

UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	3333333333	2222222222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	3333333333	2222222222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	3333333333	2222222222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUU	UUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLL	333	222
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLLLLLLLLLLLLLLL	3333333333	22222222222222
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLLLLLLLLLLLLLLL	3333333333	22222222222222
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	TTTTTTTTTTTTTTTT	IIIIIIIIII	LLLLLLLLLLLLLLLL	3333333333	22222222222222

P
S
S
S

```

IIIIII  000000  CCCCCCCC  000000  LL      LL      EEEEEEEEE  CCCCCCCC  TTTTTTTTTT
IIIIII  000000  CCCCCCCC  000000  LL      LL      EEEEEEEEE  CCCCCCCC  TTTTTTTTTT
  II     00     00  CC      00     00  LL      LL      EE          CC          TT
  II     00     00  CC      00     00  LL      LL      EE          CC          TT
  II     00     00  CC      00     00  LL      LL      EE          CC          TT
  II     00     00  CC      00     00  LL      LL      EE          CC          TT
  II     00     00  CC      00     00  LL      LL      EE          CC          TT
  II     00     00  CC      00     00  LL      LL      EE          CC          TT
  II     00     00  CC      00     00  LL      LL      EE          CC          TT
  II     00     00  CC      00     00  LL      LL      EE          CC          TT
  II     00     00  CC      00     00  LL      LL      EE          CC          TT
  II     00     00  CC      00     00  LL      LL      EE          CC          TT
IIIIII  000000  CCCCCCCC  000000  LLLLLLLLLL  LLLLLLLLLL  EEEEEEEEE  CCCCCCCC  TT          ....
IIIIII  000000  CCCCCCCC  000000  LLLLLLLLLL  LLLLLLLLLL  EEEEEEEEE  CCCCCCCC  TT          ....

```

```

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
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```

(2)	60
(3)	161
(4)	271
(5)	374
(6)	455
(7)	501
(8)	532
(9)	575

DECLARATIONS
IOCOLLECT - I/O PERFORMANCE DATA COLLECTION CONTROL
DATA COLLECT - COLLECT I/O PERFORMANCE DATA
ALLOCATE - ALLOCATE DATA COLLECTION STRUCTURE
COPY DATA - COPY DATA TO TEMPORARY BUFFER
DEALLOCATE - DEALLOCATE DATA COLLECTION STRUCTURE
DELETE - DELETE DATA COLLECTION STRUCTURE
TIMAST - TIMER AST ROUTINE

```
0000 1 .TITLE IOCOLLECT - I/O PERFORMANCE DATA COLLECTION
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: Performance Monitoring
0000 30
0000 31 : ABSTRACT:
0000 32 : IOCOLLECT is responsible for controlling I/O performance
0000 33 : monitoring and logging data to a mag tape or disk file.
0000 34
0000 35 : ENVIRONMENT:
0000 36 : MODE = KERNEL, EXEC
0000 37
0000 38 : AUTHOR: S. S. AMWAY, CREATION DATE: 24-Oct-1983
0000 39 : (ORIGINAL AUTHOR: D.N. CUTLER, CREATION DATE: 20-Nov-1977)
0000 40
0000 41 : MODIFIED BY:
0000 42 : V03-002 SSA0024 Stan Amway 10-Apr-1984
0000 43 : Moved disk geometry info in SRQ record to avoid
0000 44 : conflicting usage by newly defined field PIB$W_SRQ_ACON.
0000 45
0000 46 : V03-001 SSA0008 Stan Amway 21-Feb-1984
0000 47 : Record node name to data file as part of
0000 48 : the initialization record. This is done to aid in
0000 49 : the analysis of data recorded simultaneously on
0000 50 : several nodes in a VAXcluster.
0000 51
0000 52 :
0000 53 : When allocating PDB, check for previous monitoring
0000 54 : process abort which left a PDB allocated. This can
0000 55 : only happen if another process issues a $DELPRC
0000 56 : service or a user issues a DCL STOP PROCESS command.
0000 57 :--
```



```
0000 60      .SBTTL  DECLARATIONS
0000 61
0000 62 :
0000 63 : INCLUDE FILES:
0000 64 :
0000 65 :
0000 66      $DCDEF      ;Define device classes & types
0000 67      $DDBDEF     ;DEFINE DDB OFFSETS
0000 68      $DEVDEF     ;DEFINE DEVCHAR/DEVCHAR2 bits
0000 69      $DYNDEF     ;DEFINE DATA STRUCTURE CODES
0000 70      $IPLDEF     ;DEFINE INTERRUPT PRIORITY LEVELS
0000 71      $IODEF     ;DEFINE I/O FUNCTION CODES
0000 72      $PBHDEF     ;DEFINE PBH OFFSETS
0000 73      $PCBDEF     ;DEFINE PCB OFFSETS
0000 74      $PDBDEF     ;DEFINE PDB OFFSETS
0000 75      $PIBDEF     ;DEFINE PIB OFFSETS
0000 76      $PRDEF     ;DEFINE PROCESSOR REGISTERS
0000 77      $SYIDDEF   ;Define GETSYI item codes
0000 78      $UCBDEF     ;DEFINE UCB OFFSETS
0000 79
0000 80 :
0000 81 : MACROS:
0000 82 :
0000 83 :
0000 84 :
0000 85 : GENERATE STRING DESCRIPTOR AND TEXT
0000 86 :
0000 87 :
0000 88      .MACRO .DESCR STRING
0000 89          .PSECT TEXT, LONG, RD, WRT, NOEXE
0000 90      $START$ = .
0000 91          .ASCII \STRING\
0000 92      $END$ = .
0000 93          .PSECT DATA, LONG, RD, WRT, NOEXE
0000 94          .LONG $END$-$START$
0000 95          .LONG $START$
0000 96      .ENDM .DESCR
0000 97
0000 98 :
0000 99 : EQUATED SYMBOLS:
0000 100 :
0000 101 :
00000200 0000 102 DATBUFSIZ=512 ;DATA COLLECTION BUFFER SIZE
00000008 0000 103 NUMDATBUF=11 ;NUMBER OF DATA COLLETION BUFFERS
00001000 0000 104 OUTBUFSIZ=4096 ;OUTPUT DATA BUFFER SIZE
00000014 0000 105 FLUSHCOUNT=20 ;BUFFER FLUSH CONTROL VALUE
0000 106
0000 107 :
0000 108 : OWN STORAGE:
0000 109 :
0000 110 :
00000000 0000 111          .PSECT DATA, LONG, RD, WRT, NOEXE
00000002 0000 112 ANDMSK: .BLKW 1 ;PACKET SELECTION 'AND' MASK
00000004 0002 113 BUFLen: .BLKW 1 ;LENGTH OF OUTPUT MESSAGE
00000204 0004 114 COPBUF: .BLKB DATBUFSIZ ;DATA COLLECTION COPY BUFFER
00000205 0204 115 DCLASS: .BLKB 1 ;DEVICE CLASS MATCH BYTE
0000020D 0205 116 DELTAT: .BLKW 1 ;DELTA TIME BEFORE TEST END
```

IC
S)
SI
S)
SI
SI
SI
SI
SI
SI
SI
SI
SI
SI
SI
SI
SI
SI
TI
UC
UC
UC
UC
XC

PS
--
:
\$A
DA
\$R
CC

PH
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As

TH
11
TH
6C

```

0000020E 020D 117 DVTYPE: .BLKB 1 ;DEVICE TYPE MATCH BYTE
0000020F 020E 118 FLAGS: .BLKB 1 ;FLAGS BYTES
00000217 020F 119 FUNCTN: .BLKB 1 ;PACKET SELECTION FUNCTION MASK
0000021F 0217 120 IOSTAT: .BLKB 1 ;I/O STATUS QUADWORD
      20 021F 121 MSGLEN: .BYTE PIBSK_SRQ_SIZE ;SIZE OF START OF I/O REQUEST MESSAGE
      18 0220 122      .BYTE PIBSK_SIO_SIZE ;SIZE OF START OF I/O TRANSACTION MESSAGE
      18 0221 123      .BYTE PIBSK_EIO_SIZE ;SIZE OF END OF I/O TRANSACTION MESSAGE
      10 0222 124      .BYTE PIBSK_ERQ_SIZE ;SIZE OF END OF I/O REQUEST MESSAGE
      10 0223 125      .BYTE PIBSK_ARQ_SIZE ;SIZE OF ABORT I/O REQUEST MESSAGE
00000228 0224 126 NUMBUF: .BLKB 1 ;NUMBER OF BUFFERS OF DATA TO COLLECT
00001228 0228 127 OUTBUF: .BLKB OUTBUFSIZ ;DATA OUTPUT BUFFER
0000122A 1228 128 XORMSK: .BLKB 1 ;PACKET SELECTION 'XOR' MASK
      122A 129 SYI_LIST: ; Item List for GETSYI
      0008 122A 130      .WORD 8
      10D9 122C 131      .WORD SYIS_NODENAME
0000123A' 122E 132      .LONG NODENAME
00000000 1232 133      .LONG 0
00000000 1236 134      .LONG 0
00000000 00000000 123A 135 NODENAME:
      123A 136      .QUAD 0
      1242 137
      1242 138      .ALIGN LONG
      0000026D' 1244 139 LOCK_RANGE: ; Address range to lock into working set
000003BA' 1244 140      .LONG NON_PAGED_BEGIN
      1248 141      .LONG NON_PAGED_END
      124C 142
      124C 143 OUTFAB: $FAB FNM=<IOCOLLECT>,-
      124C 144 DNM=<IOCOLLECT.DAT>,-
      124C 145 ALQ=500,-
      124C 146 DEQ=500,-
      124C 147 FAC=<PUT,TRN>,-
      124C 148 FOP=<CBT, MXV, TEF, SQO, NEF, RWO, RWC>,-
      124C 149 MRS=OUTBUFSIZ,-
      124C 150 BLS=OUTBUFSIZ,-
      124C 151 ORG=SEQ,-
      124C 152 RFM=FIX
      129C 153 OUTRAB: $RAB FAB=OUTFAB,-
      129C 154 RAC=SEQ,-
      129C 155 MBC=<OUTBUFSIZ/512>,-
      129C 156 MBF=3,-
      129C 157 ROP=<WBH>,-
      129C 158 RBF=OUTBUF,-
      129C 159 RSZ=OUTBUFSIZ

```

```

12E0 161 .SBTTL IOCOLLECT - I/O PERFORMANCE DATA COLLECTION CONTROL
12E0 162 :++
12E0 163 : FUNCTIONAL DESCRIPTION:
12E0 164 : THIS IS THE MAIN ROUTINE RESPONSIBLE FOR
12E0 165 : CONTROLLING I/O PERFORMANCE DATA COLLECTION
12E0 166
12E0 167 : CALLING SEQUENCE:
12E0 168 : CALLS/CALLG IOCOLLECT
12E0 169
12E0 170 : INPUT PARAMETERS:
12E0 171
12E0 172 : 04(AP) = ADDRESS OF NUMBER OF BUFFERS OF DATA TO COLLECT.
12E0 173 : 08(AP) = ADDRESS OF NUMBER OF SECONDS TILL END OF DATA COLLECTION.
12E0 174 : 12(AP) = ADDRESS OF DEVICE CLASS.
12E0 175 : 16(AP) = ADDRESS OF DEVICE TYPE.
12E0 176 : 20(AP) = ADDRESS OF 'AND' MASK FOR PACKET STATUS WORD.
12E0 177 : 24(AP) = ADDRESS OF 'XOR' MASK FOR PACKET STATUS WORD.
12E0 178 : 28(AP) = ADDRESS OF QUADWORD FUNCTION MASK.
12E0 179
12E0 180 : IMPLICIT INPUTS:
12E0 181 : NONE
12E0 182
12E0 183 : OUTPUT PARAMETERS:
12E0 184 : NONE
12E0 185
12E0 186 : IMPLICIT OUTPUTS:
12E0 187 : I/O DATA COLLECTED IN FILE
12E0 188
12E0 189 : COMPLETION CODES:
12E0 190 : $$$ NORMAL
12E0 191 : Numerous error codes from other facilities
12E0 192
12E0 193 : SIDE EFFECTS:
12E0 194 : NONE
12E0 195
12E0 196 :--
12E0 197

```

```

00000000 198 .PSECT CODE, LONG, RD, NOWRT, EXE
0000 199
0000 0000 200 .ENTRY IOCOLLECT, ^M<>
0002 201
0002 202 $CREATE FAB=OUTFAB ;CREATE DATA FILE
10 50 E9 000F 203 BLBC R0, 10$
0012 204 $CONNECT RAB=OUTRAB
03 50 E8 001F 205 BLBS R0, 10$
00FD 31 0022 206 10$: BRW 30$
0025 207 15$:
0025 208
0025 209

```

```

0025 210 : INITIALIZE DATA COLLECTION PARAMETERS
0025 211 :
0025 212 :

```

```

50 00 9C 0224'CF 8C D5 0025 213 TSTL (AP)+ ;POINT TO ADDRESS OF FIRST ARGUMENT
00989680 9C D0 0027 214 MOVL @ (AP)+, W^NUMBUF ;SET NUMBER OF BUFFERS OF DATA TO COLLECT
0209'CF 51 CE 002C 215 EMUL #1000*1000*10, @ (AP)+, #0, R0 ;CALCULATE DELTA TIME IN 100 NANoseconds
0205'CF 50 CE 003A 216 MNEGL R1, W^DELTAT+4 ;SET HIGH ORDER TIME
003A 217 MNEGL R0, W^DELTAT ;SET LOW ORDER TIME

```



```

0209'CF 00 D9 003F 218 SBWC #0,W^DELTAT+4 ;
0204'CF 9C 90 0044 219 MOVB @ (AP)+,W^DCLASS ;SET DEVICE CLASS
020D'CF 9C 90 0049 220 MOVB @ (AP)+,W^DVTYPE ;SET DEVICE TYPE
0000'CF 9C B2 004E 221 MCOMW @ (AP)+,W^ANDMSK ;SET 'AND' MASK
1228'CF 9C B0 0053 222 MOVW @ (AP)+,W^XORMSK ;SET 'XOR' MASK
020F'CF 9C 7D 0058 223 MOVQ @ (AP)+,W^FUNCTN ;SET FUNCTION MASK
03 50 E8 005D 224 $LKWSET,S LOCK-RANGE ; Lock high IPL code into working set
00AE 31 0071 225 BLBS RO,16$
0074 226 BRW 30$
0074 227 16$:
0074 228 :
0074 229 : Collect UCB/device name mappings
0074 230 :
1000 8F FF 8F 6E 00 2C 0074 231 MOVCS #0,(SP),#^CO,#OUTBUFSIZ,W^OUTBUF ; Init buffer
0228'CF 007C
022C'CF DF 007F 232 PUSHAL W^OUTBUF+4 ; Address of device name counter
7E 020D'CF 9A 0083 233 MOVZBL W^DVTYPE,-(SP) ; Device type mask
7E 0204'CF 9A 0088 234 MOVZBL W^DCLASS,-(SP) ; Device class mask
0000FE0 8F DD 008D 235 PUSHL #<OUTBUFSIZ-8-12-12> ; Size of buffer
0230'CF 9F 0093 236 ; (Save room for time stamp & node ID)
05 DD 0097 237 PUSHAB W^OUTBUF+8 ; Address of buffer
05 DD 0099 238 PUSHL #5 ; Argument count
5E 18 C0 00A8 239 $CMKRNLS GET_DEVICE_INFO,(SP) ; Get UCB/device name mappings
74 50 E9 00AB 241 ADDL2 #<<5+T>*4>,SP ; Pop arguments
52 022C'CF D0 00AE 242 BLBC RO,30$
00B3 243 MOVL W^OUTBUF+4,R2 ; Get device count
00B3 244 ; (Device entry =
00B3 245 ; UCB addr + 16 char name +
00B3 246 ; quadword DEVCHAR +
00B3 247 ; byte DEVCLASS + byte DEVTYPE +
00B3 248 ; word filler)
52 52 20 C4 00B3 248 MULL2 #<4+16+8+1+1+2>,R2 ; Compute next available
52 0230'CF42 9E 00B6 249 MOVAB W^OUTBUF+8[R2],R2 ; byte in buffer
82 02 02 CE 00BC 250 MNEGL #2,(R2)+ ; Insert record type
82 0000000'GF 7D 00BF 251 MOVQ G^EXE$GQ,SYSIME,(R2)+ ; Insert starting time stamp
00C6 252 $GETSYIW,S ITMLST=SYI_LIST ; Get node identification
42 50 E9 00DD 253 BLBC RO,30$
82 03 CE 00E0 254 MNEGL #3,(R2)+ ; Insert record type
62 123A'CF 7D 00E3 255 MOVQ W^NODENAME,(R2) ; Move node name to buffer
08 12 00E8 256 BNEQ 20$ ; If no node name,
62 20202020 20202020 8F 7D 00EA 257 MOVQ #^A\ \,(R2) ; return blanks instead of 0
00F5 258 20$:
00F5 259 $PUT RAB=OUTRAB
1D 50 E9 0102 260 BLBC RO,30$
0105 261 $FLUSH RAB=OUTRAB
0D 50 E9 0112 262 BLBC RO,30$
0115 263
0115 264 $CMEXEC,S W^DATA_COLLECT ; COLLECT I/O PERFORMANCE DATA
0122 265 BLBC RO,30$ ; IF LBC DATA COLLECTION FAILURE
50 DD 0122 266 30$: PUSHL RO ; SAVE FINAL COLLECTION STATUS
0124 267 $CLOSE FAB=OUTFAB ; Close collection file
50 8ED0 0131 268 POPL RO ; RESTORE FINAL COLLECTION STATUS
04 0134 269 RET ;

```

```

0135 271 .SBTTL DATA_COLLECT - COLLECT I/O PERFORMANCE DATA
0135 272 :++
0135 273 : FUNCTIONAL DESCRIPTION:
0135 274 : THIS ROUTINE RUNS IN EXECUTIVE MODE AND COLLECTS DATA UNTIL EITHER A
0135 275 : SPECIFIED NUMBER OF DATA BUFFERS HAVE BEEN COLLECTED OR UNTIL A
0135 276 : SPECIFIED AMOUNT OF TIME HAS ELAPSED.
0135 277 :
0135 278 : INPUT PARAMETERS:
0135 279 : NONE
0135 280 :
0135 281 : IMPLICIT INPUTS:
0135 282 : PARAMETERS SUPPLIED TO/SAVED BY COLLECT
0135 283 :
0135 284 : OUTPUT PARAMETERS:
0135 285 : NONE
0135 286 :
0135 287 : IMPLICIT OUTPUTS:
0135 288 : I/O DATA FILE
0135 289 :
0135 290 :--
0135 291 :
0135 292 DATA_COLLECT:
07FC 0135 293 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
0137 294 $CMKRNL_S W^ALLOCATE ;ALLOCATE COLLECTION DATA STRUCTURE
0144 295 BLBS RO,5$ ;IF LBS SUCCESSFUL ALLOCATION
0147 296 RET ;
0148 297 :
0148 298 :
0148 299 : RESIDENT COLLECTION DATA STRUCTURE ALLOCATED
0148 300 :
0148 301 :
1000 8F FF 8F 6E 00 2C 0148 302 5$: PUSH R #^M<R2,R3,R4,R5> ; Init buffer
0228'CF 0152 303 MOVCS #0,(SP),#^C0,#OUTBUFSIZ,W^OUTBUF
53 0228'CF 9E 0155 304 POPR #^M<R2,R3,R4,R5>
59 1000 C3 9E 0157 305 MOVAB W^OUTBUF,R3 ;GET ADDRESS OF OUTPUT BUFFER
5A 14 D0 015C 306 MOVAB OUTBUFSIZ(R3),R9 ;GET ENDING ADDRESS OF OUTPUT BUFFER
03 50 E8 0161 307 MOVL #FLUSHCOUNT,R10 ;Set buffer flush control counter
00CF 31 0164 308 $SETIMR_S #2,W^DELTAT,TIMAST ;SET TIMER FOR DATA COLLECTION PERIOD
0179 309 BLBS RO,10$ ;IF LBS SUCCESSFUL
017C 310 BRW 80$
017F 311 10$: $HIBER_S ;HIBERNATE WHILE WAITING FOR BUFFER
0186 312 BBC #1,W^FLAGS,20$ ;IF CLR, SPECIFIED TIME NOT EXPIRED
018C 313 BRW 60$
018F 314 20$: $CMKRNL_S W^COPY DATA ;COPY DATA TO INTERNAL BUFFER
019C 315 BBC #0,W^FLAGS,25$ ;IF CLR, NO DATA OVERRUN
01A2 316 BRW 70$
01A5 317 25$: BLBC RO,10$ ;IF LBS NO DATA TO COPY
0004'CF 9E 01A8 318 MOVAB W^COPBUF,R8 ;GET ADDRESS OF TEMPORARY BUFFER
57 0D A8 9E 01AD 319 MOVAB PBH$K_START(R8),R7 ;GET ADDRESS OF START OF DATA
0B A8 B7 01B1 320 30$: DECB PBH$W_MSGCNT(R8) ;ANY MORE MESSAGES IN BUFFER?
D9 19 01B4 321 BLSS 20$ ;IF LSS NO
50 67 9A 01B6 322 40$: MOVZBL PIB$B_TYPE(R7),RO ;GET MESSAGE TYPE
56 021F'CF40 9A 01B9 323 MOVZBL W^MSG[EN[R0],R6 ;GET LENGTH OF MESSAGE
51 53 56 C1 01BF 324 ADDL3 R6,R3,R1 ;CALCULATE ADDRESS OF END OF MESSAGE
59 51 D1 01C3 325 CMPL R1,R9 ;BEYOND END OF BUFFER?
3A 1A 01C6 326 BGTRU 50$ ;IF GTRU YES

```

	50	00	91	01C8	327	CMPB	#PIB\$K_SRQ,R0	; START I/O REQUEST?		
		2C	12	01CB	328	BNEQ	45\$; IF NEQ NO		
01	A7	1F	01	A7	83	01CD	329	SUBB3	PIB\$B_SRQ_PRI(R7),#31,PIB\$B_SRQ_PRI(R7); CALCULATE REAL PRIORITY	
				01D3	330	ASSUME	PIB\$B_SRQ_ACCESS+2+2 EQ PIB\$K_SRQ_SIZE			
		1E	A7	B4	01D3	331	CLRW	PIB\$B_SRQ_ACCESS+2(R7) ; Pre-set for no disk geometry		
	55	14	A7	D0	01D6	332	MOVL	PIB\$L_SRQ_UCB(R7),R5 ; R5 <= UCB address		
	40	A5	01	91	01DA	333	CMPB	#DC\$_DISK,UCB\$B_DEVCLASS(R5) ; Is device a disk ?		
			19	12	01DE	334	BNEQ	45\$; BR if not		
		44	A5	B0	01E0	335	MOVW	UCB\$L_DEVDEPEND(R5),- ; Pass geometry for conversion		
		1E	A7		01E3	336		PIB\$B_SRQ_ACCESS+2(R7)		
			00	E1	01E5	337	BBC	#DEV\$0_CLO,- ; Device available cluster-wide ?		
	05	3C	A5		01E7	338		UCB\$L_DEVCHAR2(R5),42\$; (i.e., DSA or MSCP emulated disk)		
01	A7	40	8F	88	01EA	339	BISB2	#^X40,PIB\$B_SRQ_PRI(R7) ; Yes - flag as such for data reduction		
			02	E1	01EF	340	BBC	#UCB\$V_NOCNVRT,= ; Is media address in		
	05	68	A5		01F1	341		UCB\$W_DEVSTS(R5),45\$; sector/track/cylinder format ?		
01	A7	80	8F	88	01F4	342	BISB2	#^X80,PIB\$B_SRQ_PRI(R7) ; No - flag as such for data reduction		
63	67	56	28	01F9	343	45\$:	MOVW	R6,(R7),(R3) ; MOVE MESSAGE TO OUTPUT BUFFER		
	57	56	C0	01FD	344	ADDL	R6,R7 ; UPDATE ADDRESS IN TEMPORARY BUFFER			
		AF	11	0200	345	BRB	30\$;			
52	0228	'CF	9E	0202	346	50\$:	MOVAB	W^OUTBUF,R2 ; GET ADDRESS OF OUTPUT BUFFER		
				0207	347	\$PUT	RAB=OUTRAB ; Output buffer to file			
	37	50	E9	0214	348	BLBC	R0,80\$;			
	13	5A	F5	0217	349	SOBGTR	R10,55\$;			
				021A	350	\$FLUSH	RAB=OUTRAB ;			
	24	50	E9	0227	351	BLBC	R0,80\$;			
	5A	14	D0	022A	352	MOVL	#FLUSHCOUNT,R10 ; Reset buffer flush control counter			
		3C	BB	022D	353	55\$:	PUSHR	#^M<R2,R3,R4,R5> ; Re-init buffer		
1000	8F	FF	8F	6E	00	2C	022F	354	MOVCS	#0,(SP),#^CO,#OUTBUFSIZ,W^OUTBUF ;
				0237	355	POPR	#^M<R2,R3,R4,R5> ;			
		53	52	D0	023C	356	MOVL	R2,R3 ; SET STARTING ADDRESS OF OUTPUT BUFFER		
	26	0224	'CF	F5	023F	357	SOBGTR	W^NUMBUF,90\$; ALL BUFFERS COLLECTED?		
	50	00	'3C	0244	358	60\$:	MOVZWL	S^#SS\$ _NORMAL,R0 ; SET NORMAL COMPLETION STATUS		
		05	11	0247	359	BRB	80\$;			
50	0000	'8F	3C	0249	360	70\$:	MOVZWL	#SS\$ _BUFFEROVF,R0 ; SET BUFFER OVERFLOW STATUS		
		50	DD	024E	361	80\$:	PUSHL	R0 ; SAVE FINAL COMPLETION STATUS		
				0250	362	\$CMKRNLS	W^DEALLOCATE ; DEALLOCATE DATA COLLECTION STRUCTURE			
				025D	363	\$CANTIM	_S ; CANCEL TIMER			
	50	8ED0		0266	364	POPL	_R0 ; RETRIEVE FINAL COMPLETION STATUS			
		04		0269	365	RET				
	FF49	31		026A	366	90\$:	BRW	40\$;		
				026D	367					
				026D	368	*****				
				026D	369	;	BEGIN - Non-pageable code *			
				026D	370	*****				
	0000026D			026D	371	NON_PAGED_BEGIN = .				
				026D	372					

```

026D 374 .SBTTL ALLOCATE - ALLOCATE DATA COLLECTION STRUCTURE
026D 375 :++
026D 376 : FUNCTIONAL DESCRIPTION:
026D 377 : THIS ROUTINE EXECUTES IN KERNEL MODE AND IS CALLED TO ESTABLISH THE
026D 378 : DATA STRUCTURES NECESSARY TO TURN ON I/O PERFORMANCE DATA COLLECTION.
026D 379 :
026D 380 : INPUT PARAMETERS:
026D 381 : NONE
026D 382 :
026D 383 : IMPLICIT INPUTS:
026D 384 : NUMBER OF BUFFERS TO ALLOCATE
026D 385 :
026D 386 : OUTPUT PARAMETERS:
026D 387 : NONE
026D 388 :
026D 389 : IMPLICIT OUTPUTS:
026D 390 : RESIDENT DATA COLLECTION STRUCTURE ALLOCATED & INITIALIZED
026D 391 :
026D 392 :--
026D 393 :
026D 394 ALLOCATE:
026D 395 .WORD ^M<R2,R3,R4,R5,R6>
026F 396 BSBW CLEANUP ABORT ;CLEANUP FROM ANY PREVIOUS PROCESS ABORT
54 00000000'EF 007C 0272 397 MOVZBL #NUMDATBUF,R6 ;SET NUMBER OF DATA BUFFERS TO ALLOCATE
0275 398 MOVL SCH$GL_CURPCB,R4 ;GET CURRENT PROCESS PCB ADDRESS
027C 399 SETIPL #IPL$ ASTDEL ;SET IPL TO AST DELIVERY LEVEL
027F 400 MOVZBL #PDB$K_LENGTH,R1 ;GET SIZE OF PERFORMANCE DATA BLOCK
0282 401 JSB EXE$ALONONPAGED ;ALLOCATED NONPAGED MEMORY
0288 402 MOVL R2,R5 ;SAVE ADDRESS OF PERFORMANCE DATA BLOCK
028B 403 BLBC R0,30$ ;IF LBC ALLOCATION FAILURE
028E 404 MOVAB PDB$K_FREEFL(R5),PDB$K_FREEFL(R5) ;INITIALIZE FREE BUFFER LISTHEAD
0291 405 MOVAB PDB$K_FREEFL(R5),PDB$K_FREEBL(R5) ;
0295 406 MOVW R1,PDB$W_SIZE(R5) ;SET SIZE OF DATA STRUCTURE
0299 407 MOVB #DYN$C_PDB,PDB$B_TYPE(R5) ;SET DATA STRUCTURE TYPE
029D 408 CLRB PDB$B_OVERRUN(R5) ;CLEAR OVERRUN INDICATOR
02A0 409 MOVAB PDB$K_FILLFL(R5),PDB$K_FILLFL(R5) ;INITIALIZE FILLED BUFFER LISTHEAD
02A5 410 MOVAB PDB$K_FILLFL(R5),PDB$K_FILLBL(R5) ;
02AA 411 MOVL PCB$K_PID(R4),PDB$K_PID(R5) ;SET DATA COLLECTION PROCESS PID
02AF 412 CLRL PDB$K_CURBUF(R5) ;CLEAR ADDRESS OF CURRENT BUFFER
02B2 413 MOVB W^DCLASS,PDB$B_DEVCLASS(R5) ;SET DEVICE CLASS MATCH
02B8 414 MOVB W^DVTYPE,PDB$B_DEVTYPE(R5) ;SET DEVICE TYPE MATCH
02BE 415 MOVW W^ANDMSK,PDB$W_ANDM(R5) ;SET 'AND' MASK
02C4 416 MOVW W^XORMSK,PDB$W_XORM(R5) ;SET 'XOR' MASK
02CA 417 CLRW PDB$W_BUFcnt(R5) ;CLEAR FILLED BUFFER COUNT
02CD 418 MOVQ W^FUNCTN,PDB$Q_FUNC(R5) ;SET SELECTION FUNCTION MASK
02D3 419 10$: MOVZWL #DATBUFSIZ,R1 ;SET SIZE OF REQUIRED BUFFER
02D8 420 JSB EXE$ALONONPAGED ;ALLOCATE DATA COLLECTION BUFFER
02DE 421 BLBC R0,20$ ;IF LBC ALLOCATION FAILURE
02E1 422 MOVW R1,PB$W_SIZE(R2) ;SET SIZE OF DATA BUFFER
02E5 423 MOVB #DYN$C_PBH,PB$B_TYPE(R2) ;SET DATA STRUCTURE TYPE
02E9 424 CLRW PB$W_MSGCNT(R2) ;CLEAR COUNT OF COMPLETED MESSAGES
02EC 425 INSQUE (R2),PDB$K_FREEBL(R5) ;INSERT BUFFER IN FREE BUFFER LIST
02F0 426 SOBGTR R6,10$ ;ALL BUFFERS ALLOCATED?
02F3 427 SETIPL #IPL$ PERFMON ;DISABLE ALL SOFTWARE INTERRUPTS
02F6 428 TSTL PMS$GC_IOPMPDB ;PERFORMANCE DATA BLOCK ADDRESS ESTABLISHED?
02FC 429 BNEQ 20$ ;IF NEQ YES
02FE 430 MOVAB (R5),PMS$GL_IOPMPDB ;SET ADDRESS OF PERFORMANCE DATA BLOCK

```

```

50 00' 3C 0305 431 MOVZWL S^#SS$ _NORMAL,R0 ;SET NORMAL COMPLETION STATUS
    08 11 0308 432 BRB 40$ ;
50 008C 30 030A 433 20$: BSBW DELETE ;DELETE ALLOCATED STRUCTURE
    '8F 3C 030D 434 30$: MOVZWL #SS$ _INSFMEM,R0 ;SET INSUFFICIENT MEMORY STATUS
    04 0312 435 40$: SETIPL #0 ;ALLOW ALL INTERRUPTS
    0315 436 RET ;
    0316 437 ;
    0316 438 ;Local subroutine to cleanup from
    0316 439 ;any previous process abort that
    0316 440 ;left the PDB allocated
    0316 441 CLEANUP_ABORT:
    0316 442 SETIPL #IPL$ PERFORMON ;DISABLE ALL SOFTWARE INTERRUPTS
55 UG000000'GF D0 0319 443 MOVL G^PMS$GL_IOPFMPDB,R5 ;GET ADDRESS OF PERFORMANCE DATA BLOCK
    1B 13 0320 444 BEQL 10$ ;IF EQL NONE
    21 0A A5 91 0322 445 CMPB PDB$B_TYPE(R5),#DYN$C_PDB ;IS STRUCTURE A PDB ?
    15 12 0326 446 BNEQ 10$ ;BR IF NOT
    50 20 A5 D0 0328 447 MOVL PDB$L_PID(R5),R0 ;GET PID OF OWNER
    00000000'GF 16 032C 448 JSB G^EXE$IPIID_TO_PCB ;DOES PROCESS EXIST ?
    09 12 0332 449 BNEQ 10$ ;BR IF YES - DON'T DELETE PDB !
    00000000'GF D4 0334 450 CLRL G^PMS$GL_IOPFMPDB ;CLEAR ADDRESS OF PERFORMANCE DATA BLOCK
    005C 30 033A 451 BSBW DELETE ;DELETE PERFORMANCE DATA STRUCTURE
    033D 452 10$: SETIPL #0 ;ALLOW ALL INTERRUPTS
    05 0340 453 RSB
  
```

```

0341 455 .SBTTL COPY_DATA - COPY DATA TO TEMPORARY BUFFER
0341 456 :++
0341 457 : FUNCTIONAL DESCRIPTION:
0341 458 : THIS ROUTINE EXECUTES IN KERNEL MODE AND IS CALLED TO COPY THE NEXT
0341 459 : BUFFER FROM THE FILLED BUFFER POOL INTO A TEMPORARY BUFFER.
0341 460 :
0341 461 : INPUT PARAMETERS:
0341 462 : NONE
0341 463 :
0341 464 : IMPLICIT INPUTS:
0341 465 : TEMPORARY BUFFER TO COPY DATA INTO.
0341 466 :
0341 467 : OUTPUT PARAMETERS:
0341 468 :
0341 469 : RO LOW BIT CLEAR INDICATES NO DATA AVAILABLE.
0341 470 :
0341 471 : RO LOW BIT SET INDICATES DATA HAS BEEN COPIED INTO TEMPORARY BUFFER.
0341 472 :
0341 473 : IMPLICIT OUTPUTS:
0341 474 : IF DATA IS AVAILABLE, TEMPORARY BUFFER CONTAINS DATA
0341 475 :
0341 476 :--
0341 477 :
0341 478 COPY_DATA:
0341 479 .WORD *M<R2,R3,R4,R5,R6,R7>
0343 480 CLR  RO
0345 481 MOVAB W^COPBUF,R1 :ASSUME NO BUFFERS ARE FULL OF DATA
034A 482 SETIPL #IPL$ASTDEL :GET ADDRESS OF TEMPORARY BUFFER
034D 483 :RAISE PRIORITY TO AST DELIVERY LEVEL
034D 484 ASSUME DATBUFSIZ LE 512
034D 485
034D 486 TSTB (R1) :MAKE SURE TEMPORARY BUFFER IS IN MEMORY
034F 487 TSTB DATBUFSIZ-1(R1)
0353 488 SETIPL #IPL$SYNCH :RAISE IPL TO SYNCH
56 00000000'EF D0 0356 489 MOVL PMS$G[_IOPFMPDB,R6] :GET ADDRESS OF PERFORMANCE DATA BLOCK
57 0C B6 0F 035D 490 REMQUE @PDB$_FILLFL(R6),R7 :GET ADDRESS OF NEXT FILLED BUFFER
61 67 0200 8F 28 0361 491 BVS 10$ :IF VS NO FILLED BUFFER AVAILABLE
04 B6 0B A7 B4 0363 492 MOVC #DATBUFSIZ,(R7),(R1) :COPY DATA TO TEMPORARY BUFFER
04 B6 0B A7 B4 0369 493 CLRW PBH$W MSGCNT(R7) :CLEAR COMPLETED MESSAGE COUNT
04 B6 0B A7 B4 036C 494 INSQUE (R7),@PDB$_FREEBL(R6) :INSERT BUFFER IN FREE LIST
04 B6 0B A7 B4 036E 495 DECB PDB$_BUFCNT(R6) :DECREASE NUMBER OF FILLED BUFFERS
04 B6 0B A7 B4 0370 496 INCL RO :SET SUCCESS INDICATOR
04 B6 0B A7 B4 0373 497 SETIPL #0 :ALLOW INTERRUPTS
020E'CF 0B A6 88 0375 498 BISB PDB$_OVERRUN(R6),W^FLAGS :MERGE OVERRUN STATUS IN FLAG BYTE
04 037E 499 RET

```

```

037F 501 .SBTTL DEALLOCATE - DEALLOCATE DATA COLLECTION STRUCTURE
037F 502 :++
037F 503 : FUNCTIONAL DESCRIPTION:
037F 504 : THIS ROUTINE EXECUTES IN KERNEL MODE AND IS CALLED TO DEALLOCATE THE
037F 505 : DATA STRUCTURE THAT WAS ESTABLISHED BY THE ALLOCATION ROUTINE.
037F 506 :
037F 507 : INPUT PARAMETERS:
037F 508 : NONE
037F 509 :
037F 510 : IMPLICIT INPUTS:
037F 511 : ADDRESS OF PERFORMANCE DATA BLOCK.
037F 512 :
037F 513 : OUTPUT PARAMETERS:
037F 514 : NONE
037F 515 :
037F 516 : IMPLICIT OUTPUTS:
037F 517 : THE I/O PERFORMANCE DATA BLOCK AND THE CURRENT FREE AND FILLED BUFFERS
037F 518 : ARE DEALLOCATED. THE ADDRESS OF THE PERFORMANCE DATA BLOCK IS ZEROED.
037F 519 :
037F 520 :--
037F 521 :
037F 522 DEALLOCATE:
003C 037F 523 .WORD ^M<R2,R3,R4,R5>
55 00000000'EF D0 0381 524 SETIPL #IPL$ PERFMON ;DISABLE ALL SOFTWARE INTERRUPTS
08 13 0384 525 MOVL PM$GC_IOPFMPDB,R5 ;GET ADDRESS PF PERFORMANCE DATA BLOCK
00000000'EF D4 038B 526 BEQL 10$ ;IF EQL NONE
04 10 038D 527 CLRL PM$GL_IOPFMPDB ;CLEAR ADDRESS OF PERFORMANCE DATA BLOCK
0393 528 BSBB DELETE ;DELETE PERFORMANCE DATA STRUCTURE
0395 529 10$: SETIPL #0 ;ALLOW ALL INTERRUPTS
04 0398 530 RET ;

```

```

0399 532 .SBTTL DELETE - DELETE DATA COLLECTION STRUCTURE
0399 533 :++
0399 534 : FUNCTIONAL DESCRIPTION:
0399 535 : THIS ROUTINE EXECUTES IN KERNEL MODE AND IS CALLED TO ACCOMPLISH THE
0399 536 : ACTUAL DELETION OF THE I/O PERFORMANCE DATA COLLECTION STRUCTURE.
0399 537 :
0399 538 : INPUT PARAMETERS:
0399 539 :
0399 540 : R5 = ADDRESS OF PERFORMANCE DATA BLOCK.
0399 541 :
0399 542 : INTERRUPT PRIORITY LEVEL AT IPL$_PERFMON
0399 543 :
0399 544 : IMPLICIT INPUTS:
0399 545 : NONE
0399 546 :
0399 547 : OUTPUT PARAMETERS:
0399 548 : NONE
0399 549 :
0399 550 : IMPLICIT OUTPUTS:
0399 551 : THE I/O PERFORMANCE DATA STRUCTURE IS DELETED.
0399 552 :
0399 553 :--
0399 554
0399 555 DELETE:
0399 556 SETIPL #IPL$_ASTDEL ;DELETE PERFORMANCE DATA STRUCTURE
50 00 B5 OF 039C 557 10$: REMQUE @PDB$_FREEFL(R5),R0 ;LOWER TO AST DELIVERY LEVEL
0399 558 BVS 20$ ;GET NEXT BUFFER FROM FREE LIST
0399 559 BSBB 40$ ;IF VS NONE
0399 560 BRB 10$ ;DEALLOCATE BUFFER
50 0C B5 OF 03A6 561 20$: REMQUE @PDB$_FILLFL(R5),R0 ;GET NEXT BUFFER FROM FILLED LIST
0399 562 BVS 30$ ;IF VS NONE
0399 563 BSBB 40$ ;DEALLOCATE BUFFER
0399 564 BRB 20$ ;
0399 565 30$: MOVL R5,R0 ;SET ADDRESS OF LAST BLOCK
00000000'EF 03B3 566 40$: JSB EXE$DEANONPAGED ;DEALLOCATE NONPAGED MEMORY
0399 567 RSB ;
0399 568
0399 569 :*****
0399 570 : END - Non-pageable code *
0399 571 :*****
000003BA 03BA 572 NON_PAGED_END = .
0399 573

```



```

03BA 575 .SBTTL TIMAST - TIMER AST ROUTINE
03BA 576 :++
03BA 577 : FUNCTIONAL DESCRIPTION:
03BA 578 : THIS ROUTINE IS ENTERED WHEN THE DATA COLLECTION PERIOD HAS ELAPSED.
03BA 579 :
03BA 580 : INPUT PARAMETERS:
03BA 581 : NONE
03BA 582 :
03BA 583 : IMPLICIT INPUTS:
03BA 584 : TIMER EXPIRED FLAG IN OWN STORAGE
03BA 585 :
03BA 586 : OUTPUT PARAMETERS:
03BA 587 : NONE
03BA 588 :
03BA 589 : IMPLICIT OUTPUTS:
03BA 590 : THE TIMER EXPIRED FLAG IS SET AND A WAKE SYSTEM SERVICE IS EXECUTED
03BA 591 : TO WAKE THE DATA COLLECTION PROCESS.
03BA 592 :
03BA 593 :--
03BA 594
03BA 595 TIMAST:
03BA 596 .WORD ^M<>
020E'CF 02 0000 03BC 597 BISB #2,W^FLAGS ;SET TIMER EXPIRED FLAG
03BA 598 $WAKE_S ;WAKE UP CURRENT PROCESS
03BA 599 RET ;
03BA 600
03BA 601 .END

```

IOCOLLECT
Symbol table

- I/O PERFORMANCE DATA COLLECTION F 11

16-SEP-1984 02:14:47 VAX/VMS Macro V04-00
5-SEP-1984 04:36:56 [UTIL32.SRC]IOCOLLECT.MAR;1

\$\$TAB	= 0000129C	R	02	Iostat	= 00000217	R	02
\$\$TABEND	= 000012E0	R	02	IPL\$ASTDEL	= 00000002		
\$\$TMP	= 00000400			IPL\$PERFMON	= 0000000F		
\$\$TMP1	= 00000001			IPL\$SYNCH	= 00000008		
\$\$TMP2	= 000000CF			LOCKRANGE	00001244	R	02
\$\$TMPX	= 00000009	R	03	MSGLEN	0000021F	R	02
\$\$TMPX1	= 0000000D			NODENAME	0000123A	R	02
\$\$T1	= 00000001			NON_PAGED_BEGIN	= 0000026D	R	04
ALLOCATE	0000026D	R	04	NON_PAGED_END	= 000003BA	R	04
ANDMSK	00000000	R	02	NUMBUF	00000224	R	02
BUFLen	00000002	R	02	NUMDATBUF	= 0000000B		
CLEANUP_ABORT	00000316	R	04	OUTBUF	00000228	R	02
COPBUF	00000004	R	02	OUTBUFSIZ	= 00001000		
COPY_DATA	00000341	R	04	OUTFAB	0000124C	R	02
DATA_COLLECT	00000135	R	04	OUTRAB	0000129C	R	02
DATBUFSIZ	= 00000200			PB\$B_TYPE	= 0000000A		
DC\$DISK	= 00000001			PB\$K_START	= 0000000D		
DCLASS	00000204	R	02	PB\$W_MSGCNT	= 0000000B		
DEALLOCATE	0000037F	R	04	PB\$W_SIZE	= 00000008		
DELETE	00000399	R	04	PC\$SL_PID	= 00000060		
DELTAT	00000205	R	02	PDB\$B_DEVCLASS	= 00000024		
DEVSV_CLU	= 00000000			PDB\$B_DEVTYPE	= 00000025		
DVTYPE	0000020D	R	02	PDB\$B_OVERRUN	= 0000000B		
DYN\$C_PBH	= 00000020			PDB\$B_TYPE	= 0000000A		
DYN\$C_PDB	= 00000021			PDB\$K_LENGTH	= 00000034		
EXES\$[ONONPAGED	*****	X	04	PDB\$K_CURBUF	= 00000014		
EXES\$[EANONPAGED	*****	X	04	PDB\$K_FILLBL	= 00000010		
EXES\$[Q_SYSTIME	*****	X	04	PDB\$K_FILLFL	= 0000000C		
EXES\$[PID_TO_PCB	*****	X	04	PDB\$K_FREEBL	= 00000004		
FAB\$B_DNS	= 00000035			PDB\$K_FREEFL	= 00000000		
FAB\$B_FNS	= 00000034			PDB\$K_PID	= 00000020		
FAB\$C_BID	= 00000003			PDB\$Q_FUNC	= 0000002C		
FAB\$C_BLN	= 00000050			PDB\$W_ANDM	= 00000026		
FAB\$C_FIX	= 00000001			PDB\$W_BUFcnt	= 0000002A		
FAB\$C_SEQ	= 00000000			PDB\$W_SIZE	= 00000008		
FAB\$K_ALQ	= 00000010			PDB\$W_XORM	= 00000028		
FAB\$K_DNA	= 00000030			PIB\$B_SRQ_ACCESS	= 0000001C		
FAB\$K_FNA	= 0000002C			PIB\$B_SRQ_PRI	= 00000001		
FAB\$K_FOP	= 00000004			PIB\$B_TYPE	= 00000000		
FAB\$V_CBT	= 00000015			PIB\$K_ARQ_SIZE	= 00000010		
FAB\$V_CHAN_MODE	= 00000002			PIB\$K_EIO_SIZE	= 00000018		
FAB\$V_FILE_MODE	= 00000004			PIB\$K_ERQ_SIZE	= 00000010		
FAB\$V_LNM_MODE	= 00000000			PIB\$K_SIO_SIZE	= 00000018		
FAB\$V_MXV	= 00000001			PIB\$K_SRQ	= 00000000		
FAB\$V_NEF	= 0000000A			PIB\$K_SRQ_SIZE	= 00000020		
FAB\$V_PUT	= 00000000			PIB\$K_SRQ_UCB	= 00000014		
FAB\$V_RWC	= 0000000B			PM\$SGC_IOPFMPDB	*****	X	04
FAB\$V_RWO	= 00000007			PR\$IPC	= 00000012		
FAB\$V_SQO	= 00000006			RAB\$B_RAC	= 0000001E		
FAB\$V_TEF	= 0000001C			RAB\$C_BID	= 00000001		
FAB\$V_TRN	= 00000004			RAB\$C_BLN	= 00000044		
FAB\$W_GBC	= 00000048			RAB\$C_SEQ	= 00000000		
FLAGS	0000020E	R	02	RAB\$K_CTX	= 00000018		
FLUSHCOUNT	= 00000014			RAB\$K_ROP	= 00000004		
FUNCTN	0000020F	R	02	RAB\$V_WBH	= 0000000A		
GET_DEVICE_INFO	*****	X	04	SCH\$GE_CURPCB	*****	X	04
IOCOLLECT	00000000	RG	04	SS\$BUFFEROVF	*****	X	04

```

SS$ IN$F$MEM          ***** X 04
SS$ NORMAL           ***** X 04
SYIS NODENAME        = 000010D9
SYI CIST              0000122A R 02
SYS$CANTIM           ***** GX 04
SYS$CLOSE            ***** GX 04
SYS$CMEXEC           ***** GX 04
SYS$CMKRNL           ***** GX 04
SYS$CONNECT          ***** GX 04
SYS$CREATE            ***** GX 04
SYS$FLUSH            ***** GX 04
SYS$GETSYIW          ***** GX 04
SYS$HIBER            ***** GX 04
SYS$LKWSET           ***** GX 04
SYS$PUT              ***** GX 04
SYS$SETIMR           ***** GX 04
SYS$WAKE             ***** GX 04
TIMAS                000003BA R 04
UCBS$ DEVCLASS       = 00000040
UCBS$ DEVCHAR2       = 0000003C
UCBS$ DEVDEPEND       = 00000044
UCBS$ NOCNVRT         = 00000002
UCBS$ DEVSTS         = 00000068
XORMSR              00001228 R 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
DATA	000012E0 (4832.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
\$RMSNAM	00000016 (22.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
CODE	000003CD (973.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	47	00:00:00.10	00:00:00.51
Command processing	155	00:00:00.67	00:00:04.97
Pass 1	507	00:00:21.22	00:00:54.24
Symbol table sort	0	00:00:03.23	00:00:04.21
Pass 2	119	00:00:03.66	00:00:09.18
Symbol table output	18	00:00:00.15	00:00:00.34
Psect synopsis output	1	00:00:00.03	00:00:00.06
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	849	00:00:29.08	00:01:13.51

The working set limit was 1800 pages.
114315 bytes (224 pages) of virtual memory were used to buffer the intermediate code.
There were 110 pages of symbol table space allocated to hold 2123 non-local and 30 local symbols.
601 source lines were read in Pass 1, producing 23 object records in Pass 2.

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

! Macro library statistics !

<u>Macro library name</u>	<u>Macros defined</u>
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	9
^_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	33
TOTALS (all libraries)	42

2489 GETS were required to define 42 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:IOCOLLECT/OBJ=OBJ\$:IOCOLLECT MSRCS:IOCOLLECT/UPDATE=(ENH\$:IOCOLLECT)+EXECMLS/LIB

SE
VO



0429 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 100 terminal window screenshots, arranged in 10 rows and 10 columns. Each window shows a different system utility or diagnostic tool. The visible titles for several windows are:

- RTB MAP
- SETUSER MAP
- SEARCH MAP
- CHECKSUM LIS
- DISKMON LIS
- DCLDIET LIS
- TOCOLLECT LIS
- SEARCH LIS
- GETINFO LIS
- RTB LIS
- DISKQ LIS
- CKSMMSG LIS

The screenshots show various data formats, including lists of files, system status information, and diagnostic output. The text is small and dense, typical of a terminal window output.