

Lab 0, TA: Dhara Shah

What is Linux?

- The operating system that lets you use your computer the way you like, even if there are no buttons for what you want to do with your computer!
- The best way to use your computer
- Emancipation from buttons!

what is ssh?

- a road (= protocol) to connect you to a computer that you are not on, and let you use THAT computer's functionalities.

Why do we use ssh?

- you are on a long weekend break, and you want to calculate how many minutes your plane is delayed, from the display date-time to rescheduled date-time. You have a program sitting on your school computer, written in C, that calculates the minutes from these strings.
- How do you get to look at this program (a file with a bunch of C text) on your school computer on your phone?
- Even if you can look at that program, your phone doesn't have a way to compile and run C file. How can you use your school computer to run this program in C, and look at its results on your phone?

What can you do with ssh?

- see your files in the remote computer
- create, rename and delete folders (dir) and files
- modify your files
- run your code files in the remote computer
- see the results of the code that was run on the remote computer

What is PuTTY?

- Windows doesn't like you using ssh (or any protocol that you will learn in this class) to connect anywhere.
- PuTTY lets you use these protocols in its environment (the box that you see) and lets you connect to another computer

Who doesn't need PuTTY?

- Folks whose machine has mac or Linux operating system
- They can just open up the command prompt and type ssh command directly there

The most important concept of Linux: **EVERYTHING IS A DIRECTORY OR FILE**

Case study: Suppose Jhon Doe has GSU id **jdoue1@student.gsu.edu**, and has his school password **Puppies<3**

How does he ..	Command
ssh to snowball?	open PuTTY ssh jdoue1@snowball.cs.gsu.edu press enter when asked to enter his school password (Puppies<3 in his case, yours in your case)

```
dhara@ascsc-1-616-1:~$ ssh dshah8@snowball.cs.gsu.edu
dshah8@snowball.cs.gsu.edu's password:
Last login: Wed Jan 15 10:16:38 2020 from ascsc-1-616-1.dyn.gsu.edu

+ - - - - -
-          GSU Computer Science          -
-          Instructional Server          -
-          SNOWBALL (snowball.cs.gsu.edu) -
-          - - - - -                     -
+ - - - - -

[dshah8@gsuad.gsu.edu@snowball ~]$
```

see where he “landed?” (everything is a file or a directory, remember?)	pwd (shows which directory you are in) (/home/jdoue1)
---	--

```
[dshah8@gsuad.gsu.edu@snowball ~]$ pwd
/home/dshah8
[dshah8@gsuad.gsu.edu@snowball ~]$
```

make a new folder(directory) named kitties?	mkdir <dir name> mkdir kittens
See that “kittens” was made?	ls

```
[dshah8@gsuad.gsu.edu@snowball ~]$ mkdir kittens
[dshah8@gsuad.gsu.edu@snowball ~]$ ls
kittens
[dshah8@gsuad.gsu.edu@snowball ~]$
```

change the name of the directory from kittens to puppies?	mv <old dirname> <new dirname> mv kittens puppies
---	---

```
[dshah8@gsuad.gsu.edu@snowball ~]$ ls
kittens
[dshah8@gsuad.gsu.edu@snowball ~]$ mv kittens puppies
[dshah8@gsuad.gsu.edu@snowball ~]$ ls
puppies
[dshah8@gsuad.gsu.edu@snowball ~]$
```

go inside the directory puppies?	cd <dir path> cd puppies
make sure he indeed is in puppies directory?	pwd

```
[dshah8@gsuad.gsu.edu@snowball ~]$ pwd
/home/dshah8
[dshah8@gsuad.gsu.edu@snowball ~]$ cd puppies
[dshah8@gsuad.gsu.edu@snowball puppies]$ pwd
/home/dshah8/puppies
```

Make a file named "my.puppynames"? (and write a bunch of names in there?)	nano <filename> and then follow the prompt
---	---

What is vi or nano?

- Your super basic "microsoft word" :)

know that he indeed created a file named my.puppynames?	ls
---	-----------

```
[dshah8@gsuad.gsu.edu@snowball puppies]$ nano my.puppynames
[dshah8@gsuad.gsu.edu@snowball puppies]$ ls
my.puppynames
[dshah8@gsuad.gsu.edu@snowball puppies]$
```

see what is in this file (first few lines), without actually opening it?	cat <filename> cat my.puppynames
--	--

```
[dshah8@gsuad.gsu.edu@snowball puppies]$ cat my.puppynames
coda
fonz
```

Wants to make a copy for his friend?	<code>cp <file that you need to copy> <name of the copied file></code> <code>cp my.puppynames share.puppynames</code>
--------------------------------------	--

```
[dshah8@gsuad.gsu.edu@snowball puppies]$ cp my.puppynames share.puppynames
[dshah8@gsuad.gsu.edu@snowball puppies]$ ls
my.puppynames  share.puppynames
[dshah8@gsuad.gsu.edu@snowball puppies]$ cat share.puppynames
coda
fonz
```

Change the name 'share.puppynames' to 'sweetnames'	<code>mv <oldname> <newname></code> <code>mv share.puppynames sweetnames</code>
--	--

```
[dshah8@gsuad.gsu.edu@snowball puppies]$ mv share.puppynames sweetnames
[dshah8@gsuad.gsu.edu@snowball puppies]$ ls
my.puppynames  sweetnames
[dshah8@gsuad.gsu.edu@snowball puppies]$
```

what is the difference between mv and cp?

wants to delete the file sweetnames	<code>rm <filename></code> <code>rm sweetnames</code> <code>ls</code> (to see if the file is deleted indeed)
wants to step out of the current directory and go to home (the place where he lands after ssh)	<code>cd ~</code> <code>pwd</code> (to see that it indeed happened)
delete the whole directory puppies	<code>rm -r puppies</code>

What is the difference between rm and rm -r ?

What happens if you still have puppies > my.puppynames and you try rm puppynames?

What happens if you remove my.puppynames, and then try rm puppies?

Now, can you tell the difference between rm and rm -r ?

```
[dshah8@gsuad.gsu.edu@snowball ~]$ nano hello.c
[dshah8@gsuad.gsu.edu@snowball ~]$ cat hello.c
#include <stdio.h>
int main(){
printf("yaw\n");
}
[dshah8@gsuad.gsu.edu@snowball ~]$ gcc hello.c -o hello
[dshah8@gsuad.gsu.edu@snowball ~]$ ./hello
yaw
[dshah8@gsuad.gsu.edu@snowball ~]$
```