Linux Notes

1)Working with Linux command line Basics

-**Sudo** means “Super user do” run programs with admin power

-typing **Info** on the command line with display a menu with major topics Also you can hit the / key and type any key words it will show a page about that key word

Ex:  
**info  
/wget**

Or if you want to get examples of its usage you can type  
**info wget examples simple**

Command **cd** means change directory so   
Ex  
**cd /windows  
cd /windows/test**

-**ls** means list so it will list all the contents in the current directory  
**ls – a** or **ls --all** shows files even hidden files in a directory

**ls -l** displays a list of directory in long form  
**ls -lh** displays file size in human readable format  
**ls -lht** displays in descending chronological order

-**pwd** means present work directory and will display it

-**mkdir** means make directory   
example  
**mkdir test**

Make text files with **nano** command  
example  
**nano test1**  
after done typing it crtl S to save then crtl X to exit

-**cp** means copy  
Example

-rm means remove  
**rm file?** Would remove all files with a single text after the word file like **file1  
rm file**\* would remove all files with the word file in its name regardless of the following characters  
**rm \*** will delete everything in the directory

-**mv** means move  
example  
**mv ../test\* .** would move all files from the previous directory with the name text to the current directory

-to be able to run a sh file you need to input **chmod +x NameofFile.sh** and to run it **./NameofFile.sh**

**Questions**copy/move a file to another directory

**Mounting an external drive**  
-first start off by making a Directory  
**mkdir e** (can be named anything)

-then you mount it to that directory  
**sudo mount -t drvfs e: /mnt/e**

-then to unmount the drive   
**sudo unmount /mnt/e**

**Making the SSH KEY**

1) first open windows terminal using ubuntu and type  
**ssh-keygen -t ed25519 -C "*your\_email@example.com*"**  
this will make a ssh key it will ask you where to save it and chose a location  
**/home/bro**

2) then it will ask for you to make a password

3) after you put the password, it will tell you your fingerprint   
**SHA256:1mhZgUcOXOmTYJNlkKqjtUb2BOx6kTjTogrHByPYA88**

4) Once you get it go to your GitHub page and go to setting and input that fingerprint to make the ssh connection.

5)after you input your fingerprint on GitHub. Locate the ssh file with **ls -a** and list it with **ls .ssh**  
it will show it something like this  
**id\_ed25519, id\_ed25519.pub**

6) Before adding a new SSH key to the ssh-agent to manage your keys, you should have checked for existing SSH keys and generated a new SSH key.  
**eval “$(ssh.agent -s)”**

And it will show something like **agent pid 70**

7) then you have tocheck and add an identity  
**ssh-add -l**

8) if we entered the above text and says **agent has no identities** then we have to add one  
**ssh-add ~/.ssh/id\_ed25519**

9) then after it says the identity was added use the **cat** command to read the file to you  
its will look something like this  
**cat /home/user/.ssh/id\_ed25519**it will display something like this  
**ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIL8E06i/8h6c50LanPeCgVKoZQsfwZXVPvHc1stYz6Ij** [**juaguil64@gmail.com**](mailto:juaguil64@gmail.com)

**Setting up GitHub**

-first of set a folder where you want your GitHub files to go to  
**mkdir Development**

-then you have to get the ssh link from your GitHub with the **git clone** command  
**git clone git@github.com:JuanAndOnly64/Simple-Calculator.git**

-then it will ask you to yes or no to continue with the fingerprint then it will connect

-after you confirmed the fingerprint connection it will load your GitHub repository into the Development directory

-now if you **cd** it and **ls** it will show the contents of that repository

**Changes in Repository**

-now if you made a change In any of the files and want to check which one then you can use  
**git status**this will show on the terminal which files was altered in any way

-if you want to see what you changed you can run  
**git diff**this will show you what changed in the file like adding or replacing the word it will show something like this  
-Hello World  
+Goodbye World

-to upload the changes to your GitHub account you must add the altered file and you can add a commit if you want after you added it then after that you just push it and boom it uploaded  
**git add NamOfFile.txt  
git commit  
git push**

Mounting

virtual box

**ubuntu iso**

**take notes on git**

study ssh

**look into bitwarden**

look into clipboard manager ditto

copy a file remotely: scp

**Permissions**

0777

4 main numbers

1st number:

2nd number: = bro

3rd number: Group = bro

4th number: World = any user on the system

3 types of permissions:

- read

- write

- execute

000 = no permission = 0

001 = just exec = 1

010 = just write = 2

011 = write and exec = 3

100 = just read = 4

101 = read, no write, exec = 5

110 = read, and write = 6

111 = all permission = 7

**change owner**: chown

**change modifiers:** chmod

**Symbolic Link**

**Soft Link:** A soft link is something like a shortcut in Windows. It is an indirect pointer to a file or directory. Unlike a hard link, a symbolic link can point to a file or a directory on a different file system or partition.

**Hard Link:** You can think a hard link as an additional name for an existing file. Hard links are associating two or more file names with the same [inode](https://en.wikipedia.org/wiki/Inode) . You can create one or more hard links for a single file. Hard links cannot be created for directories and files on a different file system or partition.

**Example for symbolic link**  
**ln -s /home/james/transactions.txt trans.txt** = Soft Link  
-if it has the -s it will make a soft link and

l**n /home/james/transactions.txt trans.txt** = Hard Link  
-if it doesn't have -s it will be a hard link

-when creating symbolic links first you need to know what you are going to link by first going to the location of what you are going want to link then when you name the link