Date: September 23, 1982

To: Prime Personnel

From: Glenn A. Weinberg, Donald C. Slutz

Subject: AUTOPSY, The PRIMOS Crash Dump Analysis Program

Reference: PRIME Engineering Handbook (PE-T-500), 50 Series Technical Summary (PE-T-1022), The Assembly Language Programmer's Guide (FDR 3049)

Keywords:

Abstract

When PRIMOS crashes or is manually halted, it is the responsibility of the PRIMOS engineering staff to determine the cause of the problem. So much information is kept by PRIMOS on a crash dump tape that the cause of almost any crash may be determined. On the other hand, this information may be overwhelming in its magnitude, and found by only the most knowledgeable PRIMOS wizards. In order to help mere mortals analyze PRIMOS crash dumps, a program called AUTOPSY exists. This document describes that program.

PE-T-484, Rev.1 describes AUTOPSY as of version 19.0.14.

This document is a complete rewrite of the original PE-T-484, and therefore change bars have not been used.

Table of Contents

		Page
1	**Introduction**	1
1.1	Layout of this document	1
1.2	What this document does not do	1
2	**Initializing AUTOPSY**	1
2.1	Reading a crash dump from tape	1
2.2	Reading a dump which is already on disk	2
2.3	Reading maps	3
3	**Examining the dump**	4
3.1	The RPRNT command	4
3.2	The STATUS command	4
3.3	The LOCSEARCH command	5
3.4	The CHKPRT command	5
3.5	The RESTORE command	6
4	**More advanced commands**	6
4.1	The TRACE command	6
4.2	The TTYBUF command	8
4.3	The UTBL and UTENTRY commands	9
4.4	Commands to check system consistency	9
5	**Command Reference**	10
5.1	CHKPRT – Print description of last machine check	10
5.2	CLRMAP – Clear symbol table	10
5.3	COMSEARCH – Search symbol table for common block	11
5.4	DDQB – Dump Disk Queue Blocks	11
5.5	DLCB – Dump Locate Control Blocks	11
5.6	DSEMAPHORE – Dump Semaphore	11
5.7	DUMP – Dump memory contents	11
5.8	DATE – Display date header	12
5.9	ECBSEARCH – Search symbol table for an ECB	12
5.10	FROM – Load a dump from a file	12
5.11	FSCHK – Check File System hash tables	12
5.12	HELP	12
5.13	Keyprt – Decode keys and modals	13
5.14	LBSEARCH – Search symbol table for module with given LB	13
5.15	LBNames – Search symbol table for modules with given LB	13
5.16	LOCSEARCH – Search symbol table	13
5.17	LPRNT – Display N1LOCK status	14
5.18	MAP – Read map(s)	14
5.19	OTHSEARCH – Search symbol table for Other symbol	14
5.20	PAGCHK – Check page maps for consistency	14
5.21	PAUSE – Suspend AUTOPSY session	14
5.22	PBSEARCH – Search symbol table for module with given PB	14
5.23	PDUMP – Display PCB for specified user	15
5.24	PMAP – Display page maps for given segment	15

5.25 QUIT – Exit AUTOPSY 15
5.26 RDUMP – Display register save area 15
5.27 READ – Read a dump from tape 15
5.28 RESTORE – Load a segment for VPSD 15
5.29 RPRNT – Print live registers 16
5.30 STATUS – Display status information 16
5.31 SYMBOL – Display location information for given symbols 16
5.32 TRACE – Enter the TRACE subsystem 16
5.33 TTYBUF – Display terminal buffers for given user 18
5.34 UNIT – Display contents of specified unit table entry 18
5.35 UOWNER – Display owner of specified unit table entry 18
5.36 UTBL – Display unit tale of specified user 18
5.37 UTENTRY – Display specified unit table entry 19
5.38 VPSD – Invoke VPSD subsystem 20
5.39 ! – Execute PRIMOS command 20

1 Introduction

In order to analyze a PRIMOS crash dump to determine the cause of the crash, a tremendous volume of data may have to be examined. Only a very few people at PRIME know where most of this data lives, and it is a reasonably safe bet that no one person knows where everything is. Furthermore, the data which is kept and where it is kept may vary from rev to rev. In order to facilitate analysis of crash dumps, a program called AUTOPSY has been written. This program allows the dump to be examined in a logical manner, providing information about the state of the system in terms of users, semaphores, file system, etc. in a format which is useful to humans.

1.1 Layout of this document

The description of AUTOPSY proceeds in a top-down manner, describing first the procedure for initializing AUTOPSY, continuing with commands which are useful for any dump, and then describing more complex commands which may apply only to certain classes of crash dumps. Finally a detailed commands reference is provided.

1.2 What this document does not do

It is not the purpose of this document to describe the crash dump procedure; for information on this procedure, consult the *PRIME Engineering Handbook* (PE-T-500, Rev. 1). This document also assumes at least a passing familiarity with the 50-series hardware and VPSD, the symbolic debugger. For information on the hardware, refer to PE-T-500 or the *50 Series Technical Summary* (PE-T-1022); for VPSD, see *The Assembly Language Programmer's Guide* (FDR 3059).

2 Initializing AUTOPSY

When a crash dump is to be analyzed, it may be on either of two storage media: magnetic tape or disk. The procedure for initializing AUTOPSY in each case is described in this section. (Usually, AUTOPSY may be found either in CMDNC0 as the external command X.AUTOPSY, or in the directory which contains the crash dumps as AUTOPSY.SAVE.)

2.1 Reading a crash dump from tape

In order to read a crash dump from tape, first assign the tape drive on which the tape is mounted, and then invoke AUTOPSY. AUTOPSY will respond with its version number and a prompt, for example:

```
[Autopsy 19.0.14]
>
```

PE-T-484, Rev. 1

(AUTOPSY will always prompt with ">" when it is ready for the next command.) To read the dump in from tape, use the READ command, for example:

```
Read enb.03/08.1233 1
```

"enb.03/08.1233" is the name that the crash dump file will have on disk. "1" is the magnetic tape drive number which has previously been assigned. (If the tape drive is not assigned, AUTOPSY will abort with an "MTx not assigned" error. In this case, assign the drive at that point, and restart AUTOPSY with the START command.) Note that since most crash dumps are at least 512 records long (they may currently be up to 4K records long), the directory into which the dump is read should have plenty of storage available.

When the dump has been read in, AUTOPSY will display the status of the Read command, along with information about the dump, as:

```
Done!
2048 Pages in file, 2049 Tape records, 1 were duplicated.
```

If any tape errors occurred, they will be described here.

AUTOPSY next gives you general information about the crash that has just been read in. This information is described in the next section on reading dumps from disk.

2.2 Reading a dump which is already on disk

In order to read an existing crash dump, AUTOPSY may be invoked with a treename on its command line:

```
X.AUTOPSY <treename>
```

<treename> is the name of the file containing the existing crash dump. If you have already invoked AUTOPSY (or have finished examining one dump and wish to examine another), use the FROM command:

```
From enb.03/08.1233
```

When AUTOPSY has finished reading in the dump, it prints the following information (known as the "date header"):

```
Revision of PRIMOS      Time PRIMOS was built      Time of crash

                                    MAPGEN Revision

Version: 19.0.86  02/25/82**13:01:02**17.1  82-03-08.12:33:08.Mon
Crash:   ENB.03/08.1233
CPU:     P750, microcode rev. 15

                                                Dump file name

    Processor type        Revision of microcode in processor
```

2.3 Reading maps

Many of the more advanced features of AUTOPSY can be supported only if the maps describing PRIMOS are read by AUTOPSY. In order to read the maps (there are currently two: one for ring zero, and one for ring three), use the MAP command:

```
MAP [<treename_1> [<treename_2>]]
```

If no names are given to the MAP command, it looks in the top-level directory called MAPS for files with the name

```
RING(0 3).MAP.<version>
```

where <version> is the PRIMOS revision number printed in the date header. An example with no names given follows:

```
    > MAP

Reading ring 0 map for version "19.0.86".
Done!  1444 entries in symbol table:
    761 routines, 247 common blocks, 436 other symbols.

Reading ring 3 map for version "19.0.86".
Done!  2010 entries in symbol table:
    1076 routines, 257 common blocks, 677 other symbols.
```

The symbol table now contains maps describing PRIMOS revision 19.0.86, which contains 1076 procedures, 257 common blocks, and 677 other symbols. Making further use of the information contained in the symbol table will be described later.

3 Examining the dump

Now that the dump has been read in, we must look at it to determine what went wrong. The first thing to do is to find out what user was running when the system crashed.

3.1 The RPRNT command

Determining the user running at the time of the crash is done with the RPRNT (register print) command and the LIVE (live user) option. The output from the RPRNT command might look like this:

```
> RP LIVE
User # 44                Timer = -113

PB: 12(0)/74426  SB: 6003(0)/162  LB: 12(0)/74130  XB: 27(0)/55421

FR0  064000.002516.000000.000000   0.2387724758470383350-38
FR1  000002.140714.142730.145640   0.1523948815799282360+407

              (OCT)             (DEC)
A           000002                  2
B           000000                  0
L    000002.000000             131072
E    075170.066247         2054712487
X           000000                  0
Y           000000                  0
KEYS/MODALS         014000.100177
FCODE/FADDR         104000.000064     4000(3)/4226
DTAR0-3 127700.012000 140000.012501 140006.033100 176406.033064
```

From this we can tell that user 44 (octal; all numbers in AUTOPSY are assumed to be octal) was running in segment 12 at the time of the crash. More generally useful information can be gotten with the STATUS command.

3.2 The STATUS command

The STATUS command prints information about one or more processes, including user or process ID, base registers, last wait location, and wait state. The results of a "ST 44" command would be:

```
    > STAT 44

User # 44  ALEXK  PB: 12(0)/74426 SB: 6003(0)/162 LB: 12(0)/74130
                  XB: 27(0)/55421
LEVEL : 626  PRIORITY 1 USER  *READY* WAS WAITING AT: 0/546
                    *LIVE*    *MASTER*
LOKOWN: NETLCK
ABSAVE: NONE
PCBABT: NONE
```

3.3 The LOCSEARCH command

We can find out the name of the module in which the user was executing by using the LOCSEARCH command:

```
LO 12/74426

12/74426 at LB XLGCON + 270

Routine name: XLGCON
ECB: 12/75146  PB: 12/74726  LB: 12/74536
```

Now just from two commands (RP and LO) we've discovered something very interesting: user 44 seems to have been executing part of his linkage area!

3.4 The CHKPRT command

The fact that the user was running in his linkage area makes us wonder what strange things may have been happening to the machine, so we next see if there have been any machine checks with the CHKPRT command:

```
    > CH
Last check handled:

DSWPB:      12(0)/102124
DSWRMA:     6003(0)/652
DSWSTAT:    027170.174000
DSWPARITY: 007101.003504
    Memory parity error on oap, physical page 7636 (corrected)
    SSU parity error
    RMA not incremented at error
    BMA15 at error
    BMA16 at error
    ECCC on Cache Miss
    Cache cycle for execution

Check headers:

Memory parity  at 12(0)/102124, Keys = 014000, Modals = 100177
```

Aha! There has been a machine check. On the other hand, it looks like a harmless ECCC (corrected memory parity) error. However, the error did occur in segment 12, where we were executing, so we again use the LOCSEARCH command to find out exactly where the problem occurred:

```
> LO 12/102124

Routine name: X$FLDM
ECB at 12/102230, PB=12/102124, LB=12/101626
```

PE-T-484, Rev. 1

We have just discovered that the ECCC error occurred on the first instruction of a routine called X$FLDM. It just so happens that we've found the cause of the problem, because we know from experience that P750's with microcode prior to rev 23 have a microcode bug that drives them batty if they take an ECCC during an ARGT instruction! Just to be on the safe side, however, we want to take a look at the instruction at 12/102124 to make sure it is an ARGT. We do this with the RESTORE command:

3.5 The RESTORE command

```
> RESTORE 12

$ a 102124              /* In VPSD; attach to offending location
4001/ 102124 ARGT  ?    /* We do have an ARGT; release location
$ q                     /* Return to AUTOPSY
>
```

We have now solved this crash dump. Not all crashes will be this easy to solve, but even this relatively easy crash would have been difficult without commands like RPRNT and CHKPRT.

4 More advanced commands

The commands discussed in the first section are useful for almost all dumps. This section describes some of the more advanced commands available in AUTOPSY for analyzing more complex dumps.

4.1 The TRACE command

The TRACE command enters a subsystem of AUTOPSY which allows you to examine, in detail, the stack for any user on the system. TRACE lets you retrace the steps that a process took to get where it was when the system halted. At each level of the stack, TRACE displays information such as the PB, LB, and SB of the process; the size of the stack frame; and, if maps are available, the names of both the called and calling routines. Subcommands to TRACE allow you to travel freely up and down the stack or to any particular frame on it, as well as to examine all or part of any of the frames. TRACE is invoked by:

```
Trace <user> [<address>]
```

If <address> is given, the trace begins at <address>, which must be the base address of a legitimate stack frame. Otherwise, the trace begins at the current SB of the <user>. Once

within TRACE, you can use the basic commands Father to go back down the stack, and Son to go up. For instance:

```
    > TRACE 176


Level 1:  PREVSB_
Root: 6003  SB: 6003/1652  Size: 57 words  Type: 000000 (PCL)
Keys: 034000
Call at 6(0)/142670 (QUTABT+520) SB: 6003(0)/1472 LB: 6(0)/142662

(trace)> FATHER             /* Note different prompt

Level 2:  QUTABT
Root: 6003  SB: 6003/1472  Size: 111 words  Type: 000000 (PCL)
Keys: 014000
Call at 6(0)/54227 (LOGABT+261) SB: 6003(0)/1414 LB: 6(0)/53710

(trace)> F

Level 3:  LOGABT
Root: 6003  SB: 6003/1414  Size: 45 words  Type: 000000 (PCL)
Keys: 014000
Call at 6(0)/34547 (PABORT+405) SB: 6003(0)/1364 LB: 6(0)/34266

(trace)> F

Level 4:  PABORT
Root: 6003  SB: 6003/1364  Size: 23 words  Type: 000000 (PCL)
Keys: 014000
Call at 6(0)/4371 (PRCFP (et al)+177756) SB: 6003(0)/1274
                                         LB: 6(0)/5130

(trace)> SON                /* Go back a level

Level 3:  LOGABT
Root: 6003  SB: 6003/1414  Size: 45 words  Type: 000000 (PCL)
Keys: 014000
Call at 6(0)/34547 (PABORT+405) SB: 6003(0)/1364 LB: 6(0)/34266

(trace)> F 6                /* Trace through six fathers

Level 9:  INIT$U
Root: 6003  SB: 6003/604  Size: 133 words  Type: 000000 (PCL)
Keys: 014200
Call at 15(0)/11011 (PHLOGIN (et al)+515) SB: 6003(0)/162
                                          LB: 15(0)/10564

(trace)> Q                  /* Done
```

In addition to the basic Father and Son commands, TRACE allows you to: examine the stack at the current level (STACK command), display arguments to the current procedure (ARGUMENTS command), or GOTO a specific level of the stack, among others. Full details are in the commands reference section.

PE-T-484, Rev. 1

4.2 The TTYBUF command

The TTYBUF command allows you to read the terminal input and/or output buffers for each terminal user on the system, as well as for special processes which have their own buffers. Reading user 1's terminal buffers can prove especially enlightening since disk errors and such are displayed there. The syntax of the TTYBUF command is:

```
TTybuf <user> [-<buffer>] [-CRLF] [-OCTAL]
```

If no <buffer> option is given, the user's input and output buffers are printed. Otherwise, only the specified buffers are displayed. (A complete list of the available buffers is given in the commands reference section. The most useful are -INPut, to display only the user's input buffer; -OUTput, to display only the user's output buffer; and -User_1_Message, to display user 1's message buffer.) If the -CRLF option is given, lines are displayed with newline characters as they would have appeared at the terminal; if the option is not supplied, the format is as for an ASCII dump in VPSD, except that there are no spaces between the contents of each word. If the -OCTAL option is given, control characters are shown as three-digit octal numbers preceded by an uparrow (^). The buffers are displayed in chronological order, with the most recent input or output at the bottom of the display, for instance:

```
> TT 1

Input buffer (400 bytes long) for user 1 at 7(0)/101064:
  ...... .!."#.$..A CMDNC0.X.MAIL SKIP MM.M SKIP NOW -ON END.Mail
   is back......MAX 0.SE -030882 -1230.DATE.RDY..C..C.X.MAIL SKIP
  MM.DELETE MM -RPT.M SKIP NOW -ON END.HOW ABOUT NOW....MAX ALL.PH
   LM.B>PH_BOOT.RDY.M ALL NOW.LOGIN PLEASE.......????????????DATE.


Output buffer (600 bytes long) for user 1 at 7(0)/0:
  deleted...OK, OK, OK, PHANTOM is user 106..OK, OK 12:32:03  7.63
  6  6.163  level 2+..OK, ...*** SYSTEM (user 106 on ENB) at 12:32
  ...***EVERY THING IS O.K.........LINE MONITOR STARTED....OK, ..Ph
  antom 106: Normal logout at 12:32..Time used: 00h 00m connect, 0
  0m 02s CPU, 00m 02s I/O.....*** BATCH_SERVICE (user 101 on ENB)
   at 12:32...Monitor continued....08 Mar 82 12:32:56 Monday..OK,


> TT -USER_1_MESSAGE -CRLF

User 1 message buffer (600 bytes long) at 7(0)/132667:
    AN rev 1.0] started.

    *** SYSTEM (user 102 on ENB) at 12:31
    ***ERROR IN START PHANTOM***.......PHANTOM NOT STARTED..

    *** FTP (user 105 on ENB) at 12:31
    12.31.19: FTS Server - FTP started up on Monday, March 8, 1982

    *** SYSTEM (user 106 on ENB) at 12:32
    ***EVERY THING IS O.K........LINE MONITOR STARTED.

    *** BATCH_SERVICE (user 101 on ENB) at 12:32
```

```
Monitor continued.
```

4.3 The UTBL and UTENTRY commands

The UTBL command enables you to examine the contents of a unit table for any user on the system. A unit table contains a list of offsets into the unit table entry common. Each offset defines the location of a unit table entry, and each unit table entry describes a unit that the user holds open. The UTENTRY command allows you to display the contents of individual unit table entries. The format of the commands is:

```
UTbl    <user>
UTEntry <user> <unit>
```

Examples of both commands follow:

```
> UTBL 1
Unit table for user 1 (at 10(0)/1071):
System unit at 1
Unit 175 at 551
Attach points:  current at 25, home at 51, initial at 75

> UTE 1 175
Entry for unit 175 (at 40(0)/1670 [551]):
Status:    110002 (Type: SAM file  Mode: w  modified  no close)
RWlock:    5 (none)
Accesses:  2 (W)
Ldev:      0
BRA:       32146  CRA: 32146
Position:  word 234 in relative record 0
Entry at:  word 533 in directory with BRA 4622
Dir block at:   41
Hash thread:    0
```

4.4 Commands to check system consistency

Two commands, each ending in "CHK," cause AUTOPSY to automatically check certain PRIMOS databases for consistency. The FSCHK command checks the file system hash tables for circular or orphaned threads, while the PAGCHK command checks the memory maps (MMAP, HMAP, and LMAP). Messages will be printed if any problems are found.

PE-T-484, Rev. 1

5 Command Reference

This section provides a detailed commands reference guide for AUTOPSY. We hope to keep it up-to-date by providing addenda as new commands are added.

The minimum abbreviation for a command is shown in uppercase in the command name. All commands are left-substring abbreviatable. Only the first four characters are checked.

5.1 CHKPRT – Print description of last machine check

```
CHkprt
```

The CHKPRT command prints a detailed human-readable description of the last machine check which occurred, plus the check headers of any other checks. An example of the output for a recovered memory parity error (ECCC) follows:

```
> ch
Last check handled:

DSWPB:      12(0)/102124
DSWRMA:     6003(0)/652
DSWSTAT:    027170.174000
DSWPARITY: 007101.003504
    Memory parity error on oap, physical page 7636 (corrected)
    SSU parity error
    RMA not incremented at error
    BMA15 at error
    BMA16 at error
    ECCC on Cache Miss
    Cache cycle for execution

Check headers:

Memory parity  at 12(0)/102124, Keys = 014000, Modals = 100177
```

Other information may be displayed according to the nature of the check and the machine on which it occurred. For instance, the 850 communications area status word is displayed for checks occurring on P850's.

5.2 CLRMAP – Clear symbol table

```
Clrmap
```

The CLRMAP command clears the symbol table databases so that new maps may be read in. Note that once even one map has been read in, the MAP command may not be given without a treename unless CLRMAP is given first.

5.3 COMSEARCH – Search symbol table for common block

COmsearch <address>

The COMSEARCH command searches the symbol table for a common block at the specified address (wordnumber/segmentnumber). If one is found, its name and address are displayed.

5.4 DDQB – Dump Disk Queue Blocks

DDqb [<start> [<end>]] [-FREE] [-USED] [-MeTeRs]

Dumps the disk queue control blocks. If no options are specified, all blocks are displayed. Otherwise, the arguments and options act as follows:

- o <start> only: Dumps QCB number <start> (first is zero).
- o <start> <end>: Dumps QCBs <start> to <end>, inclusive.
- o -FREE: Displays blocks currently not in use.
- o -USED: Displays blocks currently in use.
- o -METERS: Displays DISKIO meters in addition to QCBs.

5.5 DLCB – Dump Locate Control Blocks

DLcb [<start> [<end>]] [-FREE] [-Hash_Table]

Dumps LOCATE control blocks. If no options are specified, all blocks are displayed. Otherwise, the arguments and options act as follows:

- o <start> only: Dumps LCB number <start> (first is zero).
- o <start> <end>: Dumps LCBs <start> to <end>, inclusive.
- o -LRU_LIST: Displays blocks in most-recently to least-recently used order.
- o -HASH_TABLE: Displays LOCATE hash table.

5.6 DSEMAPHORE – Dump Semaphore

DSemaphore <address> [<user>]

Dumps the semaphore at the specified location. If <user> is omitted, the current user is used.

5.7 DUMP – Dump memory contents

Dump <address> <last_word> [<user>]

<address> is the starting address of the area to be displayed. <last_word> is the last word number to be displayed. (DUMP may not cross segment boundaries.) If the segment to be dumped belongs to a user other than the current user, <user> specifies to which user it

belongs. The area is always dumped in octal, 8 words per line.

5.8 DATE – Display date header

DAte

The DATE command displays the date header which is printed when a file is read with the FROM command. The names of any maps are also printed.

5.9 ECBSEARCH – Search symbol table for an ECB

Ecbsearch <address>

The symbol table is scanned for an ECB at or near the given address, and if one is found its name and location are displayed.

5.10 FROM – Load a dump from a file

From <treename>

Reads the dump in file <treename> into memory and initializes the AUTOPSY databases. When the dump has been read in, displays the date header, which gives information on the revision of PRIMOS running at the time of the crash, when that system was loaded, and when the crash occurred.

5.11 FSCHK – Check File System hash tables

FSchk

Checks the file system hash tables for consistency. Prints messages describing the nature of the problem and where it occurs if circular or orphaned hash threads are found.

5.12 HELP

Help [<topic>]

If a <topic> is specified and it exists in the HELP database, detailed information on the topic is displayed at the terminal. Legal topics are all commands plus "NEW," which describes recent changes to AUTOPSY. If no <topic> is given, a brief description of all commands is given.

5.13 Keyprt – Decode keys and modals

Keyprt <keys> <modals>

Displays the decoded versions of the specified key and modal values. An example follows:

```
> keyp 14201 100176

Breakdown of Keys 014201:
    Processor mode:          64V
    C-bit:                   off
    D-bit:                   off
    L-bit:                   off
    Floating exceptions:     allowed
    Integer exceptions:      allowed
    Condition codes:         LT/NE
    In dispatcher bit:       off
    Save done bit:           on

Breakdown of Modals 100176:
    Interrupts:              enabled
    Vectored interrupt bit:  off
    Mapped I/O:              enabled
    Process exchange:        enabled
    Segmented mode:          enabled
    Current register file:   3
    Machine check mode:      All but ECCC checks
```

5.14 LBSEARCH – Search symbol table for module with given LB

LBsearch <address>

Searches the symbol table for a module with the specified LB, and returns its name, ECB address, PB and LB if one is found.

5.15 LBNames – Search symbol table for modules with given LB

LBNames <address>

Searches the symbol table for all modules with the specified LB, and returns their names.

5.16 LOCSEARCH – Search symbol table

LOcsearch <address>

Searches the symbol table for the symbol nearest to the specified address, and returns data appropriate to the type of symbol which is found, including the offset of the <address> from the symbol.

5.17 LPRNT – Display N1LOCK status

Lprnt

The LPRNT command displays the status of all the PRIMOS N1LOCKs. For each lock, the number of readers and writers waiting is displayed.

5.18 MAP – Read map(s)

Map [<treename_1> [<treename_2>]]

The MAP command reads the PRIMOS memory maps generated by SEG at load time into an internal symbol table which is then used by may of AUTOPSY's commands. If no treename is specified, AUTOPSY attempts to read both ring zero and ring three maps from files with the names RING(0 3).MAP.<version> in a top-level directory called MAPS. Only two maps may be used by AUTOPSY at one time; in order to read in new maps, you must first use the CLRMAP command to reinitialize the symbol table.

5.19 OTHSEARCH – Search symbol table for Other symbol

Othsearch <address>

Searches the symbol table for the "other" symbol (that is, neither a procedure nor a common block) nearest the specified address. Returns its name and address if found.

5.20 PAGCHK – Check page maps for consistency

PAgchk

The PAGCHK command checks the page maps (MMAP, HMAP, and LMAP) for consistency. Any anomalous entries are listed.

5.21 PAUSE – Suspend AUTOPSY session

PAUse

Suspends this session of AUTOPSY in a restartable state.

5.22 PBSEARCH – Search symbol table for module with given PB

PBsearch <address>

Searches the symbol table for the module with the PB closest to the specified address. If one is found within 1000 words, its ECB address, PB, and LB are displayed.

5.23 PDUMP – Display PCB for specified user

Pdump [<user>]

Displays the PCB (including concealed stack) for the given user (or the current user if omitted). The display is in octal, 8 word per line.

5.24 PMAP – Display page maps for given segment

PMap [<user>]

Displays the LMAP and HMAP entries for the segment specified. The page maps are displayed in octal, 8 words per line. If <user> is omitted, the current user is assumed.

5.25 QUIT – Exit AUTOPSY

Quit

Exits autopsy.

5.26 RDUMP – Display register save area

RDump [{SLAVE | AP}]

Displays the register save (RSAV) area. On P750's and below, there is only one RSAV area, and that is displayed. On P850's, the master processor's RSAV area is displayed by default. The SLAVE option dumps the slave processor's registers, while the AP option dumps the 850 communications area registers.

5.27 READ – Read a dump from tape

Read <treename> <mt_unit>

Reads a crash dump into file <treename> from magnetic tape drive <mt_unit>. The drive must have been previously assigned, or AUTOPSY will abort with a "MTx not assigned" error. After the dump has been read in, statistics on its size and the number of errors encountered are displayed.

5.28 RESTORE – Load a segment for VPSD

REStore [<user>]

Loads the specified segment into segment 4001, which is VPSD's default segment number, and then invokes the VPSD subsystem. If <user> is omitted, the current user is assumed. Note that if the is less than 4000, the current user is set to 1.

PE-T-484, Rev. 1

5.29 RPRNT – Print live registers

RPrnt [{LIVE | LAST}] [SLAVE]

Displays the live register sets for the LIVE or LAST user using the processor. On a P850 this command applies by default to the master processor; the slave processor's LIVE and LAST process' registers are examined by adding the SLAVE argument. If neither LIVE nor LAST is supplied, LIVE is assumed.

5.30 STATUS – Display status information

Status {<user> | ALL | USERS}

Displays basic status for the specified <user>, ALL processes on the system, or all processes associated with USERS (as opposed to system processes; that is, those with positive user numbers). This status includes user or process ID, current base registers, last wait location, locks held, and whether the process was running and/or on the ready list.

5.31 SYMBOL – Display location information for given symbols

SYmbol <symbol1> <symbol2> ...

Displays the type (routine, common, or other) and address information for the specified list of symbols. For common blocks and other symbols, the address is displayed. For routines, the PB, LB, and ECB addresses are listed.

5.32 TRACE – Enter the TRACE subsystem

Trace <user> [<address>]

The TRACE command enters the TRACE subsystem for the specified <user>. If an <address> is given, the trace begins at the stack <address> specified. The TRACE command traces back down the user's current stack much as DMSTK does at PRIMOS command level, except that with the TRACE subcommands you have much more control over the examination of the stack.

At each level of the stack trace, If maps have been read in, the name of the routine and the offset of the PCL in the calling routine are displayed at each level. If more than one routine has the LB of this level, "(et al)" is displayed after the name of the first routine with this LB. TRACE will automatically resolve the routine name when the next level if pushed, if possible, and will remember the name during any future passes through this level which are made during the current invocation of TRACE. If TRACE cannot resolve the name itself, commands are available which allow you to tell TRACE which name it should use.

Experienced users of AUTOPSY should note that the TRACE subsystem was completely

rewritten at version 19.0.13. The new format is documented here, with references to the old commands (which will continue to work at least for a while). It is particularly important to note that TRACE no longer operates in character-by-character mode; carriage returns are now required after commands.

Commands to the TRACE subsystem are:

Arguments
Displays arguments at this level.

Current (formerly ".")
Redisplays the current frame.

Father [<n>] (formerly "^")
Traces back to the next stack level. If <n> is given, traces back <n> levels or to the end of the stack.

Goto [<level>]
Makes <level> the current level. If omitted, <level> defaults to 1.

Help
Prints list of commands and brief description of each.

List_Procs
Displays all procedures which have the LB of the current frame's calling routine. This command is useful if the caller's name displayed by TRACE is followed by "(et al)", which indicates that multiple entry points use this LB, and that TRACE could not distinguish between them.

Quit (formerly '/')
Exits TRACE.

ReStart [<address>]
Restarts the trace at the specified <address>. If <address> is omitted, makes the current frame the base frame for future tracing. This command is useful for stacks which have more levels (257) than TRACE can currently handle.

Set_Proc <routine_name>
Allows you to tell TRACE which routine you want it to consider the one which actually made the call. May be used only when the calling routine has multiple entry points. Note that when you get to the next frame TRACE will automatically resolve the conflict if possible; therefore, it may be a good idea to go to the next level before using this command.

Son [<n>] (formerly <cr>)
Goes back to previously traced level, or <n> levels if so specified.

STack [<start> [<end>]] (formerly "M")
Dumps the current frame in octal. If specified, only relative locations <start> to <end> are dumped. <start> defaults to '10, and <end> to the end of the frame.

User <user_number> [<address>]
Starts tracing a different user at the specified <address>.

PE-T-484, Rev. 1

5.33 TTYBUF – Display terminal buffers for given user

```
TTybuf <user> [-INput] [-OUTput] [-User_1_Message]
              [-CENtronics_1] [-CENtronics_2] [-CARD_reader]
              [-Paper_Tape_Reader] [-Paper_Tape_Punch]
              [-OCTal] [-CRLF]
```

If the -CRLF option is not supplied, the format is as for an ASCII dump in VPSD, except that there are no spaces between the contents of each word. Otherwise, carriage returns are output literally. If the -OCTAL option is given, non-printing characters are displayed as three-digit octal numbers preceded by an uparrow ("^"). Buffers are displayed in chronological order, with the most recent input or output at the bottom of the display.

If none of the buffer-selection options are given, the user's input and output buffers are displayed. Otherwise, the buffers displayed are selected by the options specified as follows:

```
INPUT:  Specified user's terminal input buffer.
OUTPUT: Specified user's terminal output buffer.
U1M:    User 1 message buffer.
CEN1:   First centronics printer's buffer.
CEN2:   Second centronics printer's buffer.
CARD:   Card reader buffer.
PTR:    Paper tape reader buffer.
PTP:    Paper tape punch buffer.
```

5.34 UNIT – Display contents of specified unit table entry

Unit {<offset> | <address>} [-uNFormatted]

Displays the contents of the unit table entry which resides at the given UTCOM$ offset or absolute address. Format is as for UTENTRY, below.

5.35 UOWNER – Display owner of specified unit table entry

UOwner {<offset> | <address>} [-DISP]

Returns the owner (user and unit number) of the unit table entry which resides at the given UTCOM$ offset or absolute address. If the -DISP option is given, the contents of the unit table entry are dumped as for the UTENTRY command, below.

5.36 UTBL – Display unit tale of specified user

UTbl <user> [-uNFormatted]

Dumps the contents of the specified user's unit table. Normally this information is given only for existing units in the form "Unit <unit> at <offset>," where <unit> is the unit number and <offset> is the word offset into the unit table common area at which the unit

table entry may be found. For pre-rev 19 dumps, or if the -UNFORMATTED option is given, the unit table is dumped in octal, 8 entries per line.

5.37 UTENTRY – Display specified unit table entry

UTEntry <user> {<unit> | -CURrent | -HOME | -INITial}
[-uNFormatted]

Displays the unit table entry for <user>'s <unit>, or one of the specified attach points, in decoded form. For pre-rev 19 dumps, or if the -UNFORMATTED option is given, the entry is displayed in octal, 8 words per line.

There are three forms the formatted version of the dump may take, depending on whether the unit is a file unit, attach point unit, or remote unit. An example of each follows:

```
> ute 1 175
Entry for unit 175 (at 40(0)/1454 [335]):
Status:    1 (Type: SAM file  Mode: r)
RWlock:    1 (excl)
Accesses:  377 (PDALUXWR)
Ldev:      0
BRA:       15727  CRA: 15727
Position:  word 0 in relative record 0
Entry at:  word 3540 in directory with BRA 14
Dir block at:   11
Hash thread:    0

> ute 44 -cur
Entry for unit 200 (at 40(0)/2654 [1535]):
Status:    2404 (Type: ACL dir  Mode: ap)
Accesses:  31 (LUR)
Ldev:      0
BRA:       160346  CRA: 160346
Position:  word 200 in relative record 0
Entry at:  word 6404 in directory with BRA 1
ACL at:    word 3022 in directory with BRA 1
Quota block at: 201
Dir block at:   211
Hash thread:    2665

> ute 44 170

Entry for unit 170 (at 40(0)/3520 [2401]):
Status:    3 (Type: SAM file  Mode: rw)
Node:           1010
Unit in slave: 175
Ldev:           37
Pack name:      SYSENH
```

5.38 VPSD — Invoke VPSD subsystem

Vpsd

Invokes the VPSD subsystem. Normally this is done with the RESTORE command, above, since that command automatically restores one of the dump's segments into segment 4001, which is VPSD's default segment. If live memory or AUTOPSY's memory is to be accessed, the VPSD command may be used instead. Note, however, that if the segment number being examined is changed from 4001, subsequent calls to VPSD, even with the RESTORE command, will reference the new segment rather than the restored segment mapped into 4001.

VPSD is described fully in *The Assembly Language Programmer's Guide* (FDR 3059).

5.39 ! — Execute PRIMOS command

! <primos_command>

Executes the specified PRIMOS command, which must be internal or an EPF. Using the ! command to invoke a static mode external command may cause AUTOPSY to abort, or at least corrupt its database.