UTSC Computational Physics Workshops, Week 1

Introduction to the Linux Terminal: Useful bash Commands

Fergus Horrobin, UTSC DPES

1 Introduction

We will use the **bash** terminal. This is the default terminal on most Linux and MacOS machines. There are many other flavours of the shell such as tcsh, zsh, csh... but bash is the most common and the one I use.

When you start terminal you will see there is a $\sim$ (tilde) symbol next to the username. This means we are in the Home directory for the current user. This is where all your personal files live.

Above this directory, the files are arranged as: /home/"username" ie. all the users Home directories are stored in a folder called **home**. Above that folder is the / directory containing all the system files. We will now navigate through stuff a bit to become familiar with it.

2 Directories

Let's try some commands to naviage through the folders. TIP: Use the "Up" and "Down" keys to recall past commands.

1. **cd 'dir':** Change directories. Changes to directory 'dir' (without quotes). If blank then changes to user's home directory. Add a / before the name to start from /, otherwise starts from current location.

2. **ls:** List the contents of the current directory. Alternatively specify a directory, as in cd to list that directory.

   Add -a after to show hidden files as well: ls -a

3. **ll:** Same as ls but with more details. Try them both to see the differences.

   Can add -a like ls

4. **pwd:** Print working Directory. Shows the full path of the current folder.

3 Files

Since the goal is to get you used to writing programs, let's talk about using files in the Terminal. We will be working with text files (Python is written in a plain text file).

1. **mkdir 'name':** Make Directory. Makes a new directory in the current folder called 'name' (without quotes).

2. **touch 'filename.ext':** Makes a new blank file called 'filename' with the extension 'ext'.

3. **gedit 'filename.ext':** Opens the file 'filename.ext' in Gedit, the default Ubuntu text editor. If the file does not exist, it creates it.

4. **rm 'filename.ext':** Deletes (permanently) the file specified.

   Add -r (recursive) to delete entire directory and all files inside it (Careful!): rm -r 'name'

5. **cp 'filename.ext' 'location/filename-new.ext':** Copy the file to the new location/name.

   Add -r to copy entire directory: cp -r 'name'

6. **mv 'filename.ext' 'location/filename-new.ext':** Move the file to new location/name. If location is not specified the file is just renamed in current directory.

   Add -r for same effect as cp: mv -r 'name'

7. **vim 'filename.ext':** The traditional built in text editor. Play around with it if you like but we will use gedit here since Vim has a steep learning curve.

4 Other useful commands

1. **man 'command':** Manual. Shows the manual for how to use a command.
2. **more 'filename.ext':** See the contents of a file without editing them. Good for quickly scanning through files.
3. **locate 'keyword':** Search for files containing the given keyword.

5 Wildcards

1. *** All Wildcard** We can use this with commands such as ls, ll, cp, mv to look for or work with all files that begin or end with certain characters. Examples:

   We can see all items starting with 'b' by typing: **ll b***

   We can copy all items ending with the .png extension by typing: **cp *.png 'location'**

2. **? Replace Wildcard** Indicates that we require a character at his position exactly. We can use this to show all files with extension of length 3:

   ls *.??? gets all files where extension matches formatting .???

6 Tips

1. Try pressing tab while typing a command. The terminal will auto-complete the rest of it. If nothing happens there is probably more than one option. Press tab a second time and it will list all the options for you.
2. If you get lost in directories, just type cd with no 'directory' and you will return home.
3. Locate and wildcards will prove valuable as you accumulate more files. Practice using them.