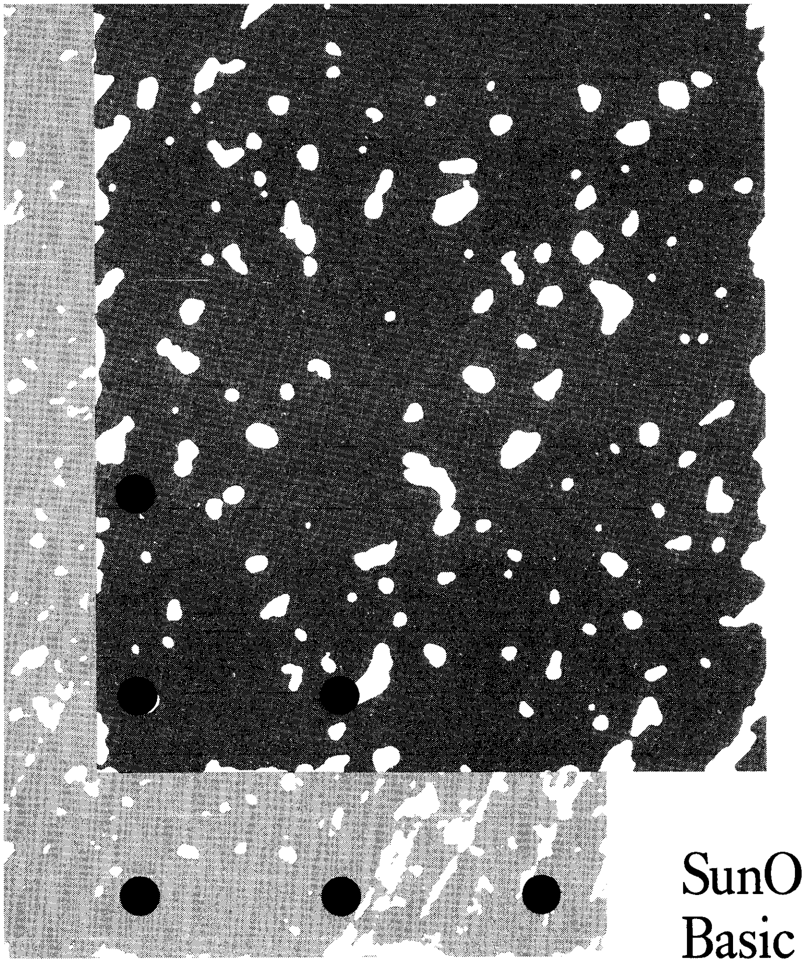


SunOS User's Guide: Basic Troubleshooting

The Sun Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees.

Contents

Figures

Preface

This manual describes problems that can arise during normal use of the SunOS operating system and a number of its utility programs. We suggest that you read *SunOS User's Guide: Getting Started* before using this manual.

NOTE *This manual does not cover procedures for system maintenance, malfunctions, or troubleshooting. For information about those topics, refer to System and Network Administration.*

Chapter 1 is an introduction. It includes a list of important problems, and steps to prevent or work around them.

Chapter 2 describes problems logging in and out.

Chapter 3 describes error messages that can happen when running commands and what they mean.

Chapter 4 describes error messages for selected commands.

Chapter 5 describes problems and error messages that can occur when editing text files.

Appendix A describes error messages that can occur when using remote systems over the network.

1

Introduction

When you begin using any new piece of equipment, problems can arise that seem to require prior experience to avoid or correct. Remember your first few encounters with a sewing machine or with car repairs?

A bit of guidance can be most helpful when you know what you are trying to accomplish, but have hit a snag. This manual is intended to provide you with pointers to get through the sorts of problems that many beginners find perplexing. The majority of problems most users face stem from simply not knowing what to do in a given situation.

This chapter describes initial steps for dealing with any problem, lists some of the most serious ones you may encounter, and describes what you can do about them. Subsequent chapters deal with problems that may come up when performing activities, such as logging in and out, running commands or editing files.

1.1. Discovering a Problem

There is never a convenient time to discover that you have a problem. As a beginning user, a problem typically comes to your attention when the system displays a message such as:

```
File not found.
```

```
Command not found.
```

or

```
Permission denied
```

Messages like this can be both perplexing and disturbing, but there is a reason behind each one. Often, the reason becomes obvious as soon as you think about it, but sometimes it takes a bit of detective work to figure out.

Since detective work can be difficult when you don't know where to look, it may help to befriend an experienced user who can diagnose problems that are especially puzzling.

Whenever a problem occurs, there are some steps that you can take to minimize the interruption and prevent any further complications from arising.

When you are confronted with an *error message*, stop what you are doing for a moment. Make note of what you are trying to accomplish and how far along you were when the problem came up. Then note the command that resulted in the error message (in most cases the command that you just typed in).

This will come in handy if you have to ask your system administrator[1] or a more experienced user for help. This will also make it easier for you to pick up from where you left off should you need to play detective.

The vast majority of error messages come as the result of mistyping a command or filename. So check the command line for spelling errors and typos.

Another common problem, even among experienced users, is being in the wrong directory when you type in a command. Check to see what directory you are in with `pwd`.

If you have occasion to use another person's account, you may have forgotten which one you are currently using. When you are logged in as someone else, files that belong to your account may not be accessible. Check to see which account you are using with `whoami`.

If the steps so far haven't revealed the source of the problem, then you are going to have to do some digging. So, along with your note about what you were doing when the problem occurred, copy down the error message exactly as it appears. This information will be important to you as you look for answers in this manual, and vital to anyone trying to help you.

By following these steps, you will have gathered enough information to answer the important questions of:

What?	What, exactly, was the error message?
When?	At which stage in your work did the problem arise, and with which command?
Where?	What directory were you in?
Who?	Whose account were you using?

This makes the questions of "Why did the problem arise?," and "How can I correct it?" much simpler to track down, especially when you need to get expert help.

[1] If your workstation is equipped with its own disk drive, then you may be able to perform tasks that we indicate should be referred to the system administrator. These tasks are described in *System and Network Administration*.

Step 1: Stop. Make a note of what you are trying to accomplish and how far along you are. Make note of the specific command that resulted in the error message.

Step 2: Check the command line for typing errors.

Step 3: Check to see what directory you are in with `pwd`.

Step 4: Check to see which account you are using with `whoami`.

Step 5: Take down the error message exactly as it appears. Check this manual for answers, and if necessary, get help from your system administrator or an experienced user.

1.2. Worst Cases

One problem that almost all beginners seem to face is anxiety about making a disastrous mistake. The fear that one may accidentally destroy valuable files or bring the system crashing down is as unnerving as it is exaggerated. (It takes more knowledge than most beginners have to do any serious harm). But it's only natural to be apprehensive when you aren't sure of the risks involved.

Perhaps the best way to overcome this anxiety is to learn what the realistic risks are and what you can do to minimize them.

It *is* possible to remove or erase the contents of a file unintentionally, but there are simple steps that you can take to help prevent this from happening. (It is also possible to make the system crash — say, by unplugging your keyboard and then plugging it back in — but it is normally quite easy to start it back up again.)

Deleted Data or Files

Perhaps the worst problem that you will encounter as a beginner is the accidental deletion of a file or a portion of its contents. Once removed, a file is gone for good, and the same is true for any contents that you erase.

Your system administrator should be making regular backup tapes of all files that change and can usually recover a recent copy of the file containing all but the current day's work. Still, losing a full day's work on an important file can be a big disappointment.

It is a good idea to check with your system administrator about the backup schedule for your system.

You can take the following steps to prevent your files from being accidentally deleted or overwritten. One thing you can do to help skirt the problem of lost or "trashed" files is to make a copy of the file from time to time — say, after every major change. This "shadow version" of your file will serve as a startup point if you should lose the original.

We assume that you are using the C shell.

If the following lines are not already contained in your `.cshrc` file, `cd` to your home directory and insert them:[2]

[2] Refer to *SunOS User's Guide: Customizing Your Environment* for more information about your `.cshrc` file.

Figure 1-1 *Safety Lines for Your* `.cshrc` *File*

```
set noclobber
alias cp        'cp -i'
alias rm        'rm -i'
alias mv        'mv -i'
```

The `set noclobber` command prevents existing files from being overwritten when you use the > symbol to redirect output. The `alias` commands indicate that `cp`, `rm`, and `mv` are to run with the `-i` option — to ask you for confirmation before removing or overwriting existing files.

To make sure that these commands take effect immediately, type

```
source .cshrc
```

on your terminal, or in each `cmdtool` or `shelltool` window currently running.

Be sure to open any `shelltool` or `cmdtool` windows that are iconic and run this command in them as well.

Step 1: Insert the lines

```
set noclobber
alias cp        'cp -i'
alias rm        'rm -i'
alias mv        'mv -i'
```

in your `.cshrc` file.

Step 2: Run the command `source .cshrc` in every `shelltool` or `cmdtool` window, or on your terminal.

System Shutdown

If your system administrator should need to take the system down while you are using it, you may see a broadcast message telling you that the system is about to shut down. Whenever the the system shuts down, it automatically saves files that you are editing with `vi`.[3] Even so, it is a good idea to type Esc :**wq** as soon as you see the broadcast message.

If you are running other programs and want to make sure that the most recent data is saved, type in the `sync` command:

```
venus%
Broadcast Message from venus!root (console) at 12:44 ...

System Going Down Immediately!
sync
venus%
```

[3] also `ex` and `ed`.

To recover the file you were editing after the system has come back up, cd to the directory you were in at the time of the crash, and type in the command:

```
vi -r filename
```

Then, before you do anything else, type in the command:

```
:w
```

This assures you that the recovered version is properly written out.

System Shutting Down:

Write and quit from vi when you see the broadcast message, or use the sync command to force data to be saved.

System Comes Back Up:

Step 1: cd to each directory containing files you were editing and:

Step 2: Use vi -r to recover preserved files.

Step 3: vi displays the date the file was preserved on its message line, after checking this date, type in the command :w to assure that the preserved version is saved.

System Crash

If the system *crashes*, or shuts down unexpectedly, files that you are editing are preserved as indicated above. Should this happen while you are using the workstation itself, the windows and mouse will cease to function and messages such as those shown below will appear at the lower left of the workstation screen. If you are logged in on a terminal, things will simply come to a halt.

If you are logged in at a terminal when the system crashes, contact your system administrator for assistance.

```
Abort at 004832
```

or

```
panic: condition
```

If the message says panic this doesn't mean that that *you* should. It means that the system's internal software encountered some sort of rare and unusual error condition. The condition often clears up when you start the system back up. Even so, it is a good idea to take down the entire message for your system administrator's information before proceeding to restart the system. Even when the system comes back up successfully, it is best to report any panic messages to your system administrator.

The message is followed by a right angle-bracket (>). This is the *monitor prompt.* To restart (or *reboot*) the workstation, type in the monitor command b when you see this prompt:

```
> b
```

The system will display a series of messages and eventually ask you to log in.[4]

Step 1: If the word `panic` appears in the error message, take down what the message said.

Step 2: When the monitor prompt appears, type in `b` to restart the system.

Step 3: If the system hasn't prompted for login in 15 minutes or so, contact your system sdministrator for help.

Step 4: Perform the steps in the previous section under *System Comes Back Up*

Unresponsive Terminal

When the terminal (or `cmdtool` or `shelltool` window) gets confused or hung, there is often no error message. Instead, the terminal may not display things properly. It may display only its prompts and messages but not what you type, or it may display only what you type and nothing else. This sometimes happens when you interrupt an interactive game with [Ctrl-C].

To clear the terminal or window, use the `reset` command as follows:

```
venus% [Line Feed]
venus% reset
venus% [Line Feed]
```

Even if the terminal doesn't display anything, it often recognizes this command, and clears up. Use [Line Feed] rather than [Return], because when it is confused, the terminal may not recognize [Return] properly.

(It's a good idea to be a bit patient with your machine, too. Many times a terminal appears to be hung, when in fact it's just slowed down due to heavy demands on it. This is especially likely if you are running several processes at once or if the network experiences particularly heavy usage.)

Error messages are often displayed when you type the first [Line Feed], and the system finally looks at everything you have typed since the last successful command.

If this doesn't clear things up and the problem is with a `cmdtool` or `shelltool` window, you may have to quit the window and open a new one. If the problem is with a terminal, you may have to get help from your system administrator. If you are logged in on a remote terminal over the phone lines, you may have to log out and dial in again. If you're on a single-user workstation, reboot it.

[4] Sometimes the system has to verify its data when it comes up again, and this can take up to 15 minutes. If you don't get a `login` prompt by then, contact your system administrator for assistance.

2

Problems With Logging In, Logging Out, and Passwords

In order to log in to the system, you must have an account set up for you by your system administrator. If this hasn't been done yet, or if a mistake is made when setting up an account, you may not be able to log in. Problems with logging out can come up when you have stopped jobs or when you attempt to log out from within a different shell (such as a `shelltool` window) than the one you logged in on.

2.1. Can't Log In

There is generally little that you can do about not being able to log in except to contact your system administrator and wait for her or him to correct the problem.

Account Not Set Up

If you have no account on a system, you cannot log in to it. If your system is connected to a network that uses the NIS, however, once your account is set up, it may be recognized by other machines on the network as well as your own. See *SunOS User's Guide: Getting Started* and *SunOS User's Guide: Customizing Your Environment* for more information, and check with your system administrator to be sure.

Can't Remember Password

If you forget your password, you cannot log in. You must ask your system administrator to change your password to something that is easy for you to remember. Once you can log in, you may want to change your password again so that it is known only to you. In general, long sequences of numbers, punctuation marks and other special characters are hard to remember, whereas words that evoke some meaning special only to you are easier to remember.

Terminal Control Characters in Password

Although you can use any character in your password, characters that can be interpreted as signals by the terminal such as [Ctrl-C], [Ctrl-Z], [Ctrl-U], [Ctrl-S], [Esc], [Tab], and in some cases [#] and [@] should be avoided. The terminal may interpret these as signals rather than as text characters, and this would preclude you from typing in the password correctly.

Capitals in the Username

If you type your username in all capitals when you log in, the system will behave as if the terminal were uppercase only. All letters will be displayed in uppercase. To avoid this, leave capital letters out of your username when you log in.

Misspelled Username or Mistyped Password

If you see the message

```
Login incorrect
```

try again. You may have miskeyed something. If on the third try you still aren't logged in, see your system administrator.

No Home Directory

If you see the message

```
No Directory! Logging in with home=/
```

This means that there is no directory set up for your use on the system. If this happens when you log in to another machine on the network, it may not matter. But if this happens when you log in to your own system, then you need to contact your system administrator to have him or her set up a home directory for you.

No Shell

If you see the message

```
No shell
```

then an error was made when you account was set up. Consult your system administrator.

Error Messages at Login

If you see an error message whenever you log in, there is probably an error in your `.login` file. If you see an error message whenever you start a new `shelltool` or `cmdtool` window, then there is probably an error in your `.cshrc` file. Since these files contain commands like those that you can type in, they are not too difficult to correct. Refer to *SunOS User's Guide: Customizing Your Environment* for more information about these files.

2.2. Can't Log Out

When you use the window system on your workstation, you are logged in as long as SunView is running — even if you quit from every window in the display. To log out, you must exit SunView first.[5]

Not Login Shell

If the message

```
Not login shell
```

appears when you attempt to log out, you can use the `exit` command to kill the current shell, `shelltool`, or `cmdtool`.

Stopped Jobs

If the system responds with the message

```
There are stopped jobs.
```

then use the `jobs` command to see which commands are waiting. If you don't care about any of them, a second command to log out will have the desired effect, and will terminate the stopped jobs as well. Refer to *SunOS User's Guide: Doing More* for more information about jobs, stopped jobs, and job control in the C shell.

[5] Logging out is one way to secure your workstation against unauthorized use. Another is to use the 'Lock Screen' item in the SunView menu. For more on security, see *SunOS User's Guide: Doing More.*

2.3. Can't Change Password

When you use `passwd` to change your password, you must enter the old password correctly, or it will reply with

```
Sorry.
```

and your password will remain the same.

If you use the `su` command to assume another user's identity,[6] you cannot change that user's password. Nor can another user change your password after gaining access with `su`.

Network Access: Your NIS Password

The NIS maintains a separate list of user accounts and passwords for everyone with access to the network. When you log in to another machine and are prompted for a password, it is asking you for your NIS password (which may be different than that on your current machine).

To change your NIS password, use the command `yppasswd`. This command works just like `passwd`. As you are prompted to do so, type in your old NIS password, your new password, and then confirm the new password. If you don't know your NIS password, the system administrator for your network can issue you a new one.

[6] See *SunOS User's Guide: Doing More* for more information.

3

Problems Running Commands

Because there are error messages specific to every command, attempting to describe each and every one would only obscure those that are more general in scope. In this chapter we describe error messages that may come up when using just about any command. The next chapter describes those issued by selected commands that are used most often. Appendix A lists error messages that can arise when using the network to gain access to other machines.

3.1. Problems With Command Line Syntax

```
... arg count
```

You specified either too many, or too few, arguments for the command. An argument follows a command and modifies it; it can be an option to that command, like `ls -F`, or a file that the command works on, like `cat namelist`, or so on. Many commands that require arguments will display the proper command syntax when just the command (no arguments) is typed in. See also `Too few arguments` and `Too many arguments`.

```
Arg list too long
```

Your command substitution[7] (or filename wildcard) expanded to form a command line of that is too long. This can happen when you use the `cat` command within backquote marks (`) or when you use `*` in a directory containing an extraordinary number of files.

```
Broken pipe
```

The command that was receiving data from a pipe terminated before the command sending data finished. This usually means that the second or subsequent command encountered an error and died. For example, this can happen when piping output through `more`, if you quit `more` before the other

[7] See *SunOS User's Guide: Getting Started* and *SunOS User's Guide: Doing More* for more information.

command is done. You may have to look up commands in the pipeline in the *SunOS Reference Manual* to figure out what went wrong.

```
exec failed:  wrong binary format
```

You're attempting to run software that will not run on the machine you're using. This is likely to occur when you have copied a program from another user running a different kind of machine, or if you run software that is specific to a particular release of SunOS. Often, you can replace the program with a copy meant for your machine.

If you have the source code for the program giving you this message, you may be able to recompile it on your own machine to make it work.

```
File exists
```

The command attempted to overwrite an existing file. Move the existing file to a new name with `mv` or remove it with `rm`, and run the command again.

```
File name too long
```

Your filename expansion string somehow expanded to a pathname with a component (directory-name or filename portion) exceeding 255 characters, or to a pathname that exceeded 1023 characters. Try specifying the pathname explicitly.

```
File not found
```

The file you attempted to look up or get access to is not in the directory. Refer to the discussion of `find` in *SunOS User's Guide: Doing More* for information on how to look up a file when you don't know the directory it is in.

```
File too large
```

The size of a file exceeded the maximum (about 10^9 bytes). If you see this message, consult your system administrator immediately.

```
Is a directory
```

You attempted to read or write on a directory as if it were a regular file. (Can happen to anyone).

```
... No match
```

The filename substitution pattern you gave did not match any files.

```
No such file or directory
```

This error occurs when a named file does not exist or when one of the directory-names in a pathname does not exist.

```
Not a directory
```

You supplied the name of a file to a command that expects a directory, such as `cd`.

```
Not owner
```

You tried to access or change permissions of a file that you don't own, you attempted to kill a process that you don't own, or you attempted to run a command that can only be run by the superuser (`root`). Refer to *SunOS User's Guide: Getting Started* and *SunOS User's Guide: Doing More* for more information.

```
Out of memory
```

There are too many processes taking up memory on your Workstation. If the system's performance is still good, then a good many processes may be idle and unnecessary. Use `ps` and `kill` to terminate idle processes that you own. Refer to *SunOS User's Guide: Getting Started* for more information about processes.

```
Permission denied
```

You attempted to either read, write, or execute a file for which you lack the proper permission. Refer to *SunOS User's Guide: Getting Started* for more information about files and permissions.

```
... Stale NFS file handle
```

The rug got pulled out from under the file you were using. The file is in a remote-mounted file system that got unmounted by *its* host system (perhaps by a crash of that host), or someone on another machine removed it, or moved a new version into it while your command was still running. Do an `ls` on the directory to see if the file is still there. If the file is there and its contents are what you expect, then rerun the command.

If the file is missing or its contents are unfamiliar, then you may need to get the version you need restored from backup. Files that are accessible from more than one machine may need to be protected using `sccs` to prevent collisions like this. Refer to *SunOS User's Guide: Doing More* for more information about `sccs`.

If no files are listed, have the system administrator remount the filesystem on the remote host, and run the command again.

```
Too many arguments
```

The command you used expected fewer arguments. For instance, you can `cd` to only one directory.

```
Too few arguments
```

The command you used expected more arguments. For instance, `mv` requires a minimum of two arguments: a filename and its new name.

```
Too many levels of symbolic links.
```

A pathname lookup involved more than 8 symbolic links. To reduce the number of symbolic links needed to resolve the pathname, use `ls -l` to look up the filename pointed to by the symbolic link. Then supply *that* name to `ls -l`, and so on until you reach the file or directory.

```
Too many links
```

You can have only 32767 hard links to the same file. Actually, if you see this message it could indicate that someone mistakenly made all those links. It is a good idea to report this to your system administrator so that the problem can be corrected.

```
Usage: ...
```

You typed in the command incorrectly, and it responded with a syntax diagram as a brief reminder. If the syntax summary is not clear to you, or if you need to know the full range of options open to you, check the *SunOS Reference Manual.*

```
... Unknown option
```

The option you specified is not valid for the command you typed. See `Too many arguments` above.

```
Unknown error
```

The system encountered something very strange. Contact your system administrator immediately. Do not attempt to correct this yourself. Your system administrator needs to see the exact sequence of events leading up to this error message.

3.2. Unforeseen Conditions

```
Disc quota exceeded.
```

You attempted to write on a file, create a new file, or create a new directory that would exceed your disc quota.

```
Read-only file system
```

You attempted to write on a file or in a directory within a filesystem that is mounted read-only.

```
... write failed, file system is full
```

```
File system full
```

There is no more room in the named filesystem. This message can come up when there are extremely large print jobs in the printer queue and you are running programs that use temporary files (such as `vi`) on a diskless Workstation. Both the printer queue and the `/tmp` directory may be in the same filesystem, which gets filled up with data for the printer. The solution is either to remove one or more print jobs or wait until those print jobs finish.

If the printer queue is not the problem, the `df` and `du` commands can be used to look at a system's file space.

```
segmentation fault
core dumped
```

```
Bus error
core dumped
```

The program failed to protect itself against an unexpected occurrence or internal error. The message `core dumped` means that the system has copied an image of the process into a file called `core`, in the current directory, for use in debugging. If you don't plan to attempt debugging, you should remove `core`, as it is often a large file.

If the program can't be debugged, the only recourse is to try rectifying the input, option, or condition that caused the problem — if such can be determined.

Sometimes segmentation faults are the result of poor memory management by programs.

4

Problems With Selected Commands

To interpret an error message produced by a specific command not contained in this list, refer to that command in the *SunOS Reference Manual.*

cat

```
cat: No such file or directory
```

Either you are in the wrong directory, the file does not exist, or you do not have permission to read it.

```
cat: input filename is output
```

You cannot `cat` the contents of *filename* to itself. Unfortunately, by the time you see this error message, the shell will have emptied out the file in preparation for writing to it. You can prevent this by having `noclobber` set, which prevents files from being written over. `noclobber` will give you the error message `File exists.` For more on `noclobber`, see Chapter 1.

cd

```
pathname: bad directory
```

You do not have permission to change to *pathname*.

chmod

```
chmod: can't access filename
```

You attempted to change permissions of a file or directory that does not exist. You may be in the wrong directory.

```
chmod: can't change filename
```

You attempted to change permissions on a file you don't own.

```
chmod: invalid mode
```

You specified the new permissions incorrectly. Perhaps you typed *filename* ahead of the permissions argument.

`cp`

```
cp: cannot open filename
```

The file specified by *filename* either doesn't exist, or it does not have read permission.

```
cp: cannot copy file to itself.
```

The two filename arguments to `cp` must represent different filenames.

```
cp: cannot create filename
```

This usually indicates that you are in a directory which you don't own, or you are trying to create a file in a directory for which you don't have write permission.

`date`

```
date: no permission
```

You must be the superuser (`root`) in order to set the date.

`df`

```
cannot open device
```

The filesystem or device you specified is not accessible.

```
bad free count
```

```
bad free block
```

Something is wrong in the file system. Notify your system administrator and do not try to fix this yourself.

`diff`

```
Files too big, try -h
```

Use the `-h` option when running `diff` on huge files.

`grep`

```
unknown flag
```

You supplied an option that `grep` does not recognize. Refer to `grep` in the *SunOS Reference Manual* for a list of valid options.

```
argument too long
```

The search pattern, or *regular expression*, on the command line is too long for `grep` to handle. You must shorten the expression, perhaps by performing the search in stages using several `grep` commands within a pipeline.

```
RE error
```

You specified regular expression (search pattern) incorrectly.

```
can't open filename
```

Either *filename* does not exist or you do not have read permission for it.

`kill`

```
No such process
```

Either the process does not exist or has terminated.

```
pid:  not owner
```

You are attempting to kill a process that someone else started. Many automatic processes are actually owned by daemons. Often you can `rlogin` in as `root` and kill processes you don't own.

`ln`

```
Cross-device link
```

You attempted to create a hard link to a file in another filesystem. You may create a symbolic link instead.

```
filename does not exist
```

The file named by the first argument does not exist.

```
File exists
```

A file by the name contained in the second argument already exists.

```
ln: name is a directory
```

Use the `-s` option to make a symbolic link to a directory.

`ls`

```
directory unreadable
```

You do not have read permission for *directory*.

```
too many files
```

```
out of memory
```

Some options of the `ls` command sort the filenames before displaying them. This message occurs when there is not enough memory available to perform the sort. Use `ps` and `kill` to eliminate extraneous (idle) processes. See *SunOS User's Guide: Getting Started* for more information.

make

```
Don't know how to make target
```

You asked make to process a target not contained in your makefile.

mkdir

```
mkdir: cannot access directory
```

directory does not exist or you don't have read access to its parent. This may come up should you try to create a subdirectory before its parent directory is created. This also appears when the parent directory does not have write access.

```
mkdir:cannot make directory dir
```

The directory specified by *dir* already exists.

man

```
No entry for item ...
```

There is no entry for the item you asked for in the online man pages or in the section you specified.

mount **and** umount

```
Block device required
```

You tried to mount something other than a filesystem.

```
Mount device busy
```

You attempted to mount filesystem that is already mounted or to dismount a device on which there is a file in use.

```
You must be root to use mount
```

You must be logged in as root (the *superuser*) to use these commands.[8] If you know the root password for your workstation, use su -f to "become root" quickly and run the command. Commands in Section 8 of the *SunOS Reference Manual* are typically run by root.

[8] mount with no arguments displays the filesystems you currently have mounted. You do not have to be

```
Too many levels of remote in path.
```

You attempted to mount a remote filesystem within another remote-mounted filesystem.

`mv`

```
can't mv directories across file systems
```

Use `cp -r` to copy the directory, its files, and any subdirectories (with their files) to the new location. Then use `rm -r` to remove directory and its files from the old location.

```
mv: directory rename only
```

This message appears if you type a command of the form:

```
mv filename olddir newdir
```

You can either move a list of filenames to a new directory, or move a single directory to a new name, but not both in the same command line.

```
mv: cannot unlink filename
```

Usually arises because you are not the owner of the directory containing *filename*.

```
mv: cannot move a directory into itself
```

The source and destination directory names must be different.

`pwd`

```
pwd: cannot open ..
```

The parent directory has somehow become read protected, and the `pwd` command cannot read it to determine the pathname of the current directory. This can happen when you have more than one `cmdtool` or `shelltool` window open and, from within another window, you change the permissions of any directory in the pathname of the current directory. The C shell command `dirs` may serve instead of `pwd`.

`root` to use `mount` this way.

`rm -r` and `rmdir`

Note: `rm -r` will *silently* remove a full directory just as long as

- You own it, and
- `noclobber` is not set.

```
Directory not empty.
```

There are still files in the directory. Use `ls -a` to see them all.

```
filename not removed
```

Either you lack write permission for the directory that contains the file or you did not answer with a `y` when asked for confirmation to remove the file.

`troff` and `nroff`

```
Core limit reached.
```

Refer to `Out of memory` in Chapter 3.

```
Segmentation fault.
core dumped.
```

A closing macro was missing. Use `checknr` to obtain a list of possible formatting errors. The message `core dumped` indicates that the system copied an image of the process where it left off for further diagnosis, if needed.

```
Typesetter busy.
```

Either `troff` was not properly installed on your system or network, your system uses a local version, or the phototypesetter is temporarily unavailable. Consult your system administrator to find out which is the case.

5

Problems Editing Text Files With `vi`

5.1. Operating Problems

The problems described in this section require you to perform specific tasks in order to avoid losing text or changes made since the file was last written.

Deleted Wrong Text or Need Previously Deleted Text

When you have made several deletions, such as when you use `dd` successively to delete lines, you may find that you want some of the deleted text back again. If you want the most recently removed line, you can use `u` (*undo*) to restore it. But suppose you want to undo the third-to-the-last deletion?

When you delete text (of a line or more), `vi` saves that text in a *buffer*. There are 9 numbered buffers that contain up to the ninth most recent deletion. To recover the *n*th most recent deletion, use the `p` command as follows:

```
"np
```

Hint: If you don't which buffer (or register) contains the text you want, start by typing

```
"1p
```

If that's not the buffer you want, type `u` (for *undo*) and a dot (.) to repeat. This will show you buffer 2. Repeat this procedure until you find your text or until you see the message, `Nothing in register` *n*.

When you delete a word or portion of a line, this text is *not* saved in a buffer. If you want to recover it, say, after adding some more text (in which case `u` won't work) use `p` (*put*).

Nested Editing Sessions

One common error in using `vi` is to run multiple copies of `vi` simultaneously on the same file. It's very easy to lose a lot of editing work this way.

You lose work because each time you run `vi` on a file, you edit a completely separate copy of the file. When you're done editing, you make that copy's changes permanent by writing it out to a hard disk. If you do simultaneous edits on the same file, then each time you end `vi` you'll write over any changes you might have made in other copies.

There are two common ways of accidentally running `vi` more than once on the same file. One way is if you are editing a file and you use [Ctrl-Z] to temporarily stop `vi` and put it in the background, and then type `vi` to go back to editing the file. If you do escape `vi` with [Ctrl-Z], do *not* go back into `vi` by typing `vi`; rather, use the `%` command, as explained in *SunOS User's Guide: Getting Started*

This puts you back into your original editing session.

The other way is use the `:!` shell escape, which allows you to run SunOS commands from within `vi`. One such command that you can run is, oddly enough, `vi`. It's okay to use `:! vi` *filename* **only** if you edit a *different* file. Do not attempt to use this shell escape to edit the file you're already editing.

Try this on a test file to see how it can happen:

CAUTION **Do not run this test on a file containing information of any value.**

Make a copy of a file and call it `test`. Run `vi` on the file.

```
venus% cp memo test
venus% vi test
(vi screen comes up)
```

Now, use `:!` to run `vi` again. When the new `vi` screen comes up, delete its contents with `dG`, then go to the first `vi` session with `:wq`.

```
:! vi test
(vi screen comes up)
dG
(text is deleted)
:wq
(previous text comes up)
```

At this point you can check to see that the written-out version of the file has less text by running the command `:! more test`

```
:! more test
(Hit Return to continue)
```

After typing Return, write and quit, then run `more test` from the C shell. Lo and behold, the contents are back! But what would have happened if you had added text rather than deleting it in the "inner" `vi` session?

If you accidentally do find yourself doing multiple edits of a single file, exit from that session with `:q!`.

Hint: If you want to have more than one copy of the file open in different windows, use `vi` to open it one, and `view` in the others. `view` is just like `vi`, except that it does not let you save any changes to the file.

Recovered Wrong Version

If you type in the `-r` option, and the file you are editing was once preserved during a crash (but not yet recovered), `vi` will recover this older and probably outdated version. If the date displayed for the file does not correspond to the most recent crash, *do not write the file and do not edit it.* Rather, check the file very carefully to see if it is up to date. If it is not, then use `:q!` to exit immediately from `vi`. Or, if in doubt, use

```
:w filename.REC
```

to save the recovered version. You can then exit `vi` and run `diff` to compare the two versions of the file.

If you have already typed `:w`, you may have to have the system administrator recover the more recent version from backup tape.

Can't Write Changes

When you edit a file for which you don't have write permission, you can still make changes to the temporary copy `vi` uses. If you don't own the file, you can write out a modified copy by supplying a filename to the `:w!` command:

```
:w! newfile
```

If you have write permission on the directory, you can exit `vi` and use `mv` to replace the existing version of the file with your modified version.

If you own the file, you can force the changes to be written out, even though `vi` complains. To force the changes to be written, use the command

```
:! chmod +w %
```

to give yourself write permission on the file. The percent sign is a symbol that stands for the name of the file currently being edited. Then use

```
:w!
```

to force the changes to be written. Of course, if you don't own the file, this won't work.

Ran Out of Disk Space

If, when you attempt to write a file, you get a message that the file system is full, then you will need to find another filesystem with more space. Use the command

```
:!df
```

to see which filesystems have space available and pick a directory in one of them. To write the contents onto a new file in this other directory, use the command

```
:w! pathname
```

Next, notify your system administrator about the full disk, so that she or he can move some of the files to another disk or remove them to tape.

If you get a message indicating that your disk quota has been reached, you may either have to forgo writing your changes or go searching for files to remove instead. (You can use Ctrl-Z to suspend `vi` while you hunt for files to remove. Use `%` to resume `vi` where you left off.)

If the system drops from `vi` into *open mode* and indicates a filesystem is full, this means that the disk containing `vi`'s temporary files filled up. Use the command

```
!ls /var/tmp
```

to see if there are files that you can remove. If you have queued up large print jobs, you may have to remove some of them or wait until they've printed, to regain disk space for `vi`.

Another directory to look in is `/var/adm`. It contains a number of files that contain records of your activities on the machine, such as the file `messages`, which keeps all the messages your system sends you. These files should be cleared out from time to time. (You must be the superuser, `root`, to do this; see *SunOS User's Guide: Doing More* for information on becoming superuser.) If you are sure that you don't need the file, use `vi` to remove its contents.

`vi` Comes Up as `ex`

You typed an interrupt signal before `vi` could draw the screen. Use the command `vi` to begin (or resume) full-screen editing.

5.2. `vi` Error Messages

Should a problem arise during an editing session, `vi` may display one of the messages listed below.

Problems Starting Up

```
Bad termcap entry
Termcap entry too long
terminal : Unknown terminal type
Block device required
```

```
Not a typewriter
```

`vi` could not figure out how to work with the terminal you are using. See you system administrator to correct this problem.

```
Block special file
Character special file
Directory
Executable
Non-ascii file
file non-ASCII
```

```
Bad file number
```

The file you attempted to edit was not a regular text file.

Problems With File Recovery

```
Can't fork to execute recovery
Can't make pipe for recovery
```

```
[Warning - file is incomplete]
```

`vi` was unable to recover all or part of a file preserved after a crash. See your system administrator. You may have to recover an earlier version of the file from backup tapes.

Problems Writing a File

```
File exists
File exists - use w! file to overwrite
```

```
[Existing file]
```

When you specify a file to write with `:w` *filename*, `vi` checks to see if a file by that name already exists. If so, you see one of these messages. Use `:w!` *filename* to overwrite the file.

```
[Read only]
File is read only
```

```
Permission denied
```

Either you are running `view` or you lack write permission for the file. In either case, see under Can't Write Changes, above.

A

Problems With Network Access

The following messages indicate a problem accessing a remote system over the network.

```
NFS server name not responding still trying
NFS server name okay
```

This is the network equivalent of "Don't touch that dial; we are experiencing technical difficulties." Your machine is having trouble connecting with the server *name*; this message often shows up when the remote server is very busy and your machine has to wait for service. Wait a bit to see if the connection is eventually made or regained. If the message continues, the remote machine may have "died."

```
Connection refused.
```

The remote machine is unavailable at present.

```
Connection timed out.
```

The remote machine failed to respond within the required time limit. Either it is too busy to accept new connections or it is restarting (*rebooting*) after a crash.

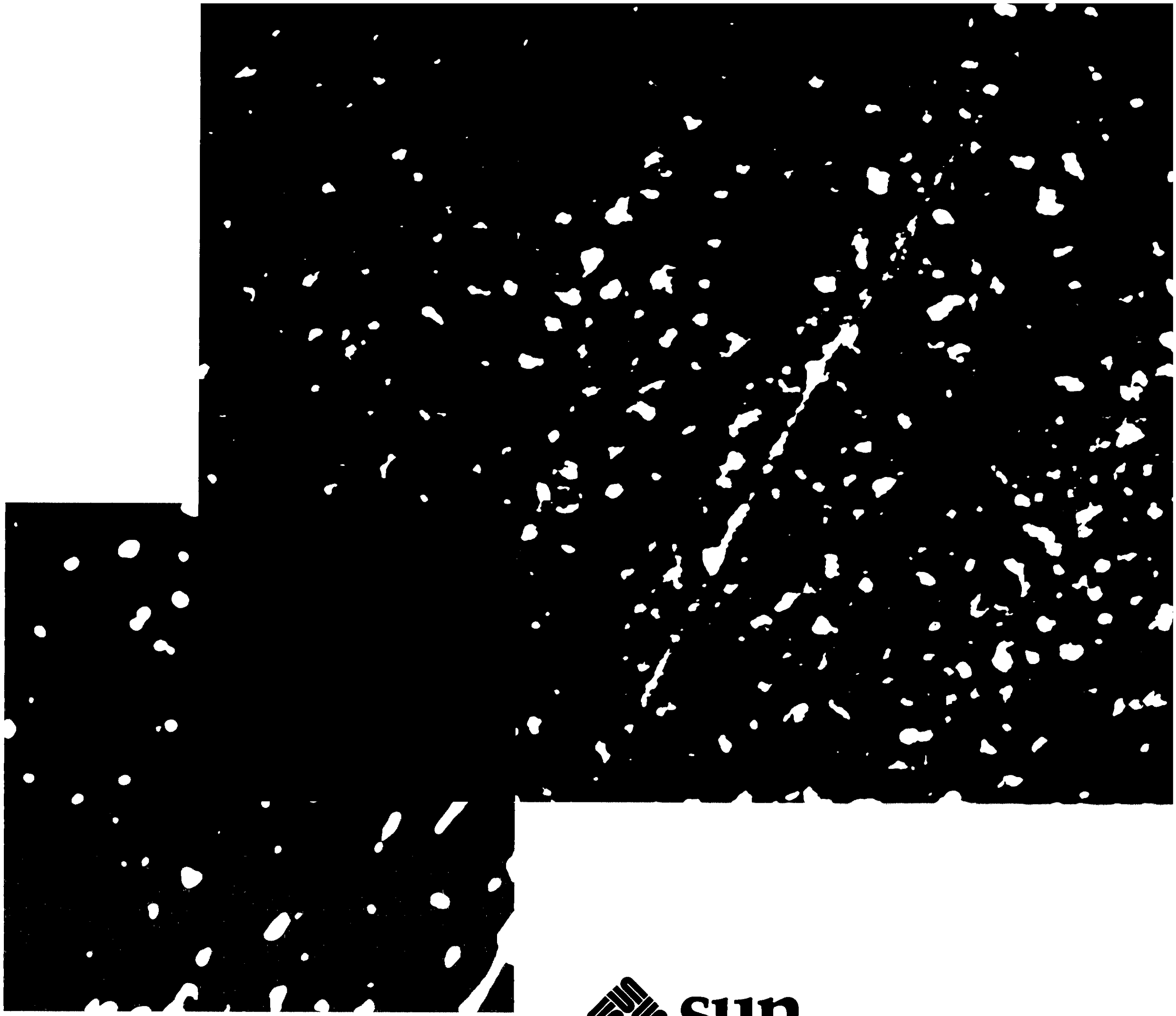
```
Network dropped connection on reset.
```

The host you were connected to crashed and rebooted, dropping the network connection to your system.

Index

Notes

Notes



sun
microsystems

Part Number: 800-3834-10
Revision A, of March 27, 1990