

More UNIX Commands

Utilities

This chapter introduces the following utilities, listed in alphabetical order:

at	cpio	diff
cmp	crontab	egrep
fgrep	od	tr
find	perl	tty
gawk	sed	ul
grep	sort	umount
gunzip	su	uniq
gzip	tar	whoami
ln	time	zcat
mount		

4.1 Introduction

In this chapter, I introduce about thirty useful utilities. Rather than describe them in a order, I've grouped them into fairly logical sets, as shown in Figure 4–1.

Section	Utilities
Filtering files	egrep, fgrep, grep, uniq
Sorting files	sort
Comparing files	cmp, diff
Archiving files	tar, cpio
Searching for files	find
Scheduling commands	at, crontab
Programmable text processing	gawk, perl
Hard and soft links	ln
Switching users	su
Transforming files	gzip, gunzip, sed, tr, ul, zcat
Looking at raw file contents	od
Mounting file systems	mount, umount
Identifying shells	whoami
Timing execution of a command	time

Figure 4–1 Advanced Linux/GNU utilities.

Utility: **grep** -hilnvw *pattern* { *fileName* } *
fgrep -hilnvwx *string* { *fileName* } *
egrep -hilnvw *pattern* { *fileName* } *

grep (Global or Get Regular Expression and Print) is a utility that allows you to search for a pattern in a list of files. If no files are specified, it searches standard input instead. *pattern* may be a regular expression. All lines that match the pattern are displayed to standard output. If more than one file is specified, each matching line is preceded by the name of the file unless the **-h** option is specified. The **-n** option precedes each matching line by its line number. The **-i** option causes the case of the patterns to be ignored. The **-l** option displays a list of the files that contain the specified pattern. The **-v** option causes **grep** to display all of the lines that don't match the pattern. The **-w** option restricts matching to occur on whole words only. **fgrep** (Fixed **grep**) is a fast version of **grep** that can only search for fixed strings. **egrep** (Extended **grep**) supports matching with regular expressions. **fgrep** supports an additional option; the **-x** option outputs only lines that are exactly equal to *string*.

For more information about regular expressions, consult the Appendix.

Figure 4-2 Description of the **grep** command.

Utility	Kind of pattern that may be searched for
fgrep	Fixed string only.
grep	Regular expression.
egrep	Extended regular expression.

Figure 4-3 The differences in the **grep** command family.

*Utility: **cmp** -ls *fileName1* *fileName2* [*offset1*] [*offset2*]*

cmp is a utility that tests whether two files are identical. If *fileName1* and *fileName2* are exactly equal, then **cmp** returns the exit code 0 and displays nothing; otherwise, it returns the exit code 1 and displays the offset and line number of the first mismatched byte. If one file is a prefix of the other, then the EOF message is displayed for the file that is shorter. The **-l** option displays the offset and values of all mismatched bytes. The **-s** option causes all output to be inhibited. The optional values *offset1* and *offset2* specify the starting offset in *fileName1* and *fileName2*, respectively, that the comparison should begin.

Figure 4–8 Description of the **cmp** command.

*Utility: **diff** -i -dflag *fileName1* *fileName2**

diff is a utility that compares two files and outputs a description of their differences. See the rest of this section for information on the format of this output. The **-i** flag makes **diff** ignore the case of the lines.

Figure 4–9 Description of the **diff** command.

Utility: **find** *pathList expression*

The **find** utility recursively descends through *pathList* and applies *expression* to every file. The syntax of *expression* is described below, together with some examples of **find**.

Figure 4–11 Description of the **find** command.

Expression	Value/action
-name <i>pattern</i>	True if the file's name matches <i>pattern</i> , which may include the shell metacharacters *, [,], and ?.
-perm <i>oct</i>	True if the octal description of the file's permission flags is exactly equal to <i>oct</i> .
-type <i>ch</i>	True if the type of the file is <i>ch</i> (b = block, c = char, etc.).
-user <i>userId</i>	True if the owner of the file is <i>userId</i> .
-group <i>groupId</i>	True if the group of the file is <i>groupId</i> .
-atime <i>count</i>	True if the file has been accessed within <i>count</i> days.

Expression	Value/action
<code>-mtime <i>count</i></code>	True if the contents of the file have been modified within <i>count</i> days.
<code>-ctime <i>count</i></code>	True if the contents of the file have been modified within <i>count</i> days or if any of its attributes have been altered.
<code>-exec <i>command</i></code>	True if the exit code from executing <i>command</i> is 0. <i>command</i> must be terminated by an escaped semicolon (\;). If you specify { } as a command-line argument, it is replaced by the name of the current file.
<code>-print</code>	Prints out the name of the current file and returns true.
<code>-ls</code>	Displays the current file's attributes (equivalent of <code>ls -dils</code>) and returns true.
<code>-cpio <i>device</i></code>	Writes the current file in cpio format to <i>device</i> and returns true.
<code>!<i>expression</i></code>	Returns the logical negation of <i>expression</i> .
<code><i>expr1</i> [-a] <i>expr2</i></code>	Short-circuiting and; if <i>expr1</i> is false, it returns false and <i>expr2</i> is not executed. If <i>expr1</i> is true, it returns the value of <i>expr2</i> .
<code><i>expr1</i> -o <i>expr2</i></code>	Short-circuiting or; if <i>expr1</i> is true, it returns true. If <i>expr1</i> is false, it returns the value of <i>expr2</i> .

Figure 4–12 find expressions. (Part 2 of 2)

Utility: tar -cftuvxz [tarFileName] fileList

tar allows you to create and access special tar-format archive files. The **-c** option creates a tar-format file. Use the **-f** option followed by a filename to specify the destination for the tar-format file. The **-v** option causes verbose output. The **-x** option allows you to extract named files, and the **-t** option generates a table of contents. The **-r** option unconditionally appends the listed files to the archive file. The **-u** option appends only files that are more recent than those already archived. The **-z** option filters the archive through **gzip** to compress or uncompress it. If the file list contains directory names, the contents of the directories are appended/extracted recursively.

Figure 4–14 Description of the **tar** command.

*Utility: gzip -cv { fileName }+
gunzip -cv { fileName }+
zcat { fileName }+*

gzip replaces a file by its compressed version, appending a “.gz” suffix. The **-c** option sends the result to standard output rather than overwriting the original file. The **-v** option displays the amount of compression that takes place.

The **gunzip** command uncompresses a file created by **gzip**.

zcat is equivalent to **gunzip -c**.

Figure 4–24 Description of the **gzip**, **gunzip**, and **zcat** commands.

Utility: **ln** -sF *original* [*newLink*]

ln -sF { *original* }+ *directory*

ln is a utility that allows you to create hard links or symbolic (soft) links to existing files.

To create a hard link between two regular files, specify the existing file label as the *original* filename and the new file label as *newLink*. Both labels will then refer to the same physical file, and this arrangement will be reflected in the hard-link count shown by the **ls** utility. The file can then be accessed via either label, and is removed from the file system only when all of its associated labels are deleted. If *newLink* is omitted, the last component of *original* is assumed. If the last argument is the name of a directory, then hard links are made from that directory to all of the specified original filenames. Hard links may not span file systems.

The **-s** option causes **ln** to create symbolic links. A symbolic link is a new file that contains a pointer (by name) to another file. A symbolic link may span file systems since there is no explicit connection to the destination file other than the name. Note that if the file pointed to by a symbolic link is removed, the symbolic-link file still exists but will result in an error if accessed.

The **-F** option allows a super-user to create a hard link to a directory.

For further information about how hard links are represented in the file system, see the discussion of Linux file systems in Chapter 13, “Linux Internals.”

Figure 4-21 Description of the **ln** command.

Utility: **whoami**


Displays the owner of a shell.

Figure 4–22 Description of the **whoami** command.

Utility: **su** [-] [*userName*] [*args*]

su creates a temporary shell with *userName*'s real and effective user/group IDs. If *userName* is not specified, "root" is assumed and the new shell's prompt is set to a # as a reminder. While you're in the subshell, you are effectively logged on as that user; when you terminate the shell with a *Control-D*, you are returned to your original shell. Of course, you must know the other user's password to use this utility. The SHELL and HOME environment variables are set from *userName*'s entry in the password file. If *userName* is not "root," the USER environment variable is also set. The new shell does not go through its login sequence unless the - option is supplied. All other arguments are passed as command-line arguments to the new shell.

Figure 4–23 Description of the **su** command.



Utility: **mount** *-options -t type [deviceName directory]*
umount *deviceName*

mount is a utility that allows you to “splice” a device’s file system into the root hierarchy. When used without any arguments, **mount** displays a list of the currently mounted devices. To specify special options, follow **-o** by a list of valid codes. These codes include **rw**, which mounts a file system for read/write, and **ro**, which mounts a file system for read-only. Use *type* to specify the file system type (e.g., ext2, ext3, etc.). The **umount** utility unmounts a previously mounted file system.

Figure 4–30 Description of the **mount** and **umount** commands.

Utility: **time** [-p] *command-line*

The **time** command can be used to report the execution time of any Linux command specified by *command-line*. Time is reported in both elapsed time and CPU time (CPU time is expressed as two values, user time and system time). When the **-p** option is used, **time** reports in the traditional UNIX format.

Figure 4-32 Description of the **time** command.

For example:

```
$ time -p sort allnames.txt >sortednames.txt
```

```
real    0m 4.18s
user    0m 1.85s
sys     0m 0.14s
$ _
```