


```

000000  SSSSSSSS  WW  WW  PPPPPPP  SSSSSSSS  CCCCCCCC  HH  HH  EEEEEEEEE  DDDDDDDD
000000  SSSSSSSS  WW  WW  PPPPPPP  SSSSSSSS  CCCCCCCC  HH  HH  EEEEEEEEE  DDDDDDDD
00 00  SS  WW  WW  PP  PP  SS  CC  HH  HH  EE  DD  DD
00 00  SS  WW  WW  PP  PP  SS  CC  HH  HH  EE  DD  DD
00 00  SS  WW  WW  PP  PP  SS  CC  HH  HH  EE  DD  DD
00 00  SS  WW  WW  PP  PP  SS  CC  HH  HH  EE  DD  DD
00 00  SSSSSS  WW  WW  PPPPPPP  SSSSSS  CC  HH  HH  EE  DD  DD
00 00  SSSSSS  WW  WW  PPPPPPP  SSSSSS  CC  HH  HH  EE  DD  DD
00 00  SS  WW  WW  PP  SS  CC  HH  HH  EE  DD  DD
00 00  SS  WW  WW  PP  SS  CC  HH  HH  EE  DD  DD
00 00  SS  WWW  WWW  PP  SS  CC  HH  HH  EE  DD  DD
00 00  SS  WWW  WWW  PP  SS  CC  HH  HH  EE  DD  DD
000000  SSSSSSSS  WW  WW  PPP  SSSSSSSS  CCCCCCCC  HH  HH  EEEEEEEEE  DDDDDDDD
000000  SSSSSSSS  WW  WW  PPP  SSSSSSSS  CCCCCCCC  HH  HH  EEEEEEEEE  DDDDDDDD

```

```

LL 111111 SSSSSSSS
LL 111111 SSSSSSSS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SSSSSS
LL 11 SSSSSS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SS
LLLLLLLLLL 111111 SSSSSSSS
LLLLLLLLLL 111111 SSSSSSSS

```



```

0000 1 .TITLE OSWPSCHED - SWAP SCHEDULER
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 9 * ALL RIGHTS RESERVED. *
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 16 * TRANSFERRED. *
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 20 * CORPORATION. *
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 24 *
0000 25 *
0000 26 *****
0000 27
0000 28 **
0000 29 : FACILITY: EXECUTIVE, BALANCE SET SWAPPER
0000 30
0000 31 : ABSTRACT:
0000 32 : OSWPSCHED SELECTS THE LOWEST PRIORITY PROCESS SUITABLE AS AN
0000 33 : OUTSWAP CANDIDATE TO MAKE MEMORY PAGES AVAILABLE BOTH FOR
0000 34 : INSWAPPING AND MAINTENANCE OF THE DESIRED NUMBER OF FREE PAGES.
0000 35
0000 36 : ENVIRONMENT:
0000 37 : MODE = KERNEL, RESIDENT
0000 38
0000 39 : AUTHOR: R. I. HUSTVEDT, CREATION DATE: 14-FEB-77
0000 40
0000 41 : MODIFIED BY:
0000 42
0000 43 : V03-010 SSA0028 Stan Amway 2-Aug-1984
0000 44 : Reverse ordering of LEF and HIB entries in OSWPSTATE.
0000 45
0000 46 : V03-009 WMC0009 Wayne Cardoza 18-Jun-1984
0000 47 : Write modified page list if limit has been set to zero.
0000 48
0000 49 : V03-008 WMC0008 Wayne Cardoza 02-Mar-1984
0000 50 : Check the PHDSM_NO_WS_CHNG flag.
0000 51
0000 52 : V03-007 SSA0014 Stan Amway 5-Mar-1984
0000 53 : Inhibit 2nd level trimming of realtime processes.
0000 54
0000 55 : V03-006 SSA0003 Stan Amway 5-Dec-1983
0000 56 : Reclaim memory loans before doing any 2nd level trimming or
0000 57 : swapping.

```

0000 58 :
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
0000 64 :
0000 65 :
0000 66 :
0000 67 :
0000 68 :
0000 69 :
0000 70 :
0000 71 :
0000 72 :
0000 73 :
0000 74 :
0000 75 :
0000 76 :
0000 77 :
0000 78 :
0000 79 :
0000 80 :
0000 81 :
0000 82 :
0000 83 :
0000 84 :
0000 85 :
0000 86 :
0000 87 :
0000 88 :
0000 89 :
0000 90 :
0000 91 :
0000 92 :
0000 93 :
0000 94 :
0000 95 :
0000 96 :
0000 97 :
0000 98 :
0000 99 :
0000 100 :
0000 101 :
0000 102 :
0000 103 :
0000 104 :
0000 105 :
0000 106 :
0000 107 :
0000 108 :
0000 109 :
0000 110 :
0000 111 :
0000 112 :
0000 113 :
0000 114 :

Added new category of outswap candidates, DORMANT. These processes are computable, but are not making any progress (as measured by the time of the last significant event for the process and compared against a threshold determined by the new SYSGEN parameter DORMANTWAIT). After SUSP processes, DORMANT processes are the most likely outswap candidates. This change solves the problem that existed when low-priority, COM processes would hold onto large amounts of memory because they were never trimmed or swapped.

Restored use of SWAPASAP flag during 1st level trimming. Added SWAPASAP flag to SUSP and DORMANT state table entries. Deleted same from all other entries.

Restored finer granularity in OSWPSTATE table for outswap candidate selection classes.

Changed use of LONGWAIT to track change in units from 6.6 ms to 1 second, EXESGL_ABSTIM as the time reference, and movement of the WAITIME field from a word in the PHD to a longword in the PCB.

V03-005 TCM0001 Trudy C. Matthews 31-Mar-1983
Change references to working set fields in PHD so that they are used as unsigned words.

V03-004 HRJ0210 Herb Jacobs 28-Feb-1982
Change priorities to shrink compute processes to quota before outswapping short waiting processes.

V03-003 HRJ0100 Herb Jacobs 29-Jun-1982
Honor DISAWS in shrinking, and add perturbation into scheduler to recover low priority compute bound processes memory.

V03-002 HRJ0061 Herb Jacobs 25-Mar-1982
Rewrite again to make it table driven to allow performance measurement of several different scheduling alternatives. Remove V02-008 code, this is accomplished as a side effect of an unsatisfiable pages needed count.

V03-001 HRJ0060 Herb Jacobs 20-Mar-1982
Add interlock to prevent this code from changing WSLAST out of process context.

V02-012 HRJ0052 Herb Jacobs 08-Jan-1982
Add sysgen parameter threshold of whether using modified page writer should be used for gaining memory.

V02-011 HRJ0051 Herb Jacobs 30-Jan-1982
Don't allow a swap candidate if the process is larger than its swap slot. This can happen if free working set list returns a resource wait condition as the result of the preparatory shrink. This should be short lived, and will fix itself on a future need.
Also detect a queue reordering as the result of ALLOCPFN

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 :--

reporting free pages; in this case restart rather than
continue.

V02-010 HRJ0049 Herb Jacobs 25-Jan-1982
Recompute of extra dynamic working set after MMG\$SHRINKWS.

V02-009 HRJ0044 Herb Jacobs 12-Jan-1982
Update for changed interface to MMG\$SHRINKWS and add SWPFAIL
count into priority consideration for outswaps.

V02-008 HRJ0024 Herb Jacobs 29-Jul-1981
Add back ability to force outswap for shortage of balance set
slots.

V02-007 HRJ0020 Herb Jacobs 20-Apr-1981
Major enhancement to attempt to shrink processes working sets
to acquire memory before resulting to swapping.

06 RIH0059 RICHARD I. HUSTVEDT 25-FEB-1980 19:45
Remove spurious modification of SWTIME.

05 RIH0057 RICHARD I. HUSTVEDT 04-FEB-1980 10:07
Remove use of SWPRATE. Moved to SWAPPER instead.

04 RIH0036 RICHARD I. HUSTVEDT 01-NOV-1979 11:24
Change wait time measurement to use SCH\$GW_IOTA, a fixed allowance
for voluntary waits to avoid undesired negative performance
feedback.

03 RIH0031 RICHARD I. HUSTVEDT 08-AUG-1979 13:17
Remove inhibition on swapping for processes holding a Mutex.

```

0000 149      .SBTTL  DECLARATIONS
0000 150      :
0000 151      : INCLUDE FILES:
0000 152      :
0000 153      :
0000 154      $CEBDEF      ; DEFINE CEB OFFSETS
0000 155      $IPLDEF     ; DEFINE INTERRUPT PRIORITY LEVELS
0000 156      $PCBDEF     ; DEFINE PCB OFFSETS
0000 157      $PHDDEF     ; DEFINE PHD OFFSETS
0000 158      $PRDEF      ; DEFINE PROCESSOR REGISTERS
0000 159      $STATEDEF   ; DEFINE STATE VALUES
0000 160      $WQHDEF     ; DEFINE WAIT QUEUE HEADER OFFSETS
0000 161      :
0000 162      :
0000 163      : MACROS:
0000 164      :
0000 165      :
0000 166      :
0000 167      : OUTSWP - MACRO TO GENERATE ORDERED ACTION TABLE FOR
0000 168      : NON-EXECUTABLE STATES.
0000 169      :
0000 170      : OUTSWP STATE,[<FLAG,FLAG,FLAG...>]
0000 171      :
0000 172      : WHERE: STATE = STATE NAME LESS SCH$C PREFIX
0000 173      : FLAG   = ONE OF THE SWAP SCHEDULING FLAGS
0000 174      :         _V_CEF = COMMON EVENT FLAG STATE
0000 175      :         _V_DIOCNT = REQUIRE NONZERO DIOCNT
0000 176      :         _V_LONGWAIT = ONLY CONSIDER PROCESSES IN A LONG WAIT
0000 177      :         _V_SHORTWAIT = ONLY CONSIDER PROCESSES IN A SHORT WAIT
0000 178      :         _V_INQUAN = OBSERVE INITIAL QUANTUM, IGNORE
0000 179      :         PROCESS IF PCB$V_INQUAN IS SET
0000 180      :         _V_NDIOCNT = IGNORE PROCESS IF PCB$W_DIOCNT IS NONZERO
0000 181      :         _V_PRIORITY = OBSERVE PRIORITY OF POSSIBLE OUTSWAP
0000 182      :         CANDIDATE RELATIVE TO INSWAP CANDIDATE
0000 183      :         _V_COMPUTE = USED TO FLAG START OF COM PROCESS SCAN
0000 184      :         _V_SWPOGOAL = REDUCE PROCESS PAST QUOTA BEFORE SWAPPING
0000 185      :         _V_SWAPASAP = WHEN A PROCESS IN THIS STATE REACHES A
0000 186      :         SWAPPABLE SIZE, EXIT BY SWAPPING IT
0000 187      :         _V_DORMANT = ONLY CONSIDER PROCESS IF IT IS DORMANT
0000 188      :
0000 189      : .MACRO OUTSWP,STATE,FLAGS
0000 190      : .BYTE SCH$C,'STATE
0000 191      : TMP...=0
0000 192      : .IRP FLGS,<FLAGS>
0000 193      : TMP...=TMP...+<1@_V_'FLGS>
0000 194      : .ENDR
0000 195      : .WORD TMP...
0000 196      : .ENDM OUTSWP
0000 197      :
0000 198      : EQUATED SYMBOLS:
0000 199      :
0000 200      :
0000 201      : _YIELD 0,<- ; DEFINE CONTROL BITS
0000 202      : CEF,- ; COMMON EVENT FLAG WAIT -MUST BE BIT 0
0000 203      : DIOCNT,- ; REQUIRE NON-ZERO DIOCNT
0000 204      : LONGWAIT,- ; PROCESS WAIT MUST BE LONG
0000 205      : SHORTWAIT,- ; PROCESS WAIT MUST BE SHORT

```

```

0000 206      INQUAN,-           ; OBSERVE INQUAN FLAG
0000 207      NDIOCNT,-        ; DIOCNT MUST BE ZERO
0000 208      PRIORITY,-      ; OBSERVE PRIORITY
0000 209      COMPUTE,-       ; START OF COMPUTE PROCESSES
0000 210      SWPOGOAL,-      ; DESIRED SIZE IS SWPOUTPGCNT
0000 211      SWAPASAP,-      ; SWAP PROCESSES IN THIS STATE FIFO
0000 212      DORMANT,-       ; PROCESS MUST BE DORMANT
0000 213      >
0000 214      ;
0000 215      ; OWN STORAGE:
0000 216      ;
0000 217      ;
00000000 218      .PSECT $$$220, LONG
00000000 219      SWP$GL_SWTIME:: ; EARLIEST TIME FOR NEXT EXCHANGE SWAP
00000000 220      .LONG 0
00000000 221      SAVED_R3:
00000000 222      .LONG        ; Used by COMPUTE scan to save state
00000000 223      ;
00000000 224      .PSECT $OSWPSCHED, BYTE
0000 225      OSWPSTATE:
0000 226      OUTSWP  SUSP, <SWAPASAP> ; SUSPENDED
00 0003 227      .BYTE 0 ; ATTEMPT OUTSWAPS BEFORE PROCEEDING
0004 228      OUTSWP  COM, <DORMANT, COMPUTE, SWAPASAP> ; DORMANT PROCESSES
00 0007 229      .BYTE 0 ; ATTEMPT OUTSWAPS BEFORE PROCEEDING
0008 230      OUTSWP  HIB, <LONGWAIT, SWPOGOAL> ; HIBERNATING, LONG WAIT
0008 231      OUTSWP  LEF, <NDIOCNT, LONGWAIT, SWPOGOAL> ; LOCAL EVENT FLAG WAIT, LONG WAIT
00 000E 232      .BYTE 0 ; ATTEMPT OUTSWAPS BEFORE PROCEEDING
000F 233      OUTSWP  CEF, <CEF, NDIOCNT, SWPOGOAL> ; COMMON EVENT FLAG WAIT
00 0012 234      .BYTE 0 ; ATTEMPT OUTSWAPS BEFORE PROCEEDING
0013 235      OUTSWP  HIB, <SHORTWAIT, SWPOGOAL> ; HIBERNATING
0016 236      OUTSWP  LEF, <NDIOCNT, SHORTWAIT, SWPOGOAL> ; LOCAL EVENT FLAG WAIT
00 0019 237      .BYTE 0 ; ATTEMPT OUTSWAPS BEFORE PROCEEDING
001A 238      OUTSWP  FPG, <PRIORITY> ; FREE PAGE WAIT
001D 239      OUTSWP  COLPG, <PRIORITY> ; COLLIDED PAGE WAIT
00 0020 240      .BYTE 0 ; ATTEMPT OUTSWAPS BEFORE PROCEEDING
0021 241      OUTSWP  MWAIT ; MUTEX WAIT
00 0024 242      .BYTE 0 ; ATTEMPT OUTSWAPS BEFORE PROCEEDING
0025 243      OUTSWP  CEF, <CEF, DIOCNT, INQUAN, PRIORITY> ; COMMON EVENT WAIT -DIOCNT NONZERO
0028 244      OUTSWP  LEF, <DIOCNT, INQUAN, PRIORITY> ; LOCAL EVENT WAIT -DIOCNT NONZERO
00 002B 245      .BYTE 0 ; ATTEMPT OUTSWAPS BEFORE PROCEEDING
002C 246      OUTSWP  PFW, <INQUAN, PRIORITY> ; PAGE FAULT WAIT
002F 247      OUTSWP  COM, <INQUAN, COMPUTE> ; COMPUTE PROCESSES
FF 0032 248      .BYTE -1 ; END OF TABLE

```



```

0033 250 .SBTTL SCH$OSWPSCHED - OUT SWAP SCHEDULER
0033 251 :++
0033 252 : FUNCTIONAL DESCRIPTION:
0033 253 : SCH$OSWPSCHED SCANS IN LOWEST PRIORITY ORDER FOR PROCESSES THAT
0033 254 : CAN BE USED TO ACQUIRE MEMORY FROM TO SATISFY THE DEFICIT.
0033 255 : IF AFTER SHRINKING PROCESSES FAIL, A PROCESS WILL BE SELECTED
0033 256 : FOR OUTSWAPPING.
0033 257 :
0033 258 : CALLING SEQUENCE:
0033 259 : BSB/JSB SCH$OSWPSCHED
0033 260 : IPL = IPL$_SYNCH
0033 261 :
0033 262 : INPUT PARAMETERS:
0033 263 : SWP$GB ISWPRI - PRIORITY OF INSWAP PROCESS
0033 264 : FP - NEGATIVE NUMBER OF DESIRED PAGES
0033 265 :
0033 266 : IMPLICIT INPUTS:
0033 267 : ALL STATE QUEUE HEADERS AND ATTACHED PROCESS CONTROL BLOCKS.
0033 268 :
0033 269 : OUTPUT PARAMETERS:
0033 270 : R4 - PCB ADDRESS OF OUTSWAP CANDIDATE
0033 271 : 0=> NONE
0033 272 : R5 - PHD ADDRESS OF OUTSWAP CANDIDATE
0033 273 : R0,R1,R2,R3,R6,R7,R8,R9 DESTROYED
0033 274 :
0033 275 : IMPLICIT OUTPUTS:
0033 276 : SOME PROCESSES WORKING SETS ARE LIKELY TO BE SHRUNK.
0033 277 : PCB$V RES OF OUTSWAP CANDIDATE IS CLEARED AND A SWPOUT EVENT
0033 278 : HAS BEEN REPORTED IF AN OUTSWAP IS TO OCCUR.
0033 279 :
0033 280 : COMPLETION CODES:
0033 281 : NONE
0033 282 :
0033 283 : SIDE EFFECTS:
0033 284 : NONE
0033 285 :
0033 286 :--
0033 287 : .ENABLE LSB
0033 288 :
5E 04 C0 0033 289 80$: ADDL #4,SP ; POP CONTINUE ADDRESS OFF STACK
54 D4 0036 290 90$: CLRL R4 ; GET PROCESS TO SWAP, IF ANY
05 0038 291 RSB ; RETURN TO SWAPPER PROPER
0039 292 :
0039 293 RETRY: ; USED IN CASE ALLOCPFN REORDERS QUEUES
5E 04 C0 0039 294 ADDL #4,SP ; POP CONTINUE ADDRESS FROM STACK AND
003C 295 ; TRY AGAIN
003C 296 SCH$OSWPSCHED:: ; SCHEDULE OUTSWAP
0232 30 003C 297 BSBW MMG$MPWCHECK ; WILL STARTING MPW SATISFY NEED?
5D D5 003F 298 TSTL FP ; FP CONTAINS DEFICIT
F3 18 0041 299 BGEQ 90$ ; BRANCH IF NO MORE DEFICIT
0000'CF 94 0043 300 CLRB W^SCH$GB_RESCAN ; CLEAR RESCAN NEEDED FLAG
0047 301 :
0047 302 : Begin 1st level trimming
0047 303 :
53 B6 AF 9E 0047 304 MOVAB OSWPSTATE,R3 ; INITIALIZE STATE SCAN AT TOP
52 53 D0 004B 305 10$: MOVL R3,R2 ; RESET TO NEW SUB SECTION
28 13 004E 306 BEQL 24$ ; BRANCH IF AT END OF TABLE

```

```

00F6 30 0050 307 BSBW CANDIDATE ; GET A CANDIDATE TO BE SHRUNK
F6 13 0053 308 BEQL 10$ ; BRANCH IF NONE, TRY NEXT SUB SECTION
58 18 A5 08 A5 A3 0055 309 20$ SUBW3 PHD$W_WSLIST(R5),PHD$W_WSQUOTA(R5),R8
58 B6 005B 310 INCW R8 ; SIZE TO SHRINK WORKING SET TO
01D9 30 005D 311 BSBW SHRINKWS ; RECLAIM MEMORY FROM IT
D1 0000'CF 00' D1 18 0060 312 BGEQ 80$ ; IF NO DEFICIT, QUIT TRIMMING
06 56 09 E0 0062 313 BBS S^#SCH$V_REORD,W^SCH$GB_RESCAN,RETRY ; BRANCH IF FPG STATE CHANGE
006C 30 006C 314 BBC # V SWAPASAP,R6,22$ ; IF A FULL PROCESS SWAP IS DESIRED
53 59 E8 006F 315 BSBW 100$ ; AND IT IS CURRENTLY POSSIBLE,
9E 16 0072 316 BLBS R9,60$ ; DO IT NOW
DF 12 0074 317 22$ JSB @(SP)+ ; TRY FOR THE NEXT ONE
D3 11 0076 318 BNEQ 20$ ; BRANCH IF THERE IS ONE
0078 319 BRB 10$
0078 320 ;
0078 321 ;
0078 322 ;
53 85 AF 9E 0078 323 24$ MOVAB OSWPSTATE,R3 ; RESTART STATE SCAN AT TOP
52 53 D0 007C 324 25$ MOVL R3,R2 ; RESET TO NEW SUB SECTION
B5 13 007F 325 BEQL 90$ ; BRANCH IF AT END OF TABLE
00C5 30 0081 326 BSBW CANDIDATE ; GET A CANDIDATE TO BE SHRUNK
F6 13 0084 327 BEQL 25$ ; BRANCH IF NONE, TRY NEXT SUB SECTION
0161 30 0086 328 30$ BSBW SWAPQUANCHK ; CHECK SWAPABILITY AND INITIAL QUANTUM
27 59 E9 0089 329 BLBC R9,40$ ; BRANCH IF SHOULD NOT SWAP IT NOW
OF 24 A4 18 E0 008C 330 BBS #PCB$V_DISAWS,PCB$L_STS(R4),32$ ; BRANCH IF CAN'T SHRINK IT
OB A4 10 91 0091 331 CMPB #16,PCB$B_PRI(R4) ; Real-time process ?
09 1A 0095 332 BGTRU 32$ ; BR if yes - can't shrink RT process
58 0000'CF D0 0097 333 MOVL W^SWP$GL_SWPPGCNT,R8 ; VALUE TO SHRINK CANDIDATE TO
08 56 08 E0 009C 334 BBS # V SWPOGOAL,R6,35$ ; BRANCH IF THIS IS DESIRED GOAL SIZE
58 18 A5 08 A5 A3 00A0 335 32$ SUBW3 PHD$W_WSLIST(R5),PHD$W_WSQUOTA(R5),R8
58 B6 00A6 336 INCW R8 ; SIZE TO SHRINK WORKING SET TO
018E 30 00A8 337 35$ BSBW SHRINKWS ; RECLAIM MEMORY FROM IT
86 0000'CF 00' E0 00AD 338 BGEQ 80$ ; RETURN WITH NO ONE TO OUTSWAP
9E 16 00B3 339 BBS S^#SCH$V_REORD,W^SCH$GB_RESCAN,RETRY ; BRANCH IF FPG STATE CHANGE
CF 12 00B5 340 40$ JSB @(SP)+ ; TRY FOR NEXT ONE
00BF 30 00B7 341 BNEQ 30$ ; BRANCH IF THERE IS ONE
C0 13 00BA 342 BSBW CANDIDATE ; GET A CANDIDATE TO BE OUTSWAPPED
1D 10 00BC 343 50$ BEQL 25$ ; NO ONE TO OUTSWAP
04 59 E8 00BE 344 BSBB 100$ ; CHECK SWAPABILITY AND INITIAL QUANTUM
9E 16 00C1 345 BLBS R9,60$ ; BRANCH TO SWAP IT
F5 11 00C3 346 JSB @(SP)+ ; TRY FOR NEXT ONE
00C5 347 BRB 50$ ; IS THERE ANOTHER ONE?
00C5 348
00 24 A4 00 D4 00C5 349 60$ CLRL R2 ; NULL PRIORITY INCREMENT
E7 00C7 350 BBCCI #PCB$V_RES,PCB$L_STS(R4),70$ ; CLEAR RESIDENT FLAG
0000'CF 0000'CF B0 00C0 351 70$ RPTEVT SWPOUT ; REPORT SWAPOUT EVENT
SE 04 C0 00D7 352 MOVW W^SCH$GW_SWPFAIL,W^SCH$GW_SWPFCNT ; RESET FAILURE COUNTER
05 00DA 353 ADDL #4,SP ; POP CONTINUE ADDRESS OFF STACK
00DB 354 RSB ; WITH PCB IN R4, PHD IN R5
59 34 A4 36 A4 A1 00DB 355 ADDW3 PCB$W_PPGCNT(R4),PCB$W_GPGCNT(R4),R9 ; GET SIZE OF PROCESS
52 A5 59 B1 00E1 356 100$ CMPW R9,PHD$W_SWAPSIZE(R5) ; DO WE HAVE ENOUGH SPACE TO SWAP IT?
03 1B 00E5 357 BLEQU 110$ ; BRANCH IF YES
59 D4 00E7 358 CLRL R9 ; SET FAILURE
05 00E9 359 RSB ; RETURN
00FD 30 00EA 360 BSBW SWAPQUANCHK ; CHECK SWAPABILITY AND INITIAL QUANTUM
05 00ED 361 110$ RSB
00EE 362
00EE 363

```

Begin 2nd level trimming and swapping

```

00EE 364 .DISABLE LSB
00EE 365 .ENABLE LSB
00EE 366 COMPUTE:
06 56 0A E1 00EE 367 BBC # V DORMANT,R6,5$ ; COMPUTE STATE SCAN
0000'CF B5 00F2 368 TSTW W*SCH$GW DORMANTWAIT ; Prevent DORMANT process scan
54 13 00F6 369 BEQL NEXTSTATE ; if DORMANTWAIT = 0
0004'CF 53 D0 00F8 370 5$: MOVL R3,W*SAVED R3 ; Save state for scan continuation
51 0000'CF 9E 00FD 371 MOVAB W*SCH$AQ COMOH,R1 ; POINT TO LONGWORD BEYOND PRI=0 HDR
53 71 7D 0102 372 10$: MOVQ -(R1),R3 ; GET QUEUE HEADER
2E 13 0105 373 BEQL 50$ ; NONE, SCAN FINISHED
54 51 D1 0107 374 20$: CMPL R1,R4 ; CHECK FOR EMPTY QUEUE
F6 13 010A 375 BEQL 10$ ; YES, NEXT QUEUE (HIGHER PRIORITY)
07 56 0A E1 010C 376 BBC # V DORMANT,R6,30$ ; Must process be DORMANT ?
00F7 30 0110 377 BSBW DORMANT_CHECK ; Process must be DORMANT - Is it ?
2A 19 0113 378 BLSS 55$ ; If LSS, terminate scan
18 13 0115 379 BEQL 40$ ; If EQL, bad candidate, continue scan
0117 380 ; (If GTR, good candidate)
0000'CF 0B A4 91 0117 381 30$: CMPB PCB$B_PRI(R4),W*SWP$GB_1SWPRI ; COMPARE WITH INSWAP CANDIDATE
16 19 011D 382 BLSS 50$ ; BRANCH IF CAN'T SWAP
55 6C A4 D0 011F 383 MOVL PCB$L_PHD(R4),R5 ; MMG ROUTINES WANT PHD IN R5
0123 384 ASSUME PHD$M_NO_WS_CHNG LE 255
36 A5 40 8F 93 0123 385 BITB #PHD$M_NO_WS_CHNG,PHD$W_FLAGS(R5) ; ARE WE ALLOWED TO TOUCH IT
05 12 0128 386 BNEQ 40$ ; NO
55 55 D0 012A 387 MOVL R5,R5 ; SET Z=0
9E 16 012D 388 JSB @($P)+ ; RETURN WITH CANDIDATE, WE MAY CONTINUE
54 04 A4 D0 012F 389 40$: MOVL PCB$L_SQBL(R4),R4 ; BLINK THROUGH QUEUE
D2 11 0133 390 BRB 20$ ; AND CHECK FOR EMPTY
0135 391
0000'CF B7 0135 392 50$: DECW W*SCH$GW_SWPFCNT ; COUNT A FAILURE
04 18 0139 393 BGEQ 55$ ; EXIT IF NO UNDERFLOW
0000'CF B4 013B 394 CLRW W*SCH$GW_SWPFCNT ; LIMIT VALUE TO ZERO
53 0004'CF D0 013F 395 55$: MOVL W*SAVED R3,R3 ; Restore state and continue
06 11 0144 396 BRB NEXTSTATE
0146 397
53 D4 0146 398 60$: CLRL R3 ; INDICATE NONE FOUND, Z BIT SET, END
05 0148 399 70$: RSB ; AND RETURN
0149 400
53 52 D0 0149 401 CANDIDATE:
0149 402 MOVL R2,R3 ; INIT STATE SCAN AT CURRENT SUB SECTION
014C 403 NEXTSTATE: ; TRY NEXT STATE
014C 404 ; (MOVZBL not used so that the two
51 51 D4 014C 405 CLRL R1 ; conditional tests will work correctly)
83 90 014E 406 MOVB (R3)+,R1 ; GET NEXT STATE TO TRY
F5 13 0151 407 BEQL 70$ ; BRANCH IF AT END OF STATE SUB SECTION
F1 19 0153 408 BLSS 60$ ; Branch if at end of table
56 83 3C 0155 409 MOVZWL (R3)+,R6 ; FETCH STATE FLAGS WORD
92 56 07 E0 0158 410 BBS # V COMPUTE,R6,COMPUTE ; TRY COM IF CANT FIND NON-EXECUTABLE
1A 56 00 E1 015C 411 BBC #-V-CEF,R6,110$ ; CONTINUE IF NOT CEF WAIT
57 0000'CF 7E 0160 412 MOVAQ W*SCH$GQ_CEBHD,R7 ; GET ADDRESS OF LISTHEAD
50 14 A7 DE 0165 413 MOVAL CEB$L_WQFL(R7),R0 ; AND MAKE WORKING COPY
0169 414 NEXTCEB: ; NEXT CEB
E0 56 E9 0169 415 BLBC R6,NEXTSTATE ; BR IF V CEF CLEAR
50 EC A0 D0 016C 416 MOVL -CEB$L_WQFL(R0),R0 ; FLINK FORWARD
57 50 D1 0170 417 CMPL R0,R7 ; CHECK FOR END OF CEB LIST
D7 13 0173 418 BEQL NEXTSTATE ; AT END IF EQL
50 14 C0 0175 419 ADDL #CEB$L_WQFL,R0 ; POINT TO WAIT QUEUE
OA 11 0178 420 BRB 120$ ; ELSE SCAN CEB WAIT QUEUE

```

```

50 0000'CF41 7E 017A 421 110$: MOVAQ W^SCH$AQ_WQHDR[R1],R0 ; COMPUTE ADDRESS OF WAIT QUEUE HDR
50 6041 DE 0180 422 MOVAL (R0)[R1],R0 ; AS SCH$AQ_WQHDR+12*R1
54 04 A0 D0 0184 423 120$: MOVL WQH$L_WQBL(R0),R4 ; GET TAIL OF QUEUE
0188 424
0188 425 .DISABLE LSB
0188 426
50 54 D1 0188 427 WSCAN: CMPL R4,R0 ; CHECK FOR END OF QUEUE
DC 13 0188 428 BEQL NEXTCEB ; TRY NEXT STATE OR CEB
53 24 A4 E9 018D 429 BLBC PCB$S_STS(R4),NEXTPR ; NOT RESIDENT, NEXT IN QUEUE
40 A4 3E A4 B1 0191 430 CMPW PCB$W_DIOCNT(R4),PCB$W_DIOCM(R4) ; CHECK IT FOR ZERO
06 13 0196 431 BEQL 10$ ; IF ZERO, IS THAT PERMITTED?
48 56 05 E0 0198 432 BBS #V_NDIOCNT,R6,NEXTPR ; BR IF NON-ZERO NOT PERMITTED
04 11 019C 433 BRB 20$ ; CONTINUE IF NON-ZERO ALLOWED
42 56 01 E0 019E 434 10$: BBS #V_DIOCNT,R6,NEXTPR ; BR IF ZERO DIOCNT NOT PERMITTED
OE 56 06 E1 01A2 435 20$: BBC #V_PRIORITY,R6,30$ ; BR IF PRIORITY CAN BE IGNORED
0000'CF 0B A4 91 01A6 436 CMPB PCB$B_PRI(R4),W^SWP$GB_ISWPRI ; COMPARE WITH INSWAP PRIORITY
06 18 01AC 437 BGEQ 30$ ; BRANCH IF LESS IMPORTANT THAN ISWP
0000'CF B5 01AE 438 TSTW W^SCH$GW_SWPFCNT ; CHECK FOR FAILURE COUNT EXHAUSTED
30 14 01B2 439 BGTR NEXTPR ; BRANCH IF NOT GOOD CHOICE YET
56 0C 93 01B4 440 30$: BITB #<1@_V_SHORTWAIT>!<1@_V_LONGWAIT>,R6 ; CHECK WAIT TIME?
18 13 01B7 441 BEQL 50$ ; BRANCH IF NOT
55 0000'CF 3C 01B9 442 MOVZWL W^SCH$GW_LONGWAIT,R5 ; Get long wait threshold
55 0118 C4 C0 01BE 443 ADDL2 PCB$S_WAITIME(R4),R5 ; R5 <= wait time + delta
55 0000'CF D1 01C3 444 CMPL W^EXE$GL_ABSTIM,R5 ; HOW LONG HAVE WE BEEN WAITING?
02 56 03 E1 01C8 445 BBC #V_SHORTWAIT,R6,40$ ; BRANCH IF NON- SHORT WAIT ALLOWED
02 56 02 E1 01CE 447 40$: BBC #V_LONGWAIT,R6,50$ ; BRANCH IF NON- LONG WAIT ALLOWED
10 1F 01D2 448 BLSSU NEXTPR ; BRANCH IF WAIT INTERVAL IS SHORT
55 6C A4 D0 01D4 449 50$: MOVL PCB$S_PHD(R4),R5 ; MMG ROUTINES WANT PHD IN R5, SET Z=0
01D8 450 ASSUME PHD$M_NO_WS_CHNG LE 255
36 A5 40 8F 93 01D8 451 BITB #PHD$M_NO_WS_CHNG,PHD$W_FLAGS(R5) ; ARE WE ALLOWED TO TOUCH IT
05 12 01DD 452 BNEQ NEXTPR ; NO
55 55 D0 01DF 453 MOVL R5,R5 ; SET Z=0
9E 16 01E2 454 JSB @($P)+ ; RETURN SO WE CAN CONTINUE
54 04 A4 D0 01E4 455 NEXTPR: MOVL PCB$S_SQBL(R4),R4 ; BLINK THROUGH QUEUE
9E 11 01E8 456 BRB WSCAN ; AND CONTINUE SCAN
01EA 457
01EA 458 SWAPQUANCHK: ; CHECK VALIDITY OF PROCESS CHOICE
18 24 A4 59 D4 01EA 459 CLRL R9 ; ASSUME BAD CHOICE
12 56 04 E0 01EC 460 BBS #PCB$V_PSWAPM,PCB$S_STS(R4),20$ ; BRANCH IF NOT SWAPPABLE
OD 24 A4 03 E1 01F1 461 BBC #V_INQUAN,R6,10$ ; BRANCH IF INQUAN CAN BE IGNORED
OF 0000'CF 91 01F5 462 BBC #PCB$V_INQUAN,PCB$S_STS(R4),10$ ; BRANCH IF INQUAN SATISFIED
06 15 01FF 463 CMPB W^SWP$GB_ISWPRI,#15$ ; IS THIS A REAL TIME INSWAP?
0000'CF B5 0201 464 BLEQ 10$ ; BRANCH IF SO
02 14 0205 465 TSTW W^SCH$GW_SWPFCNT ; CHECK FOR FAILURE COUNT EXHAUSTED
59 D6 0207 466 20$: BGTR 20$ ; BRANCH IF NOT GOOD CHOICE YET
05 0209 467 10$: INCL R9 ; SET GOOD CHOICE
020A 468 20$: RSB ; RETURN WITH STATUS IN R9
020A 469
020A 470 DORMANT_CHECK: ; Check if candidate is DORMANT
55 7E 01 CE 020A 471 MNEGL #1,-($P) ; Preset terminate scan status
55 0B A4 90 020D 472 MOVB PCB$B_PRI(R4),R5 ; Fetch priority of candidate
55 10 91 0211 473 CMPB #16,R5 ; Real-time process ?
20 1A 0214 474 BGTRU 10$ ; BR if yes
58 1F 0000'CF 83 0216 475 SUBB3 W^SYS$GB_DEFPRI,#31,R8 ; Convert to internal form for test
58 55 91 021C 476 CMPB R5,R8 ; Is this a low-priority job ?
15 1F 021F 477 BLSSU 10$ ; BR if not

```

58	0000	6E	D6	0221	478	INCL	(SP)	:	Set bad candidate, continue status
58	0118	CF	3C	0223	479	MOVZWL	W^SCH\$GW_DORMANTWAIT,R8	:	Get dormant wait threshold
58	0000	C4	C0	0228	480	ADDL2	PCB\$L_WAITTIME(R4),R8	:	Has process had a significant
		CF	D1	0220	481	CPL	W^EXE\$GL_ABSTIM,R8	:	event recently ?
		02	1F	0232	482	BLSSU	10\$:	BR if not
		6E	D6	0234	483	INCL	(SP)	:	Indicate that process is DORMANT
		8E	D5	0236	484	TSTL	(SP)+	:	Set status in condition codes
			05	0238	485	RSB		:	

```

0239 487 .SBTTL SHRINK WORKING SET
0239 488 :++
0239 489 : FUNCTIONAL DESCRIPTION:
0239 490 : SHRINK IS USED TO MAKE PAGES AVAILABLE FROM OTHER PROCESSES THAT
0239 491 : HAVE BORROWED PAGES TO SATISFY HIGH FAULTING RATES OR TO SHRINK
0239 492 : A SWAPPABLE PROCESS TO THE DESIRED SIZE FOR SMALLER SWAPS.
0239 493 :
0239 494 : CALLING SEQUENCE:
0239 495 : BSB/JSB SHRINKWS
0239 496 :
0239 497 : INPUT PARAMETERS:
0239 498 : R4 - PCB OF TARGET PROCESS
0239 499 : R5 - PHD OF TARGET PROCESS
0239 500 : R8 - SIZE TO ATTEMPT TO SHRINK PROCESS TO
0239 501 : FP - NEGATIVE NUMBER OF TOTAL PAGES WANTED BY SYSTEM
0239 502 :
0239 503 : IMPLICIT INPUTS:
0239 504 : WORKING SET LIST
0239 505 :
0239 506 : OUTPUT PARAMETERS:
0239 507 : FP UPDATED
0239 508 : R9 DESTROYED
0239 509 :
0239 510 : IMPLICIT OUTPUTS:
0239 511 : TARGET PROCESS WORKING SET SHRUNK UP TO R8 INPUT VALUE
0239 512 :
0239 513 : COMPLETION CODES:
0239 514 : POSITIVE INDICATES SWAPPER REQUEST SATISFIED
0239 515 :
0239 516 : SIDE EFFECTS:
0239 517 : THIS ROUTINE COULD POTENTIALLY DUMP A LOT OF TRAFFIC INTO THE
0239 518 : PAGING SYSTEM, ESPECIALLY THE MODIFIED PAGE WRITER.
0239 519 :
0239 520 :--
0239 521 :
0239 522 :

```

```

0239 522 SHRINKWS:
59 58 58 3C 0239 523 MOVZWL R8,R8 ; GET INPUT SIZE IN A LONGWORD
59 59 50 A5 3C 023C 524 MOVZWL PHD$W WSSIZE(R5),R9 ; GET CURRENT SIZE
59 58 59 C3 0240 525 SUBL3 R9,R8,R9 ; CAN WE GET ANY PAGES FROM PROCESS?
28 18 0244 526 BGEQ 30$ ; BRANCH IF NOT
51 59 0F BB 0246 527 PUSHR #*M<R0,R1,R2,R3> ; SAVE SWAPPER OUTSWAP SCHEDULER REGS
0000'CF DD 0248 528 MOVL R9,R1 ; NUMBER OF PAGES TO SHRINK PROCESS BY
0000'CF 00' 88 024B 529 PUSHL W^SCH$GL_FREECNT ; SAVE OLD FREE LIST COUNT
0000'CF 00' 88 024F 530 BISB S^*MMG$M_NOLASTUPD!MMG$M_NOWAIT,W^MMG$GB_FREWFLGS ; DISABLE
0000'CF FDA9' 30 0254 531 BSBW MMG$SHRINKWS ; WSLAST UPDATE AND MP LIST SIZE WAIT
0000'CF 00' 8A 0257 533 BICB S^*MMG$M_NOLASTUPD!MMG$M_NOWAIT,W^MMG$GB_FREWFLGS ; REENABLE
FDA1' 30 025C 534 ; WSLAST UPDATE AND MP LIST SIZE WAIT
59 0000'CF 8E C3 025C 535 BSBW MMG$EXTRADYNWS ; RECALC EXTRA DYNAMIC WORKING SET
5D 59 C0 025F 536 SUBL3 (SP)+,W^SCH$GL_FREECNT,R9 ; CALC NUMBER OF PAGES FREED
02 18 0265 537 ADDL R9,FP ; HAVE WE SATISFIED DEFICIT?
05 10 0268 538 BGEQ 20$ ; BRANCH IF YES
0F BA 026A 539 BSBB MMG$MPWCHECK ; WILL STARTING MPW SATISFY NEED?
5D 05 026C 540 POPR #*M<R0,R1,R2,R3> ; RESTORE OSWPSCHED REGISTERS
05 05 026E 541 TSTL FP ; SET DEFICIT INDICATION
05 0270 542 RSB ; RETURN

```

```

0271 544 .SBTTL START MODIFIED PAGE WRITER TEST
0271 545 :++
0271 546 : FUNCTIONAL DESCRIPTION:
0271 547 : THIS ROUTINE IS CALLED BY THE SWAPPER TO OPTIONALLY START THE
0271 548 : MODIFIED PAGE WRITER IF IT WILL SOLVE THE CURRENT DEFICIT PROBLEM.
0271 549 :
0271 550 : CALLING SEQUENCE:
0271 551 : BSB/JSB MMG$MPWCHECK
0271 552 :
0271 553 : INPUT PARAMETERS:
0271 554 : FP - NEGATIVE NUMBER OF TOTAL PAGES WANTED BY SYSTEM
0271 555 :
0271 556 : IMPLICIT INPUTS:
0271 557 : MODIFIED PAGE WRITER VALUES
0271 558 :
0271 559 : OUTPUT PARAMETERS:
0271 560 : FP UPDATED TO REFLECT HOW MUCH IS EXPECTED BY RUNNING MPW
0271 561 :
0271 562 : IMPLICIT OUTPUTS:
0271 563 : MODIFY PAGE WRITER LOW THRESHOLD MAY BE SET TO START MPW
0271 564 :
0271 565 : COMPLETION CODES:
0271 566 : NONE
0271 567 :
0271 568 : SIDE EFFECTS:
0271 569 : NONE
0271 570 :
0271 571 :--
0271 572 :
0271 573 MMG$MPWCHECK::
0000'CF B5 0271 574 TSTW W^SCH$GL_MFY LIM ; HAS LIMIT BEEN SET TO ZERO - LOW WORD
OF 12 0275 575 BNEQ 10$ ; NO CONTINUE WITH NORMAL CHECKS
0000'CF D5 0277 576 TSTL W^SCH$GL_MFYCNT ; MAKE SURE SOMETHING IS REALLY THERE
2B 13 027B 577 BEQL 30$ ; NOTHING TO DO
50 5D 01 78 027D 578 ASHL #1,FP,RO ; GET RID OF HIGH BIT
03 12 0281 579 BNEQ 10$ ; IT WASN'T THE "FREE BALANCE SLOT" FLAG
5D D4 0283 580 CLRL FP ; SIGNAL THE START OF THE MPW
05 0285 581 RSB
0000'CF 0000'CF D1 0286 582 10$: CMPL W^MPW$GL_THRESH,W^SCH$GL_MFYCNT; IS MPW VALID FOR GETTING MEMORY
19 14 028D 583 BGTR 30$ ; BRANCH IF NOT
50 0000'CF 0000'CF C3 028F 584 SUBL3 W^SCH$GL_MFYLOLIM,W^SCH$GL_MFYCNT,RO; WILL WRITING MODIFIED
OF 15 0297 585 BLEQ 30$ ; PAGES HELP?, BR IF NONE
50 5D C0 0299 586 ADDL FP,RO ; ARE THERE ENOUGH?
0A 19 029C 587 BLSS 30$ ; BR IF NOT AND AVOID FLUSHING LIST
0000'CF 0000'CF B0 029E 588 MOVW W^SCH$GL_MFYLOLIM,W^SCH$GL_MFY LIM ; RESET LOWER THRESHOLD
5D 50 D0 02A5 589 MOVL RO,FP ; INDICATE NEW DEFICIT
05 02A8 590 50$: RSB ; EXIT, MPW WILL START IF NECESSARY
02A9 591
02A9 592 .END

```

OSWPSCHED
Symbol table

- SWAP SCHEDULER

H 2

16-SEP-1984 00:40:12 VAX/VMS Macro V04-00
5-SEP-1984 03:45:41 [SYS.SRC]OSWPSCHED.MAR;1

Page 13
(5)

SY
VO

```

BIT... = 0000000B
CANDIDATE = 00000149 R 03
CEBSL_WQFL = 00000014
COMPUTE = 000000EE R 03
DORMANT_CHECK = 0000020A R 03
EVTS_SWPOUT ***** X 03
EXESGL_ABSTIM ***** X 03
MMGSEXTRADYNWS ***** X 03
MMG$GB_FREWFLGS ***** X 03
MMG$MPQCHECK 00000271 RG 03
MMG$M_NOLASTUPD ***** X 03
MMG$M_NOWAIT ***** X 03
MMG$SHRINKWS ***** X 03
MPW$GL_THRESH ***** X 03
NEXTCEB 00000169 R 03
NEXTPR 000001E4 R R 03
NEXTSTATE 0000014C R R 03
OSWPSTATE 00000000 R 03
PCBSB_PRI = 0000000B
PCBSL_PHD = 0000006C
PCBSL_SQBL = 00000004
PCBSL_STS = 00000024
PCBSL_WAITIME = 00000118
PCBSV_DISAWS = 00000018
PCBSV_INQUAN = 00000003
PCBSV_PSWAPM = 00000004
PCBSV_RES = 00000000
PCBSW_DIOCNT = 0000003E
PCBSW_DIOLM = 00000040
PCBSW_GPGCNT = 00000034
PCBSW_PPGCNT = 00000036
PHDSM_NO_WS_CHNG = 00000040
PHDSW_FLAGS = 00000036
PHDSW_SWAPSIZE = 00000052
PHDSW_WSLIST = 00000008
PHDSW_WSQUOTA = 00000018
PHDSW_WSSIZE = 00000050
RETRY 00000039 R 03
SAVED_R3 00000004 R 02
SCH$AQ_COMOH ***** X 03
SCH$AQ_WQHDR ***** X 03
SCH$C_CEF = 00000003
SCH$C_COLPG = 00000001
SCH$C_COM = 0000000C
SCH$C_FPG = 0000000B
SCH$C_HIB = 00000007
SCH$C_LEF = 00000005
SCH$C_MWAIT = 00000002
SCH$C_PFW = 00000004
SCH$C_SUSP = 00000009
SCH$GB_RESCAN ***** X 03
SCH$GL_FREECNT ***** X 03
SCH$GL_MFYCNT ***** X 03
SCH$GL_MFY LIM ***** X 03
SCH$GL_MFYLOLIM ***** X 03
SCH$GQ_CEBHD ***** X 03
SCH$GW_DORMANTWAIT ***** X 03

```

```

SCH$GW_LONGWAIT ***** X 03
SCH$GW_SWPFAIL ***** X 03
SCH$GW_SWPFCNT ***** X 03
SCH$OSWPSCHED 0000003C RG 03
SCH$RSE ***** X 03
SCH$V_REORD ***** X 03
SHRINKWS 00000239 R 03
SIZ... = 00000001
SWAPQUANCHK 000001EA R 03
SWP$GB_ISWPRI ***** X 03
SWP$GL_SWPPGCNT ***** X 03
SWP$GL_SWTIME 00000000 RG 02
SYSSGB_DEFPRI ***** X 03
TMP... = 00000090
WQH$L_WQBL = 00000004
WSCAN 00000188 R 03
-V_CEF = 00000000
-V_COMPUTE = 00000007
-V_DIOCNT = 00000001
-V_DORMANT = 0000000A
-V_INQUAN = 00000004
-V_LONGWAIT = 00000002
-V_NDIOCNT = 00000005
-V_PRIORITY = 00000006
-V_SHORTWAIT = 00000003
-V_SWAPASAP = 00000009
-V_SWPOGOAL = 00000008

```

! 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	00000008 (8.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
\$OSWPSCHED	000002A9 (681.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.04	00:00:01.24
Command processing	120	00:00:00.48	00:00:04.34
Pass 1	216	00:00:05.92	00:00:22.00
Symbol table sort	0	00:00:00.71	00:00:01.73
Pass 2	123	00:00:01.70	00:00:04.64
Symbol table output	11	00:00:00.08	00:00:00.08
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	505	00:00:08.96	00:00:34.07

The working set limit was 1200 pages.
33660 bytes (66 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 456 non-local and 39 local symbols.
592 source lines were read in Pass 1, producing 19 object records in Pass 2.
18 pages of virtual memory were used to define 17 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	7
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	13

505 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:OSWPSCHED/OBJ=OBJ\$:OSWPSCHED MSRC\$:OSWPSCHED/UPDATE=(ENH\$:OSWPSCHED)+EXECMLS/LIB

This image displays a grid of 100 small, illegible document thumbnails arranged in 10 rows and 10 columns. The thumbnails are arranged in a grid pattern. Several thumbnails contain legible text, including:

- Row 2, Column 5: PAGEFAULT LIS
- Row 4, Column 1: OSMPSCHE LIS
- Row 5, Column 8: PARAMETER LIS
- Row 6, Column 1: PLSYSVECT LIS
- Row 10, Column 5: PAGEFILE LIS

The remaining thumbnails are too small and faded to read.