ENTRY

This is a draft

WHAT IS LINUX

Linux is an operating system like Windows or Mac OS. This operating system is very light weight allowing for small tech devices to run it at full strength. Debian and Ubuntu are some of the most popular types of Linux operating systems. Linux isn't like other operating systems. Windows and Mac use what is called a GUI. Take the file explorer of windows for example. This shows you in a nice picture perfect format the files on your computer. Linux, however, uses what is called the terminal. This is a text based version of the file explorer. This is where you will move around your computer making and reading files.

--get them into a command terminal shell--

Let's start simple

Type the following command

echo Hello World

You can see that the terminal responded with "Hello World"

The echo command relays whatever is after it back into the terminal.

Type the following command

ls

ls is also known as list. This command lists all the files and folders that are around you.

if theres a file in the area

you can see that there is a x.txt when running the ls command. That is a file. But how do we read the file? That's where cat comes in. The cat command allows you to read the content of the file.

Type the following command

cat x.txt

as you can see the terminal read off the file and displayed it.

if there's only folders

You can see that the command responded with a bunch of folders, they are highlighted in blue.

if we want to move into a folder, we use the cd command

Type the following command

cd Documents/

now type ls

You can see that the response of ls after doing the cd command is different. Now you are listing what is in the documents folder

Now you can use the cat command to read the files that are in this folder.

if you want to go back to the previous folder you were in you use the .. with cd

Type the following command

cd ..

then type

ls

now you can see that the ls responded with the same results from when you first typed ls

let's turn it up a notch

Type the following command

echo Hello World && ls

you can see that the terminal responded with hello world then the content around you

The && command is used between two commands

This will run the first command then run the second command

there is a similar command which is just one & symbol, this will run both commands at the same time

Type the following command

echo Hello there > hi.txt

Read the hi.txt file

the > allowed us to create a file and put the results of the response into the file. if the file already exists, its going to replace the content

Type the following command

echo general kenobi >> hi.txt

cat hi.txt

we can see ‘hello there’ is still in the file. using >> adds the response from the terminal to the end of the file

commands learned

echo

ls

cat

cd

& - run two commands at once

&& - run the first command, then the next one

'>' - redirect output to somewhere

'>>' - appends the output instead of replace

INTERMEDIATE

TREE

pwd

whoami

privileges

owners

groups

su

whos root

Now that you understand some of the basic maneuvers of the linux terminal, lets get to know the environment you're in. You can think of the linux operating system as a tree. the root of the tree is the / folder. this folder holds every file in the system. users will be in the home folder in the / folder.

Type the following command

pwd

This command prints the working directory. This is the folder you are currently in. That is also called an absolute path. Absolute paths allow you to call, move, execute, anywhere in the linux machine.

Type the following command

cd /

This brings you directly to the / folder. This is technically an absolute path. an absolute path is the direct path to a folder. if you wanted to get to the absolute path of the toor user folder you would type.

/home/toor

You can be anywhere in the computer and go directly to this folder with this command.

Inside linux, commands can have multiple capabilities, this is done with the use of switches.

Type the following command

ls -la

If you remember the ls command, it lists the files in this directory. However we added the ‘-la’ switch with it. This can also be broken up into 2 switches ‘-l -a’. This would do the same thing.

The -l is asking for the long format for the list command. The -a is asking for all files.

The ls command does not initially print the hidden files. Hidden files are files that start with a . at the start of the name. This is commonly used with configuration files, bash files that the user does not necessarily need to use that often but still needs to be in their home folders.

The -l as I said before lists the files in long format. This format shows the permissions of a file, owner of the file, group owner of the file, the size of the file, the date and time of last edit, and then the name of the file.

The permissions of a file can be rather confusing.

The permissions are seen as 10 “-”

The first - is signifying the type of the file that it is. The most common types you are going to see is ‘-’ and ‘d’, there are others but you won't see them that often.

The ‘-’ is signifying that it is a regular file such as a text file.

The ‘d’ is signifying that it is a directory, or folder.

The directories are going to allow you to move in and out of them

The next 9 are used as permissions of the different users of the machine. These are split into 3 sections. The 3 sections consist of 3 ‘-’ in them.

The fist dash in these sections is the read permission. If this ‘-’ has a ‘r’ in its place, then the user this section is have the permission to read the

The next dash is the write section, this dash signifies whether or not this user can edit this file. Again, if there is a ‘w’ in this dash slot, then the user can write to this file, if there is a ‘-’ then they do not have permission to write to it. The last slot is the executable permission, this slot will allow the user section to execute the file. This will show an ‘x’ in the slot if the user has that permission, if not, again, there will be a dash.

If you are interested in seeing the type of file a file is you can use the ‘file’ command.

Type the following command

file hi.txt

The 3 sections are dedicated to different users. The first section is the permissions of the owner of the file. The next section is for the group of the file, and the last section is the permissions of every other user.

-rw-r–r– root root

In this example, this is a regular file. The owner can read and write to this file. The group owner can read the file. Everyone else can read the file. In this example root is the owner and group owner of this file so only root can write.

The user part is self explanatory, they are the owner of the file. The group owner is the group assigned to a file. Think of it as a company and their different divisions of workers. The group owner may be the division of the IT department. The IT employees will be a part of the group IT. So if a file says that IT is the group ownership, anyone in the IT group can do what is in the group section of the permissions.

Root is the king user. They can edit, move, delete, read, or configure anything in the computer. They’re very powerful and dangerous user. If you have root access to a machine, you completely own that machine. When you are in this user, you do not need permission for anything. Some commands will require you to have root permissions to do them. This is normal for commands that can be dangerous if done improperly. To be able to do the command you can run it one of two ways. Either changing to the root user or running sudo in front of the command.

Sudo goes before the command and it is stating that as the root user I want to do this command. If you have the permission to run a certain command as sudo, it will ask for your password then run the command. You may not have permission to run the command with sudo, in that case, you would need to change to the root user and run the command again.

To change users, you type ‘su “user”’. This will ask for the password of the user you want to change to and then change you into that user.

HARD

Now let's start with some more challenging commands.

Text editors. These will allow you to change the content of files in the command line. There are two that are the most commonly used, nano and vim. Nano is more user friendly and self explanatory than vim. Vim is built as an upgrade to the old vi text editor. Vi was the linux text editor made back in 1978 so some of the old linux users are going to prefer the vim editor because it is what they are used to.

Type the following command

Nano hi.txt

You should now see that you can edit the content of the hi.txt file. When you want to save, type CTRL + s. When you want to exit type, CTRL + x.

Linux can be a very complicated system to try to navigate and find the file that you are looking for. Let's say that you moved a file to another part of the computer a long time ago but you dont remember where. The find command can help you do that.

Type the following command

Find / -name hi.txt

This will tell you the absolute path of all the hi.txt files in the entire machine. The command starts with the find command, the location you want to start looking from, ‘/’ which means the whole computer, -name is saying i want files that are named the following where we put hi.txt.

This command can be very complicated to use and many types of options in it. If you want to learn more about a command you can use the man page of the command.

Type the following command

Man file

This will show you a manual to the command file and help you find what switch you need to complete what you want to do.

There are some files that are extremely long, maybe millions of lines. That can be very frustrating to search through the whole file to find a certain line of code. Grep is used to help find strings in a file.

Type the following command

Grep there hi.txt

This will show any occurrences of there in hi.txt. Again this command can be very complicated, using a lot of switches. To learn more of the command, you can look into the man page of the command.

We now have chmod. This command changes the permissions of a file.

Type the following command

Chmod +w hi.txt

You can now see that the hi.txt has added writing permissions to all sections. But this gives it to all sections, what if we want it only added to the group section. We can do it in the same format we just did, but there is a simpler syntax. This is the number code of permissions.

Type the following command

chmod 664 hi.txt

664 is saying that I want the file to be read and writable by the owner and the group but only readable to everyone else. Each number correlates to a section of the permissions.

First # is the owner, next # is the group, and the last # is everyone else. These numbers are the totals of the permissions that you want.

4 - read

2 - write

1 - execute

Total what you want as permission for that section and put it in the slot. So if you want a file to have read,write,execute for the user, group, and everyone it would be 777. If you want no one to have permissions of any kind, the code would be 000.

Commands im avoiding though id like to add

mv

rm

chown

mkdir

ifconfig

help

Advanced

Ports

ifconfig