EN.540.635 "Software Carpentry"

Lecture 3 - Linux, SSH, and SFTP

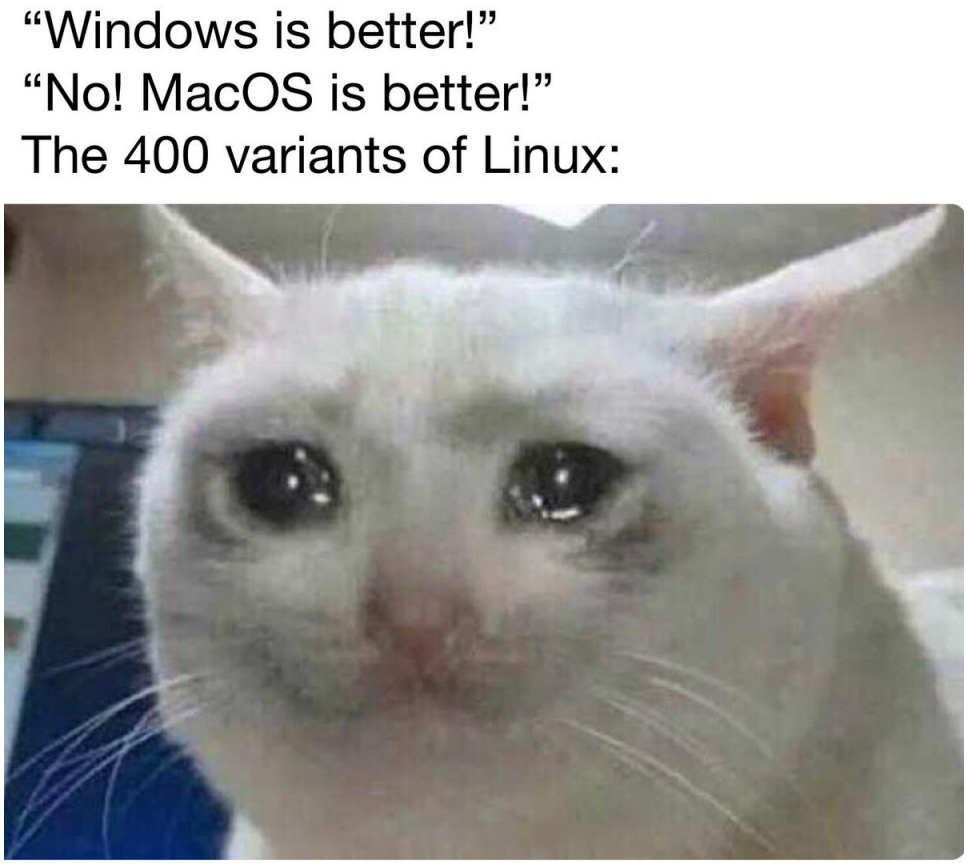

Figure 1: Credit: @ProgrammersMeme

Introduction

Now that we have seen what goes into computers on the hardware level, let us take a look at the software level. What makes it such that when you hit the power button, you can communicate with a machine? Here we will discuss an overview of the Operating System (OS) as a whole, with a final focus on Linux and networking.

Operating Systems

What exactly is an OS? An OS is many things that come together to allow a computer to run. An example OS would be "Windows 10", or "macOS Catalina", or "Ubuntu - Bionic Beaver". So, if an OS runs a computer, what exactly does that entail?

1. Kernel - This interfaces with the CPU and Memory. It manages memory accessibility in the RAM for programs and manages which programs get access to the hardware resources. Generally written in C because of the portability.
2. User Interface - Allows the user to interact with the hardware
3. Networking - Network protocols for sharing resources between computers. As such there is use of a common protocol. Network services such as SSH and SFTP.
4. Security - Distinguish whether certain commands should be processed or not based on user identity and therefore processes for authentication of identity should be present.

Linux

There are various flavours (distributions) of Linux OS's. That is, using the Linux Kernel, we have many different OS's. Some examples include Ubuntu, Fedora, CentOS, Debian, and Arch Linux. These OS's are widely used, from Academia to Industry. Linux is used in every Android phone, most Super Computers, and SpaceX (to name a few applications). Interestingly enough, the first major movie to be produced on a Linux machine was the Titanic, and over 95% of hardware at animation and visual effect studios use Linux. Linux is such a prolific OS in the modern day that it becomes near criminal for those in STEM degrees to not have an exposure to it.

It's not hard to understand why Linux has become so popular. With being open-source (based on the GPLv2 License), and the fact that you can use something (for free) and sell it later on (as long as you follow stipulations in the license), Linux has become the most widely used OS! Note, it still lags considerably in personal use machines (making up 1.8% in 2016), with Windows taking up the vast majority (89.7%), and Mac OS the remainder (8.5%).

GPLv2 License
You may copy, distribute and modify the software as long as you track changes/dates in source files. Any modifications to or software including (via compiler) GPL–licensed code must also be made available under the GPL along with build \& installinstructions.

SSH and SFTP

Secure SHell (SSH) and Secure File Transfer Protocols (SFTP) are safe ways of making a connection between two computers. They involve sharing something called "keys" to encrypt messages between the two machines, allowing for a secure way of remote-accessing a computer, as well as transferring files back-and-forth. On a Windows machine, we can very easily accomplish both via the MobaXTerm software. On Linux/MacOS, however, we must open up the terminal and connect manually. This is simple though, and can be done as follows:

```
ssh -X user@remoteHost
sftp user@remoteHost
```

Note the -X in the ssh call. This simply adds something called "X11-Forwarding", so that we can forward a display from the machine we are connecting to, to our screen. For instance, if you ssh into another machine (say, ssh -X henry@login.marcc.jhu.edu), you may want to open Firefox and see the browser. This will only happen if the -X flag is specified.

Now, when connecting to sftp this way, you must keep track of two disconnected file systems. That is, using ls and cd you can maneuver the file system of the machine you connected to, and using lls and lcd you can maneuver the file system of the machine you are connecting from. For example, assume I am in the directory /fs/home/henry on my laptop, and I connect to a cluster that puts me in folder /x/y/z. If I did lcd ../, my local directory would then be /fs/home. If I did ls ./, I would see the files and folders in the /x/y/z folder of the machine I am connected to. Now, if I want to get a file from the machine I am connected to, and bring it to my computer, I would use the get command. Similarly, if I want to put a file from my computer onto the machine I am connected to, I would use the put command. Thus, to get a file named "test" from the machine I am connected to (that is, test exists on the cluster, NOT on my laptop), I would use get test. To put a file named "newTest" onto the machine I'm connected to (that is, newTest exists locally on my laptop), I would use put newTest.