


```

UU      UU  EEEEEEEEE  TTTTTTTTT  MM      MM  EEEEEEEEE  MM      MM  YY      YY  000000  11
UU      UU  EEEEEEEEE  TTTTTTTTT  MM      MM  EEEEEEEEE  MM      MM  YY      YY  000000  11
UU      UU  EE          TT          MMMM   MMMM  EE          MM      MM  YY      YY  00      00  1111
UU      UU  EE          TT          MMMM   MMMM  EE          MM      MM  YY      YY  00      00  1111
UU      UU  EE          TT          MM  MM   MM  EE          MM      MM  YY      YY  00      0000  11
UU      UU  EE          TT          MM  MM   MM  EE          MM      MM  YY      YY  00      0000  11
UU      UU  EEEEEEEEE  TT          MM      MM  EEEEEEEEE  MM      MM  YY      YY  00      00  11
UU      UU  EEEEEEEEE  TT          MM      MM  EEEEEEEEE  MM      MM  YY      YY  00      00  11
UU      UU  EE          TT          MM      MM  EE          MM      MM  YY      YY  0000  00  11
UU      UU  EE          TT          MM      MM  EE          MM      MM  YY      YY  0000  00  11
UU      UU  EE          TT          MM      MM  EE          MM      MM  YY      YY  00      00  11
UU      UU  EE          TT          MM      MM  EE          MM      MM  YY      YY  00      00  11
UUUUUUUUUU  EEEEEEEEE  TT          MM      MM  EEEEEEEEE  MM      MM  YY      YY  000000  111111
UUUUUUUUUU  EEEEEEEEE  TT          MM      MM  EEEEEEEEE  MM      MM  YY      YY  000000  111111

```

```

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
LLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLL IIIIII  SSSSSSSS

```

(2)	53	Include Files
(2)	60	Macros
(2)	69	Equates
(2)	80	Declarations
(3)	95	ASCII string definitions
(4)	123	Main Program
(4)	157	AST Handler
(5)	163	System Service Exception Handler
(6)	271	RMS Error Handler
(7)	334	Error Exit

```

0000 1 .Title UETMEMY01 PAGING LOAD
0000 2 .ident 'V04-000'
0000 3 .default displacement WORD
0000 4 .enable suppression
0000 5
0000 6 *****
0000 7 *
0000 8 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 9 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 10 * ALL RIGHTS RESERVED. *
0000 11 *
0000 12 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 13 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 14 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 15 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 16 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 17 * TRANSFERRED. *
0000 18 *
0000 19 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 20 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 21 * CORPORATION. *
0000 22 *
0000 23 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 24 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 25 *
0000 26 *
0000 27 *****
0000 28
0000 29 ++
0000 30 ABSTRACT: The UETMEMY01 module does an EXPAND REGION to the size of the
0000 31 maximum working set size. It then touches the pages within
0000 32 that region randomly by writing to them (PAGE FAULT) for a
0000 33 set time period.
0000 34 The intent is to use, (or misuse), the virtual memory
0000 35 management facility very inefficiently in order to simulate
0000 36 a heavy user environment. The program is intended to be one
0000 37 part of a test package aimed at using up system resources.
0000 38
0000 39 ENVIRONMENT: User mode only.
0000 40 No particular privileges are required by the module.
0000 41
0000 42 --
0000 43
0000 44 AUTHOR: Fred Matthes CREATION DATE: December, 1981
0000 45
0000 46 MODIFIED BY:
0000 47
0000 48 V03-001 RNH0001 Richard N. Holstein, 26-Jun-1984
0000 49 Take advantage of new UETP error message codes.
0000 50
0000 51 **

```

```

0000 53      .sbttl Include Files
0000 54      :
0000 55      INCLUDE FILES:
0000 56      :
0000 57      SYSS$LIBRARY:LIB.MLB      for general definitions
0000 58      SHRLIB$:UETP.MLB        for UETP definitions
0000 59      :
0000 60      .sbttl Macros
0000 61      :
0000 62      MACROS:
0000 63      :
0000 64      $schdef                    ; Condition handler frame definitions
0000 65      $shrdef                    ; Shared messages definitions
0000 66      $jpiddef                   ; Job process information definitions
0000 67      $uetpdef                   ; UETP definitions
0000 68      $stsdef                    ; Status return definitions
0000 69      .sbttl Equates
0000 70      :
0000 71      Equated symbols
0000 72      :
00000001 0000 73 RMS_K = 1
00740000 0000 74 UETP = UETPS$FACILITY@STSSV_FAC_NO ; define UETP facility code
007410E0 0000 75 UETPS$ABENDD = UETP!SHRS$ABENDD ; Define the UETP message codes
00741038 0000 76 UETPS$BEGINDD = UETP!SHRS$BEGINDD
00741080 0000 77 UETPS$ENDEDD = UETP!SHRS$ENDEDD
00741130 0000 78 UETPS$TEXT = UETP!SHRS$TEXT
00000084 0000 79 TEXT_BUFFER = 132 ; buffer size
0000 80      .sbttl Declarations
0000 81      :
0000 82      Declarations
0000 83      :
FFFFFFFF B8797400 0000 84 TWO_MIN: .long -10*1000*1000*120,-1 ; 120 seconds
          00 0008 85 TIME_FLAG: .byte 0
          71269417 0009 86 SEED: .long ^X71269417
          00000200 000D 87 VAS_PAGCNT: .long 512 ; # pages we grab each time
00000000 00000000 0011 88 BEG$END_ADD: .quad ; for ret beg/end address
          0004 0019 89 GTLIST: .word 4
          0416 001B 90 .word JPI$WSEXTENT ; max working set size
          0000000D 001D 91 .long VAS_PAGCNT
          00000000 0021 92 .long 0
          00000000 0025 93 .long 0 ; end of list

```

```

        65 6C 69 66 00000031'010E0000' 0029 95 .sbtll ASCII string definitions
64 72 6F 63 65 72 0000003D'010E0000' 0029 96 FILE: .ASCID /file/ ; Fills in RMS_ERR_STRING
        0035 97
        0035 98 RECORD: .ASCID /record/ ; Fills in RMS_ERR_STRING
        0043 99
        0043 100 RMS_ERR_STRING: ; Announces an RMS error
66 20 6E 69 20 72 6F 72 72 65 20 53 0043 101 .ASCID /RMS !AS error in file !AD/
        44 41 21 20 65 6C 69 005D
        0064 102 FAO_BUF: ; FAO output string descriptor
0000 0084 0064 103 .WORD TEXT_BUFFER,0
00000074' 0068 104 .ADDRESS BUFFER
        006C 105
        006C 106 BUFFER_PTR: ; Fake .ASCID buffer for misc. strings
0000 0084 006C 107 .WORD TEXT_BUFFER,0 ; A word for length, a word for desc.
00000074' 0070 108 .ADDRESS BUFFER
        0074 109
000000F8 0074 110 BUFFER: .BLKB TEXT_BUFFER ; FAO output and other misc. buffer
        00F8 111
00000000 00F8 112 STATUS: .LONG 0 ; Status value on program exit
        00FC 113
00000100 00FC 114 MSG_BLOCK: ; Auxiliary $GETMSG info
        0100 115 .BLKB 4
00000000 0100 116 ARG_COUNT: ; Argument counter used by ERROR_EXIT
        0104 117 .LONG 0
00000000 0104 118 ERROR_COUNT: ; Cumulative error count at runtime
        0108 119 .LONG 0
4D 45 4D 54 45 55 00000110'010E0000' 0108 120 PROCESS_NAME: ; Process name
        31 30 59 0116 121 .ASCID /UETMEMY01/

```

```

0000 0119 123
6D 01D1'CF DE 0119 124
0118 125
0120 126
0129 127
013E 128
013E 129
013E 130
FEBC CF C3 0151 131
FEBD CF 0155 132
53 0158 133
53 D6 0159 134
55 53 F7 8F 78 015B 135
52 FEAD CF D0 0160 136
53 00000080 8F D0 0165 137
54 55 4E 016C 138
FE95 CF 94 016F 139
0173 140
0173 141
57 D4 0186 142
0186 143
0188 144
FE7D CF DF 0188 145
00000000'GF 01 FB 018C 146
56 50 54 45 0193 147
56 56 4A 0197 148
56 53 C4 019A 149
6246 57 D0 019D 150
E3 57 55 F2 01A1 151
DC FE5F CF E9 01A5 152
01AA 153
50 00000000'8F D0 01B9 154
01C0 155
01C9 156
01C9 157
01C9 158
FE38 CF 01 0000 01C9 159
88 01CB 160
04 01D0 161

```

```

Main Program
.entry UETMEMY01,^M<>
mval SSERROR,(FP)
$SETSFM_S enbflg = #1
$GETJPI_S itmlst = GTLIST
$EXPREG_S pagcnt = VAS_PAGCNT,-
retadr = BEG$END_ADD,-
region = #0
subl3 BEG$END_ADD,-
BEG$END_ADD+4,-
R3
incl R3
ashl #-9,R3,R5
movl BEG$END_ADD,R2
movl #128,R3
cvtfl R5,R4
clrb TIME_FLAG
$SETIMR_S daytim = TWO_MIN,-
astadr = SET_TIMEOUT_FLAG
OUTER_LOOP:
clrl R7
LOOP:
pushal SEED
calls #1,G^MTH$RANDOM
mulf3 R4,R0,R6
cvtfl R6,R6
mull R3,R6
movl R7,(R2)[R6]
aoblss R5,R7,LOOP
blbc TIME_FLAG,OUTER_LOOP
$DELTVA_S inadr = BEG$END_ADD
movl #SS$NORMAL,R0
$EXIT_S R0
.sbtll AST Handler
SET_TIMEOUT_FLAG:
.word 0
bisb #1,TIME_FLAG
RET

```

```

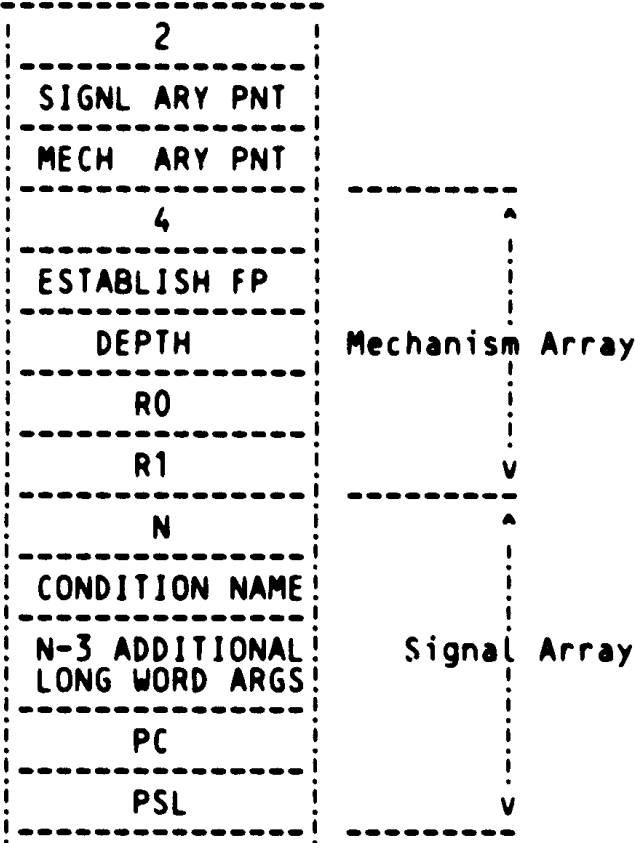
; entry mask
; Declare exception handler
; Enable system service failure mode
; get # of pages
; expand our region
; fig # bytes actually got
; <last-first> + 1
; (div by 512) loop count
; address
; longwords per page
; page offset into VAS
; clear time signal flag
; start the clock ticking
; plant seed
; get random number
; 0 to .999 x PAGES
; fixed point
; byte offset into VAS
; touch the page (FAULT IT)
; any time left?
; release all expanded pages
; see Y'all
; AST handler

```

```

01D1 163 .SBTTL System Service Exception Handler
01D1 164 :++
01D1 165 : FUNCTIONAL DESCRIPTION:
01D1 166 : This routine is executed if a software or hardware exception occurs or
01D1 167 : if a LIB$SIGNAL system service is used to output a message.
01D1 168 : Information about this method of handling messages and errors can be
01D1 169 : found in the VMS COMMON RUN-TIME manual and in the VMS SYSTEM SERVICE
01D1 170 : manual.
01D1 171 :
01D1 172 : CALLING SEQUENCE:
01D1 173 : Entered via an exception from the system
01D1 174 :
01D1 175 : INPUT PARAMETERS:
01D1 176 : ERROR_COUNT = previous cumulative error count
01D1 177 :
01D1 178 : AP ---->
01D1 179 :
01D1 180 :     2
01D1 181 :     SIGNAL ARY PNT
01D1 182 :     MECH ARY PNT
01D1 183 :
01D1 184 :     4
01D1 185 :     ESTABLISH FP
01D1 186 :     DEPTH
01D1 187 :     R0
01D1 188 :     R1
01D1 189 :     N
01D1 190 :     CONDITION NAME
01D1 191 :     N-3 ADDITIONAL
01D1 192 :     LONG WORD ARGS
01D1 193 :     PC
01D1 194 :     PSL
01D1 195 :
01D1 196 : IMPLICIT INPUTS:
01D1 197 : NONE
01D1 198 :
01D1 199 : OUTPUT PARAMETERS:
01D1 200 : NONE
01D1 201 :
01D1 202 : IMPLICIT OUTPUTS:
01D1 203 : NONE
01D1 204 :
01D1 205 : COMPLETION CODES:
01D1 206 : NONE
01D1 207 :
01D1 208 : SIDE EFFECTS:
01D1 209 : NONE
01D1 210 :
01D1 211 :
01D1 212 :
01D1 213 :
01D1 214 :
01D1 215 :
01D1 216 :
01D1 217 :
01D1 218 :
01D1 219 : --

```




```

01D1 220
OFFC 01D1 221 SSERROR:
      01D1 222 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; Entry mask
      01D3 223
      01D3 224 $SETSFM_S ENBFLG = #0 ; Disable SS failure mode for PUTMSG
      01DC 225 $SETAST_S ENBFLG = #0 ; ASTs can play havoc with cond handler
56 04 AC D0 01E5 226 MOVL CHF$SIG_ARG1(R6) ; Get the signal array pointer
      OC 10 ED 01E9 227 CMPZV #STSSV_FAC_NO,#STSS$FAC_NO,- ; Is this a message from LIB$SIGNAL?
00000074 BF 04 A6 01EC 228 CHF$SIG_NAME(R6),#UETPS_FACILITY
      25 12 01F3 229 BNEQ 10$ ; BR if this is a system exception
      66 02 C2 01F5 230 SUBL2 #2,CHF$SIG_ARGS(R6) ; Drop the PC and PSL
      01F8 231 $PUTMSG_S MSGVEC = CHF$SIG_ARGS(R6) ; Print the message
      0207 232 $SETSFM_S ENBFLG = #1 ; Enable SS failure mode
      0210 233 $SETAST_S ENBFLG = #1 ; Enable ASTs
      04 0219 234 RET ; Return to the program
      021A 235 10$:
      01 OC 10 ED 021A 236 CMPZV #STSSV_FAC_NO,#STSS$FAC_NO,- ; Is it an RMS failure?
      01 08 A6 021D 237 CHF$SIG_ARG1(R6),#RMS_R
      0A 12 0220 238 BNEQ 20$ ; BR if not
      04 0222 239 $SETAST_S ENBFLG = #1 ; Enable ASTs
      022B 240 RET ; Yes, RMS_ERROR gets to handle them
      022C 241 20$:
FEC6 CF 04 A6 D0 022C 242 MOVL CHF$SIG_NAME(R6),STATUS ; Save the status
      58 D4 0232 243 CLRL R8 ; Assume for now it's not SS failure
FEBB CF 00000000 8F D1 0234 244 CMPL #SS$SSFAIL,STATUS ; But is it a System Service failure?
      3B 12 023D 245 BNEQ 40$ ; BR if not - no special case message
      023F 246 $GETMSG_S MSGID = CHF$SIG_ARG1(R6),- ; Get SS failure code associated text
      023F 247 MSGLEN = BUFFER_PTR,-
      023F 248 BUFADR = FAO_BUF,-
      023F 249 FLAGS = #14,-
      023F 250 OUTADR = MSG_BLOCK
FEA2 CF 95 0257 251 TSTB MSG_BLOCK+1 ; Get FAO arg count for SS failure code
      17 13 025B 252 BEQL 30$ ; Don't use $GETMSG if no $FAO args...
FE0B CF DF 025D 253 PUSHAL BUFFER_PTR ; ...else build up...
      01 DD 0261 254 PUSHL #1 ; ...a message describing...
00741130 8F DD 0263 255 PUSHL #UETPS_TEXT ; ...why the System Service failed
      08 A6 F0 0269 256 INSV CHF$SIG_ARG1(R6),- ; Give the message...
      00 026C 257 #STSSV_SEVERITY,- ; ...the correct severity code
      6E 03 026D 258 #STSS$SEVERITY,(SP)
      58 03 D0 026F 259 MOVL #3,R8 ; Count the number of args we pushed
      06 11 0272 260 BRB 40$
      0274 261 30$:
      08 A6 DD 0274 262 PUSHL CHF$SIG_ARG1(R6) ; Save SS failure code
      58 01 D0 0277 263 MOVL #1,R8 ; Count the number of args we pushed
      027A 264 40$:
57 66 04 C5 027A 265 MULL3 #4,CHF$SIG_ARGS(R6),R7 ; Convert longwords to bytes
      SE 57 C2 027E 266 R7,SP ; Save the current signal array...
6E 04 A6 57 28 0281 267 MOVCS R7,CHF$SIG_NAME(R6),(SP) ; ...on the stack
      7E 66 58 C1 0286 268 ADDL3 R8,CHF$SIG_ARGS(R6),-(SP) ; Push the current arg count
      0078 31 028A 269 BRW ERROR_EXIT ; Exit in error

```

```

028D 271 .SBTTL RMS Error Handler
028D 272 :++
028D 273 : FUNCTIONAL DESCRIPTION:
028D 274 : This routine handles error returns from RMS calls.
028D 275 :
028D 276 : CALLING SEQUENCE:
028D 277 : Called by RMS when a file processing error is found.
028D 278 :
028D 279 : INPUT PARAMETERS:
028D 280 : NONE
028D 281 :
028D 282 : IMPLICIT INPUTS:
028D 283 : The FAB or RAB associated with the RMS call.
028D 284 :
028D 285 : OUTPUT PARAMETERS:
028D 286 : NONE
028D 287 :
028D 288 : IMPLICIT OUTPUTS:
028D 289 : Error message
028D 290 :
028D 291 : COMPLETION CODES:
028D 292 : NONE
028D 293 :
028D 294 : SIDE EFFECTS:
028D 295 : Program may exit, depending on severity of the error.
028D 296 :
028D 297 :--
028D 298
028D 299 RMS_ERROR:
OFFC 028D 300 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; Entry mask
028F 301
56 04 AC DO 028F 302 MOVL 4(AP),R6 ; See whether we're dealing with...
0000'C6 00'8F 91 0293 303 CMPB #FAB$C_BID,FAB$B_BID(R6) ; ...a FAB or a RAB
57 FD8A CF DE 0299 304 BNEQ 10$ ; BR if it's a RAB
58 56 DO 029B 305 MOVAL FILE,R7 ; FAB-specific code: text string...
0000'C6 DD 02A0 306 MOVL R6,R8 ; ...address of FAB...
0000'C6 DD 02A3 307 PUSHL FAB$L_STV(R6) ; ...STV field for error...
FE46 CF 0000'C6 DD 02A7 308 PUSHL FAB$L_STS(R6) ; ...STS field for error...
19 11 02AB 309 MOVL FAB$L_STS(R6),STATUS ; ...and save the error code
02B2 310 BRB COMMON ; FAB and RAB share other code
02B4 311 10$:
57 FD7D CF DE 02B4 312 MOVAL RECORD,R7 ; RAB-specific code: text string...
58 0000'C6 DO 02B9 313 MOVL RAB$L_FAB(R6),R8 ; ...address of associated FAB...
0000'C6 DD 02BE 314 PUSHL RAB$L_STV(R6) ; ...STV field for error...
0000'C6 DD 02C2 315 PUSHL RAB$L_STS(R6) ; ...STS field for error...
FE2B CF 0000'C6 DO 02C6 316 MOVL RAB$L_STS(R6),STATUS ; ...and save the error code
02CD 317 COMMON:
5A 0000'C8 9A 02CD 318 MOVZBL FAB$B_FNS(R8),R10 ; Get the file name size
02D2 319 $FAO_S CTRSTR = RMS_ERR_STRING,- ; Common code, prepare error message...
02D2 320 OUTLEN = BUFFER_PTR,-
02D2 321 OUTBUF = FAO_BUF,-
02D2 322 P1 = R7,-
02D2 323 P2 = R10,-
02D2 324 P3 = FAB$L_FNA(R8)
FD7B CF DF 02ED 325 PUSHAL BUFFER_PTR ; ...and arguments for ERROR_EXIT...
00741130 01 DD 02F1 326 PUSHL #1 ; ...
8F DD 02F3 327 PUSHL #UETPS_TEXT ; ...

```

		00	EF	02F9	328
		03		02FB	329
59	FDF9	CF		02FC	330
	6E	59	88	0300	331
		05	DD	0303	332

```
EXTZV #STSSV_SEVERITY,-
        #STSSS_SEVERITY,-
        STATUS=R9
BISB2 R9,(SP)
PUSHL #5
```

```
; ...get the severity code...
; ...and add it into the signal name
; Current arg count
```

```

0305 334 .sbtll Error Exit
0305 335 :++
0305 336 : FUNCTIONAL DESCRIPTION:
0305 337 : This routine prints an error message and exits.
0305 338 :
0305 339 : CALLING SEQUENCE:
0305 340 : MOVx error status value,STATUS
0305 341 : PUSHx error specific information on the stack
0305 342 : PUSHL current argument count
0305 343 : BRW ERROR_EXIT
0305 344 :
0305 345 : INPUT PARAMETERS:
0305 346 : Arguments to LIB$SIGNAL, as above
0305 347 :
0305 348 : IMPLICIT INPUTS:
0305 349 : NONE
0305 350 :
0305 351 : OUTPUT PARAMETERS:
0305 352 : Message to SYS$OUTPUT and SYS$ERROR
0305 353 :
0305 354 : IMPLICIT OUTPUTS:
0305 355 : Program exit
0305 356 :
0305 357 : COMPLETION CODES:
0305 358 : NONE
0305 359 :
0305 360 : SIDE EFFECTS:
0305 361 : NONE
0305 362 :
0305 363 :--
0305 364 :
0305 365 ERROR_EXIT:
0305 366
0305 367 $SETAST_S ENBFLG = #0 ; ASTs can play havoc with messages
FDEC CF 08 8E C1 030E 368 ADDL3 (SP)+,#8,ARG_COUNT ; Get total # args, pop partial count
0305 369 INCL ERROR_COUNT ; Keep running error count
0305 370 PUSHL #0 ; Push the time parameter
0305 371 P SHAL PROCESS_NAME ; Push test name...
000F0002 8F DD 031E 372 PL 4L #^XF0002 ; ...arg count...
007410E2 8F DD 0324 373 PUSHL #UETPS$ ABENDD!STSSK_ERROR ; ...and signal name
0305 374 PUSHL ERROR_COUNT ; finish off arg list...
0305 375 PUSHAL PROCESS_NAME ; ...
0305 376 PUSHL #^X10002 ; ...
0305 377 PUSHL #UETPS$ ERBOXPROC!STSSK_ERROR ; ...for error box message
00000000'GF FDBE CF FB 033E 378 CALLS ARG_COUNT,G^LIB$SIGNAL ; Truly bitch
0305 379
0305 380 BISL #STSSM_INHIB_MSG,STATUS ; Don't print messages twice
0305 381 $EXIT_S STATUS ; Exit in error
0305 382 .end UETMEMY01

```

UETMEMY01
Symbol table

PAGING LOAD

F 9

16-SEP-1984 00:29:44 VAX/VMS Macro V04-00
5-SEP-1984 04:36:14 [UETPSY.SRC]UETMEMY01.MAR;1

Page 10
(7)

UE
V04

\$\$T1	=	00000000		
\$\$T2	=	00000006		
ARG_COUNT		00000100	R	01
BEGSEND_ADD		00000011	R	01
BUFFER		00000074	R	01
BUFFER_PTR		0000006C	R	01
CHFSL_SIGARGLST	=	00000004		
CHFSL_SIG_ARG1	=	00000008		
CHFSL_SIG_ARGS	=	00000000		
CHFSL_SIG_NAME	=	00000004		
COMMON		000002CD	R	01
ERROR_COUNT		00000104	R	01
ERROR_EXIT		00000305	R	01
FABSB_BID		*****	X	01
FABSB_FNS		*****	X	01
FABSC_BID		*****	X	01
FABSL_FNA		*****	X	01
FABSL_STS		*****	X	01
FABSL_STV		*****	X	01
FAO_BUF		00000064	R	01
FILE		00000029	R	01
GLIST		00000019	R	01
JPIS_WSEXTENT	=	00000416		
LIBSSIGNAL		*****	X	01
LOOP		00000188	R	01
MSG_BLOCK		000000FC	R	01
MTH\$RANDOM		*****	X	01
OUTER_LOOP		00000186	R	01
PROCESS_NAME		00000108	R	01
RABSL_FAB		*****	X	01
RABSL_STS		*****	X	01
RABSL_STV		*****	X	01
RECORD		00000035	R	01
RMS_ERROR		0000028D	R	01
RMS_ERR_STRING		00000043	R	01
RMS_K	=	00000001		
SEED		00000009	R	01
SET_TIMEOUT_FLAG		000001C9	R	01
SHRS_ABENDD	=	000010E0		
SHRS_BEGINN	=	00001038		
SHRS_ENEDDD	=	00001080		
SHRS_TEXT	=	00001130		
SS\$NORMAL		*****	X	01
SS\$SSFAIL		*****	X	01
SSERROR		000001D1	R	01
STATUS		0C 000FB	R	01
STSSK_ERROR	=	00000002		
STSSM_INHIB_MSG	=	10000000		
STSSS_FAC_NO	=	0000000C		
STSSS_SEVERITY	=	00000003		
STSSV_FAC_NO	=	00000010		
STSSV_SEVERITY	=	00000000		
SYSSDELIVA		*****	GX	01
SYSEXIT		*****	GX	01
SYSEXPRG		*****	GX	01
SYSSFAO		*****	X	01
SYSSGETJPI		*****	GX	01

SYSSGETMSG	*****	GX	01
SYSSPUTMSG	*****	GX	01
SYSSSETAST	*****	GX	01
SYSSSETIMR	*****	GX	01
SYSSSETSFM	*****	GX	01
TEXT_BUFFER	=	00000084	
TIME_FLAG		00000008	R 01
TWO_MIN		00000000	R 01
UETMEMY01		00000119	RG 01
UETP	=	00740000	
UETPS_ABENDD	=	007410E0	
UETPS_BEGINN	=	00741038	
UETPS_ENEDDD	=	00741080	
UETPS_ERBOXPROC	=	00748020	
UETPS_FACILITY	=	00000074	
UETPS_TEXT	=	00741130	
VAS_PAGCNT		0000000D	R 01

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS :	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK :	00000358 (859.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$ABSS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.06	00:00:00.53
Command processing	137	00:00:00.71	00:00:05.76
Pass 1	214	00:00:04.84	00:00:12.21
Symbol table sort	0	00:00:00.41	00:00:00.83
Pass 2	93	00:00:01.20	00:00:02.62
Symbol table output	13	00:00:00.08	00:00:00.23
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	497	00:00:07.34	00:00:22.21

The working set limit was 1350 pages.
24038 bytes (47 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 337 non-local and 5 local symbols.
382 source lines were read in Pass 1, producing 16 object records in Pass 2.
26 pages of virtual memory were used to define 24 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[SHRLIB]UETP.MLB;1	1
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	20
TOTALS (all libraries)	21

468 GETS were required to define 21 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:UETMEMY01/OBJ=OBJ\$:UETMEMY01 MSRCS:UETMEMY01/UPDATE=(ENHS:UETMEMY01)+EXECMLS/LIB+SHRLIB\$:UETP/LIB

