Linux Command-Line Reference

General

`tab`: **THE** most useful thing. Ever. Use tab to auto-complete filenames and command names when using the command line. Left pinky FTW!

⬆, ⬇: Navigate command-line history back and forth, using the up- and down- arrow keys respectively.

`ctrl`+`r`: Search through your command-line history.

`man`: Very useful command; shows manual pages for other commands, e.g.: `man ls`, gives you information about the `ls` command.

`*`: Used as a wildcard to expand filenames in the command-line. e.g.: `ls *.jpg`, shows all jpg image files in the current directory.

Navigation

`cd`: Change directory; this command is used to navigate through directories in the command-line. e.g.: `cd some-directory/`, goes into a directory called `some -directory`.

`ls`: List; this command is used to list all files and directories inside a directory. e.g.: `ls *.jpg`, lists all jpg image files in the current directory. `ls -la *.jpg`, lists all jpg image files in the current directory and displays their names and attributes as a list.

`pwd`: Print Working Directory; this command prints the absolute path of the directory where you are.

`find`: Finds files starting from a specified directory. e.g.: `find some-directory -name "*.jpg"`, finds all jpg image files inside `some-directory` and inside every directory inside `some-directory`.

`locate`: Finds files in computer without starting at current directory. e.g.: `locate *.jpg`, finds all jpg image files in the filesystem.

`.`: The single period refers to your working directory, where you actually are.

`..`: The double period is used to represent the previous (parent) directory. e.g.: `cd ..`, goes out from the current directory, and up to the directory that contains the current directory.

`/`: The single slash refers to the root directory, where your filesystem tree starts

File operations

`cp`: Copy; this command copies files. If there's already a file with the name given, that file is overwritten (deleted). e.g.: `cp image.jpg copy.jpg`, makes a copy of the file `image.jpg`, and names the copy `copy.jpg`.

`mv`: Move; this command moves/renames files. e.g.: `mv image.jpg new-name.jpg`, renames the file `image.jpg` to `new-name.jpg`.

`mkdir`: Make Directory; this command is used to create new directories within the current directory. e.g.: `mkdir new-directory`, makes a new directory named `new-directory`.

`rm`: Remove; this command deletes files. e.g.: `rm image.jpg`, removes the file named `image.jpg`. `rm -f *.jpg`, removes every jpg image file in current directory. `rm -rf new-directory`, removes the directory named `new-directory` recursively (with every file and directory inside of it).

`file`: This command is used to determine the type of a file. e.g.: `file image.jpg`, prints out `image.jpg: JPEG image data, JFIF standard 1.01`.

Text viewing/editing

`cat`: From conCATenating; this command displays a (text) file on the terminal. e.g.: `cat poems.txt`, prints the text in `poems.txt` to the terminal, line by line. `cat poems.txt names.txt`, prints the text on both files `poems.txt` and `names.txt` to the terminal, line by line, one after the other.

`more`: This commands also displays text files on the terminal, but waits for user input before moving to the next page. e.g.: `more poems.txt`.

`grep`: This command searches for words in text files and displays the entire line where they're found. e.g.: `grep love poems.txt`, searches for the word *love* in the file `poems.txt` and prints out the lines where *love* is found.

`nano`: Command line text editor. e.g.: `nano poems.txt`, opens a text file named `poems.txt` to be edited on the command-line. If the file doesn't exist, it creates an empty file to be edited.

Disk space

`du`: Disk Usage; displays how much disk space is being used by the current directory, and any directory inside the current directory. e.g.: `du -h`, prints out disk usage information; the `-h` option makes it "human-readable" (using bytes instead of disk "blocks").

`df`: Disk Free; displays the ammount of space available in the whole system, and any other hard drives attached to it (USB, SD, etc). e.g.: `df -h`, prints out available space information; the `-h` option, again, makes it "human-readable".

Networking

`lynx`: Old-school text-based web browser.

`wget`: This command is used to download internet files from the command-line, without using a browser. e.g.: `wget http://www.fact.co.uk/images/fact/fact-logo-white.gif`, downloads the FACT logo to your computer.

`ifconfig`: This command prints out information about your network connections (WiFi, cable, router information). It can be used to determine your computer's IP address (it's address on the internet or local network). e.g.: `ifconfig`, prints out all the info; look for the `inet addr:` parameters.

`ping`: This command sends a *hello* message to another computer, and computes how long it takes for the other computer to reply. It needs an IP address or url address as a parameter. e.g.: `ping 192.168.103.101`, pings the local computer at address `192.168.103.101`. `ping fact.co.uk` pings the server hosting FACT's website. To quit ping, hit `ctrl`+`c`.

`ssh`: This command lets you actually connect to remote computers, and use their command-line to run programmes. e.g.: `ssh pi@192.168.103.101`, would connect you to the computer with IP address `192.168.103.101`, and log you in as user `pi` (given that you have the correct password).

`nc`: More general than `ping`, this command lets you send and receive any kind of message over the network. e.g.: `nc -l 8888` opens up a connection at port 8888 on your own computer. Anyone can connect to your computer using that port, and start transferring data, with the command `nc 192.168.103.101 8888` (assuming your ip is `192.168.103.101`).

Camera/Images

`raspistill`: This programme lets you take pictures using the camera, from the command-line. e.g.: `raspistill -o image.jpg`, takes a picture and saves it as `image.jpg`.

`raspivid`: This programme lets you record video using the camera, from the command-line. e.g.: `raspivid -o video.mov -t 10000`, records 10 seconds of video (specified with `-t 10000`), and saves it in a file called `video.mov`.

`fbi`: Frame Buffer Image-viewer; displays an image when working in the command line. e.g.: `fbi image.jpg`, opens the image file named `image.jpg`. This command doesn't work in the X environment.

`gpicview`: This programme lets you open an image while working in the X environment. e.g.: `gpicview image.jpg`, opens the image file named `image.jpg`. Only works while in an X environment.

`omxplayer`: This programme is a general media player that can be used to play video files from the command line. e.g.: `omxplayer video.mov`, starts playing the video file called `video.mov`. Works both in pure command-line and the X environment.