

The image is a large, symmetrical, abstract graphic composed of the letters 'S' and 'Y' arranged in a grid-like pattern. The letters are black on a white background. The overall shape is roughly rectangular, with a central vertical column of 'Y's and 'S's. The left and right sides are composed of 'S's, with some 'Y's interspersed. The top and bottom edges are also composed of 'S's. The central vertical column is the tallest, with 'Y's at the top and 'S's at the bottom. The overall effect is a complex, stylized letter 'M' or 'W'.

SY
VO[illegible]

(1) 120
(1) 172
(1) 272
(1) 663
(1) 699
(1) 789

DECLARATIONS
EX\$DELPRC - DELETE PROCESS SYSTEM SERVICE
DELETE - PERFORM DELETE ACTIONS IN CONTEXT OF PROCESS
DELPAGE - DELETE PAGE
TERMMBX - SEND MESSAGE TO TERMINATION MAILBOX
Return Unused CPU Time Limit to Parent

```
0000 1      .TITLE SYSDELPRC - DELETE PROCESS SYSTEM SERVICE
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, SYSTEM SERVICES
0000 30
0000 31 : ABSTRACT:  SYSDELPRC IMPLEMENTS THE DELETE PROCESS SYSTEM SERVICE
0000 32 :            WHICH CAUSES A PROCESS TO BE DELETED FROM THE SYSTEM AFTER
0000 33 :            RELEASING ALL OF ITS RESOURCES.
0000 34
0000 35 : ENVIRONMENT:
0000 36 :            MODE=KERNEL
0000 37
0000 38 : AUTHOR:    R. I. HUSTVEDT, CREATION DATE: 30-DEC-76
0000 39
0000 40 : MODIFIED BY:
0000 41
0000 42 :            V03-019 TMK0002      Todd M. Katz      18-Apr-1983
0000 43 :            Delete the job-wide logical name table and all names contained
0000 44 :            therein after dismounting all mounted volumes ( the table is
0000 45 :            actually deleted after decrementing the interactive and batch
0000 46 :            job counts as appropriate ). We were deleting the job table
0000 47 :            before dismounting all privately mounted volumes. This was
0000 48 :            causing random paged-pool corruption whenever a logical name had
0000 49 :            been specified during the mounting of the private volume. This
0000 50 :            logical name is now placed into the job table, instead of into
0000 51 :            the process logical name table. Therefore, it was being deleted
0000 52 :            twice - once as a result of the cleanup of the job table prior
0000 53 :            to its deletion, and once as a result of the dismount of the
0000 54 :            privately mounted disk.
0000 55
0000 56 :            V03-018 HH0011      Hai Huang      4-Apr-1984
0000 57 :            Lock I/O database when flushing the job-wide mount list.
```


0000	58	:	
0000	59	:	
0000	60	:	V03-017 HH0002 Hai Huang 1-Feb-1984
0000	61	:	Add job-wide mount support, i.e. do not issue ioc\$dismount
0000	62	:	if the process is a sub-process.
0000	63	:	
0000	64	:	V03-016 CDS0002 Christian D. Saether 14-Dec-1983
0000	65	:	Modify the bugcheck that looks for assigned channels
0000	66	:	to ignore the special xqp channel.
0000	67	:	
0000	68	:	V03-015 LJK0254 Lawrence J. Kenah 7-Dec-1983
0000	69	:	Add two consistency checks.
0000	70	:	
0000	71	:	V03-014 TMK0001 Todd M. Katz 12-Oct-1983
0000	72	:	If the process being deleted is not a sub-process, then
0000	73	:	delete the job-wide logical name table after the deletion
0000	74	:	of all subprocesses has completed.
0000	75	:	
0000	76	:	V03-013 LJK0248 Lawrence J. Kenah 30-Aug-1983
0000	77	:	Reorder EXE\$DELPRC procedure so that setting of DELPEN bit
0000	78	:	and queuing of DELETE AST occur while remaining at SYNCH.
0000	79	:	
0000	80	:	V03-012 WMC0002 Wayne Cardoza 15-JUL-1983
0000	81	:	Batch count.
0000	82	:	
0000	83	:	V03-011 WMC0011 Wayne Cardoza 15-Jun-1983
0000	84	:	Decrement interactive user count when appropriate.
0000	85	:	
0000	86	:	V03-010 RSH0032 R. Scott Hanna 26-May-1983
0000	87	:	Fix linker truncation error.
0000	88	:	
0000	89	:	V03-009 ACG0335 Andrew C. Goldstein, 10-May-1983 19:20
0000	90	:	Deallocate extended rights list when present
0000	91	:	
0000	92	:	V03-008 LJK0202 Lawrence J. Kenah 19-Apr-1983
0000	93	:	Fix several broken word displacements
0000	94	:	
0000	95	:	V03-007 CWH1002 CW Hobbs 24-Feb-1983
0000	96	:	Modified to support extended pids. Use SCH\$C_SWPPPIX to locate
0000	97	:	the swapper pid.
0000	98	:	
0000	99	:	V03-006 DMW4025 DMWalp 7-Jan-1983
0000	100	:	Fixed broken branch
0000	101	:	
0000	102	:	V03-005 CDS0001 C Saether 22-Oct-1982
0000	103	:	Add process deletion defer mechanism. Initial user F11BXQP.
0000	104	:	
0000	105	:	V03-004 ACG0301 Andrew C. Goldstein, 21-Oct-1982 13:41
0000	106	:	Restore FILCNTNONZ bugcheck
0000	107	:	
0000	108	:	V03-003 WMC0001 Wayne Cardoza 26-Aug-1982
0000	109	:	Reset privileged library vectors after rundown.
0000	110	:	
0000	111	:	V03-002 LJK0169 LAWRENCE J. KENAH 2-Jun-1982
0000	112	:	Correct incorrect register bug when returning unused CPU
0000	113	:	time. Make extremely rare error path perform in a way
0000	114	:	that makes sense.

SYSDELPRC
V04-000

- DELETE PROCESS SYSTEM SERVICE

D 9

16-SEP-1984 01:57:52 VAX/VMS Macro V04-00
5-SEP-1984 03:52:26 [SYS.SRC]SYSDELPRC.MAR;1

Page 3
(1)

0000 115 :
0000 116 :
0000 117 :
0000 118 :--

V03-001 ACG0178

Remove FILECNTNONZ bugcheck

Andrew C. Goldstein,

1-Apr-1982 11:21


```
0000 120      .SBTTL  DECLARATIONS
0000 121      :
0000 122      : INCLUDE FILES:
0000 123      :
0000 124      :
0000 125      $ACBDEF      ; DEFINE AST CONTROL BLOCK OFFSETS
0000 126      $ACCDEF      ; DEFINE TERMINATION MESSAGE OFFSETS
0000 127      $ACMDEF      ; DEFINE ACCOUNTING MESSAGE OFFSETS
0000 128      $ARBDEF      ; DEFINE ACCESS RIGHTS BLOCK
0000 129      $CCBDEF      ; DEFINE CHANNEL CONTROL BLOCK
0000 130      $IDDEF       ; DEFINE I/O FUNCTION CODES
0000 131      $IPLDEF      ; DEFINE INTERRUPT PRIORITY LEVELS
0000 132      $JIBDEF      ; DEFINE JOB INFORMATION BLOCK OFFSETS
0000 133      $MSGDEF      ; DEFINE SYSTEM MESSAGE TYPE CODES
0000 134      $PCBDEF      ; DEFINE PCB OFFSETS
0000 135      $PFNDEF      ; DEFINE PFN CONSTANTS
0000 136      $PHDDEF      ; DEFINE PROCESS HEADER OFFSETS
0000 137      $PRDEF       ; DEFINE PROCESSOR REGISTER NAMES
0000 138      $PRIDEF      ; DEFINE PRIORITY INCREMENTS
0000 139      $PTEDEF      ; DEFINE PTE FIELDS
0000 140      $PSLDEF      ; DEFINE PSL FIELDS AND VALUES
0000 141      $RSNDEF      ; DEFINE RESOURCE NUMBERS
0000 142      $SECDEF      ; DEFINE SECTION TABLE ENTRY
0000 143      $SSDEF       ; DEFINE SYSTEM SERVICE STATUS CODES
0000 144      $UCBDEF      ; DEFINE UNIT CONTROL BLOCK OFFSETS
0000 145      :
0000 146      :
0000 147      : MACROS:
0000 148      :
0000 149      :
0000 150      :
0000 151      : EQUATED SYMBOLS:
0000 152      :
0000 153      :
0000001C 0000 154 ACB_L_CPULIM = ACB$L_KAST + 4
00000020 0000 155 ACB_L_CPUTIM = ACB_L_CPULIM + 4
00000010 0000 156 JOB_STRING_LEN = 16
0000 157      :
0000 158      :
0000 159      : OWN STORAGE:
0000 160      :
0000 161      :
00000000 162      .PSECT  Y$EXEPAGED
0000 163      :
0000 164 LNM_SYSTEM_DIRECTORY_DESC: ; Descriptor of System Directory Table
59 53 24 4D 4E 4C 00000008'010E0000' 0000 165      .ASCID  /LNM$SYSTEM_DIRECTORY/
4F 54 43 45 52 49 44 5F 4D 45 54 53 000E
59 52 001A
001C 166
001C 167 JOB_FAO: ; $FAO control string descriptor for
4F 4A 24 4D 4E 4C 00000024'010E0000' 001C 168      .ASCID  /LNM$JOB_!XL/ ; construction of job-wide table name
4C 58 21 5F 42 002A
002F 169
00000000 170      .PSECT  AEXENONPAGED
```



```
0000 172 .SBTTL EXESDELPRC - DELETE PROCESS SYSTEM SERVICE
0000 173 :++
0000 174 : FUNCTIONAL DESCRIPTION:
0000 175 : EXESDELPRC IMPLEMENTS THE DELETE PROCESS SYSTEM SERVICE WHICH CAUSES
0000 176 : A PROCESS TO BE DELETED FROM THE SYSTEM. IF THE SPECIFIED
0000 177 : PROCESS EXISTS, A KERNEL AST IS QUEUED FOR IT TO
0000 178 : PERFORM THE DELETION PROCESSING IN THE CONTEXT OF THE TARGET
0000 179 : PROCESS. A PROCESS THEREFORE DELETES ITSELF.
0000 180 :
0000 181 : CALLING SEQUENCE:
0000 182 : CALLG  ARGLIST,EXESDELPRC
0000 183 :
0000 184 : INPUT PARAMETERS:
0000 185 : PIDADR(AP) - ADDRESS OF THE PROCESS IDENTIFICATION OF THE
0000 186 : PROCESS TO BE DELETED.
0000 187 : PRCNAM(AP) - ADDRESS OF STRING DESCRIPTOR FOR PROCESS LOGICAL NAME
0000 188 : STRING.
0000 189 : R4 - PCB ADDRESS OF CURRENT PROCESS
0000 190 :
0000 191 : IMPLICIT INPUTS:
0000 192 : CURRENT PROCESS PCB AND PHD
0000 193 : PCB OF TARGET PROCESS
0000 194 :
0000 195 : OUTPUT PARAMETERS:
0000 196 : R0 - COMPLETION STATUS CODE
0000 197 : @PIDADR(AP) - PROCESS IDENTIFICATION OF DELETED PROCESS
0000 198 :
0000 199 : IMPLICIT OUTPUTS:
0000 200 : NONE
0000 201 :
0000 202 : COMPLETION CODES:
0000 203 : SSS_NORMAL - SUCCESSFUL COMPLETION
0000 204 : SSS_NONEXPR - NONEXISTENT PROCESS
0000 205 : SSS_NOPRIV - NO PRIVILEGE, TARGET PROCESS IS NOT A SUBPROCESS
0000 206 : AND ISSUING PROCESS DOES NOT HAVE PROCESS CONTROL
0000 207 : PRIVILEGE FOR GROUP OR WORLD.
0000 208 : SSS_INSFMEM - INSUFFICIENT DYNAMIC MEMORY AVAILABLE FOR SERVICE
0000 209 : AND RESOURCE WAIT MODE DISABLED.
0000 210 :
0000 211 : SIDE EFFECTS:
0000 212 : A KERNEL AST WILL BE ENQUEUED FOR THE PROCESS TO BE
0000 213 : DELETED WHICH MAY CAUSE RESCHEDULING TO OCCUR.
0000 214 :
0000 215 :--
0000 216 :
0000 217 .ENTRY EXESDELPRC,^M<R2,R3,R4,R5,R6,R7> ; Save R2 through R7
0000 218 BSBW EXESALLOCIRP ; Allocate AST control block
0000 219 BLBC R0,10$ ; Return error if none
0000 220 MOVL R2,R5 ; Set address for SCH$QAST
0000 221 BSBW EXESNAMPID ; Convert name/PID to PCB address
0000 222 BLBC R0,30$ ; Exit if error
0000 223 CMPW R1,S^#SCH$C_SWPPIX ; Do not allow SWAPPER or NULL
0000 224 BLEQU 20$ ; Exit with error if either of those
0000 225 :
0000 226 : If the target process is already marked for delete, the system service
0000 227 : simply returns a success code to the caller after deallocating the ACB.
0000 228 : Note that SSS_NORMAL was loaded into R0 by the success path of EXESNAMPID.
```

00FC 0000 217
FFFB' 30 0002 218
34 50 E9 0005 219
55 52 D0 0008 220
FFF2' 30 000B 221
34 50 E9 000E 222
00' 51 B1 0011 223
2A 1B 0014 224
0016 225
0016 226
0016 227
0016 228


```
2A 24 A4 01 E2 0016 229
0016 230 BBSS #PCBSV_DELPEN,PCBSL_STS(R4),30$
001B 231
001B 232 ; The following three instructions are the guts of the $RESUME system service.
001B 233 ; That same functionality must be reproduced here while remaining at IPL
001B 234 ; SYNCH so the system service cannot be used.
001B 235
001B 236 ASSUME PCBSV_RESPEN LE 7
001B 237
24 A4 20 88 001B 238 BISB #<1@PCBSV_RESPEN>,PCBSL_STS(R4) ; Set resume pending bit
52 02 9A 001F 239 MOVZBL #PRI$_RESAVL,R2 ; Set priority increment class
0022 240 RPTEVT RESUME ; Report RESUME event to scheduler
0026 241
10 A5 0000002F'EF 9E 0026 242 MOVAB L^DELETE,ACBSL_AST(R5) ; Set address for delete action
0B A5 94 002E 243 CLRB ACBSB_RMOD(R5) ; Mark as normal kernel mode AST
0C A5 60 A4 D0 0031 244 MOVL PCBSL_PID(R4),ACBSL_PID(R5) ; Set PID of target
0036 245
0036 246 ; R2 has already been loaded with the correct priority increment class
0036 247 ; and R2 is preserved by SCH$RSE (called from within the RPTEVT macro)
0036 248 ; so that a
0036 249 :
0036 250 : MOVZBL #PRI$_RESAVL,R2 ; Set priority increment
0036 251 :
0036 252 ; is not necessary here.
0036 253
50 FFC7' 30 0036 254 BSBW SCH$QAST ; And queue ast for process
01 3C 0039 255 MOVZWL #SS$_NORMAL,R0 ; Set normal status
003C 256 10$: SETIPL #0 ; Enable rescheduling
003F 257 RET ; ... and return
0040 258
0040 259 ; The following error paths must deallocate the unused ACB before returning.
0040 260
50 08E8 8F 3C 0040 261 20$: MOVZWL #SS$_NONEXPR,R0 ; NULL and SWAPPER cannot be deleted
50 50 DD 0045 262 30$: PUSHL R0 ; Save error status
50 55 D0 0047 263 MOVL R5,R0 ; Get address of ACB
FFB3' 30 004A 264 BSBW EXESDEANONPAGED ; Deallocate packet back to pool
50 8ED0 004D 265 POPL R0 ; Retrieve saved error status
EA 11 0050 266 BRB 10$ ; Join common exit code
0052 267
0052 268 ; The rest of the code in this module can exist in a pageable part of the exec
0052 269
0000002F 270 .PSECT Y$EXEPAGED
```

```
002F 272 .SBTTL DELETE - PERFORM DELETE ACTIONS IN CONTEXT OF PROCESS
002F 273 :++
002F 274 : FUNCTIONAL DESCRIPTION: DELETE EXECUTES AS THE RESULT OF A KERNEL
002F 275 : AST INITIATED BY THE DELETE PROCESS SYSTEM SERVICE. IT PERFORMS
002F 276 : THE ACTIONS NECESSARY TO DELETE A PROCESS AND RETURN ITS RESOURCES.
002F 277 : TERMINATION MESSAGES ARE SENT TO THE TERMINATION MAILBOX UNIT
002F 278 : SPECIFIED WHEN THE PROCESS WAS CREATED AND TO THE SYMBIONT
002F 279 : MANAGER IF ACCOUNTING IS NOT DISABLED FOR THIS PROCESS.
002F 280 :
002F 281 : CALLING SEQUENCE:
002F 282 : (SAME EFFECT AS) DCLAST ASTADR=DELETE MODE=KERNEL
002F 283 :
002F 284 : INPUT PARAMETERS:
002F 285 : NONE
002F 286 :
002F 287 : OUTPUT PARAMETERS:
002F 288 : NONE
002F 289 :
002F 290 : IMPLICIT INPUTS:
002F 291 : PCB OF CURRENT PROCESS (LOCATED VIA SCH$GL_CURPCB)
002F 292 : PHD OF CURRENT PROCESS
002F 293 :
002F 294 : IMPLICIT OUTPUTS:
002F 295 : NONE
002F 296 :
002F 297 : SIDE EFFECTS:
002F 298 : NONE
002F 299 : --
002F 300 :
002F 301 : DELETE:
002F 302 : .WORD 0
002F 303 : MOVL G^SCH$GL_CURPCB, R4
002F 304 : BICB2 #1, PCB$B_ASTACT(R4)
002F 305 : JSB L^SCH$NEW[VL]
002F 306 4$: SETIPL 8$
002F 307 : TSTB PCB$B_DPC(R4)
002F 308 : BEQL 7$
002F 309 : PUSHL #0
002F 310 : MOVL #RSN$ASTWAIT, R0
002F 311 : JSB L^SCH$RWAIT
002F 312 : BRB 4$
002F 313 7$: SETIPL #0
002F 314 : BBCC #PCB$V_SSRWAIT, PCB$L_STS(R4), 10$ ; ENABLE RESOURCE WAIT
002F 315 : BRB 10$
002F 316 8$: .LONG IPL$SYNCH
002F 317 : ASSUME .-4$-LE 512 ; MAKE SURE CODE DOESN'T PAGE AT SYNCH
002F 318 10$: ; IF NOT TRUE, IPL PAGE FAULT HACK NG
002F 319 : -----
002F 320 :
002F 321 : INVOKE THE USER RUNDOWN SERVICE(S)
002F 322 :
002F 323 : The user service is invoked in KERNEL mode with a JSB and must
002F 324 : exit with a RSB.
002F 325 :
002F 326 : R4 - Pointer to current PCB
002F 327 : R7 - Access mode of call to rundown routine (always 0)
002F 328 : ACMODE(AP) - always 0
```

54 00000000'GF 0000 D0 0031 303
OC A4 01 8A 0038 304
00000000'EF 16 003C 305
2A A4 95 0042 306
OD 13 0049 307
00 00 004C 308
50 01 D0 004E 309
00000000'EF 16 0050 310
E7 11 0053 311
06 24 A4 0A E5 0059 312
04 11 005B 313
00000008 005E 314
0063 315
0065 316
0069 317
0069 318
0069 319
0069 320
0069 321
0069 322
0069 323
0069 324
0069 325
0069 326
0069 327
0069 328


```
0069 329 :
0069 330 : The above parameters are the same as in the call from SYSRUNDWN.
0069 331 :
0069 332 :-----
0069 333 :
57 D4 0069 334 CLRL R7 ; KERNEL MODE RUNDOWN
5C DD 006B 335 PUSHL AP
7E D4 006D 336 CLRL -(SP) ; ANOTHER KERNEL MODE ACCESS PARAMETER
01 DD 006F 337 PUSHL #1
5C 5E D0 0071 338 MOVL SP,AP
55 00000000'9F D0 0074 339 MOVL @#CTL$GL_USRUNDWN, R5 ; GET PER-PROCESS USER RUNDOWN VECTOR
02 13 007B 340 BEQL 15$ ; NOT PRESENT, SKIP ON
65 16 007D 341 JSB (R5) ; CALL THRU THE VECTOR(S)
55 00000000'GF D0 007F 342 15$: MOVL G^EXE$GL_USRUNDWN, R5 ; GET SYSTEM-WIDE USER RUNDOWN VECTOR
02 13 0086 343 BEQL 18$ ; NOT PRESENT, SKIP ON
65 16 0088 344 JSB (R5) ; CALL THRU THE VECTOR(S)
5E 08 C0 008A 345 18$: ADDL #8,SP ; CLEAN UP
5C 8E D0 008D 346 MOVL (SP)+,AP
00000000'EF 16 0090 347 JSB L^EXE$RESETVEC ; RESET THE PRIVILEGED LIBRARY VECTORS
0096 348 :
0096 349 :-----
0096 350 :
0096 351 : RMS RUNDOWN
0096 352 :
0096 353 :-----
0096 354 :
5E FF78 CE 9E 0096 355 MOVAB -136(SP),SP ; ALLOCATE STRING BUFFER
6E 80 8F 9A 009B 356 20$: MOVZBL #128,(SP) ; SET COUNT FOR STRING BUFFER
04 AE 08 AE DE 009F 357 MOVAL 8(SP),4(SP) ; FORM ADDRESS PART OF DESCRIPTOR
02 DD 00A4 358 PUSHL #2 ; LAST CHANCE RMS CLEANUP
04 AE 9F 00A6 359 PUSHAB 4(SP) ; SET ADDRESS OF STRING BUFFER
00000000'9F 02 FB 00A9 360 CALLS #2,@#SYS$RMSRUNDWN ; DEFAULT CLOSE OF OPEN RMS FILES
E8 50 E9 00B0 361 BLBC R0,20$ ; REPEAT IF MORE TO CLOSE
00B3 362 :
00B3 363 :-----
00B3 364 :
00B3 365 : FIND AND DELETE ALL SUB-PROCESSES
00B3 366 :
00B3 367 :-----
00B3 368 :
00B3 369 SUBDELETE: ; DELETE SUB-PROCESSES
42 A4 B5 00B3 370 TSTW PCB$W_PRCNT(R4) ; ANY SUB-PROCESSES?
42 13 00B6 371 BEQL 50$ ; NO, CONTINUE
56 00000000'EF 3C 00B8 372 MOVZWL SCH$GL_MAXPIX,R6 ; INITIALIZE INDEX FOR PROCESS SCAN
53 00000000'FF46 D0 00BF 373 10$: MOVL @L^SCH$GL_PCBVEC[R6],R3 ; GET A PCB ADDRESS
1C A3 60 A4 D1 00C7 374 CMPL PCB$L_PID(R4),PCB$L_OWNER(R3) ; DO WE OWN IT
0C 12 00CC 375 BNEQ 20$ ; NO, TRY ANOTHER
00CE 376 $DELPRC_S PCB$L_EPID(R3) ; MARK IT FOR DELETE USING THE EXTENDED PID
E2 56 F5 00DA 377 20$: SOBGTR R6,10$ ; CONTINUE
7E DC 00DD 378 30$: MOVPSL -(SP) ; SAVE PSL
00DF 379 DSBINT W^SUBWIPL,R6 ; BLOCK SYSTEM EVENTS, SAVE IPL IN R6
42 A4 B5 00E7 380 TSTW PCB$W_PRCNT(R4) ; CHECK COUNT OF SUB-PROCESSES STILL RUNNING
0B 13 00EA 381 BEQL 40$ ; DONE
50 01 3C 00EC 382 MOVZWL #RSN$ASTWAIT,R0 ; SET RESOURCE NUMBER
00000000'EF 16 00EF 383 JSB L^SCH$RWAIT ; AND WAIT FOR AN AST
E6 11 00F5 384 BRB 30$ ; CHECK AGAIN WHEN AWAKENED
00F7 385 40$: SETIPL R6 ; RESTORE IPL
```

```
04 11 00FA 386 50$: : DONE, ALL SUB-PROCESSES DELETED
00000008 00FA 387 : CONTINUE WITH RUNDOWN
00FC 388 SUBWIPL: :
00FC 389 : .LONG IPL$_SYNCH : MARKER FOR END OF TEMP NON-PAGED
0100 390 :
0100 391 :-----
0100 392 :
0100 393 : RUNDOWN PROCESS ACTIVITY
0100 394 :
0100 395 :-----
0100 396 :
0100 397 RUNDWN: : RUNDOWN PROCESS ACTIVITY
00 DD 0100 398 : SET MODE FOR RUNDOWN
00000000'EF 01 FB 0102 399 : RUNDOWN AT KERNEL MODE
55 00000000'9F 01 DO 0109 400 : GET PROCESS HEADER ADDRESS
52 55 20 A5 C1 0110 401 : GET ADDRESS OF SECTION TABLE
7E D4 0115 402 : ASSUME NO PAGES WILL BE DELETED
5E 10 C2 0117 403 : SAVE ROOM FOR INRANGE AND RETRANGE
53 5E DO 011A 404 : REMEMBER SCRATCH AREA ADDRESS
56 24 A5 32 011D 405 : GET INDEX TO LAST SEC TBL ENTRY USED
5E 11 0121 406 : JOIN COMMON LOOP
18 A246 D5 0123 407 4$: : ANY PAGES MAPPED?
55 19 0127 408 : BR IF NOTHING MAPPED TO THIS SECTION
0129 409 :
0129 410 : DO A $DELTVA FOR THIS SECTION.
0129 411 :
04 A3 08 A7 57 6246 DE 0129 412 : MOVAL (R2)[R6],R7 : ADDRESS OF SECTION TABLE ENTRY
04 A3 00CC C5 18 00 EF 012D 413 : EXTZV #SEC$V_VPX,#SEC$S_VPX,SEC$S_VPXPFC(R7),4(R3) : GET START VPN
50 00000000'EF 07 9C 013E 414 : CMPZV #PHD$V_POLR,#PHD$S_POLR,PHD$S_POLRASTL(R5),4(R3) : IS VA IN P0?
04 A3 04 A3 50 C2 0146 415 : BGTRU 5$ : BR IF IT IS IN P0 SPACE
04 A3 04 A3 16 00 EF 014A 416 : ROTL #7,SGN$GL_PTPAGCNT,R0 : GET LONGWD OFFSET FROM BEGIN OF PAGTBL
04 A3 04 A3 09 78 0151 417 : SUBL R0,4(R3) : GET VPN IN P1 SPACE
04 A3 04 A3 09 78 0157 418 5$: : EXTZV #SEC$V_VPX,#SEC$S_VPX,4(R3),4(R3) : MASK OFF SIGN BITS
63 1C A7 09 78 0157 419 : ASHL #9,4(R3),4(R3) : GET VA OF FIRST PAGE MAPPED
63 04 A3 C0 015C 420 : ASHL #9,SEC$S_PAGCNT(R7),(R3) : GET # OF BYTES IN SECTION
08 A3 FFFFFFFF 8F D1 0160 421 : ADDL2 4(R3),(R3) : GET VA OF LAST PAGE MAPPED
10 A3 01 DO 0170 422 : DECL (R3) : GET ADR OF LAST BYTE MAPPED
56 08 C0 017E 423 : $DELTVA_S (R3),8(R3) : DELETE PAGES FOR THIS SECTION
5E 10 C0 0183 424 : CMPL #-1,8(R3) : WERE ANY PAGES DELETED?
8E D5 0186 425 : BEQL 6$ : BR IF NONE DELETED
2B 13 0188 426 : MOVL #1,16(R3) : INDICATE SECOND RUNDWN REQUIRED
00 DD 018A 427 6$: : ADDL2 #SEC$C_LENGTH@-2,R6 : GET OFFSET TO NEXT SECTION TABLE ENTRY
01 FB 018C 428 8$: : BLSS 4$ : GO CHECK NEXT SECTION
0193 429 : ADDL #16,SP : CLEAN INRANGE AND RETRANGE OFF STACK
0193 430 : TSTL (SP)+ : WERE ANY PAGES DELETED?
0193 431 : BEQL 55$ : NO, THEN DON'T BOTHER WITH 2ND RUNDWN
0193 432 : PUSHL #0 : SET MODE FOR RUNDOWN
0193 433 : CALLS #1,SYSS$RUNDWN : RUNDOWN AT KERNEL MODE
0193 434 :
0193 435 :*****
0193 436 :
0193 437 : SCAN THE CHANNEL TABLE AND LOOK FOR ASSIGNED
0193 438 : CHANNELS. THERE SHOULD BE NONE.
0193 439 :
0193 440 :*****
55 00000000'9F 3C 0193 441 :
0193 442 10$: : MOVZWL @#CTL$GW_CHINDX,R5 : GET MAXIMUM INDEX + 16
```



```
00000000'9F 55 19 13 019A 443 BEQL 55$ : NO CHANNEL TABLE
55 55 CE 019C 444 MNEGL R5,R5 : CONVERT TO NEGATIVE OFFSET
09 C1 019F 445 ADDL3 #CCB$B_AMOD, -
53 01A6 446 @#CTL$GL-CCB$BASE,R3 : COMPUTE ADDRESS OF HIGHEST CHANNEL
6345 95 01A7 447 40$: TSTB (R3)[R5] : IS THIS CHANNEL STILL ASSIGNED?
04 15 01AA 448 BLEQ 50$ : BRANCH IF NOT.
01AC 449
01AC 450 :
01AC 451 : A NEGATIVE ACCESS MODE IN THE CCB IS USED TO RESERVE THE CHANNEL
01AC 452 : FOR USE BY THE F11BXQP TO PERFORM LOGICAL I/O ON BEHALF OF VIRTUAL
01AC 453 : I/O FUNCTIONS THAT IT PROCESSES (THE XQP MODIFIES THE CCB$B_UCB AND
01AC 454 : CCB$B_AMOD FIELDS PRIOR TO ACTUALLY USING IT). THE CHANNEL DOES NOT
01AC 455 : HAVE TO BE DEASSIGNED BECAUSE NO DEVICE REFERENCE COUNTS REFLECT ITS
01AC 456 : EXISTENCE.
01AC 457 :
01AC 458
01AC 459 BUG_CHECK FILCNTNONZ,FATAL : IT REALLY SHOULD HAVE WORKED
01B0 460
55 10 C0 01B0 461 50$: ADDL2 #CCB$C_LENGTH,R5 : NEXT CHANNEL
F2 19 01B3 462 BLSS 40$ : LOOP THRU
01B5 463 55$:
01B5 464
01B5 465 :*****
01B5 466 : END OF BUG TRAP
01B5 467 :*****
01B5 468
01B5 469 DISMOUNT ALL MOUNTED VOLUMES
01B5 470 :
1C A4 D5 01B5 471 TSTL PCB$B_OWNER(R4) : IS THIS A SUB-PROCESS ?
2A 12 01B8 472 BNEQ 20$ : IF NEQ YES, SKIP IOC$DISMOUNT
56 0080 C4 D0 01BA 473 MOVL PCB$B_JIB(R4),R6 : GET JIB ADDRESS
00000000'EF 16 01BF 474 JSB L^SCH$IOLOCKW : LOCK I/O DATABASE
56 00 B6 0F 01C5 475 REMQUE @JIB$B_MTLFL(R6),R6 : GET A VOLUME TO BE DISMOUNTED
10 1D 01C9 476 BVS 15$ : NONE, FINISHED WITH DISMOUNT LIST
00000000'EF 16 01CB 477 JSB L^SCH$IOUNLOCK : UNLOCK I/O DATABASE
01D1 478 SETIPL #0 : DROP IPL
53 D4 01D4 479 CLRL R3 : UNLOAD UPON DISMOUNT
FE27' 30 01D6 480 BSBW IOC$DISMOUNT : DISMOUNT IT
B8 11 01D9 481 BRB 10$ : AND TRY FOR ANOTHER
01DB 482 15$: REF LABEL
00000000'EF 16 01DB 483 JSB L^SCH$IOUNLOCK : UNLOCK I/O DATABASE
01E1 484 SETIPL #0 : DROP IPL
01E4 485 20$:
01E4 486 $DALLOC,S ACMODE=#0 : DEALLOCATE ALL
40 A4 3E A4 B1 01EF 487 CMPW -PCB$B_DIOCNT(R4),PCB$B_DIOLM(R4) : WAIT FOR DIRECT I/O COMPLETION
EE 12 01F4 488 BNEQ 20$
3C A4 3A A4 B1 01F6 489 30$: CMPW PCB$B_BIOCNT(R4),PCB$B_BIOLM(R4) : AND BUFFERED I/O
F9 12 01FB 490 BNEQ 30$
01FD 491
01FD 492 :-----
01FD 493 : DECREMENT INTERACTIVE AND BATCH JOB COUNT WHEN APPROPRIATE
01FD 494 :
01FD 495 :-----
01FD 496
1C A4 D5 01FD 497 TSTL PCB$B_OWNER(R4)
48 12 0200 498 BNEQ RELQUOTA : WE DON'T DO IT FOR A SUBPROCESS
499
```

```
08 24 A4 19 E1 0202 500 BBC #PCBSV_INTER,PCBSL_STS(R4),59$ : NOT INTERACTIVE
00000000'EF B7 0207 501 DECW SY$SGW_IJOBcnt ; ONE LESS INTERACTIVE JOB
OB 11 020D 502 DELETE_JT
OE E1 020F 503 59$: BBC #PCBSV_BATCH,- ; NOT BATCH JOB
06 24 A4 0211 504 PCBSL_STS(R4),DELETE_JT
00000000'EF B7 0214 505 DECW SY$SGW_BJOBcnt ; ONE LESS BATCH JOB
021A 506
021A 507
021A 508
021A 509
021A 510
021A 511
021A 512
021A 513
021A 514
5E FO AE 9E 021A 515 DELETE_JT: ; DELETE THE JOB-WIDE LOGICAL NAME TABLE
6E 9F 021E 516 MOVAB ~JOB_STRING_LEN(SP),SP ; MAKE ROOM ON STACK FOR TABLE NAME
10 DD 0220 517 PUSHAB (SP)- ; CONSTRUCT A DESCRIPTOR OF THE BUFFER
53 5E DO 0222 518 PUSHL #JOB_STRING_LEN ; WHICH IS TO CONTAIN THE TABLE NAME
0225 519 MOVL SP,R3 ; SAVE THE ADDRESS OF THE DESCRIPTOR
0225 520 $FAO_S - ; CONSTRUCT THE JOB-WIDE TABLE NAME
0225 521 CTRSTR = JOB_FAO, -
0225 522 OUTBUF = (R3), -
0225 523 P1 = PCBSL_JIB(R4)
0238 524
0238 525 $DELLNM_S - ; DELETE THE JOB-WIDE LOGICAL NAME TABLE
0238 526 LOGNAM = (R3), -
0238 527 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
5E 18 CO 0247 528 ADDL2 #JOB_STRING_LEN+8,SP- ; RESTORE THE STACK
024A 529
024A 530
024A 531
024A 532
024A 533
024A 534
024A 535
024A 536 RELQUOTA: ; RELEASE QUOTAS
56 70 A4 94 024A 536 CLRBL PCB$T_LNAME(R4) ; REMOVE PROCESS NAME
1C A4 DO 024D 537 MOVL PCBSL_OWNER(R4),R6 ; GET PID OF OWNER
4E 13 0251 538 BEQL MESSAGES ; IF EQL, NOT A SUBPROCESS
54 DD 0253 539 PUSHL R4 ; SAVE PCB ADDRESS
56 56 3C 0255 540 MOVZWL R6,R6 ; EXTRACT PIX FROM OWNER PID
54 00000000'FF46 DO 0258 541 MOVL @L^SCH$GL_PCBVEC[R6],R4 ; AND GET PCB
00000000'EF 16 0260 542 JSB L^EXE$ALLOCIRP ; ALLOCATE AST CONTROL BLOCK
33 50 E9 0266 543 BLBC R0,20$ ; SKIP QUOTA RETURN IF ERROR
55 52 DO 0269 544 MOVL R2,R5 ; SET ADDRESS OF QUOTA AST BLOCK
18 A5 04F3'CF 9E 026C 545 MOVAB W^RETQUOTA,ACBSL_KAST(R5) ; SET AST ADDRESS FOR SPECIAL
0272 546 ; KERNEL AST TO RETURN QUOTAS
OB A5 80 8F 90 0272 547 MOVBL #<1@ACBSV_KAST>,ACBSB_RMOD(R5) ; FLAG AS SPECIAL KERNEL AST
OC A5 60 A4 DO 0277 548 MOVL PCBSL_PID(R4),ACBSL_PID(R5) ; SET TARGET PID FOR AST
52 6E DO 027C 549 MOVL (SP),R2 ; GET PCB ADDRESS FOR PROCESS BEING DELETED
51 00000000'9F DO 027F 550 MOVL @#CTL$GL_PHD,R1 ; GET PROCESS HEADER ADDRESS
1C A5 5C A1 0110 C1 C3 0286 551 SUBL3 PHD$S_EXTRACPU(R1),PHD$S_CPULIM(R1),ACB_L_CPULIM(R5) ; SAVE CPULIMIT-LESS BONUS
028E 552
20 A5 38 A1 DO 028E 553 MOVL PHD$S_CPUTIM(R1),ACB_L_CPUTIM(R5) ; AND ACCUMULATED CPU TIME
0293 554
52 02 3C 0293 555 MOVZWL #PRIS_RES AVL,R2 ; SET PRIORITY INCREMENT CLASS
00000000'EF 16 0296 556 JSB L^SCH$QAST ; QUEUE AST FOR CREATOR
```



```
10 BA 029C 557 20$: POPR #^M<R4> ; RESTORE PCB ADDRESS
    029E 558 SETIPL #0 ; DROP IPL TO ISSUE SYSTEM SERVICE REQUESTS
    02A1 559
    02A1 560 -----
    02A1 561 :
    02A1 562 : TERMINATION MESSAGES
    02A1 563 :
    02A1 564 :
    02A1 565 :
    32 A4 B5 02A1 566 MESSAGES: ; SEND TERMINATION MESSAGES
    03 13 02A4 567 TSTW PCB$W_TMBU(R4) ; TERMINATION MAILBOX SPECIFIED ?
    015A 30 02A6 568 BEQL 10$ ; IF EQL, NO
    05 24 A4 0E E1 02A9 569 BSBW TERMMSG ; SEND MESSAGE TO TERMINATION MAILBOX
    00 24 A4 0A E5 02AE 570 10$: BBC #PCB$V_BATCH,PCB$L_STS(R4),20$ ; SEND MESSAGE TO JOB CONTROLLER
    55 D4 02B3 571 BBCC #PCB$V_SSRWAIT,PCB$L_STS(R4),20$ ; IF BC, NOT A BATCH JOB
    00000000'EF 16 02B5 572 20$: CLRL R5 ; ENABLE RESOURCE WAIT MODE
    02BB 573 JSB EXE$PRCDELMSG ; NOT A SPECIAL KERNEL AST ROUTINE
    02BB 574 ; SEND PROCESS DELETE MESSAGE
    02BB 575 -----
    02BB 576 :
    02BB 577 : DESTROY PROCESS
    02BB 578 :
    02BB 579 :
    02BB 580 :
    02BB 581 :
    02BB 582 .ENABLE LSB
    02BB 583
    02BB 584 DESTROY:
    5E 10 C2 02BE 585 SETIPL #IPL$ ASTDEL ; NO MORE ASTS AFTER THIS POINT
    52 5E D0 02C1 586 SUBL #16,SP ; 4 LONGWORDS OF STORAGE
    62 40000000 8F D0 02C4 587 MOVL SP,R2 ; SAVE BASE
    04 A2 7FFFFFFF 8F D0 02CB 588 MOVL #<1030>,(R2) ; SET STARTVA
    00 DD 02D3 589 MOVL #<<1031>-1>,4(R2) ; END AT VECTOR PAGE
    08 A2 7F 02D5 590 ; WE CAN'T USE THE MACRO SINCE THE P1 VECTORS WILL DISAPPEAR
    62 7F 02D8 591 $DELTVA_S (R2),8(R2) ; DELETE FROM KERNEL STACK TO VECTORS
    80000000'EF 03 FB 02DA 592 PUSHL #0
    02E1 593 PUSHAQ 8(R2)
    02E1 594 PUSHAQ (R2)
    02E6 595 CALLS #3,SYS$DELTVA-P1SYSVECTORS+^X80000000 ; DELETE KSTK TO VECTORS
    07 02E6 596 SETIPL W^DESTROY_SYNCH ; ENTER & EXIT DELTVA AT IPL(ASDEL)
    02E7 597 SVPCTX ; BLOCK SYSTEM EVENTS
    600 02E7 598 ; AND LOCK PAGABLE CODE
    55 6C A4 D0 02EB 601 MOVZWL PCB$L_PHD(R4),R5 ; SAVE PROCESS CONTEXT
    5A 42 A5 3C 02EF 602 MOVZWL PHD$W-PHVINDEX(R5),R10 ; AND GO TO INTERRUPT STACK
    53 00000000'EF 9E 02F6 603 MOVAB L^SCH$GL NULLPCB,R3 ; GET PROCESS HEADER ADDRESS
    00000000'EF 53 D0 02FD 604 MOVL R3,L^SCH$GL_CURPCB ; GET BALANCE SLOT INDEX
    51 00000000'FF4A 3C 0305 605 MOVZWL @L^PHV$GL_PTXBAS[R10],R1 ; GET POINT TO NULL PROCESS PCB
    00000000'FF41 53 D0 030D 606 MOVL R3,@L^SCH$GL_PCBVEC[R1] ; SET AS CURRENT PCB
    59 0C A5 3C 0311 607 MOVZWL PHD$W_WSLOCK(R5),R9 ; GET PROCESS INDEX
    58 08 A5 3C 0314 608 CLRL (R5)[R9] ; RELEASE PROCESS INDEX SLOT
    58 6548 DE 0318 609 MOVZWL PHD$W_WSLIST(R5),R8 ; POINT TO FIRST NON PERM PAGE
    52 88 D0 031C 610 MOVAL (R5)[R8],R8 ; MARK WITH A ZERO
    0A 13 031F 611 40$: MOVL (R8)+,R2 ; LONGWORD INDEX TO START OF WORKING SET LIS
    F8 52 E9 0321 612 BEQL 60$ ; FORM BYTE ADDRESS OF START OF WSLIST
    613 BLBC R2,40$ ; GET VA OF PERM PAGE
    ; DONE, NONE LEFT
    ; IGNORE IF NOT VALID
```



```

      F6 19 0324 614      BLSS 40$      : OR IF HEADER PAGES (SYSTEM ADDRESS)
009A 30 0326 615      BSBW DELPAGE      : DELETE PAGE
      F1 11 0329 616      BRB 40$      : CONTINUE
      032B 617
50 10 B4 0F 032B 618 60$: REMQUE @PCBSL_ASTQFL(R4),R0 : REMOVE PENDING AST CONTROL BLOCKS
      OF 1D 032F 619      BVS 70$      : NONE LEFT
      05 E0 0331 620      BBS #ACBSV NODELETE,- : BR. IF NODELETE BIT IS SET. THIS
      F5 0B A0 0333 621      ACBSB_RMOD(R0),60$ : SHOULD NOT HAPPEN BUT IF IT DOES MAY
      0336 622      : RESULT IN LOSING POOL (WHICH IS
      0336 623      : PREFERABLE OVER A DOUBLE DEALLOCATE).
      02 10 0336 624      BSBB DEANONPAGED      : DEALLOCATE SPACE FOR ACB
      F1 11 0338 625      BRB 60$      : AND CONTINUE
      033A 626
      00000000'EF 17 033A 627 DEANONPAGED:      :
      0340 629      JMP L^EXES$DEANONPAGED      : JUMP TO REACH REAL DEANONPAGED
50 20 A4 D0 0340 630 70$: MOVL PCBSL_WSSWP(R4),R0 : GET VBN OF AREA TO FREE IN SWAPFILE
      1C 13 0344 631      BEQL 80$      : NONE, CONTINUE
51 52 A5 3C 0346 632      MOVZWL PHD$W_SWAPSIZE(R5),R1 : SIZE OF AREA TO FREE
53 50 08 18 EF 034A 633      EXTZV #24,#8,R0,R3 : SET SWAP FILE NUMBER
50 50 18 00 EF 034F 634      EXTZV #0,#24,R0,R0 : GET VBN OF AREA
53 00000000'FF43 D0 0354 635      MOVL @L^MMG$GL_PAGSWPVC[R3],R3 : GET ADDRESS OF PFL CONTROL BLOCK
      00000000'EF 16 035C 636      JSB L^MMG$DEALCLOCPAGFIL : DEALLOCATE AREA
50 00AC C4 D0 0362 637 80$: MOVL PCBSQ_PRIV+ARB$SL_RIGHTSLIST+8(R4),R0 : GET EXTENDED RIGHTS LIST
      02 13 0367 638      BEQL 85$      : BRANCH IF NONE
      CF 10 0369 639      BSBB DEANONPAGED      : DEALLOCATE IT
50 0080 C4 D0 036B 640 85$: MOVL PCBSL_JIB(R4),R0 : GET ADDRESS OF JIB FOR RELEASE
      0C 13 0370 641      BEQL 90$      : BR IF NONE
      44 A0 B7 0372 642      DECW JIB$W_PRCNT(R0) : ONE LESS PROCESS IN THIS TREE
54 A0 60 A4 D1 0375 643      CMPL PCBSL_PID(R4),JIB$SL_MPID(R0) : IS THIS THE MASTER PROCESS?
      02 12 037A 644      BNEQ 90$      : BR IF NOT
      BC 10 037C 645      BSBB DEANONPAGED      : DEALLOCATE JIB
50 1C A4 3C 037E 646 90$: MOVZWL PCBSL_OWNER(R4),R0 : GET PIX FROM OWNER UIC
      14 13 0382 647      BEQL 100$     : BR IF NOT A SUB-PROCESS
50 00000000'FF40 D0 0384 648      MOVL @L^SCH$GL_PCBVECC[R0],R0 : GET ADDRESS OF OWNER PCB
      42 A0 B7 038C 649      DECW PCBSW_PRCNT(R0) : DECREMENT COUNT OF SUB-PROCESSES
50 01 D0 038F 650      MOVL #RSNS$ASTWAIT,R0 : NOTIFY PARENT PROCESS THAT ITS
      00000000'EF 16 0392 651      JSB L^SCH$RAVAIL : SUBPROCESS COUNT HAS BEEN DECREMENTED
50 54 D0 0398 652 100$: MOVL R4,R0 : SET ADDRESS OF PCB FOR RELEASE
      9D 10 039B 653      BSBB DEANONPAGED      : AND DELETE IT
      00000000'EF B7 039D 654      DECW L^SWP$GW_BALCNT : DECREASE COUNT IN BALANCE SET
      00000000'EF 16 03A3 655      JSB L^SCH$SWPWAKE : WAKE SWAPPER
00000000'FF4A 00 B2 03A9 656      MCOMW #0,@L^PHV$GL_PIXBAS[R10] : INDICATE PROCESS HEADER TO BE DELETED
      00000000'EF B6 03B1 657      INCW L^SCH$GW_DELPHDCT : FLAG SWAPPER'S ATTENTION
      00000000'EF B7 03B7 658      DECW L^SCH$GW_PROCCNT : ONE LESS PROCESS NOW
      00000000'EF 17 03BD 659      JMP L^SCH$SCRED : AND SCHEDULE FOR ANOTHER PROCESS
      03C3 660
      03C3 661      .DISABLE LSB
```



```
03C3 663 .SBTTL DELPAGE - DELETE PAGE
03C3 664 :++
03C3 665 : FUNCTIONAL DESCRIPTION:
03C3 666 : DELPAGE DELETES THE CONTENTS OF THE SPECIFIED PAGE AND PLACES
03C3 667 : THE PFN ON THE FREE PAGE LIST.
03C3 668 :
03C3 669 : CALLING SEQUENCE:
03C3 670 : BSB DELPAGE
03C3 671 :
03C3 672 : INPUT PARAMETERS:
03C3 673 : R2 - VIRTUAL ADDRESS OF PAGE TO RELEASE
03C3 674 : R4 - PCB ADDRESS OF PROCESS OWNING PAGE
03C3 675 : R5 - PROCESS HEADER ADDRESS OF PROCESS OWNING PAGE
03C3 676 :
03C3 677 : OUTPUT PARAMETERS:
03C3 678 : NONE
03C3 679 :
03C3 680 :--
03C3 681 :
03C3 682 DELPAGE:
03C3 683 JSB MMG$SVAPTECHK ; DELETE PAGE
03C9 684 BBCC #PTESV_VALID,(R3),50$ ; GET SVA OF PTE FOR PAGE
03CD 685 BBCC #PTESV_MODIFY,(R3),10$ ; EXIT IF NOT VALID
03D1 686 10$: EXTZV #0,#PTES PFN,(R3),R0 ; CLEAR MODIFY INDICATOR
03D6 687 DECF L^,,JSB ; GET PFN FROM PTE
03E5 688 BGTR 50$ ; DROP REFERENCE COUNT ON PAGE
03E7 689 BICB #PFNSM_MODIFY,@L^PFNSAB_ ; BR IF CANT RELEASE YET
03F0 690 20$: JSB MMG$DE[CONPFN STATE[R0] ; MAKE SURE NO MODIFY BIT
03F6 691 ASSUME PFNSC_FREPAGLST EQ 0 ; DELETE CONTENT OF PFN
03F6 692 CLRL R2 ; INDICATE FREE LIST
03F8 693 JSB MMG$INSPFNH ; INSERT AT HEAD OF FREE LIST
03FE 694 50$: RSB ;
03FF 695 ;
03FF 696 DESTROY_SYNCH: ; END OF NONPAGABLE AREA
03FF 697 .LONG IPL$_SYNCH ; SYNCHRONIZATION IPL TO BLOCK SYSTEM EVENTS
```

00000000'EF 16
31 63 1F E5
00 63 1A E5
50 63 15 00 EF
17 14
00000000'FF40 80 8F 8A
00000000'EF 16
52 D4
00000000'EF 16
05
00000008 03FF


```
0403 699 .SBTTL TERMMBX - SEND MESSAGE TO TERMINATION MAILBOX
0403 700
0403 701 -----
0403 702
0403 703 BUILD PROCESS TERMINATION MESSAGE AND SEND TO TERMINATION
0403 704 MAILBOX UNIT.
0403 705 -----
0403 706
0403 707
0403 708 TERMMSG:
0403 709
55 5A 5E DO 0403 709 MOVL SP,R10 ; SEND TERMINATION MESSAGES
7E 68 A4 DD 0406 710 PUSHL PCB$E_OWNER(R4) ; SAVE STACK STATE
00000000'9F DO 0409 711 MOVL @#CTL$GL_PHD,R5 ; SEND EXTENDED PID OF SUBPROCESS OWNER (0 =
00000000'9F 7D 0410 712 MOVQ @#CTL$GL_LOGIN,-(SP) ; GET PROCESS HEADER ADDRESS
00000000'9F DD 0417 713 ASSUME ACC$E_OWNER EQ ACC$Q_LOGIN+8 ; LOGIN TIME
54 A5 DD 0417 714 PUSHL @#CTL$GL_VOLUMES ; COUNT OF MOUNTED VOLUMES
58 A5 DD 041D 715 ASSUME ACC$Q_LOGIN EQ ACC$E_VOLUMES+4
00000000'9F DD 041D 716 PUSHL PHD$E_DIOCNT(R5) ; DIRECT I/O COUNT
00 00 DD 0420 717 ASSUME ACC$E_VOLUMES EQ ACC$E_DIOCNT+4
00 00 DD 0420 718 PUSHL PHD$E_BIOCNT(R5) ; BUFFERED I/O COUNT
00 00 DD 0423 719 ASSUME ACC$E_DIOCNT EQ ACC$E_BIOCNT+4
4C A5 DD 0423 720 PUSHL @#CTL$GL_WSPEAK ; PEAK WORKING SET SIZE
38 A5 DD 0429 721 ASSUME ACC$E_BIOCNT EQ ACC$E_WSPEAK+4
00 00 DD 0429 722 PUSHL #0 ; PEAK PAGING FILE USAGE
4C A5 DD 042B 723 ASSUME ACC$E_WSPEAK EQ ACC$E_PGFLPEAK+4
38 A5 DD 042B 724 PUSHL PHD$E_PAGEFLTS(R5) ; TOTAL PROCESS PAGE FAULTS
00 00 DD 042E 725 ASSUME ACC$E_PGFLPEAK EQ ACC$E_PAGEFLTS+4
50 0000000C'9F 9E 0431 726 PUSHL PHD$E_CPUTIM(R5) ; SAVE ACCUMULATED CPU TIME FOR PROCESS
7E 70 7D 0431 727 ASSUME ACC$E_PAGEFLTS EQ ACC$E_CPUTIM+4
50 0000'CO 9E 0431 728 MOVAB @#CTL$T_USERNAME+12,R0 ; GET BASE ADDRESS
7E 60 7D 0438 729 MOVQ -(R0),-(SP) ; LAST HALF OF USER NAME
50 0000'CO 9E 043B 730 PUSHL -(R0) ; FIRST LONGWORD OF USER NAME
7E 60 7D 043D 731 MOVAB W^<CTL$T_ACCOUNT-CTL$T_USERNAME>(R0),R0 ; POINT PAST ACCOUNT
7E 60 7D 0442 732 ASSUME ACC$E_CPUTIM EQ ACC$E_USERNAME+12
7E 60 7D 0442 733 MOVQ (R0),-(SP) ; SET ACCOUNT INTO RECORD
7E 00000000'EF 7D 0445 734 ASSUME ACC$T_USERNAME EQ ACC$T_ACCOUNT+8
00 00 DD 0445 735 MOVQ EXES$Q_SYSTIME,-(SP) ; LOGOUT TIME
00 00 DD 044C 736 ASSUME ACC$T_ACCOUNT EQ ACC$Q_TERMTIME+8
64 A4 DD 044C 737 PUSHL #0 ; JOBID
00000000'9F DD 044E 738 ASSUME ACC$Q_TERMTIME EQ ACC$E_JOBID+4
03 03 DD 044E 739 PUSHL PCB$E_EPID(R4) ; EXTENDED PROCESS ID
00000000'9F DD 0451 740 ASSUME ACC$E_JOBID EQ ACC$E_PID+4
03 03 DD 0451 741 PUSHL @#CTL$GL_FINALSTS ; FINAL STATUS OF PROCESS
03 03 DD 0457 742 ASSUME ACC$E_PID EQ ACC$E_FINALSTS+4
03 03 DD 0457 743 PUSHL #MSG$_DELPROC ; MESSAGE TYPE
03 03 DD 0459 744 ASSUME ACC$E_FINALSTS EQ ACC$W_MSGTYP+4
57 5E DO 0459 745 MOVL SP,R7 ; MESSAGE ADDRESS
7E 3A303030 3030424D 8F 7D 045C 746 MOVZWL PCB$W_TMBU(R4),R0 ; TERMINATION MAILBOX UNIT
53 07 AE 9E 0460 747 MOVQ #^A/MB00000:/,-(SP) ; SKELETON FOR DEVICE STRING
52 50 50 0A 7B 046B 748 CLRL R1 ; INIT FOR EDIV
52 50 50 0A 7B 046D 749 MOVAB 7(SP),R3 ; POINTER TO LSD OF DEVICE NAME
52 50 50 0A 7B 0471 750 EDIV #10,R0,R0,R2 ; CONVERT A DECIMAL DIGIT
52 50 50 0A 7B 0476 751 ADDB R2,-(R3) ; SET DIGIT INTO NAME STRING IN ASCII
50 05 0479 752 TSTL R0 ; CHECK FOR END
5E 12 047B 753 BNEQ 10$ ; NOT YET, CONVERT ANOTHER DIGIT
08 DD 047D 754 PUSHL SP ; BUILD STRING DESCRIPTOR FOR
08 DD 047F 755 PUSHL #8 ; MAILBOX NAME
```



```
58 5E DO 0481 756      MOVL SP,R8      ; AND SAVE POINTER TO NAME DESCRIPTOR
59 7E DE 0484 757      MOVAL -(SP),R9    ; RESERVE SPACE FOR CHANNEL
      0487 758      :
      0487 759      : CHECK FOR PRESENCE OF CHANNEL TABLE, IF NON-EXISTENT (I.E. PROCESS DELETED
      0487 760      : BEFORE THE CREATION OF IT) USE A PAGE OF SUPERVISOR STACK FOR IT.
      0487 761      :
00000000'9F B5 0487 762      TSTW @#CTL$GW_NMIOCH    ; IS THERE A VALID CHANNEL TABLE?
      25 12 048D 763      BNEQ 20$              ; YES, SKIP TO ASSIGN
      54 DD 048F 764      PUSHL R4              ; SAVE PCB ADDRESS
53 00000008'9F DO 0491 765      MOVL @#CTL$AL STACK+<4*PSL$C_SUPER>,R3    ; PICK UP SUPER SP
      FO A3 9E 0498 766      MOVAB -CCB$C_LENGTH(R3),-
00000000'9F 049B 767      @#CTL$GL CCBBASE      ; MAKE IT TABLE BASE
0200 8F 00 6E 00 2C 04A0 768      MOVCS #0,(SP),#0,#512,-512(R3); CLEAR IT OUT
      FE00 C3 04A7 769      POPL R4              ; RESTORE PCB ADDRESS
      54 8ED0 04AA 770      MOVZWL #<512/CCB$C_LENGTH>-1,- ; SET NUMBER OF CHANNELS
      1F 3C 04AD 771      @#CTL$GW_NMIOCH
00000000'9F 04AF 772      :
      04B4 773      : ASSIGN CHANNEL TO MAILBOX AND WRITE THE MESSAGE
      04B4 774      :
      04B4 775      20$:
      04B4 776      $ASSIGN_S DEVNAM=(R8),-      ; ATTEMPT TO ASSIGN CHANNEL TO
      04B4 777      CHAN=(R9)      ; MAILBOX
21 50 E9 04C1 778      BLBC R0,30$      ; UNABLE TO ASSIGN, IGNORE
      04C4 779      $QIO_S CHAN=(R9),-      ; SEND TERMINATION MESSAGE
      04C4 780      FUNC=#<IO$_WRITEVBLK!IO$_M_NOW>,-
      04C4 781      P1=(R7),-      ; MESSAGE ADDRESS
      04C4 782      P2=#ACC$C_TERMLEN; MESSAGE SIZE
      04E5 783      30$:
      04E5 784      $DASSGN_S CHAN=(R9)      ; DEASSIGN CHANNEL
5E 5A DO 04EF 785      MOVL -R10,SP      ; RESTORE STACK STATE
      05 04F2 786      RSB
      04F3 787
```

```
04F3 789      .SUBTITLE      Return Unused CPU Time Limit to Parent
04F3 790
04F3 791      ;+
04F3 792      : Functional Description:
04F3 793
04F3 794      RETQUOTA is entered via a special kernel AST directed to the
04F3 795      creator of a subprocess. The AST control block contains the
04F3 796      unused CPU time limit to be returned to the creator.
04F3 797
04F3 798      The CPU limit of the parent has any unused CPU time from the
04F3 799      subprocess added to it.
04F3 800
04F3 801      Input Parameters:
04F3 802
04F3 803      R4      PCB Address of Current (Parent) Process
04F3 804      R5      Pointer to AST Control Block
04F3 805
04F3 806      ACB_L_CPULIM(R5)      Original CPU Limit of Subprocess
04F3 807      ACB_L_CPUTIM(R5)     CPU Time Used by Subprocess
04F3 808
04F3 809      Output Parameter:
04F3 810
04F3 811      PHD$L_CPULIM(R1)      The CPU limit of the parent is adjusted
04F3 812                          according to the following rules.
04F3 813
04F3 814      If the subprocess has unused CPU time limit (a credit) ,
04F3 815      that credit is added to the parent's CPU limit.
04F3 816
04F3 817      If the subprocess exceeded its CPU time allotment, an attempt
04F3 818      is made to pass the deficit to the parent.
04F3 819
04F3 820      If the deficit is smaller than the parent's CPU limit, the
04F3 821      deficit is simply subtracted from CPU limit.
04F3 822
04F3 823      If the deficit is larger than the parent's CPU limit, the
04F3 824      parent is forced into the CPU time expiration path by
04F3 825      loading its CPUTIM into its CPULIM field, indicating
04F3 826      that its CPU time limit has expired. (Note that loading CPULIM
04F3 827      into CPUTIM has the same effect but causes double accounting
04F3 828      for the CPU time already charged to the subprocess.)
04F3 829
04F3 830      Side Effects:
04F3 831
04F3 832      The AST control block is released to nonpaged pool.
04F3 833
04F3 834      Environment:
04F3 835
04F3 836      IPL$_ASTDEL while carrying pointer to AST control block.
04F3 837      :-
04F3 838
04F3 839      RETQUOTA:
04F3 840      MOVL      @#CTL$GL PHD,R1      ; Get safe pointer to process header
04F3 841      TSTL      PHD$L_CPULIM(R1)    ; Check for no quota
04F3 842      BEQL      20$                 ; Skip return if no quota
04F3 843      SUBL3     ACB_L_CPUTIM(R5),ACB_L_CPULIM(R5),R0 ; Get difference
04F3 844      BGEQU     10$                 ; Excess or deficit (Branch if excess)
04F3 845      MNEGL     R0,-(SP)             ; Handle deficit very carefully
```

```
51  00000000'9F  D0 04F3 840
      5C A1      D5 04FA 841
      1C        13 04FD 842
50  1C A5  20 A5  C3 04FF 843
      10        1E 0505 844
      7E 50     CE 0507 845
```


5C A1	8E	D1	050A	846		CMPL	(SP)+,PHD\$\$_CPULIM(R1)	; Can creator accommodate deficit?
	07	1F	050E	847		BLSSU	10\$	; Branch if yes
5C A1	38 A1	D0	0510	848		MOVL	PHD\$\$_CPUTIM(R1),PHD\$\$_CPULIM(R1)	; Otherwise, force CPU time
			0515	849				; expiration by setting limit to time
	04	11	0515	850		BRB	20\$	; Deallocate AST control block
			0517	851				
5C A1	50	C0	0517	852	10\$:	ADDL2	R0,PHD\$\$_CPULIM(R1)	; Give back excess
50	55	D0	051B	853	20\$:	MOVL	R5,R0	; Get address for release
00000000	'GF	17	051E	854		JMP	G^EXE\$DEANONPAGED	; Return via EXE\$DEANONPAGED
			0524	855				
			0524	856		.END		

SYSDELPRC
Symbol table

- DELETE PROCESS SYSTEM SERVICE

G 10

16-SEP-1984 01:57:52 VAX/VMS Macro V04-00
5-SEP-1984 03:52:26 [SYS.SRC]SYSDELPRC.MAR;1

Page 19
(1)

\$\$T1	=	00000001		
\$\$T2	=	00000004		
ACBSB_RMOD	=	0000000B		
ACBSL_AST	=	00000010		
ACBSL_KAST	=	00000018		
ACBSL_PID	=	0000000C		
ACBSV_KAST	=	00000007		
ACBSV_NODELETE	=	00000005		
ACB_L_CPULIM	=	0000001C		
ACB_L_CPUTIM	=	00000020		
ACCSL_TERMLEN	=	00000054		
ACCSL_BIOCNT	=	0000003C		
ACCSL_CPUTIM	=	0000002C		
ACCSL_DIOCNT	=	00000040		
ACCSL_FINALSTS	=	00000004		
ACCSL_JOBID	=	0000000C		
ACCSL_OWNER	=	00000050		
ACCSL_PAGEFLTS	=	00000030		
ACCSL_PGFLPEAK	=	00000034		
ACCSL_PID	=	00000008		
ACCSL_VOLUMES	=	00000044		
ACCSL_WSPEAK	=	00000038		
ACCSQ_LOGIN	=	00000048		
ACCSQ_TERMTIME	=	00000010		
ACCSL_ACCOUNT	=	00000018		
ACCSL_USERNAME	=	00000020		
ACCSW_MSGTYP	=	00000000		
ARBSL_RIGHTSLIST	=	00000020		
BUGS_FILCNTNONZ	*****		X	02
CCBSB_AMOD	=	00000009		
CCBSL_LENGTH	=	00000010		
CTLSAL_STACK	*****		X	02
CTLSGL_CCBASE	*****		X	02
CTLSGL_FINALSTS	*****		X	02
CTLSGL_PHD	*****		X	02
CTLSGL_USRUNDWN	*****		X	02
CTLSGL_VOLUMES	*****		X	02
CTLSGL_WSPEAK	*****		X	02
CTLSGQ_LOGIN	*****		X	02
CTLSGW_CHINDX	*****		X	02
CTLSGW_NMIOCH	*****		X	02
CTLST_ACCOUNT	*****		X	02
CTLST_USERNAME	*****		X	02
DEANONPAGED	0000033A	R		02
DELETE	0000002F	R		02
DELETE_JT	0000021A	R		02
DELPAGE	000003C3	R		02
DESTROY	000002BB	R		02
DESTROY_SYNCH	000003FF	R		02
EVT\$ RESUME	*****		X	03
EXESALLOCI	*****		X	03
EXESDEANONPAGED	*****		X	03
EXESDELPRC	00000000	RG		03
EXESGL_USRUNDWN	*****		X	02
EXESGQ_SYSTIME	*****		X	02
EXESNAMPID	*****		X	03
EXESPRCDELMG	*****		X	02

EXESRESETVEC	*****	X	02
IOSM_NOW	=	00000040	
IOS_WRITEVBLK	=	00000030	
IOCSDISMOUNT	*****	X	02
IPLS_ASTDEL	=	00000002	
IPLS_SYNCH	=	00000008	
JIBSL_MPID	=	00000054	
JIBSL_MTLFL	=	00000000	
JIBSW_PRCNT	=	00000044	
JOB_FAO	0000001C	R	02
JOB_STRING_LEN	=	00000010	
LNH_SYSTEM_DIRECTORY_DESC	00000000	R	02
MESSAGES	000002A1	R	02
MMGSDEALLOCPAGFIL	*****	X	02
MMGSDELCONPFN	*****	X	02
MMGSGL_PAGSWPVC	*****	X	02
MMGSINSPFNH	*****	X	02
MMGSREFCNTNEG	*****	X	02
MMGSVAPTECHK	*****	X	02
MSG\$ DELPROC	=	00000003	
P1SYSVECTORS	*****	X	02
PCBSB_ASTACT	=	0000000C	
PCBSB_DPC	=	0000002A	
PCBSL_ASTQFL	=	00000010	
PCBSL_EOWNER	=	00000068	
PCBSL_EPID	=	00000064	
PCBSL_JIB	=	00000080	
PCBSL_OWNER	=	0000001C	
PCBSL_PHD	=	0000006C	
PCBSL_PID	=	00000060	
PCBSL_STS	=	00000024	
PCBSL_WSSWP	=	00000020	
PCBSQ_PRIV	=	00000084	
PCBST_LNAME	=	00000070	
PCBSV_BATCH	=	0000000E	
PCBSV_DELPEN	=	00000001	
PCBSV_INTER	=	00000019	
PCBSV_RESPEN	=	00000005	
PCBSV_SSRWAIT	=	0000000A	
PCBSW_BIOCNT	=	0000003A	
PCBSW_BIOLM	=	0000003C	
PCBSW_DIOCNT	=	0000003E	
PCBSW_DIOLM	=	00000040	
PCBSW_PRCNT	=	00000042	
PCBSW_TMBU	=	00000032	
PFNSAB_STATE	*****	X	02
PFNSAW_REFcnt	*****	X	02
PFNSC_FREPAGLST	=	00000000	
PFNSM_MODIFY	=	00000080	
PHDSL_BIOCNT	=	00000058	
PHDSL_CPULIM	=	0000005C	
PHDSL_CPUTIM	=	00000038	
PHDSL_DIOCNT	=	00000054	
PHDSL_EXTRACPU	=	00000110	
PHDSL_POLRASTL	=	000000CC	
PHDSL_PAGEFLTS	=	0000004C	
PHDSL_PSTBASOFF	=	00000020	

SYSDELPRC
Symbol table

- DELETE PROCESS SYSTEM SERVICE

H 10

16-SEP-1984 01:57:52 VAX/VMS Macro V04-00
5-SEP-1984 03:52:26 [SYS.SRC]SYSDELPRC.MAR;1

Page 20
(1)

PHD\$S_POLR	= 00000018		
PHD\$V_POLR	= 00000000		
PHD\$W_PHVINDEX	= 00000042		
PHD\$W_PSTLAST	= 00000024		
PHD\$W_SWAPSIZE	= 00000052		
PHD\$W_WSLIST	= 00000008		
PHD\$W_WSLOCK	= 0000000C		
PHV\$GC_PIXBAS	*****	X	02
PR\$IPC	= 00000012		
PRIS_RES AVL	= 00000002		
PSL\$C_SUPER	= 00000002		
PTE\$S_PFN	= 00000015		
PTE\$V_MODIFY	= 0000001A		
PTE\$V_VALID	= 0000001F		
RELQUOTA	0000024A	R	02
RETQUOTA	000004F3	R	02
RSNS_ASTWAIT	= 00000001		
RUNDWN	00000100	R	02
SCH\$C_SWPPIX	*****	X	03
SCH\$GC_CURPCB	*****	X	02
SCH\$GL_MAXPIX	*****	X	02
SCH\$GL_NULLPCB	*****	X	02
SCH\$GL_PCBVEC	*****	X	02
SCH\$GW_DELPDCT	*****	X	02
SCH\$GW_PROCCNT	*****	X	02
SCH\$IOLOCKW	*****	X	02
SCH\$IOUNLOCK	*****	X	02
SCH\$NEWLVL	*****	X	02
SCH\$QAST	*****	X	03
SCH\$RAVAIL	*****	X	02
SCH\$RSE	*****	X	03
SCH\$RWAIT	*****	X	02
SCH\$SCHED	*****	X	02
SCH\$SWPWAKE	*****	X	02
SEC\$C_LENGTH	= 00000020		
SEC\$C_PAGCNT	= 0000001C		
SEC\$C_REFCNT	= 00000018		
SEC\$C_VPXPFC	= 00000008		
SEC\$S_VPX	= 00000016		
SEC\$V_VPX	= 00000000		
SGN\$GC_PTPAGCNT	*****	X	02
SS\$_NOEXPR	= 000008E8		
SS\$_NORMAL	= 00000001		
SUBDELETE	000000B3	R	02
SUBWIPL	000000FC	R	02
SWP\$GW_BALCNT	*****	X	02
SYSSASSIGN	*****	GX	02
SYSSDALLOC	*****	GX	02
SYSSDASSGN	*****	GX	02
SYSSDELLNM	*****	GX	02
SYSSDELPRC	*****	GX	02
SYSSDELTVA	*****	GX	02
SYSSFAO	*****	X	02
SYSSGW_BJOBCNT	*****	X	02
SYSSGW_IJOBCNT	*****	X	02
SYSSQIO	*****	GX	02
SYSSRMSRUNDWN	*****	X	02

SYSSRUNDWN
TERMMSG
TMP...

***** X 02
00000403 R 02
= 00000000

+-----+
! 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
YSEXEPAGED	00000524 (1316.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
AEXENONPAGED	00000052 (82.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.03	00:00:02.38
Command processing	110	00:00:00.53	00:00:05.52
Pass 1	494	00:00:19.82	00:01:01.05
Symbol table sort	0	00:00:03.34	00:00:10.41
Pass 2	173	00:00:03.97	00:00:12.52
Symbol table output	21	00:00:00.19	00:00:00.91
Psect synopsis output	8	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	838	00:00:27.92	00:01:32.83

The working set limit was 1950 pages.

113562 bytes (222 pages) of virtual memory were used to buffer the intermediate code.

There were 110 pages of symbol table space allocated to hold 2048 non-local and 46 local symbols.

856 source lines were read in Pass 1, producing 25 object records in Pass 2.

46 pages of virtual memory were used to define 44 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	18
_\$255\$DUA28:[SYS.LIB]STARLET.MLB;2	23
TOTALS (all libraries)	41

2279 GETS were required to define 41 macros.

There were no errors, warnings or information messages.

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

0383 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SYSRMPSC
LIS

SYSDCLEXH
LIS

SYSDEVALC
LIS

SYSCURTIM
LIS

SYSDBGSC
LIS

SYSENQDEQ
LIS

SYSDCLCMH
LIS

SYSDERLMB
LIS

SYSDBSSGN
LIS

SYSDELPRC
LIS