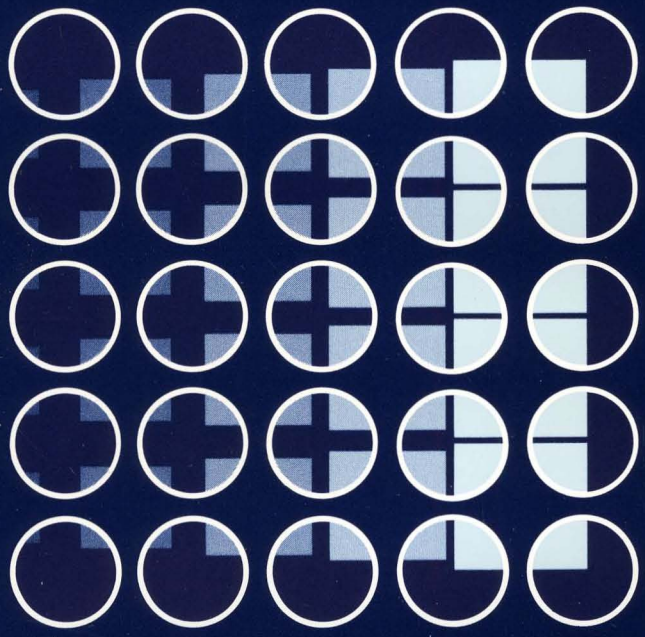
ULTRIX-32™
Programmer's Manual

ULTRIX-32™
Guidelines for System Management

Order No. AA-BG59A-TE



digital software

ULTRIX-32™
Guidelines for System Management

Order No. AA-BG59A-TE

digital equipment corporation, merrimack, new hampshire

First printing, May 1984

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 postage-paid READER'S COMMENTS form on the last page of this document requests your critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	ULTRIX-32
DECUS	UNIBUS
MASSBUS	VAX
PDP	VMS
ULTRIX	VT
ULTRIX-11	digital™

UNIX is a trademark of AT&T Bell Laboratories.

Information herein is derived from copyrighted material as permitted under a license agreement with AT&T Bell Laboratories.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California. We acknowledge the Electrical Engineering and Computer Sciences Departments at the Berkeley Campus of the University of California for their role in its development.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. We acknowledge the following individuals for their role in its development:

Eric Allman, Ken Arnold, Ozalp Babaoglu, Scott B. Baden, Jerry Berkman, John Breedlove, Earl T. Cohen, Robert P. Corbett, Mike Curry, Steve Feldman, Tom Ferrin, John Foderaro, Susan L. Graham, Charles Haley, Robert R. Henry, Andy Hertzfeld, Mark Horton, S.C. Johnson, William Joy, Howard Katseff, Peter Kessler, Jim Kleckner, J.E. Kulp, James Larus, Kevin Layer, Mike Lesk, Steve Levine, Jeff Levinsky, Louise Madrid, M. Kirk McKusick, Colin L. McMaster, Mikey Olson, Geoffrey Peck, Ed Pelegri-Llopart, Rob Pike, Dave Presotto, John F. Reiser, Asa Romberger, Bill Rowan, Jeff Schreibman, Eric P. Scott, Greg Shenaut, Eric Shienbrood, Kurt Shoens, Keith Sklower, Helge Skrivervik, Al Stanberger, Ken Thompson, Michael C. Toy, Richard Tuck, Bill Tuthill, Mike Urban, Edward Wang, David Wasley, Joseph Weizenbaum, Jon L. White, Glenn Wichman, Niklaus Wirth.

ULTRIX-32 Documentation Set

1. Organization

The ULTRIX-32 documentation set is organized into four separate binders. The documentation in each binder is so organized to better meet the needs of three separate audiences in the performance of their respective tasks. The ULTRIX-32 documentation set comprises:

Programmer's Manual Binder 1 -- General Users

Section 1 -- Commands
Documentation for all user-invoked programs (commands)

Section 6 -- Games
Documentation for all user-invoked game programs

Programmer's Manual Binder 2 -- Programmers

Section 2 -- System Calls
Documentation for the system calls (entries into the kernel)

Section 3 -- Subroutines
Documentation for the library subroutines

Section 5 -- File Formats and Conventions
Documentation for the output and system file structures

Section 7 -- Macro Packages and Language Conventions
Documentation for miscellaneous information

Programmer's Manual Binder 3A -- System Managers

Installation Guide
Documentation for installing an ULTRIX-32 system

Building an ULTRIX-32 System with the Config Program
Documentation for configuring an executable kernel image

4.2BSD Line Printer Spooler
Documentation for installing the line printer spooling system

Sendmail Installation and Operations Guide
Documentation for installing and operating the sendmail system

UUCP Installation and Administration
: Documentation for installing and administering the uucp system

Programmer's Manual Binder 3B -- System Managers

Guidelines for System Management
: Documentation for maintaining an installed ULTRIX-32 system

Section 4 -- Special Files
: Documentation for the special files (I/O devices and drivers)

Section 8 -- Maintenance Commands
: Documentation for the system maintenance programs (commands)

2. Man(1) Format

Each numbered section in Binder 1, Binder 2, and Binder 3B contains an introduction and separate entries that correspond to those created by the man(1) command. Each entry, following the order in their respective binders, describes a user command or game; a system call, library subroutine, file structure, or macro package; and a special file or maintenance command.

Regardless of their respective binder, all entries have a single, consistent format. First, the header provides information that quickly identifies the entry: name and section number (enclosed in parentheses). For example, AARDVARK(6) identifies the aardvark(6) game (Binder 1). Then, the listed subsections provide specific information about the entry. Although each entry lists only those subsections that are applicable, seven main subsections may be used. Finally, the footer provides paging information: section and consecutive page number. For example, the footers for section 6 (Binder 1) are 6-1 through 6-35.

The seven main subsections are:

NAME
This subsection lists the exact name and a short description of its function.

SYNTAX
This subsection lists the complete syntax. Boldface indicates literals. A minus sign (-) indicates command options. Ellipses (...) indicate that the preceding argument may be repeated. Square brackets [ ] indicate optional arguments.

DESCRIPTION
This subsection provides a detailed description of function and background.

FILES
This subsection lists those related files that either are part of or are used during execution.

DIAGNOSTICS
This subsection lists those diagnostic messages that may be produced. Since most self-explanatory messages are not listed, this subsection is not comprehensive.

RESTRICTIONS
This subsection lists those restrictions that are known to apply.

SEE ALSO
This subsection lists the names of the related entries and other documentation.

3. Conventions

The following conventions apply specifically to these documents:

Installation Guide (Binder 3A)
Building an ULTRIX-32 System with the Config Program (Binder 3A)
UUCP Installation and Administration (Binder 3A)
Guidelines for System Management (Binder 3B)

bold	Literals are printed in **bold type**. Literals frequently indicate a specific com mand option and should be entered exactly as printed.
case	The ULTRIX-32 system differentiates between uppercase and lowercase. Therefore, enter uppercase only where specifically indicated by an example c the command syntax.
color	Examples are printed in color. Examples represent command sequences c information that the user enters from the terminal.
<CTRL/*X*>	Terminal control characters are represented by <CTRL/*X*>, where *X* is a singl character. To generate a terminal control characters, hold down the CTR key while entering the character.
<DELETE>	The DELETE key or ERASE character is represented by <DELETE>.
italics	Substitutable parameters are printed in *italics*.
<RETURN>	The RETURN key is printed as <RETURN>. To invoke a command, enter th command sequence and depress the RETURN key.
#	The superuser prompt (normally a #) is displayed at the console when th system is in single-user mode or at a terminal when the superuser is logged in
>>>	The console subsystem prompt is represented by three right angle bracket >>>. For further information about console commands, read the *VA Hardware Manual.*

GUIDELINES FOR SYSTEM MANAGEMENT

Table of Contents

1. ULTRIX-32 System Manager

As system manager, you are responsible for maintaining your installed ULTRIX-32 system. More specifically, you are responsible for:

Daily Operations

- System shutdown
- File system backup
- File system restore
- System boot

System Administration Files

- /etc/passwd -- Grant system access
- /etc/group -- Change group security levels
- /etc/ttys -- Enable or disable terminals
- /etc/fstab -- Maintain file system table entries
- /usr/lib/aliases -- Maintain sendmail aliases file
- /usr/lib/crontab -- Change the system clock daemon
- /etc/motd -- Change the Message-of-the-Day

System Performance

- File system utilization
- File system data
- Line printer system
- System scheduling priority
- System accounting information
- System crashes
- File system inconsistencies

1.1. Guidelines for System Management

This manual is intended to assist you in developing system maintenance procedures. This manual, however, is not a trouble-shooting guide. Instead, it provides introductory discussions of the major maintenance issues that periodically require your attention. These discussions present guidelines from which you can develop specific procedures for your site.

This manual also is introductory and referential. After introducing you to a given maintenance topic, it refers you to the accompanying command documentation for more specific information.

Finally, this manual assumes that you have system management responsibilities; that you have knowledge of your system configuration, of your controller/drive unit numbers and naming conventions, and of the `vi(1)` editor; and that you have superuser privileges.

Therefore, you should familiarize yourself with the contents of this manual. Then, as the need arises, you should refer to the relevant section.

1.2. Superuser Privileges

Most system management activities require you to be logged in as the superuser.

When logged in as the superuser (root login), the system invokes your requests without performing the normal security checks. The superuser has complete access to any file as well as complete control of any process on the system. For example, the superuser can override all file mode permissions or kill any executing process.

To maintain system security, therefore, you should restrict the superuser login to those with the required system knowledge and those with system management responsibilities. Then, to increase security further, you should change the superuser password regularly.

You automatically have superuser privileges at the console when the system is in single-user mode. During multi-user mode, you have superuser privileges at any terminal from which you use su(1) with either the root or no name specified. Although the system prompts are site specific, most systems indicate superuser status by changing the prompt from either a $ or a % to a #.

For further information, read the su(1) command documentation.

Note: By first logging in as yourself and then substituting your user ID to root, you create a traceable accounting record of when and who assumed superuser status. This method may be used to increase further superuser and system security.

2. Guidelines for Daily Operations

This section provides guideline procedures for maintaining day-to-day system operations. It discusses:

- Shutdown procedure
- Backup procedure
- Restore procedure
- Boot procedure

2.1. Shutdown Procedure

On occasion, routine system maintenance may require you to shut down multi-user mode. The exact shutdown procedure that you use depends on whether you want to shutdown and remain in single-user mode, want to shutdown and halt the hardware, or want to shut down multi-user mode and reboot.

2.1.1. Shutdown Multi-User

When you want to shutdown multi-user mode (for example, to back up file systems), you should:

- Shutdown multi-user mode
- Unmount all file systems
- Sync root file system

2.1.1.1. Step 1 -- Shutdown Multi-User Mode

To shutdown multi-user mode, use the shutdown(8) command with no options specified. For example:

/etc/shutdown *time reason*

This command logs the specified reason in both /usr/adm/shutdownlog and /etc/nologin. Then, it notifies current users of the impending shutdown. Five minutes prior to the specified *time*, it disables login(1). (Users who attempt to log in are notified of the reason for the shutdown from /etc/nologin.) Finally, at the designated *time*, shutdown(8) shuts down multi-user mode.

Once the system displays a superuser prompt (#), the system is back to single-user mode. The system console is open with the superuser account active. All other terminals are disabled. But all file systems still are mounted.

For further information, read the login(1) and shutdown(8) command documentation.

Note: During multi-user reboot, /etc/rc automatically removes /etc/nologin and re-enables user logins.

2.1.1.2. Step 2 -- Unmount File Systems

To unmount all file systems, use the umount(8) command with the −a option specified. For example:

```
/etc/umount -a
```

This command unmounts those file systems named in /etc/fstab and leaves only the root file system available. (If you have mounted a file system that is not defined in /etc/fstab, use the umount(8) command with its name specified to unmount it.)

For further information, read the umount(8) command documentation.

2.1.1.3. Step 3 -- Sync Root File System

To sync the root file system, use the sync(8) command. For example:

```
/etc/sync
```

This command flushes the system buffers and writes out to disk the in-memory information for the root file system.

Having sufficiently prepared the single-user mode, you then may proceed with the desired maintenance prodecdure.

For further information, read the sync(8) and update(8) command documentation.

2.1.2. Shutdown and Halt

When you want to shutdown multi-user mode and halt the hardware (for example, for scheduled field service), you should use the shutdown(8) command with the −h option specified. For example:

```
/etc/shutdown -h time reason
```

This command logs the specified reason in both /usr/adm/shutdown and /etc/nologin. Then, it notifies current users of the impending shutdown. Five minutes prior to the specified *time*, it disables login(1). (Users who attempt to log in are notified of the reason for the shutdown from /etc/nologin.) Finally, at the specified *time*, shutdown(8) shuts down multi-user mode and halts the hardware.

For further information, read the shutdown(8) command documentation.

Notes: 1. During mutli-user reboot, /etc/rc automatically removes /etc/nologin and re-enables user logins.

2. The fasthalt(8) and halt(8) commands provide alternative shutdown procedures, but they are not recommended for daily operations. For further information, read the fasthalt(8) and halt(8) command documentation.

2.1.3. Shutdown and Reboot

When you want to shutdown multi-user mode and reboot immediately, you should use the shutdown(8) command with the −r option specified. For example:

/etc/shutdown -r *time reason*

This command logs the specified reason in both /usr/adm/shutdown and /etc/nologin. Then, it notifies current users of the impending shutdown. Five minutes prior to the specified *time*, it disables login(1). (Users who attempt to log in are notified of the reason for the shutdown from /etc/nologin.) Finally, at the specified *time*, shutdown(8) shuts down multi-user mode, syncs the file systems, and immediately reboots multi-user mode.

For further information, read the shutdown(8) command documentation.

Notes: 1. During multi-user reboot, /etc/rc automatically removes /etc/nologin and re-enables user logins.

2. The fastboot(8) and reboot(8) commands provide alternative shutdown procedures, but they are not recommended for daily operations. For further information, read the fastboot(8) and reboot(8) command documentation.

2.2. Backup Procedure

To guard against data loss caused by a hardware failure, a system crash, or a human error, you should back up your file systems regularly.

Because dump(8) performs incremental dumps, you should develop a backup strategy that minimizes both the time and number of tapes required. In addition, because serious file system inconsistencies can occur when dumping a file system that is active, you should shutdown multi-user mode before backing up your file systems.

For further information, read **Shutdown Multi-User**.

2.2.1. Backup Strategy

Your backup strategy depends on the needs and resources of your installation. Because dump(8) performs incremental backups that can, if unplanned, waste both time and tape, you should develop a practical dump sequence.

One simple dump sequence would start with a monthly level 0 dump followed by weekly level 5 dumps and daily level 9 dumps. This sequence would ensure that those files that had changed since the same or lower dump level would be dumped to tape.

The following provides a sample monthly dump sequence.

```
/etc/dump 0key... filesystem  (1st of month)
/etc/dump 9key... filesystem
      .
      .
      .
/etc/dump 5key... filesystem  (1st full week)
/etc/dump 9key... filesystem
      .
      .
      .
/etc/dump 5key... filesystem  (last full week)
/etc/dump 9key... filesystem
      .
      .
      .
```

The dump(8) command uses the date on which a given level is specified as a time stamp for determining which files are to be dumped to tape. With level 0 the highest and level 9 the lowest in order of precedence, dump(8) compares the date a given file was last modified with that of the specified dump level. Then, it dumps only those files that have been changed since the appropriate date (time stamp of a same or lower level number).

For further information, read the dump(8) command documentation.

Note: For an alternative backup strategy, read "Installing and Operating 4.2BSD on the VAX" in *ULTRIX-32 Supplementary Documents, Volume III.*

2.3. Restore Procedure

The exact restore procedure that you use depends on whether you want to restore a complete file system or want to restore individual files.

2.3.1. Restore File System

When you want to restore a complete file system, you first should shutdown multi-user mode. Then, you should:

- Create target file system
- Mount target file system
- Change directory
- Restore from backup tape(s)
- Change directory
- Unmount restored file system
- Check restored file system
- Dump restored file system

For further information, read **Shutdown Multi-User** and **Invoke Multi-User Mode**.

2.3.1.1. Step 1 -- Create Target File System

To ensure sufficient free blocks for the files on the dump tape(s), use the newfs(8) command on an umounted (raw) device. For example:

/etc/newfs *special type*

This command creates a new, empty target file system on the specified device.

For more information, read the newfs(8) command documentation.

Note: Again, you are to execute newfs(8) on an unmounted file system. Otherwise, you will crash your system.

2.3.1.2. Step 2 -- Mount Target File System

After creating a target file system, use the mount(8) command to mount it. For example:

/etc/mount *special directory*

This command mounts the target file system on the designated directory and makes this file system available for use.

For more information, read the mount(8) command documentation.

2.3.1.3. Step 3 -- Change Directory

Once the target file system is available for use, use the cd(1) command to position yourself at its root directory. For example:

cd *directory*

This command positions you at the root directory of the target file system.

For more information, read the cd(1) command documentation.

2.3.1.4. Step 4 -- Restore File System

Having sufficiently prepared the target file system, you then should restore the file system from its most recent and most complete dump tape(s). That is, for a single reel dump tape, use the restore(8) command with r key specified. For example:

/etc/restore r

This command restores the file system from a single reel dump tape. (If the dump tape is on multiple reels, however, you should specify the R key.)

For more information, read the restore(8) command documentation.

2.3.1.5. Step 5 -- Change Directory

After restoring the file system, use the cd(1) command to reposition youself back at the root of the file system, /. For example:

```
cd /
```

This command repositions you back at /.

For further information, read the cd(1) command documentation.

2.3.1.6. Step 6 -- Unmount Restored File System

After repositioning yourself at the root of the file system (/), use the umount(8) command to unmount the restored file system. For example:

/etc/umount *filesystem*

This command unmounts the specified file system and prepares it for the next step.

For further information, read the unmount(8) command documentation.

2.3.1.7. Step 7 -- Check File System

After unmounting the restored file system, use the fsck(8) command without any options specified to check it for inconsistencies. For example:

/etc/fsck *filesystem*

When executed without options specified, fsck(8) checks the named file system, notifies you of all inconsistencies, prompts for a response to its suggested course of action, and proceeds accordingly. When unsure of the consequences of your response, you should answer no. By answering no, however, you leave the condition uncorrected but create a summary from which you can better decide on a plan of action.

For further information, read **Guidelines for File System Consistency** and the fsck(8) command documentation.

2.3.1.8. Step 8 -- Dump File System

Finally, use the dump(8) command to create a new level 0 dump of the restored file system. For example:

/etc/dump 0*key... filesystem*

This command creates a new level 0 dump (epoch) of the specified file system. This dump not only reflects the file system as it now exists on disk, but it also prepares it for future incremental dumps.

For further information, read the dump(8) command documentation.

2.3.2. Restore Individual Files

When you want to restore individual files, you may leave the system in multi-user mode but should:

- Change to target directory
- Restore file from backup tape

2.3.2.1. Step 1 -- Change Directory

To position yourself at the appropriate target directory, use the cd(1) command and specify the root directory of the target file system. For example:

```
cd directory
```

This command places you at the root directory of the target file system.

For further information, read the cd(1) command documentation.

2.3.2.2. Step 2 -- Restore File

After establishing the desired directory, use the restore(8) command with the i key specified. For example:

```
/etc/restore i
```

This command invokes restore(8) for interactive execution and allows you to search through the mounted dump tape for the desired file(s).

For further information, read the restore(8) command documentation.

2.4. Boot Procedure

After most crashes, the system normally reboots itself. However, if the system is unable to reboot itself, you may have to boot manually from the console. The exact boot command that you issue depends not only on whether you want the system to come up in single-user or multi-user mode but also on your system hardware.

2.4.1. Boot Single-User Mode

When system maintenance requires you to boot single-user mode (for example, to dump or restore file systems), you may enter one of the following commands at the console subsystem.

On a 11/730 or 11/780, you may enter:

```
>>> B xxS
```

This single command sequence provides all the information needed to boot single-user mode. The *xx* represents the naming convention used for the system disk (for example, hp, hk, ra, or rb) from which the system boots single-user mode.

Alternatively, on a 11/730 or 11/780, you also may enter:

```
>>> B ANY
```

This command first prompts for the device and file name. Then, following your response, it boots single-user mode.

On a 11/750, you may enter:

```
>>> B/2
```

This command uses the default device indicated by the front panel switch setting and boots single-user mode.

Alternatively, on a 11/750, you also may enter:

```
>>> B/3
```

This command loads the boot image from the default device indicated from the front panel setting. Then, it prompts for device and file name. Following your response, it boots single-user mode.

When booting single-user mode manually at the console, you should always check the root and all other file systems that you plan to use. To check these file systems, use the fsck(8) command. For example:

```
/etc/fsck filesystem...
```

When executed without any options specified, fsck(8) checks the named file system(s), notifies you of all inconsistencies, prompts for a response to its suggested course of action, and proceeds accordingly. When unsure of the consequences of your response, you should answer no. By answering no, you leave the inconsistency uncorrected but create a summary from which you can better decide on a plan of action.

After correcting all reported inconsistencies, you may mount all required file systems and begin activity.

For further information, read **Guidelines for File System Consistency** and the fsck(8) command documentation.

2.4.1.1. Single-User Mode

During single-user mode, the system comes up with the root file system mounted and sh(1) running at the console. All other file systems are unmounted, and all configured terminals are disabled. Therefore, you have access only to those files on

the root file system.

The superuser account is active at the console. But because login(1) has not been invoked, the superuser name is not logged in. Therefore, although sh(1) has read in .profile, login(1) has not done a full environmental initialization for the superuser account. Under certain circumstances (for example, executing cd(1) without specifying a directory), you may encounter unexpected results.

For further information, read the login(1), sh(1), and init(8) command documentation.

2.4.1.2. Invoke Multi-User Mode

When you wish to invoke multi-user mode without having to reboot, you should:

- Unmount all file systems
- Check all file systems
- End Single-User Mode

2.4.1.2.1. Step 1 -- Unmount File Systems

To unmount all file systems, use the unmount(8) command with the −a option specified. For example:

```
/etc/umount -a
```

This command unmounts all file systems listed in /etc/fstab and leaves only the root file system available. (If you have mounted a file system that is not defined in /etc/fstab, use the umount(8) command with its name specified to unmount it.)

For further information, read the umount(8) command documentation.

Note: The umount(8) command most often fails if a progam or user is active on a mounted file system. Therefore, you should be at the root directory of the file system (/) and should have no active program on any mounted file system.

2.4.1.2.2. Step 2 -- Check File Systems

After unmounting all file systems, use the fsck(8) command to check them for inconsistencies. For example:

```
/etc/fsck
```

This command checks the file systems that are specified in /etc/fstab and normally are mounted during multi-user mode. It notifies you of all inconsistencies, prompts for a response to its suggested course of action, and proceeds accordingly. When unsure of the consequences of your response, you should answer no. By answering no, you leave the inconsistency uncorrected but create a summary from which you can better decide on a plan of action.

For further information, read **Guidelines for File System Consistency** and the fsck(8) command documentation.

2.4.1.2.3. Step 3 -- End Single-User Mode

After correcting all reported inconsistencies, enter a <CTRL/D> at the console.

The <CTRL/D> ends the single-user session. Once single-user mode ends, the system initialization program, init(8), automatically invokes the multi-user startup file, /etc/rc. During execution, /etc/rc then invokes /etc/rc.local. When these multi-user startup file successfully complete execution, the system comes up in multi-user mode.

For further information, read **Multi-User Mode** and the init(8) command documentation.

2.4.2. Boot Multi-User Mode

When circumstances require you to boot multi-user mode, you may enter one of the following commands at the console subsystem.

On all machines, you may enter:

```
>>> B
```

This single command uses the default boot device and boots multi-user mode. On a 11/730 or 11/780, the default device is specified by the DEFBOO.CMD command file on your console floppy or cassette. On a 11/750, the default device is determined by the front panel switch. Then, after the multi-user startup files (/etc/rc and /etc/rc.local) successfully complete execution, the system comes up in multi-user mode.

Alternatively, on a 11/730 or 11/780, you also may enter:

```
>>> B xxM
```

This single command sequence provides all the information to boot multi-user mode. That is, *xx* represents the naming convention used for the system disk (for example, hp, hk, ra, or rb) from which it boots muti-user mode. Again, after the multi-user startup files (/etc/rc and /etc/rc.local) successfully complete execution, the system comes up in multi-user mode.

Alternatively, on a 11/750, you also may enter:

```
>>> B/1
```

This command loads the boot image from the default device indicated by the front panel setting. Then, it prompts for the device and file name. Following your response, it boots multi-user mode. Again, after the multi-user startup files (/etc/rc and /etc/rc.local) successfully complete execution, the system comes up in multi-user mode.

2.4.2.1. Multi-User Mode

During multi-user startup, /etc/rc automatically invokes fsck(8), which checks the file systems defined in /etc/fstab. If fsck(8) successfully completes execution, /etc/rc continues and starts up multi-user mode. However, if fsck(8) notifies you of unexpected inconsistencies, /etc/rc exits multi-user startup, and the system remains in single-user mode.

When the system remains in single-user mode, you should immediately reinvoke fsck(8) without any options specified on the file systems with reported inconsistencies. For example:

/etc/fsck *filesystem...*

When executed without options specified, fsck(8) checks the named file system(s), notifies you of all inconsistencies, prompts for a response to its suggested course of action, and proceeds accordingly. When unsure of the consequences of your response, you should answer no. By answering no, you leave the inconsistency uncorrected but create a summary from which you can better decide on a plan of action.

After correcting all reported inconsistencies, you should invoke or reboot multi-user mode.

Once multi-user mode starts up, the system comes up with the root as well as all other file systems specified in /etc/fstab mounted and with the appropriate terminals listed in /etc/ttys enabled. All users with accounts in /etc/passwd then may log in.

For further information, read **Invoke Multi-User Mode, Shutdown and Reboot, Guidelines for File System Consistency**, and the fsck(8) command documentation.

3. Guidelines for System Administration Files

This section provides guideline procedures for maintaining the system administration files. It discusses:

- /etc/passwd -- Adding new users
- /etc/group -- Adding/deleting groups or group members
- /etc/ttys -- Enabling/disabling terminals
- /etc/fstab -- Adding/deleting file system table entries
- /usr/lib/aliases -- Adding/deleting sendmail aliases
- /usr/lib/crontab -- Modifing the clock daemon table
- /etc/motd -- Creating a message-of-the-day

3.1. Add New Users

Before a user attempts to login to your system, you should prepare the proper login environment. For each user, you should:

- Create a user login entry
- Create a home directory
- Copy distributed startup files
- Change user and group IDs

3.1.1. Step 1 -- Create User Login Entry

When a user attempts to login, the system checks the password file, /etc/passwd, for an entry beginning with that user name. If such an entry exists, the system begins the login process. For each user that is to access your system, use the vipw(8) command to create an entry in /etc/passwd.

Before creating entries in /etc/passwd, you should familiarize yourself with its format. Each entry in /etc/passwd contains information that the system uses in determining login permission and in setting up the user's initial process. Each entry contains seven fields of information, which are delimited by colons.

- The first field contains the user's login name (1-8 characters). The system uses this name in establishing login permission. When creating a new entry, enter the user's login name followed by a colon.
- The second field contains the user's encrypted password, if used. The system uses this password in verifying login permission. When creating a new entry, leave this field blank, that is, simply add a colon. Then, with the passwd(1) command, either you or the user later may set a password.
- The third field contains the user's identification number (UID). Once login permission is granted, the system internally translates the login name to this user ID number and uses it in identifying the user's processes and in determining owner access permissions to files. When creating a new entry, enter the next UID in sequence followed by a colon.

- The fourth field contains the user's appropriate group identification number (GID). Again, once login permission is granted, the system internally establishes the user's group ID number and uses it in determining group access permission to files. A user may simultaneously belong to a maximum 8 groups. When creating a new entry, however, enter the GID number of the user's predominant group followed by a colon.
- The fifth field contains the user's personal information. The finger(1) command uses this information. When creating a new entry, enter the user's full name followed by a colon. Then, with the chfn(1) command, the user later may supply additional information.
- The sixth field contains the absolute pathname to the user's home (initial working) directory. After establishing the appropriate user and group IDs, the system uses this pathname to place the user in the named directory. When creating a new entry, enter the pathname to the appropriate directory followed by a colon.
- The seventh field contains the absolute pathname to the command that is to be executed immediately upon conclusion of the login process. This normally is a version of the shell (command interpreter). When creating a new entry, enter the pathname to csh(1) or sh(1). If nothing is specified, the system automatically invokes /bin/sh. With the chsh(1) command, the user later may change this default login shell.

The following provides a sample entry from /etc/passwd.

```
jones::100:4:David Jones:/usr/users/jones:/bin/csh
```

This entry contains information for a user whose login name is jones; whose password field is left blank; whose user ID and group ID are 100 and 4, respectively; whose name field contains the user's full name; whose assigned home directory is /usr/users/jones; and whose initial process is the csh(1) shell.

For further information, read the chfn(1), chsh(1), finger(1), passwd(1), and vipw(8) command documentation.

3.1.2. Step 2 -- Create Home Directory

After granting login permission and establishing the appropriate user and group IDs, the system attempts to place a user in the designated home directory. For each user, you also should create a directory with the name specified in the password file in /usr/users.

To create a user home directory, use the mkdir(1) command with the full pathname

specified. For example, specifying /usr/users/jones:

```
mkdir /usr/users/jones
```

This command creates an empty directory with the name jones in the users subdirectory of the /usr file system.

For further information, read the cd(1) and mkdir(1) command documentation.

Note: If you do not create a home directory for a user, the system allows the user to login and places the user in the root directory of the file system, /.

3.1.3. Step 3 -- Copy Distributed Startup Files

After placing the user in the appropriate home directory, the system invokes the designated initial process. Usually, this process is either csh(1) or sh(1). These shell commands look in the user's home directory for their startup files. For each user, you also should copy the required startup files into the appropriate home directory.

On your distributed system, /usr/skel contains prototypes of all of the startup files that a user initially needs. To copy these files from /usr/skel to the user's home directory, use the cp(1) command with the following arguments specified.

```
cp /usr/skel/.??* /usr/users/name
```

Once the shell expands this pattern to file names, this command copies .cshrc, .login, .mailrc, .profile, and .project from /usr/skel to /usr/users/*name*.

Once copied, you should edit these files and add the defaults that apply for your site. To edit each startup file, use the vi(1) command. The user later can edit these files to suit individual preferences.

For further information, read the cp(1), csh(1), finger(1), mail(1), sh(1), and vi(1) command documentation.

3.1.4. Step 4 -- Change User and Group IDs

After creating a user home directory and copying the required startup files, you also should change their assigned user and group IDs.

When a directory or a file is created, the system records the user and group ID numbers of the user who created it in the inode that describes it. Since you created both the user's home directory and the startup files, the system has recorded your user and group IDs for them. For the user to have full owner access privileges to the home directory and these startup files, you should change their assigned user and group IDs.

To change the appropriate user and group IDs, use the chown(8) and chgrp(1) com-

mands with the following arguments specified.

/etc/chown *name* /usr/users/*name* /usr/users/*name*/.??*

and

chgrp *group* /usr/users/*name* /usr/users/*name*/.??*

These commands change the user and group IDs assigned to the newly created directory and startup files.

For further information, read the chgrp(1) and chown(8) command documentation.

3.2. Add/Delete Groups or Group Members

When necessary, you may either have to add new groups to your system or have to add new users to existing groups. To add new groups or group members, you should edit the system group file, /etc/group. Then, either create a new group entry or add the user name to an existing one.

Before editing /etc/group, you should familiarize yourself with its format. Each entry in /etc/group contains information that the system uses when translating individual group IDs to names and in verifying simultaneous membership in multiple groups. Each entry contains four fields of information, which are delimited by colons.

- The first field contains the group name. When creating a new group entry, enter the group name followed by a colon.
- The second field contains an encrypted password. Currently, this field is not used but should not be left blank. When creating a new entry, add an asterisk (*) followed by a colon. (An asterisk eliminates group password matching.)
- The third field contains the group identification number (GID). The system uses this number in determining group access permissions to files. When creating a new entry, enter the appropriate group ID followed by a colon.
- The fourth field contains the names of the current members of the group, which are delimited by commas. When creating a new entry or adding new members, enter each name followed by a comma.

To add a new group, execute vi(1) on /etc/group to create a new entry. The following is a sample entry from /etc/group.

```
research:*:25:jones,wilson
```

This entry contains information for the research group. The password field contains an asterisk and eliminates password matching. The group ID is 25, and current members are Jones and Wilson.

To add new members to an existing group, use vi(1) on /etc/group to add the users' login names to the appropriate list. For example, adding two names to the last sample

entry:

```
research::25:jones,wilson,smith,thompson
```

This entry now lists four current members: Jones, Wilson, Smith, and Thompson.

When you have to delete either an existing group or an existing member from a group, use vi(1) on /etc/group to delete the appropriate entry or name(s).

For further information, read the group(5) file description.

3.3. Enable/Disable Terminals

When necessary, you may have to enable or disable a configured terminal for daily operations. To enable or disable a configured terminal, you should edit the terminal initialization file, /etc/ttys. Then, change information listed for the appropriate terminal entry.

Before editing /etc/ttys, you should familiarize yourself with its format. Each entry in /etc/ttys contains information that init(8) uses during system boot in determining if a terminal is to be opened for login(1). Each entry in /etc/ttys contains three fields of information, which are undelimited.

- The first field contains either 0 or 1. If a 0 is specified, init(8) ignores that entry during system initialization. If a 1 is specified, init(8) attempts to open that terminal for login(1).
- The second field contains either a single number or character. The value indicates the terminal's baud rate and initial settings. For further information, read the gettytab(5) file documentation.
- The third field contains the terminal's special file name, as listed in /dev.

To enable a previously configured terminal, use vi(1) on /etc/ttys to change the first field of the appropriate entry to a 1. Similarly, to disable a previously configured terminal, use vi(1) on /etc/ttys to change the first field of the appropriate entry to a 0.

When you next boot your system, init(8) automatically will act on the changed entry accordingly. To implement a change during multi-user mode, use the kill(1) command to send a hangup signal to init(8). For example:

```
kill -HUP 1
```

This command sends a hangup signal to init(8) which rescans /etc/ttys and processes only thoses entries that have been changed.

For further information, read the kill(1) and init(8) command documentation.

3.4. Add/Delete File System Table Entries

When necessary, you may have to modify the file system table, /etc/fstab, to reflect required changes (for example, adding a new file system or changing the order in which your file systems are to be mounted, dumped, or checked). To modify the file system table, you should edit /etc/fstab. Then, either create a new entry or

add/change information for an existing one.

Before editing /etc/fstab, you should familiarize yourself with its format. Each entry in /etc/fstab contains default information for the mount(8), dump(8), and fsck(8) command. Each entry in /etc/fstab contains five fields of information, which are delimited by colons.

- The first field contains the file system's special file name. When creating a new entry, enter the appropriate block special file name followed by a colon.
- The second field contains the absolute pathname to the directory on which the file system is mounted. This is used as default information for the mount(8) command. When creating a new entry, enter the appropriate pathname followed by a colon.
- The third field contains the file system mode: rw (read-write), ro (read only), rq (read-write with quotas), sw (swap), and xx (ignore). If sw is specified and if the file system has been configured for such use, swapon(8) (invoked by /etc/rc) makes it part of the system swap space. If xx is specified, the file system is ignored, that is, not processed by mount(8), dump(8), or fsck(8). When creating a new entry, enter the appropriate mode followed by a colon.
- The fourth field contains the file system dump frequency (every *n*th day). This is used as default information for the dump(8) command. When creating a new entry, enter the appropriate frequency rate followed by a colon.
- The fifth field contains the file system pass number. This is used as default order for the fsck(8) command. Normally, the root file system has a pass number of 1. Then, the remaining file systems are assigned higher pass numbers that allow efficient, simultaneous parallel checks. When creating a new entry, first consider that all file systems within a single drive should have different numbers and that file systems on different drives may be checked at the same time, hence, have the same pass number. Then, enter the appropriate pass number.

To add a new entry to the file system table, use vi(1) on /etc/fstab to create a new entry. The following is a sample entry from /etc/fstab.

```
/dev/hp0h:/usr/users:rw:1:3
```

This entry lists information for a file system that resides on a disk drive's h partition, that is mounted on the /usr/users directory, that is mounted read-write, that is dumped every day, and that is checked on the third pass.

To change an existing entry in the file system table, use vi(1) on /etc/fstab to change the appropriate field of information.

For further information, read the fstab(5) documentation.

Note: Since mount(8), dump(8), and fsck(8) process the entries in the order that they appear, the order of the entries in /etc/fstab is important. For example, mount(8) will fail if it is told to mount a file system on a directory that itself

is not yet mounted. You should make sure that each entry in /etc/fstab is listed in a logical order.

3.5. Add/Delete Sendmail Aliases

When necessary, you may have to add user names to the sendmail(8) aliases file, /usr/lib/aliases. To add user name to this file, you should edit /usr/lib/aliases. Then, either create a new alias entry or add new name to an existing one.

Before editing /usr/lib/aliases, you should familairize yourself with its format. Each entry in /usr/lib/aliases contains information that sendmail(8) uses in routing messages to users when an alias is specified. Each entry in /usr/lib/aliases contains two fields of information, which are delimited by colons.

- The first field contains the alias name. When creating a new entry, enter the alias followed by a colon.
- The second field contains the user login names, which are delimited by commas. This information lists the user names to which the mail is to be routed when the alias is specified. When creating a new entry, enter each user name followed by a comma.

To add user names to the alias file, use vi(1) on /usr/lib/aliases to create an entry under the #people comment that lists the appropriate alias and name(s). The following is a sample entry from /usr/lib/aliases.

```
research:jones,wilson,smith,thompson
```

This entry lists the names (Jones, Wilson, Smith, and Thompson) that sendmail(8) routes messages to when the reseach alias is specified.

To make these additions immediately become part of the sendmail(8) aliases data base, use the newaliases(1) command. For example:

```
newaliases
```

This command reads the new information added to /usr/lib/aliases and rebuilds the sendmail(8) aliases data base.

For further information, read "SENDMAIL Installation and Operation Guide" in *ULTRIX-32 Supplementary Documents, Volume III* as well as the newaliases(1), aliases(5), and sendmail(8) documentation.

3.6. Modify Clock Daemon Table

When necessary, you may have to change the clock daemon table, /usr/lib/crontab. Once invoked during multi-user startup, the system clock daemon, cron(8), wakes up every 60 seconds and executes those commands listed in /usr/lib/crontab (daemon table) that are scheduled for that time. To change the clock daemon table, you should edit /usr/lib/crontab. Then, either create a new entry or change an existing one.

Before editing /usr/lib/crontab, you should familiarize yourself with its contents. Each entry in /usr/lib/crontab contains information that specifies a time and command sequence that is to be executed regularly. Each entry contains six fields of information, which are delimited by blanks.

- The first field specifies the exact minute that the command sequence is to be executed. When creating a new entry, enter the exact minute (0-59) followed by a blank.
- The second field specifies the hour of the day on which the command sequence is to executed. When creating a new entry, enter the exact hour (0-23) followed by a blank.
- The third field specifies the day of the month on which the command sequence is to be executed. When creating a new entry, enter the day number (1-31) followed by a blank.
- The fourth field specifies the month of the year on which the command sequence is to be executed. When creating a new entry, enter the month number (1-12) followed by a blank.
- The fifth field specifies the day of the week on which the command sequence is to be executed. When creating a new entry, enter the weekday number (1-7, Monday-Sunday) followed by a blank.
- The sixth field specifies the command sequence that is to be executed. When creating a new entry, enter the complete command sequence.

In addition, the first five fields may specify either a single time indicator, a multiple time indicator, a time range, or an asterisk. A single time indicator may consist of one or two consecutive digits (for example, 3 or 33). A multiple time indicator consist of a string of indicators separated by commas (for example, 5,10,15,20). A time range consists of two indicators separated by a dash (for example, 5-20). An asterisk represents all times.

To change the system clock daemon table, use vi(1) on /usr/lib/crontab. Then, either add a new entry or make the appropriate changes to an existing one. The following is a sample entry from /usr/lib/crontab.

```
0,30 * * * * /usr/lib/atrun
```

This entry specifies that cron(8) now is to invoke /usr/lib/atrun daily on the hour and half hour only.

For further information, read /etc/rc and the cron(8) command documentation.

3.7. Create a Message-of-the-Day

When necessary, you may have to provide all system users with information that is relevant to that day's operation. When you need to send revelant information to all users, you may create a message-of-the-day.

When you choose to create a message-of-the-day, use vi(1) on /etc/motd to overwrite its contents with the new information. The system displays the content of /etc/motd at the user's terminal after each login. The system continues to display this same message until you either change or delete the contents of /etc/motd.

For further information, read the vi(1) command documentation.

4. Guidelines for System Performance

This section provides guideline procedures for managing system performance. It discusses:

- Managing file system utilization
- Managing file system data
- Managing your line printer system
- Managing system scheduling priority
- Managing system accounting information

4.1. Manage File System Utilization

You should regularly monitor the disk usage of your configured file system to ensure an adequate amount of free space. To ensure adequate free space, you should:

- Check available free space
- Check disk usage
- Free disk space
- Verify disk quotas (if imposed)

4.1.1. Step 1 -- Check Available Free Space

To ensure sufficient free space for your configured file systems, you should regularly check the amounts of disk free space available on the file systems listed in /etc/fstab. To check your disk free space, use the df(1) command. For example:

```
df
```

This command reports of the amount of free space available on the file systems that are defined in /etc/fstab. It reports the file system's configured size (Kbytes), the amount presently utilized, the amount presently available, the percentage utilized, and the directory on which it is mounted.

Normally, your file systems are configured with a minimum free space percentage established. When constructing a new file system, the newfs(8) command reserves a minimum percentage of the space from normal use (default 10%). This default percentage allows for a report in excess of 100%. In interpreting the free space report, therefore, you should look for significant changes in file system disk usage.

For further information, read the df(1) and newfs(8) command documentation.

4.1.2. Step 2 -- Check Disk Usage

After determining that a file system has insufficient space available, you should try to determine how its space is being utilized. You should determine which users have used the most space and are most likely to be able to free up disk space.

To display a summary report of how space is being used on a file system, use the

du(1) command with the −s option specified. For example:

```
du -s path/*
```

This command displays a summary report of the number of blocks used by each main subdirectory in the specified file system. Normally, this information is sufficient to determine which users have the most disk space.

For further information, read the du(1) command documentation.

4.1.3. Step 3 -- Free Disk Space

One way to free up space on a file system is to transfer it to a larger file system. If this option is possible, you first should make a level 0 dump the file system. Then, you should restore it on the larger target file system.

If this option is impossible, you should use the information gathered during the previous step and request that users remove their large, unused files. If there is still an insufficient amount of free space, you should request that users of that file system remove their obsolete files and dump to tape their infrequently used files.

For further information, read **Backup Procedure** and **Restore File System**.

4.1.4. Step 4 -- Verify Disk Quotas

If you are enforcing user disk quotas, you should verify your quota system periodically. To compare the established limits with the actual usage, use:

quot(8)	To display actual block usage for each user
quotacheck(8)	To verify actual block usage is consistent with established limits
repquota(8)	To display both actual disk usage and established limits

Then, if you find it necessary to change the established quotas, use the edquota(8) command. This command allows you to set or change the usage limits for each user.

For further information, read "Disc Quotas in a UNIX Environment" in *ULTRIX-32 Supplementary Documents, Volume III* as well as the edquota(8), quotacheck(8), quotaon(8), quotaoff(8), and repquota(8) command documentation.

4.2. Manage File System Data

When necessary, you may either need to move a file system from one disk pack to another or need to merge files from one file system into another.

4.2.1. Move File System

When moving a file system from one disk pack to another, you should:

- Dump file system (level 0)
- Restore file system

For further information on these procedures, read **Backup Procedure** and **Restore File System**.

4.2.2. Merge Files

When merging individual files from one file system into another, you should extract the appropriate file and restore it to the target file system. To extract and restore individual files, use the tar(1) command. For example:

```
tar key file
```

The tar(1) command extracts the named file. Then, you can use tar(1) to restore the file from tape to the target file system.

For further information, read the tar(1) command documentation.

4.3. Manage Your Line Printer System

During multi-user startup, lpd(8) automatically activates your configured line printer system. To manage your line printer system, use the lpc(8) command. For example:

```
/etc/lpc
```

This command allows you to control the activity of the line printers and spooler queues listed in /etc/printcap. More specifically, it allows you to:

- Enable/disable a printer
- Enable/disable a spooler queue
- Alter order of queued jobs
- Display printer, queue, or daemon status

For further information, read /etc/rc and the printcap(5), lpc(8), and lpd(8) command documentation.

Note: For a detailed explanation of the line printer system, read "4.2BSD Line Printer Spooler Manual" in *ULTRIX-32 Supplementary Documents, Volume III.*

4.4. Manage Process Scheduling Priority

As the superuser, you can alter the system's process scheduling priority.

To alter the priority of given processes, use the renice(8) command. For example:

```
/etc/renice +5 -u jones
```

This command slows down all processes that are owned by Jones.

For further information, read the renice(8) command documentation.

4.5. Manage System Information

During daily operations, your system accumulates system accounting information that you can use in evaluating day-to-day operations. With this information, you can keep track of:

- User logins

- Command usage
- Printer/plotter usage

4.5.1. User Logins

The system automatically maintains two login accounting files: /etc/utmp and /usr/adm/wtmp. The system records all active logins in /etc/utmp and accumulates a user login history in /usr/adm/wtmp.

Periodically, you should generate a report of user logins. To generate this report, use the ac(8) command. For example:

```
/etc/ac
```

This command displays a report of the system's current login history (/usr/adm/wtmp).

Over time, /usr/adm/wtmp increases in size. To manage space on your /usr file system, you should periodically print out a copy of this report and condense /usr/adm/wtmp. To condense /usr/adm/wtmp, use th cp(1) command with the following arguments specified. For example:

```
cp /dev/null /usr/adm/wtmp
```

This command copies /dev/null to /usr/adm/wtmp, that is, reduces /usr/adm/wtmp to a zero length file.

For further information, read the ac(8) command documentation.

Note: The system automatically enables login accounting, but it accumulates a login history only if /usr/adm/wtmp exists. To disable the system login history, remove /usr/adm/wtmp.

4.5.2. Command Usage

During multi-user startup, /etc/rc normally enables system process accounting. When process accounting is enabled, the system records information on each executed process (command) in /usr/adm/acct.

Periodically, you should generate a report listing command usage. To generate this report, use the sa(8) command. For example:

```
/etc/sa
```

This command displays the contents of the system's current process accounting file, /usr/adm/acct.

Over time, /usr/adm/acct increases in size. To manage space on your /usr file system, you should periodically condense the process accounting information. To con-

dense /usr/adm/acct, use the sa(8) with the −s option specified. For example:

```
/etc/sa -s
```

This command merges the current information in /usr/adm/acct into the process history file, /usr/adm/savacct.

For further information, read the sa(8) command documentation.

Note: If you want to disable process accounting, remove /usr/adm/acct.

4.5.3. Printer/Plotter Usage

Your system normally records all printer/plotter information in the default accounting file named in /etc/printcap.

Periodically, you should generate a report of your printer/plotter usage. To generate this report, use the pac(8) command. For example:

```
/etc/pac
```

This command displays a report detailing usage per user: number of pages printed, feet of paper consumed, and total estimated cost.

For further information, read the pac(8) command documentation.

Note: The system enables printer/plotter accounting only if /etc/printcap names a default accounting file. If you do not want printer/plotter accounting enabled, remove the entry in /etc/printcap that names this file.

4.5.4. Active System Information

In addition to those commands that are used to display accumulated system accounting information, the system has a number of commands that you can use to display active system statistics:

iostat(1)	To display a report of current I/O statistics
ps(1)	To display a report of the system's process status
uptime(1)	To display a report of how long the system has been up
vmstat(1)	To display a report of virtual memory statistics
w(1)	To display a report of currently active user and what they are doing
pstat(8)	To display various system tables

For further information, read the iostat(1), ps(1), uptime(1), vmstat(1), w(1), and pstat(8) command documentation.

5. Guidelines for a Crash Recovery

This section provides discussions of what happens when your system crashes and what the most likely causes are.

5.1. When Your System Crashes

Before coming down, the system writes a panic message to the console and attempts to update all file system information. Then, it writes two files to the end of the primary swap area. The first, vmcore.*n*, is a dump of memory. The second, vmunix.*n*, is a copy of the kernel image, vmunix. Finally, the system attempts to reboot.

During a reboot, the system automatically checks for file system inconsistencies. If fsck(8) exits without notifying you of unexpected inconsistencies, the system continues to reboot multi-user mode. If during the reboot /etc/rc.local invokes savecore(8), the system copies the dump from the primary swap area to files in the specified directory.

Currently, the distributed /etc/rc.local contains an entry for savecore(8). However, in an effort to make savecore(8) a site-dependent option, this entry is commented out. That is, as distributed, /etc/rc.local currently contains an entry for but does not invoke savecore(8).

If you choose to enable savecore(8) and create crash dumps, you should:

- Verify sufficient space on /usr
- Remove comment characters from savecore(8) entry

  or

- Determine new core file directory (file system)
- Make new core file directory
- Change directory argument for savecore(8) entry

Although these dump files may be helpful in determining the cause of the crash, they will use up space on the specified file system. To save space on this file system as well as to create a permanent record of these dump files, you should copy them to tape and then remove them from the specified directory.

To create a permanent copy of these core files, use the tar(1) command to extract these files. For example:

tar *key path*/vmunix.*n path*/vmcore.*n*

This command copies the specified core files to tape.

Then, to conserve space on the specified file system, use the rm(1) command to

remove them. For example:

rm *path*/vmunix.*n* *path*/vmcore.*n*

This command removes the specified files and frees up space on that file system.

For further information, read the rm(1), tar(1), reboot(8), and savecore(8) command documentation.

5.2. Panic Messages -- Most Likely Causes

The system panic messages are intended to provide you with some indication of what caused your system to crash.

During daily operations, the system constantly monitors its own internal status. It performs an number of internal consistency checks, and, if one of these checks fail, the system voluntarily crashes.

The following provides a brief list of the most common panics.

IO err in push
hard IO err in swap

Cause: The system encountered an I/O error either when it attempted to read file system information from a disk or when it attempted to write to the paging device.

timeout table overflow

Cause: Currently, if this table runs out of entries, the system crashes.

KSP not valid
SBI fault
CHM? in kernel

Cause: These messages may result either from a hardware failure or from a inconsistency in the system.

trap type %d, code=%d, pc=%x

Cause: The system encountered the specified error trap. Possible trap types include:

0 reserved addressing fault
1 privileged instruction fault
2 reserved operand fault
3 bpt instruction fault
4 xfc instruction fault
5 system call trap
6 arithmetic trap
7 ast delivery trap
8 segmentation fault
9 protection fault

10 trace trap
11 compatibility mode fault
12 page fault
13 page table fault

init died

Cause: The system initialization program, init(8), must be running (but not necessarily active) at all times.

6. Guidelines for File System Consistency

This section provides discussions of how file system inconsistencies occur, how during daily operations they normally are corrected, and how you should proceed when fsck(8) cannot correct them easily or cleanly.

6.1. File System Inconsistencies

File system inconsistencies most often occur when your system crashes. A power failure, a hardware failure, a software error, or a human error can bring your system down, and a system crash is likely to produce file system inconsistencies.

Before it comes down, the system attempts to update all file system information. For efficiency, the system keeps copies of this information for all active file systems in memory. The system's in-memory buffer cache contains all free block lists, free inode lists, modified data blocks, modified inodes, and modified super blocks 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, update(8) wakes up every 30 seconds and invokes sync(2). When it crashes, the system may not completely update the disk resident file system information. Then, inconsistencies between this summary information and the file system's actual status exist.

For further information, read the sync(2) and update(8) command documentation.

Note: For detailed description of the ULTRIX-32 file system, read "A Fast File System for Unix" in *ULTRIX-32 Supplementary Documentation, Volume III.*

6.2. The fsck Command: Invoked by /etc/rc

Normally, your file systems are checked for inconsistencies every time that you reboot. During a reboot, /etc/rc automatically invokes fsck(8). When it is so invoked, fsck(8) checks for and corrects only those inconsistencies that can be corrected easily and cleanly.

If fsck(8) encounters inconsistencies that cannot be corrected easily or quickly, /etc/rc exits multi-user startup, and your system remains in single-user mode. When this occurs, you should reinvoke fsck(8) for interactive execution and correct these inconsistencies immediately.

6.3. The fsck Command: Interactive Execution

When invoked for interactive execution, fsck(8) checks your file systems. Then, as it encounters each inconsistency, fsck(8) not only displays a diagnostic message that indicates the type of inconsistency found, but it also prompts you for a response to its suggested corrective action. You can answer either yes or no to this prompt.

6.3.1. A Yes Response

When you answer yes, fsck(8) proceeds with its suggested course of action and attempts to correct the inconsistency. When necessary, fsck(8) relinks all allocated but unlinked files to the lost+found directory for the appropriate file system. When it relinks a file, fsck(8) names it by its inode number.

When fsck(8) relinks a file, you should try to determine its owner and the directory in which it belongs. By using ls(1) with the −l option specified, you can determine relevant information from the inode. By using file(1), you can determine its text file type. By talking with the owner, you can try to determine which directory it belongs in. After determining its appropriate directory, you should either link the file to it or remove the file from the lost+found directory.

Note: The fsck(8) command requires a lost+found directory in each file system. Normally, newfs(8) automatically creates this directory in each file system. However, if during operations one of these directories are inadvertently removed, use the mklost+found(8) command make this required directory.

6.3.2. A No Response

When you answer no, fsck(8) leaves the inconsistency uncorrected but continues checking the file system(s). When unsure of the consequences of your response, you should answer no. Although this will leave the inconsistency uncorrected, it will create an summary from which you can better decide on a course of action.

For further information, read "Fsck -- The Unix File System Check Program" in *ULTRIX-32 Supplementary Documents, Volume III* and the fsck(8) command documentation.

Notes: 1. If fsck(8) tells you to reboot the system after correcting the root file system, you should enter a <CTRL/P> immediately. This command returns you to the console subsystem and allows you to boot multi-user mode.

2. The fsck(8) command has made the other distributed file system maintenance commands virtually obsolete. However, for further information, read the clri(8), dcheck(8,) dumpfs(8), icheck(8), and ncheck(8) command documentation.

7. System Overview

This section provides a quick overview of the file system hierarchy of your distributed ULTRIX-32 operating system.

7.1. Root File System

/ root directory for root file system

/bin directory for utility programs (see also /usr/bin)

as
  assembler
cc
  C compiler executive (see also /lib/ccom, /lib/cpp, /lib/c2)
csh
  C shell
    .
    .
    .

/dev directory for devices (4)

MAKEDEV
  shell script to create special files
MAKEDEV.local
  site specific part of MAKEDEV
console
  main console
hp*
  disks
rhp*
  raw disks
ra*
  UNIBUS disks
tty*
  terminals
    .
    .
    .

/etc directory for maintenance programs and administrative files

ac
  login accounting program
chown
  change file owner program
cron
  clock daemon

disktab
  disk characteristics and partition table
dump
  dump program
dumpdates
  dump history file
edquota
  edit user quotas program
fsck
  file system consistency check program
fstab
  file systems table
getty
  terminal mode program
gettytab
  terminal configuration table
group
  groups file
hosts
  host name to network address mapping file
init
  system initialization program (parent of all processes)
lpc
  line printer control program
lpd
  line printer daemon
motd
  message-of-the-day file
mount
  mount program
mtab
  mounted file system table
networks
  network name to network number mapping file
newfs
  construct new file system program
pac
  printer/plotter accounting program
passwd
  password file
quotacheck
  program that verifies (checks) file system quotas
quotaon
  program for turning on file system quotas

quotaoff
  program for turning off file system quotas
phones
  phone numbers for remote hosts file
printcap
  line printer configuration table
protocols
  name to number mapping file
rc
  shell script that brings system to multi-user mode
rc.local
  shell script that is site dependent portion of *rc*
remote
  names and description of remote hosts file
renice
  alter priorities program
repquota
  program for summarizing file system quotas
restore
  restore program
sa
  system accounting program
savecore
  save core dump program
sendmail
  send mail program
services
  network services definition file
shutdown
  shut down multi-user mode program
sync
  file system synchronization program
termcap
  terminal capabilities file
ttys
  terminal initialization file
ttytype
  terminal type table
umount
  unmount file systems program
update
  update file systems program
utmp
  current login history file

vipw
edit password file program
.
.
.

/lib directory object libraries (see also /usr/lib)

ccom
C compiler proper
cpp
C preprocessor
c2
C code improver
libc.a
system calls and standard I/O (2,3,3S)
.
.
.

/lost+found directory for reconnecting unlinked files from root file system

/sys symbolic link, normally to /usr/sys

/tmp directory for temporary files (see also /usr/tmp)

e*
temp files for *ed*(1)
ctm*
temp files for *cc*(1)
.
.
.

/usr general purpose directory, normally on which the /usr file system is mounted (see description below)

/vmunix kernel image

7.2. /usr File System

/usr root directory for /usr file system

/usr/adm directory for administrative information

crash
directory for crash dump files
vmcore.?,vmunix.?
crash dump files

lpacct
  line printer accounting file
messages
  hardware error messages file
tracct
  phototypesetter accounting file
vaacct, vpacct
  varian and versatec accounting file
wtmp
  login history file

.
.
.

/usr/bin directory for utility programs (keeps /bin small)

/usr/dict directory for word lists

spellhist
  history file
words
  word list

.
.
.

/usr/doc directories containing files for the Vol.2 documentation

as
  assembler manual
c
  C manual

.
.
.

/usr/games directory for games

hangman
  hangman game
lib
  library directory for games

.
.
.

/usr/guest directory for guest accounts

/usr/include directory for standard #include files

a.out.h
  object file layout
math.h
  math (3M)
stdio.h
  standard I/O
sys
  symbolic link to /sys/h (system generation #include files)

.
.
.

/usr/lib directory for object libraries (keeps /lib small)

atrun
  scheduler for *at*(1)
aliases
  user aliases file for *mail*(1)
crontab
  system clock daemon table
font
  directory for *nroff*(1) and *troff*(1) fonts
lint
  directory for *lint*(1) utility files
tmac
  directory for *troff*(1) macros
units
  directory of *units*(1) conversion tables
uucp
  directory for *uucp*(1C) programs and data

.
.
.

/usr/lost+found
  directory for reconnecting unlinked files from /usr file system.

/usr/man directory for unformatted and preformatted man pages

man1
  directory for section 1 (unformatted)
man2
  directory for section 2 (unformatted)
man3
  directory for section 3 (unformatted)

.
.

cat1
  directory for section 1 (preformatted)
cat2
  directory for section 2 (preformatted)
cat3
  directory for section 3 (preformatted)

.
.
.

/usr/mdec directory for ULTRIX-32 boot files

/usr/msgs directory for *msgs*(1) messages

/usr/new directory for binaries of new versions of programs

/usr/preserve directory for editor temp files preserved after crashes/hangups

/usr/pub directory for binaries of user programs

/usr/skel directory for sample startup files

.cshrc
  csh(1) startup file
.login
  csh(1) login startup file
.mailrc
  mail(1) startup file
.profile
  sh(1) startup file
.project
  finger(1) information file

/usr/spool directory for delayed execution files

at
  directory used by *at*(1)
lpd
  directory used by *lpr*(1)
    lock
      present when line printer is active
    cf*
      copy of file to be printed, if necessary
    df*
      line printer daemon control file
    tf*
      transient control file

mail
  directory for *mail*(1) mailboxes
    name
      mail file for *name* user
    name.lock
      lock file (exists while *name* is receiving mail)
uucp
  directory for *uucp*(1) work files
    LOGFILE
      summary log

/usr/src directory for generic sources

usr.bin
  directory for user sources
    troff
      directory for nroff and troff sources
        term
          directory of description files for new printers

/usr/sys directory for system files

BINARY
  directory for *make*(1) system object files
cassette
  directory of files for boot cassette
conf
  directory of *config*(8) configuration files
data
  directory for drive partition tables
floppy
  directory of files for floppy disk
h
  directory for system #include files
mdec
  directory of headers for 11/750 boot blocks
net
  directory for general network files
netimp
  directory for IMP network files
netinet
  directory for DARPA internet network files
netpup
  directory for PUP network files
stand
  directory for standalone boot binaries

sys
  directory for machine dependent system files
vax
  directory for VAX specific system files
vaxif
  directory of network interface drivers for the VAX
vaxmba
  directory of drivers for devices on the MASSBUS
vaxuba
  directory of drivers for devices on the UNIBUS

/usr/tmp symbolic link to /tmp

8. ULTRIX-32 Special Files

Section 4 of the *ULTRIX-32 Programmer's Manual Binder 3B* provides documentation for the special files. It discusses:

acc(4)	- ACC LH/DH IMP interface
ad(4)	- Data Translation A/D converter
arp(4P)	- DARPA Internet address resolution protocol
autoconf(4)	- Diagnostics from the autoconfiguration code
bk(4)	- Line discipline for machine-machine communication
cons(4)	- Console interface
css(4)	- DEC IMP-11A LH/DH IMP interface
ct(4)	- Phototypesetter interface
de(4)	- DEUNA 10 Mb Ethernet Interface
dh(4)	- DH-11/DM-11 communications multiplexer
dmc(4)	- DEC DMC-11/DMR-11 point-to-point communications device
dmf(4)	- DMF-32, terminal multiplexor
dn(4)	- DN-11 autocall unit interface
drum(4)	- Paging device
dz(4)	- DZ-11 communications multiplexer
ec(4)	- 3Com 10 Mb/s Ethernet interface
en(4)	- Xerox 3 Mb/s Ethernet interface
fl(4)	- Console floppy interface
hk(4)	- RK6-11/RK06 and RK07 moving head disk
hp(4)	- MASSBUS disk interface
ht(4)	- TM-03/TE-16,TU-45,TU-77 MASSBUS magtape interface
hy(4)	- Network Systems Hyperchannel interface
ik(4)	- Ikonas frame buffer, graphics device interface
il(4)	- Interlan 10 Mb/s Ethernet interface
imp(4)	- 1822 network interface
imp(4P)	- IMP raw socket interface
inet(4F)	- Internet protocol family
intro(4)	- Introduction to special files and hardware support
intro(4N)	- Introduction to network facilities
ip(4P)	- Internet protocol
kg(4)	- KL-11/DL-11W line clock
kmem(4)	- Main memory
lo(4)	- Software loopback network interface
lp(4)	- Line printer
mem(4)	- Main memory
mt(4)	- TM78/TU-78 MASSBUS magtape interface
mtio(4)	- UNIX magtape interface
null(4)	- Data sink
pcl(4)	- DEC CSS PCL-11 B Network Interface

ps(4) - Evans-Sutherland Picture System 2 graphics interface
pty(4) - Pseudo terminal driver
pup(4F) - Xerox PUP-I protocol
pup(4P) - PUP raw socket interface
rx(4) - DEC RX02 floppy disk interface
tcp(4P) - Internet transmission protocol
tm(4) - TM-11/TE-10 magtape interface
ts(4) - TS-11 magtape interface
tty(4) - General terminal interface
tu(4) - VAX 11/730 and 11/750 TU58 console cassette interface
uda(4) - UDA-50 disk controller interface
udp(4P) - Internet user datagram protocol
un(4) - Ungermann-Bass interface
up(4) - Unibus storage module controller/drives
ut(4) - UNIBUS TU45 tri-density tape drive interface
uu(4) - TU58/DECtape II UNIBUS cassette interface
va(4) - Benson-Varian interface
vp(4) - Versatec interface
vv(4) - Proteon proNET 10 Megabit ring

9. ULTRIX-32 Maintenance Commands

Section 8 of the *ULTRIX-32 Programmer's Manual Binder 3B* provides documentation for the maintenance commands. It discusses:

ac(8) - Login accounting
accton(8) - System accounting
analyze(8) - Virtual UNIX postmortem crash analyzer
arcv(8) - Convert archives to new format
arff(8) - Archiver/copier for floppy
badsect(8) - Create files to contain bad sectors
bugfiler(8) - File bug reports in folders automatically
catman(8) - Create cat(1) files for manual
chown(8) - Change owner
clri(8) - Clear file i-node
comsat(8) - Biff server
config(8) - Build system configuration files
cron(8) - Clock daemon
dcheck(8) - File system directory consistency check
diskpart(8) - Calculate default disk partition sizes
dmesg(8) - Collect system diagnostic messages
drtest(8) - Standalone disk test program
dump(8) - Incremental file system dump
dumpfs(8) - Dump file system information
edquota(8) - Edit user quotas
fastboot(8) - Reboot system without checking disks
fasthalt(8) - Halt system without checking the disks
format(8) - Format disk packs
fsck(8) - File system consistency check and repair
ftpd(8) - DARPA Internet file transfer protocol server
gettable(8) - Get NIC format host table
getty(8) - Set terminal mode
halt(8) - Stop the processor
htable(8) - Convert NIC standard format host tables
icheck(8) - File system storage consistency check
implog(8) - IMP log interpreter
implogd(8) - IMP logger
init(8) - Process control initialization
intro(8) - Introduction to system maintenance and operation commands
kgmon(8) - Generate dump of system's profile buffers
lpc(8) - Line printer control program
lpd(8) - Line printer daemon
makedev(8) - Make system special files
makekey(8) - Generate encryption key

mkfs(8) - Construct a file system
mklost+found(8) - Make a lost+found directory for fsck
mknod(8) - Build special file
mkproto(8) - Construct a prototype file system
mount(8) - Mount file system
ncheck(8) - Generate names from i-numbers
newfs(8) - Construct a new file system
pac(8) - Printer/ploter accounting information
pstat(8) - Print system facts
quot(8) - Summarize file system ownership
quotacheck(8) - File system quota consistency checker
quotaon(8) - Turn file system quotas on
quotaoff(8) - Turn file system quotas off
rc(8) - Command script for auto-reboot and daemons
reboot(8) - UNIX bootstrapping procedures
rdump(8) - File system dump across network
renice(8) - Alter priority of running processes
repquota(8) - Summarize quotas for a file system
restore(8) - Incremental file system restore
rexecd(8) - Remote execution server
rlogind(8) - Remote login server
rmt(8) - Remote magtape protocol
route(8) - Manipulate routing tables
routed(8) - Network routing daemon
rrestore(8) - Restore file system across network
rshd(8) - Remote shell server
rwhod(8) - System status server
rxformat(8) - Format floppy disks
sa(8) - System accounting
savecore(8) - Save a core dump of the operating system
sendmail(8) - Send mail over the internet
setifaddr(8) - Set network interface address
shutdown(8) - Close down the system at a given time
sticky(8) - Executable files with persistent text
swapon(8) - Specify additional device for paging and swapping
sync(8) - Update the super block
syslog(8) - Log systems messages
telnetd(8) - DARPA TELNET protocol server
tftpd(8) - DARPA file transfer protocol server
trpt(8) - Tranliterate protocol trace
tunefs(8) - Tune an existing file system
umount(8) - Unmount file system
update(8) - Update file system super block

uuaids(8)	- UUCP administrative utilities
uuclean(8)	- Clean UUCP spool directory
uumon(8)	- Monitor UUCP system
vipw(8)	- Edit the password file

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA and Puerto Rico call **800–258–1710**

In Canada call **800–267–6146**

In New Hampshire, Alaska or Hawaii call **603–884–6660**

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.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION
A&SG 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, Northboro, Massachusetts 01532

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

ULTRIX - 32
Programmer's Manual
AA-BG59A-TE

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

Documentation Manager
ULTRIX-32™ Documentation Group
MK02-1/H10
Continental Blvd.
Merrimack, N.H.
03054

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line

Notes:

Notes:

Notes:

Notes:

digital™