



```

SSSSSSSS YY YY SSSSSSSS UU UU NN NN WW WW IIIIII NN NN DDDDDDDD
SSSSSSSS YY YY SSSSSSSS UU UU NN NN WW WW IIIIII NN NN DDDDDDDD
SS SS YY YY SS SSSSSSSS UU UU NN NN WW WW II NN NN DD DD
SS SS YY YY SS SSSSSSSS UU UU NN NN WW WW II NN NN DD DD
SS SS YY YY SS SSSSSSSS UU UU NN NN WW WW II NN NN DD DD
SSSSSSS YY YY SSSSSSSS UU UU NN NN WW WW II NN NN DD DD
SSSSSSS YY YY SSSSSSSS UU UU NN NN WW WW II NN NN DD DD
SS SS YY YY SS SSSSSSSS UU UU NN NN WW WW II NN NN DD DD
SSSSSSSS YY YY SSSSSSSS UUUUUUUUUU NN NN WWW WWW IIIIII NN NN DDDDDDDD
SSSSSSSS YY YY SSSSSSSS UUUUUUUUUU NN NN WW WW IIIIII NN NN DDDDDDDD

```

```

LL LL IIIIII SSSSSSSS
LL LL IIIIII SSSSSSSS
LL LL II
LL LL II
LL LL II
LL LL II
LL LL II
LL LL II
LL LL II
LL LL II
LL LL II
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```

(1) 86  
(1) 294

UNWIND PROCEDURE CALL STACK  
CALCULATE VALUE OF SP BEFORE CALL

G 15

```
0000 1 .TITLE SYSUNWIND - SYSTEM SERVICE UNWIND PROCEDURE CALL STACK
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 D. N. CUTLER 16-DEC-76
0000 29
0000 30 MODIFIED BY:
0000 31
0000 32 V03-006 ACG0383 Andrew C. Goldstein, 19-Dec-1983 16:15
0000 33 Fix register use bug in skipping active signal
0000 34
0000 35 V03-005 ACG0357 Andrew C. Goldstein, 9-Sep-1983 18:08
0000 36 Turn off T bit when unwinding through AST's and
0000 37 into the frame of the exception.
0000 38
0000 39 V03-004 ACG0348 Andrew C. Goldstein, 4-Aug-1983 17:07
0000 40 Fix unwinding into frame of exception
0000 41
0000 42 V03-003 ACG0336 Andrew C. Goldstein, 16-May-1983 15:48
0000 43 Add signal/stop longword to signal args in $$$_UNWIND
0000 44
0000 45 V03-002 KDM0002 Kathleen D. Morse 28-Jun-1982
0000 46 Add $PSLDEF.
0000 47
0000 48 V03-001 ACG0294 Andrew C. Goldstein, 10-Jun-1982 18:22
0000 49 Fix storing of PC into AST frame when unwinding thru AST
0000 50
0000 51
0000 52 **
0000 53
0000 54 SYSTEM SERVICE UNWIND PROCEDURE CALL STACK
0000 55
0000 56 MACRO LIBRARY CALLS
0000 57
```

```
0000 58
0000 59 $CHFDEF ;DEFINE CONDITION HANDLING ARGLIST OFFSETS
0000 60 $PSLDEF ;DEFINE PROGRAM STATUS LONGWORD
0000 61 $SSDEF ;DEFINE SYSTEM STATUS VALUES
0000 62
0000 63 ;
0000 64 ; LOCAL SYMBOLS
0000 65 ;
0000 66 ; ARGUMENT LIST OFFSET DEFINITIONS
0000 67 ;
0000 68
00000004 0000 69 DEPADR=4 ;ADDRESS OF NUMBER OF FRAMES TO UNWIND
00000008 0000 70 NEWPC=8 ;CHANGE OF FLOW FINAL RETURN ADDRESS
0000 71
0000 72 ;
0000 73 ; CALL FRAME OFFSET DEFINITIONS
0000 74 ;
0000 75
00000000 0000 76 HANDLER=0 ;CONDITION HANDLER ADDRESS
00000004 0000 77 SAVPSW=4 ;SAVED PSW FROM CALL
00000006 0000 78 SAVMSK=6 ;REGISTER SAVE MASK
00000008 0000 79 SAVAP=8 ;SAVED AP REGISTER IMAGE
0000000C 0000 80 SAVFP=12 ;SAVED FP REGISTER IMAGE
00000010 0000 81 SAVPC=16 ;SAVED PC REGISTER IMAGE
00000014 0000 82 SAVRG=20 ;OTHER SAVED REGISTER IMAGES
0000 83
00000000 84 .PSECT YEXEPAGED, BYTE ;PAGED PSECT
```

```

0000 86 .SBTTL UNWIND PROCEDURE CALL STACK
0000 87 :+
0000 88 : EXESUNWIND - UNWIND PROCEDURE CALL STACK
0000 89 :
0000 90 : THIS SERVICE PROVIDES THE CAPABILITY TO UNWIND THE PROCEDURE CALL STACK
0000 91 : TO A SPECIFIED DEPTH AFTER A HARDWARE- OR SOFTWARE-DETECTED EXCEPTION
0000 92 : CONDITION HAS BEEN SIGNALLED. OPTIONALLY A CHANGE OF FLOW RETURN ADDRESS
0000 93 : MAY ALSO BE SPECIFIED. THE ACTUAL UNWIND IS NOT PERFORMED IMMEDIATELY BY
0000 94 : THE SERVICE, BUT RATHER THE RETURN ADDRESSES IN THE CALL STACK ARE MODIFIED
0000 95 : SUCH THAT WHEN THE CONDITION HANDLER RETURNS THE UNWIND OCCURS.
0000 96 :
0000 97 : INPUTS:
0000 98 :
0000 99 :     DEPADR(AP) = ADDRESS OF NUMBER OF FRAMES TO UNWIND.
0000 100 :     NEWPC(AP) = CHANGE OF FLOW FINAL RETURN ADDRESS.
0000 101 :
0000 102 :     R4 = CURRENT PROCESS PCB ADDRESS.
0000 103 :
0000 104 : OUTPUTS:
0000 105 :
0000 106 :     R0 LOW BIT CLEAR INDICATES FAILURE TO FULLY UNWIND CALL STACK.
0000 107 :
0000 108 :     R0 = SSS_ACCVIO - CALL STACK NOT ACCESSIBLE TO CALLING ACCESS
0000 109 :     MODE.
0000 110 :
0000 111 :     R0 = SSS_INSFRAME - INSUFFICIENT CALL FRAMES TO UNWIND TO
0000 112 :     SPECIFIED DEPTH.
0000 113 :
0000 114 :     R0 = SSS_NOSIGNAL - NO SIGNAL IS CURRENTLY ACTIVE TO UNWIND.
0000 115 :
0000 116 :     R0 = SSS_UNWINDING - UNWIND ALREADY IN PROGRESS.
0000 117 :
0000 118 :     R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
0000 119 :
0000 120 :     R0 = SSS_NORMAL - NORMAL COMPLETION.
0000 121 : -
0000 122 :
0000 123 .ENTRY EXESUNWIND, ^M<R2,R3,R4,R5>
6D 0000'CF 003C 0002 124 MOVAB W^EXESSIGTORET, (FP) ;ESTABLISH CONDITION HANDLER
54 5D D0 0007 125 MOVL FP,R4 ;SET ADDRESS OF FIRST FRAME TO EXAMINE
000A 126
000A 127 :
000A 128 : SEARCH CALL STACK FOR A FRAME THAT WAS CREATED BY A CALL FROM THE SIGNAL
000A 129 : DISPATCH VECTOR OR BY A CALL FROM THE UNWIND SIGNAL DISPATCHER.
000A 130 :
000A 131 :
50 0900 8F 3C 000A 132 MOVZWL #SS$ NOSIGNAL, R0 ;ASSUME NO SIGNAL ACTIVE
54 0C A4 D0 000F 133 10$: MOVL SAVFP(R4), R4 ;GET ADDRESS OF PREVIOUS FRAME
19 13 0013 134 BEQL 20$ ;IF EQL END OF CALL STACK
10 A4 00000004'8F D1 0015 135 CMPL #SYS$CALL_HANDL+4, SAVPC(R4) ;CALL FROM CONDITION HANDLER DISPATCHER?
10 13 001D 136 BEQL 30$ ;IF EQL YES
10 A4 00000122'8F D1 001F 137 CMPL #CALLUNWIND+4, SAVPC(R4) ;CALL FROM UNWIND SIGNAL DISPATCHER?
E6 12 0027 138 BNEQ 10$ ;IF NEQ NO
50 0928 8F 3C 0029 139 MOVZWL #SS$_UNWINDING, R0 ;SET ALREADY UNWINDING
04 002E 140 20$: RET ;
002F 141
002F 142 :

```

```

002F 143 ; SET TO UNWIND PROCEDURE CALL STACK TO SPECIFIED DEPTH
002F 144 ;
002F 145 ;
53 04 AC D0 002F 146 30$: MOVL DEPADR(AP),R3 ;GET ADDRESS OF NUMBER OF FRAMES TO UNWIND
05 13 0033 147 BEQL 40$ ;IF EQL NONE SPECIFIED
53 63 D0 0035 148 MOVL (R3),R3 ;GET NUMBER OF FRAMES TO UNWIND
OF 11 0038 149 BRB 50$ ;
52 54 D0 003A 150 40$: MOVL R4,R2 ;COPY CURRENT FRAME ADDRESS
011A 30 003D 151 BSBW OLDSP ;CALCULATE VALUE OF SP BEFORE CALL
52 0C A2 D0 0040 152 MOVL CHFSL_MCHARGLIST+4(R2),R2 ;GET ADDRESS OF MECHANISM ARGUMENT LIST
53 08 A2 01 C1 0044 153 ADDL3 #1,CHFSL_MCH_DEPTH(R2),R3 ;CALCULATE DEPTH OF ESTABLISHER'S CALLER
53 15 0049 154 50$: BLEQ 90$ ;IF LEQ NO FRAMES TO REMOVE
50 00FC'CF DE 004B 155 MOVAL W*STARTUNWIND,R0 ;SET CONDITION HANDLER UNWIND ADDRESS
0087 30 0050 156 BSBW SETPC ;
0053 157 ;
0053 158 ;
0053 159 ; SCAN THROUGH SPECIFIED NUMBER OF FRAMES SETTING EACH FRAME TO UNWIND ON RETURN
0053 160 ;
0053 161 ;
54 0C A4 D0 0053 162 60$: MOVL SAVFP(R4),R4 ;GET ADDRESS OF PREVIOUS FRAME
7B 13 0057 163 BEQL 100$ ;IF EQL INSUFFICIENT FRAMES
53 D7 0059 164 DECL R3 ;ANY MORE FRAMES TO CONSIDER?
05 14 005B 165 BGTR 65$ ;BRANCH IF YES
04 AC D5 005D 166 TSTL DEPADR(AP) ;ARE WE UNWINDING TO CALLER OF ESTABLISHER?
3C 13 006C 167 BEQL 90$ ;BRANCH IF YES - DON'T TOUCH HANDLER FRAMES
10 A4 00000004'8F D1 0062 168 65$: CMPL #SYSSCALL_HANDL+4,SAVPC(R4) ;CALL FROM CONDITION DISPATCHER?
26 12 006A 169 BNEQ 80$ ;IF NEQ NO
52 54 D0 006C 170 MOVL R4,R2 ;COPY ADDRESS OF CURRENT FRAME
00E8 30 006F 171 BSBW OLDSP ;CALCULATE VALUE OF SP BEFORE CALL
52 0C A2 D0 0072 172 MOVL CHFSL_MCHARGLIST+4(R2),R2 ;GET ADDRESS OF MECHANISM ARGUMENT LIST
08 A2 D5 0076 173 TSTL CHFSL_MCH_DEPTH(R2) ;CHECK IF THIS IS A VECTORED HANDLER
17 19 0079 174 BLSS 80$ ;IF SO, DON'T SKIP ANY FRAMES
55 04 A2 D0 007B 175 MOVL CHFSL_MCH_FRAME(R2),R5 ;GET ESTABLISHER FP
55 54 D1 007F 176 70$: CMPL R4,R5 ;UNWOUND TO ESTABLISHER FRAME?
DE 13 0082 177 BEQL 65$ ;IF EQL YES
50 04 'AF DE 0084 178 MOVAL B*LOOPUNWIND,R0 ;SET FRAME UNWIND ADDRESS
50 10 0088 179 BSBW SETPC ;
54 0C A4 D0 008A 180 MOVL SAVFP(R4),R4 ;GET ADDRESS OF PREVIOUS FRAME
EF 12 008E 181 BNEQ 70$ ;LOOP AS LONG AS LINKAGE IS VALID
42 11 0090 182 BRB 100$ ;ZERO FP - BAD STACK
0092 183 ;
53 D5 0092 184 80$: TSTL R3 ;ANY MORE FRAMES TO CONSIDER?
08 15 0094 185 BLEQ 90$ ;IF LEQ NO
50 04 'AF DE 0096 186 MOVAL B*LOOPUNWIND,R0 ;SET FRAME UNWIND ADDRESS
3E 10 009A 187 BSBW SETPC ;
B5 11 009C 188 BRB 60$ ;
009E 189 ;
009E 190 ;
009E 191 ; MODIFY CHANGE OF FLOW RETURN IF NEW ADDRESS SPECIFIED
009E 192 ;
009E 193 ;
50 08 AC D0 009E 194 90$: MOVL NEWPC(AP),R0 ;GET CHANGE OF FLOW RETURN ADDRESS
00000004'8F 2C 13 00A2 195 BEQL 95$ ;IF EQL NONE SPECIFIED
10 A4 D1 00A4 196 CMPL SAVPC(R4),#SYSSCALL_HANDL+4 ;CHECK IF THIS IS A HANDLER FRAME
20 12 00AC 197 BNEQ 94$ ;BRANCH IF NOT
00AE 198 ;
00AE 199 ; TO HERE IF THE FRAME BEING UNWOUND TO IS THE FRAME IN WHICH THE SIGNAL

```

```

00AE 200 : OCCURRED (DETECTED BY THE FACT THAT SAVPC IN THE FRAME ABOUT TO BE
00AE 201 : PLUGGED IS THE HANDLER CALL SITE). IF WE ARE STORING A NEW PC HERE,
00AE 202 : WE HAVE TO TAKE SPECIAL HANDLING SO THAT THE SIGNAL ARGUMENT LIST IS
00AE 203 : REMOVED BEFORE ALLOWING EXECUTION TO PROCEED. THIS IS DONE BY STORING
00AE 204 : THE DESIRED NEW PC IN THE PC OF THE SIGNAL VECTOR, AND SETTING THE
00AE 205 : RETURN PC TO A PIECE OF CODE THAT WILL CLEAN UP THE EXCEPTION.
00AE 206 :
      52 54 DO 00AE 207 10$: MOVL R4,R2 ;GET POINTER TO STACK FRAME
      00A6 30 00B1 208 BSBW OLDSP ;FIND THE EXCEPTION ARG LIST
      52 08 A2 DO 00B4 209 MOVL CHF$L_SIGARGLST+4(R2),R2 ;GET SIGNAL VECTOR
      51 FF A2 9A 00B8 210 MOVZBL -1(R2),R1 ;GET ORIGINAL SIG ARG COUNT
6241 FC A241 50 DO 00BC 211 MOVL R0,-4(R2)[R1] ;STORE PC IN NEXT TO LAST LONGWORD
      (8000010 8F CA 00C1 212 BICL #PSL$M_CM!PSL$M_FPD!PSL$M_TBIT!PSL$M_TP,(R2)[R1]
      00C9 213 ;CLEAN OUT OCD INSTRUCTION CONTEXT
      50 0155'CF 9E 00C9 214 MOVAB W^REMSIGNAL,R0 ;ADDRESS OF CLEANUP ROUTINE
      00CE 215
      50 0A 10 00CE 216 94$: BSBB SETPC ;SET NEW FINAL RETURN ADDRESS
      50 01 3C 00D0 217 95$: MOVZWL #SS$ _NORMAL,R0 ;SET NORMAL COMPLETION
      04 00D3 218 RET ;
      50 012C 8F 3C 00D4 219 100$: MOVZWL #SS$ _INSFRAME,R0 ;SET INSUFFICIENT FRAMES
      04 00D9 220 RET ;
      00DA 221
      00DA 222 :
      00DA 223 : SUBROUTINE TO STORE UNWIND PC. IT CHECKS IF THE FRAME BEING ALTERED
      00DA 224 : IS AN AST CALL FRAME. RATHER THAN PLUG ITS RETURN PC, WE LET
      00DA 225 : IT RETURN TO THE AST DISPATCHER, WHO WILL DISMISS THE AST. INSTEAD,
      00DA 226 : WE PLUG THE INTERRUPT PC OF THE AST, SO THE REI GOES BACK TO
      00DA 227 : LOOPUNWIND TO CONTINUE WITH THE AST DISMISSED.
      00DA 228 :
      00DA 229
00000000'8F 10 A4 D1 00DA 230 SETPC: CMLP SAVPC(R4),#EXESASTRET ;CHECK IF FRAME IS AN AST
      05 13 00E2 231 BEQL 10$ ;BRANCH IF YES
      10 A4 50 DO 00E4 232 MOVL R0,SAVPC(R4) ;SET FRAME UNWIND ADDRESS
      05 00E8 233 RSB ;
      00E9 234
      52 54 DO 00E9 235 10$: MOVL R4,R2
      006B 30 00EC 236 BSBW OLDSP ;FIND THE START OF THE AST ARG LIST
      10 A2 50 DO 00EF 237 MOVL R0,16(R2) ;AND STUFF THE AST PC
14 A2 (8000010 8F CA 00F3 238 BICL #PSL$M_CM!PSL$M_FPD!PSL$M_TBIT!PSL$M_TP,20(R2)
      00FB 239 ;CLEAN OUT OCD INSTRUCTION CONTEXT
      05 00FB 240 RSB ;
      00FC 241 :
      00FC 242 : UNWIND HANDLER FRAME
      00FC 243 :
      00FC 244
      50 0C AE DO 00FC 245 STARTUNWIND: ;START OF ACTUAL UNWIND
      50 0C A0 7D 0100 246 MOVL CHF$L_MCHARGLST+4(SP),R0 ;GET ADDRESS OF MECHANISM ARGUMENT LIST
      0104 247 MOVQ CHF$L_MCH_SAVR0(R0),R0 ;RESTORE REGISTERS R0 AND R1
      0104 248
      0104 249 :
      0104 250 : UNWIND CALL FRAME SIGNALING CONDITION HANDLER IF ONE IS SPECIFIED
      0104 251 :
      0104 252
      0104 253 .ENABL LSB
      0104 254 LOOPUNWIND: ;UNWIND CALL FRAME
      6D 05 0104 255 TSTL (FP) ;CONDITION HANDLER SPECIFIED?
      1E 13 0106 256 BEQL 10$ ;IF EQL NO

```



```

7E 0920 8F 3C 0108 257 MOVZWL #SS$_UNWIND,-(SP) ;PUSH UNWIND SIGNAL CONDITION
      01 DD 010D 258 PUSHL #1 ;PUSH NUMBER OF SIGNAL ARGUMENTS
      02 DD 010F 259 PUSHL #2 ;PUSH STOP INDICATOR
      03 BB 0111 260 PUSHR #*M<R0,R1> ;PUSH REGISTERS R0 AND R1
      00 DD 0113 261 PUSHL #0 ;PUSH FRAME DEPTH
      5D DD 0115 262 PUSHL FP ;PUSH FRAME ADDRESS
      04 DC 0117 263 PUSHL #4 ;PUSH NUMBER OF MECHANISM ARGUMENTS
      6E DF 0119 264 PUSHAL (SP) ;PUSH ADDRESS OF MECHANISM ARGUMENTS
      1C AE DF 011B 265 PUSHAL 28(SP) ;PUSH ADDRESS OF SIGNAL ARGUMENTS
      011E 266 CALLUNWIND: ;SIGNAL UNWIND
      00 BD 02 FB 011E 267 CALLS #2,@(FP) ;CALL CONDITION HANDLER
      50 OC AE 7D 0122 268 MOVQ CH$_L_MCH_SAVRO(SP),R0 ;RETRIEVE NEW VALUES FOR R0 AND R1
      5C 14 AD DE 0126 269 10$: MOVAL SAVRG(FP),AP ;GET ADDRESS OF REGISTER SAVE AREA
      03 06 AD E9 012A 270 BLBC SAVMSK(FP),20$ ;IF LBC R0 NOT SAVED
      8C 50 DO 012E 271 MOVL R0,(AP)+ ;SAVE R0 FOR SUBSEQUENT RESTORATION
      03 06 AD 01 E1 0131 272 20$: BBC #1,SAVMSK(FP),30$ ;IF CLR, R1 NOT SAVED
      6C 51 DO 0136 273 MOVL R1,(AP) ;SAVE R1 FOR SUBSEQUENT RESTORATION
00000000'8F 10 AD D1 0139 274 30$: CMPL SAVPC(FP),#EX$_ASTRET ;ABOUT TO UNWIND AN AST?
      11 12 0141 275 BNEQ 40$ ;BRANCH IF NOT
      06 BB 0143 276 PUSHR #*M<R1,R2> ;SAVE R1 AND R2
      52 5D DO 0145 277 MOVL FP,R2
      10 10 0148 278 BSBB OLDSP ;FIND THE AST PARAMETER LIST
      51 8ED0 014A 279 POPL R1 ;GET BACK R1
      08 A2 50 7D 014D 280 MOVQ R0,8(R2) ;STUFF R0 AND R1 SO THEY WILL PASS THROUGH
      52 8ED0 0151 281 POPL R2 ;RESTORE R2
      04 0154 282 40$: RET
      0155 283
      0155 284
      0155 285 ; UNWIND TRANSFERS TO HERE WHEN A NEW PC IS GIVEN IN UNWINDING TO THE
      0155 286 ; FRAME IN WHICH THE SIGNAL OCCURRED. WE MUST CLEAN THE EXCEPTION OFF
      0155 287 ; THE STACK BEFORE RESUMING EXECUTION.
      0155 288
      0155 289 REMSIGNAL:
      8E D5 0155 290 TSTL (SP)+ ;REMOVE JSB RETURN ADDRESS
      FEAG' 31 0157 291 BRW EX$_CONTSIGNAL ;LET EXCEPTION MODULE DO THE REST
      015A 292 .DSABL LSB

```



SYUNWIND  
Symbol table

```

CALLUNWIND          = 0000011E R    02
CHFSL_MCHARGLST    = 00000008
CHFSL_MCH_DEPTH    = 00000008
CHFSL_MCH_FRAME    = 00000004
CHFSL_MCH_SAVRO    = 0000000C
CHFSL_SIGARGLST    = 00000004
DEPADR             = 00000004
EXESASTRET         ***** X    02
EXESCONTSIGNAL     ***** X    02
EXESSIGTORET       ***** X    02
EXESUNWIND         = 00000000 RG   02
HANDLER            = 00000000
LOOPUNWIND         = 00000104 R    02
NEWPC              = 00000008
OLDSP              = 0000015A R    02
PSLSM_CM           = 80000000
PSLSM_FPD          = 08000000
PSLSM_TBIT         = 00000010
PSLSM_TP           = 40000000
REMSIGNAL          = 00000155 R    02
SAVAP              = 00000008
SAVFP              = 0000000C
SAVMSK             = 00000006
SAVPC              = 00000010
SAVPSW            = 00000004
SAVRG              = 00000014
SETPC              = 000000DA R    02
SSS_IN$FRAME       = 0000012C
SSS_NORMAL         = 00000001
SSS_NOSIGNAL       = 00000900
SSS_UNWIND         = 00000920
SSS_UNWINDING      = 00000928
STARTUNWIND        = 000000FC R    02
SYSSCALL_HANDL     ***** X    02
  
```

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

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
YEXEPAGED	0000017A ( 378.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.25
Command processing	106	00:00:00.57	00:00:01.13
Pass 1	218	00:00:04.65	00:00:05.72
Symbol table sort	0	00:00:00.70	00:00:00.72
Pass 2	70	00:00:01.11	00:00:01.26
Symbol table output	6	00:00:00.05	00:00:00.05

Psect synopsis output 0 00:00:00.02 00:00:00.02  
 Cross-reference output 0 00:00:00.00 00:00:00.00  
 Assembler run totals 431 00:00:07.17 00:00:09.15

The working set limit was 1200 pages.  
 26747 bytes (53 pages) of virtual memory were used to buffer the intermediate code.  
 There were 30 pages of symbol table space allocated to hold 479 non-local and 21 local symbols.  
 309 source lines were read in Pass 1, producing 17 object records in Pass 2.  
 10 pages of virtual memory were used to define 9 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYS.LIB]STARLET.MLB;2	6
TOTALS (all libraries)	6

529 GETS were required to define 6 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SYSUNWIND/OBJ=OBJ\$:SYSUNWIND MSRC\$.SYSUNWIND/UPDATE=(ENH\$:SYSUNWIND)+EXECML\$/LIB


