*TAB can autofill in a file name (ex: type in FIL then hit TAB, FILENAME appears
** which SOFTWARE, will tell you location of software (ex: which vim or which emac)
*** ls -l vs. ls -alh/ll, one is minimal info, the other shows hidden files

DS5110/CS5501: Linux tools, shell commands notes

1. ssh: ssh -i private_key USERNAME@ip_address (ex: public IPv4 DNS)
2. pwd: print working directory, tells u which directory you're in
3. ls: lists the content of a specified directory
4. touch: touch FILENAME, creates a file (can check with ls command)
5. nano/vim/emacs: nano FILENAME --> ^==ctrl, ex: ctrl+G==help
   vim FILENAME, 3 text editors
6. apt: apt update, updates repo address list, might need to change permissions (ex: sudo apt update) then can (sudo) apt install SOFTWARE
7. pip3: package installer for python3 (python related dependencies like Dask, Jupyter Notebook, etc.), install with sudo apt install ...smth pip3
8. wget: can download from link, wget LINK
9. mv: mv OLDNAME NEWNAME, change name of file or move file from one place to another
10. cp: cp SOURCE DESTINATION, copy file (source) with new name (destination)
11. scp:
12. cat: cat FILENAME, displays the contents of a file
13. head/tail: head/tail FILENAME, prints out first/last 10 lines of file
    (ex: head -n 24 FILENAME, show first 24 lines...)
14. mkdir:
15. man:
16. cd: cd DESTINATION, change directory, one dot (.) is cd and two dots (..) is parent working directory (back one step)
    (ex: cd .. ==> takes you back one step),
    cd by itself brings you back to home or cd ~ does the same
17. sudo/su:
18. chmod:
19. python3:
20. which: which SOMETHING, checks if you have it ==> nothing output means you don't
21. echo: echo $PATH, prints out the path information connected using :
22. |:

23. >: FILENAME > output, can change standard output (ex: file > pdf or file > txt)
24. >>: _____
25. &>: _____
26. wc: grep PATTERN FILENAME | wc ==> | connects two different commands aka the pipe operator, counts number of occurrences of pattern
27. grep: grep PATTERN FILE, can print out all lines that matches pattern
28. find: _____
29. &: _____
30. ps: _____
31. kill: _____
32. pkill: _____
33. htop: _____
34. df: _____
35. du: _____
36. lsof: _____

script.sh (can be an executable file)

#! shebang line, specify interpreter, (ex: ../bash)

wget filepath, url, etc.

commands (ex: grep | wc etc.)

Permissions:

chmod +x FILENAME.sh (changes it to executable)

adding a new path to PATH:

-go to the new path directory then:

vim .profile

source .profile

echo $PATH