Linux Quick Command Reference

This appendix contains a quick reference to Linux commands. For a more detailed description of any command, use the `man` command. For example, if you need information on the `ls` command, type

```
man ls
```

Alternatively, you can type the command with the `--help` flag to get a quick reference for that command. For example, for a quick reference for the `date` command, type

```
date --help
```

Note that this list is not exhaustive. A complete list of all Linux commands would comprise a whole new book the size of this volume. However, it includes those commands that I feel are most important.

`adduser` To add a user to the system. This is only available to the superuser ("root"). To remove a user, see `userdel`.

`cat` To print out the contents of a file to standard output. For example, `cat errors` prints out the contents of the file errors. If the file is a binary file and not an ASCII text file, this looks very ugly. Note that if you want to display file contents to a screen, you're better off using `more` or `less`.

`cd` To change directory. For example, `cd /usr/local/bin` to go to directory /usr/local/bin; `cd ..` to go from /usr/local/bin to /usr/local; `cd ~mike` to go to Mike's home directory, or `cd` to go to your own home directory.

`chfn` To change your personal information. Your personal information includes a real name, office location, and phone numbers. See `finger`. Note that this feature is altered or suppressed on many systems. You can also display more personal information by creating .plan and .project files.

`chgrp` To change the group to which a file belongs. For example, to turn file suzanne.c over to group software, type

```
chgrp software suzanne.c
```

See `chmod`, `groupadd`, `groupdel`, `chown`.

`chmod` To change the access permissions of a file or directory. There are three types of permission (read, write, execute) and three groups to give permission to (owner, group, everybody). One way to change permissions is to give a numerical code to indicate permissions. The octal codes are as follows:

Code	Read	Write	Execute
0	no	no	no
1	no	no	yes
2	no	yes	no
3	no	yes	yes
4	yes	no	no
5	yes	no	yes
6	yes	yes	no
7	yes	yes	yes

The octal code is three digits, the first for the owner, the second for the group, and the third for everybody. For example, to change the permissions on file resume.txt so that anyone can read it and you (the owner) can write it, use

```
chmod 644 resume.txt
```

See also `chgrp`, `chown`.

`chown` To change the owner of a file. For example, to make Monica the owner of file resume.html, type

```
chown monica resume.html
```

See also `adduser`, `userdel`, `chmod`, `chgrp`.

`cjpeg` To compress a raw image file (a bitmap) into a JPEG. See `xv`.

`clear` To clear the display screen.

`cp` To copy a file from one directory to another. For example, type

```
cp resume.html ~/docs
```

to copy resume.html to the docs subdirectory in your own directory; or type

```
cp *.html /dev/null
```

to copy all files in the local directory ending with the .html extension to /dev/null.

`cryptdir` To encrypt a directory; not available on all systems.

`date` To display the present date and time.

`decryptdir` To decrypt a directory; not available on all systems.

`dir` To list all contents of a directory. Not available on all systems. See `ls`.

`df` To display information on a file system, including space usage. If you specify a particular file residing on the target file system, you will get information on that particular system. For example, if you want information on the file system where resume.html resides, type

```
df resume.html
```

You can also get information on a file system by specifying the device file for that system (for example, /dev/hda1). Using `df` alone gives you information on all mounted file systems.

`dselect` The Debian package manager; a text-based GUI for easy installation, configuration, and upgrading of Debian packages. Not available with all distributions. See `rpm`, `tar`.

`du` To display the disk usage of each specified file. To learn how much disk space resume.html takes up, type

```
du resume.html
```

To learn how much disk space the directory /dev takes up, type

```
du /dev
```

`elm` To run a mail reader called ELM. Not available on all systems. See `mail`, `mutt`.

`emacs` To run a text file editor called Emacs. Not available on all systems. See the `emacs` command reference for details.

`exit` To exit a shell session. If you exit from your login shell, this command logs you out.

`find` To locate a file in the directory system. See `locate`, `updatedb`.

`finger` To obtain personal information on a user. To get personal information on Mike, type

```
finger mike
```

This feature is disabled on many systems. See `chfn`.

`free` To display information on memory use. Use the `-b` flag to display the information in bytes, `-k` for kilobytes, `-m` for megabytes. Use the `-o` option to remove buffer information. Use `-t` to display totals.

`fsck` To check for errors in a file system, and if desired, to repair them. This operation is only allowed to the system administrator.

`grep` To find a specified pattern in a specified file and print out all lines that contain this pattern. For example, if you want to print out all lines that contain the word `program` in resume.html, type

```
grep program resume.html
```

`groupadd` To add a user group to the system. Users in the same group can access some of the same files and directories. See `chmod`, `chgrp`, `groupdel`.

`groupdel` To remove a user group from the system. See `groupadd`.

`gunzip` To decompress a gzipped file. See `gzip`.

`gzip` To compress a file. `gzip -d` is the same as `gunzip`. Depending on the demand, one can set the effectiveness (and time cost) of the compression. `gzip -1` is the fastest compression, `gzip -9` is the best.

`head` To print the first 10 lines of every specified file. If a specified file is binary instead of ASCII text, this will look ugly.

`history` To display previously executed commands. Some systems keep command histories going back to the instant the account was created; others purge histories or restrict them to only so many commands.

`hostname` To echo the name of the system.

`id` To display the user ID of a user. To get your UID, type

```
id
```

To display Mike's UID, type

```
id mike
```

`kill` To halt a running process. In order to use this command you need to know the PID for that process; you can obtain this using the `ps` command. For example, if you want to exit Netscape, you could type

```
ps -fa
```

to find the Netscape process, and if Netscape has the PID 512, type

```
kill 512
```

to end the process. Usually used with the `-9` option.

`less` To display an ASCII text file one page at a time, allowing the user to scroll back and forth and to search for patterns. If the file is not ASCII text, this can look very ugly. See `cat`, `more` for similar programs.

`locate` To locate a file in the file system, using the file database. See `updatedb`, `find`.

`lock` To lock a Linux machine while you step away to the bathroom. In order to get back on the machine, you need to type your password. If you are using a shared machine, keeping a machine locked for more than fifteen or twenty minutes will get your fellow users very angry at you.

`logout` To end a session. If you have another shell session running, this command will tell you to use `exit` instead.

`lpr` To print a file. For example, if you want to print out the file resume.txt, type

```
lpr resume.txt
```

Usually you need to specify a particular printer with the `-P` option if there are several printers on the local network. For example, to print resume.txt to the printer Beth, type

```
lpr -PBeth resume.txt
```

`ls` To list the contents of a directory or the properties of a particular file. To list the contents of the local directory, type

```
ls
```

For the contents of /usr/local/bin, type

```
ls /usr/local/bin
```

for the properties of every file in directory docs, type

```
ls -l docs
```

`mail` To read or send email. This is a very basic mail program, and is command-line based. See `elm`, `mutt`.

`man` To display detailed information on a particular command or concept (called a *manual page*, or simply a *man page*). Although man pages are standard on most UNIX workstations, they are optional. Some installers forego man pages because they take up considerable disk space. For example, `man ls` prints out very detailed information on the `ls` command.

`mesg` To toggle whether or not you accept user messages (for example, via the talk utility). To accept user messages, type

```
mesg y
```

To block user messages, type

```
mesg n
```

`mkdir` To create a directory. For example, to create a directory docs in your present directory, type

```
mkdir docs
```

to create a new directory /usr/local/bin/ircii, type

```
/usr/local/bin/ircii
```

`more` To display the contents of a file, one page at a time. If the file is not ASCII text, this will look ugly. See `cat`, `less` for similar programs.

`mount` To mount a disk onto the file system; in particular, to mount a floppy disk, CD-ROM, or network file system. For example, to mount a floppy disk onto the file system, type

```
mount /dev/fd0 /mnt/floppy
```

To unmount a disk, use `umount`.

`mutt` To run a mail reader called MUTT. Not available on all systems. See also `elm`, `mail`.

`mv` To move a file to a new directory, deleting the old copy. For example, to move resume.html from the local directory to /home/httpd/mike/, type

```
mv resume.html /home/httpd/mike/.
```

to move a file from your local directory to the directory one level down (say, to /usr/local if you are in /usr/local/bin), type

```
mv resume.html ..
```

You can also use `mv` to rename a file. For example: to rename resume.html to resume.html.old, type

```
mv resume.html resume.html.old
```

`netconf` To configure the system's network connections. This is a difficult operation and is only available to the system administrator.

`nice` To run a program with deference to other operations. Because Linux is a multi-user, multitasking operating system, very computationally expensive programs can inhibit the ability of other users or processes to get work finished. The `nice` command instructs the kernel to yield to more important, less expensive operations. For example, `sort war-and-peace.txt` could take hours, making work almost impossible for others; `nice -19 sort war-and-peace.txt` means that the sort operation takes longer but that other operations execute much more quickly, and others can get work finished. `-1` is the least amount of "niceness," `-19` is the most.

`passwd` To change a password. To change your password, type

```
passwd
```

If you are a superuser you can change another person's password by typing `passwd` *`login-id`*; for example, to change Mike's password, type

```
passwd mike
```

`pwd` To find out your local directory. For example, if you are in /usr/local/bin, `pwd` will print out `/usr/local/bin`.

`ps` To list processes that you are running. `ps` alone will only list processes that you are running in your present shell. `ps -u` lists CPU and memory usage as well.

`rm` To remove designated file(s). For example, to remove resume.txt, type

```
rm resume.txt
```

to remove all HTML files in a particular directory, type

```
rm *.html
```

to remove both monica and suzanne, type

```
rm monica suzanne
```

`rmdir` To remove a specified directory. For example, to remove the directory /usr/local/bin/ircii, type

```
rmdir /usr/local/bin/ircii
```

`rpm` To use the Red Hat Package Manager to install or uninstall software or components. Red Hat packages are one common means for distributing Linux packages, the other one being compressed tape archives ("tar balls"). See `dselect`, `gzip`, `gunzip`, `tar`.

`rwho` To list all users on the local network. See `who`.

`sort` To sort and list the contents of an ASCII file. The sort takes place on a line-by-line basis. For example, if the file test contains:

```
do ra mi fa
so la ti
do
```

then `sort test` will produce

```
do
do ra me fa
so la ti
```

It is possible to use `sort` on multiple files, in which case `sort` will print out the sorted content of *both* files simultaneously. For example, suppose that the following are the contents of two files

FILE1	FILE2
do ra mi fa	she'll be coming around
so la ti	the mountain when she
do	comes

`sort FILE1 FILE2` produces

```
comes

do

do ra mi fa
she'll be coming around
so la ti
the mountain when she
```

If two files are already sorted, but you want to merge them in sorted order, use
`sort -m`.

`su` To switch users. For example, `su mike` switches the user to Mike. `su` alone switches to the superuser. `su` requires the proper password for the new user.

`startx` To start the X Window System. This is not available on all Linux machines.

`tail` To print the last ten lines of each specified file. If the file is a binary instead of ASCII text, this will look rather ugly.

`talk` To chat with someone through an xterm (command line) session. This feature is useful but practically antique; most such communications are better handled by email or by an IRC program. To chat with Mike, type
`talk mike`

See `mesg`.

`tar` To create a *tape archive* of several files, suitable for storage on a backup system or for download. Gzipped tape archives (or "tar balls") are one common means of distributing Linux packages. See also `gzip`, `gunzip`, `rpm`. To tar ball the contents of an entire directory and name the tar ball stinky, type
`tar -cfv stinky.tar *`

To place the contents of a tar ball in the local directory, type
`tar -xvf stinky.tar`

Note that `tar -zcvf stinky.tar *` creates the stinky.tar.gz tar ball and then gzips it; `tar -zxvf stinky.tar.gz` first gunzips the gzipped tar ball `stinky.tar.gz` and then unspools it in the local directory. Note that `tar` doesn't particularly care if you use the .tar or .tar.gz extensions; the program will do its job regardless of the name of the file. However, it is easier for a human to figure out that resume-collection.tar.gz is a gzipped tar ball, than to try to figure out that resume-collection (with no extensions) is a gzipped tar ball.

`tin` To run a news reader called TIN. Not available on all systems.

`touch` To create an empty file. To create empty files monica and suzanne, type
`touch monica suzanne`

If you use touch on an already existing file, it will update the file's last-modified time stamp.

`umount` To remove a disk from the file system. For example, to remove a floppy disk from the file system, type
`umount /dev/fd0`

or
`umount /mnt/floppy`

Note that this is `umount` and *not* `unmount`; the *n* is missing.

`uname` To display information about the system. In particular `uname -a` displays all available information about the system (that is, operating system, machine name, kernel release, node name, time and date, and hardware type).

`updatedb` To force an update of the file database. The locate command uses this database to find files quickly. See `locate`.

`uptime` To display the time, the total running time of the system, and the process load of the system.

`userconf` To configure user information in some detail. This is a complicated operation and is available only to the system administrator. See `adduser`, `userdel`. Note that this is not available with all distributions.

`userdel` To remove a user from the system. Although this removes the user, it does not remove the user's files. One can remove, for example, most of Mike's files with `rm -rf ~mike`, but this removes Mike's home directory; there is no guarantee that Mike does not have files elsewhere, or that Mike's directory is not storing files that belong to someone else. See `adduser`.

`vi` To edit a specified file using the vi text editor. See the `vi` command reference for details. A better version of vi is vim, which is a recent improvement of the `vi` environment.

`w` or `who` To list all users on the system. See `rwho`.

`whoami` To echo your present login ID.

`Xconfigurator` To configure your X Window System. `Xconfigurator` and XF86Config XF86Setup use graphical interfaces; XF86Config uses a XF86Setup command-line interface. This is available only to the superuser. X is tricky; consult Chapter 15, "Video Hardware."

`xv` To display most types of graphics in the X Window environment. For example, to view mike.jpg, type

```
xv mike.jpg
```

This will return an error unless you are running X Window. See `startx`.