF PROPER BUILD PACKET ROUTINE
00000000 0000 311 .LONG 0 ;
00000000 0004 312 IOEA: .LONG 0 ; I/O END ACTION RETURN
00000000 0008 313 RWSSWP: .LONG 0 ; REMAINING WS SWP ADDRESS
00000000 000C 314 RSVAPTE: .LONG 0 ; REMAINING SVA OF PTE
0000 0010 315 RPGCNT: .WORD 0 ; REMAINING PAGE COUNT
0000 0012 316 OSWPPGS: .WORD 0 ; OUTSWAP PAGE COUNT
00000000 0014 317 OSWPPCB: .LONG 0 ; PCB ADDRESS OF OUTSWAP PROCESS
0000 0018 318 SWP$GW_BALCNT:: ; COUNT OF PROCESSES IN BALANCE SET
FFFF 0018 319 .WORD -1 ; EXCLUDING NULL PROCESS AND SWAPPER
0000 001A 320 SCH$GW_SWPFCNT:: ; COUNT OF SUCCESSIVE SWAP
0000 001A 321 .WORD 0 ; SCHEDULE FAILURES.
00000000 322 ;
00000000 323 .PSECT $$$260,5 ; WRITABLE, HIGH USE PSECT
0000 324 :
0000 325 :
0000 326 : LNM$SYSTEM_DIRECTORY - THE SYSTEM DIRECTORY LOGICAL NAME TABLE.
0000 327 :
0000 328 :
0000 329 LNM$SYSTEM_DIRECTORY::
00000000 0000 330 .LONG 0 ; FORWARD LINK
00000000 0004 331 .LONG 0 ; BACK LINK
0000 0008 332 .WORD LNM_SYS_DIR_SIZ ; SIZE OF STRUCTURE
40 000A 333 .BYTE DYN$C_LNM ; TYPE OF STRUCTURE
00 000B 334 .BYTE PSL$C_KERNEL ; KERNEL ACCESS MODE
0000002B 000C 335 .ADDRESS LNM_SYSTEM_DIR_LNMTH ; DIRECTORY TABLE HEADER ADDRESS
19 0010 336 .BYTE LNMBSM_NO_ALIAS!- ; DIRECTORY TABLES CAN NOT BE ALIASED
0011 337 LNMBSM_TABLE!- ; DIRECTORIES ARE TABLES
0011 338 LNMBSM_NODELETE ; DIRECTORIES CAN NOT BE DELETED
SF 4D 45 54 53 59 53 24 4D 4E 4C 00 0011 339 .ASCII "LNM$SYSTEM_DIRECTORY" ; NAME OF DIRECTORY TABLE
59 52 4F 54 43 45 52 49 44 001D
14 0011
0026 340
02 0026 341 .BYTE LNMX$M_TERMINAL ; FLAGS BYTE. NO MORE TRANSLATIONS
82 0027 342 .BYTE LNMX$C_TABLE ; TRANSLATION INDEX ( SPECIAL TABLE )
0000 0028 343 .WORD 0 ; TRANSLATION HASH CODE
25 002A 344 .BYTE LNMTH$K_LENGTH ; SIZE OF TABLE HEADER BLOCK
002B 345
002B 346 LNM_SYSTEM_DIR_LNMTH::
03 002B 347 .BYTE LNMTH$M_SHAREABLE!- ; DIRECTORY IS A SHAREABLE TABLE
002C 348 LNMTH$M_DIRECTORY ; TABLE IS A DIRECTORY TABLE
00000000 002C 349 .LONG 0 ; ADDRESS OF HASH TABLE
00000058 0030 350 .ADDRESS LNM_SYSTEM_DIR_ORB ; ADDRESS OF OBJECT RIGHTS BLOCK
00000000 0034 351 .ADDRESS LNM$SYSTEM_DIRECTORY ; ADDRESS OF CONTAINING LNM BLOCK
00000000 0038 352 .LONG 0 ; ADDRESS OF PARENT TABLE
00000000 003C 353 .LONG 0 ; ADDRESS OF CHILD TABLE
00000000 0040 354 .LONG 0 ; ADDRESS OF SIBLING TABLE
0000002B 0044 355 .ADDRESS LNM_SYSTEM_DIR_LNMTH ; ADDRESS OF TABLE HOLDING QUOTA
7FFFFFFF 0048 356 .LONG ^X7FFFFFFF ; INITIAL QUOTA ( POSITIVE INFINITY )
7FFFFFFF 004C 357 .LONG ^X7FFFFFFF ; REMAINING QUOTA ( POSITIVE INFINITY )
0050 358

```

```

04 0050 359          .BYTE  LNMYSM_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$C_ORB       ; BLOCK TYPE
00 0063 367          .BYTE  0          ; NO ACL AS YET
00000000 0064 368          .LONG  0          ; ZERO RESERVED WORD & REF COUNT
00000000 0068 369          .QUAD  0          ; OBJECT DOES NOT HAVE AN ACCESS MODE
00000008 0070 370          .LONG  ^X00000008     ; SYSTEM PROTECTION IS RWE
00000008 0074 371          .LONG  ^X00000008     ; OWNER PROTECTION IS RWE
0000000E 0078 372          .LONG  ^X0000000E     ; GROUP PROTECTION IS R
0000000E 007C 373          .LONG  ^X0000000E     ; WORLD PROTECTION IS R
00000000 0080 374          .LONG  0 0          ; NULL INITIAL ACL
00'00'00'00'00'00'00'00'00'00'00'00'00' 0088 375          .BYTE  0[ORB$S_MIN_CLASS] ; MINIMUM CLASSIFICATION MASK
00'00'00'00'00'00'00'00'00'00'00'00'00' 0094
00'00'00'00'00'00'00'00'00'00'00'00'00' 009C 376          .BYTE  0[ORB$S_MAX_CLASS] ; MAXIMUM CLASSIFICATION MASK
00'00'00'00'00'00'00'00'00'00'00'00'00' 00A8
00B0 377
00B0 378          .ALIGN  5
00000068 00C0 379 LNM_SYS DIR_ORB_SIZ = . - LNM SYSTEM DIR ORB
000000C0 00C0 380 LNM_SYS_DIR_SIZ = . - LNM$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$C_KERNEL    ; KERNEL ACCESS MODE
0000002B' 00CC 392          .ADDRESS LNM_SYSTEM_DIR_LNMTH ; DIRECTORY TABLE HEADER ADDRESS
09 00D0 393          .BYTE  LNMBSM_NO_ALIAS!- ; TABLE CAN NOT BE ALIASED
00D1 394          .LONG  LNMBSM_TABLE ; TABLE
5F 4D 45 54 53 59 53 24 4D 4E 4C 00' 00D1 395          .ASCIIC "LNM$SYSTEM_TABLE" ; TABLE NAME
45 4C 42 41 54 00DD
10 00D1
00E2 396
02 00E2 397          .BYTE  LNMYSM_TERMINAL ; FLAGS BYTE. NO MORE TRANSLATIONS
82 00E3 398          .BYTE  LNMYSM_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          .LONG  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

```

WORKING SET SWAPPER  
DECLARATIONS

H 3

16-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1

Page 9  
(4)

```

00000000 0104 412 .LONG 0 ; INITIAL QUOTA ( POOLED )
00000000 0108 413 .LONG 0 ; REMAINING QUOTA ( POOLED )
          010C 414
          04 010C 415 .BYTE LNMYSM_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$C_ORB ; BLOCK TYPE
          00 011B 423 .BYTE 0 ; NO ACL AS YET
00000000 011C 424 .LONG 0 ; ZERO RESERVED WORD & REF COUNT
00000000 0120 425 .QUAD 0 ; OBJECT DOES NOT HAVE AN ACCESS MODE
00000008 0128 426 .LONG ^X00000008 ; SYSTEM PROTECTION IS RWE
00000008 012C 427 .LONG ^X00000008 ; OWNER PROTECTION IS RWE
0000000E 0130 428 .LONG ^X0000000E ; GROUP PROTECTION IS R
0000000E 0134 429 .LONG ^X0000000E ; WORLD PROTECTION IS R
00000000 0138 430 .LONG 0,0 ; NULL INITIAL ACL
00'00'00'00'00'00'00'00'00'00'00'00'00'00'00'00'00'0140 431 .BYTE 0[ORB$S_MIN_CLASS] ; MINIMUM CLASSIFICATION MASK
          00'00'00'00'00'00'00'00'00'00'00'00'00'014C
00'00'00'00'00'00'00'00'00'00'00'00'00'00'00'00'00'0154 432 .BYTE 0[ORB$S_MAX_CLASS] ; MAXIMUM CLASSIFICATION MASK
          00'00'00'00'00'00'00'00'00'00'00'00'00'0160
          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$LOWUSE ; PAGED PSECT AT END OF SYS.EXE
0000 440
49 4E 49 53 59 53 00000008'010E0000' 0000 441 IMGDESC:.ASCID /SYSINIT.EXE/ ; SYSTEM INITIALIZATION PROCESS
      45 58 45 2E 54 000E
3A 30 41 50 4F 0000001B'010E0000' 0013 442 TTODESC:.ASCID /OPA0:/
      0020 443
      0020 444
      0020 445 ; DESCRIPTORS AND CHARACTER STRING BUFFERS FOR THE LOGICAL NAME TABLE NAMES,
      0020 446 ; LOGICAL NAMES, AND LOGICAL NAME EQUIVALENCE STRINGS THAT ARE CREATED AT
      0020 447 ; SYSTEM INITIALIZATION TIME.
      0020 448 ;
      0020 449
49 44 24 4D 4E 4C 00000028'010E0000' 0020 450 LNM_DIRECTORIES_DESC:
      53 45 49 52 4F 54 43 45 52 002E 451 .ASCID /LNMS$DIRECTORIES/
      0037 452
      0037 453 LNM_FILE_DEV_DESC:
49 46 24 4D 4E 4C 0000003F'010E0000' 0037 454 .ASCID /LNMS$FILE_DEV/
      56 45 44 5F 45 4C 0045
      004B 455
      004B 456 LNM_PERMANENT_MAILBOX_DESC:
45 50 24 4D 4E 4C 00000053'010E0000' 004B 457 .ASCID /LNMS$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 /LNMS$TEMPORARY_MAILBOX/
4C 49 41 4D 5F 59 52 41 52 4F 50 4D 0086
      58 4F 42 0092
      0095 469
      0095 470 LOG_G_DESC:
00000009' 0095 471 .LONG LOG_GROUP_LENGTH
000001AB' 0099 472 .ADDRESS LOG_GROUP
      009D 473
      009D 474 LOG_P_DESC:
0000000B' 009D 475 .LONG LOG_PROCESS_LENGTH
000001B4' 00A1 476 .ADDRESS LOG_PROCESS
      00A5 477
      00A5 478 LOG_S_DESC:
0000000A' 00A5 479 .LONG LOG_SYSTEM_LENGTH
000001BF' 00A9 480 .ADDRESS LOG_SYSTEM
      00AD 481
      00AD 482 SYS_DISK_DESC:
49 44 24 53 59 53 000000B5'010E0000' 00AD 483 .ASCID /SYS$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 487
00D2 488 TRNLOG_GS_DESC:
00D2 489 .ASCII /TRNLOG$_GROUP_SYSTEM/
47 4F 4C 4E 52 54 000000DA'010E0000' 00E0
54 53 59 53 5F 50 55 4F 52 47 5F 24 00EC
4D 45 00EE 490
00EE 491 TRNLOG_PG_DESC:
00EE 492 .ASCII /TRNLOG$_PROCESS_GROUP/
47 4F 4C 4E 52 54 000000F6'010E0000' 00FC
52 47 5F 53 53 45 43 4F 52 50 5F 24 0108
50 55 4F 010B 493
010B 494 TRNLOG_PS_DESC:
010B 495 .ASCII /TRNLOG$_PROCESS_SYSTEM/
47 4F 4C 4E 52 54 00000113'010E0000' 0119
59 53 5F 53 53 45 43 4F 52 50 5F 24 0125
4D 45 54 53 0129 496
0129 497 TRNLOG_PGS_DESC:
0129 498 .ASCII /TRNLOG$_PROCESS_GROUP_SYSTEM/
47 4F 4C 4E 52 54 00000131'010E0000' 0137
52 47 5F 53 53 45 43 4F 52 50 5F 24 0143
4D 45 54 53 59 53 5F 50 55 4F 014D 499
014D 500 LNM_GROUP:
014D 501 .ASCII /LNM$GROUP/
50 55 4F 52 47 24 4D 4E 4C 0156 502 LNM_GROUP_LENGTH = . - LNM_GROUP
00000009 0156 503
0156 504 LNM_JOB:
0156 505 .ASCII /LNM$JOB/
42 4F 4A 24 4D 4E 4C 015D 506 LNM_JOB_LENGTH = . - LNM_JOB
00000007 015D 507
015D 508 LNM_PROCESS:
015D 509 .ASCII /LNM$PROCESS/
53 53 45 43 4F 52 50 24 4D 4E 4C 0168 510 LNM_PROCESS_LENGTH = . - LNM_PROCESS
00000008 0168 511
0168 512 LNM_PROCESS_DIRECTORY:
0168 513 .ASCII /LNM$PROCESS_DIRECTORY/
5F 53 53 45 43 4F 52 50 24 4D 4E 4C 0174
59 52 4F 54 43 45 52 49 44 017D 514 LNM_PROCESS_DIRECTORY_LENGTH = . - LNM_PROCESS_DIRECTORY
00000015 017D 515
017D 516 LNM_SYSTEM:
017D 517 .ASCII /LNM$SYSTEM/
4D 45 54 53 59 53 24 4D 4E 4C 0187 518 LNM_SYSTEM_LENGTH = . - LNM_SYSTEM
0000000A 0187 519
0187 520 LNM_SYSTEM_DIRECTORY:
0187 521 .ASCII /LNM$SYSTEM_DIRECTORY/
44 5F 4D 45 54 53 59 53 24 4D 4E 4C 0193
59 52 4F 54 43 45 52 49 019B 522 LNM_SYSTEM_DIRECTORY_LENGTH = . - LNM_SYSTEM_DIRECTORY
00000014 019B 523
019B 524 LNM_SYSTEM_TABLE:
019B 525 .ASCII /LNM$SYSTEM_TABLE/
54 5F 4D 45 54 53 59 53 24 4D 4E 4C 01A7
45 4C 42 41 01AB 526 LNM_SYSTEM_TABLE_LENGTH = . - LNM_SYSTEM_TABLE
00000010 01AB 527
01AB 528 LOG_GROUP:
01AB 529 .ASCII /LOG$GROUP/
50 55 4F 52 47 24 47 4F 4C 01B4 530 LOG_GROUP_LENGTH = . - LOG_GROUP
00000009 01B4 531
01B4

```

```

53 53 45 43 4F 52 50 24 47 4F 4C 01B4 532 LOG_PROCESS:
0000000B 01B4 533 .ASCII /LOG$PROCESS/
01BF 534 LOG_PROCESS_LENGTH = . - LOG_PROCESS
01BF 535
01BF 536 LOG_SYSTEM:
4D 45 54 53 59 53 24 47 4F 4C 01BF 537 .ASCII /LOG$SYSTEM/
0000000A 01C9 538 LOG_SYSTEM_LENGTH = . - LOG_SYSTEM
01C9 539
01C9 540 :
01C9 541 : ATTRIBUTE, ACCESS MODE AND ITEM BUFFERS WHICH ARE PASSED BY REFERENCE.
01C9 542 :
01C9 543
00000001 01C9 544 EXEC_MODE: ; EXECUTIVE ACCESS MODE BUFFER
01CD 545 .LONG PSL$C_EXEC
01CD 546
00000000 01CD 547 KERNEL_MODE: ; KERNEL ACCESS MODE BUFFER
01CD 548 .LONG PSL$C_KERNEL
01D1 549
00000002 01D1 550 SUPER_MODE: ; SUPERVISOR ACCESS MODE BUFFER
01D1 551 .LONG PSL$C_SUPER
01D5 552
00000001 01D5 553 LNM_NO_ALIAS: ; NO_ALIAS ATTRIBUTE BUFFER
01D5 554 .LONG LNMSM_NO_ALIAS
01D9 555
00000200 01D9 556 TERMINAL_BUFFER: ; TERMINAL ATTRIBUTES ITEM BUFFER
01D9 557 .LONG LNMSM_TERMINAL
01DD 558
01DD 559 :
01DD 560 : ITEM LISTS FOR THE CREATION OF THE LOGICAL NAMES SETUP AT SYSTEM
01DD 561 : INITIALIZATION TIME.
01DD 562 :
01DD 563
0004 01DD 564 DIRECTORIES LIST: ; ITEM LIST FOR LNMSDIRECTORIES
0003 01DD 565 .WORD 4 ; TERMINAL ATTRIBUTES ITEM
000001D9 01DF 566 .WORD LNMS_ATTRIBUTES
00000000 01E1 567 .ADDRESS TERMINAL_BUFFER
01E5 568 .LONG 0
01E9 569
0015 01E9 570 .WORD LNM_PROCESS_DIRECTORY_LENGTH ; LNMSPROCESS_DIRECTORY STRING ITEM
0002 01EB 571 .WORD LNMS_STRING
00000168 01ED 572 .ADDRESS LNM_PROCESS_DIRECTORY
00000000 01F1 573 .LONG 0
01F5 574
0014 01F5 575 .WORD LNM_SYSTEM_DIRECTORY_LENGTH ; LNMS$SYSTEM_DIRECTORY STRING ITEM
0002 01F7 576 .WORD LNMS_STRING
00000187 01F9 577 .ADDRESS LNM_SYSTEM_DIRECTORY
00000000 01FD 578 .LONG 0
0201 579
00000000 0201 580 .LONG 0 ; END OF ITEM LIST
0205 581
000B 0205 582 FILE_DEV_SUPER_LIST: ; ITEM LIST FOR SUPERVISOR LNMSFILE_DEV
0002 0207 583 .WORD LNM_PROCESS_LENGTH ; LNMSPROCESS STRING ITEM
0000015D 0209 584 .WORD LNMS_STRING
00000000 020D 585 .ADDRESS LNM_PROCESS
0211 586 .LONG 0
0007 0211 587
0211 588 .WORD LNM_JOB_LENGTH ; LNMSJOB STRING ITEM

```

```

0002 0213 589 .WORD LNMS_STRING
00000156' 0215 590 .ADDRESS LNM_JOB
00000000 0219 591 .LONG 0
          021D 592
0009 021D 593 .WORD LNM_GROUP_LENGTH ; LNMSGROUP STRING ITEM
0002 021F 594 .WORD LNMS_STRING
0000014D' 0221 595 .ADDRESS LNM_GROUP
00000000 0225 596 .LONG 0
          0229 597
000A 0229 598 .WORD LNM_SYSTEM_LENGTH ; LNMSYSTEM STRING ITEM
0002 022B 599 .WORD LNMS_STRING
0000017D' 022D 600 .ADDRESS LNM_SYSTEM
00000000 0231 601 .LONG 0
          0235 602
00000000 0235 603 .LONG 0 ; END OF ITEM LIST
          0239 604
          0239 605 FILE_DEV_EXEC_LIST: ; ITEM LIST FOR EXECUTIVE LNMSFILE_DEV
          0239 606 PERMANENT_MAILBOX_LIST: ; ITEM LIST FOR LNMSPERMANENT_MAILBOX
000A 0239 607 .WORD LNM_SYSTEM_LENGTH ; LNMSYSTEM STRING ITEM
0002 023B 608 .WORD LNMS_STRING
0000017D' 023D 609 .ADDRESS LNM_SYSTEM
00000000 0241 610 .LONG 0
          0245 611
00000000 0245 612 .LONG 0 ; END OF ITEM LIST
          0249 613
          0249 614 LOG_G_LIST: ; ITEM LIST FOR LOG$GROUP
0009 0249 615 .WORD LNM_GROUP_LENGTH ; LNMSGROUP STRING ITEM
0002 024B 616 .WORD LNMS_STRING
0000014D' 024D 617 .ADDRESS LNM_GROUP
00000000 0251 618 .LONG 0
          0255 619
00000000 0255 620 .LONG 0 ; END OF ITEM LIST
          0259 621
          0259 622 LOG_P_LIST: ; ITEM LIST FOR LOG$PROCESS
000B 0259 623 .WORD LNM_PROCESS_LENGTH ; LNMSPROCESS STRING ITEM
0002 025B 624 .WORD LNMS_STRING
0000015D' 025D 625 .ADDRESS LNM_PROCESS
00000000 0261 626 .LONG 0
          0265 627
0007 0265 628 .WORD LNM_JOB_LENGTH ; LNMSJOB STRING ITEM
0002 0267 629 .WORD LNMS_STRING
00000156' 0269 630 .ADDRESS LNM_JOB
00000000 026D 631 .LONG 0
          0271 632
00000000 0271 633 .LONG 0 ; END OF ITEM LIST
          0275 634
          0275 635 LOG_S_LIST: ; ITEM LIST FOR LOG$SYSTEM
000A 0275 636 .WORD LNM_SYSTEM_LENGTH ; LNMSYSTEM STRING ITEM
0002 0277 637 .WORD LNMS_STRING
0000017D' 0279 638 .ADDRESS LNM_SYSTEM
00000000 027D 639 .LONG 0
          0281 640
00000000 0281 641 .LONG 0 ; END OF ITEM LIST
          0285 642
          0285 643 SYSTEM_LIST: ; ITEM LIST FOR LNMSYSTEM
0004 0285 644 .WORD 4 ; TERMINAL ATTRIBUTES ITEM
0003 0287 645 .WORD LNMS_ATTRIBUTES

```



```

00000009' 0289 646 .ADDRESS TERMINAL_BUFFER
00000000 028D 647 .LONG 0
          0291 648
          0010 0291 649 .WORD LNM_SYSTEM_TABLE_LENGTH ; LNM$SYSTEM_TABLE STRING ITEM
          0002 0293 650 .WORD LNM$_STRING
00000198' 0295 651 .ADDRESS LNM_SYSTEM_TABLE
00000000 0299 652 .LONG 0
          029D 653
00000000 029D 654 .LONG 0 ; END OF ITEM LIST
          02A1 655
          0007 02A1 656 TEMPORARY_MAILBOX_LIST: ; ITEM LIST FOR LNM$TEMPORARY_MAILBOX
          0002 02A3 657 .WORD LNM_JOB_LENGTH ; LNM$JOB STRING ITEM
          00000156' 02A5 658 .WORD LNM$_STRING
00000000 02A9 659 .ADDRESS LNM_JOB
          02AD 660 .LONG 0
          00000000 02AD 661 .LONG 0 ; END OF ITEM LIST
          02B1 662
          0009 02B1 663 TRNLOG_GS_LIST: ; ITEM LIST FOR TRNLOG$GROUP_SYSTEM
          0002 02B3 664 .WORD LOG_GROUP_LENGTH ; LOG$GROUP STRING ITEM
          000001AB' 02B5 665 .WORD LNM$_STRING
00000000 02B9 666 .ADDRESS LOG_GROUP
          02BD 667 .LONG 0
          000A 02BD 668 .WORD LOG_SYSTEM_LENGTH ; LOG$SYSTEM STRING ITEM
          0002 02BF 669 .WORD LNM$_STRING
000001BF' 02C1 670 .ADDRESS LOG_SYSTEM
00000000 02C5 671 .LONG 0
          02C9 672
00000000 02C9 673 .LONG 0 ; END OF ITEM LIST
          02CD 674
          000B 02CD 675 TRNLOG_PG_LIST: ; ITEM LIST FOR TRNLOG$PROCESS_GROUP
          0002 02CF 676 .WORD LOG_PROCESS_LENGTH ; LOG$PROCESS STRING ITEM
          000001B4' 02D1 677 .WORD LNM$_STRING
00000000 02D5 678 .ADDRESS LOG_PROCESS
          02D9 679 .LONG 0
          0009 02D9 680 .WORD LOG_GROUP_LENGTH ; LOG$GROUP STRING ITEM
          0002 02DB 681 .WORD LNM$_STRING
000001AB' 02DD 682 .ADDRESS LOG_GROUP
00000000 02E1 683 .LONG 0
          02E5 684
00000000 02E5 685 .LONG 0 ; END OF ITEM LIST
          02E9 686
          000B 02E9 687 TRNLOG_PS_LIST: ; ITEM LIST FOR TRNLOG$PROCESS_SYSTEM
          0002 02EB 688 .WORD LOG_PROCESS_LENGTH ; LOG$PROCESS STRING ITEM
          000001B4' 02ED 689 .WORD LNM$_STRING
00000000 02F1 690 .ADDRESS LOG_PROCESS
          02F5 691 .LONG 0
          000A 02F5 692 .WORD LOG_SYSTEM_LENGTH ; LOG$SYSTEM STRING ITEM
          0002 02F7 693 .WORD LNM$_STRING
000001BF' 02F9 694 .ADDRESS LOG_SYSTEM
00000000 02FD 695 .LONG 0
          0301 696
00000000 0301 697 .LONG 0 ; END OF ITEM LIST
          0305 700
          0301 701
          0305 702

```

```

000B 0305 703 TRNLOG_PGS_LIST: ; ITEM LIST FOR TRNLOG$ PROCESS_GROUP_SYSTEM
0002 0305 704 .WORD LOG_PROCESS_LENGTH ; LOG$PROCESS STRING ITEM
000001B4' 0307 705 .WORD LNMS_STRING
00000000 0309 706 .ADDRESS LOG_PROCESS
0311 707 .LONG 0
0311 708
0009 0311 709 .WORD LOG_GROUP_LENGTH ; LOG$GROUP STRING ITEM
0002 0313 710 .WORD LNMS_STRING
000001AB' 0315 711 .ADDRESS LOG_GROUP
00000000 0319 712 .LONG 0
031D 713
000A 031D 714 .WORD LOG_SYSTEM_LENGTH ; LOG$SYSTEM STRING ITEM
0002 031F 715 .WORD LNMS_STRING
000001BF' 0321 716 .ADDRESS LOG_SYSTEM
00000000 0325 717 .LONG 0
0329 718
00000000 0329 719 .LONG 0 ; END OF ITEM LIST
032D 720
032D 721 :
032D 722 : ARGUMENT LISTS FOR THE $CRELNMS. THIS SYSTEM SERVICES CAN NOT BE DIRECTLY
032D 723 : ISSUED AT SYSTEM INITIALIZATION BECAUSE THE SWAPPER DOES NOT HAVE A P1 SPACE
032D 724 : WITH SYSTEM SERVICE VECTORS; HOWEVER, IT MAYBE CALLED DIRECTLY. SETUP AN
032D 725 : ARGUMENT LIST FOR EACH AND EVERY DIRECT CALL.
032D 726 :
032D 727 :
032D 728 DIRECTORIES_ARG: ; ARGUMENT LIST FOR LNMS$DIRECTORIES
032D 729 $CRELNM -
032D 730 ACMODE = KERNEL_MODE, -
032D 731 ATTR = LNM_NO_ALIAS, -
032D 732 ITMLST = DIRECTORIES_LIST, -
032D 733 LOGNAM = LNM_DIRECTORIES_DESC, -
032D 734 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
0345 735
0345 736 FILE_DEV_EXEC_ARG: ; ARGUMENT LIST FOR EXECUTIVE LNMS$FILE_DEV
0345 737 $CRELNM -
0345 738 ACMODE = EXEC_MODE, -
0345 739 ITMLST = FILE_DEV_EXEC_LIST, -
0345 740 LOGNAM = LNM_FILE_DEV_DESC, -
0345 741 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
035D 742
035D 743 FILE_DEV_SUPER_ARG: ; ARGUMENT LIST FOR SUPERVISOR LNMS$FILE_DEV
035D 744 $CRELNM -
035D 745 ACMODE = SUPER_MODE, -
035D 746 ITMLST = FILE_DEV_SUPER_LIST, -
035D 747 LOGNAM = LNM_FILE_DEV_DESC, -
035D 748 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
0375 749
0375 750 LOG_G_ARG: ; ARGUMENT LIST FOR LOG$GROUP
0375 751 $CRELNM -
0375 752 ACMODE = KERNEL_MODE, -
0375 753 ITMLST = LOG_G_LIST, -
0375 754 LOGNAM = LOG_G_DESC, -
0375 755 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
038D 756
038D 757 LOG_P_ARG: ; ARGUMENT LIST FOR LOG$PROCESS
038D 758 $CRELNM -
038D 759 ACMODE = KERNEL_MODE, -

```

```

038D 760 ITMLST = LOG_P_LIST, -
038D 761 LOGNAM = LOG_P_DESC, -
038D 762 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
03A5 763
03A5 764 LOG_S_ARG: ; ARGUMENT LIST FOR LOG$SYSTEM
03A5 765 $CRELNM -
03A5 766 ACMODE = KERNEL_MODE, -
03A5 767 ITMLST = LOG_S_LIST, -
03A5 768 LOGNAM = LOG_S_DESC, -
03A5 769 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
038D 770
038D 771 PERMANENT_MAILBOX_ARG: ; ARGUMENT LIST FOR LNM$PERMANENT_MAILBOX
038D 772 $CRELNM -
038D 773 ACMODE = KERNEL_MODE, -
038D 774 ITMLST = PERMANENT_MAILBOX_LIST, -
038D 775 LOGNAM = LNM_PERMANENT_MAILBOX_DESC, -
038D 776 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
03D5 777
03D5 778 SYSTEM_ARG: ; ARGUMENT LIST FOR LNM$SYSTEM
03D5 779 $CRELNM -
03D5 780 ACMODE = KERNEL_MODE, -
03D5 781 ATTR = LNM_NO_ALIAS, -
03D5 782 ITMLST = SYSTEM_LIST, -
03D5 783 LOGNAM = LNM_SYSTEM_DESC, -
03D5 784 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
03ED 785
03ED 786 TEMPORARY_MAILBOX_ARG: ; ARGUMENT LIST FOR LNM$TEMPORARY_MAILBOX
03ED 787 $CRELNM -
03ED 788 ACMODE = KERNEL_MODE, -
03ED 789 ITMLST = TEMPORARY_MAILBOX_LIST, -
03ED 790 LOGNAM = LNM_TEMPORARY_MAILBOX_DESC, -
03ED 791 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
0405 792
0405 793 TRNLOG_GS_ARG: ; ARGUMENT LIST FOR TRNLOG$_GROUP_SYSTEM
0405 794 $CRELNM -
0405 795 ACMODE = KERNEL_MODE, -
0405 796 ITMLST = TRNLOG_GS_LIST, -
0405 797 LOGNAM = TRNLOG_GS_DESC, -
0405 798 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
041D 799
041D 800 TRNLOG_PG_ARG: ; ARGUMENT LIST FOR TRNLOG$_PROCESS_GROUP
041D 801 $CRELNM -
041D 802 ACMODE = KERNEL_MODE, -
041D 803 ITMLST = TRNLOG_PG_LIST, -
041D 804 LOGNAM = TRNLOG_PG_DESC, -
041D 805 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
0435 806
0435 807 TRNLOG_PS_ARG: ; ARGUMENT LIST FOR TRNLOG$_PROCESS_SYSTEM
0435 808 $CRELNM -
0435 809 ACMODE = KERNEL_MODE, -
0435 810 ITMLST = TRNLOG_PS_LIST, -
0435 811 LOGNAM = TRNLOG_PS_DESC, -
0435 812 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
044D 813
044D 814 TRNLOG_PGS_ARG: ; ARGUMENT LIST FOR TRNLOG$_PROCESS_GROUP_SY
044D 815 $CRELNM -
044D 816 ACMODE = KERNEL_MODE, -

```

```
044D 817 ITMLST = TRNLOG_PGS_LIST, -
044D 818 LOGNAM = TRNLOG_PGS_DESC, -
044D 819 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
0465 820
0000 0180 821 .PSECT $$$260 ; WRITABLE PSECT
0180 822 ; ITMLST MUST BE FOLLOWING TWO CRELNM
0180 823
0130 824 SYS_DISK_ARG: ; ARGUMENT LIST FOR SYS$DISK
0180 825 $CRELNM -
0180 826 ACMODE = EXEC_MODE, -
0180 827 LOGNAM = SYS_DISK_DESC, -
0180 828 TABNAM = LNM_SYSTEM_DESC
0198 829
0198 830 SYS_SYSDEVICE_ARG: ; ARGUMENT LIST FOR SYS$SYSDEVICE
0198 831 $CRELNM -
0198 832 ACMODE = EXEC_MODE, -
0198 833 LOGNAM = SYS_SYSDEVICE_DESC, -
0198 834 TABNAM = LNM_SYSTEM_DESC
```

```

0180      836
00000465 837      .PSECT YF$LOWUSE          ; PAGED PSECT AT END OF SYS.EXE
0465      838
0465      839
0465      840 : DEFINE A QUOTA LIST TO BE USED BY VARIOUS PIECES OF THE SYSTEM WHEN
0465      841 : CREATING A SPECIAL SYSTEM PROCESS, LIKE A FILES-11 ACP. EVERY QUOTA
0465      842 : IS MENTIONED EXPLICITLY. NOTE THAT THIS LIST CAN BE TAILORED BY
0465      843 : COPYING IT TO SOME TEMPORARY LOCATION AND APPENDING NEW QUOTA ITEMS
0465      844 : TO THE END OF THE LIST. THE $CREPRC SYSTEM SERVICE USES THE LAST
0465      845 : VALUE OF A SPECIFIED QUOTA IN THE LIST WHEN IT CREATES A PROCESS.
0465      846 : NOTE THAT THE END OF THE LIST MUST BE TERMINATED BY A ZERO BYTE,
0465      847 : AND THAT THE LENGTH OF THE LIST, AS GIVEN BY PQL$C_SYSPQLLEN, DOES NOT
0465      848 : INCLUDE THE LIST TERMINATOR.
0465      849 :
0465      850
0465      851 PQL$AB_SYSPQL::                ; SYSTEM PROCESS QUOTA LIST
01      0465 852      .BYTE PQL$_ASTLM      ; PROCESS AST LIMIT
0000000A 0466 853      .LONG 10
02      046A 854      .BYTE PQL$_BIOLM     ; PROCESS BUFFERED I/O LIMIT
0000000A 046B 855      .LONG 10
03      046F 856      .BYTE PQL$_BYTLM    ; PROCESS BUFFERED I/O BYTE LIMIT
00008000 0470 857      .LONG 32768
04      0474 858      .BYTE PQL$_CPULM    ; PROCESS CPU TIME LIMIT
00000000 0475 859      .LONG 0           ; ZERO IMPLIES NO LIMIT
05      0479 860      .BYTE PQL$_DIOLM    ; PROCESS DIRECT I/O LIMIT
0000000A 047A 861      .LONG 10
06      047E 862      .BYTE PQL$_FILLM    ; PROCESS OPEN FILE LIMIT
0000003C 047F 863      .LONG 60
07      0483 864      .BYTE PQL$_PGFLQUOTA ; PROCESS PAGE FILE QUOTA
00004E20 0484 865      .LONG 20000
08      0488 866      .BYTE PQL$_PRCLM    ; PROCESS SUBPROCESS CREATION LIMIT
00000008 0489 867      .LONG 8
09      048D 868      .BYTE PQL$_TQELM   ; PROCESS TIMER QUEUE ENTRY LIMIT
00000008 048E 869      .LONG 8
0B      0492 870      .BYTE PQL$_WSDEFAULT ; PROCESS DEFAULT WORKING SET SIZE
00000064 0493 871      .LONG 100
0A      0497 872      .BYTE PQL$_WSQUOTA  ; PROCESS WORKING SET QUOTA
000000C8 0498 873      .LONG 200
0D      049C 874      .BYTE PQL$_WSEXTENT ; PROCESS WORKING SET EXTENT LIMIT
000003E8 049D 875      .LONG 1000
0C      04A1 876      .BYTE PQL$_ENQLM   ; PROCESS LOCK LIMIT
00000064 04A2 877      .LONG 100
0E      04A6 878      .BYTE PQL$_JTQUOTA  ; JOB-WIDE LOGICAL NAME TABLE QUOTA
00000400 04A7 879      .LONG 1024
00      04AB 880 10$: .BYTE PQL$_LISTEND ; END OF PROCESS QUOTA LIST
04AC      881
00000046 04AC 882 PQL$C_SYSPQLLEN == 10$ - PQL$AB_SYSPQL ; LENGTH OF LIST (MINUS TERMINATOR)

```

```

04AC 885      .SBTTL EXESSWAPINIT - INITIALIZATION AND STARTUP FOR SWAPPER
04AC 886
04AC 887      :++
04AC 888      : FUNCTIONAL DESCRIPTION:
04AC 889      : EXESSWAPINIT IS ENTERED WHEN THE SWAPPER PROCESS IS FIRST
04AC 890      : SCHEDULED AFTER A SYSTEM BOOT/STARTUP. THIS TRANSFER OCCURS
04AC 891      : VIA THE INITIAL PC VALUE BUILT INTO THE HARDWARE PCB FOR THE
04AC 892      : SWAPPER PROCESS. R4 CONTAINS THE ADDRESS OF THE SWAPPER PCB.
04AC 893      :
04AC 894      :--
04AC 895
04AC 896 EXESSWAPINIT::      ; SWAPPER INITIALIZATION
04AC 897
04AC 898      : INITIALIZE PAGED POOL.
04AC 899
5B 00000000'GF  DO 04AC 900      MOVL    G^EXESGL_PAGED,R11      ; POINT TO START OF PAGED POOL
8B 00000000'GF  D4 04B3 901      CLRL    (R11)+      ; ZAP FORWARD LINK
6B 00000000'GF  DO 04B5 902      MOVL    G^SGN$GL_PAGEDYN,(R11) ; AND SET SIZE
04BC 903
04BC 904
04BC 905      : ALLOCATE LOGICAL NAME HASH TABLE. THE NUMBER OF ENTRIES IN THE HASH TABLE
04BC 906      : MUST BE A POWER OF TWO. SO THE ALLOCATED SIZE IS THE SMALLEST POWER OF
04BC 907      : TWO LARGER THAN THE SYSGEN PARAMETER.
04BC 908
00000000'FF  DD 04BC 909      PUSHL   @LNMSAL_HASHTBL      ; SAVE ADDR OF CRELNM ITMLST BLOCKS FOR
04C2 910      : "SYSSDISK" AND "SYSSSYSDEVICE"
57 00000000'GF  DO 04C2 911      MOVL    #1, R8      ; DO THIS TWICE
57 57 07 07 51  D4 04D6 915      CLRL    R1      ; CLEAR A REGISTER
00 51 57 51  D0 04D8 916      BBSS   R7,R1,50$   ; THE SIZE OF THE TABLE ROUNDED UP
00000000'GF  D0 04DC 917 50$: MOVL    R1,G^LNMSGL_HTBLSIZES[R8]; WRITE BACK THE CORRECT VALUE
DE 58 F4 04E4 918      SOBGEQ  R8,40$   ; LOOP TWO TIMES
04E7 919
04E7 920      : INITIALIZE THE SYSTEM SPACE HASH TABLE.
04E7 921
51 00000000'GF  DO 04E7 922      MOVL    G^LNMSGL_HTBLSIZES,R1 ; SIZE OF TABLE IN ENTRIES
51 00000000'GF  DE 04EE 923      MOVAL   @#LNMHSH$K_BUCKET[R1],R1; MULT BY 4 AND ADD OVERHEAD
00000000'GF  16 04F6 924      JSB     G^EXESALOPAGED      ; ALLOCATE MEMORY
62 51 00 00 8F 00 2C 04FE 926      MOVCS   #0,#0,#0,R1,(R2)    ; SAVE REGISTERS DESTROYED BY MOVCS
12 BA 0505 927      POPR    #^M<R1,R4>      ; RESTORE REGISTERS DESTROYED BY MOVCS
0507 928      : NOTE: THAT R2 COMES BACK AS R4
50 00000000'GF  C3 0507 929      SUBL3   #1,G^LNMSGL_HTBLSIZES,R0 ; CALC UPPER BOUND OF HASH INDEX
64 50 D2 050F 930      MCOML  R0,LMHSH$L_MASK(R4)    ; STORE HASH INDEX MASK
08 A4 51 B0 0512 931      MOVW   R1,LMHSH$W_SIZE(R4)    ; STORE SIZE IN STRUCTURE HEADER
0A A4 38 90 0516 932      MOVB   #DYN$C_RSHT,LMHSH$B_TYPE(R4)
00000000'FF  64 9E 051A 934      MOVAB  (R4),@LNMSAL_HASHTBL ; STORE STRUCTURE TYPE
0521 935      : STUFF WAY POINTER TO TABLE
0521 936      : NOTE: THAT THE HASH TABLE HAS BEEN
0521 937      : INITIALIZED TO ZERO
0521 938
0521 939      : FIX UP THE SYSTEM LOGICAL NAME DIRECTORY, AND INSERT IT IN INTO THE
0521 940      : APPROPRIATE HASH BUCKET OF THE SHAREABLE LOGICAL NAME HASH TABLE.
0521 941

```

```

53 00000000'EF 9E 0521 942
0000002C'EF 64 DE 0521 943
                                MOVAB LNMS$SYSTEM DIRECTORY,R3
                                MOVAL (R4),LNMS$SYSTEM_DIR_LNMTH+LNMTM$HASH
                                ; HASH TABLE ADDRESS IN LNMTM
                                ; GET SIZE OF DIRECTORY NAME
                                MOVZBL LNMB$T_NAME(R3),R0
                                ; GET ADDRESS OF DIRECTORY NAME
                                MOVAB LNMB$T_NAME+1(R3),R1
                                ; HASH THE DIRECTORY NAME
                                JSB LNMS$HASH
                                ; MODIFY THE INDEX TO BE IN RANGE
                                BICL2 LNMSH$SL_MASK(R4),R0
                                MOVAL (R3),LNMSH$C_BUCKET(R4)[R0]
                                ; INSERT DIRECTORY LNMB IN HASH TABLE
                                MOVAL LNMSH$C_BUCKET(R4)[R0],LNMB$SL_BLINK(R3)
                                ; INSERT HASH TBL ENTRY IN DIRECT LNMB
                                052F 945
                                052F 946
                                0533 947
                                0537 948
                                CA 053D 949
                                DE 0540 950
                                0545 951
                                DE 0545 952
                                054B 953
                                054B 954
                                054B 955
                                054B 956
                                054B 957
                                054B 958
                                054B 959
                                :
                                : FIXUP THE SYSTEM LOGICAL NAME TABLE, LNMS$SYSTEM TABLE, AND INSERT IT INTO THE
                                : APPROPRIATE HASH BUCKET OF THE SYSTEM LOGICAL NAME HASH TABLE.
                                :
                                51 000000C0'EF 9E 054B 960
                                00000000'FF D0 0552 961
                                000000E8'EF 0558 962
                                055D 963
                                D4 055D 964
                                16 055F 965
                                JSB G^LNMS$INSLOGTAB
                                ; RETRIEVE SYSTEM TABLE LNMB ADDRESS
                                ; MOVE THE ADDRESS OF THE SHAREABLE
                                ; LOGICAL NAME HASH TABLE INTO THE
                                ; SYSTEM TABLE'S TABLE HEADER
                                ; NO SPECIAL INSERTION ATTRIBUTES
                                ; APPROPRIATELY INSERT LNMS$SYSTEM_TABLE
                                0565 966
                                0565 967
                                0565 968
                                0565 969
                                0565 970
                                0565 971
                                0565 972
                                CALLG - ; CREATE LNMS$DIRECTORIES
                                DIRECTORIES ARG, -
                                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                056E 974
                                056E 975
                                056E 976
                                CALLG - ; CREATE EXECUTIVE LNMS$FILE_DEV
                                FILE DEV EXEC ARG, -
                                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                0577 978
                                0577 979
                                0577 980
                                CALLG - ; CREATE SUPERVISOR LNMS$FILE_DEV
                                FILE DEV SUPER ARG, -
                                @#SYS$CRELNM-PTSYSVECTORS+^X80000000
                                0580 982
                                0580 983
                                0580 984
                                CALLG - ; CREATE LOG$GROUP
                                LOG G ARG, -
                                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                0589 986
                                0589 987
                                0589 988
                                CALLG - ; CREATE LOG$PROCESS
                                LOG P ARG, -
                                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                0592 990
                                0592 991
                                0592 992
                                CALLG - ; CREATE LOG$SYSTEM
                                LOG S ARG, -
                                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                0598 994
                                0598 995
                                0598 996
                                CALLG - ; CREATE LNMS$PERMANENT_MAILBOX
                                PERMANENT MAILBOX ARG, -
                                @#SYS$CRELNM-P1SYSVECTORS+^X80000000
                                80000000'9F FDC4 CF FA 0598 997
                                05A4 998

```

```

      80000000'9F  FE2D CF  FA 05A4 999
      80000000'9F  FE3C CF  FA 05A4 1000
      80000000'9F  FE4B CF  FA 05A4 1001
      80000000'9F  FE5A CF  FA 05AD 1002
      80000000'9F  FE69 CF  FA 05AD 1003
      80000000'9F  FE78 CF  FA 05AD 1004
      00000194'EF  54 8ED0 05AD 1005
      000001AC'EF  64 DE 05AD 1006
      00000180'EF  0000'C4 DE 05B6 1007
      00000198'EF  51 00'8F 9A 05B6 1008
      00000000'EF  50 54 DO 05B6 1009
      00000000'EF  16 05BF 1010
      05DA 1011
      05DA 1012
      05DA 1013
      05DA 1014
      05DA 1015
      05DA 1016
      05DA 1017
      05DA 1018
      05DA 1019
      05DA 1020
      05DA 1021
      05DA 1022
      05DA 1023
      05DA 1024
      05DA 1025
      05DA 1026
      05DA 1027
      05DA 1028
      05DD 1029
      05E4 1030
      05E4 1031
      05ED 1032
      05ED 1033
      05ED 1034
      05ED 1035
      05F8 1036
      05F8 1037
      05F8 1038
      05F8 1039
      0603 1040
      0603 1041
      0603 1042
      0607 1043
      060A 1044
      0610 1045
      0610 1046
      0610 1047
      0610 1048
      0610 1049
      0610 1050
      0610 1051
      0610 1052
      0610 1053
      0610 1054
      0610 1055

CALLG - ; CREATE LNM$SYSTEM
      SYSTEM ARG, -
      @#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG - ; CREATE LNM$TEMPORARY_MAILBOX
      TEMPORARY_MAILBOX ARG, -
      @#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG - ; CREATE TRNLOG$_GROUP_SYSTEM
      TRNLOG GS ARG, -
      @#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG - ; CREATE TRNLOG$_PROCESS_GROUP
      TRNLOG PG ARG, -
      @#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG - ; CREATE TRNLOG$_PROCESS_SYSTEM
      TRNLOG PS ARG, -
      @#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG - ; CREATE TRNLOG$_PROCESS_GROUP_SYSTEM
      TRNLOG PGS ARG, -
      @#SYS$CRELNM-P1SYSVECTORS+^X80000000

:
: CREATE TWO STARTUP LOGICAL NAMES.
:
:
POPL R4 ; RECOVER ADDR OF CRELOG BLOCK
MOVAL (R4),SYS_DISK_ARG+CRELNM$_ITMLST
: STUFF THE ADDRESS OF THE ITEM LIST
MOVAL BDL$_SYSDLOG(R4),SYS_SYSDEVICE_ARG+CRELNM$_ITMLST
: STUFF THE ADDRESS OF THE ITEM LIST

CALLG - ; CREATE SYS$DISK
      SYS_DISK ARG, -
      @#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG - ; CREATE SYS$SYSDEVICE
      SYS_SYSDEVICE ARG, -
      @#SYS$CRELNM-P1SYSVECTORS+^X80000000

MOVZBL #BDL$_CRELNM_ITMLST,R1 ; GET THE SIZE OF THE STRUCTURE
MOVL R4,R0 ; MOVE STRUCTURE ADDR INTO CORRECT REG
JSB EXE$DEANONPGDSIZ ; RETURN THE MEMORY

:
: CREATE INITIAL PROCESSES
:
: THE $CREPRC S MACRO CANNOT BE USED BECAUSE THAT MACRO GENERATES A
: CALL THROUGH THE P1 SYSTEM SERVICE VECTOR PAGES AND THE SWAPPER DOES
: NOT HAVE A P1 SPACE. THE SENSE OF THE CREATE PROCESS CALL IS THE
: FOLLOWING.
:
: $CREPRC_S INPUT=TTODESC,- ;

```



```

                                0610 1056 :
                                0610 1057 :
                                0610 1058 :
                                0610 1059 :
                                0610 1060 :
                                0610 1061 :
                                0610 1062 :
                                DD 0610 1063
00080020 7E D4 0612 1064
                                DD 0614 1065
                                DD 061A 1066
                                7E 7C 061C 1067
                                DD 061E 1068
F9EF CF 7F 0620 1069
                                DD 0624 1070
                                DD 0626 1071
F9D4 CF 7F 0628 1072
                                DD 062C 1073
80000000'9F OC FB 062E 1074
00000000'GF 17 0635 1075

                                OUTPUT=TTODESC,-;
                                ERROR=TTODESC,-;
                                IMAGE=IMGDESC,-;
                                UIC=#^X80020,-;
                                STSFLG=#<PRCSM_NOACNT!PRCSM_SSRWAIT>,-;
                                BASPRI=#2;

                                PUSHL #<PRCSM_NOACNT!PRCSM_SSRWAIT>
                                CLRL -(SP)
                                PUSHL #^X80020
                                PUSHL #2
                                CLRQ -(SP)
                                PUSHL #0
                                PUSHAQ TTODESC
                                PUSHL (SP)
                                PUSHL (SP)
                                PUSHAQ IMGDESC
                                PUSHL #0
                                CALLS #12,@#SYSS$CREPRC-P1SYSVECTORS+^X8000000
                                JMP G^LOOP ; JUMP OFF TO THE MAIN LOOP

```

```

063B 1078      .SBTTL SWAPPER - MAIN LOOP
063B 1079
063B 1080      :++
063B 1081      : FUNCTIONAL DESCRIPTION:
063B 1082      : THE MAIN LOOP OF THE SWAPPER IS EXECUTED WHENEVER THE SWAPPER IS AWAKENED
063B 1083      : FOR ANY REASON. EACH OF THE FUNCTIONAL ROUTINES WILL CHECK TO SEE IF
063B 1084      : THEY HAVE ANY ACTION TO PERFORM.
063B 1085      :--
063B 1086
00000000 1087
2E 10 0000 1088 LOOP: .PSECT $AEXENONPAGED ; NON-PAGED PSECT
FFFB' 30 0002 1089 BSBW BALANCE ; BALANCE FREE PAGE COUNT
0088 30 0005 1090 BSBW MMG$WRTMFYPAG ; WRITE MODIFIED PAGES
0000'CF D5 0008 1091 BSBW SWAPSCHEM ; SCHEDULE SWAP
06 13 000C 1092 TSTL W^EXESGL_PFATIM ; CHECK FOR POWER FAIL TIME
00000000'EF 16 000E 1093 JSB EXESPOWERAST ; GIVE ANY REQUIRED POWER FAIL ASTS
54 0000'CF D0 0014 1094 15$: MOVL W^SCH$GL_CURPCB,R4 ; GET PROPER PCB ADDRESS
52 0000'CF 7E 0019 1095 MOVAQ W^SCH$GQ_HIBWQ,R2 ; AND ADDRESS OF WAIT QUEUE HEADER
001E 1096 SETIPL #IPL$ SYNCH ; BLOCK SYSTEM EVENTS WHILE CHECKING
05 24 A4 0C E4 0021 1097 BBSC #PCBS$WAKEPEN,PCBS$L_STS(R4),20$ ; TEST AND CLEAR WAKE PENDING
00 DD 0026 1098 PUSHL #0 ; NULL PSL
FFD5' 30 0028 1099 BSBW SCH$WAITK ; WAIT WITH STACK CLEAN
DO 11 002E 1101 20$: SETIPL #0 ; DROP IPL
0030 1102 BRB LOOP ; CHECK FOR WORK TO DO
.DISABLE LSB

```

```

      0030 1105 .SBTTL BALANCE FREE PAGE COUNT
      0030 1106
      0030 1107 :++
      0030 1108 : FUNCTIONAL DESCRIPTION:
      0030 1109 : BALANCE WILL ENSURE THAT THE FREE PAGE LIST HAS AT LEAST THE NUMBER OF
      0030 1110 : PAGES SPECIFIED BY THE PARAMETER FREELIM. IF NOT, PAGES WILL BE MADE
      0030 1111 : AVAILABLE BY EITHER WRITING MODIFIED PAGES OR OUTSWAPPING PROCESSES.
      0030 1112 : IF SUFFICIENT FREE PAGES ARE AVAILABLE, THEN A CHECK IS MADE FOR
      0030 1113 : DELETED PROCESS HEADERS IN NEED OF CLEANUP.
      0030 1114 :--
      0030 1115
      0030 1116 BALANCE:
      0030 1117 Cmpl W^SGN$GL_FREELIM,W^SCH$GL_FREECNT ; BALANCE FREE PAGE COUNT
53 0000'CF 0000'CF 0A 15 0037 1118 BLEQ 5$ ; ARE WE HERE DUE TO FREELIM?
      0039 1119 SUBL3 W^SGN$GL_FREEGOAL,W^SCH$GL_FREECNT,R3 ; BRANCH IF NOT
      0041 1120 BLSS 20$ ; SUFFICIENT FREE PAGES?
      0043 1121 5$: TSTW W^SCH$GW_DELPHDCT ; NO, MUST ACQUIRE SOME
      0047 1122 BEQL 10$ ; CHECK FOR DELETED PROCESS HEADERS
      0049 1123 CLRL R3 ; NONE, EXIT
      004B 1124 BRB 25$ ; INDICATE NO FREE PAGES NEEDED
      004D 1125 10$: RSB ; IN BALANCE, RETURN
      004E 1126 20$:
50 17 0000'CF 00' E0 004E 1127 BBS S^#SCH$V_MPW,W^SCH$GB_SIP,25$ ; MODIFIED PAGE WRITING ACTIVE
      0000'CF 0000'CF 0D 15 0054 1128 SUBL3 W^SCH$GL_MFYLOLIM,W^SCH$GL_MFYCNT,R0 ; HOW MUCH WILL WRITING PAGES
      50 53 005C 1129 BLEQ 25$ ; NONE, MUST OUTSWAP
      005E 1130 ADDL R3,R0 ; YIELD RELATIVE TO WHAT WE NEED?
      0061 1131 BLSS 25$ ; NOT ENOUGH, MUST OUTSWAP
      0000'CF 0000'CF 08 19 0061 1131 BLSS 25$ ; NOT ENOUGH, MUST OUTSWAP
      0063 1132 MOVL W^SCH$GL_MFYLOLIM,W^SCH$GL_MFYLIM ; TRIGGER MODIFIED PAGE WRITING
      006A 1133 RSB ; AND EXIT TO LET IT HAPPEN
      006B 1134 :
      006B 1135 : MUST OUTSWAP, FIRST CHECK FOR SWAP IN PROGRESS SINCE SWAPPER IS
      006B 1136 : NOT RE-ENTRANT. IF PURGING DELETED HEADERS, THEN THE NUMBER OF
      006B 1137 : REQUIRED PAGES (IN R3) WILL BE SET TO ZERO. AN INFINITE INSWAP PRIORITY
      006B 1138 : WILL BE ASSUMED WHILE BALANCING THE NUMBER FO FREE PAGES.
      006B 1139 :
      DC 0000'CF 00' E2 006B 1140 25$: BBSS S^#SCH$V_SIP,W^SCH$GB_SIP,10$ ; EXIT IF SWAPPER ALREADY BUSY
      0071 1141 SETIPL #IPL$ SYNCH ; BLOCK SYSTEM EVENTS
      3FC0 8F BB 0074 1142 PUSHR #^M<R6,R7,R8,R9,R10,R11,AP,FP> ; SAVE NON-STANDARD REGISTERS
      0000'CF 94 0078 1143 CLRB W^SWP$GB_ISWPRI ; SET PRIORITY FOR SWAP SCHEDULE
      5D 53 D0 007C 1144 MOVL R3,FP ; GET AND TEST FREE PAGE DEFICIT
      0018'CF 08 18 007F 1145 BGEQ 30$ ; NONE, PURGING DELETED HEADERS
      0081 1146 TSTW W^SWP$GW_BALCNT ; CHECK FOR SINGULAR BALANCE SET
      0085 1147 BNEQ 30$ ; NO, CAN OUTSWAP
      5D D4 0087 1148 CLRL FP ; PREVENT OUTSWAP SCHEDULE
      008A 31 0089 1149 30$: BRW OUTSWAP ; TRY TO FORCE AN OUTSWAP

```

```

008C 1152      .SBTTL SCHEDULE SWAP
008C 1153
008C 1154      :++
008C 1155      : FUNCTIONAL DESCRIPTION:
008C 1156      : SWAPSCHEID IS CALLED BY THE MAIN LOOP OF THE SWAPPER PROCESS TO CHECK
008C 1157      : ELIGIBLE INSWAP CANDIDATES AND TO PROVIDE MEMORY NEEDED FOR THEIR
008C 1158      : INSWAP. A QUICK EXIT WILL BE TAKEN IF THE SWAPPER IS ALREADY BUSY.
008C 1159      : NO OUTSWAP WILL BE NEEDED IF THE NUMBER OF REQUIRED PAGES CAN BE
008C 1160      : TAKEN FROM THE FREE PAGE LIST LEAVING AT LEAST FREELIM STILL FREE.
008C 1161      : OTHERWISE OUTSWAP WILL BE ENTERED TO MAKE PAGES AVAILABLE BY ANY
008C 1162      : MEANS NECESSARY.
008C 1163      :--
008C 1164
008C 1165 QEMPTY: BUG_CHECK QUEUEEMPTY,FATAL      ; EMPTY QUEUE OR NOT A PCB
0090 1166
0090 1167 SWAPSCHEID:
0090 1168      SETIPL #IPL$ SYNCH      ; BLOCK SYSTEM EVENTS
52  OF 0000'CF 00' E2 0093 1169      BBSS S^SCH$V_SIP,W^SCH$GB_SIP,5$ ; EXIT IF SWAP IN PROGRESS
   0000'CF 20 00 EA 0099 1170      FFS #0,#32,W^SCH$GL_COMOQS,R2; FIND HIGHEST PRIORITY QUEUE
   00 0000'CF 00' 0A 12 00A0 1171      BNEQ 10$ ; FOUND ONE
   00 0000'CF 00' E5 00A2 1172      BBCC S^SCH$V_SIP,W^SCH$GB_SIP,5$ ; CLEAR SWAP IN PROGRESS
   00A8 1173 5$: SETIPL #0 ; DROP IPL
   00AB 1174      RSB ; AND RETURN
   00AC 1175
   00AC 1176 10$:
53  3FC0 8F BB 00AC 1177      PUSHR #M<R6,R7,R8,R9,R10,R11,AP,FP> ; SAVE REGS OTHER THAN R0-R5
   0000'CF42 7E 00B0 1178      MOVAQ W^SCH$AQ_COMOH[R2],R3 ; COMPUTE ADDRESS OF QUEUE HEADER
   54 63 D0 00B6 1179      MOVL (R3),R4 ; GET PCB ADDRESS
   00B9 1180
   00B9 1181 : THE FOLLOWING CHECK IS NEEDED DUE TO THE ODIOS MISLEADING SYMPTOMS THAT
   00B9 1182 : MIGHT OTHERWISE RESULT.
   00B9 1183
   0A A4 OC 91 00B9 1184      CMPB #DYN$C_PCB,PCB$B_TYPE(R4); IS THIS A GOOD PCB?
   CD 12 00BD 1185      BNEQ QEMPTY ; BUGCHECK IF NOT
   00BF 1186
   00BF 1187 : DETERMINE THE SIZE OF THE INSWAP CANDIDATE, TAKING INTO ACCOUNT THE FACT
   00BF 1188 : THAT THE PROCESS HEADER MIGHT ALREADY BE RESIDENT.
   00BF 1189
   50 36 A4 3C 00BF 1190      MOVZWL PCB$W_PPGCNT(R4),R0 ; COUNT OF PROCESS PAGES
   5A 34 A4 3C 00C3 1191      MOVZWL PCB$W_GPGCNT(R4),R10 ; COUNT OF GLOBAL PAGES
   07 24 A4 12 E1 00C7 1192      ADDL R0,R10 ; SUM PAGE COUNTS
   50 30 A4 3C 00CA 1193      BBC #PCB$V_PHDRES,PCB$L_STS(R4),15$ ; CONTINUE IF HEADER NON-RESIDENT
   5A 50 C2 00CF 1194      MOVZWL PCB$W_APTCNT(R4),R0 ; GET ACTIVE PAGE TABLE COUNT
   00D3 1195      SUBL R0,R10 ; SUBTRACT RESIDENT HEADER PAGES FROM REQUIR
50 0000'CF 0000'CF C3 00D6 1196 15$:
   0000'CF 0B A4 90 00DE 1197      SUBL3 W^SCH$GL_FREELIM,W^SCH$GL_FREECNT,R0 ; COMPUTE PAGES AVAILABLE
   5D 50 5A C3 00E4 1198      MOVB PCB$B_PRI(R4),W^SWP$GB_ISWPRI ; SAVE PRIORITY OF INSWAP
   001A'CF 0000'CF 0A 19 00E8 1200      SUBL3 R10,R0,FP ; WILL PROCESS FIT?
   0441 31 B0 00EA 1201      BLSS 20$ ; NO, MUST OUTSWAP
   00F1 1202      MOVW W^SCH$GW_SWPFAIL,W^SCH$GW_SWPFCNT ; RESET FAILURE COUNTER
   00F4 1203      BRW INSWAP ; YES PERFORM SWAP
   00F4 1204
   00F4 1205 : IF INSWAPPING A NON-REAL TIME PROCESS, THEN CHECK TO SEE IF ITS CURRENT
   00F4 1206 : PRIORITY IS THE DEFAULT BACKGROUND PRIORITY. IF SO, THEN DELAY AT LEAST
   00F4 1207 : SWAPRATE INTERVAL SINCE THE LAST INSWAP. THE EFFECT WILL BE TO AVOID FILLING
   00F4 1208 : THE BALANCE SET WITH CRUNCHING PROCESSES IMMEDIATELY.

```

```

          OB A4 10 91 00F4 1209 20$:
          1C 14 00F4 1210
51 1F 0000'CF 83 00F8 1211
          51 52 91 00FA 1212
          11 19 0100 1213
          51 11 0103 1214
51 00000000'EF D0 0105 1215
0000'CF 51 D1 010C 1216
          03 1A 0111 1217
          08CB 31 0113 1218
          0116 1219 40$:

```

```

CMPB #16,PCBSB_PRI(R4) : SCHEDULE OUTSWAP
BGTR 40$ : IS THIS A REAL TIME PROCESS?
SUBB3 W^SYSSGB_DEFPRI,#31,R1 : BR IF SO
CMPB R2,R1 : CONVERT PRIORITY TO INTERNAL FORM
BLSS 40$ : IS THIS A CRUNCHER OR LOW PRIORITY JOB?
MOVL EXE$GQ,SYSIME,R1 : BR IF NOT
CMPL R1,W^SWP$GL_SWTIME : GET CURRENT TIME IN APPROX. 10MS UNITS
BGTRU 40$ : HAS INTERVAL ELAPSED?
BRW SWAPEXIT : BR IF YES
: CAN'T DO SWAP NOW
:

```

0116 1222  
0116 1223  
0116 1224  
0116 1225  
0116 1226  
0116 1227  
0116 1228  
0116 1229  
0116 1230  
0116 1231  
0116 1232  
0116 1233  
0116 1234  
0116 1235  
0116 1236  
0116 1237  
0116 1238  
0116 1239  
0116 1240  
0116 1241  
0116 1242  
011E 1243  
0121 1244  
0126 1245  
0128 1246  
012B 1247  
0131 1248  
0133 1249  
0139 1250  
013D 1251  
013F 1252  
0143 1253  
0145 1254  
0147 1255  
0149 1256  
014C 1257  
014F 1258  
0153 1259  
0155 1260  
0158 1261  
015A 1262  
015C 1263  
015E 1264  
015E 1265  
015E 1266  
015E 1267  
015E 1268  
015E 1269  
015E 1270  
0161 1271  
0163 1272  
0165 1273  
0168 1274  
016B 1275  
016B 1276  
016B 1277  
016B 1278

.SBTTL OUTSWAP

SCHEDULE AND PERFORM OUTSWAPS IF POSSIBLE

FUNCTIONAL DESCRIPTION:

THE OUTSWAP STRATEGY IS TO FREE PROCESS HEADERS FOR OUTSWAP PROCESSES,  
USE AVAILABLE MODIFIED PAGES (AFTER WRITING THEM) AND FINALLY AS A LAST  
RESORT OUTSWAP ANOTHER PROCESS. ONLY ONE OF THESE ACTIONS WILL BE TAKEN  
AT A TIME THEN THE SCHEDULING SITUATION WILL BE RE-EVALUATED. THE VALUE  
IN FP INDICATES THE SIZE OF THE PAGE DEFICIT AND WILL BE SET POSITIVE IF  
ENTERED FROM BALANCE TO FREE DELETED PROCESS HEADERS.

INPUT:

FP - NEGATIVE VALUE WILL PERMIT PROCESS OUTSWAP  
ZERO OR POSITIVE WILL PURGE HEADERS ONLY.

OUTSWAP:

SUBL3 #1,SGN\$GL\_BALSETCT,R8 ; TRY TO OUTSWAP  
MCOML #0,R9 ; INIT INDEX FOR BALANCE SLOT SCAN  
10\$: TSTW @W^PHV\$GL\_REFCBAS[R8] ; INDICATE NO FREE LIST PURGE CANDIDATE  
BNEQ 12\$ ; IS SLOT IN NEED OF CLEANUP?  
BRW 60\$ ; CONTINUE IF NOT RELEASABLE  
12\$: CVTWL @W^PHV\$GL\_PIXBAS[R8],R4 ; GO RELEASE PAGE TABLES AND HEADER  
BLEQ 15\$ ; GET PROCESS INDEX  
MOVL @W^SCH\$GL\_PCBVEC[R4],R4 ; DELETED PROCESS OR VACANT SLOT  
BLBS PCB\$L\_STSTR4),20\$ ; GET PCB ADDRESS FOR PIX  
15\$: BEQL 20\$ ; SKIP IF PROCESS IS RESIDENT  
TSTB W^EXE\$GQ\_SYSTIME ; VACANT SLOT  
BEQL 17\$ ; ADD 1 IN 8 RANDOMNESS TO DECISION  
TSTL R9 ; BRANCH ON LOW PROBABILITY  
BGEQ 20\$ ; CHECK FOR REMEMBERED INDEX  
17\$: MOVL R8,R9 ; YES DONT OVERWRITE  
20\$: SOBGEQ R8,10\$ ; SAVE BALANCE SLOT NUMBER OF CANDIDATE  
TSTB W^EXE\$GQ\_SYSTIME+1 ; TRY ALL SLOTS  
BEQL 21\$ ; ADD 1 IN 256 RANDOMNESS TO DECISION  
MOVL R9,R8 ; BRANCH ON VERY LOW PROBABILITY  
BGEQ 24\$ ; GET AND TEST SLOT INDEX FOR SECONDARY CAND  
21\$: TSTL FP ; BR IF SLOT FOR CLEANUP  
BGEQ 22\$ ; CHECK FOR HEADER PURGE  
; EXIT IF SO

; SINCE THERE WAS NO HEADER TO FREE, WE MUST NOW WRITE MODIFIED PAGES OR OUTSWAP  
; SOME PROCESS. ONLY IF MODIFIED PAGES (MFYCNT-LOLIM) WILL TOTALLY SATISFY OUR  
; NEEDS WILL THEY BE WRITTEN. OTHERWISE THE LEAST USEFUL (BY SOME OPINION) PROCESS  
; WILL BE OUTSWAPPED AND THE SITUATION RECONSIDERED.

BSBW SCH\$OSWPSCHED ; SCHEDULE OUTSWAP  
TSTL R4 ; ANY CANDIDATE?  
BNEQ 23\$ ; YES  
BRW SWAPEXIT ; ELSE EXIT AND TRY LATER  
22\$: BRW 70\$ ; GO DO OUTSWAP  
23\$:

; A HEADER SLOT IN NEED OF CLEANUP WAS FOUND, NOW SCAN THE FREELIST FOR ALL  
; PAGES WHOSE PTE BACK POINTERS PLACE THEM WITHIN THIS HEADER. DELETE THE  
; CONTENT OF THOSE PAGES VIA MMG\$DELCONPFN TO FINALLY REDUCE THE REFERENCE

58 0000000'EF 01 C3  
59 00 D2  
0000'DF48 B5  
03 12  
00D1 31  
54 0000'DF48 32  
0A 15  
54 0000'DF44 D0  
OF 24 A4 E8  
0D 13  
0000'CF 95  
04 13  
59 D5  
03 18  
59 58 D0  
D2 58 F4  
0001'CF 95  
05 13  
58 59 D0  
11 18  
5D D5  
07 18  
FE9F' 30  
54 D5  
03 12  
0879 31  
0094 31

```

016B 1279 ; COUNT BINDING THE HEADER.
016B 1280 ;
57 0000'CF 02 9C 016B 1281 24$: ROTL #2,W^SWP$GL_BSLLOTSZ,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 PAGTBLPTE
04 BE41 9F 018B 1288 PUSHAB @4(SP)[R1] ; ANS SAVE PTE HIGH LIMIT
018F 1289
018F 1290 ;
018F 1291 ;
018F 1292 ;
018F 1293 ;
018F 1294 ;
018F 1295 ;
018F 1296 ;
018F 1297 ;
018F 1298 ASSUME PFNSC_FREPAGLST EQ 0 ;
018F 1299 ASSUME PFNSC_MFYPAGLST EQ 1 ;
018F 1300 ASSUME PFNSC_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 #PFNSC_BADPAGLST,R7 ; INITIALIZE LOOP SCAN TO BADPAGLST
50 0000'CF47 D0 019B 1305 25$: MOVL W^PFNS$AL_HEAD[R7],R0 ; GET HEAD OF LIST TO START SCAN
3E 13 01A1 1306 BEQL 45$ ; NO PAGES, DONE
01A3 1307 30$: PFN REFERENCE -
01A3 1308 MOVZWL <@W^PFNS$AL_FLINK[R0],R9>,- ; GET FORWARD LINK
01A3 1309 LONG OPCODE=MOVL,-
01A3 1310 IMAGE=SYS_NONPAGED
53 0000'DF40 D0 01A9 1311 MOVL @W^PFNS$AL_PTE[R0],R3 ; GET SVA OF PTE FOR PAGE
01AF 1312 ASSUME PFNSC_PPGTBL EQ 4
01AF 1313 ASSUME PFNSC_GPGTBL EQ 5
51 0000'DF40 01 02 EF 01AF 1314 EXTZV #2,#1,@W^PFNS$AB_TYPE[R0],R1 ; GET PAGE TABLE BIT
08 AE41 53 D1 01B7 1315 CMPL R3,8(SP)[R1] ; COMPARE WITH LOW LIMIT
6E41 53 1F 01BC 1316 BLSSU 40$ ; OUT OF RANGE
1E 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 @PFNS$AL_BAK[R0],R2 ; GET BACKING STORE ADDRESS
09 52 16 E0 01CF 1321 BBS #PTE$V_TYPO,R2,40$ ; LEAVE MODIFIED SECTION PAGES
52 57 D0 01D3 1322 35$: MOVL R7,R2 ; SET LIST NUMBER FOR DELETE
FE27' 30 01D6 1323 BSBW MMGSREMPFN ; REMOVE PAGE FROM FREE LIST
0895 30 01D9 1324 BSBW RELDELPAGE ; RELEASE PAGE DELETING CONTENT
50 59 D0 01DC 1325 40$: 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$GL_REF CBAS[R8] ; DID WE FREE PROCESS HEADER
0E 13 01EC 1330 BEQL 60$ ; 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 ;
01EE 1335 ;
1. SOME OF THE TRANSITION PAGES ARE ON THE MODIFIED LIST.
2. THERE IS I/O IN PROGRESS ON THE TRANSITION PAGES.

```

```

01EE 1336 ; TO COVER THE FORMER CASE (SINCE WE CANT REALLY TELL), THE MODIFIED LIST MUST
01EE 1337 ; BE TOTALLY FLUSHED. HOWEVER THIS IS ACTUALLY QUITE RARE.
01EE 1338 ;
0000'CF D4 01EE 1339 CLRL W^SCH$GL_MFYLOLIM ; FORCE ENTIRE MODIFY LIST TO BE WRITTEN
0000'CF B4 01F2 1340 CLRW W^SCH$GL_MFYLIM ; CLEAR PART OF HI LIMIT, NOT PART THAT
59 00 D2 01F6 1341 ; INDICATES MODIFIED WRITING IN PROGRESS
FF50 31 01F6 1342 MCOML #0,R9 ; NO, TRY FOR ANOTHER
0117 31 01F9 1343 BRW 20$ ; NOW ATTEMPT CLEANUP
01FC 31 01FC 1344 60$: BRW RELPHD ; GO RELEASE PROCESS HEADER
01FF 1345 70$:
01FF 1346 ;
01FF 1347 ; R4 - PCB OF OUTSWAP CANDIDATE, ALREADY MARKED NON-RESIDENT
01FF 1348 ;
01FF 1349 ;
55 6C A4 D0 01FF 1350 MOVL PCB$L_PHD(R4),R5 ; GET PROCESS HEADER ADDRESS
0203 1351 ;
0018'CF B7 0203 1352 DECW W^SWP$GW_BALCNT ; DECREASE NUMBER IN BALANCE SET
0857 30 0207 1353 BSBW OSINIT ; INIT REGISTERS FOR SCAN
30 A4 B4 020A 1354 CLRW PCB$W_APTCNT(R4) ; INITIALIZE ACTIVE PAGE TABLE COUNT
57 08 A5 :C 020D 1355 MOVZWL PHD$W_WSLIST(R5),R7 ; WS INDEX FOR PERM PAGES
56 12 A5 3C 0211 1356 MOVZWL PHD$W_WSLAST(R5),R6 ; END OF WORKING SET LIST
0215 1357 ;
0215 1358 ; REGISTER CONVENTIONS FOR OWSLOOP ARE:
0215 1359 ;
0215 1360 ; R0 - PFN
0215 1361 ; R1 - SCRATCH, WSLX
0215 1362 ; R2 - WORKING SET LIST ENTRY (VIRTUAL ADDRESS+FLAGS)
0215 1363 ; R3 - SVA OF PTE FOR WORKING SET LIST ENTRY
0215 1364 ; R4 - PCB ADDRESS
0215 1365 ; R5 - PHD ADDRESS
0215 1366 ; R6 - END INDEX TO WORKING SET LIST
0215 1367 ; R7 - WSLX (WORKING SET LIST INDEX)
0215 1368 ; R8 - PTE CONTENT
0215 1369 ; R9 - WORKING POINTER TO SWP$AL_MAP
0215 1370 ; R10 - PTE$M_VALID!PTE$C_ERKW
0215 1371 ; R11 - BASE ADDRESS OF SWP$AL_MAP
0215 1372 ;
0215 1373 OWSLOOP: ; OUTSWAP WS LOOP
52 6547 D0 0215 1374 MOVL (R5)[R7],R2 ; GET WORKING SET LIST ENTRY
15 52 E9 0219 1375 BLBC R2,NOTVALID ; SKIP IF NOT VALID
FDE1' 30 021C 1376 BSBW MMG$SVAPTECHK ; CONVERT VA TO SVA OF PTE
021F 1377 ;
021F 1378 ; R3 <- SVA OF PTE FOR VA IN R2
021F 1379 ;
58 63 D0 021F 1380 MOVL (R3),R8 ; GET CONTENT OF PTE
02 19 0222 1381 BLSS 10$ ; CONTINUE IF VALID PAGE
52 D7 0224 1382 DECL R2 ; CLEAR VALID FLAG
50 52 E0 8F 8A 0226 1383 10$: BICB #^C<WSL$M_VALID!WSL$M_PAGTYP!WSL$M_PFNLOCK>,R2; ISOLATE INTERESTING
58 15 00 EF 022A 1384 EXTZV #PTE$V_PFN,#PTE$S_PFN,R8,R0 ; GET PFN FROM PTE
06 10 022F 1385 BSBW OSDISPAT$CH ; DISPATCH ON PAGE TYPE
0231 1386 NOTVALID:
E0 57 56 F3 0231 1387 AOBLEQ R6,R7,OWSLOOP ; PROCESS ENTIRE WORKING SET LIST
21 11 0235 1388 BRB PROCWRT ; DONE WITH WORKING SET LIST, RESET HEADER
0237 1389 OSDISPAT$:
0237 1390 ASSUME WSL$V_VALID EQ 0
0237 1391 ASSUME WSL$V_PAGTYP EQ 1
0237 1392 ASSUME WSL$V_PFNLOCK EQ 4

```



				0237	1393	ASSUME	PFNSC_PROCESS	EQ	0	
				0237	1394	ASSUME	PFNSC_SYSTEM	EQ	1	
				0237	1395	ASSUME	PFNSC_GLOBAL	EQ	2	
				0237	1396	ASSUME	PFNSC_GBLWRT	EQ	3	
				0237	1397	ASSUME	PFNSC_PPGTBL	EQ	4	
				0237	1398	ASSUME	PFNSC_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		PROCVVALID,-			: 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	PCBSL 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	PCBSW 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	026E	1424	CMPW	R4,PHDSW 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 1431
027F 1432
027F 1433
027F 1434
027F 1435
027F 1436
027F 1437
0000'CF D6 027F 1438
0808 30 0283 1439
04 50 E8 0286 1440
0289 1441
028D 1442 20$:
028D 1443
028D 1444
55 6C A4 D0 0290 1445
58 42 A5 3C 0294 1446
50 89 5A CB 0298 1447 30$:
02 0000'DF40 03 00 ED 029C 1448
62 13 02A4 1449
56 0000'DF40 D0 02A6 1450
66 84000000 8F CA 02AC 1451
0000'DF40 B5 02B3 1452
25 13 02B8 1453
0000'DF40 80 8F 88 02BA 1454
03 00 ED 02C1 1455
04 0000'DF40 02C4 1456
14 12 02C9 1457
02CB 1458
2B 12 02D5 1459
52 02 9A 02D7 1460
FD23' 30 02DA 1461
23 11 02DD 1462
02DF 1463 40$:
05 12 02E9 1464
FD12' 30 02EB 1465 50$:
12 11 02EE 1466
07 0000'DF40 03 00 ED 02F0 1467 55$:
02F8 1468
08 12 02F8 1469
0000'DF40 03 00 F0 02FA 1470
93 57 F5 02FC 1471
000E 31 0302 1472 60$:
0305 1473
0308 1474 80$:
0308 1475
C9 11 0314 1476 BRB 40$
INCL W*SWPSGL_OSWPCNT : ACCOUNT FOR OUTSWAP
BSBW SWPWRITE : WRITE HEADER AND BODY
BLBS R0,20$ : CONTINUE IF NO I/O ERROR
BUG_CHECK OUTSWPERR,FATAL : **** OUT SWAP I/O ERROR
BSBW RELINIT : INIT REGISTERS FOR RELEASE LOOP
MOVL PCB$P_PHD(R4),R5 : GET POINTER TO PHD
MOVZWL PHD$W-PHVINDEX(R5),R8 : GET PROCESS HEADER SLOT INDEX
BICL3 R10,(R9)+,R0 : GET PAGE NUMBER TO RELEASE
CMPZV #PFNSV_PAGTYP,#PFNSS_PAGTYP,@W^PFNSAB_TYPE[R0],#PFNSC_GLOBAL :
BEQL 80$ : PAGE IS GLOBAL, COMPLEX CLEANUP
MOVL @W^PFNSAL_PTE[R0],R6 : GET POINTER TO PAGE TABLE FOR PAGE
BICL #<PTESM_VALID!PTESM_MODIFY>,(R6); : CLEAR VALID AND MODIFY
TSTW @W^PFNSAW_SWPVBN[R0] : WAS I/O IN PROGRESS?
BEQL 40$ : NO, DONT MARK PAGE MODIFIED
BISB #PFNSM_MODIFY,@W^PFNSAB_STATE[R0] : MARK PAGE MODIFIED
CMPZV #PFNSV_LOC,#PFNSS_LOC,- : IF THIS WAS READ IN PROGRESS
@W^PFNSAB_STATE[R0],#PFNSC_RDERR ; AND IS NOW PAGE READ ERROR
40$
BNEQ :
DECREf : AND IF THIS IS THE LAST REFERENCE
BNEQ 60$ :
MOVZBL #PFNSC_BADPAGLST,R2 : THEN DIVERT THE PAGE TO
BSBW MMGSINSPFNT : THE BAD PAGE LIST
BRB 60$ :
DECREf : DECREMENT REFERENCE COUNT FOR PAGE
BNEQ 55$ : NOT RELEASABLE YET
BSBW MMGSRELPFN : RELEASE PFN AS APPROPRIATE
BRB 60$ : GO FOR NEXT PAGE
CMPZV #PFNSV_LOC,#PFNSS_LOC,@W^PFNSAB_STATE[R0],- ;
#PFNSC_ACTIVE : IS STATE ACTIVE?
BNEQ 60$ : NO, THEN LEAVE UNCHANGED
INSV #PFNSC_RELPEND,- : MAKE STATE BE RELEASE PENDING
#PFNSV_LOC,#PFNSS_LOC,@W^PFNSAB_STATE[R0]; : IF SOME I/O OUTSTANDING
60$: SOBGTR R7,30$ : NEXT PAGE IN LIST
BRW RELPHD : RELEASE PROCESS HEADER IF POSSIBLE
80$: DECshr GTR=60$,- : DECREASE SHARE COUNT FOR PAGE
BRB 40$ : RELEASE PAGE TO FREE LIST IF REFCNT=0

```

.SBTTL RELPHD - RELEASE PROCESS HEADER

0316 1479  
0316 1480  
0316 1481  
0316 1482  
0316 1483  
0316 1484  
0316 1485  
0316 1486  
0316 1487  
0316 1488  
0316 1489  
0316 1490  
0316 1491  
0316 1492  
0316 1493  
0316 1494  
0316 1495  
0316 1496  
0316 1497  
0316 1498  
0316 1499  
0316 1500  
0316 1501  
0316 1502  
0316 1503  
0316 1504

```

:++
: FUNCTIONAL DESCRIPTION:
: RELPHD CHECKS THE REFERENCE COUNT ON THE PROCESS HEADER
: AND RELEASES THE PAGE TABLES FROM THE PROCESS HEADER WHEN ALL
: OF THEIR PAGES HAVE BEEN RELEASED. THE PAGE TABLES ARE FIRST WRITTEN
: TO THE SWAP IMAGE IF THEY ARE MARKED AS UPDATED.
:
: CALLING SEQUENCE:
: BRW/JMP RELPHD
:
: INPUT PARAMETERS:
: R8 - BALANCE SLOT INDEX FOR HEADER TO BE RELEASED
:
: OUTPUT PARAMETERS:
: R0-R7,R9,R10 VOLATILE
:
: SIDE EFFECTS:
: THE PAGE TABLES FROM THE PROCESS HEADER MAY BE WRITTEN TO THE
: SWAP IMAGE FOR THE PROCESS IF THEY HAVE BEEN UPDATED.
:--

```

RELPHD:

```

TSTW @W*PHV$GL_REFCBAS[R8] ; SEE IF PROCESS HEADER IS RELEASABLE
BEQL 5$ ; YES, FREE ACTIVE PAGE TABLES
BRW OSWPEXIT ; NO, TRY LATER
5$: MOVL W*SWP$GL_BSLOTSZ,R7 ; SET ITERATION COUNT TO WHOLE BALANCE SLOT
MOVL R7,R8,R1 ; GET LONG WORD OFFSET TO SLOT
MOVAL @W*SWP$GL_BALSPT[R1],R6 ; POINT TO BASE OF THIS SLOT
BSBW OSINIT ; INIT REGISTERS FOR SCAN
CVTWL @W*PHV$GL_PIXBAS[R8],R4 ; GET INDEX TO PROCESS IN SLOT
BLSS 12$ ; BR IF DELETED PROCESS
54: MOVL @W*SCH$GL_PCBVEC[R4],R4 ; AND TRANSLATE TO PCB ADDRESS
MOVL PCB$$_PHD[R4],R5 ; GET PROCESS HEADER ADDRESS
MOVL R8,PCB$$_PHD[R4] ; INDICATE NO PHD FOR PROCESS
SUBL R5,PHD$$_POBR[R5] ; UNBIAS MEMORY MANAGEMENT BASE REGISTERS
SUBL R5,PHD$$_P1BR[R5] ; FOR BOTH P0 AND P1 SPACE
BBCC #PCB$$_PRDRES,PCB$$_STS[R4],7$ ; MARK PHD NON-RESIDENT
7$: MOVL PHD$$_WSLX[R5],AP ; GET POINTER TO WSLX SAVE AREA
MOVAL (R5)[AP],AP ; AND CONVERT TO BYTE ADDRESS
MOVL PHD$$_BAK[R5],FP ; GET POINTER TO BACKING STORE VECTOR
MOVAL (R5)[FP],FP ; AND CONVERT TO BYTE ADDRESS
CLR W PHD$$_EMPTPG[R5] ; CLEAR COUNT OF EMPTY WSL PAGES
10$: MOVL (R6)+,(FP)+ ; COPY ENTRY FROM SPT
BLSS 15$ ; BR IF VALID
BNEQ 11$ ; BR IF NOT EMPTY WSL PAGE
IN CW PHD$$_EMPTPG[R5] ; COUNT EMPTY WSL PAGES
11$: CLRL -4(R6) ; ZAP INVALID ENTRY TO NO-ACCESS
CLR W (AP)+ ; AND CLEAR WSLX VALUE FOR PAGE
BRB 20$
12$: BRW DELPHD ; FINISH DELETE FOR PROCESS
15$: EXTZV #PTES$_PFN,#PTES$_PFN,-4(FP),R0 ; GET PFN FOR VALID ENTRY
BEQL 11$ ; DEMAND ZERO OR NULL PTE
INSV @W*PFNSAL_BAK[R0],#PTES$_PGFLVB,#PTES$_PGFLVB,-4(FP) ; SAVE BACKU

```

```

0000'DF48 B5 0316 1505
03 13 0318 1506
00E1 31 031D 1507
57 0000'CF DO 0320 1508
51 58 57 C5 0325 1509
56 0000'DF41 DE 0329 1510
072F 30 032F 1511
54 0000'DF48 32 0332 1512
43 19 0338 1513
54 0000'DF44 DO 033A 1514
55 6C A4 DO 0340 1515
6C A4 58 DO 0344 1516
00C8 C5 55 C2 0348 1517
00D0 C5 55 C2 034D 1518
00 24 A4 12 E5 0352 1519
5C 48 A5 DO 0357 1520
5C 654C DE 035B 1521
5D 44 A5 DO 035F 1522
5D 654D DE 0363 1523
00D8 C5 B4 0367 1524
8D 86 DO 036B 1525
10 19 036E 1526
04 12 0370 1527
00D8 C5 B6 0372 1528
FC A6 D4 0376 1529
BC B4 0379 1530
23 11 037B 1531
0084 31 037D 1532
50 FC AD 15 00 EF 0380 1533
EE 13 0386 1534
FC AD 16 00 0000'DF40 F0 0388 1535

```

```

0391 1536
0391 1537
0391 1538
0391 1539
89 5A 50 C9 0397 1540
00 FC AD 1F E2 0398 1541
      C8 57 F5 03A0 1542 20$:
      59 58 C2 03A3 1543
52 20 A4 D0 03A6 1544
0014'CF 54 D0 03AA 1545
54 59 1E 9C 03AF 1546
0012'CF 54 B0 03B3 1547
      53 58 D0 03B8 1548
      0000'CF D6 03BB 1549
      06CC 30 03BF 1550
      04 50 E8 03C2 1551
      068B 30 03C5 1552
      58 6C A4 D0 03C9 1553 30$:
50 89 5A CB 03D0 1555 40$:
56 0000'DF40 D0 03D4 1556
      66 50 D0 03DA 1557
      0000'DF40 B7 03DD 1558
      04 13 03E2 1559
      03E4 1560
      53 56 D0 03E8 1561 50$:
      0683 30 03EB 1562
      66 D4 03EE 1563
      DD 57 F5 03F0 1564
0000'DF48 01 AE 03F3 1565
      0000'DF48 B4 03F9 1566
      6C A4 D4 03FE 1567
      0401 1568
      0401 1569
05D3 31 0401 1570 OSWPEXIT:
                                BRW  SWAPRETRY

```

PFN REFERENCE -  
<@W^PFNSAx WSLX[R0],(AP)+>,- ; AND WORKING SET LIST INDEX  
LONG OPCODE=CVTLW,-  
IMAGE=SYS NONPAGED  
R0,R10,(R9)+ ; SET INTO SWAPPER MAP  
#PTEBV\_VALID,-4(FP),20\$ ; MARK PAGE VALID FOR INSWAP PURPOSES  
R7,10\$ ; SCAN ENTIRE BALANCE SLOT  
R11,R9 ; COMPUTE NUMBER OF PAGES \* 4  
PCB\$L WSSWP(R4),R2 ; WORKING SET SWAP SLOT  
R4,W^OSWPPCB ; SAVE PCB ADDRESS FOR SLOT OWNER  
#<32-2>,R9,R4 ; DIVIDE COUNT BY 4  
R4,W^OSWPPGS ; SAVE COUNT OF OUTSWAP PAGES  
R11,R3 ; SET SVA OF MAP FOR I/O  
W^SWP\$GL\_HOSWPCNT ; ACCOUNT FOR HEADER OUTSWAP  
SWWRITE ; WRITE ACTIVE PAGE TABLES  
R0,30\$ ; CONTINUE IF NO ERROR  
BUG CHECK APTWRTERR,FATAL ; \*\*\*\* ACTIVE PAGE TABLE SWAP I/O ERROR  
BSBW RELINIT ; INIT REGISTERS FOR RELEASE LOOP  
PCB\$L PHD(R4),R8 ; RESTORE BALANCE SLOT INDEX  
R10,(R9)+,R0 ; ISOLATE PAGE FRAME NUMBER  
@W^PFNSAL\_PTE[R0],R6 ; GET PTE ADDRESS  
R0,(R6) ; MAKE PTE CORRECT BUT INVALID  
@W^PFNSAW\_REFCNT[R0] ; DROP REFERENCE COUNT  
50\$ ; MUST BE ZERO  
BUG CHECK APTREFHIGH,FATAL ; INCONSISTENT PAGE TABLE REFERENCE COUNT  
MOV[ R6,R3 ; SVAPTE FOR DELCON  
BSBW REDELPAGE ; RELEASE PAGE THROUGH DELCONPFN  
CLRL (R6) ; SET NO ACCESS ON PFN  
SOBGTR R7,40\$ ; CONTINUE FOR ALL ACTIVE PAGE TABLES  
MNEGW #1,@W^PHV\$GL\_REFCBAS[R8] ; MARK BALANCE SLOT AVAIL  
CLRW @W^PHV\$GL\_PIXBAS[R8] ; AND SET PIX TO NULL  
CLRL PCB\$L\_PHD(R4) ; AND SEVER CONNECTION WITH PROCESS  
; OUTSWAP COMPLETE  
; RETRY SWAP SCHEDULE AFTER OUTSWAP

```

0404 1573      .SBTTL DELPHD - DELETE PROCESS HEADER FOR DELETED PROCESS
0404 1574
0404 1575      :
0404 1576      : FUNCTIONAL DESCRIPTION:
0404 1577      : DELPHD IS ENTERED BY RELPHD IF THE PROCESS INDEX ASSOCIATED WITH
0404 1578      : THE BALANCE SLOT IS NEGATIVE INDICATING THE PROCESS HAS BEEN DELETED.
0404 1579      : NOW THAT THE REFERENCE COUNT FOR THE HEADER IS ZERO, ALL PAGES AND
0404 1580      : BACKING STORE PAGES CAN BE RELEASED PERMITTING RELEASE OF THE BALANCE
0404 1581      : SLOT. AT THIS POINT THE SPT ENTRIES ARE VALID WITH A PFN, DEMAND ZERO,
0404 1582      : OR BACKING STORE ADDRESS FORM. THERE ARE NO REMAINING TRANSITION PAGES.
0404 1583      :
0404 1584      : INPUT PARAMETERS:
0404 1585      : R1 - PRODUCT OF SGN$C_B$SLOTSZ * BALANCE SLOT INDEX
0404 1586      : R6 - ADDRESS OF FIRST SPT ENTRY FOR THIS BALANCE SLOT
0404 1587      : R7 - SGN$C_B$SLOTSZ
0404 1588      : R8 - BALANCE SLOT INDEX
0404 1589      : R10 - MASK OF PTE$M_VALID!PTE$M_MODIFY!PTE$C_ERKW
0404 1590      :
0404 1591      :
0404 1592      : DELPHD:
55 51 09 9C 0404 1593      ROTL #9,R1,R5      : COMPUTE OFFSET TO PHD FROM BASE
55 0000'CF C0 0408 1594      ADDL W^SWP$GL,BALBASE,R5 : FORM PHD ADDRESS
5B 1F A5 9A 040D 1595      MOVZBL PHD$B_PAGFIL(R5),R11 : GET PAGING FILE NUMBER
50 86 D0 0411 1596 10$:      MOVL (R6)+,R0      : GET PTE FROM SPT
2C 13 0414 1597      BEQL 40$      : BR IF EMPTY
04 19 0416 1598      BLSS 20$      : BR IF VALID
16 50 1A E0 0418 1599      BBS #PTE$V_TYP1,R0,25$ : BR IF TYPE 1 (BACKING STORE)
50 5A CA 041C 1600 20$:      BICL R10,R0      : ISOLATE PFN
1E 13 041F 1601      BEQL 30$      : SKIP DEMAND ZERO PTE
59 0000'DF40 D0 0421 1602      MOVL @W^PFNSAL_BAK[R0],R9 : GET BACKUP ADDRESS
FF A6 84 BF 8A 0427 1603      BICB #<<PTE$M_VALID!PTE$M_MODIFY>>-24>,-1(R6) ; CLEAR VALID AND MODIFY
0642 30 042C 1604      BSBW REDELPAE      : RELEASE PAGE
50 50 59 D0 042F 1605      MOVL R9,R0      : GET BACKUP ADDRESS
50 50 16 00 EF 0432 1606 25$:      EXTZV #PTE$V_PGFLVB,#PTE$S_PGFLVB,R0,R0 ; GET PAG FIL VB
06 13 0437 1607      BEQL 30$      : BR IF NONE
53 5B D0 0439 1608      MOVL R11,R3      : SET PAGING FILE NUMBER FOR RELEASE
FBC1' 30 043C 1609      BSBW MMG$DALCPAGFIL : DEALLOCATE PAGING FILE PAGE
FC A6 D4 043F 1610 30$:      CLRL -4(R6)      : ZAP SPT ENTRY
CC 57 F5 0442 1611 40$:      SOBGTR R7,10$   : RELEASE ENTIRE HEADER
0445 1612      INVALID      : INVALIDATE HEADER
0000'DF48 01 AE 0448 1613      MNEGW #1,@W^PHV$GL_REFCBAS[R8] : MARK SLOT EMPTY
0000'DF48 B4 044E 1614      CLRW @W^PHV$GL_PIXBAS[R8] : POINT OWNER PIX AT NULL PROCESS
0000'CF B7 0453 1615      DECW W^SCH$GW_DELPHDCT : ACCOUNT FOR DELETED HEADER
057D 31 0457 1616      BRW SWAPRETRY   : AND RETRY SWAP ATTEMPT

```

```

045A 1619 .SBTTL GBLTRANS/GBLVALID/GBLWRTVALID - HANDLE GLOBAL PAGES
045A 1620
045A 1621 :
045A 1622 : GBLTRANS HANDLES THE CASE OF A GLOBAL PAGE IN TRANSITION,
045A 1623 : WHICH IMPLIES THAT THE PAGE HAS BEEN FAULTED BUT IS NOT YET
045A 1624 : RESIDENT. THE WORKING SET LIST ENTRY FOR THIS PAGE WILL BE
045A 1625 : DELETED AND THE PAGE WILL HAVE TO BE FAULTED AGAIN.
045A 1626 :
045A 1627 :
045A 1628 GBLTRANS: ; TRANSITION GLOBAL PAGE
045A 1629 GBLWRTTRANS: ; TRANSITION WRITABLE GLOBAL PAGE
50 15 00 0000'DF40 F0 045A 1630 INSV @W^MMG$GL_GPTBASE[RO],#PTE$V_PFN,#PTE$S_PFN,RO ; GET GLOBAL PFN FRO
0462 1631
0462 1632 .ENABL LSB
0462 1633 GBLDROP: ; DROP GLOBAL PAGE FROM WORKING SET
51 57 DO 0462 1634 MOVL R7,R1 ; GET WSL INDEX FOR RELEASE
53 DD 0465 1635 PUSHL R3 ; SAVE SVAPTE FOR FOLLOWING DECPTRF
FB96' 30 0467 1636 BSBW MMG$DELWSLEX ; DELETE WSL GIVEN INDEX
08 BA 046A 1637 POPR #^M<R3> ; RESTORE SVAPTE
FB91' 30 046C 1638 BSBW MMG$DECPTRF ; AND DROP PAGE TABLE REFERENCE
046F 1639 DECSHR GTR=10$,- ; DECREASE SHARE COUNT
046F 1640 IMAGE_FLAG=SYS_NONPAGED
047B 1641 PROCDROP:
047B 1642 DECFR GTR=20$ ; AND REF COUNT IF LAST SHARER
FB76' 30 0487 1643 BSBW MMG$RELPFN ; RELEASE PAGE IF LAST REFERENCE
048A 1644 10$:
07 0000'DF40 03 00 05 048A 1645 RSB ; RETURN FOR NEXT PAGE
048B 1646 20$: CMPZV #PFNSV_LOC,#PFNSS_LOC,@W^PFNSAB_STATE[RO],- ;
0493 1647 #PFNSC_ACTIVE ; CHECK FOR ACTIVE STATE
08 12 0493 1648 BNEQ 30$ ; NO, THEN LEAVE STATE UNCHANGED
03 03 F0 0495 1649 INSV #PFNSC_RELPEND,- ; SET STATE TO RELEASE PENDING IF
0000'DF40 03 00 0497 1650 #PFNSV_LOC,#PFNSS_LOC,@W^PFNSAB_STATE[RO] ; I/O OUTSTANDING
05 049D 1651 30$: RSB
049E 1652 .DSABL LSB
049E 1653
049E 1654 :
049E 1655 : GBLVALID HANDLES A VALID, NON-WRITABLE, PAGE.
049E 1656 :
049E 1657 GBLVALID: ; VALID GLOBAL PAGE
049E 1658 :
049E 1659 : IF THE GLOBAL PAGE IS BEING ACTIVELY SHARED THEN IT WILL BE DROPPED
049E 1660 : FROM THE WORKING SET AND REFAULTED LATER (PRESUMABLY WITHOUT I/O).
049E 1661 :
08 6D 05 E0 049E 1662 BBS #WSL$V WSLOCK,(FP),10$ ; DON'T DROP PAGES LOCKED IN WORKING SET
04A2 1663 PFN_REFERENCE -
04A2 1664 CMPW <#1,@W^PFNSAx SHRCNT[RO]>,- ; IS THERE ACTIVE SHARING OF THIS PA
04A2 1665 LONG_OPCODE=CMPL,-
04A2 1666 IMAGE=SYS_NONPAGED
25 12 04A8 1667 BNEQ GBLWRTVALID ; YES, DROP IT AND REFAULT LATER
04AA 1668 :
04AA 1669 : OTHERWISE THE PAGE WILL BE WRITTEN TO THE SWAP IMAGE SINCE IT IS
04AA 1670 : UNLIKELY TO BE RESIDENT UPON INSWAP.
04AA 1671 :
89 5A 50 C9 04AA 1672 10$: BISL3 RO,R10,(R9)+ ; SET IN SWAPPER MAP FOR OUT SWAP
FB4F' 30 04AE 1673 BSBW MMG$DECPTRF ; DROP PAGE TABLE REFERENCE FOR PAGE
04B1 1674 GBLRESET: ; RESET SLAVE PTE TO GPTX FORMAT
51 0000'DF40 0000'CF C3 04B1 1675 SUBL3 W^MMG$GL_GPTBASE,@W^PFNSAL_PTE[RO],R1 ; GET GPTX FOR PAGE

```

```

51 51 1E 9C 04BA 1676 ROTL #<32-2>,R1,R1 ; AND CONVERT TO CORRECT SCALE
04BE 1677 ASSUME PTE$V_TYPO EQ PTE$S_GPTX ;
52 63 00 51 16 E2 04BE 1678 BBSS #PTE$V_TYPO,R1,10$ ; MARK AS GLOBAL
845FFFFFF 8F CB 04C2 1679 10$: BICL3 #<PTE$M_VALID ! - ; OBTAIN PERMANENT BITS FOR PTE
04CA 1680 PTE$M_TYPO - ; BY CLEARING ALL OTHERS
04CA 1681 PTE$M_TYPI - ;
04CA 1682 PTE$M_PFN>, (R3),R2 ; TO FORM TRANSITION GLOBAL PTE
63 52 51 C9 04CA 1683 BISL3 R1,R2,(R3) ; MUST SET ENTIRE PTE AT ONE TIME
04CE 1684 ; SO THAT I/O CAN SEE CONSISTENT PTE
05 04CE 1685 RSB ; RETURN FOR NEXT PAGE
04CF 1686
04CF 1687 :
04CF 1688 : GBLWRTVALID HANDLES THE CASE OF A WRITABLE GLOBAL PAGE.
04CF 1689 : SUCH PAGES ARE DROPPED FROM THE WORKING SET BEFORE OUTSWAPPING
04CF 1690 : AND MUST BE SUBSEQUENTLY RE-FAULTED.
04CF 1691 :
04CF 1692 GBLWRTVALID: ; VALID WRITABLE GLOBAL PAGE
04CF 1693 BBCC #PTE$V_MODIFY,(R3),10$ ; TEST AND CLEAR MODIFY BIT IN SLAVE PTE
0000'DF40 07 63 1A E5 04CF 1694 BISB #PFNSM_MODIFY,@W^PFNSAB_STATE[RO] ; AND SAVE MODIFY STATE
80 8F 88 04D3 1694 BSBB GBLRESET ; RESET PTE
D5 10 04DA 1695 10$: BRB GBLDROP ; DELETE WORKING SET LIST ENTRY
84 11 04DC 1696

```

```

04DE 1699 .SBTTL PROCTRANS - PROCESS PAGE IN TRANSITION
04DE 1700
04DE 1701
04DE 1702
04DE 1703
04DE 1704
04DE 1705
04DE 1706
04DF 1707
04DE 1708
04E1 1709
04E6 1710
04E8 1711
04EB 1712
04EE 1713
04F0 1714
04F0 1715
04F0 1716
04F0 1717
04F0 1718
04F0 1719
04F0 1720
04F0 1721
04F0 1722
04F0 1723
04F0 1724
04F0 1725
04F0 1726
04F7 1727
04FB 1728
04FF 1729
04FF 1730
0505 1731
0507 1732
050A 1733
050E 1734
050E 1735
0512 1736
0515 1737
051B 1738
051B 1739
051F 1740
051F 1741
051F 1742
051F 1743
051F 1744
0525 1745
0526 1746
0526 1747
0526 1748

04 0000'DF40 03 00 ED
      08 12
      51 57 DO
      FB12' 30
      8B 11

04 0000'DF40 07 E4
      04 63 1A E1
      00 6D 08 E2

0000'DF40 01 B1
      14 13
      04 52 E9
      OD 6D 08 E1

      51 59 5B C3
      51 04 C6
0000'DF40 51 B0
      89 5A 50 C9

0000'DF40 10 88
      05 05

04DE 1707 PROCTRANS:
04DE 1708 CMPZV #PFNSV_LOC,#PFNS$LOC,- ; PROCESS PAGE IN TRANSITION
04E1 1709 @W^PFNS$AB_STATE[R0],#PFNS$C_RDERR ; IF THIS PAGE COULD NOT
04E6 1710 BNEQ PROCVALID ; BE SUCCESSFULLY READ
04E8 1711 MOVL R7,R1 ; DROP IT FROM THE WORKING SET
04EB 1712 BSBW MMG$DELWSLEX ; DELETE THE WSL ENTRY GIVEN WSL INDEX
04EE 1713 BRB PROCDROP ; AND RELEASE THE PFN IF LAST REF

PROCVALID HANDLES THE CASE OF A VALID PROCESS PAGE WHICH INCURS
SOME SPECIAL PROCESSING IF THERE IS I/O IN PROGRESS. AN I/O IN
PROGRESS PAGE IS SWAPPED WITH THE BODY OF THE PROCESS TO RESERVE
SPACE FOR IT IN THE SWAP IMAGE AND IS LATER WRITTEN WITH CORRECT
CONTENT BY THE MODIFIED PAGE WRITER TO THIS RESERVED SPACE IN THE
SWAP IMAGE.

PROCVALID: ; PROCESS VALID PAGE
.ENABL LSB

10$
BBSC #PFNSV_MODIFY,@W^PFNS$AB_STATE[R0],20$ ; BR IF PAGE MODIFIED
BBC #PTESV_MODIFY,(R3),30$ ; BR IF PAGE NOT MODIFIED
BBSS #WSLSV_MODIFY,(FP),30$ ; SET WORKING SET MODIFIED BIT

30$
CMPW #1,@W^PFNS$AW_REFCNT[R0] ; CHECK FOR I/O OUTSTANDING
BEQL 40$ ; NO, NONE
BLBC R2,SETWRTBAK ; BRANCH IF TRANSITION PAGE
BBC #WSLSV_MODIFY,(FP),40$ ; DONT WRITE UNMODIFIED PAGES

SETWRTBAK: ; SET PAGE FOR WRITE BACK TO SWAP FILE
SUBL3 R11,R9,R1 ; GET OFFSET TO PAGE IN SWAP MAP
DIVL #4,R1 ; SCALE BACK TO PAGE NUMBER
MOVW R1,@W^PFNS$AW_SWPVBN[R0] ; SET OFFSET INTO SWAP IMAGE LESS APTCNT

40$:
BISL3 R0,R10,(R9)+ ; PUT PAGE IN SWAPPER MAP

; SET DELETE CONTENT FLAG TO CAUSE PAGE TO BE PLACED AT HEAD
; OF FREE PAGE LIST AND CONTENT FORGOTTEN.

DELCON: BISB #PFNSM_DELCON,@W^PFNS$AB_STATE[R0] ; SET TO DELETE CONTENT
RSB ; RETURN FOR NEXT PAGE
.DSABL LSB

WSLERR: BUG_CHECK IVWSETLIST,FATAL ; INVALID WORKING SET LIST ENTRY

```



052A 1751  
052A 1752  
052A 1753  
052A 1754  
052A 1755  
052A 1756  
052A 1757  
052A 1758  
052A 1759  
052A 1760  
052A 1761  
052D 1762  
0530 1763  
0534 1764

.SBTTL PAGE TABLE WORKING SET LIST ENTRIES

PAGE TABLE AND PROCESS HEADER ENTRIES IN THE WORKING SET LIST  
ARE IGNORED DURING THE PROCESS BODY OUTSWAP SCAN OF THE WORKING  
SET LIST.

30 A4 B6  
6D 55 C2  
00 6D 1E E2  
05

PPGTBLTRANS:  
PPGTBLVALID:  
INCU  
SUBL  
BBSS  
RSB

PCBSW APTCNT(R4)  
R5,(FP)  
#VASV\_SYSTEM,(FP),10\$

: TRANSITION PAGE TABLE  
: VALID PAGE TABLE  
: ACCUMULATE ACTIVE PAGE TABLE COUNT  
: UNBIAS WSL VA FOR PAGE TABLE  
: BUT FORCE SYSTEM BIT ON IN VA  
: RETURN

```

0535 1767 .SBTTL INSWAP
0535 1768
0535 1769 -----
0535 1770 : PERFORM REQUESTED INSWAP
0535 1771 :
0535 1772 : INPUTS:
0535 1773 : R4 - PCB ADDRESS OF INSWAP CANDIDATE
0535 1774 : -----
0535 1775
0535 1776 INSWAP:
55 6C A4 D0 0535 1777 : PERFORM INSWAP
06 13 0535 1778 : GET CURRENT PROCESS HEADER SLOT
58 42 A5 3C 0539 1778 : NONE, MUST ALLOCATE ONE
2F 11 053B 1779 : GET BALANCE SLOT INDEX
58 D4 053F 1780 : AND CONTINUE
0000'DF48 B5 0541 1781 10$: CLRL R8 : INIT INDEX FOR BALANCE SLOT SEARCH
OF 19 0543 1782 20$: TSTW @W^PHV$GL_REF(BAS[R8]) : CHECK FOR EMPTY
F1 58 00000000'EF F2 0548 1783 : BLSS 30$ : YES, GOT ONE
5D 01 1F 9C 054A 1784 : AOBLS SGN$GL_BALSETCT,R8,20$ : TRY ALL BALANCE SET SLOTS
0552 1785 : ROTL #31,#1,FP : SET FLAG TO PERMIT OUTSWAPPING
0556 1786 : OF PROCESSES
0556 1787 : BRW OUTSWAP : OUTSWAP IF NECESSARY TO GET SLOT
0000'DF48 60 A4 B0 0559 1788 30$: MOVW PCB$S_PID(R4),@W^PHV$GL_PIXBAS[R8] : SET PIX FOR BALANCE SET SL
0000'DF48 B4 0560 1789 : CLRW @W^PHV$GL_REF(BAS[R8]) : AND BUMP REFERENCE COUNT
50 58 0000'CF C5 0565 1790 : MULL3 W^SWP$GL_BSLOTSZ,R8,R0 : COMPUTE BALANCE SLOT OFFSET
6C A4 50 09 9C 056B 1791 : ROTL #9,R0,PCB$S_PHD(R4) : MAKE BYTE OFFSET
0570 1792 : 40$: CLRL R9 : POSITIVE UNTIL I/O COMPLETE
59 D4 0570 1793 50$: BSBW MMG$ALLOCPFN : INITIALIZE SWAPPER MAP INDEX
FA8B' 30 0572 1794 : TSTL R0 : ALLOCATE A PAGE
50 D5 0575 1795 : BGEQ 60$ : MAKE SURE IT WAS ALLOCATED
04 18 0577 1796 : BUG CHECK INSNFREPAG,FATAL : YES, CONTINUE
0579 1797 : INCQ @W^PFNS$AW_REFCNT[R0] : INSUFFICIENT FREE PAGES
0000'DF40 B6 057D 1798 60$: MOV B #PFNS$ ACTIVE,@W^PFNS$AB_STATE[R0] : REFERENCE PAGE
0000'DF40 07 90 0582 1799 : BISL3 #<PTE$C_ERKW!PTESM_VALID>,R0,@W^SWP$GL_MAP[R9] : AND MARK IT ACTIVE
0000'DF49 50 B0000000 8F C9 0588 1800 : AOBLS R10,R9,50$ : MARK VALID, WRITABL
DB 59 5A F2 0593 1801 : CLRL @W^SWP$GL_MAP[R9] : REPEAT FOR ACL REQUIRED PAGES
0000'DF49 D4 0597 1802 : 50$: : 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$GW_BALCNT : ADD ONE PROCESS TO BALANCE SET
0000'CF 54 D0 05A0 1808 : MOV R4,W^SWP$GL_INPCB : SAVE POINTER TO IN SWAP PCB
0000'CF 5A D0 05A5 1809 : MOV 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 : PERFORM INSWAP I/O OPERATION
05AF 1813 :
05AF 1814 : -----
05AF 1815 :
05AF 1816 :
52 20 A4 D0 05AF 1817 : MOVL PCB$S_WSSWP(R4),R2 : GET SWAP IMAGE DISK ADDRESS
07 24 A4 2A 15 05B3 1818 : BLEQ COPYSHELL : BRANCH IF SHELL IN SWAP
50 30 A4 12 E1 05B5 1819 : BBC #PCB$V_PHDRES,PCB$S_STS(R4),70$ : SWAP EVERYTHING IF HEADER NON-RES
50 52 50 C0 05BA 1820 : MOVZWL PCB$W_XPTCNT(R4),R0 : GET ACTIVE PAGE TABLE COUNT
53 0000'DF DE 05BE 1821 : ADDL R0,R2 : ADD PAGE TABLE COUNT
54 5A D0 05C1 1822 70$: MOVAL @W^SWP$GL_MAP,R3 : SVA OF PAGE TABLE FOR I/O
05C6 1823 : MOVL R10,R4 : NUMBER OF PAGES TO READ

```

```

0000'CF 5A C0 05C9 1824 ADDL2 R10,W^SWP$GL_ISWPPAGES ; UPDATE TOTAL PAGES INSWAPPED
      0000'CF D6 05CE 1825 INCL W^SWP$GL_ISWPCNT ; BUMP INSWAP COUNTER
      04B3 30 05D2 1826 BSBW SWPREAD ; PERFORM READ
      04 50 E8 05D5 1827 BLBS RO,80$ ; BRANCH IF NO ERROR IN READ
      0089 31 05D8 1828 BUG_CHECK INSWAPERR,FATAL ; **** BUGCHECK ON I/O ERROR
      05DC 1829 80$: BRW- SETUP ; SET UP PROCESS IN BALANCE SLOT
      05DF 1830
      05DF 1831 COPYSHELL:
54 0000'CF DE 05DF 1832 MOVAL W^MMG$AL_SYSPCB,R4 ; ADDRESS OF SYSTEM PCB
55 6C A4 D0 05E4 1833 MOVL PCB$L_PHD(R4),R5 ; ADDRESS OF SYSTEM PROCESS HEADER
56 0000'CF D0 05E8 1834 MOVL W^SWP$GL_SHELLIO,R6 ; GET I/O PAGE COUNT FOR SHELL
52 00000000'EF 9E 05ED 1835 MOVAB L^SWP$GL_SHELLBAS,R2 ; GET ADDRESS OF SHELL
      FA09' 30 05F4 1836 BSBW MMG$SVAPTECHK ; GET ADDRESS OF PAGE TABLE ENTRY
      0048 8F BB 05F7 1837 PUSHR #^M<R3,R6> ; SAVE SVAPTE AND PAGE COUNT FOR LATER
      58 D4 05FB 1838 CLRL R8 ; SET FLAG INDICATING NO I/O NEEDED
57 0103 8F 3C 05FD 1839 MOVZWL #256+3,R7 ; SET FLAGS TO LOCK ONLY VALID OR
      0602 1840 ; TRANSITION PAGES AND CREATE OTHERS
      0602 1841 ; WITHOUT ZEROING THE PHYSICAL PAGE
      52 02 90 0602 1842 MOVB #WSL$C_SYSTEM,R2 ; SET PAGE TYPE IN LOW BITS
0000'CF 00' 88 0605 1843 BISB S^MMG$M_NOWAIT,W^MMG$GB FREWFLGS ; PREVENT FREWLSE MWAIT
      F9F3' 30 060A 1844 10$: BSBW MMG$IOLOCKPAG ; LOCK THE PAGE INTO SYSTEM WORKING SET
      04 50 E8 060D 1845 BLBS RO,20$ ; BRANCH IF SUCCEEDED
      58 50 88 0610 1846 BUG_CHECK INSNFREPAG,FATAL ; INSUFFICIENT FREE PAGES
52 0200 C2 9E 0614 1847 20$: BISB RO,R8 ; SET FLAG (BIT 1) IF WE HAVE TO I/O IT
      53 04 C0 061C 1849 ADDL #4,R3 ; BUMP VA TO NEXT PAGE
      E8 56 F5 061F 1850 SOBGTR R6,10$ ; BUMP PTE TO NEXT ENTRY
      18 BA 0622 1851 POPR #^M<R3,R4> ; LOOP THROUGH THE PAGES
      16 58 01 E1 0624 1852 BBC #1,R8,40$ ; RECOVER SVAPTE AND PAGE COUNT
0000'CF 00' 8A 0628 1853 BICB S^MMG$M_NOWAIT,W^MMG$GB FREWFLGS ; ALLOW FREWLSE MWAIT
      52 D4 062D 1854 CLRL R2 ; SHELL IS PAGE FILE 0 AND VBN 0
      0456 30 062F 1855 BSBW SWPREAD ; PERFORM SHELL READ
      04 50 E8 0632 1856 BLBS RO,30$ ; BRANCH IF NO ERROR IN READ
      0000'CF 00' 88 0635 1857 BUG_CHECK INSWAPERR,FATAL ; **** BUGCHECK ON I/O ERROR
      0639 1858 30$: BISB S^MMG$M_NOWAIT,W^MMG$GB FREWFLGS ; PREVENT FREWLSE MWAIT
      063E 1859 40$: SETIPL #IPL$_ASTDEL ; ALLOW RESCHEDULE AND PAGEFAULTS WHILE
      0641 1860 ; COPYING SHELL BUT NOT COMPLETION ASTS
56 0000'CF D0 0641 1861 MOVL W^SWP$GL_SHELLIO,R6 ; GET I/O PAGE COUNT FOR SHELL
57 00000000'EF 9E 0646 1862 MOVAB L^SWP$GL_SHELLBAS,R7 ; GET ADDRESS OF SHELL
50 56 09 78 064D 1863 ASHL #9,R6,RO- ; GET BYTE COUNT
      51 D4 0651 1864 CLRL R1 ; FORM DESTINATION VA
61 67 50 28 0653 1865 MOV3 RO,(R7),(R1) ; COPY THE SHELL TO LOCATION 0
      0657 1866 SETIPL #IPL$_SYNCH ; BACK TO BLOCKING IPL
0000'CF 00' 8A 065A 1867 BICB S^MMG$M_NOWAIT,W^MMG$GB FREWFLGS ; ALLOW FREWLSE MWAIT
      51 56 7D 065F 1868 MOVQ R6,R1 ; SET UP COUNT AND VA OF SHELL AGAIN
      F998' 30 0662 1869 BSBW MMG$SVAPTECHK ; GET ADDRESS OF PAGE TABLE ENTRY
      F998' 30 0665 1870 BSBW MMG$UNLOCK ; DROP THE REFERENCE COUNTS
      0668 1871 ; CONTINUE PROCESS CREATION

```









```

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

```



```

51 51 15 00 EF 08D6 2159
      08DB 2160
      08DB 2161
      08DB 2162
      08DB 2163
      04 12 08E0 2164
      08E2 2165
      08E6 2166 35$:
      08E6 2167
      08E6 2168
      08E6 2169
      08EB 2170 40$:
      08EB 2171
      08EB 2172
      08EB 2173
      0000'DF40 52 17 FE85 31 08F0 2174
      52 867FFFFF 8F CA 08FB 2175 50$:
      52 52 5B CB 0902 2177
      61 52 50 C9 0905 2178
      0000'DF40 51 D0 0909 2179
      0000'DF40 07 90 090F 2180
      0000'DF40 02 90 0915 2181
      AE 11 091B 2182
      091D 2183
      091D 2184 55$:
      091D 2185
      091D 2186
      091D 2187
      091D 2188
      52 FC A9 5A CB 091D 2189
      0000'DF42 0000'DF40 D0 0922 2190
      0000'DF42 07 90 092B 2191
      0000'DF42 02 90 0931 2192
      0937 2193
      0937 2194
      0937 2195
      0937 2196
      0000'DF40 D4 0940 2197
      012C 30 0945 2198
      0948 2199
      61 15 50 52 D0 0948 2200
      0000'DF40 50 F0 094B 2201
      0000'DF40 51 D0 0950 2202
      61 5B C8 0956 2203
      90 11 0959 2204
      095B 2205
      0000'DF40 10 88 095B 2206 60$:
      3C BB 0961 2207
      54 0000'CF D0 0963 2208
      F695' 30 0968 2209
      52 0000'CF 7E 0968 2210
      00 DD 0970 2211
      F68B' 30 0972 2212
      0975 2213
      0978 2214
      50 FC A9 3C BA 0978 2214
      5A CB 097A 2215
  
```

```

EXTZV #PTESV PFN,#PTESS_PFN,R1,R1 ; EXTRACT PFN
      PFN REFERENCE -
TSTW <@W^PFNSAx SHRcnt[R1]>,- ; CHECK FOR FIRST REFERENCE TO PTABL
      LONG OPCODE=TSTL -
      IMAGE=SYS_NONPAGED
BNEQ 35$ ; NO
BUG_CHECK GBLPAGSZRO,FATAL ; GLOBAL PAGE SHARE COUNT ZERO
      PFN REFERENCE -
INCW <@W^PFNSAx SHRcnt[R1]>,- ; RAISE GLOBAL PAGE TABLE SHARE COUN
      LONG OPCODE=INCL -
      IMAGE=SYS_NONPAGED
INCW <@W^PFNSAx SHRcnt[R0]>,- ; RAISE SHARE COUNT FOR GLOBAL PAGE
      LONG OPCODE=INCL -
      IMAGE=SYS_NONPAGED
BRW RECONNECT ; RECONNECT AND REFERENCE PAGE TABLE
EXTZV #PFNSV BAK,#PFNSS BAK,R2,@W^PFNSAL BAK[R0] ; SAVE BACKING ADDR
BICL #^C<PTESM_PROT!PTESM_OWN>,R2 ; SAVE PROTECTION AND OWNER FIELDS
BISL R1,R2 ; SET PTE VALID
BISL3 R0,R2,(R1) ; AND STORE WITH PFN IN GPT
MOVL R1,@W^PFNSAL PTE[R0] ; SET SVAGPTE IN PFN DATA BASE
MOVB #PFNSC_ACTIVE,@W^PFNSAB_STATE[R0] ; SET STATE TO ACTIVE
MOVB #PFNSC_GLOBAL,@W^PFNSAB_TYPE[R0] ; AND TYPE TO GLOBAL
BRB 30$ ; NOW GO SETUP SLAVE PTE

; PAGE READ ERROR IN GPTE
; THE PFN IN THE GPTE WILL BE DEALLOCATED
; THE GPTE WILL BE ALTERED TO USE THE PFN FROM THE INSWAP IMAGE
; THE DATA BASE WILL BE ADJUSTED AS APPROPRIATE

BICL3 R10,-4(R9),R2 ; GET SWAP IMAGE PFN.
MOVL @W^PFNSAL BAK[R0],@W^PFNSAL BAK[R2] ; COPY BACKING STORE.
MOVB #PFNSC_ACTIVE,@W^PFNSAB_STATE[R2] ; SET STATE TO ACTIVE.
MOVB #PFNSC_GLOBAL,@W^PFNSAB_TYPE[R2] ; SET TYPE TO GLOBAL.
      PFN REFERENCE - ; COPY SHARE COUNT.
MOVW <@W^PFNSAx SHRcnt[R0],@W^PFNSAx SHRcnt[R2]>,-
      LONG OPCODE=MOVL,-
      IMAGE=SYS_NONPAGED
CLRL @W^PFNSAL_PTE[R0] ; SETUP FOR AND RELEASE
BSBW RELPAGE ; READ ERROR PFN.
; NOBODY CAN USE IT.
MOVL R2,R0 ; SETUP NEW MASTER PFN.
INSV R0,#PTESV PFN,#PTESS_PFN,(R1) ; PLANT PFN IN GPTE.
MOVL R1,@W^PFNSAL_PTE[R0] ; PLANT PTE IN DATABASE.
BISL R11,(R1) ; MAKE PTE VALID.
BRB 40$ ; JOIN COMMON CODE.

BISB #PFNSM_COLLISION,@W^PFNSAB_TYPE[R0] ; FLAG COLLISION FOR PAGAREA
PUSHR #^M<R2,R3,R4,R5> ; SAVE REGS OVER WAIT
MOVL W^SCH$GL_CURPCB,R4 ; AND SET PCB ADDRESS
BSBW SCH$NEWLVL ; SET ASTLVL CORRECTLY
MOVAQ W^SCH$GQ_COLPGWQ,R2 ; GET ADDRESS OF WAIT QUEUE
PUSHL #0 ; NULL KERNEL MODE PSL
BSBW SCH$WAITK ; WAIT WITH NO CALL FRAME
SETIPL #IPL$SYNCH ; BLOCK SYSTEM EVENTS
POPR #^M<R2,R3,R4,R5> ; RESTORE REGS
BICL3 R10,-4(R9),R1 ; RESTORE CURRENT PFN
  
```

```

FEE6 31 097F 2216 BRW GLOBAL ; AND ATTEMPT TO REASSOCIATE PAGE
0982 2217
0982 2218
0982 2219
0982 2220
0982 2221
0982 2222
0982 2223
0982 2224
0986 2225
0989 2226
098C 2227
098E 2228
0990 2229
0994 2230
0996 2231
0998 2232
09A1 2233
09A5 2234
09AA 2235
09AE 2236
09B6 2237
09BA 2238
09BE 2239
09C3 2240
09C5 2241
09CC 2242
09D4 2243
09D7 2244
09D7 2245
09DC 2246
09E1 2247
09E1 2248
09E1 2249
09E7 2250
09E7 2251
09E7 2252
09EB 2253
09EE 2254

SET PROPER AST LEVEL
SETASTLVL:
MOVAL PCBSL_ASTQFL(R4),R3 ; GET POINTER TO HEAD OF AST QUEUE
MOVL (R3),R2 ; GET POINTER TO FIRST AST CONTROL BLOCK
CMPL R3,R2 ; IS LIST EMPTY?
BEQL 20$ ; YES, DONE
CLRL R0 ; ASSUME KERNEL MODE
MOVB ACBSB_RMOD(R2),R3 ; GET ACTUAL MODE
BLSS 10$ ; BR IF SPECIAL KERNEL AST
EXTZV #ACBSV_MODE,#ACBSS_MODE,R3,R0 ; GET ACCESS MODE
BICB3 PCBSB_ASTACT(R4),PCBSB_ASTEN(R4),R1 ; CHECK FOR DELIVERABILITY
BBC R0,R1,20$ ; BR IF NOT PRESENTLY DELIVERABLE
MOVB R0,PHDSB_ASTLVL(R5) ; SET AST LEVEL FOR PROCESS
BISL #<<1@PCBSV_RES>!<1@PCBSV_INQUAN>>,PCBSL_STS(R4) ; MARK PROCESS RESID
MOVW SCH$GW_QUAN,PHD$W_QUANT(R5) ; AND GIVE NEW QUANTUM
MOVZBL PCBSB_PRI(R4),R0 ; GET CURRENT PRIORITY OF PROCESS
SUBB3 R0,#3T,R1 ; COMPUTE EXTERNAL PRIORITY FOR COMPARE
CMPB R1,W^SYSS$GB_DEFPRI ; IS THIS A 'CRUNCHER'?
BGTR 30$ ; NO, CONTINUE
MOVL EXE$GQ_SYSTIME,R1 ; GET CURRENT TIME IN APPROX. 10MS UNITS
ADDL3 W^SCH$GL_SWPRATE,R1,W^SWP$GL_SWTIME ; SET NEW CRUNCHER INTERVAL
BSBW SCH$CHSEP ; CHANGE TO RESIDENT COMPUTE
SWAPRETRY: ; RETRY SWAP SCHEDULING
MOVL W^SCH$GL_CURPCB,R4 ; GET PCB ADDRESS
BBSSI #PCBSV_WAKEPEN,PCBSL_STS(R4),20$ ; SET TO CANCEL HIBER

10$:
20$:
30$:
40$:
50$:
60$:
70$:
80$:
90$:
A0$:
B0$:
C0$:
D0$:
E0$:
F0$:

.DSABL LSB
SWAPEXIT: ; EXIT SWAPPER
BBCC S^#SCH$V_SIP,W^SCH$GB_SIP,10$ ; CLEAR SWAP IN PROGRESS

10$:
SWAPEXITA: ; ALTERNATE EXIT, LEAVING SIP SET
POPR #*M<R6,R7,R8,R9,R10,R11,AP,FP> ; RESTORE REGISTERS
SETIPL #0 ; DROP IPL
RSB ;

```

```

09EF 2257      .SBTTL FILLPHD - FILL SPT ENTRIES TO MAP PHD
09EF 2258
09EF 2259      :++
09EF 2260      : FUNCTIONAL DESCRIPTION:
09EF 2261      : FILLPHD SETS THE PTE ENTRIES FOR THE PROCESS HEADER INTO THE
09EF 2262      : SPT.
09EF 2263
09EF 2264      : CALLING SEQUENCE:
09EF 2265      : BSB/JSB FILLPHD
09EF 2266
09EF 2267      : INPUT PARAMETERS:
09EF 2268      : R3 - POINTER TO FIRST SPT ENTRY FOR PHD
09EF 2269      : R9 - ADDRESS OF SWAPPER MAP ENTRY TO BE MOVED TO SPT
09EF 2270      : R10 - PTE$C_ERKW!PTE$M_VALID!PTE$M_MODIFY
09EF 2271
09EF 2272      : OUTPUT PARAMETERS:
09EF 2273      : R5 - ZERO
09EF 2274      : R6 - DESTROYED
09EF 2275      : R9 - UPDATED
09EF 2276      : R11 - DESTROYED
09EF 2277      : AP - DESTROYED
09EF 2278      : FP - DESTROYED
09EF 2279      :--
09EF 2280
09EF 2281      FILLPHD:
09EF 2282
09EF 2283      CLRL R5 ; SET PHD ADDRESS TO SWAPPER PO SPACE
09F1 2284      INVALID ; TO SEE CORRECT PROCESS HEADER IN SWAPPER P
09F4 2285      CLRL R11 ; INIT HEADER PAGE INDEX
56 00D8 C5 D4 09F6 2286      MOVZWL PHD$W_EMPTPG(R5),R6 ; GET COUNT OF EMPTY PAGES
56 56 09 78 09FB 2287      ASHL #9,R6,R6 ; CONVERT TO BYTE OFFSET
5C 48 A5 D0 09FF 2288      MOVL PHD$W_WSLX(R5),AP ; FORM BASE ADDRESS FOR WSLX
5C 654C DE 0A03 2289      MOVAL (R5)[AP],AP ; SAVE VECTOR FOR PHD
5D 44 A5 D0 0A07 2290      MOVL PHD$W_BAK(R5),FP ; FORM BASE ADDRESS FOR BACKING STORE ADDRESS
5D 654D DE 0A0B 2291      MOVAL (R5)[FP],FP ; VECTOR
5C 56 C2 0A0F 2292      SUBL R6,AP ; ACCOUNT FOR EMPTY PAGES
5D 56 C2 0A12 2293      SUBL R6,FP ; BY SUBTRACTING THEIR SPACE
56 0000'CF D0 0A15 2294      MOVL W^$WPG$GL_BSLLOTSZ,R6 ; SET ITERATION COUNT FOR ENTIRE HEADER
83 8D D0 0A1A 2295      10$: MOVL (FP)+,(R3)+ ; SET BACKUP FORM OF PTE IN SPT SLOT
50 89 5A CB 0A1D 2296      BGEQ 30$ ; DONE IF NOT VALID
0000'DF4C 73 DE 0A1F 2297      BICL3 R10,(R9)+,R0 ; GET PAGE FROM SWAPPER MAP
51 51 63 17 00 EF 0A23 2298      MOVAL -(R3),@W^PFNSAL_PTE[R0] ; SET PTE BACK POINTER
08 18 1F A5 F0 0A29 2299      EXTZV #PFNSV_BAK,#PFNSB_BAK,(R3),R1 ; ISOLATE BACKING STORE ADDRESS
0000'DF40 51 D0 0A2E 2300      INSV PHD$B_PAGFIL(R5),#PFNSV_PGFLX,#PFNSB_PGFLX,R1 ; ADD FILE NUMBER
0A34 2301      MOVL R1,@W^PFNSAL_BAK[R0] ; SAVE IN PFN DATA BASE
0A3A 2302      PFN REFERENCE -
0A3A 2303      MOVW <(AP)[R11],@W^PFNSAx_WSLX[R0]>,- ; SAVE WORKING SET LIST INDE
0A3A 2304      LONG OPCODE=MOVZWL,-
0A3A 2305      IMAGE=SYS_NONPAGED
83 5A 50 C9 0A41 2306      BISL3 R0,R10,(R3)+ ; SET VALID PTE FOR PAGE
0000'DF40 87 8F 90 0A45 2307      MOVB #<PFNSC_ACTIVE!PFNSM_MODIFY>,@W^PFNSAB_STATE[R0] ; MARK PAGE ACTIVE
0000'DF40 04 90 0A4C 2308      MOVB #PFNSC_PPGTBL,@W^PFNSAB_TYPE[R0] ; STORE TYPE IN PFN DATA BAS
C4 5B 56 F2 0A52 2309      30$: AOBLSS R6,R11,10$ ; FILL ENTIRE PROCESS HEADER
05 0A56 2310      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 (SWPSAL_MAP)
0A57 2331      : R10 - PTE$C ERKW!PTE$M_VALID!PTE$M_MODIFY
0A57 2332      : R11 - BASE ADDRESS FOR SWAPPER MAP (SWPSAL_MAP)
0A57 2333
0A57 2334      :--
0A57 2335
0A57 2336      RELINIT:
54 0014'CF DO 0A57 2337      MOVL W^OSWPPCB,R4      ; RELEASE LOOP INITIALIZATION
57 0012'CF 3C 0A5C 2338      MOVZWL W^OSWPPGS,R7      ; GET PCB ADDRESS OF OUT SWAP PROCESS
0A61 2339      BRB OSINIT      ; AND PAGE COUNT FOR RELEASE LOOP
                                ; FALL INTO OSINIT

```

```

0A61 2342      .SBTTL OSINIT - OUTSWAP SCAN REGISTER INITIALIZATION
0A61 2343
0A61 2344      :++
0A61 2345      : FUNCTIONAL DESCRIPTION:
0A61 2346      : OSINIT SETS UP REGISTERS FOR PAGE TABLE SCANS REQUIRED DURING
0A61 2347      : OUTSWAPPING.
0A61 2348
0A61 2349      : INPUT PARAMETERS:
0A61 2350      : NONE
0A61 2351
0A61 2352      : OUTPUT PARAMETERS:
0A61 2353      : R9 - BASE ADDRESS OF SWAPPER MAP (SWPSAL_MAP)
0A61 2354      : R10 - PTE$C_ERKW!PTE$M_VALID
0A61 2355      : R11 - BASE ADDRESS OF SWAPPER MAP (SWPSAL_MAP)
0A61 2356
0A61 2357      :--
0A61 2358
0A61 2359 OSINIT:
59 0000'DF DE 0A61 2360 MOVAL @W^SWPSGL_MAP,R9 ; SET BASE OF SWAPPER MAP
5B 59 DO 0A66 2361 MOVL R9,R11 ; AND MAKE REFERENCE COPY
SA B4000000 8F DO 0A69 2362 MOVL #<PTE$C_ERKW!PTE$M_VALID!PTE$M_MODIFY>,R10 ; MASK TO VALIDATE SWAP P
05 0A70 2363 RSB ; RETURN

```

```

0A71 2366      .SBTTL RELPAGE - RELEASE DUPLICATE PAGE
0A71 2367
0A71 2368      :++
0A71 2369      : FUNCTIONAL DESCRIPTION:
0A71 2370      : RELPAGE RELEASES A PHYSICAL PAGE WHICH DUPLICATES A PAGE ALREADY
0A71 2371      : PRESENT FOR THE PROCESS. THIS SITUATION CAN ARISE DUE TO A PARTIAL
0A71 2372      : INSWAP OR A GLOBAL PAGE WHICH IS ALREADY PRESENT.
0A71 2373
0A71 2374      : CALLING SEQUENCE:
0A71 2375      : BSB/JSB RELPAGE
0A71 2376
0A71 2377      : INPUT PARAMETERS:
0A71 2378      : R0 - PFN TO RELEASE
0A71 2379      : R3 - SVA OF PTE (RELDELPAGE ONLY)
0A71 2380
0A71 2381      : OUTPUT PARAMETERS:
0A71 2382      : R1 - PRESERVED (RELPAGE ONLY)
0A71 2383      : R2 - PRESERVED (RELPAGE ONLY)
0A71 2384      : R3 - PRESERVED (RELPAGE ONLY)
0A71 2385
0A71 2386      :--
0A71 2387
0A71 2388      RELDELPAGE:
0A71 2389      BSBW      MMG$DELCONPFN      : RELEASE PAGE THROUGH DELCONPFN
0A74 2390      RELPAGE:      : DELETE PAGE CONTENT AND INIT PFN DATA
0A74 2391      PUSHHR   #^M<R1,R2,R3>      : RELEASE PAGE
0A76 2392      CLRB     @W^PFNS$AB_STATE[R0]  : PRESERVE REGISTERS
0A7B 2393      CLRW     @W^PFNS$AW_REFcnt[R0]  : INIT PFN DATA FOR RELEASE
0A80 2394      ASSUME   PFNSC_FREPAGLST EQ 0  : ZERO REFERENCE COUNT
0A80 2395      CLRL     R2                        : INDICATE FREELIST
0A82 2396      BSBW     MMG$INSPFNH              : RELEASE PFN TO HEAD OF FREE LIST
0A85 2397      POPR     #^M<R1,R2,R3>          : RESTORE REGISTERS
0A87 2398      RSB                          : AND RETURN TO CALLER

```

```

F58C' 30
OE BB
0000'DF40 94
0000'DF40 B4
52 D4
F57B' 30
OE BA
05 OA87

```

```

0A88 2401          .SBTTL  SWPREAD/SWPWRITE - SWAPPER I/O ROUTINES
0A88 2402
0A88 2403 :++
0A88 2404 : FUNCTIONAL DESCRIPTION:
0A88 2405 : SWPREAD AND SWPWRITE PERFORM THE DETAIL WORK REQUIRED TO READ
0A88 2406 : OR WRITE A SET OF CONTIGUOUS PAGES IN A WORKING SET SWAP IMAGE.
0A88 2407 : THE CALL TO EITHER SWPREAD OR SWPWRITE IS ACTUALLY A CO-ROUTINE
0A88 2408 : CALL WHICH RETURNS ONLY AFTER ALL SEGMENTS OF THE I/O OPERATION
0A88 2409 : HAVE BEEN PERFORMED. THIS RETURN IS EFFECTED BY A SPECIAL KERNEL
0A88 2410 : AST.
0A88 2411
0A88 2412 : CALLING SEQUENCE:
0A88 2413 : BSB/JSB SWPREAD/SWPWRITE
0A88 2414
0A88 2415 : INPUT PARAMETERS:
0A88 2416 : R0 - SWAP FILE INDEX
0A88 2417 : R2 - WSSWP FORM DISK ADDRESS
0A88 2418 : R3 - SYSTEM VIRTUAL ADDRESS OF PTE
0A88 2419 : R4 - PAGE COUNT
0A88 2420
0A88 2421 : 00(SP) - RETURN ADDRESS AFTER I/O COMPLETION
0A88 2422 : 04(SP) - SAVED R6
0A88 2423 : 08(SP) - SAVED R7
0A88 2424 : 12(SP) - SAVED R8
0A88 2425 : 16(SP) - SAVED R9
0A88 2426 : 20(SP) - SAVED R10
0A88 2427 : 24(SP) - SAVED R11
0A88 2428 : 28(SP) - SAVED AP
0A88 2429 : 32(SP) - SAVED FP
0A88 2430 : 36(SP) - SAVED IPL
0A88 2431 : 40(SP) - RETURN TO PREVIOUS THREAD
0A88 2432
0A88 2433 : IMPLICIT INPUTS:
0A88 2434 : PAGE FILE TABLE ENTRY (PFL) SELECTED BY WSSWP INPUT
0A88 2435
0A88 2436 : OUTPUT PARAMETERS:
0A88 2437 : R0 - COMPLETION STATUS OF I/O OPERATION
0A88 2438 :
0A88 2439 :--
0A88 2440
0A88 2441          .ENABL  LSB
0A88 2442 SWPREAD:          PUSHAB  W*EXESBLDPKTSWPR          : SWAP READ INITIATION
0A88 2443          BRB      10$          : SET ADDRESS OF BUILD PACKET ROUTINE
0A88 2444 SWPWRITE:          PUSHAB  W*EXESBLDPKTSWPW          : SWAP WRITE INITIATION
0A88 2445          BRB      10$          : SET ADDRESS OF BUILD PACKET ROUTINE
51 0000'CF 9F 0A8E 2446 10$:  MOVAB  W*IOROUTINE,R1          : ADDRESS OF I/O DATA
81 0000'CF 9E 0A92 2447 10$:  MOVQ   (SP)+,(R1)+          : SAVE I/O END ACTION ADDRESS
52 08 8E 7D 0A97 2448 10$:  POPR   #*M<R6,R7,R8,R9,R10,R11,AP,FP>: RESTORE REGISTERS OTHER THAN STANDAR
3FC0 8F BA 0A9A 2449 15$:  EXTZV #24,#8,R2,R0          : GET $WAP FILE INDEX
50 52 08 18 EF 0A9E 2450 15$:  MOVL  @W*MMG$GL_PAGSWPVC[R0],R0; GET BASE ADDRESS OF PAGE FILE TABLE
50 55 0000'DF40 D0 0AA3 2451 15$:  MOVZBL #127,R5          : SET I/O SIZE
55 7F 8F 9A 0AA9 2452 15$:  CMPL  R4,R5          : COMPARE REMAINING PGCNT WITH MAX TRANSFER
55 54 D1 0AAD 2453 15$:  BGTR  20$          : USE MAXIMUM TRANSFER
55 03 14 0AB0 2454 15$:  MOVL  R4,R5          : SET TRANSFER TO REMAINING PAGES
81 55 54 D0 0AB2 2455 20$:  ADDL3 R5,R2,(R1)+          : SAVE UPDATED DISK ADDRESS
81 52 55 C1 0AB5 2456 20$:  MOVAL (R3)[R5],(R1)+          : AND UPDATED SAVPTE
81 6345 DE 0AB9 2457

```

					OABD	2458		SETIPL	#0	:	DROP IPL
	61	54	55	A3	OACO	2459		SUBW3	R5,R4,(R1)	:	SAVE REMAINING PAGE COUNT
			53	DD	OAC4	2460		PUSHL	R3	:	SAVE SVAPTE
			OC A0	DD	OAC6	2461		PUSHL	PFL\$WINDOW(R0)	:	GET WINDOW ADDRESS
	7E	55	09	9C	OAC9	2462		ROTL	#9,R5--(SP)	:	CONVERT PAGES TO BYTE COUNT
7E	52	18	00	EF	OACD	2463		EXTZV	#0,#24,R2--(SP)	:	AND ISOLATE BLOCK NUMBER
		6E	10 A0	CO	OAD2	2464		ADDL	PFL\$VBN(R0),(SP)	:	ADD BASE VBN
	54	0000	'CF	DO	OAD6	2465		MOVL	W\$SCH\$GL CURPCB,R4	:	SET PCB ADDRESS
	55	0000	'DF	OF	OADB	2466		REMQUE	@W\$IOCS\$G_IRPFL,R5	:	GET A PACKET IF POSSIBLE
			06	1C	OAE0	2467		BVC	30\$	:	BR IF ONE AVAILABLE
			F51B'	30	OAE2	2468		BSBW	EXE\$ALLOCIRP	:	ALLOCATE ONE THE LONG WAY
			55 52	DO	OAE5	2469		MOVL	R2,R5	:	SET PACKET ADDRESS IN PROPER REGISTER
23 A5	14 A5	FB	'AF	9E	OAE8	2470	30\$:	MOVAB	B\$IODONE,IRP\$L ASTPRM(R5)	:	SET ADDRESS FOR COMPLETION
	1F	0000	'CF	83	OAE D	2471		SUBB3	W\$SWP\$GB PRI0,#31,IRP\$B_PRI(R5)	:	SET PRIORITY FOR TRANSFER
			OF	BA	OAF4	2472		POPR	#*M<R0,RT,R2,R3>	:	RESTORE VBN,BYTECNT,WINDOW,SVAPTE
			0000'DF	16	OAF6	2473		JSB	@W\$IOROUTINE	:	CALL READ OR WRITE ROUTINE
				05	OAF A	2474		RSB		:	AND RETURN TO ORIGINAL CALLER
					OAFB	2475				:	
					OAFB	2476	IODONE:			:	CONTINUATION CALLED AS KERNEL AST
			38 A5	DD	OAFB	2477		PUSHL	IRP\$L_MEDIA(R5)	:	SAVE COMPLETION STATUS
			50 55	DO	OAFE	2478		MOVL	R5,R0	:	SET PACKET ADDRESS FOR RELEASE
			F4FC'	30	OB01	2479		BSBW	EXE\$DEANONPAGED	:	AND RELEASE IT
			50 8E	DO	OB04	2480		MOVL	(SP)+,R0	:	RESTORE STATUS
					OB07	2481		SETIPL	#IPL\$ SYNCH	:	BLOCK SYSTEM EVENTS
			11 50	E9	OB0A	2482		BLBC	R0,60\$	:	EXIT IF ERROR
	51	0008	'CF	9E	OB0D	2483		MOVAB	W\$RWSSWP,R1	:	GET ADDRESS OF REMAINING TRANSFER PARAMS
		52	61	7D	OB12	2484		MOVQ	(R1),R2	:	RESTORE WSSWP,SVAPTE TO R2,R3
	54	0B	'A1	3C	OB15	2485		MOVZWL	B^<RPGCNT-RWSSWP>(R1),R4	:	AND REMAINING PAGE COUNT
			03	13	OB19	2486		BEQL	60\$	:	DONE IF NO MORE PAGES REMAIN
			FF80	31	OB1B	2487		BRW	15\$	:	CONTINUE IF MORE PAGES REMAIN
			3FC0 8F	BB	OB1E	2488	60\$:	PUSHR	#*M<R6,R7,R8,R9,R10,R11,AP,FP>	:	SAVE NON-STANDARD REGISTERS
			0004'DF	17	OB22	2489		JMP	@W\$IOEA	:	AND CONTINUE SWAP
					OB26	2490				:	
					OB26	2491		.DSABL	LSB	:	
					OB26	2492		.END		:	



SWAPPER  
Symbol table

WORKING SET SWAPPER

N 6

16-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1

\$\$ARGS	= 00000005			GBLWRTVALID	000004CF	R	05
\$\$T1	= 00000018			GLOBAL	00000868	RR	05
...PFN	= 00000A3A	R	05	IMGDESC	00000000	RR	04
ACBSB_RMOD	= 0000000B			INSWAP	00000535	R	05
ACBSB_MODE	= 00000002			IOC\$GL_IRPFL	*****	X	05
ACBSV_MODE	= 00000000			IODONE	00000AFB	RR	05
BALANCE	00000030	R	05	IOEA	00000004	RR	02
BDLSL_SYSDLOG	*****	X	04	IOROUTINE	00000000	R	02
BDLSS_CRELNM_ITMLST	*****	X	04	IPLS_ASTDEL	= 00000002		
BUGS_APTREFHIGH	*****	X	05	IPLS_SYNCH	= 00000008		
BUGS_APTWRTERR	*****	X	05	IRPSB_PRI	= 00000023		
BUGS_GBLPAGSZRO	*****	X	05	IRPSL_ASTPRM	= 00000014		
BUGS_ICPAGELOC	*****	X	05	IRPSL_MEDIA	= 00000038		
BUGS_INSNFREPAGE	*****	X	05	KERNEC_MODE	000001CD	R	04
BUGS_INSSWPFIL	*****	X	05	LNMSAL_HASHTBL	*****	X	04
BUGS_INSWAPERR	*****	X	05	LNMSGH_HTBLSIZS	*****	X	04
BUGS_IWSETLIST	*****	X	05	LNMSHASH	*****	X	04
BUGS_OUTSWPERR	*****	X	05	LNMSINSLOGTAB	*****	X	04
BUGS_QUEUEEMPTY	*****	X	05	LNMSM_NO_ALIAS	= 00000001		
BUGS_ZEROPAGE	*****	X	05	LNMSM_TERMINAL	= 00000200		
COPYSHLL	000005DF	R	05	LNMSYSTEM_DIRECTORY	00000000	RG	03
CRELNMS_ACMODE	= 00000010			LNMS_ATTRIBUTES	= 00000003		
CRELNMS_ATTR	= 00000004			LNMS_STRING	= 00000002		
CRELNMS_ITMLST	= 00000014			LNMSB_ACMODE	= 0000000B		
CRELNMS_LOGNAM	= 0000000C			LNMSB_FLAGS	= 00000010		
CRELNMS_NARGS	= 00000005			LNMSB_TYPE	= 0000000A		
CRELNMS_TABNAM	= 00000008			LNMSBL_BLINK	= 00000004		
DELCON	0000051F	R	05	LNMSBL_FLINK	= 00000000		
DELPHD	00000404	RR	05	LNMSBL_TABLE	= 0000000C		
DIRECTORIES_ARG	0000032D	RR	04	LNMSM_NODELETE	= 00000010		
DIRECTORIES_LIST	000001DD	R	04	LNMSM_NO_ALIAS	= 00000001		
DYN\$C_LNM	= 00000040			LNMSM_TABLE	= 00000008		
DYN\$C_ORB	= 00000049			LNMSM_NAME	= 00000011		
DYN\$C_PCB	= 0000000C			LNMSM_SIZE	= 00000008		
DYN\$C_RSHT	= 00000038			LNMSM\$B_TYPE	= 0000000A		
EXE\$A[LOCIRP	*****	X	05	LNMSM\$C_BUCKET	= 0000000C		
EXE\$A[OPAGED	*****	X	04	LNMSM\$K_BUCKET	= 0000000C		
EXE\$B[DPKTSWPR	*****	X	05	LNMSM\$SL_MASK	= 00000000		
EXE\$B[DPKTSWPW	*****	X	05	LNMSM\$W_SIZE	= 00000008		
EXE\$DEANONPAGED	*****	X	05	LNMT\$B_FLAGS	= 00000000		
EXE\$DEANONPGDSIZ	*****	X	04	LNMT\$K_LENGTH	= 00000025		
EXE\$GL_PAGED	*****	X	04	LNMT\$SL_BYTES	= 00000021		
EXE\$GL_PFATIM	*****	X	05	LNMT\$SL_BYTESLM	= 0000001D		
EXE\$GQ_SYSTIME	*****	X	05	LNMT\$SL_CHILD	= 00000011		
EXE\$PODERAST	*****	X	05	LNMT\$SL_HASH	= 00000001		
EXE\$SWAPINIT	000004AC	RG	04	LNMT\$SL_NAME	= 00000009		
EXEC_MODE	000001C9	R	04	LNMT\$SL_ORB	= 00000005		
FILE_DEV_EXEC_ARG	00000345	R	04	LNMT\$SL_PARENT	= 0000000D		
FILE_DEV_EXEC_LIST	00000239	R	04	LNMT\$SL_QTABLE	= 00000019		
FILE_DEV_SUPER_ARG	0000035D	R	04	LNMT\$SL_SIBLING	= 00000015		
FILE_DEV_SUPER_LIST	00000205	R	04	LNMT\$M_DIRECTORY	= 00000002		
FILLPHD	000009EF	R	05	LNMT\$M_SHAREABLE	= 00000001		
GBLDROP	00000462	R	05	LNMT\$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	= FFFFFFF82		
GBLWRTTRANS	0000045A	R	05	LNMX\$M_TERMINAL	= 00000002		

SWAPPER  
Symbol table

WORKING SET SWAPPER

B 7

16-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1

Page 55  
(28)

SYS  
V04

LNMYSM_XEND	= 00000004			MMGSM_NOWAIT	*****	X	05
LNMYSM_XLATION	= 00000004			MMG\$REFCNTNEG	*****	X	05
LNMYSW_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\$WRMFPYAG	*****	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	OPS_CMP1	= 000000D1		
LNM_PROCESS_DIRECTORY	00000168	R	04	OPS_CMPW	= 000000B1		
LNM_PROCESS_DIRECTORY_LENGTH	= 00000015			OPS_CVTLW	= 000000F7		
LNM_PROCESS_LENGTH	= 0000000B			OPS_DECL	= 000000D7		
LNM_SYSTEM	0000017D	R	04	OPS_DECLW	= 000000B7		
LNM_SYSTEM_DESC	00000068	R	04	OPS_INCL	= 000000D6		
LNM_SYSTEM_DIRECTORY	00000187	R	04	OPS_INCW	= 000000B6		
LNM_SYSTEM_DIRECTORY_DESC	00000070	R	04	OPS_MOVL	= 000000D0		
LNM_SYSTEM_DIRECTORY_LENGTH	= 00000014			OPS_MOVW	= 000000B0		
LNM_SYSTEM_DIR_LNMTH	0000002B	RG	03	OPS_MOVZW	= 0000003C		
LNM_SYSTEM_DIR_ORB	00000058	R	03	OPS_TSTL	= 000000D5		
LNM_SYSTEM_LENGTH	= 0000000A			OPS_TSTW	= 000000B5		
LNM_SYSTEM_TABLE	0000019B	R	04	ORBSB_FLAGS	= 0000000B		
LNM_SYSTEM_TABLE_LENGTH	= 00000010			ORBSB_TYPE	= 0000000A		
LNM_SYS_DIR_ORB_SIZ	= 00000068			ORBSK_LENGTH	= 00000058		
LNM_SYS_DIR_SIZ	= 000000C0			ORBSL_ACL_COUNT	= 00000028		
LNM_TEMPORARY_MAILBOX_DESC	00000078	R	04	ORBSL_ACL_DESC	= 0000002C		
LOG_GROUP	000001AB	R	04	ORBSL_ACL_MUTEX	= 00000004		
LOG_GROUP_LENGTH	= 00000009			ORBSL_GRP_PROT	= 00000020		
LOG_G_ARG	00000375	R	04	ORBSL_OWNER	= 00000000		
LOG_G_DESC	00000095	R	04	ORBSL_OWN_PROT	= 0000001C		
LOG_G_LIST	00000249	R	04	ORBSL_SYS_PROT	= 00000018		
LOG_PROCESS	000001B4	R	04	ORBSL_WOR_PROT	= 00000024		
LOG_PROCESS_LENGTH	= 0000000B			ORBSQ_MODE_PROT	= 00000010		
LOG_P_ARG	0000038D	R	04	ORBSR_MAX_CLASS	= 00000044		
LOG_P_DESC	0000009D	R	04	ORBSR_MIN_CLASS	= 00000030		
LOG_P_LIST	00000259	R	04	ORBSS_MAX_CLASS	= 00000014		
LOG_SYSTEM	000001BF	R	04	ORBSS_MIN_CLASS	= 00000014		
LOG_SYSTEM_LENGTH	= 0000000A			ORBSW_REF_COUNT	= 0000000E		
LOG_S_ARG	000003A5	R	04	ORBSW_SIZE	= 0000000B		
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\$DACPAGFIL	*****	X	05	OUTSWAP	00000116	R	05
MMG\$DECPTRF	*****	X	05	OVSLOOP	00000215	R	05
MMG\$DELCONPFN	*****	X	05	P1SYSVECTORS	*****	X	04
MMG\$DELWSLEX	*****	X	05	PCBSB_ASTACK	= 0000000C		
MMG\$GB_FREWFLGS	*****	X	05	PCBSB_ASTEN	= 0000000D		
MMG\$GL_GPTBASE	*****	X	05	PCBSB_PRI	= 0000000B		
MMG\$GL_PAGSWPVC	*****	X	05	PCBSB_TYPE	= 0000000A		
MMG\$GL_SPTBASE	*****	X	05	PCBSL_ASTOFL	= 00000010		
MMG\$INSPFNH	*****	X	05	PCBSL_PHD	= 0000006C		
MMG\$INSPFNT	*****	X	05	PCBSL_PHYPCB	= 00000018		
MMG\$IOLOCKPAG	*****	X	05	PCBSL_PID	= 00000060		

SWAPPER  
Symbol table

WORKING SET SWAPPER

C 7

16-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1

Page 56  
(28)

SYS  
V04

PCBSL_STS	=	00000024		
PCBSL_WSSWP	=	00000020		
PCBSV_INQUAN	=	00000003		
PCBSV_PHDRES	=	00000012		
PCBSV_RES	=	00000000		
PCBSV_WAKEPEN	=	0000000C		
PCBSW_APTCNT	=	00000030		
PCBSW_GPGCNT	=	00000034		
PCBSW_PPGCNT	=	00000036		
PERMANENT_MAILBOX_ARG		0000038D	R	04
PERMANENT_MAILBOX_LIST		00000239	R	04
PFLSL_VBN	=	00000010		
PFLSL_WINDOW	=	0000000C		
PFNSAB_STATE		*****	X	05
PFNSAB_TYPE		*****	X	05
PFNSAL_BAK		*****	X	05
PFNSAL_HEAD		*****	X	05
PFNSAL_PTE		*****	X	05
PFNSAW_REFCNT		*****	X	05
PFNSAW_SUPVBN		*****	X	05
PFNSAX_FLINK		*****	X	05
PFNSAX_SHRCNT		*****	X	05
PFNSAX_WSLX		*****	X	05
PFNSC_ACTIVE	=	00000007		
PFNSC_BADPAGLST	=	00000002		
PFNSC_FREPAGLST	=	00000000		
PFNSC_GBLWRT	=	00000003		
PFNSC_GLOBAL	=	00000002		
PFNSC_GPGTBL	=	00000005		
PFNSC_MFY PAGLST	=	00000001		
PFNSC_PPGTBL	=	00000004		
PFNSC_PROCESS	=	00000000		
PFNSC_RDERR	=	00000004		
PFNSC_RDINPROG	=	00000006		
PFNSC_RELPEND	=	00000003		
PFNSC_SYSTEM	=	00000001		
PFNSC_WRTINPROG	=	00000005		
PFNSM_COLLISION	=	00000010		
PFNSM_DELCON	=	00000010		
PFNSM_MODIFY	=	00000080		
PFNSS_BAK	=	00000017		
PFNSS_LOC	=	00000003		
PFNSS_PAGTYP	=	00000003		
PFNSS_PGFLX	=	00000008		
PFNSV_BADPAG	=	00000005		
PFNSV_BAK	=	00000000		
PFNSV_LOC	=	00000000		
PFNSV_MODIFY	=	00000007		
PFNSV_PAGTYP	=	00000000		
PFNSV_PGFLX	=	00000018		
PHDSB_ASTLVL	=	000000CF		
PHDSB_PAGFIL	=	0000001F		
PHDSL_BAK	=	00000044		
PHDSL_POBR	=	000000C8		
PHDSL_P1BR	=	000000D0		
PHDSL_PCB	=	00000078		
PHDSL_WSLX	=	00000048		

PHDSV_NOACCVIO	=	00000003		
PHDSW_EMPTPG	=	00000008		
PHDSW_FLAGS	=	00000036		
PHDSW_PHVINDE	=	00000042		
PHDSW_PTCNTACT	=	00000070		
PHDSW_QUANT	=	0000003C		
PHDSW_SWAPSIZE	=	00000052		
PHDSW_WSLAST	=	00000012		
PHDSW_WSLIST	=	00000008		
PHVSGE_PIXBAS		*****	X	05
PHVSGE_REFCBAS		*****	X	05
PPGTBLTRANS		0000052A	R	05
PPGTBLVALID		0000052A	R	05
PQLSAB_SYSPQL		00000465	R	04
PQLSC_SYSPQLLEN	=	00000046	R	G
PQLS_ASTLM	=	00000001		
PQLS_BIOLM	=	00000002		
PQLS_BYTLM	=	00000003		
PQLS_CPULM	=	00000004		
PQLS_DIOLM	=	00000005		
PQLS_ENQLM	=	0000000C		
PQLS_FILLM	=	00000006		
PQLS_JTQUOTA	=	0000000E		
PQLS_LISTQUO	=	00000000		
PQLS_PGFLQUOTA	=	00000007		
PQLS_PRCLM	=	00000008		
PQLS_TQELM	=	00000009		
PQLS_WSDEFAULT	=	00C0000B		
PQLS_WSEXTENT	=	0000000D		
PQLS_WSQUOTA	=	0000000A		
PRS_TPL	=	00000012		
PRS_TBIA	=	00000039		
PRCSM_NOACNT	=	00000008		
PRCSM_SSRWAIT	=	00000001		
PROCDROP		00000478	R	05
PROCTRANS		000004DE	R	05
PROCVALID		000004F0	R	05
PROCWRT		00000258	R	05
PSLSC_EXEC	=	00000001		
PSLSC_KERNEL	=	00000000		
PSLSC_SUPER	=	00000002		
PTESC_ERKW	=	30000000		
PTESC_URKW	=	70000000		
PTESM_MODIFY	=	04000000		
PTESM_OWN	=	01800000		
PTESM_PFN	=	001FFFFF		
PTESM_PROT	=	78000000		
PTESM_TYPO	=	00400000		
PTESM_TYPI	=	04000000		
PTESM_VALID	=	80000000		
PTESS_GPTX	=	00000016		
PTESS_FFN	=	00000015		
PTESS_PGFLVB	=	00000016		
PTESV_GPTX	=	00000000		
PTESV_MODIFY	=	0000001A		
PTESV_PFN	=	00000000		
PTESV_PGFLVB	=	00000000		

SWAPPER  
Symbol table

WORKING SET SWAPPER

D 7

16-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1

Page 57  
(28)

SYS  
V04

PTESV_TYPO	= 00000016			SWPSGL_ISPAGCNT	*****	X	05
PTESV_TYP1	= 0000001A			SWPSGL_ISWPCNT	*****	X	05
PTESV_VALID	= 0000001F			SWPSGL_ISWPPAGES	*****	X	05
QEMPTY	0000008C	R	05	SWPSGL_MAP	*****	X	05
RECONNECT	00000778	R R	05	SWPSGL_OSWPCNT	*****	X	05
REDELPAGE	00000A71	R R	05	SWPSGL_PHDBASVA	*****	X	05
RELINIT	00000A57	R R	05	SWPSGL_SHELIO	*****	X	05
RELPAGE	00000A74	R R	05	SWPSGL_SHELLBAS	*****	X	05
RELPHD	00000316	R R R	05	SWPSGL_SWTIME	*****	X	05
RPGCNT	00000010	R R R	02	SWPSGW_BALCNT	00000018	RG	02
RSVAPTE	000C000C	R R	02	SWPSGW_IBALSETX	*****	X	05
RWSSWP	00000008	R	02	SWPS_SHELINIT	*****	X	05
SCHSAQ_COMOH	*****	X	05	SWPREAD	00000A88	R	05
SCHSCHSEP	*****	X	05	SWWRITE	00000A8E	R	05
SCHSGB_SIP	*****	X	05	SYSSCRELMN	*****	X	04
SCHSGL_COMOQS	*****	X	05	SYSSCREPRC	*****	X	04
SCHSGL_CURPCB	*****	X	05	SYSSGB_DEFPRI	*****	X	05
SCHSGL_FREECNT	*****	X	05	SYSTEM_ARG	000003D5	R	04
SCHSGL_FREELIM	*****	X	05	SYSTEM_LIST	00000285	R R	04
SCHSGL_MFYCNT	*****	X	05	SYSTEM_TABLE	000000C0	R R	03
SCHSGL_MFY LIM	*****	X	05	SYSTEM_TABLE_LNMTN	000000E7	R R	03
SCHSGL_MFYLOLIM	*****	X	05	SYSTEM_TABLE_ORB	00000110	R	03
SCHSGL_PCBVEC	*****	X	05	SYSTEM_TABLE_ORB_SIZ	= 00000070		
SCHSGL_SWPRATE	*****	X	05	SYSTEM_TABLE_SIZE	= 000000C0		
SCHSGQ_COLPGWQ	*****	X	05	SYS_DISK_ARG	00000180	R	03
SCHSGQ_HIBWQ	*****	X	05	SYS_DISK_DESC	000000AD	R R	04
SCHSGW_DELPHDCT	*****	X	05	SYS_SYSDEVICE_ARG	00000198	R R	03
SCHSGW_QUAN	*****	X	05	SYS_SYSDEVICE_DESC	000000BD	R R	04
SCHSGW_SWPFAIL	*****	X	05	TEMPORARY_MAILBOX_ARG	000003ED	R R	04
SCHSGW_SWPFCNT	0000001A	RG	02	TEMPORARY_MAILBOX_LIST	000002A1	R R	04
SCHSNEQLVL	*****	X	05	TERMINAL_BUFFER	000001D9	R	04
SCHSOSWPSCHED	*****	X	05	TMP...	= 00000001		
SCHSV_MPW	*****	X	05	TRNLOG_GS_ARG	00000405	R	04
SCHSV_SIP	*****	X	05	TRNLOG_GS_DESC	000000D2	R R	04
SCHSWAITK	*****	X	05	TRNLOG_GS_LIST	000002B1	R R	04
SETASTLVL	00000982	R R	05	TRNLOG_PGS_ARG	0000044D	R R	04
SETUP	00000668	R R	05	TRNLOG_PGS_DESC	00000129	R R	04
SETWRTBAK	0000050E	R	05	TRNLOG_PGS_LIST	00C^9305	R R	04
SGNSGL_BALSETCT	*****	X	05	TRNLOG_PG_ARG	000u041D	R R	04
SGNSGL_FREEGOAL	*****	X	05	TRNLOG_PG_DESC	000000EE	R R	04
SGNSGL_FREELIM	*****	X	05	TRNLOG_PG_LIST	000002CD	R R	04
SGNSGL_PAGEDYN	*****	X	04	TRNLOG_PS_ARG	00000435	R R	04
SGNSGL_PHDPAGCT	*****	X	05	TRNLOG_PS_DESC	0000010B	R R	04
SPACEFAIL	00000254	R	05	TRNLOG_PS_LIST	000002E9	R R	04
SUPER_MODE	000001D1	R R	04	TTODESC	00000013	R	04
SWAPEXIT	000009E1	R R	05	VASS_VPN	= 00000015		
SWAPEXITA	000009E7	R R	05	VASV_SYSTEM	= 0000001F		
SWAPRETRY	000009D7	R R	05	VASV_VPN	= 00000009		
SWAPSCHEM	00000090	R	05	WLSM_SYSTEM	= 00000002		
SWPSGB_ISWPRI	*****	X	05	WLSM_MODIFY	= 00000100		
SWPSGB_PRI0	*****	X	05	WLSM_PAGTYP	= 0000000E		
SWPSGL_BALBASE	*****	X	05	WLSM_PFNLOCK	= 00000010		
SWPSGL_BALSPT	*****	X	05	WLSM_VALID	= 00000001		
SWPSGL_BSLOTSZ	*****	X	05	WLSM_WSLOCK	= 00000020		
SWPSGL_HISWPCNT	*****	X	05	WLSV_MODIFY	= 00000008		
SWPSGL_HOSWPCNT	*****	X	05	WLSV_PAGTYP	= 00000001		
SWPSGL_INPCB	*****	X	05	WLSV_PFNLOCK	= 00000004		

SWAPPER  
Symbol table

WORKING SET SWAPPER

E 7

'6-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1

Page 5A  
(28)

SYS  
V04

WSLSV_VALID	=	00000000		
WSLSV_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
\$ABSS	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
YFSLOWUSE	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:[SYS.LIB]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 MSRCS:SWAPPER/UPDATE=(ENHS:SWAPPER)+EXECMLS/LIB

