ULTRIX-32 Guide to System Disk Maintenance

Order No. AA-ME93A-TE

ULTRIX-32 Operating System, Version 3.0

Digital Equipment Corporation

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

Contents

About This Manual

1 System Disks

Figures

Tables

About This Manual

The objective of this manual is to provide you with information on handling disk space. The guide presents background information on system disks, describes how to partition disks, and explains dynamic bad block replacement.

Audience

The Guide to System Disk Maintenance 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(1) or ed(1). You do not need to be a programmer to use this guide.

Organization

This manual consists of four chapters, an appendix and an index. The chapters are:

Chapter 1: System Disks
Provides an overview on disks and describes default values for system disks.

Chapter 2: Managing Disk Space
Explains how to check disk space and use, and how to obtain additional space on a disk.

Chapter 3: Disk Partitioning
Explains how to partition disks using the chpt command.

Chapter 4: Bad Block Replacement
Describes how the system replaces bad blocks and explains how you can force bad block replacement, if necessary.

Related Documents

You should have the hardware documentation for your system and peripherals.

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 cat(1) command. This indicates that you can find the material on the cat 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.
italics	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-32 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 Disks 1

This chapter provides background information about the defaults established on your system disk when you perform a basic installation. If you change your system disk defaults during an advanced installation or by using the chpt command, some of the information in this chapter may no longer apply. Refer to chpt(8) in the ULTRIX Reference Pages and Chapter 3 of this manual for more information on changing disk partitions.

The following sections describe:

- Disk organization
- Disk partition defaults
- Selecting disk partitions
- Paging and swapping
- Allocating swap space
- File system organization
- Setting up /var
- System directories
- User directories

1.1 Disk Organization

A disk consists of storage units called sectors. A disk sector is usually 512 bytes, and sectors are grouped into a maximum of eight partitions. However, not all disks contain the same number of partitions or the same partition sizes. See ra(4,) hp(4,) and rd(4) in the ULTRIX Reference Pages for a listing of supported disks and partition sizes.

Disk partitions form logical disk boundaries that separate each file system. Each partition can contain one file system, and several file systems can reside on the same disk. A file system can be smaller but not larger than the partition size.

1.1.1 Disk Partition Defaults

When you perform a basic installation, the install procedure establishes certain disk partition defaults for you. For example:

- Partition a of the system disk (usually drive 0) contains the root file system.
- Partition b of the system disk is reserved for use as a paging and swapping area, and for crash dumps.
- Partition g of most system disks is reserved for the /usr file system. (On RA82 and RA90 disks, the default area for /usr is partition d.) If partition g of the system disk is too small to contain the /usr file system, some subdirectories of /usr can be placed on another partition or, if available, on another disk.
- Partition h of the system disk is reserved for the /usr/users file system. However, if the system disk does not have a partition h, then the default is to put /usr/users on partition g (subordinate to the /usr file system).

1.1.2 Selecting Disk Partitions

By selecting the file systems to be placed in each partition, you can monitor growth and activity of the disk. If you have multiple disks, you can divide users across the disks for better input and output response. You can also accommodate growth by placing the file systems you expect to expand on partitions that are large enough to accommodate that growth. See Chapter 3 of this manual for details on selecting and changing disk partitions.

1.2 Paging and Swapping

The kernel needs disk space for paging and swapping to provide a time-sharing environment and virtual memory to VAX users. Swapping occurs when there is not enough physical memory for a process or processes in the system. Paging occurs to bring data into memory, to copy data out of memory, or to accommodate other processes that require physical memory.

Programs do not have to be written to fit within certain boundaries of physical memory. The hardware is designed to process programs in lengths called pages. If a particular page of memory is not modified during processing, it is overwritten when that memory is needed for something else. When the page of memory is needed again, it can be read in from the original file on disk. If a page of memory is modified during processing, it is written out to disk so that the copy on disk reflects the page in memory. This writing and reading of modified pages of memory is one of the ways the system uses the paging and swapping area on disk.

The kernel keeps as many processes (programs being executed by users) in physical memory as it can, and lets each one execute during its own particular time slice. Swapping occurs when a new process needs to be loaded and executed, but there is no room for it in memory. An old process is swapped out to disk, to wait its turn for processing again. When that time comes, the process is swapped into memory to continue.

1.2.1 Allocating Swap Space

The area reserved for swapping should be substantially larger than the total available physical memory in the system. A rule of thumb is to give at least two times the amount of swap space as you have physical memory. If the swap area is too small, the system will not be able to make use of all the available physical memory. If you receive messages indicating there is not enough memory, you probably need a larger partition size for the swap area.

By default, the installation procedure automatically allocates partition b of the system disk for paging and swapping. If you determine that you need more paging and swapping space, you could use a different partition. If this is impossible, you could change the size of partition b on the system disk by using the chpt command as described in Chapter 3. You do not usually need to change the partitions of the disk. The existing disk partitions will generally meet the needs of your system's users.

You can also use a second partition on a second disk for paging and swapping. In this case, the kernel interleaves its paging and swapping operations between the two disks, and your system can run faster than it would with only one disk. This is done by making the necessary assignments in the configuration file. For information on how to define this configuration, refer to the Guide to System Configuration File Maintenance.

Local paging and swapping is available to clients who have a local disk and are operating in a diskless environment. Refer to the Guide to Diskless Management Services for details.

Note

> Avoid selecting partition a of any disk for use as a swap partition. Partition information for the entire disk resides in the superblock of the a partition's file system. If a customized partition table has previously been defined for the disk, the information will be destroyed when other data is swapped to partition a.

1.3 File System Organization

ULTRIX-32 supports two file systems: the ULTRIX File System (UFS), and the Network File System (NFS). This section describes the ULTRIX file system. For a discussion of NFS, see the Guide to the Network File System.

The root file system is divided into sharable data, /usr, and nonsharable data, root (/). This separation by data use enables server systems to share the /usr file system efficiently with client systems.

The file systems are further divided into the following content areas: static data files, variable data files, and executable data files.

- Static data files remain fairly constant over time. Sharable static data, like library routines, documentation macros, and uucp files are in /usr/lib which is symbolically linked to /lib. The /usr/etc directory contains sharable system maintenance tools and daemons. Nonsharable, machine specific data files, such as aliases, crontab, and sendmail.cf are in /etc.
- Variable data files are dynamic files that change in size, such as spool files. Variable data files reside in /var or /usr/var. On server systems where the user files are shared with diskless clients, all variable data files should be in /var, and /var should be set up as a separate file system.
- Executable data files are machine architecture-specific files. All but a select set of executables can be shared by most systems. Shared executables, such as ls, mail, and pwd are in /usr/bin. Single user executables, such as init and fsck are in /bin.

The file system organization simplifies system management by placing volatile files of the same type in the same directories. Separating files by content simplifies the system management task. Unlike the area reserved for variable data, the areas reserved for executables or static data will not change in size. This organization also greatly improves system debugging, since all the logging files are contained in the variable areas.

1.3.1 Setting Up /var

When you perform the basic installation, there are only two file systems, / and /usr. Because of the space limitations in this configuration, /var does not exist. Files and directories that would normally reside in /var are located in /usr/var, which is symbolically linked to /var.

When you perform an advanced installation, the software asks you if you want to set up a separate file system for /var. If you also set up /usr/users as a separate file system, you can mount /usr as a read only file system. This file system organization is preferred if you are setting up a system that will share /usr. It provides the most secure access to shared files and simplifies system management tasks by limiting extremely dynamic files to one area.

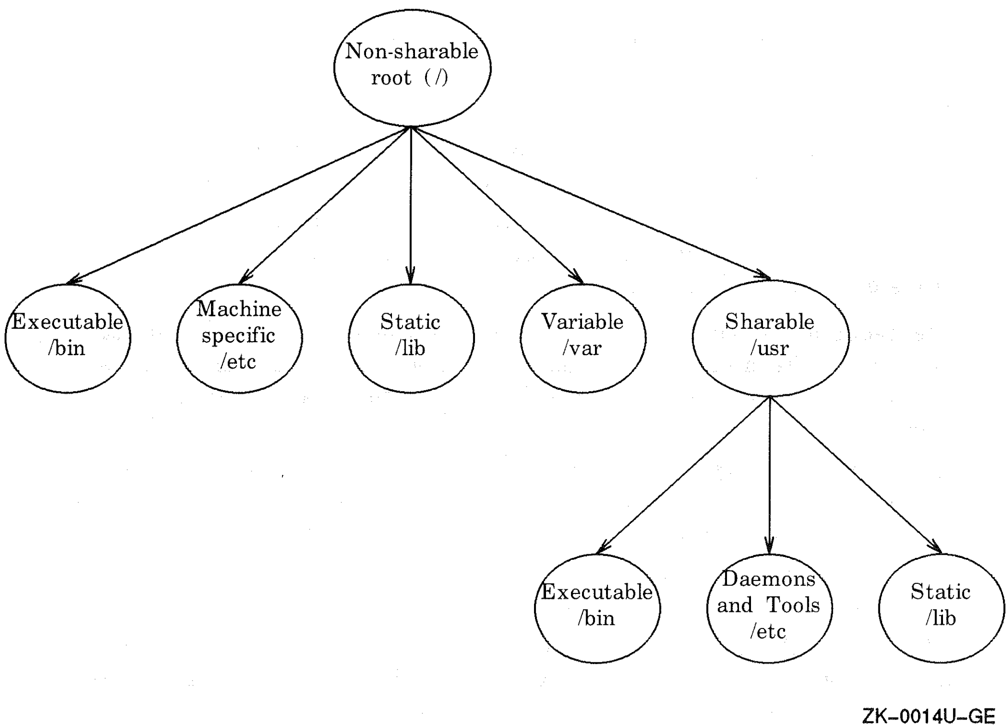


Figure 1-1: File System Organization

1.4 System Directories

In addition to the directories shown in Figure 1-1, the system needs certain directories for standard operations. The following list describes these directories, their use, and - where relevant - the recommended space allocation.

- The /var/adm directory contains data files generated by administrative programs such as system accounting and the error logger. The data and files in /var/adm can vary widely across systems and over time. The /var/adm/acct file, for example, can easily grow by 5 Kbytes a day. In addition, /var/adm is the default location for the system crash directory, /var/adm/crash.

  We recommend that /var/adm have a partition size equivalent to roughly two times the amount of memory on your system.

- The /var/spool directory contains files being held for spooling. For example, files for the line printer, mail messages, or network file transfers are held in this directory.

  We recommend that you monitor the variable growth of /var/spool to determine the amount of space needed at your site. If your users place heavy demands on these facilities, then allocate space accordingly.

- The /tmp directory is used by various system and user programs for temporary files. For example, the vi editor creates a temporary file in the /tmp directory.
- The /usr/etc directory contains sharable system maintenance tools and daemons, such as config, talkd, and nfsd.
- The /usr/hosts directory contains a file for each node on the local area network. The /usr/hosts/MAKEHOSTS command creates these files, which are linked to /bin/rsh, the remote shell program. Each of these files typically requires one Kbyte of disk space.
- The /usr/man directory contains the source for documents printed in the ULTRIX Reference Pages.
- The /usr/users is the default for the home directory for each of the system users. In an environment where /usr is shared with client systems, /usr/users should be set up as a separate file system. If you installed your system with the Advanced Installation, you were provided with the option of creating a separate file system for /usr/users. See the following section, User Directories, for more information on this file system.

1.5 User Directories

By default, the login (home) directory for users is subordinate to /usr/users. You can change this default so /usr/users contains one or more subdirectories for your users. This section explains how to reorganize the /usr/users file system according to the requirements of your users.

In general, any user who logs in to the system has access to the files in his home directory.

- On local systems, the default locations for individual home directories are in subdirectories of /usr/users. For example, a user named mjadams could have a home directory with the pathname, /usr/users/mjadams.
- On a remote client system that shares the /usr/users file system, users can mount the home directory on any established mount point.
- On client systems that run Yellow Pages, the home directory mount point on the client should have the same pathname as the pathname for the home directory on the server system. For example, if the pathname of the home directory, rjones, is /usr/users/rjones, you must create a directory for /usr/users/rjones on your client system as the mount point. Because Yellow Pages systems share the password file, the home directory pathname, userid, and groupid must match on the client and the server systems. Refer to the Guide to the Yellow Pages Service for more information on Yellow Pages.

1.5.1 Creating a User Directory Tree Structure

Changing the default structure requires that you know how to set up directory tree structures and that you understand the needs of your users. You can create a directory tree structure for users in the /usr directory with as few or as many branches as you want. Some sites, for example, create home directories for all users within one subdirectory, as shown in Figure 1-2. If the space required by all your users fits on one disk partition, there is no reason to divide the directories. Other sites may require more complex allocations, as shown in Figure 1-3. If you need more space for /usr/users, you can move the file system to another partition. (For more information on moving file systems, see Chapter 2.)

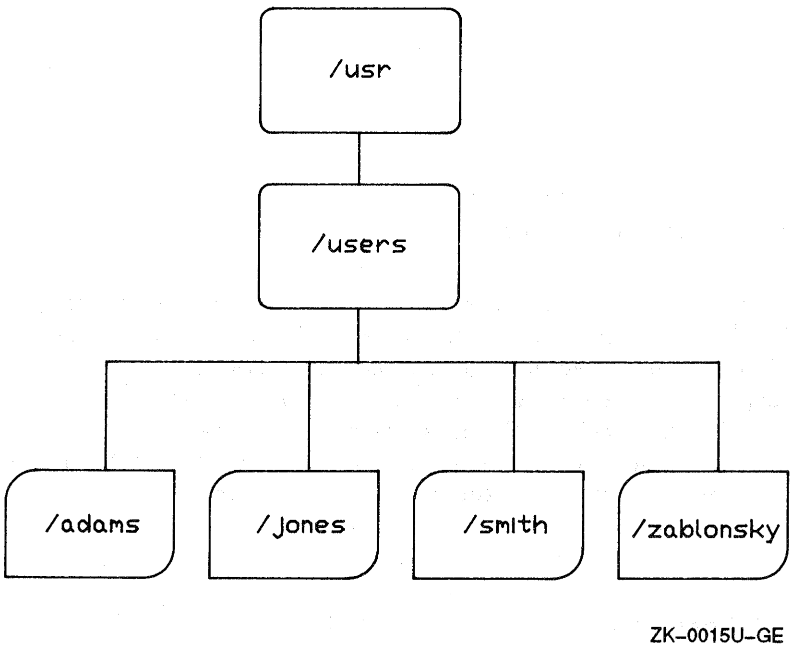
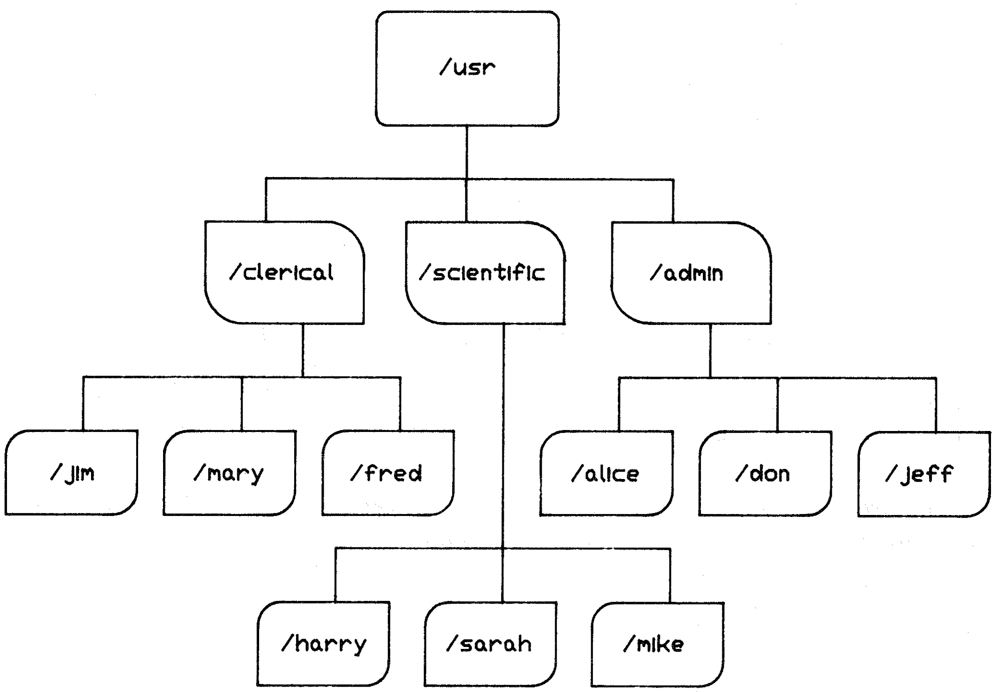


Figure 1-2: Subdirectories of /usr, One-Branch Tree

1.5.2 Sample User Directory Tree Structure

This section explains how you can organize user directories into three subdirectories for three types of users, all with different requirements. This example is illustrated in Figure 1-3.

In this example, users are divided into three categories according to disk use: clerical, scientific, and administrative.



ZK-0016U-GE

Figure 1-3: Subdirectories of /usr, Three-Branch Tree

1.5.2.1 Clerical Users – Assume that there are 10 clerical workers on the system, and they each have at least 25 files on hand at any given time. These users generate nroff source and output files for memos, papers, and reports. We will assume that these files require an average of 30 Kbytes each.

The average load, in this case, is 7.5 Mbytes, which you could put on an 8-Mbyte partition. To allow for growth and periods of high production, you could enlarge the size of an 8-Mbyte partition or allocate the directory to a larger partition for this group.

1.5.2.2 Scientific Users – Assume there are 15 graduate students in the group of scientific users and they generate C program files — source, object, and executable. These users keep an average of 10 modest-sized programs (20 Kbytes for source, 10 Kbytes for object, and 15 Kbytes for executable programs), totaling 6.75 Mbytes. Also assume there is one professor who keeps 10 of his programs and data bases around at any given time (he generates huge data files of about 1,000 Kbytes); he would require around 11.2 Mbytes. The total requirement for the scientific staff would be around 18 Mbytes. Allowing for growth, the scientific staff might require a medium-sized partition on a large disk (partition d or e on an RA81 disk, for example).

1.5.2.3 Administrative Users – The administrative staff's needs are also fairly large, but they have a special security requirement. They keep salary and personnel files that must remain private. To maintain this privacy, they should be assigned a partition on a disk with removable packs.

Use the newfs command to create new file systems on partitions that are large enough to hold them and any anticipated growth. Refer to newfs(8) in the ULTRIX Reference Pages. You should keep in mind, however, that the size of a file system is defined by the size of the partition on which it resides. If you put a small file system into an excessively large partition, the file system would be wasting the excess disk space on that partition. For example, on RP07 disks, you might have mounted a large file system on partition g because this partition overlaps partitions d, e, and f. By using partition g, the file system will have more room than if you use partition d, e, or f. To review the current disk partition setup, use the chpt program. See Chapter 3 of this manual and the ULTRIX Reference Pages for details on the chpt command.

Managing Disk Space 2

This chapter discusses methods of monitoring file system use and managing file system data. It explains how to check the disk space and use and describes different methods you can use to obtain additional disk space.

2.1 Monitoring File System Use

To ensure an adequate amount of free disk space, regularly monitor the disk use of your configured file system. To ensure adequate free disk space:

- Check available free space using df
- Check disk use using du -s
- Verify disk quotas (if imposed) using quot

2.1.1 Checking Available Free Space

To ensure sufficient free space for your configured file systems, you should regularly check the amount of free disk space available in all of the mounted file systems. To see how much free disk space there is, use the df(1) command. This command reports the amount of free space available on all of the currently mounted file systems. When invoked, the df command generates a listing similar to the following:

```
# df

Filesystem      total    kbytes   kbytes   percent
   node         kbytes    used     free    used     Mounted on
/dev/ra0a        7429     6141      546     92%     /
/dev/ra1a        7429      437     6250      7%     /tmp
/dev/ra1g      175015   143520    13994     91%     /usr
/dev/ra0e       30519    14342    13126     52%     /usr/spool
/dev/ra0h      313233   134316   147594     48%     /usr/staff
```

For each file system, the df command reports the file system's configured size (Kbytes), the amount presently used, the amount presently available (free), the percentage used, and the directory on which the file system is mounted.

You can also use the df command with the -i option to display both the free and the used inodes. For more information on the df command and its options, see df(1) in the ULTRIX Reference Pages.

File systems are usually configured with a minimum percentage of free space established. When constructing a new file system, the newfs command reserves a minimum percentage of the space (the default is 10 percent). This default percentage allows for a report in excess of 100 percent. When interpreting the free space report, therefore, you should look for significant changes in file system disk use. A file system is considered to have insufficient space when the percentage used exceeds 90.

2.1.2 Checking Disk Use

After determining that a file system has insufficient space available, check how its space is being used. The du program pinpoints disk space allocation by user. With this information you can decide who is using the most space and who should free up disk space.

To display a summary of how space is being used on a file system, use the du command with the -s option specified. The following example shows how to display a summary of the space use by all subdirectories (accounts) in the /usr/users file system:

```
# du -s /usr/users/*
```

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

You can also use the quot command, to list the number of blocks in the named file system owned by each user. For more information on checking disk space use, refer to quot(8) and du(1) in the ULTRIX Reference Pages.

2.1.3 Verifying Disk Quotas

If you are enforcing user disk quotas, you should verify your quota system periodically. You can use the following commands to compare the established limits with actual use: quot, quotacheck, and repquota.

The quot, command displays the actual block use for each user. The quotacheck command verifies that the actual block use is consistent with established limits. The repquota command displays both the actual disk use and the established limits.

If you find it necessary to change the established quotas, use the edquota command. This command allows you to set or change the limits for each user.

Note

> To enable automatic quotas, set the third field of the /etc/fstab file to rq for the listed file system.

For further information on disk quotas refer to the *ULTRIX-32 Supplementary Documents*, as well edquota(8), quotacheck(8), quotaon(8), quotaoff(8), and repquota(8) in the ULTRIX Reference Pages.

2.2 Obtaining Additional Disk Space

Once you have checked disk space and use, you may need to establish additional space on the disk or on a file system. You can use the following methods, depending on your system and the needs of its users:

- Delete unused or obsolete files
- Move a file system
- Move files to another file system
- Change partition tables

A complete discussion of changing partition tables is contained in Chapter 3.

2.2.1 Deleting Unused or Obsolete Files

You should request that your system users remove any unused or obsolete files. If there is still an insufficient amount of free space, request that users of that file system dump their infrequently used files.

For further information on backing up and restoring files, see Guide to System Backup and Restore.

2.2.2 Moving a File System

When moving a file system from one disk pack to another, follow these steps:

1. Dump the file system (level 0) with the dump command.
2. Create a clean file system with the newfs command.
3. Restore the file system with the restore command.

The Guide to System Backup and Restore contains information on backup and restore procedures for your system.

2.2.3 Moving Files to a Larger File System

There are two methods of moving files to a larger file system: moving all the files and merging individual files. However, before you move any files, make a level 0 dump of the entire file system.

When transfering all the files to a larger file system, restore the level 0 dump on the larger target file system.

When merging individual files from one file system into another, create the appropriate file and restore it to the target file system. To create and restore individual files, use the tar command. For example, to create the file named file1, type:

```
# tar c file1
```

In this example, the tar command creates file1 on the default output media, usually the default tape device (dev/rmt0h).

Then, after creating files on the default output media, use the tar command with the xp options to restore (extract) the file from the tape device to the target file system. For example,

```
# tar xp file1
```

This command extracts file1 from the default tape device to the current working directory using the pathname on the tar archive media. The p is a superuser option that causes tar to use the original protection code assigned to file1.

Disk Partitioning 3

This chapter provides the information you need to change the partition sizes of your system disks. In general, you allocate disk space during the initial installation or when adding disks to your configuration. However, there are cases when it is necessary to change the partition sizes on your system disks in order to accommodate changes at your site and to improve system performance.

The following sections describe:

- Basic planning and preparation
- The disk partitioning scheme used by ULTRIX systems
- The chpt(8) program
- Step by step procedures and examples

For a review of the concepts pertaining to ULTRIX system disks, see Chapter 1 of this manual.

3.1 Preparing to Change Disk Partitions

Before making any changes to the partitions on your system disks, you should:

1. Assess the number of file systems you need and the amount of space each requires, including potential growth space. Here, you might ask:
    - Are users logically and optimally grouped within /usr/users?
    - Are other files and directories logically and optimally grouped within file systems?
    - Does this setup satisfy our current needs?
    - Does this setup satisfy our projected needs?

2. Analyze the statistics available to you regarding current system performance. Here you might ask:
   - Does the current disk partitioning setup optimize the average and peak demands of system users?
   - Would a different setup improve system performance?
3. Review the current disk partition setup and file system allocation. For example, you should know:
   - The device type and partition defaults for all disks at your site
   - The current size of each partition on the disk(s)
   - The current location of each file system on the disk(s)
4. Back up all files systems that exist on a disk that you intend to reconfigure.

3.2 The ULTRIX Disk Partitioning Scheme

Each device driver on the system contains a set of default partition tables with one table for each type of disk on the system. This set of tables is the same for all ULTRIX systems:

- For all disks, the default partition table is copied into the active partition table for that particular disk when the disk is first opened, or whenever the disk volume is changed (for example, the drive is turned off and on).
- For some disk packs, the default partition tables are copied into their respective active partition tables every time the disk is opened.

The active set of partition tables contains one table for each disk configured into the system.

3.2.1 Partition Table Values

The ULTRIX partitioning scheme allows you to use the values for the default partition tables or to create or modify one or all of the partition table values for individual disks. You can continue to use the default values in the active partition tables in the device driver, or you can modify the partition table for a particular disk. There are two ways to override the default values:

1. Use the chpt(8) command
2. Mount a disk pack which already contains a different partition table in the superblock of its a partition.

For example, making a new partition table allows you to have two RM05 disks on the system with different partitions, one using the default values for all RM05 disks, the other using its own partition table, which is set up according to your space needs. You can port the modified disk to any other ULTRIX system, or you can modify the partition table for particular disk.

3.2.2 Reading the Partition Tables

The first time a disk is opened, the device driver copies the disk's default table in the driver that is to be used. The device driver then checks the superblock of the disk's a partition to see if the disk has an existing partition table. If there is no partition table, the driver uses the default values.

If the device driver finds a partition table in the superblock, the driver copies that table into the disk's active partition table in the device driver. This overwrites the default values that were originally copied into the active table.

The active set of partition tables in the device driver are the ones that the system uses for each disk, whether a partition table exists in the disk's superblock or not. Remember, the active set of partition tables in the device driver contains one table for each disk on the system for that driver. Each table is a copy of either the default table in the driver or the individual partition table in the superblock of the a partition of the disk.

There are two reasons why a disk might not have its own partition table:

1. The a or c partition does not have a file system for a partition table to reside in.
2. An individual partition table was never created for the disk.

3.2.3 Rules for Changing Partition Tables

Rules and guidelines to follow when changing disk partitions are:

- You must have superuser privileges to use chpt.
- Back up all the file systems before changing the tables if there is any data on the disk. The new partition will overwrite the old file systems, destroying the data.

- You cannot change the offset (beginning sector) or shrink any partition on a mounted file system or on a file system that has an open file descriptor.
- If you need only one partition on the entire disk, use the existing partition c.
- A disk's partition table always resides in the superblock of partition a. Therefore, you must have an a partition with a file system on it before you can change the tables.

  If you plan to change the partitions other than a, and there is no file system in partition a, create a file system on partition a before changing any others (use newfs to create a file system). Partition a must begin at the start of the disk (sector 0).
- If your disk is divided into cylinder numbers, translate them into sector numbers before using the change feature of the chpt command (—p). Here is the basic calculation:

```
(No. sect/track) * (No. tracks/cyl) = (No. sect/cyl)

(No. sect/cyl) * (cyl No.) = sect
```

For example, suppose you have an RM05 disk and you need the beginning sector number of the fifth cylinder. You can find the number of sectors per track and tracks per cylinder by using the diskpart(8) command:

```
# /etc/diskpart rm05
rm05: #sectors/tracks=51, #tracks/cylinders=14
      #cylinders=1248
partition     size        range
    a         15884        0 -  26
    b         33440       27 -  81
    c        500384        0 - 822
    d         15884      562 - 588
    e         55936      589 - 680
    f         86176      681 - 825
    g        158528      562 - 822
    h        291346       82 - 561
```

Now you can perform the calculation:

```
51 * 14 = 714

714 * 5 = 3570
```

Note

Be careful when changing partition tables; you could overwrite data on the file systems or make the system inefficient. If the partition tables become corrupted while you are changing the partition sizes, you can return to the default partition table with the —d option of the chpt command.

3.3 Reviewing the Current File System Assignments

Before you change the partition tables or reallocate disk space, review the current file system assignments on your disk. To do this, use the df command.

The df command displays the current file system assignments, showing the partitions and space being used by the system. In addition, it displays the free space available on all file systems listed in /etc/fstab, and lists the partitions for each disk.

For example, if you have two file systems mounted on an RA81 system disk, when you use the df command, you will see something like this:

```
# df
Filesystem      total     kbytes    kbytes    percent
  node          kbytes     used      free      used     Mounted on
/dev/ra0a         7429      5371      1315      80%       /
/dev/ra0h       388950    118000    270950      30%       /usr
```

As this example shows, the df command gives you a broad view of the disk partition assignments that have been made. Note that any disk without file systems mounted on it does not show in the df display.

Refer to the ULTRIX Reference Pages for more information on df(1).

3.4 Reviewing the Current Disk Partition Setup

Before changing the size of a disk partition, review the current assignment. To do this, log in as superuser and use this format:

chpt -q *device*

This command invokes the chpt program. Including the –q option indicates that you want to view the partition sizes. The *device* variable indicates the specific disk you are checking.

Specify the device with its directory name (/dev), followed by the raw device name and partition a or c. For example, to check current partition sizes on an RA81, type:

```
# chpt  q /dev/rra0a
```

The chpt program responds with output similar to this:

```
Current partition table:
partition   bottom     top      size    overlap
    a            0    15883    15884    c
    b        15884    49323    33440    c,h
    c            0   891071   891072    a,b,d,e,f,g,h
    d       340670   356553    15884    h,c
    e       356554   412489    55936    h,c
    f       412490   891071   478582    h,c
    g        49324   131403    82080    c
    h       131404   891071   759668    c,d,e,f
```

3.5 Changing Partition Sizes

To change disk partition sizes with the chpt command, follow these steps:

1. Display the current partition table.
2. Unmount any active file systems.
3. Back up all data on the disk.
4. Calculate the new partition parameters.
5. Change the partition parameters.
6. Make and mount file systems as necessary.
7. Restore any backed up files to the new partitions.
8. Check the consistency of the new file system.

When following these steps, use the chpt command, which lets you change the partition table of an individual disk without rebuilding the kernel and rebooting the system.

The chpt format is:

/etc/chpt −a −q −d −v −p*x* *offset size ... device*

For more information on changing partition sizes, see chpt(8) in the ULTRIX Reference Pages.

The following sections explain the procedure and give examples for:

- Changing partition sizes on an RA81 disk
- Changing partition sizes on an RM05 disk

3.5.1 Changing Partition Sizes on an RA81 Disk

The following example shows the correct way to change a partition table on a running system. It then shows what happens if you try to change partitions incorrectly so you can avoid these mistakes.

Assume that you have an RA81 disk on drive 7 of your system and that you want to customize it to your system's needs by expanding partition a to include all the space in partition b.

Step 1: Display the Current Partition Table

Use the chpt command with the q option to see the status of the partition table:

```
# chpt  q /dev/rra7a
/dev/rra7a
Current partition table:
partition      bottom      top       size      overlap
    a          0           15883     15884     c
    b          15884       49323     33440     c,h
    c          0           891071    891072    a,b,d,e,f,g,h
    d          340670      356553    15884     c,h
    e          356554      412489    55936     c,h
    f          412490      891071    478582    c,h
    g          49324       131403    82080     c
    h          131404      891071    759668    c,d,e,f
```

Step 2: Unmount Active File Systems

Use the mount command to see which file systems are mounted on which devices:

```
# /etc/mount
```

The mount program responds with information such as:

```
hp0g on /usr
hp0h on /usr/staff
ra7a on /mnt
```

From this response you learn that the /mnt file system is mounted on partition a of the RA81 disk. This is the disk where you want to change partition sizes. Unmount /mnt as follows:

```
# /etc/umount /dev/ra7a
```

Step 3: Back Up the Disk

You should back up the entire disk before changing partition sizes.

In this example no files need to be backed up. If, however, files did need to be backed up, you would use the dump command to back up the files to another media.

Step 4: Calculate the New Partition Parameters

This step shows how to expand partition a of an RA81 disk so that it encompasses partition b. By looking at the partition table generated by the chpt command, you can see that you need to extend the size of partition a to the ending sector of partition b. Here is the formula that you can use:

```
(new length of a) = (beginning of b) + (size of b)
```

Here is the calculation:

```
49324 = 15884 + 33440
```

Step 5: Change the Partition Parameters

Shown here are three examples of changing partitions in an RA81 pack. Method A shows how to use the chpt command correctly with the -b option. Methods B and C show how **not** to change partitions.

Method A – Correct – Here is a correct way to extend the length of an active partition. This example shows how to extend the a partition by the size of partition b:

```
# chpt -v -pa 0 49324 /dev/rra7a
/dev/rra7a
Current partition table:
partition   bottom     top       size      overlap
    a       0          49323     49324     b,c
    b       15884      49323     33440     c,h
    c       0          891071    891072    a,b,d,e,f,g,h
    d       340670     356553    15884     c,h
    e       356554     412489    55936     c,h
    f       412490     891071    478582    c,h
    g       49324      131403    82080     c
    h       131404     891071    759668    c,d,e,f
```

Method B – Incorrect – One of the rules for changing partitions is that the chpt command does not let you shrink an active (mounted) partition. Here is what happens if you try:

```
# chpt -pa 0 800 /dev/rra7a
chpt: cannot set partition table in driver:
Mount device busy
```

Method C – Incorrect – Another rule is that you can never change the starting sector (offset) of partition a:

```
# chpt -pa 6 800 /dev/rra7a
chpt: cannot set partition table in driver:
Mount device busy
```

Step 6: Make and Mount File Systems

Then, place the file system on the partition, using the newfs command.

```
# /etc/newfs /dev/rra7a ra81
Warning: partition table overriding /etc/disktab
Warning: 656 sector(s) in last cylinder unallocated
/dev/rra7a:  49324 sectors in 70 cylinders of 14 tracks,
51 sectors 25.3Mb in 5 cyl groups
(16 c/g, 5.85Mb/g, 2048 1/g)
superblock backups (for fsck -b#) at:
 32, 11520, 23008, 34496, 45984,
```

If you are ready to use the file system, remount it on the partition with the mount command:

```
# /etc/mount /dev/ra7a /mnt
```

Step 7: Restore Backed Up Files

In our example, there are no files to restore. If there were, you would use the restore command.

Step 8: Check File Consistency

Check the consistency of the restored files using the fsck command.

3.5.2 Changing Partition Sizes on an RM05 Disk (Example)

Although this example is for an RM05 disk, the principles of this example apply to any disk that begins each partition at the beginning of a sector.

Here is a scenario that requires changes to the partition table. Assume that you have an RM05 disk on your system and you want to customize it to your system's needs. Assume your system requires twice as much swap space as the default partition table allows. Therefore, you need to double the size of partition b. One way to extend the length of partition b is by using space in the beginning of partition h.

For this example, assume that the RM05 disk has just been configured into the system and that no data is yet on the disk. Therefore, there is no need to back up or restore any files.

Keep in mind that RM05 disks always begin each partition at the beginning of a cylinder. Therefore, the starting sector of one partition may not immediately follow the last sector of the previous partition. This creates unaddressable disk space.

On an RM05 disk, there are 32 sectors per track and 19 tracks. Thus there are 608 sectors per cylinder (32*19). There are 823 cylinders.

Here is the default table, found in hp(4) for an RM05 on drive 0:

Disk	**Start**	**Length**	**Cylinders**
hp0a	0	15884	0-26
hp0b	16416	33440	27-81
hp0c	0	500384	0-822
hp0d	341696	15884	562-588
hp0e	358112	55936	589-680
hp0f	414048	86240	681-822
hp0g	341696	158592	562-822
hp0h	49856	291346	82-561

Here is how you want the partitions to be divided. The lengths of the partitions being modified are highlighted:

Disk	**Start**	**End**	**Length**	**Cylinders**
hp0a	0	15884	15884	0-26
hp0b	16416	83295	**66880***	27-136
hp0c	0	500384	500384	0-822
hp0d	341696	357579	15884	562-588
hp0e	358112	414047	55936	589-680
hp0f	414048	500287	86240	681-822
hp0g	341696	500287	158592	562-822
hp0h	83296	341202	**257906***	137-561

Now that you have determined what changes are necessary, follow these steps:

Step 1: Display the Current Partition Table

You must know how the disk is currently partitioned. To view the current table, use the chpt command, as follows:

```
# /etc/chpt  q /dev/rhp0a
/dev/rhp0a
No partition table found in superblock...
using default table from device driver.
Current partition table:
partition    bottom     top      size     overlap
    a        0          15883    15884    c
    b        16416      49855    33440    c,h
    c        0          500383   500384   a,b,d,e,f,g,h
    d        341696     357579   15884    c,g
    e        358112     414047   55936    c,g
    f        414048     50028    86240    c,g
    g        341696     500287   158592   c,d,e,f
    h        49856      341201   291346   c
```

In this example, chpt indicates that there is no partition table in the superblock of partition a. Therefore, the chpt command displays the partition table found in the device driver for the RM05 disk on drive 0.

Before you can change the partition table, you must copy the RM05 table in the driver into the superblock of partition a of the disk. The –a option of the chpt command does this for you:

```
# chpt -a /dev/rhp0a
```

Step 2: Unmount Active File Systems

Use the mount command to check for mounted file systems.

In this example, a new RM05 disk is being added to the system. Therefore, you know there are no file systems mounted on any partitions. If a file system did exist, you would unmount it using the umount command.

Step 3: Back Up File Systems

You should back up any file systems existing on the partitions to be changed. (For data protection, it is recommended that you back up all data on the disk before changing partition sizes.)

Because the example calls for adding a new RM05 disk to the system, there is no data on the disk. Therefore, in this example, you do not need to back up any files.

Step 4: Calculate the New Partition Parameters

To calculate the new partition parameters, use the sector numbers (size) that the chpt q command provided. You can do your calculations on a piece of scratch paper, with a calculator, or on the system. The following example shows the formula for doubling the size of partition b and reducing the size of partition h to reflect the new size of partition b. The formula for this example is:

```
(new size of b) = (current size of b) x 2

(beginning of h) = (size of new b) + (beginning size of b)

(size of h) = (ending sector of h) - (beginning sector of h +1)
```

The calculation for this example is:

```
 66880 =  33440 x 2
 83296 =  66880 + 16416
257906 = 341201 - 83297
```

You now have all the necessary data for changing partitions b and h.

Step 5: Change the Partition Table Parameters

First, use the chpt command to double the size of partition b:

```
# chpt -v -pb 16416 66880 /dev/rhp0a
/dev/rhp0a
New partition table:
partition     bottom     top       size      overlap
    a         0          15883     15884     c
    b         16416      83295     66880     c,h
    c         0          500383    500384    a,b,d,e,f,g,h
    d         341696     357579    15884     c,g,h
    e         358112     414047    55936     c,g,h
    f         414048     500287    86240     c,g,h
    g         341696     500287    158592    c,d,e,f
    h         49856      341201    291346    c
```

Second, use the chpt command to decrease partition h by the size that partition b was expanded:

```
# chpt -v -ph 83296 257906 /dev/rhp0a
/dev/rhp0a
New partition table:
partition     bottom     top      size     overlap
    a         0          15883    15884    c
    b         16416      83295    66880    c,h
    c         0          500383   500384   a,b,d,e,f,g,h
    d         341696     357579   15884    c,g,h
    e         358112     414047   55936    c,g,h
    f         414048     500287   86240    c,g,h
    g         341696     500287   158592   c,d,e,f
    h         83296      341201   257906   c
```

You have now changed the partition table.

Step 6: Make and Mount File Systems

Before you can mount file systems on the partitions, you must make the necessary file systems using the newfs command. Make new file systems for all partitions whose bottom addresses you changed with the chpt command. In this example, because b is swap space, you only need to make a new file system for partition h:

```
# /etc/newfs /dev/rhp0h rm05
```

Now you can mount any necessary file systems on the various partitions.

Step 7: Restore Backed Up Files

If you backed up any files before changing the partition tables, restore those files to their proper file systems using the restore command.

In this example, no files were backed up so there are no files to restore.

Bad Block Replacement 4

This chapter describes the replacement of bad blocks on DIGITAL Storage Architecture (DSA) disks attached to MicroVAX and VAX processors. This chapter discusses:

- Automatic bad block replacement – how the ULTRIX-32 operating system does dynamic bad block replacement
- Disk error messages interpretation– how the system detects, records, and reports errors
- Forced error recovery– how to recover the data associated with a bad block
- The radisk utility – how to replace, clear, and scan blocks on a disk manually

4.1 Automatic Bad Block Replacement

There are three strategies to replace bad blocks for DSA disks: controller-initiated, host-initiated, and media replacement. The controller-initiated and host-initiated strategies involve replacing a bad block with a good block reserved for this purpose by the DSA disk subsystem. The third strategy involves replacing the medium. Device support for these three strategies is summarized in Table 4-1.

4.1.1 Controller-Initiated and Host-Initiated Strategies

The controller-initiated and the host-initiated bad block replacement strategies have the same result, although they are implemented differently. In the controller-initiated strategy, the hardware confirms and replaces a detected bad block. In the host-initiated strategy, the controller notifies the host software of a bad block. Then the host software confirms and replaces the bad block.

Table 4-1: Device Support for Bad Block Strategies

Strategy	Controller(s)	Device(s)
Controller-initiated	RQDX1, RQDX2, RQDX3	RD31, RD32, RD51, RD52, RD53, RD54
	Small VAX disk controller*	RD31, RD32, RD53, RD54
Host-initiated	UDA50A, KDA50, KDB50	RA60, RA80, RA81, RA82
	HSC50, HSC70	RA70, RA90
Media replacement	RQDX1, RQDX2, RQDX3	RX50, RX33
	RUX50	RX50
	small VAX disk controller	RX50, RX33

*The small VAX disk controller is used by the VAXstation 2000 and MicroVAX 2000 processors. The software driver for the small VAX disk controller emulates RQDX3-style controller-initiated bad block replacement.

4.1.2 The Media Replacement Strategy

The RX50 and RX33 devices have no means of replacing bad blocks. If a hard error with a bad block is reported, the diskette must be replaced.

1. First try the diskette in another drive to be sure the problem is with the diskette and not the drive.
2. If you can read the diskette, make a copy of it, since the original diskette may be marginal. File the original and use the copy.
3. If the diskette is bad, recover as much of the data as you can, and recreate the remainder of the diskette.

4.2 Disk Error Messages Interpretation

This section describes how the system detects and reports disk errors and how to interpret bad block error messages.

4.2.1 Logical Block Numbers

The DIGITAL Storage Architecture (DSA) does not use the actual physical address of sectors on the medium. Instead, a sector on the medium is addressed by a logical block number (LBN). An LBN is the basic addressable unit of the user-accessible data area of the disk. The first LBN on a disk is numbered 0. The highest numbered LBN on a disk is numbered one less than the number of LBNs in the user-accessible area of the disk.

The LBN reported in an error message is relative to the start of the physical disk unit and is that of the first bad block encountered.

4.2.2 Detecting and Reporting Disk Errors

The disk controller detects and reports data errors when bad blocks cause erroneous data to be transferred from the drive to the controller. Error Correction Code (ECC) schemes are used to detect and correct any erroneous data found. These schemes apply statistical algorithms to dynamically correct erroneous data and to successfully complete requested I/O operations. If the data from a bad block is not correctable, the controller performs a series of rereads and other recovery techniques and attempts to obtain the data and deliver it without error.

Using ECC schemes, the hardware can correct an error when it is within a certain limit. If the error is equal to or below the limit, the controller can correct it and complete the I/O operation successfully without a retry. If the error is above the limit, the controller cannot correct the data. After repeated retries, the I/O operation fails and the failure is reported as a hard error to the host operating system. The block is bad and needs to be replaced.

Errors that are correctable but that are above a given threshold are also reported to the host operating system. These marginal blocks are replaced to prevent future loss of data.

If the data read from the bad block resulted in an uncorrectable ECC error, the replacement block is written with the forced error indicator. A forced error indicator warns that the data may be corrupted.

4.2.3 Examining the Errorlog File

The messages in the kernel errorlog buffer are logged to the errorlog file on disk, where they are accessed by the uerf command. The default errorlog file is /usr/adm/syserr/syserr.*hostname*.

To produce an error report containing driver error messages from the errorlog file, run the uerf command with the -r option:

```
# /etc/uerf -r 250
```

This command accesses ASCII messages. Driver error messages are logged in ASCII rather than binary format.

The -r option does not produce error messages for the VAXstation 2000 or MicroVAX 2000 disk driver. To produce an error report with messages for the sdc driver, use this command:

```
# /etc/uerf -c oper | grep sd?
```

The ? should be the unit number.

To diagnose a problem, study the types of errors logged. For example, a sudden incidence of hard errors reported at random places on a device may indicate an electrical or mechanical problem.

For more information on uerf(8) see the ULTRIX Reference Pages and Guide to the Error Logger System.

Note

Defective hardware can cause good blocks to be reported as bad. The system may be replacing blocks that would not be considered bad if the hardware was functioning properly. Check your hardware first if an error report displays an unusual number of bad blocks.

4.2.4 Interpreting Error Messages

The uda and sdc device drivers report bad block information to the errorlog file. Several types of error messages can be accessed from this file.

The following sections describe the disk error messages that may appear in the errorlog file. The examples are in the terse output format.

Note

The text of an ASCII error message, when reported in the brief or full output format, will generate more than one entry in the errorlog file. These ASCII text messages may not print sequentially.

4.2.4.1 Bad Block Reported Message – When an error results from a bad block reported condition, this message appears on the error report:

```
sd0: Bad Block Reported at LBN 79
        LBN 79 replaced
```

This message identifies the device and the location of the LBN and indicates that the block is bad and has been replaced.

4.2.4.2 More Bad Blocks Not Reported Message – When the controller encounters more than one bad block during a read of multiple blocks, it returns the LBN of the first block only. The following message appears on the error report:

```
uda50a0: unit 1, Bad Block Reported at LBN 42345
         More Bad Blocks NOT Reported!
         unit 1, LBN 42345 replaced
```

This message indicates that the reported block is bad and has been replaced. It warns that there is at least one more bad block. A similar message reports the next bad block and replaces it.

Note

The VAXstation 2000 and MicroVAX 2000 processors do not produce this message.

4.2.4.3 Transient Error Messages – Transient errors are caused by malfunctioning hardware or by marginal media. If your hardware is properly functioning, transient errors recorded in the errorlog file repeatedly for one LBN may indicate an impending bad block.

To check for transient errors, use the **uerf** command to produce an error report. Here is an example of a transient error message:

```
kdb501: unit 5, Bad Block Reported at LBN 179368
        unit 5, Transient error on LBN 179368, Block not bad
```

This message indicates that the reported block is not bad and has not been replaced.

When transient errors occur repeatedly, the controller's recovery mechanism may eventually fail to read the data successfully. If the block on the medium degrades, and the error level becomes too high to be corrected by ECC schemes, the transient error may become a hard error.

4.2.4.4 Forced Error Condition – During a bad block replacement operation, data is read from the bad block and written to the replacement block. When valid data cannot be read from the bad block and written to the replacement block, the forced error condition is set for the replacement block to indicate corrupted data. This message occurs when the hardware reads a block written with the forced indicator set:

```
ra1a: hard error sn 123
ra1a: Force Error Modifier set LBN 12345
```

The first line of this example appears at the console and the entire message is logged to the errorlog file.

This message means that the data in the block may be corrupted and should be restored. Restoring the data causes the block to be rewritten; this clears the forced error condition because the data is restored to a correct state.

4.3 Forced Error Recovery

When a read of a bad block cannot be successfully completed, the controller sets the forced error indicator in the replacement block. This indicator warns that the data written in the replacement block may be corrupted. If a read is successfully completed, the forced error indicator is not set and the replaced block is considered clean.

4.3.1 Discovering Forced Errors

To find out if there are any forced error conditions on a disk, run the uerf command with the -r option:

```
# /etc/uerf -r 250
```

This command accesses ASCII messages. Driver error messages are logged in ASCII rather than binary format.

The -r option does not produce error messages for the VAXstation 2000 or MicroVAX 2000 processors. To produce an error report with messages for the sdc driver, use this command:

```
# /etc/uerf -c oper | grep sd?
```

The ? should be the unit number.

4.3.2 Restoring Data

When you discover a forced error indicator on a disk, you may not know what type of data is associated with the block generating the error. Before you restore a block reporting a forced error, you should determine what kind of information the block contains:

1. Shut down multiuser mode. Use the shutdown command to bring the system to single-user mode.

2. Use the icheck command with the -b option to discover how the block is used.

   You need the sector number of the bad block from the errorlog message and the specification of the partition within which the bad block occurred; do not use the LBN.

   ```
   # icheck -b 2300 /dev/rra0g
   ```

3. If the icheck command produces an inode number, indicating the block is part of a file, specify the ncheck command with the -i option to determine the corresponding file. Use the inode number and the partition specification:

   ```
   # ncheck -i 354 /dev/rra0g
   ```

   Output from this command shows the file name or names associated with the bad block.

4. If the block with the forced error is a data file, copy the data to a new file and delete the old corrupted file. If the data is still corrupt, restore the file from a backup. Then use the fsck command to clean the file system. If the errors are still not cleared, run fsck again. You may need to run fsck several times.

5. If the forced error is in the superblock, use the backup superblock (fsck -b). See the fsck command for more information.

6. If the forced error is in an inode or a cylinder group block, first try the fsck command. If fsck does not work, decide whether to restore the file system or clear the block. If you decide to clear the block, use the radisk utility.

4.4 The radisk Utility

The radisk utility allows you to replace, clear, and scan blocks on a disk manually. However, you must have superuser privileges, be in single-user mode, and have all the file systems except root unmounted when using this command.

The radisk command has three options:

- -c clear
- -r replace
- -s scan

Note

> The VAXstation 2000 and MicroVAX 2000 processors do not support the -r option of the radisk command.

The format for the radisk command is:

radisk -option *LBN length special*

LBN is the number of the logical block. The length is the number or blocks from the specified LBN that you want to scan or clear. Special refers to the raw device special file (/dev/rra3c, for example). For the -c and -r options, the special file specified indicates an unmounted c partition of a character device special file. The -s option will accept any valid partition on the disk. The system produces an error message with both an LBN and a sector number.

4.4.1 The Clear Option

The clear option clears the forced error indicator in the specified area of a disk. The LBN can then be read without generating an error, even though the block may contain corrupted data.

This command line clears the forced error indicator from LBN 12334 to the last block of the partition:

```
# radisk -c 12334 -1 /dev/rra3c
```

Use **-1** to refer to the end of a disk partition. In this example, **-1** represents the number of LBNs from LBN 12334 to the end of the disk.

Note

> Clearing the block will remove the forced error indicator. However, the data in the block must still be considered corrupted. To maintain data integrity, check the data, and if necessary, restore it from backup. See the Guide to System Backup and Restore for information on restoring data.

4.4.2 The Replace Option

The -r option forces the replacement of blocks on a disk.

This option replaces a single block on a DSA disk. It enables you to replace a block that records repeated transient errors before it goes bad, thus saving the data.

For example, this command line forces LBN 12345 to be replaced:

```
# radisk -r 12345 /dev/rra3c
```

This option applies to host-initiated bad block replacement only; it does not apply to controller-initiated replacement.

Note

Replacing blocks affects performance. As blocks are replaced, the speed of the system is affected. In addition, there are a limited number of replacement blocks available on a disk.

4.4.3 The Scan Option

The -s option scans a disk for bad blocks. Use this option after reformatting or when the disk is new. The scan option reads the specified range of blocks and reports any errors. If a bad block is found, the block is replaced and the scan option restarts and rescans the specified range until it can complete without reporting any bad blocks.

When a forced error condition is found, radisk reports the LBN and continues to scan. It does not restart and the forced error indicator is not cleared. Use the -c option to clear a forced error condition reported during a scan.

This command line scans the g partition starting at LBN 12345 for 2000 blocks:

```
# radisk -s 12345 2000 /dev/rra0g
```

For the -s option, an LBN of 0 means the first LBN of the special file's partition.

For more information on the scan option, see the radisk(8) command in the ULTRIX Reference Pages.

Device Mnemonics A

This appendix identifies and defines the mnemonics that are used to attach any hardware or software device to your system. The mnemonics are used by the /dev/MAKEDEV shell script to create the character or block special files that represent each of the devices. The mnemonics also appear in the system configuration file as described in the Guide to System Configuration File Maintenance.

Table A-1 lists the mnemonics in seven categories: generic, consoles, disks, tapes, terminals, modems, and printers. The generic category lists the mnemonics of a general nature and includes memory, null, trace, and tty devices. The consoles category lists the system console devices that the ULTRIX operating system uses. The disks, tapes, terminals, modems, and printers categories identify the appropriate mnemonics for those devices.

The description heading in Table A-1 identifies the corresponding device name. It does not define the mnemonic's use. For detailed information on the use of each mnemonic in relation to both the MAKEDEV script and the system configuration file, refer to the reference pages in Section 4 of the ULTRIX Reference Pages. If on-line reference pages are available, you can also use the man command. For instance, if you enter at the system prompt:

```
# man ra
```

the system displays the reference page for the Mass Storage Control Protocol (MSCP) disk controller driver. Where appropriate, the SYNTAX section of the reference page defines the device's syntax as it appears, or should appear, in the config file. Refer to /dev/MAKEDEV for additional software device mnemonics that MAKEDEV uses. Refer to MAKEDEV(8) in the ULTRIX Reference Pages for a description of the MAKEDEV utility.

You should note that Table A-1 uses the convention of an asterisk (*) beside a mnemonic and a question mark (?) beside a device name to mean a variable number. The range of the variable number is dependent on the particular device.

Table A-1: Devices Supported by MAKEDEV

Category	Mnemonic	Description
Generic	boot*	Boot and std devices by cpu number; e.g., boot750
	mvax*	All MicroVAX setups; e.g., mvax2000
	vaxstation*	A VAXstation 2000 setup; e.g., vaxstation2000
	std	Standard devices below with all console subsystems:
	drum	Kernel drum device
	errlog	Error log device
	kUmem	Kernel Unibus/Q-bus virtual memory
	kmem	Virtual main memory
	mem	Physical memory
	null	A null device
	trace	A trace device
	tty	A tty device
	local	Customer specific devices
Consoles	console	System console interface
	crl	Console RL02 disk interface for VAX 86?0
	cs*	Console RX50 floppy interface for VAX 8??0
	ctu*	Console TU58 cassette interface for VAX 11/750
	cty*	Console extra serial line units for VAX 8??0
	cfl	Console RX01 floppy interface for 11/78?
	ttycp	Console line used as auxiliary terminal port
Disks	hp*	MASSBUS disk interface for RM?? drives
	ra*	UNIBUS/Q-bus/BI/HSC MSCP disk controller interface
	ese*	UNIBUS/Q-bus/BI/HSC MSCP electronic ESE20 disk
	rb*	UNIBUS IDC RL02 disk controller interface for RB?? drives
	rd*	VAXstation 2000 and MicroVAX 2000 RD type drives
	rz	SCSI disks (RZ22/RZ23/RZ55/RRD40)
	rk*	UNIBUS RK?? disk controller interface
	rl*	UNIBUS/Q-bus RL?? disk controller interface
	rx*	VAXstation 2000 and MicroVAX 2000 RX type drives
Tapes	mu*	TU78 MASSBUS magtape interface
	tms*	UNIBUS/Q-bus/BI/HSC TMSCP tape controller interface
	rv*	UNIBUS/Q-bus/BI/HSC TMSCP optical disk
	ts*	UNIBUS/Q-bus TS11/TS05/TU80 magtape interface
	tu*	TE16/TU45/TU77 MASSBUS magtape interface
	st*	VAXstation 2000 and MicroVAX 2000 TZK50 cartridge tape

Category	Mnemonic	Description
	tz*	SCSI tapes (TZ30/TZK50)
Terminals	cxa*	Q-bus cxa16
	cxb*	Q-bus cxb16
	cxy*	Q-bus cxt08
	dfa*	Q-bus DFA01 comm multiplexer
	dhq*	Q-bus DHQ11 comm multiplexer
	dhu*	UNIBUS DHU11 comm multiplexer
	dhv*	Q-bus DHV11 comm multiplexer
	dmb*	BI DMB32 comm multiplexer including dmbsp serial printer/plotter
	dhb*	BI DHB32 comm multiplexer
	dmf*	UNIBUS DMF32 comm multiplexer including dmfsp serial printer/plotter
	dmz*	UNIBUS DMZ32 comm multiplexer
	dz	UNIBUS DZ11 and DZ32 comm multiplexer
	sh*	MicroVAX 2000, 8 serial line expansion option
	ss*	VAXstation 2000 and MicroVAX 2000 basic 4 serial line unit
	dzq*	Q-bus DZQ11 comm multiplexer
	dzv*	Q-bus DZV11 comm multiplexer
	lta*	Sets of 16 network local area terminals (LAT)
	pty*	Sets of 16 network pseudoterminals
	qd*	Q-bus VCB02 (QDSS) graphics controller/console
	qv*	Q-bus VCB01 (QVSS) graphics controller/console
	sm*	VAXstation 2000 monochrome bitmap graphics/console
	sg*	VAXstation 2000 color bitmap graphics console
Modems	dfa*	DFA01 integral modem communications device.
Printers	dmbsp*	BI DMB32 serial printer/plotter
	dmfsp*	UNIBUS DMF32 serial printer/plotter
	lp*	UNIBUS LP11 parallel line printer
	lpv*	Q-bus LP11 parallel line printer

Index

V

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, Westminster, Massachusetts 01473

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

ULTRIX - 32
Guide to System
Disk Maintenance
AA-ME93A-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

Digital Equipment Corporation
Documentation Manager
ULTRIX Documentation Group
ZKO3-3/X18
Spit Brook Road
Nashua, N.H.
03063

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line

ULTRIX - 32
Guide to System
Disk Maintenance
AA-ME93A-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

Digital Equipment Corporation
Documentation Manager
ULTRIX Documentation Group
ZKO3-3/X18
Spit Brook Road
Nashua, N.H.
03063

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

Cut Along Dotted Line