**Introduction**

Linux is an operating system with many 'flavors', the most common of which are Ubuntu, Debian, and Red Hat distributions. A distribution is a specific type of Linux OS which has specific ways of installing software and running the OS.

Linux is used in many server applications, such as for big data, serving websites, and running critical software processes for organizations.

**Commands**

pwd – print working directory, shows you where you are in the filesystem

cd – change folder

cd ..– navigate to the parent folder

cd folder – navigate to folder in the current directory

cd ./folder– navigate to folder in the current directory

cd /home/user/folder – navigate to the folder using an explicit path

ls – list files/folders within the current directory

ls -al– list command with options. a means show hidden files, l means show with more details

rm – remove a file

rm -r file\_name– remove a directory and its subcontents. -r means recursive

rm file\_name – remove a file

touch – creates a file

touch my\_new\_file.txt

sudo – log in as super user. Allows you to access root permissions

head – shows top of a file

cat – scrolls through file page by page

tail – shows end of a file

man – bring up manual for a command. I’d recommend you google the command instead of reading the man page

which – locates a file or a command

which python– shows you where python is if it’s on the path

whereis – shows more files than which

echo – prints out whatever follows

export – set an environment variable

source – runs a file

source ~/.bashrc-- ~ means home directory, bashrc is your startup script. When you configure things yourself, you want to export path variables in that bashrc script.

chmod – change permissions for read write and execute

chmod -R 777 name\_of\_file\_or\_directory – gives all users read write and execute access

chown – changes group and owner of files or folders

Usability

ctrl + c – cancels a running process

ctrl + d – quits a process

pressing tab will auto complete things. Press it a bunch of times if you have a lot of things with the same string matching.

**Environment Variables**

Environment variables store information for the OS to use. A simple example is the home directory, or $HOME. We can examine the contents of this variable with echo $HOME. Note we need a dollar sign before the variable name. If we want to create a new environment variable, we can do export NEW\_VAR=testvar. We can check that it worked with echo $NEW\_VAR. If an environment variable doesn't exist or have a value, it'll print out a blank line.

Another important example is the PATH environment variable. This stores filepaths/directories where the OS will looks for commands when you type them. Often this includes /usr/bin, where many executable files are. We can see the PATH environment variable with echo $PATH. To use an environment variable in the terminal, we need to put a dollar sign before it. Environment variables are present in all OSs (Windows, Mac, Linux), and the PATH in particular is in all of them. You'll see it has filepaths separated by colons in Linux. We can add a new folder to our path like so:

export PATH=/usr/bin:$PATH

In that case, we added the /usr/bin folder to the PATH, and our OS will check the /usr/bin folder first any time it's looking for a command.

We can also store environment variables in a few files:

* ~/.bashrc
* ~/.profile

The '~' is the same as $HOME, our home directory. The ~/.bashrc file is loaded every time we create a new terminal, and the ~/.profile is loaded each time we log in. We can re-run these with source ~/.bashrc if we change them and want the changes loaded.

**Users and file/folder ownership**

There are users within our OS, just like with Mac and Windows. These users can be created and deleted. The "ultimate" user is 'root', which has full power to do anything on the system. When we run commands with sudo before them, we are running it as root.

When doing ls -l, we see the owners of files, as well as other information. We won't go into it now, but we can change ownership with chown, and change permissions (e.g. can a file be executed) with chmod.

**Installing software**

The different Linux distributions install software slightly differently. Ubuntu and Debian distros use the aptitude package manager with .deb files. These are like .exe files in Windows, or .pkg files in Mac. We can update and upgrade our current software in Ubuntu/Debian with:

sudo apt-get update -- checks for updates

sudo apt-get upgrade -- upgrades current software

sudo apt-get install -- installs new software

Another Linux distribution, Red Hat (and a subversion of Red Hat, CentOS) uses the yum package manager and .rpm files. So if you find yourself with some unknown Linux distribution, then you can try the commands apt and yum to see which one works. This will at least tell you if it's an Ubuntu/Debian version or Red Hat / CentOS type Linux. Updating software with yum works like:

yum update