

**ULTRIX-32
Guide to
System Crash Recovery**

Order No. AA-ME94A-TE

ULTRIX-32 Operating System, Version 3.0

Digital Equipment Corporation


Copyright © 1987, 1988 Digital Equipment Corporation
All Rights Reserved.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

DEC	Q-bus	VAX
DECnet	RT	VAXstation
DECUS	ULTRIX	VMS
MASSBUS	ULTRIX-11	VT
MicroVAX	ULTRIX-32	ULTRIX Worksystem Software
PDP	UNIBUS	

UNIX is a registered trademark of AT&T in the USA and other countries.

IBM is a registered trademark of International Business Machines Corporation.

MICOM is a registered trademark of Micom System, Inc.

This manual was written and produced by the ULTRIX Documentation Group in Nashua, New Hampshire.

Contents

About This Manual

Audience	v
Organization	v
Related Documents	v
Conventions	vi

1 System Crash Recovery

1.1 System Crashes and the Dump Process	1-1
1.1.1 Establishing Crash Dumps	1-2
1.1.2 Creating a Copy of the Dump Files	1-3
1.2 Maintaining File System Consistency	1-4
1.2.1 Identifying File System Inconsistencies	1-4
1.2.2 Invoking the fsck Command Using <code>/etc/rc</code>	1-4
1.2.3 Executing the fsck Command Interactively	1-4
1.2.4 Restoring Pseudoterminals Invoked by <code>/etc/rc.local</code>	1-5

2 Forcing a Crash Dump

2.1 Starting the Crash Dump Routine Manually	2-1
2.2 Forcing a Segmentation Fault	2-3
2.3 Initializing the Processor	2-4

Index

About This Manual

This guide provides information on recovering from a system crash using the ULTRIX-32 utilities. It also presents guidelines from which you can develop specific crash recovery procedures for your site.

Audience

The *ULTRIX-32 Guide to System Crash Recovery* is written for the person responsible for managing and maintaining an ULTRIX-32 system. It assumes that this individual is familiar with ULTRIX-32 commands, the system configuration, the system's controller/drive unit number assignments and naming conventions, and an editor such as vi or ed. You do not need to be a programmer to use this guide.

Organization

This manual consists of the following two chapters:

- Chapter 1: System Crash Recovery
Explains what the system does when a crash occurs.

- Chapter 2: Forcing a Crash Dump
Explains three ways that you can force a crash dump to occur when the system hangs.

Related Documents

You should have the hardware documentation for your system and peripherals, the *VAX Architecture Handbook*, and the *VAX Hardware Handbook*.

Conventions

The following conventions are used in this manual:

special	In text, each mention of a specific command, option, partition, pathname, directory, or file is presented in this type.
command(x)	In text, cross-references to the command documentation include the section number in the reference manual where the commands are documented. For example: See the <code>cat(1)</code> command. This indicates that you can find the material on the <code>cat</code> command in Section 1 of the reference pages.
literal	In syntax descriptions, this type indicates terms that are constant and must be typed just as they are presented.
<i>italics</i>	In syntax descriptions, this type indicates terms that are variable.
[]	In syntax descriptions, square brackets indicate terms that are optional.
. . .	In syntax descriptions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
function	In function definitions, the function itself is shown in this type. The function arguments are shown in italics.
UPPERCASE	The ULTRIX system differentiates between lowercase and uppercase characters. Enter uppercase characters only where specifically indicated by an example or a syntax line.
example	In examples, computer output text is printed in this type.
example	In examples, user input is printed in this bold type.
%	This is the default user prompt in multiuser mode.
#	This is the default superuser prompt.
>>>	This is the console subsystem prompt.
.	In examples, a vertical ellipsis indicates that not all of the lines of the example are shown.
.	
.	
<KEYNAME>	In examples, a word or abbreviation in angle brackets indicates that you must press the named key on the terminal keyboard.

<CTRL/x>

In examples, symbols like this indicate that you must hold down the CTRL key while you type the key that follows the slash. Use of this combination of keys may appear on your terminal screen as the letter preceded by the circumflex character. In some instances, it may not appear at all.

System Crash Recovery 1

This chapter discusses system crashes. It explains what happens during a system crash, how the dump process works, and how to recover from a crash. In addition, this chapter describes how to perform a file system consistency check after a system crash.

1.1 System Crashes and the Dump Process

The system monitors its own internal status and performs a number of internal consistency checks. If an internal check shows inconsistencies, the system prints panic messages to the console and then crashes. The panic messages help you determine the cause of the crash.

Prior to a system crash, but after a panic message is displayed, the system updates all file system information. The system then performs a core dump of the memory image to the dump device specified in the configuration file. The partition size of the dump device defines the size of the dump area.

If the dump device cannot contain the entire core dump, the system performs a partial crash dump. A partial crash dump only saves the vital information that helps you determine why the system crash occurred. For example, if the memory on your system is 9 Mbytes and your dump area is 5 Mbytes, the system creates a partial dump that is 5 Mbytes in size.

If your dump device is the default swap device, and your system is creating partial dumps, increase the amount of space in the swap device. See the *ULTRIX-32 Guide to System Configuration File Maintenance* for more information.

After the system dumps the raw memory image, the system reboots itself and invokes `/etc/fsck` to check for file system inconsistencies during the

reboot process.

Note

If the `fsck` command finds and corrects any corruption in the root (`/`) file system, press the `HALT` button or type `CTRL/P` to halt your processor. (The method you use depends on your processor type.) This returns you to the console prompt subsystem and allows you to reboot the system.

The `fsck` command can exit without notifying you of unexpected inconsistencies found on the root (`/`) file system. The system continues to reboot multiuser mode even if it finds unexpected inconsistencies it can fix in other file systems.

1.1.1 Establishing Crash Dumps

You establish crash dumps by specifying a `savecore` entry in the `/etc/rc.local` file. During the reboot process, the `/etc/rc.local` file invokes the `savecore` utility with the default `savecore` entry. The default `savecore` entry in the `/etc/rc.local` file is:

```
/etc/savecore /usr/adm/crash > /dev/console
```

This entry instructs `savecore` to save the errorlog files, the main memory (`vmcore`), and the kernel image (`vmunix`) after the crash. In large VAX systems, the `>/dev/console` portion of the entry instructs `savecore` to redirect any messages to the console. A `savecore` entry with the `-e` option instructs `savecore` to save only the error messages and to append them to the errorlog file. For example:

```
/etc/savecore -e /usr/adm/crash > /dev/console
```

To disable `savecore` execution, enter a number sign (`#`) in the leftmost column of the `savecore` entry in the `/etc/rc.local` file.

The following two methods can enable full crash dumps:

Method One

1. Verify that there is sufficient space in the directory specified in the `savecore` entry of the `/etc/rc.local` file. The default directory is `/usr/adm/crash`.

Note

If the directory specified by the `savecore` entry does not exist or if it is too small to hold the errorlog files, `vmcore` and `vmunix`, the `savecore` utility does nothing and a message describing the situation is not issued.

2. Ensure that you do not have the `-e` option in the `savecore` entry in the `/etc/rc.local` file.

Method Two

1. Determine a new directory (file system) to contain the dump files and create it if it does not already exist.
2. Change the directory argument for the `savecore` entry to reflect the new directory.
3. Ensure that you do not have the `-e` option in the `savecore` entry in the `/etc/rc.local` file.

If the `savecore` entry is not enabled in the `/etc/rc.local` file, but you want to create the crash dump files, you can do so manually as follows:

1. Boot the system to single-user mode.
2. Execute the `savecore` command. For example, to create the crash dump file in the `/usr/adm/crash` directory, make sure that the directory exists and then enter:

```
# /etc/savecore /usr/adm/crash
```

After a system crash, use the `adb` command to examine the crash dump or partial crash dump files. The dump files can help determine the cause of the crash, but they also use space on the specified file system. To save space and to create a permanent record of the dump files, copy the files to tape and then remove them from the specified directory.

See `savecore(8)` in the *ULTRIX-32 Reference Pages* for more information.

1.1.2 Creating a Copy of the Dump Files

To create a permanent copy of the dump files, use the `tar` command to extract the files. To copy dump files to tape using the `tar` command, use the following format:

```
tar c path/vmunix.n path/vmcore.n
```

The *path* is the directory pathname specified in the `/etc/rc.local` file such as `/usr/adm/crash`. The *n* specifies the number of the crash. Each time a system crash occurs, *n* is incremented by one. For example, if *path* is `/usr/adm/crash` and *n* is 1, type the following command line:

```
# tar c /usr/adm/crash/vmunix.1 /usr/adm/crash/vmcore.1
```

After you specify the tar command, use the rm command to remove the dump files and to conserve space on the specified file system. The following example shows how to remove the dump files. In this example, the dump files are located in /usr/adm/crash and *n* is 1.

```
# rm /usr/adm/crash/vmunix.1 /usr/adm/crash/vmcore.1
```

For further information, see the rm(1) and tar(1) commands in the *ULTRIX-32 Reference Pages*.

1.2 Maintaining File System Consistency

This section discusses how file system inconsistencies occur, how they are corrected during daily operations, and how to proceed if the fsck command cannot correct the inconsistencies.

1.2.1 Identifying File System Inconsistencies

Before the system crashes, it tries to update all file system information. The system keeps copies of the information for all active file systems in memory. The system's in-memory buffer cache contains copies of the recently used free block lists, free inode lists, modified data blocks, and the modified inodes of the mounted file systems. It also keeps all the modified superblocks of the mounted file systems.

To coordinate the changes recorded in these in-memory copies with the permanent summary information, the system periodically updates all file system information. That is, the update command executes every 30 seconds and invokes the sync system routine. However, when the system crashes, the disk-resident file system information may not be completely updated. If this occurs, inconsistencies exist between the summary information and the actual status of the file system. These can be corrected during the reboot process.

1.2.2 Invoking the fsck Command Using /etc/rc

Unless your system has a clean shut down, the fsck command checks the file systems for inconsistencies each time the system reboots. The /etc/rc file automatically invokes the fsck command to check and correct those inconsistencies that can be fixed easily.

If the fsck command encounters inconsistencies that cannot be corrected easily, /etc/rc exits multiuser startup and your system remains in single-user mode. You are instructed to run the fsck command manually. This allows you to correct specific file system inconsistencies immediately.

1.2.3 Executing the fsck Command Interactively

The `fsck` command checks your file systems when invoked for interactive execution. As it encounters each inconsistency, the `fsck` command displays a diagnostic message that indicates the type of inconsistency found and prompts you for a response to the displayed corrective action. You must answer either yes or no to this prompt.

If you answer yes to a corrective action prompt, the `fsck` command attempts to implement the corrective action. In addition, if necessary, the `fsck` command relinks all allocated, but unlinked files to the `lost+found` directory for the appropriate file system. To relink a file, the `fsck` command uses the file's inode number as its name.

If the `fsck` command relinks a file, you should determine the file's owner and the directory in which it belongs as follows:

1. Use the `ls` command with the `-i` option to gather information about the file's inode number.
2. Use the `file` command to determine the file type.
3. Contact the owner of the file and determine which directory the file belongs in. You can then move the file from the `lost+found` directory to the correct directory.

Note

The `fsck` command requires a `lost+found` directory in each file system. The `newfs` command creates this directory in each file system. However, if during operations one of these directories is inadvertently removed, use the `mklost+found` command to create this directory.

If you answer no to the corrective action prompt, the `fsck` command continues to check for other inconsistencies and creates a summary that enables you to determine your own corrective measures. If the `fsck` command can provide alternate correctives actions, it continues to prompt you for a response.

For more information, see the `fsck(8)` and `mklost+found(8)` commands in the *ULTRIX-32 Reference Pages*.

Note

If the `fsck` command tells you to reboot the system after correcting the root file system, press the `HALT` button or type `CTRL/P` (depending on your processor type). This returns you to the console subsystem prompt and allows you to boot multiuser mode.

The `fsck` command has made the other file system maintenance commands obsolete. However, for further information, see `clri(8)`, `dcheck(8)`, `dumpfs(8)`, `icheck(8)`, and `ncheck(8)` in the *ULTRIX-32 Reference Pages*.

1.2.4 Restoring Pseudoterminals Invoked by `/etc/rc.local`

After a system crash, ownership and permissions of pseudoterminals are restored to normal by the `/etc/rc.local` file. When the system returns to multiuser mode, ownership is root and permissions are 666.

Forcing a Crash Dump 2

Usually, the system reboots itself after a crash occurs. If the system does not reboot, a condition may exist that prevents the crash dump routine, doadump, from executing properly. For example, the system cannot execute the crash dump routine when an invalid interrupt stack in the kernel address space exists. Should this condition exist, you must force a crash dump as follows:

- Start the crash dump routine manually
- Force a segmentation fault
- Initialize the processor

Each successive method yields less information about the cause of crash because more of the machine state is altered. As you move through each method, you can assume that the cause of the crash is more serious. Starting a crash dump routine manually is the preferred course of action. If you cannot manually start a crash dump, force a segmentation fault. Avoid initializing the processor unless an attempt to force a segmentation fault does not work.

The following sections describe the procedures you must follow for each method of forcing a crash dump. You must be in console mode to force a crash dump. To enter console mode, press the HALT button or type CTRL/P (depending on your processor type).

2.1 Starting the Crash Dump Routine Manually

When you start a crash dump manually, you cannot change the current machine state. This is the suggested course of action. Use the following steps to start the crash dump:

1. Find the address of the dump routine by examining the fourth physical long word of the restart parameter block (RPB). For example:

```
>>>E/P/L 4
P 00000004 00001E00
```

The system displays the physical address location of the dump routine.

2. Examine the program counter (PC) which contains the address of the next instruction to be executed and stored in general register F. For example:

```
>>>E/G F
G 0000000F 80001EAD
```

3. Examine the Process Status Longword (PSL) which contains the execution state of the processor at the time that the crash occurred. For example:

```
>>> E PSL
M 00000000 04C10004
```

See the *VAX Hardware Handbook* for more information on the bit meanings in the PSL.

4. Set the PSL to Interrupt Stack with an interrupt priority level (IPL) 31. This sets the processor to run on the interrupt stack and blocks interrupts. For example:

```
>>>D PSL 041F0000
>>>
```

5. Start execution of the dump routine. For example:

```
>>>S 80001E00
```

Note that bit 31 has been changed to reflect the virtual address of the crash dump routine obtained in Step 1. This is a necessary change because the processor is still set to run in virtual memory mode.

At this point, the system should execute the dump routine, reboot itself, and place the core dump `vmunix.n` and `vmcore.n`, in the *ULTRIX-32* file system, the location of which is specified by the `savecore` entry in the `/etc/rc.local` file. The `n` specifies the dump number which is an incremental number beginning at zero. The number is incremented system by 1 with each successive dump.

To analyze the crash dump use the `adb` and the `nm` commands. See `adb(1)` and `nm(1)` in the *ULTRIX-32 Reference Pages* for more information.

2.2 Forcing a Segmentation Fault

If you cannot manually start the crash dump routine, set up a condition that forces a segmentation fault and instructs the processor to continue. To force a segmentation fault, you must set the program counter (PC) to an address that is outside of the process address space, such as PC -1. This causes the processor to synchronize the disks; however, some of the current machine state is changed.

Before you set the PC to an invalid address such as -1, examine the PC and stack pointers because these change when you force the segmentation fault.

Use the following steps to force a segmentation fault:

1. Examine the PC stored in general register F. For example:

```
>>>E/G F
G 0000000F 80001EAD
```

2. Examine the process status longword (PSL). For example:

```
>>>E PSL
M 00000000 04C10004
```

3. Display and record the kernel stack pointer (KSP) because this changes when you force a segmentation fault. The KSP is stored in internal register 0. For example:

```
>>>E/I 0
I 00000000 7FFFFDAC
```

4. Display and record the user stack pointer (USP) because this changes when you force a segmentation fault. The USP is stored in internal register 3. For example:

```
>>>E/I 3
I 00000003 7FFFE2F4
```

5. Display and record the interrupt stack pointer (ISP) because this changes when you force a segmentation fault. The ISP is stored in internal register 4. For example:

```
>>>E/I 4
I 00000004 80000C00
```

6. Set the PC to -1. For example:

```
>>>D/G F FFFFFFFF
```

7. Set the PSL to interrupt priority level 31 to block interrupts. For example:

```
>>>D PSL 001F0000  
>>>
```

8. Instruct the processor to continue. For example:

```
>>>C
```

The processor should execute the crash dump routine and to reboot itself. In addition, the crash dump data is placed in the designated area.

2.3 Initializing the Processor

If neither of the previous methods force a crash dump, you may be able to do so by initializing the processor before starting the dump routine. This action sets the processor to a known state by setting the PSL to run on the interrupt stack and the IPL to 31. In addition, the processor disables memory mapping.

Using this method, however, affects more of the machine state. Depending on your processor, the initialization may corrupt the following:

- The Interrupt Stack Pointer (ISP)
- The Kernel Stack Pointer (KSP)
- The P0 space base register (P0BR)
- The P0 space length register (P0LR)
- The P1 space base register (P1BR)
- The P1 space length register (P1LR)

See the *VAX Architecture Handbook* for more information on the ISP, KSP, and the P0 and P1 address spaces.

Use the following steps to initialize the processor:

1. Examine the restart parameter block (RPB) to obtain the dump address. For example:

```
>>>E/P/L 4  
P 00000004 00001E00
```

The processor displays the dump address.

2. Initialize the processor. For example:

```
>>>I  
>>>
```

3. Start execution of the dump. For example:

```
>>>S 1E00
```

Note that when you initialize the processor, you must specify the physical address of the dump routine because the processor is not running in virtual memory mode.

This method should cause the system to produce a crash dump, reboot itself, and place the crash dump in the ULTRIX-32 file system as defined by the savecore entry in the /etc/rc.local file.

Index

C

crash dump

- completing, 1-3
- enabling, 1-2 to 1-3
- errorlog files, 1-2
- forcing, 2-1 to 2-5
- forcing manually, 2-1 to 2-2
- forcing segmentation fault, 2-3 to 2-4
- initializing the processor, 2-4 to 2-5
- kernel image, 1-2
- main memory, 1-2
- recovery, 1-1 to 1-6

D

dump file

- copying to tape, 1-3e
- removing from file system, 1-4e

F

file system

- maintaining consistency, 1-4 to 1-6
- system crashes and, 1-4

fsck command

- invoking automatically, 1-4
- invoking for interactive execution, 1-5 to 1-6

fsck command (cont.)

- lost+found directory and, 1-5
- rebooting the system and, 1-5n

P

pseudoterminal

- system crash and, 1-6

S

savecore command

- rc.local file entry, 1-2e

system crash

- forcing a crash dump, 2-1 to 2-5
- recovery, 2-1 to 2-5

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
and New Hampshire,
Alaska or Hawaii
call **800-DIGITAL**

In Canada
call **800-267-6215**

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
100 Herzberg Road
Kanata, Ontario K2K 2A6
Attn: Direct Order Desk

INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION
PSG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminister, Massachusetts 01473

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809-754-7575

Reader's Comments

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement. _____

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or _____

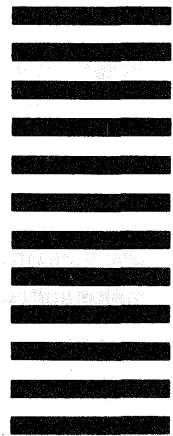
Country

-----Do Not Tear - Fold Here and Tape-----

digital



No Postage
Necessary
if Mailed in the
United States



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Documentation Manager
ULTRIX Documentation Group
ZK03-3/X18
Spit Brook Road
Nashua, N.H.

03063

-----Do Not Tear - Fold Here and Tape-----

Cut Along Dotted Line