
```
000000  P P P P P P P P  C C C C C C C C  C C C C C C C C  R R R R R R R R  A A A A A A  S S S S S S S S  H H  H H
000000  P P P P P P P P  C C C C C C C C  C C C C C C C C  R R R R R R R R  A A A A A A  S S S S S S S S  H H  H H
00      00  PP      PP  CC      CC      CC      RR      RR  AA      AA  SS      SS      HH      HH
00      00  PP      PP  CC      CC      CC      RR      RR  AA      AA  SS      SS      HH      HH
00      00  PP      PP  CC      CC      CC      RR      RR  AA      AA  SS      SS      HH      HH
00      00  PP      PP  CC      CC      CC      RR      RR  AA      AA  SS      SS      HH      HH
00      00  P P P P P P  CC      CC      CC      R R R R R R  AA      AA  S S S S S S  H H H H H H H H
00      00  P P P P P P  CC      CC      CC      R R R R R R  AA      AA  S S S S S S  H H H H H H H H
00      00  PP      PP  CC      CC      CC      RR      RR  A A A A A A A A  SS      SS      H H      H H
00      00  PP      PP  CC      CC      CC      RR      RR  A A A A A A A A  SS      SS      H H      H H
00      00  PP      PP  CC      CC      CC      RR      RR  AA      AA  S S S S S S  H H      H H
00      00  PP      PP  CC      CC      CC      RR      RR  AA      AA  S S S S S S  H H      H H
000000  PP      PP  C C C C C C C C  C C C C C C C C  R R      RR  AA      AA  S S S S S S  H H      H H
000000  PP      PP  C C C C C C C C  C C C C C C C C  R R      RR  AA      AA  S S S S S S  H H      H H

L L      I I I I I I  S S S S S S S S
L L      I I I I I I  S S S S S S S S
L L      I I      S S
L L      I I      S S
L L      I I      S S
L L      I I      S S
L L      I I      S S S S S S
L L      I I      S S S S S S
L L      I I      S S
L L      I I      S S
L L      I I      S S
L L      I I      S S
L L L L L L L L L L  I I I I I I  S S S S S S S S
L L L L L L L L L L  I I I I I I  S S S S S S S S
```


OPCCRASH
Table of contents

-- CRASH SYSTEM IMAGE

L 2

16-SEP-1984 01:59:50 VAX/VMS Macro V04-00

Page 0

(2) 73

CRASH SYSTEM ROUTINE

```
0001 1 .TITLE OPCCRASH -- CRASH SYSTEM IMAGE
0001 2 .IDENT 'V04-000'
0001 3
0001 4 *****
0001 5 *
0001 6 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0001 7 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0001 8 * ALL RIGHTS RESERVED.
0001 9 *
0001 10 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0001 11 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0001 12 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0001 13 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0001 14 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0001 15 * TRANSFERRED.
0001 16 *
0001 17 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0001 18 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0001 19 * CORPORATION.
0001 20 *
0001 21 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0001 22 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0001 23 *
0001 24 *
0001 25 *****
0001 26 ++
0001 27
0001 28 FACILITY:
0001 29
0001 30 VAX/VMS OPERATOR COMMUNICATIONS PROCESS
0001 31
0001 32 ABSTRACT:
0001 33
0001 34 CRASH SYSTEM WITH BUG CHECK
0001 35
0001 36 AUTHOR: R.HEINEN 12-JUN-78 (V06)
0001 37
0001 38 MODIFIED BY:
0001 39
0001 40 V03-005 DWT0226 David W. Thiel 24-Jul-1984
0001 41 Correct wait for modified page flush.
0001 42 Improve addressing modes.
0001 43 Remove quorum disk check when dismounting system disk.
0001 44
0001 45 V03-004 PRB0329 Paul Beck 11-Apr-1984 1:20
0001 46 Don't dismount system disk if it's also the cluster
0001 47 quorum disk and its votes are needed to maintain the quorum.
0001 48
0001 49 V03-003 DWT0205 David W. Thiel 05-Apr-1984
0001 50 Update interface for quorum adjustment.
0001 51
0001 52 V03-002 PRB0323 Paul Beck 11-Mar-1984 16:17
0001 53 Add support for cluster-wide shutdown and for quorum reduction
0001 54 when a cluster node is permanently removed from the cluster.
0001 55
0001 56 V03-001 KDM0002 Kathleen D. Morse 28-Jun-1982
0001 57 Added $PRDEF.
```

OPCCRASH
V04-000

-- CRASH SYSTEM IMAGE

N 2

16-SEP-1984 01:59:50 VAX/VMS Macro V04-00
5-SEP-1984 02:29:43 [OPCOM.SRC]OPCCRASH.MAR;1

Page 2
(1)

0001 58 :
0001 59 :--
0001 60
0001 61
0001 62
0001 63
0001 64
0001 65
0001 66
0001 67
0001 68
0001 69
0001 70
0001 71

\$CCBDEF
\$DEVDEF
\$IODEF
\$IPLDEF
\$PRDEF
\$RSNDEF
\$RVTDEF
\$UCBDEF
\$VCBDEF
\$CLUBDEF
\$CSBDEF

: DEFINE CCB STRUCTURE
: DEFINE DEVICE CHARACTERISTICS
: DEFINE I/O FUNCTION CODES
: DEFINE IPL LEVELS
: DEFINE PROCESSOR REGISTERS
: DEFINE WAIT REASON CODES
: DEFINE RVT STRUCTURE
: DEFINE UCB STRUCTURE
: DEFINE VCB STRUCTURE
: DEFINE CLUB STRUCTURE
: DEFINE CSB STRUCTURE


```
0001 73 .SBTTL CRASH SYSTEM ROUTINE
00000000 74 .PSECT OPCOM_CRASH,NOWRT
0000 75 :++
0000 76 : THIS IMAGE IS RUN TO CRASH THE SYSTEM WITH A BUGCHECK
0000 77 :
0000 78 : FUNCTIONAL DESCRIPTION:
0000 79 :
0000 80 : INPUTS:
0000 81 :
0000 82 :     NONE
0000 83 :
0000 84 : OUTPUTS:
0000 85 :
0000 86 :     NONE
0000 87 :--
00000000 88 .PSECT OPCOM_CODE,PAGE
0000 89 .ALIGN PAGE
0000 90 :
0000 91 : MESSAGE TO PRINT
0000 92 :
65 72 20 72 6F 74 61 72 65 70 4F 00' 0000 93 CRASH_MESSAGE: .ASCIC /Operator requested system shutdown/
74 73 79 73 20 64 65 74 73 65 75 71 000C
6E 77 6F 64 74 75 68 73 20 6D 65 0018
22 0000
00000027 0023 94 CHANNEL: .BLKL 1 ; channel number for dismount QIO's
4E 55 24 43 50 4F 0000002F'010E0000' 0027 95 UNLOAD: .ASCID 'OPC$UNLOAD' ; logical name of unload flag
44 41 4F 4C 0035
45 52 24 43 50 4F 00000041'010E0000' 0039 96 REBOOT: .ASCID 'OPC$REBOOT' ; logical name of reboot flag
54 4F 4F 42 0047
4C 43 24 43 50 4F 00000053'010E0000' 004B 97 CLUSTER_SHUT: .ASCID 'OPC$CLUSTER_SHUTDOWN' ; logical name of cluster shutdown
4F 44 54 55 48 53 5F 52 45 54 53 55 0059
4E 57 0065
45 52 24 43 50 4F 0000006F'010E0000' 0067 98 CLUSTER_REMOVE: .ASCID 'OPC$REMOVE_NODE' ; logical name of remove node from cluster
45 44 4F 4E 5F 45 56 4F 4D 0075
00000004 007E 99 NAMDESC: .LONG 4 ; descriptor for result string
CC000086' 0082 100 .ADDRESS NAMBUF
0000008A 0086 101 NAMBUF : .BLKL 1 ; buffer for unload flag
008A 102 :
008A 103 : Define timer for 1/10 second
008A 104 :
FFFFFFF FFF0BDC0 008A 105 ONE_TENTH: .LONG -1*1000*1000*1, -1
0092 106
0092 107
0000 0092 108 .ENTRY CRASH,^M<>
0094 109 $CMKRNLS B^10$ ; GOTO KERNEL MODE
04 00A0 110 RET
00A1 111
00A1 112
0000 00A1 113 10$: .WORD ^M<>
00A3 114 :
00A3 115 : We must cause the caches for the system disk(s) to be flushed. This is
00A3 116 : done by assigning a channel, marking the volume for dismount, and issuing
00A3 117 : a dismount QIO. We also set the unload bit in each UCB according to the
00A3 118 : unload flag provided from the shutdown procedure. If the system disk is a
00A3 119 : volume set, do this for all the volumes. The channel is assigned by hand,
00A3 120 : to circumvent any logical name screw-ups that could get in the way.
00A3 121 :
```



```
00000000'GF 16 00A3 122 JSB G^IOCS$FFCHAN ; get an available channel number
69 50 E9 00A9 123 BLBC R0,80$ ; if no channel, tough luck
FF72 CF 51 B0 00AC 124 MOVW R1,CHANNEL ; store channel number
09 A2 01 90 00B1 125 MOVW #1,CCBS$B AMOD(R2) ; and mark in use
50 FF6E CF 7E 00B5 126 MOVAQ UNLOAD,R0 ; get address of OPC$UNLOAD string
012A 30 00BA 127 BSBW GET FLAG ; get the boolean value of OPC$UNLOAD
5A 50 D0 00BD 128 MOVL R0,R10 ; save the value in R10
51 00000000'GF D0 00C0 129 MOVL G^EXE$GL SYSUCB,R1 ; get system UCB address
50 34 A1 D0 00C7 130 MOVL UCB$$_VCB(R1),R0 ; get VCB address...
53 20 A0 D0 00CB 131 MOVL VCB$$_RVT(R0),R3 ; to get RVT address
55 01 D0 00CF 132 MOVL #1,R5 ; assume 1 volume
54 D4 00D2 133 CLRL R4 ; clear loop and RVT index
53 51 D1 00D4 134 CMPL R1,R3 ; if RVT eql UCB,
0B 13 00D7 135 BEQL 40$ ; then this is not a volume set
55 0B A3 9A 00D9 136 MOVZBL RVT$B_NVOLS(R3),R5 ; get count of volumes
51 44 A344 D0 00DD 137 30$: MOVL RVT$$_UCBLST(R3)[R4],R1 ; get next UCB address
2D 13 00E2 138 BEQL 70$ ; branch if not mounted
01 0C 5A F0 00E4 139 40$: INSV R10,#UCBS$V_UNLOAD,#1,- ; set the unload bit accordingly
64 A1 D0 00E8 140 UCB$$_STS(R1) ;
00 38 A1 15 E2 00EA 141 MOVL R1,CCBS$$_UCB(R2) ; "assign" the channel
C8 54 55 F2 00ED 142 BBSS #DEV$V_DMT,UCB$$_DEVCHAR(R1),60$ ; mark volume for dismount
00F2 143 60$: $QIOW S CHAN=CHANNEL,FUNC=#IOS$_ACPCONTROL!IOSM_DMOUNT
0111 144 70$: AOBLS$ R5,R4,30$ ; loop through RVT
0115 145 ;
0115 146 ; SET THE REBOOT FLAG DEPENDING ON THE BOOLEAN VALUE OF OPC$REBOOT.
0115 147 ;
50 FF20 CF 7E 0115 148 80$: MOVAQ REBOOT,R0 ; get lognam descriptor address
00CA 30 011A 149 BSBW GET FLAG ; get the value of OPC$REBOOT
01 00' 50 F0 011D 150 INSV R0,S^#EXE$V_BUGREBOOT,#1,- ; set the system reboot flag accordingly
00000000'GF 0121 151 G^EXE$GL_FLAGS ;
0126 152 ;
0126 153 ; WRITE MODIFIED PAGE LIST AND
0126 154 ; WAIT FOR IT TO COMPLETE
0126 155 ;
7E DC 0126 156 MOVPSL -(SP) ; RESUME PSL MUST BE ON STACK
0128 157 SETIPL B^140$ ; RAISE TO IPL SYNCH
00000000'GF D4 012C 158 CLRL G^SCH$GL_MFY LIM ; FORCE SWAPPER ACTIVITY
00000000'GF D4 0132 159 CLRL G^SCH$GL_MFYLO LIM ;
54 00000000'GF D0 0138 160 MOVL G^SCH$GL_CURPCB,R4 ; PCB ADDRESS
50 50 0B 3C 013F 161 MOVZWL #RSN$ MP[EMPTY],R0 ; EVENT = MODIFIED PAGE LIST EMPTY
00000000'GF 16 0142 162 JSB G^SCH$RWAIT ; WAIT FOR EVENT
0148 163 ;
0148 164 ; If we're in a cluster, test for two special cases:
0148 165 ; 1. Cluster-wide shutdown.
0148 166 ; 2. Node being removed from cluster necessitating quorum reduction.
0148 167 ;
00000000'GF D5 0148 168 100$: TSTL G^CLUS$GL_CLUB ; Are we in a VAXcluster?
16 13 014E 169 BEQL 109$ ; if EQL, no.
50 FEF7 CF 7E 0150 170 MOVAQ CLUSTER_SHUT,R0 ; get lognam descriptor address
008F 30 0155 171 BSBW GET FLAG ; get the value of OPC$CLUSTER_SHUTDOWN
14 50 E8 0158 172 BLBS R0,T20$ ; If LBS, a cluster shutdown
50 FF08 CF 7E 015B 173 MOVAQ CLUSTER_REMOVE,R0 ; get logname descriptor address
0084 30 0160 174 BSBW GET FLAG ; get the value of OPC$CLUSTER_REMOVE
23 50 E8 0163 175 BLBS R0,200$ ; If LBS, removing node: reduce quorum
0166 176 ;
0166 177 ; BUG CHECK SYSTEM
0166 178 ;
```



```
0166 179 109$: SETIPL B^110$ ; LOCK OUT INTERRUPTS
016A 180 BUG CHECK OPERATOR,TYPE=FATAL ; CRASH THE SYSTEM
1F 016E 181 110$: .BYTE IPL$POWER ; END OF CODE THAT RUNS AT ELEVATED IPL
016F 182 :
016F 183 : *** CLUSTER-WIDE SHUTDOWN ***
016F 184 :
016F 185 120$: SETIPL B^140$ ; Make sure code won't page at SYNCH
0173 186 :
0173 187 : First, notify the connection manager that this is a cluster-wide shutdown.
0173 188 :
00000000'GF 16 0173 189 JSB G^CNX$SHUTDOWN ; Inform cnx man
0179 190 :
0179 191 : The connection manager will tell all other nodes that we are shutting down.
0179 192 : Other nodes will do the same; as they do, CSB$V_SHUTDOWN gets set. When
0179 193 : this flag is set for all active nodes in the cluster, we're done.
0179 194 :
0179 195 : Fortunately, the connection manager does all that, including the bugcheck.
0179 196 : All we have to do is give it room to breathe.
0179 197 :
0179 198 SETIPL #0 ; Desynchronize/allow paging
017C 199 130$: $HIBER_S ; Wait for connection manager to crash
F7 11 0183 200 BRB 130$ ; ...
0185 201
00000008 0185 202 140$: .LONG IPL$SYNCH ; End of locked page(s)
0189 203 :
0189 204 :
0189 205 : *** REDUCE QUORUM, USING THIS NODE'S VOTES ***
0189 206 :
0189 207 200$: SETIPL B^230$ ; Synchronize with connection manager
52 00000000'GF D0 018D 208 MOVL G^CLUGL CLUB,R2 ; Get address of CLUB
51 10 A2 D0 0194 209 MOVL CLUB$L_LOCAL_CSB(R2),R1 ; Get address of local CSB.
0198 210 :
0198 211 : We have located the CSB for this node. Use its votes to calculate
0198 212 : a new value for Quorum.
0198 213 :
0198 214 : New quorum calculation:  $Q' = (2Q+1-V)/2$ 
0198 215 :
51 50 A1 3C 0198 216 MOVZWL CSB$W VOTES(R1),R1 ; V
50 20 A2 3C 019C 217 MOVZWL CLUB$W QUORUM(R2),R0 ; Q
50 50 01 78 01A0 218 ASHL #1,R0,R0 ; 2Q
50 50 50 D6 01A4 219 INCL R0 ; 2Q+1
51 50 51 C2 01A6 220 SUBL2 R1,R0 ; 2Q+1-V
51 50 FF 8F 78 01A9 221 ASHL #-1,R0,R1 ; R1 = new quorum
01AE 222 :
01AE 223 : Tell the connection manager to set a new quorum. The connection manager
01AE 224 : must handle this since it is done in a coordinated fashion across the
01AE 225 : cluster. The quorum value is passed in R1.
01AE 226 :
00000000'GF 16 01AE 227 JSB G^CNX$CHANGE_QUORUM ; Set new active quorum
53 50 D0 01B4 228 MOVL R0,R3 ; Save status -- failure means try again lat
01B7 229 :
01B7 230 : Wait for the connection manager to effect the change (CLUB$V_ADJ_QUORUM flag)
01B7 231 :
01B7 232 SETIPL #0 ; Desynchronize/allow paging
01BA 233 :
01BA 234 : At this point, we wait in kernel mode for a short time, then test to see
01BA 235 : if the connection manager has completed its work.
```



```
01BA 236 ;
01BA 237 210$: $SETIMR_S - ; Define a tick
01BA 238 efn = S^#EXESC_SYSEFN - ; ...use system event flag
01BA 239 daytim = ONE_TENTH ; ...shouldn't be perceptible
14 50 E9 01C9 240 BLBC R0,220$ ; ...
01CC 241 $WAITFR_S -
01CC 242 efn = S^#EXESC_SYSEFN ; Wait a tick
08 50 E9 01D5 243 BLBC R0,220$ ; ...
01D8 244
DA 1C A2 AE 53 E9 01D8 245 BLBC R3,200$ ; Branch if a retry is required
1B E0 01DB 246 BBS #CLUB$V_ADJ_QUORUM, - ; If BS, not done yet.
FF83 31 01E0 247 CLUB$L_FLAGS(R2),210$
01E3 248 220$: BRW 109$ ; BugCheck
01E3 249
00000008 01E3 250 230$: .LONG IPL$_SYNCH ; End of page locking.
```

```
01E7 252 :++
01E7 253 : GET_FLAG
01E7 254 :
01E7 255 : Functional description:
01E7 256 :
01E7 257 :     This routine will translate a given logical name, and return a
01E7 258 :     status code that will reflect whether the logical name was set
01E7 259 :     to the DCL equivalent of the boolean condition TRUE. To wit, if
01E7 260 :     the logical name translates to a string that begins with a 'T',
01E7 261 :     'Y', or 'I', then return a success status.
01E7 262 :
01E7 263 : Input:
01E7 264 :
01E7 265 :     R0 = address of the logical name descriptor
01E7 266 :
01E7 267 : Output:
01E7 268 :
01E7 269 :     None.
01E7 270 :
01E7 271 : Implicit output:
01E7 272 :
01E7 273 :     The output buffer described by the RSLBUF parameter to the $STRNLOG
01E7 274 :     system service will contain the (possibly truncated) translation of
01E7 275 :     the input logical name. Only the first character is significant.
01E7 276 :
01E7 277 : Routine value:
01E7 278 :
01E7 279 :     The low bit of R0 will reflect the boolean value of the logical name.
01E7 280 :     Note that if the logical name is not defined, then the value is false.
01E7 281 : --
01E7 282 :
01E7 283 : GET_FLAG:
01E7 284 :     $STRNLOG_S      LOGNAM=(R0),-
01E7 285 :                     RSLBUF=NAMDESC ; Translate the logical name
01E7 286 :                     BLBC R0,13$    ; (only the first 4 bytes are returned)
01E7 287 :                     CMPW #SS$_NOTRAN,R0 ; Exit if error
01E7 288 :                     BEQL 13$      ; Did the name translate?
01E7 289 :                     MOVL #1,R0    ; Exit if no translation
01E7 290 :                     MOVZBL NAMBUF,R1 ; Assume TRUE
01E7 291 :                     CMPB #^A/Y/,R1 ; Get the first byte of the string
01E7 292 :                     BEQL 14$      ; Is it a 'Y'?
01E7 293 :                     CMPB #^A/y/,R1 ;
01E7 294 :                     BEQL 14$      ; Is it a 'y'?
01E7 295 :                     CMPB #^A/T/,R1 ;
01E7 296 :                     BEQL 14$      ; Is it a 'T'?
01E7 297 :                     CMPB #^A/t/,R1 ;
01E7 298 :                     BEQL 14$      ; Is it a 't'?
01E7 299 :                     CMPB #^A/I/,R1 ;
01E7 300 :                     BEQL 14$      ; Is it a 'I'?
01E7 301 :                     CLRL R0        ; Indicate FALSE
01E7 302 :                     RSB           ; Return
01E7 303 :
01E7 304 :
01E7 305 : .END      CRASH
```

50 2C 50 E9 01FC 286 BLBC R0,13\$

50 0000'8F B1 01FF 287 CMPW #SS\$_NOTRAN,R0

50 25 13 0204 288 BEQL 13\$

51 50 01 D0 0206 289 MOVL #1,R0

51 FE79 CF 9A 0209 290 MOVZBL NAMBUF,R1

51 59 8F 91 020E 291 CMPB #^A/Y/,R1

51 19 13 0212 292 BEQL 14\$

51 79 8F 91 0214 293 CMPB #^A/y/,R1

51 13 13 0218 294 BEQL 14\$

51 54 8F 91 021A 295 CMPB #^A/T/,R1

51 0D 13 021E 296 BEQL 14\$

51 74 8F 91 0220 297 CMPB #^A/t/,R1

51 07 13 0224 298 BEQL 14\$

51 31 91 0226 299 CMPB #^A/I/,R1

51 02 13 0229 300 BEQL 14\$

51 50 D4 022B 301 CLRL R0

51 05 05 022D 302 RSB

51 022E 303

51 022E 304

51 022E 305

OPCCRASH
Symbol table

-- CRASH SYSTEM IMAGE

G 3

16-SEP-1984 01:59:50 VAX/VMS Macro V04-00
5-SEP-1984 02:29:43 [OPCOM.SRC]OPCCRASH.MAR;1

Page 8
(3)

```

$ST1 = 00000000
BUG$ OPERATOR = 00000000 X 04
CCBS$ AMOD = 00000009
CCBS$ UCB = 00000000
CHANNEL = 00000023 R 04
CLUS$GL CLUB = 00000000 X 04
CLUS$GL FLAGS = 0000001C
CLUS$GL LOCAL CSB = 00000010
CLUS$V ADJ QUORUM = 00000018
CLUS$W QUORUM = 00000020
CLUSTER REMOVE = 00000067 R 04
CLUSTER SHUT = 0000004B R 04
CNX$CHANGE QUORUM = 00000000 X 04
CNX$SHUTDOWN = 00000000 X 04
CRASH = 00000092 RG 04
CRASH MESSAGE = 00000000 R 04
CSBS$VOTES = 00000050
DEVS$ DMT = 00000015
EXESC SYSEFN = 00000000 X 04
EXES$GL FLAGS = 00000000 X 04
EXES$GL SYSUCB = 00000000 X 04
EXES$V BUGREBOOT = 00000000 X 04
GET FLAG = 000001E7 R 04
IOS$ DMOUNT = 00000400
IOS$ ACPCONTROL = 00000038
IOC$FFCHAN = 00000000 X 04
IPL$ POWER = 0000001F
IPL$ SYNCH = 00000008
MSG ACCOUNT = 00000028
MSG DEVCNT = 00000010
MSG DEVNAME = 00000011
MSG DEVUNIT = 0000000E
MSG HDR = 00000032
MSG LINK = 00000000
MSG MSG = 0000000C
MSG NUMBER = 00000008
MSG PRI = 00000030
MSG PRIV = 00000010
MSG REPLYMBX = 0000000E
MSG SIZE = 000000CC
MSG UIC = 00000018
MSG USRNAME = 0000001C
NAMBUF = 00000086 R 04
NAMDESC = 0000007E R 04
ONE TENTH = 0000008A R 04
OPC$C REPEAT = 00000005
OPC$GL MS MAXSZ = 0000009A
OTV ACTIVE = 00000000
OT DESC = 00000010
OT LINK = 00000000
OT NAME = 00000018
OT SERVICE = 00000008
OT SIZE = 00000028
OT STATUS = 0000000C
PR$ IPL = 00000012
REBOOT = 00000039 R 04
RSNS MPLEMPTY = 0000000B

```

```

RVT$B NVOLS = 0000000B
RVT$GL UCBLST = 00000044
SCH$GL CURPCB = 00000000 X 04
SCH$GL MFYLIM = 00000000 X 04
SCH$GL MFYLOLIM = 00000000 X 04
SCH$R WAIT = 00000000 X 04
SS$ NOTRAN = 00000000 X 04
STV TIMER = 00000000
SYS$CMKRNL = 00000000 GX 04
SYS$HIBER = 00000000 GX 04
SYS$QIOW = 00000000 GX 04
SYS$SETIMR = 00000000 GX 04
SYS$STRNLOG = 00000000 GX 04
SYS$WAITFR = 00000000 GX 04
UCB$GL DEVCHAR = 00000038
UCB$GL VCB = 00000034
UCB$V UNLOAD = 0000000C
UCB$W STS = 00000064
UNLOAD = 00000027 R 04
VCB$GL RVT = 00000020

```

OPC
V04

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

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
OPCOM_DEF	000000CC (204.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$ABSS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
OPCOM_CRASH	00000000 (0.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE
OPCOM_CODE	0000022E (558.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC PAGE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	12	00:00:00.06	00:00:00.43
Command processing	97	00:00:00.80	00:00:09.06
Pass 1	356	00:00:15.57	00:00:51.17
Symbol table sort	0	00:00:02.62	00:00:05.20
Pass 2	60	00:00:02.55	00:00:07.04
Symbol table output	9	00:00:00.11	00:00:00.18
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	536	00:00:21.74	00:01:13.12

The working set limit was 1350 pages.

85571 bytes (168 pages) of virtual memory were used to buffer the intermediate code.

There were 90 pages of symbol table space allocated to hold 1678 non-local and 18 local symbols.

402 source lines were read in Pass 1, producing 19 object records in Pass 2.

33 pages of virtual memory were used to define 32 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[OPCOM.OBJ]OPCOMLIB.MLB;1	0
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	10
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	19
TOTALS (all libraries)	29

1846 GETS were required to define 29 macros.

There were no errors, warnings or information messages.

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

0290 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

