

# How to Linux

January 20, 2014

## Introduction

Linux is an operating system just like Windows, Android, and Mac OS. In fact Android and Mac OS use linux as part of their system (they just have more expensive and fancy graphics). Linux is very popular among scientists as it is free, fast, and very customizable.

This document is to help prepare you for using linux in Astronomy 150 and 250 labs. Hopefully you can continue to use it in your career as a reference sheet if you cannot remember how to do something.

## Navigation

Once you log into one of many different *flavours* of linux (you are probably using iceman or fedora on a lab computer; I personally prefer ubuntu), you have access to an interface similar to Windows or Mac, with some sort of starting button in one of the four corners of the screen. What you can do with a mouse in this interactive screen can also be done by typing into a command line.

First things first, go to the menu in the corner and open a *terminal* or *command line*. A little window should pop up, and have a symbol like `~>` or `~: $` in the top left. The `~` symbol means you are in the *home* directory of the user you are logged in as. This would be the equivalent of going to Windows and opening up the folder named after your user name. It generally has folders called Documents, Music, Downloads, etc. Open up your home folder in the interactive folder by going to the file system and clicking your user name or clicking 'home'.

## Contents list - ls

To list the contents of the directory, all you have to do is type **ls**. This will list everything. Type **ls** and see what you find in your home directory. Does this match with what you see in your filesystem window?

If you want to be a little more advanced, you can look for items with specific characters in their name by using the wildcard **\*** character. Say you wanted to find the folders that started with **D**, you would type **ls D\***. Linux would then go look for all the files and folders that started with the letter 'D'. However, let's say you wanted to find that one data file you can't remember what you called. You know it starts with 'tb' and ends with '.dat', but you don't know the rest. All you would need to type is **ls tb\*.dat**, and all the files that start with 'tb' and end with '.dat' in your current directory will appear.

## Make directory - mkdir

To make a directory, type **mkdir nameofdirectory**. For example, if you wanted to make a folder called *Learn*, you would type **mkdir Learn**. Make a directory called *LearnLinux*. To check if you made the directory properly, list the contents of your home folder. Also, check your interactive screen. Did it show up there as well?

## Change directory - cd

If you want to change your directory, you simply type **cd nameofdirectory**. For example, if we wanted to open the folder **THISDIR** in our home directory, we would type **cd THISDIR**. Enter the *LearnLinux* folder you have created. Do the same on the file system graphical interface.

If you want to auto-complete the directory name once you have the first few letters correct, you can always hit the tab button. It makes things more efficient while navigating. If there is more than one option, hitting tab twice quickly will display the remaining auto-complete options.

Furthermore, if there are subdirectories that you want to go to, you can use one instance of **cd**. For example, say you wanted to change to a subfolder called **Sub** in the folder **Fold**, you could type **cd Fold/Sub** and it would take you directly to the subfolder. You can place as many subdirectories in the destination, provided they are all separated by a '/'.

## List current working directory - pwd

Let's say you forgot where you are on the computer. In Windows, you usually just look at the path, or the directories you have clicked, displayed at the top of your window. In linux, all you have to do is type **pwd**. Find out what directory you are currently in and check that it agrees with the filesystem window.

## Text editing

Two text editors are generally used. In the lab manual, *pico* or *nano* are suggested, and are based within the terminal. However, you might prefer a more graphical interface, in which case *gedit* or *kedit* are also options. Gedit and Kedit are similar to notepad in windows. To run either editor simply type **nano FILENAME** or **gedit FILENAME &** into to the terminal to open the file called FILENAME.

### Nano or Pico

Nano or pico are identical text editors (one might be a new version, but there is no difference between the two), with very simple interfaces. Anything you want to type, you can type. However, when you want to do an action (save, cut out a line, search for a word, etc.), you must use the commands listed at the bottom of the screen. The carat (^) represents the 'Ctrl' button on the keyboard. As an example, to save and exit you would hit 'Ctrl' and 'x' at the same time. Nano will then ask what you want to call the file (the default would be the name of the file, i.e. FILENAME above; you can change it here), and then will save. after you hit enter.

Try making a file in your home directory with nano or pico called 'My-file.txt'. Type your favourite colour in the file, and save/exit. Check to see if your file exists both via the command line (**ls**) and in the filesystem window.

### gedit

Just like notepad in windows, gedit has all the same functions in the top menu bar. However, to start it, you type **gedit FILENAME &**. The & at the end opens the program up in the background so you have access to the terminal while you use gedit. If you don't want this feature, try running

**gedit FILENAME** and typing in the terminal. You won't be able to until you close gedit.

Open up your file Myfile.txt with gedit and add 'colour favourite is' after the colour you chose. Save the file and quit gedit. Now trying opening it from the filesystem window and see if you can see what you've added.

## Display contents - cat

You can display the contents of a file in the terminal using **cat FILENAME**, where FILENAME is the name of the file you want to see. Try running it on your file Myfile.txt. What do you see?

## Copy file - cp

If you wanted to make a carbon copy of a file (say to back it up, or you wanted to edit one thing in it but keep the original file), you can use **cp FILENAME NEWFILENAME**, where NEWFILENAME is the new version of the file. Copy your Myfile.txt file to MyBetterfile.txt and display it's contents with 'cat' to ensure the copying worked. This is identical to right click and copying the file in the filesystem window (try it to convince yourself).

## Delete file -rm

A file can be removed using **rm FILENAME**. Since we now have two files, why don't your remove the MyFile.txt file as your other file is now Better. NOTE: this removes the file from the system **PERMANENTLY**, and cannot be recovered with the recycle bin.

## Column editing -awk

You can do column operations with a command language program called awk. It is a bit tricky to summarize in single paragraph, but try typing this as an example **awk '{print "My",\$3,\$2,\$4,\$1}' MyBetterfile.txt**. What does the output say? In this example, we rearranged the columns (denoted by \$#, where # is the column number). You can also do mathematical operations and other fancy things with awk, but are beyond the scope of this tutorial.