Appendix E. Linux Command Quick Reference

The following list describes some of the most useful and popular Linux commands. Consult the man page for each command to learn about additional arguments and details of operation.

`adduser` *`userid`*
Creates a new userid, prompting for necessary information (requires `root` privileges).

`apropos` *`keyword`*
Searches the manual pages for occurrences of the specified keyword and prints short descriptions from the beginning of matching manual pages.

`at` *`time`* , `at -f` *`file time`*
Executes commands entered via `stdin` (or, by using the alternative form, the specified file) at the specified time. The time can be specified in a variety of ways; for example, in hour and minute format *`hh`* : *`mm`* or in hour, minute, month, day, and year format *`hh`* : *`mm mm`* / *`dd`* / *`yy`* .

`atq`
Prints descriptions of jobs pending via the **at** command.

`atrm` *`job`*
Cancels execution of a job scheduled via the **at** command. Use the **atq** command to discover the identities of scheduled jobs.

`bg` , `bg` *`jobs`*
Places the current job (or, by using the alternative form, the specified jobs) in the background, suspending its execution so that a new user prompt appears immediately. Use the **jobs** command to discover the identities of background jobs.

`cal` *`month year`*

Prints a calendar for the specified month of the specified year.

`cat` *files*
Prints the contents of the specified files.

`cd` , `cd` *directory*
Changes the current working directory to the user's home directory or the specified directory.

`chgrp` *group files* , `chgrp -R` *group files*
Changes the group of the specified files to the specified group. The alternative form of the command operates recursively, changing the group of subdirectories and files beneath a specified directory. The group must be named in the */etc/groups* file, maintained by the **newgroup** command.

`chmod` *mode files* , `chmod -R` *mode files*
Changes the access mode of the specified files to the specified mode. The alternative form of the command operates recursively, changing the mode of subdirectories and files beneath a specified directory.

`chown` *userid files* , `chown -R` *userid files*
Changes the owner of the specified files to the specified userid. The alternative form of the command operates recursively, changing the owner of subdirectories and files beneath a specified directory

`clear`
Clears the terminal screen.

`cmp` *file1 file2*
Compares two files, reporting all discrepancies. Similar to the **diff** command, though the output format differs.

`cp` *file1 file2* , `cp` *files directory* , `cp -R` *files directory*
Copies a file to another file or directory, or copies a subdirectory and all its files to another directory.

`date` , `date` *date*

Displays the current date and time or changes the system date and time to the specified value, of the form `MMddhhmmyy` or `MMddhhmmyyyy`.

`df`
Prints the amount of free disk space on each mounted filesystem.

`diff` *file1 file2*
Compares two files, reporting all discrepancies. Similar to the `cmp` command, though the output format differs.

`dmesg`
Prints the messages resulting from the most recent system boot.

`du` , `du` *directories*
Prints the amount of disk space used by the current directory (or the specified directories) and its (their) subdirectories.

`echo` *string* , `echo -n` *string*
Prints the specified text on the standard output stream. The *–n* option causes omission of the trailing newline character.

`fdformat` *device*
Formats the media inserted in the specified floppy disk drive. The command performs a low-level format only; it does not create a filesystem. To create a filesystem, issue the **mkfs** command after formatting the media.

`fdisk` *device*
Edits the partition table of the specified hard disk.

`fg` , `fg` *jobs*
Brings the current job (or the specified jobs) to the foreground.

`file` *files*
Determines and prints a description of the type of each specified file.

`find` *path* `-name` *pattern* `-print`
Searches the specified path for files with names matching the specified pattern (usually enclosed in single quotes) and prints their names. The **find** command has

many other arguments and functions; see the online documentation.

`finger` *users*

Prints descriptions of the specified users.

`free`

Displays the amount of used and free system memory.

`ftp` *hostname*

Opens an FTP connection to the specified host, allowing files to be transferred. The FTP program provides subcommands for accomplishing file transfers; see the online documentation.

`grep` *pattern files* , `grep -i` *pattern files* , `grep -n` *pattern files* , `grep -v` *pattern files*

Search the specified files for text matching the specified pattern (usually enclosed in single quotes) and print matching lines. The *–i* option specifies that matching is performed without regard to case. The *–n* option specifies that each line of output is preceded by the file name and line number. The *–v* option reverses the matching, causing non-matched lines to be printed.

`gzip` *files* , `gunzip` *files*

Compress (or expand) the specified files. Generally, a compressed file has the same name as the original file, followed by `.gz` .

`head` *files*

Prints the first several lines of each specified file.

`hostname` , `hostname` *name*

Displays (or sets) the name of the host.

`info`

Launches the GNU Texinfo help system.

`init` *run_level*

Changes the system run level to the specified value (requires `root` privileges).

`insmod` *module*

Dynamically loads the specified module (requires `root` privileges).

`jobs`
Displays all background jobs.

`ispell` *files*
Checks the spelling of the contents of the specified files.

`kill` *process_ids* , `kill -` *signal* *process_ids* , `kill -l`
Kills the specified processes, sends the specified processes the specified signal (given as a number or name), or prints a list of available signals.

`killall` *program* , `killall -` *signal* *program*
Kills all processes that are instances of the specified program or sends the specified signal to all processes that are instances of the specified program.

`ln` *old new* , `ln -s` *old new*
Creates a hard (or soft) link associating a new name with an existing file or directory.

`locate` *pattern*
Locates files with names containing the specified pattern. Uses the database maintained by the **updatedb** command.

`lpq`
Prints the entries of the print queue.

`lpr` *files*
Prints the specified files.

`lprm` *job*
Cancels printing of the specified print queue entries. Use **lpq** to determine the contents of the print queue.

`ls` , `ls` *files* , `ls -a` *files* , `ls -l` *files* , `ls -lR` *files*
Lists (non-hidden) files in the current directory or the specified files or directories. The *–a* option lists hidden files as well has non-hidden files. The *–l* option causes the list to include descriptive information, such as file size and modification date. The *–R* option recursively lists the subdirectories of the specified directories.

`mail`
Launches a simple mail client that permits sending and receiving email messages.

`man` *`title`* , `man` *`section title`*
Prints the specified man page.

`mkdir` *`directories`* , `mkdir -p` *`directories`*
Creates the specified directories. The *–p* option causes creation of any parent directories needed to create a specified directory.

`mkfs -t` *`type device`*
Creates a file system of the specified type (such as `ext2` or `msdos`) on the specified device (requires `root` privileges).

`mkswap` *`device`*
Creates a Linux swap space on the specified hard disk partition (requires `root` privileges).

`more` *`file`*
Lets the user peruse a file too large to be displayed as a single screen (page) of output. The **more** command provides many subcommands that let the user navigate the file. For example, the **Space** key moves forward one page, the **b** key moves back one page, and the **q** key exits the program.

`mount` , `mount` *`device directory`* , `mount -o` *`option`* `-t` *`type device directory`*
Prints the mounted devices or mounts the specified device at the specified mount point (generally a subdirectory of `/mnt`). The mount command consults `/etc/fstab` to determine standard options associated with a device. The command generally requires `root` privileges. The *–o* option allows specification of a variety of options; for example, `ro` for read-only access. The *–t* option allows specification of the filesystem type (for example, `ext2`, `msdos`, or `iso9660`, the filesystem type generally used for CD-ROMs).

`mv` *`paths target`*
Moves the specified files or directories to the specified target.

`newgroup` *`group`*
Creates the specified group.

`passwd` , `passwd` *`user`*

Changes the current user's password, or that of the specified user (requires `root` privileges). The command prompts for the new password.

`ping` *`host`*

Sends an echo request via TCP/IP to the specified host. A response confirms that the host is operational.

`pr` *`files`*

Formats the specified files for printing, by inserting page breaks and so on. The command provides many arguments and functions.

`ps` , `ps -Aux`

Displays the processes associated with the current userid or displays a description of each process.

`pwd`

Prints the absolute path corresponding to the current working directory.

`reboot`

Reboots the system (requires `root` privileges).

`reset`

Clears the terminal screen and resets the terminal status.

`rm` *`files`* , `rm -i` *`files`* , `rm -f` *`files`* , `rm -if` *`files`* , `rm -rf` *`files`*

Deletes the specified files or (when the *–r* option is specified) recursively deletes all subdirectories of the specified files and directories. The *–i* option causes the command to prompt for confirmation; the *–f* option suppresses confirmation. Because deleted files cannot generally be recovered, the *–f* option should be used only with extreme care, particularly when used by the `root` user.

`rmdir` *`directories`* , `rmdir -p` *`directories`*

Deletes the specified empty directories or (when the *–p* option is specified) the empty directories along the specified path.

`shutdown` *`minutes`* , `shutdown -r` *`minutes`*

Shuts down the system after the specified number of minutes elapses (requires `root` privileges). The *–r* option causes the system to be rebooted once it has shut down.

`sleep` *time*
Causes the command interpreter to pause for the specified number of seconds.

`sort` *files*
Sorts the specified files. The command has many useful arguments; see the online documentation.

`split` *file*
Splits a file into several smaller files. The command has many arguments; see the online documentation.

`su` , `su` *user* , `su -` , `su -` *user*
Changes the current userid to `root` or to the specified userid (the latter requires `root` privileges). The – option establishes a default environment for the new userid.

`swapon` *device*
Enables use of the specified device for swapping (requires `root` privileges).

`swapoff` *device*
Disables use of the specified device for swapping (requires `root` privileges).

`sync`
Completes all pending input/output operations (requires `root` privileges).

`tail` *file* , `tail -` *n* *file* , `tail -f` *file*
Prints the last several lines of the specified files. The *–n* option specifies the number of lines to be printed. The *–f* option causes the command to continuously print additional lines as they are written to the file.

`talk` *user*
Launches a program that allows a chat-like dialog with the specified user.

`tar cvf` *tar_file files* , `tar zcvf` *tar_file files*

Creates a tar file with the specified name, containing the specified files and their subdirectories. The `z` option specified that the tar file will be compressed.

`tar xvf` *`tar_file`* , `tar zxvf` *`tar_file`*
Extracts the contents of the specified tar file. The `z` option specified that the tar file has been compressed.

`telnet` *`host`*
Opens a login session on the specified host.

`top`
Prints a display of system processes that's continually updated until the user presses the `q` key.

`traceroute` *`host`*
Uses echo requests to determine and print a network path to the host.

`umount` *`device`*
Unmounts the specified filesystem (generally requires `root` privileges).

`uptime`
Prints the system uptime.

`w`
Prints the current system users.

`wall`
Prints a message to each user except those who've disabled message reception. Type **Ctrl-D** to end the message.

`wc` *`files`*
Prints the number of characters, words, and lines in the specified files.

Table 5-1 identifies Linux commands that perform functions similar to MS-DOS commands. The operation of the Linux command is not generally identical to that of the corresponding MS-DOS command. See the index to this book or the Linux online documentation for further information about Linux commands.

Table E-1. MS-DOS Commands and Related Linux Commands

MS-DOS	Linux
`ATTRIB`	`chmod`
`CD`	`cd`
`CHKDSK`	`df`, `du`
`DELTREE`	`rm -R`
`DIR`	`ls -l`
`DOSKEY`	(built-in; no need to launch separately)
`EDIT`	`ae`, `vi`, and so on
`EXTRACT`	`tar`
`FC`	`cmp`, `diff`
`FDISK`	`fdisk`
`FIND`	`grep`
`FORMAT`	`fdformat`
`MORE`	`more`
`MOVE`	`mv`
`SORT`	`sort`
`START`	`at`, `bg`

MS-DOS	Linux
`XCOPY, XCOPY32`	`cp`

Previous chapter

‹ Using Loadable Ethernet Drivers

Next chapter

F. Open Publication License ›