**Linux Commands**

**Terminal Navigation Commands**

* **&&** — This one is so basic that it’s not even technically a command. If you ever want to run multiple commands in sequential order, just stick this in between each one. For example, [command1] && [command2] will first run [command1] then immediately follow it with [command2]. You can chain as many commands as you want.
* **!** — Repeats a recently used command. Best to use it in conjunction with the history command. You can use !n to repeat the n-th command in history. You can also use !-n to repeat the command that happened n commands ago.
* [**cd**](http://ss64.com/bash/cd.html) — Changes the current terminal directory.
* **clear** — Clears the terminal screen.
* [**history**](http://ss64.com/bash/history.html) — Displays a list of all recently used commands. You can also cycle through recently used commands by pressing the Up and Down arrow keys in the terminal.
* [**ls**](http://ss64.com/bash/ls.html) — Displays a list of all files in the current terminal directory. You can modify it with parameters to specify some other directory or to change the format of the list.
* [**man**](http://ss64.com/bash/man.html) — Displays a help page (from the manual) based on your search query. Very useful for learning how to use a command you don’t recognize or when you forget the parameters for an infrequently used command. If you’re ever confused, turn to man.
* [**pwd**](http://ss64.com/bash/pwd.html) — Displays the current terminal directory as an absolute path.
* **whatis** — Displays brief descriptions of command line programs. Think of it like a simplified version of man when you aren’t sure what a command does but don’t need the full manual on how to use it.

## File Management Commands

* [**cat**](http://ss64.com/bash/cat.html) — When used on a single text file, it will display the contents of that file. When used on two or more text files, it will display all of their contents in sequential order. Use the redirection operator (“**>**“) to combine multiple text files into one text file.
* [**chmod**](http://ss64.com/bash/chmod.html)**/**[**chown**](http://ss64.com/bash/chown.html) — The chmod command changes the read, write, and execute permissions of a file while the chown command changes the user and/or user group that owns a file.
* [**cp**](http://ss64.com/bash/cp.html) — Makes a copy of a file. By default, the copy appears in the current terminal directory, but you can also specify the destination directory as well.
* [**find**](http://ss64.com/bash/find.html) — Searches a specific directory (or your entire system) to find files that match given set of criteria. There are dozens of options, including filename, filetype, filesize, permissions, owners, date created, date modified, etc.
* [**grep**](http://ss64.com/bash/grep.html) — Searches a specific file or set of files to see if a given string of text exists, and if it does, tells you where the text exists in those files. This command is extremely flexible (e.g. use wildcards to search all files of a given type) and particularly useful for programmers (to find specific lines of code).
* [**locate**](http://ss64.com/bash/locate.html) — Searches the entire system for files or directories that match the search query, then outputs the absolute paths for each match. By default, it only searches in directories for which you have permissions. This is the simplest and fastest way to find a file.
* [**mkdir**](http://ss64.com/bash/mkdir.html)**/**[**rmdir**](http://ss64.com/bash/rmdir.html) — Creates or deletes a directory, by default in the current terminal directory but a target directory can be specified as well. When deleting, the directory must be completely empty.
* [**mv**](http://ss64.com/bash/mv.html) — Moves a file from one directory to another, and you can specify a different name for the file in the target directory. You can use this command to rename a file by moving it to the same directory but with a different filename.
* **nano/emacs/vim** — The three main terminal text editors that exist on nearly all Linux systems, ordered by increasing complexity. Newbies should stick to nano as both emacs and vim are wildly complex (and wildly powerful).
* [**rename**](http://ss64.com/bash/rename.html) — Changes the name of a file or a set of files. Comes with a lot of interesting parameters, allowing you to automatically rename a bunch of files according to a pattern.
* [**rm**](http://ss64.com/bash/rm.html) — Removes files. With a certain parameter, it can be used to wipe the entire contents of a specified directory. It can also be used to delete several files that all match a certain filename pattern.
* [**touch**](http://ss64.com/bash/touch.html) — Changes the date accessed or date modified of the given file to right now.
* **wget** — Downloads the file or page at the given web URL.
* [**zip**](http://ss64.com/bash/zip.html)**/**[**gzip**](http://ss64.com/bash/gzip.html)**/**[**tar**](http://ss64.com/bash/tar.html) — Various formats for compressing and decompressing file archives.

**System Management Commands**

* **apt** — While apt isn’t a command in itself, there are three commands that you must know to make full use of APT: add-apt-repository ([for locating third-party packages](https://www.makeuseof.com/tag/top-7-ppas-repositories-add-ubuntu-based-systems/)), apt-get (for actually installing packages), and apt-cache (for searching your repositories).
* [**bg**](http://ss64.com/bash/bg.html)**/**[**fg**](http://ss64.com/bash/fg.html) — Sends a foreground job to run in the background or a background job to run in the foreground. For more on jobs, see the jobs command.
* [**df**](http://ss64.com/bash/df.html) — Displays how much space is used and free on your system.
* **free** — Displays how much RAM is used and free on your system.
* [**ip**](http://ss64.com/bash/ip.html) — Displays useful network details such as your IP address, network interfaces, bandwidth usage, and more. Can also be used to configure network-related settings.
* [**jobs**](http://ss64.com/bash/jobs.html) — Displays all current jobs and their statuses. A job is just a representation of a running process or group of processes.
* [**kill**](http://ss64.com/bash/kill.html)**/**[**killall**](http://ss64.com/bash/killall.html) — You can use kill to end a process according to its process ID (often used in conjunction with the ps command) whereas you can use killall to end all processes whose names match your query.
* [**mount**](http://ss64.com/bash/mount.html)**/umount** — Attaches and detaches a separate filesystem to your system’s main filesystem. Mostly used to make external devices, like hard drives or USB drives, interactable with your computer.
* [**ps**](http://ss64.com/bash/ps.html) — Displays a list of currently running processes. By default, it only lists processes started under your current user, but parameters exist to find and filter all kinds of processes.
* [**sudo**](http://ss64.com/bash/sudo.html)**/gksudo** — Prepending sudo allows you to run any command as superuser (e.g. sudo [command1]). If you want to run a graphical program with superuser privileges, use gksudo followed by the executable file for the program.
* [**top**](http://ss64.com/bash/top.html) — Displays a list of currently running processes, sorted by how much CPU each processes uses. Unlike ps, this command regularly updates in real-time. Basically a terminal equivalent to Task Manager.
* [**uname**](http://ss64.com/bash/uname.html) — Displays core system information depending on the parameters you use, such as kernel name and version, machine hardware, and operating system.
* **uptime** — Displays time elapsed since last boot.
* [**whereis**](http://ss64.com/bash/whereis.html) — Finds the location of the executable file for a given program.
* [**whoami**](http://ss64.com/bash/whoami.html) — Displays the current user name. Comes in handy when you’re switching between users with the su command and you lose track of who you are at the moment.