Introduction to Linux

David Croft

October 26, 2016

Abstract

This week we are going to be looking at the Linux operating system and how to get started in it.

1 Introduction

Way back in the day, the only way to use computers was to use command line interfaces. A Command Line Interface (CLI) was a simple text prompt where you had to type the commands that you wanted the computer to perform. Today most computers use Graphical User Interfaces instead where you click, drag and interact with a visual interface. While Graphical User Interfaces (GUIs) might be easier to use, they are not half as powerful as a good CLI.

Today we are going to be looking at the Unix terminal, this is the CLI found on Unix, Linux and other Unix-like computers (e.g. Macs). Windows also has a CLI but its pretty poor. It does have an alternative in the form of Powershell and Windows 10 has started offering the Linux CLI too. Understanding the Linux/Unix CLI will be invaluable when working with your Raspberry Pi (RPi) or Android devices.

2 Raspberry Pi

Time to set up your RPi.

Lab work: Hardware setup.

1. Make sure that you have connected the HDMI cable.
2. Make sure that you have attached you keyboard, mouse and inserted the MicroSD card.

   ! Don't forget the MicroSD, it goes on the underside of the RPi.
3. Finally attach the power cable.
4. When the NOOBS window appears select Raspbian.
   - It may take a while so be patient.

Your RPi is now set up. Once the RPi finished changing its settings it should reboot. Once it has finished rebooting if...

...you are looking at a desktop. Go to Menu → Accessories → Terminal.

...you are looking at a CLI. Login, the username is *pi* and the password is *raspberry*.

Lab work:

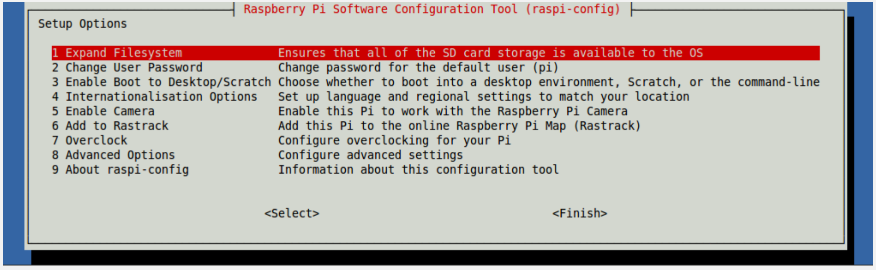
5. At the terminal type the command...

   ```
   $ sudo raspi-config
   ```

   If you are prompted for a password it will be *raspberry*.

   ! Don't include the $.

6. This should bring up a blue screen.



```
Raspberry Pi Software Configuration Tool (raspi-config)
Setup Options

  1 Expand Filesystem              Ensures that all of the SD card storage is available to the OS
  2 Change User Password           Change password for the default user (pi)
  3 Enable Boot to Desktop/Scratch Choose whether to boot into a desktop environment, Scratch, or the command-line
  4 Internationalisation Options   Set up language and regional settings to match your location
  5 Enable Camera                  Enable this Pi to work with the Raspberry Pi Camera
  6 Add to Rastrack                Add this Pi to the online Raspberry Pi Map (Rastrack)
  7 Overclock                      Configure overclocking for your Pi
  8 Advanced Options               Configure advanced settings
  9 About raspi-config             Information about this configuration tool

                 <Select>                                     <Finish>
```

7. Change your password to something more secure.

   ! Make sure you remember it!

8. Under Advanced Options

   - Set the hostname (to your university login name).
   - Enable SSH.

2.1 User accounts

Because of a limited number of network sockets in the labs we are going to suggest that while in the labs each project group works together on a single RPi. In order to let this happen you are going to need to create extra user accounts on one of your groups RPis.

Lab work: Create user accounts

9. Type the command:

   ```
   $ sudo adduser NAME
   ```

   Where NAME is the name of the new user you want to create.

10. Fill in the details that it asks for, you dont need to worry about rooms and phone numbers.

    - Do this for every member in your project group but you only need to do this on one of your RPi.

On Linux systems only certain users have the power to do things like install software or create accounts, these are known as superusers. On the RPis only the *pi* user will have these powers, so if you want to do stuff like install software then you will need to do it as *pi*. There are ways to give these powers to other users but that will not be covered in this lab.

3 Secure SHell (SSH)

One of the great things about the terminal is how easy it makes remote administration. It is possible to do this using GUI interfaces using remote desktop tools such as RDP, VNC and TeamViewer but these are often slower and limited to a single person connected at a time.

The tool that is mostly commonly used is SSH which stands for Secure SHell. It's a method of securely connecting to the shell (another term for command line) on another machine. Command line, terminal and shell all technically refer to different things but are realistically interchangeable in conversation.

Leave your RPi powered on and connected to the network but these next steps are done on a normal desktop PC. If you are in the labs then you are going to want to leave one RPi connected to the network and have everyone in your project group SSH into that one RPi.

Lab work: Secure SHell

1. Download an SSH program.
   - MobaXterm[a] has lots of features and has a free version.
   - Putty[b] is free and very widely used.

     I'm going to recommend you stick with MobxXterm for the moment but it's your choice.
2. Enter `HOSTNAME.coventry.ac.uk` as the hostname where `HOSTNAME` should be whatever you just set the RPi hostname to be.
   - If you are attempting this at home you will want to use `HOSTNAME.local` although this working depends on your home network.
3. At the prompts fill in your username and your password.
   - You should now be connected to the terminal on your RPi.

[a] `http://mobaxterm.mobatek.net/`
[b] `http://www.chiark.greenend.org.uk/~sgtatham/putty`

Once you have successfully SSH'd into your RPi the terminal will behave just like you were typing commands into it directly. Not that impressive since your RPi is sitting next to you but it could be anywhere in the world and still work!

4 Commands

Now that we are able to connect to our RPis it is time to learn how to navigate their file system. The file system is the way that the files and directories on your computer are stored and arranged.

- `ls` - list the files in the current directory.
- `cd NAME` - move into the directory called NAME.
  - `cd ..` - go up one directory.
- `cp SOURCE DEST` - copy the file/directory SOURCE to DEST.
- `mv SOURCE DEST` - move the file/directory SOURCE to DEST.
- `rm FILE` - remove the file named FILE.
  - `rm -r DIRECTORY` - remove the directory called DIRECTORY.
- `touch FILE` - create an empty file called FILE.
- `mkdir DIRECTORY` - make a directory called DIRECTORY.

Lab work: Navigating the directory tree.

11. Type the command:

    ```
    $ mkdir Documents
    ```

    You have now created a directory called Documents.

12. Type the command:

    ```
    $ ls
    ```

    What do you see? `ls` is an example of a command line tool. It's a simple program that performs a small, but important task. In this case listing the contents of a directory.

13. Type the command:

    ```
    $ cd Documents
    ```

    You are now in the Documents folder.

14. Type the command:

    ```
    $ pwd
    ```

    This stands for Present Working Directory, it tells you your current location in the directory tree.

15. Type the command:

    ```
    $ cd ..
    ```

    This stands for Change Directory, the `..` means that we should go up a directory.

16. You should now be back in your home directory. Use `pwd` again to confirm.

17. Create and navigate to a folder called Downloads.

Navigating the file system from the command line is easy once you get the hang on it but confusing to start with. So we are going to practise using the ls and cd commands in the magical world of Terminus.

Lab work: Play Terminus.

18. Go to `http://www.mprat.org/Terminus/`[a].

    ! The game doesnt work in Internet Explorer, it does work in Chrome. Other browsers are unknown.

19. See if you can find the town square. Once you are confident navigating Terminus it's time to come back to the real world.

[a] Backup location `http://web.mit.edu/mprat/Public/web/Terminus/Web/main.html`

Lab work: Dealing with files.

20. Create a file in your Documents directory using the `touch` command.

21. Check that the file is actually in the directory.

22. Rename the file you just created to test.txt.

23. Type the command:

    ```
    $ ls -l
    ```

    This is an example of using a command line flag. Flags are a way of configuring the behaviour of a program. In this case -l tells ls to list additional information about the directory contents.

5 Pipes

The real power of a CLI is that it allows you to combine multiple commands in order to achieve complex tasks. Piping is achieved using the pipe (|) character.

Lab work: Pipes

24. Type the command:

    ```
    $ ls | wc -l
    ```

    This is an example of combining commands and works like this:

    `ls` lists all the files.

    The | character (pipe) sends the output of that command to the input of the next command.

    `wc` is a program that counts the number of characters/words/lines, in this case the `-l` flag is telling it to count lines.

Lab work: Combining commands.

25. Download random.txt from moodle.

26. Figure out how to get the file onto your RPi (this is much easier if you are using MobaXterm).

27. Find the largest number in the file. You may want to investigate the `tail`, `cat` and `sort` commands.

    - How would you have done it using a GUI?

5.1 Redirection

Sometimes you are going to want to save the results of your commands to a file instead of just printing them to the console. This is called redirection and is achieved using `>` symbol.

Extended work: Redirection

28. Type the command:

    ```
    $ echo Hello World!
    ```

    What happens?

29. If we want to save this output to a file we have to type:

    ```
    $ echo "Hello World!" > hello.txt
    ```

    `>` tells the terminal that we want to put the output into the file hello.txt. Open hello.txt and check that the text is in there.

If you redirect to a file that already exists then that file will be overwritten. If you want to append to the end of a file then you can use `>>` instead.

Extended work: Redirection

30. Type the command:

    ```
    $ echo Hello again! >> hello.txt
    ```

31. Open hello.txt again and check that it's been updated.

32. Try and delete your hello.txt file.

    ! A word of warning, if something is deleted from the terminal, there is no Recycle Bin to save you if you make a mistake.

6 Deliverables

At the end of this week you are expected to have completed the following tasks:

1. Have a good understanding of moving around the directory tree from the command line (`cd`, `pwd`, `ls` etc.)
2. Be able to manage files (`touch`, `cat`, `rm`, redirection etc.)
3. Be able to chain commands together (`|`)
4. Have set up your Rpi and be able to SSH to it.
5. Blog update - Discuss how Terminus, a good old text adventure game, relates to command line tools and navigation. What do you think about the command line as an option?

6.0.1 Extension task

Investigate shell scripting, areas to look at would include:

- `sed`
- `grep`
- `for`
- `if`
- `export`