**25+ Linux Commands Raspberry Pi Users Need to Know**

By [Les Pounder](https://www.tomshardware.com/author/les-pounder)published June 07, 2019

(Image credit: Shutterstock)

Most people run Linux on [Raspberry Pi](https://www.tomshardware.com/news/raspberry-pi) and its "official" operating system is a version of Debian called Raspbian that's been modified just for the Pi. From a typical user perspective, the default Raspbian desktop user interface is similar to Windows, with a main menu and system tray on the desktop along with icons for common applications.

However, like all forms of Linux, Raspbian has a powerful command line interface that gives you a lot more control over the computer than you can get using the GUI. And many important tasks are either easier or only possible via commands. That's why, below we've listed the most important commands for navigating the file system, installing software, editing files and monitoring performance.

**Opening a Terminal**

To open a terminal from the Windows GUI, either click the Terminal icon (it looks like a tiny monitor) or hit CTRL+ ALT + T. Once open, you will see a black screen appear with a flashing prompt. If you connect to your Pi via SSH or you already booted to the command prompt, you don't need to open terminal, because you're already there.

[**It looks like fire, but it's actually a light bulb!Luoccia**](https://www.luoccia.com/products/led-flame-effect-light-bulb-with-gravity-sensing-effect?utm_source=taboola&utm_medium=referral&tblci=GiBaiZ_hyLbR3jRfpVNLFSZd7LGNtq8CzIXhE_xVXuXleiDV8Fkoyo_96JXh0-4s#tblciGiBaiZ_hyLbR3jRfpVNLFSZd7LGNtq8CzIXhE_xVXuXleiDV8Fkoyo_96JXh0-4s)

pi@raspberrypi:~ $

This prompt is telling us that we are logged in as a user called *pi* and that our machine is called *raspberry pi*. The $ refers to our permissions, in that we have permission to edit any file or folder in our home directory, which in this case is */home/pi/*.

In our home directory we can store our work, projects, pictures etc. But we cannot harm the underlying filesystem as we do not have permission to do so. To make system wide changes we either need to be a user called “root” which has similar powers to administrator on Windows, or we need to use *sudo*(see below) to temporarily give us extra permissions. In this guide we shall cover this and much more.

So let's start our adventure by testing out a few commands and learning more about home our Raspberry Pi works.

**sudo - Super User Do**

By default, Raspbian, like all forms of Linux doesn't give you the admin privileges you need to perform some core tasks like installing software. However, by prefacing any command with the word "sudo," you can have admin rights for that execution. To use "sudo," you will have to be in the "subdoers" permission group, but the good news is that the default Raspberry Pi user is already in this group.

For example, to upgrade your operating system, you'd type:

sudo apt upgrade

**Configuring Your Pi**

Whether you want to change your password, enable VNC, overclock your Raspberry Pi or set your Wi-Fi network from the terminal, the easiest way is by using the configuration tool. To launch it, simply type:

sudo raspi-config

**Navigating the file system**

Moving around the filesystem is something we take for granted in a GUI environment. But with the terminal we can do everything, and with great speed and precision. We just need to know the correct commands. If you don't have permission to perform any of these actions on a particular file or directory, prefacing the command with sudo will probably let you.

**pwd - Print working directory**

This command will show the full path to the directory we are in, for example */home/pi/.*

pwd

Advertisement

**ls - List directory content**

This command is used to list the contents of a directory.

List files in your current directory.

ls

List the files in another directory, such as /var/log:

ls /var/log

See hidden files and directories, in a long list with extra details.

ls -lha

List all files of a certain type, for example .py Python files.

ls \*py

**cd - Change directory**

Using this command we can move around the filesystem. For example to move from our home directory to *Downloads*

cd Downloads

Move to a directory in another part of the filesystem, for example */var/log*.

cd /var/log

Go back to the previous directory that we were in.

cd -

Go back to our home directory.

cd ~

**Working with files**

Sometimes we need to take a peek inside a file, look for a specific command, error or bug and with these commands we can do just that all from the terminal.

**cat - Print files to the terminal**

Print the contents of a file to the terminal, for example a Python file.

cat test.py

Print contents of the file to the terminal with line numbers.

cat -n test.py

Advertisement

**less - Print files to the terminal**

This command will print the contents of a file in sections and we can scroll through the file using the arrow keys, Page Up / Down and Home / End.

less /var/log/syslog

**grep - Looking inside a file**

To search inside a file for a specific word / section of text. Typically used with log files when looking for issues. In this example we use *lscpu* to print the details of the CPU which is passed via a pipe | to *grep* which we instruct to look for *“MHz”*.

lscpu | grep “MHz”

**Edit a file**

For when you quickly need to edit a config file, Python code or just write a to do list.

**nano**

*Nano*is the easiest command-line editor for beginners.

Create a new file, for example *newfile.txt.*

nano newfile.txt

Edit an existing file, for example *test.py.*

nano test.py

Inside nano we navigate using the arrow keys and it works just like a regular text editor.

Save your work.

CTRL + OConfirm filename, Press Enter

Exit nano.

CTRL + X

System Resources & Management

Managing our operating system and checking system resources is standard practice in the terminal. Here we show a few commands to get you started.

**htop - Display system processes**

Shows the current CPU load, RAM usage and running system processes. Installed by default on Raspbian. Useful for closing non responsive applications.

htop

Advertisement

**free - Show amount of free and used RAM**

This command will tell us how much RAM is in use, and what is free for applications. Using the -m option we can set the values in MB.

free -m

**dmesg - Monitor kernel events**

The kernel is the core of the operating system and with dmesg we can see what events are happening behind the scenes. Useful for debugging issues with devices.

dmesg

**File Management**

Moving, deleting, copying and creating new files and directories are some of the most basic actions that we need to do. From the terminal we can do that and much more.

**mv - Move / rename a file**

This command offers two functions. We can move a file from one location to another. For example here we move *test.py*to the *Documents*directory.

mv test.py Documents/

The command can also be used to rename a file or directory. Here we rename *test.py* to *test2.py.*

mv test.py test2.py

**rm - Delete a file**

With this command we can delete files and directories. In this example we delete the file *test.py.*

rm test.py

**cp - Copy a file**

To copy a file, for example test.py to our *Documents*directory.

cp test.py Documents/

To copy a directory, for example /home/pi/*test2*to*/home/pi/Documents/*we need to use the *-r*option to copy everything across.

cp -r test2/ Documents/

**mkdir - Create a directory**

Create a new directory to store work. For example let's create a directory called *Work* in our home directory.

mkdir Work

**Software Installation**

Just like any computer we need to make sure that our software is up to date and on our Raspberry Pi the tool to do just that is called apt.

**apt - Install and manage software**

Apt, the Advanced Packaging Tool. The app store of Linux. To use apt we will need to use *sudo* as it will make changes to the operating system.

First we update the list of installable software.

sudo apt update

Then we can install a specific application, for example to install vlc.

sudo apt install vlc

Or we can upgrade all of the software on our Raspberry Pi. Note that for this command we pass the *-y* option to automatically agree to install every package. But this is optional.

sudo apt upgrade -y

Network Connectivity & Internet

Checking that your Raspberry Pi is connected to the Internet is a basic yet crucial task. It enables us to debug our IoT projects and watch YouTube videos.

**ping - Check that we are connected**

The *ping*command is used to test that our Raspberry Pi is connected to the Internet / home network.

We can send a ping to a website.

ping google.com

Advertisement

Or to an IP address such as Google’s DNS server.

ping 8.8.8.8

Or for internal connectivity checks we can send a ping to devices on our home network. This example assumes that our IP range is 192.168.1.1 but your range may be different.

ping 192.168.1.1

**hostname - Get the IP address of your Raspberry Pi**

The easiest way to find the IP address of our Raspberry Pi is using *hostname*with the -I (uppercase i) which will show all IP addresses (WiFi and Ethernet)

hostname -I

**Curl - Transfer data over a network**

With this command we can transfer a file to and from our Raspberry Pi. For example if we wanted to download an image from a website we would use *curl*along with the *-o option*to create a file named *image.jpg.*

curl http://link-to-theimage.com/image.jpg -o image.jpg

The *curl*command is particularly useful for downloading installation scripts to automatically install add on boards. But it should be used with caution and any code reviewed before it is used.

**Time Savers**

The Linux terminal has many secrets and tricks, all designed to save you time and become a keyboard ninja.

**history**

The *history* command will display the history of the commands entered in the terminal. When used it will output all of the commands at once as a long list.

history

At the start of each line in the list is a number and we can use this number to run that command once again. But we must precede the number with an exclamation mark.

!117

**history with grep**

By using a pipe “|” with the *history* command we can send the output of *history*to *grep* where we can then search for specific commands. Here we look for all the occurences of “apt” in *history.*

history | grep “apt”

**CTRL + R search**

Using this we can interactively search through our command history for a specific command. To start we press CTRL + R together and then start typing part of the command. For example we have just used the *history | grep “apt”*command so we can now press CTRL + R and start typing *history* and the search will find that command.

**TAB completion**

Think of this as “auto complete” for the terminal. The TAB key is located just above the Caps Lock key and we can use TAB completion to help complete long commands, directory listings. If we type in the first few letters of a command, for example *his*and press TAB it will complete the command to show *history*.

But if we wanted to complete a long directory path, for example */usr/lib/python3/dist-packages*then we could start typing */usr/lib/*and then press TAB to show us all the directories available along that path. We could then start typing *python3 and by pressing TAB a few more times, the command will narrow down the options that we can use.*

**Arrow keys**

Another way to search through your *history* is to use the up and down arrow keys. With these keys we can go backwards and forwards through our command history and when the correct command is displayed, press Enter to run.

**alias**

Using this command we can create shortcuts / short commands from much longer ones. For example here we create an alias called *updater* and use that to call two commands. The first will update our list of installable software, and if that works successfully, denoted by using “&&” to chain the commands together, then it will run the upgrade on our Raspberry Pi.

alias updater=”sudo apt update && sudo apt upgrade -y”

Now we can run our update command just by typing *updater* in the terminal. Note that once the Raspberry Pi is switched off, this alias is deleted.