1. **The Shell**

What is the shell? The shell is basically a program that takes your commands from the keyboard and sends them to the operating system to perform. If you’ve ever used a GUI, you’ve probably seen programs such as “Terminal” or “Console” these are just programs that launch a shell for you.

almost all Linux distributions will default to the bash shell. There are other shells available such as ksh, zsh, tsch, but we won’t get into any of those.

Let’s jump right in! Depending on the distribution your shell prompt might change, but for the most part it should adhere to the following format:

username@hostname:current\_directory

pete@icebox:/home/pete $

Notice the $ at the end of the prompt? Different shells will have different prompts, in our case the $ is for a normal user using Bash, Bourne or Korn shell, you don't add the prompt symbol when you type the command, just know that it's there.

Let’s start with a simple command, echo. The echo command just prints out the text arguments to the display.

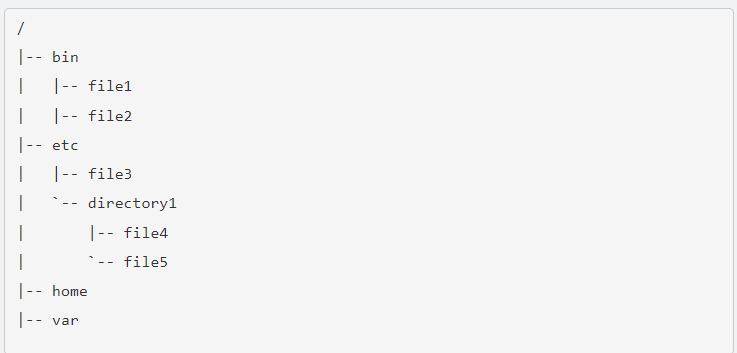
$ echo Hello World

**Exercises**

Try some other Linux commands and see what they output:

1. $ date
2. $ whoami
3. **PWD (Print working Directory)**

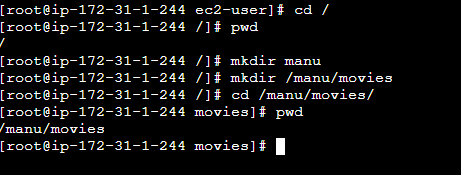
Everything in Linux is a file, as you journey deeper into Linux you’ll understand this, but for now just keep that in mind. Every file is organized in a hierarchical directory tree. The first directory in the filesystem is aptly named the root directory. The root directory has many folders and files which you can store more folders and files, etc. Here is an example of what the directory tree looks like:



The location of these files and directories are referred to as paths. If you had a folder named home with a folder inside of it named manu and another folder in that folder called Movies, that path would look like this: /home/manu/Movies, pretty simple huh?

Navigation of the filesystem, much like real life is helpful if you know where you are and where you are going. To see where you are, you can use the pwd command, this command means “print working directory” and it just shows you which directory you are in, note the path stems from the root directory.

$ pwd



**3. cd (Change Directory)**

Now that you know where you are, let’s see if we can move around the filesystem a bit. Remember we’ll need to navigate our way using paths. There are two different ways to specify a path, with absolute and relative paths.

* Absolute path: This is the path from the root directory. The root is the head honcho. The root directory is commonly shown as a slash. Every time your path starts with / it means you are starting from the root directory. For example, /home/manu/Desktop.
* 
* Relative path: This is the path from where you are currently in filesystem. If I was in location /home/manu/Documents and wanted to get to a directory inside Documents called taxes, I don’t have to specify the whole path from root like /home/manu/Documents/taxes, I can just go to taxes/ instead.



Now that you know how paths work, we just need something to help us change to the directory we want to. Luckily, we have cd or “change directory” to do that.

So now I've changed my directory location to /home/manu/movies.

Now from this directory I have a folder inside called Hawaii, I can navigate to that folder with:

$ cd Hawai



Notice how I just used the name of the folder? It’s because I was already in /home/manu/movies.

It can get pretty tiring navigating with absolute and relative paths all the time, luckily there are some shortcuts to help you out.

* **.** (current directory). This is the directory you are currently in.
* **..** (parent directory). Takes you to the directory above your current.
* **~** (home directory). This directory defaults to your “home directory”. Such as /home/manu.
* **-** (previous directory). This will take you to the previous directory you were just at.

