# Intro to Linux

## Basic Terminal

| | |
|---|---|
| **ssh** | • short for **s**ecure **sh**ell<br>• **usage:** `ssh [host]@[computer].[otherIPStuff]`<br>  ○ for lab computers: `ssh [CSID]@[comp].cs.utexas.edu`<br>  ○ can get a list of active computers from the UTCS web page by searching “lab machines”<br>• works natively for Macs and Linux machines<br>• for Windows machines will need to use Putty<br>  ○ WinSCP is also a great program for Windows<br>• there is also a secure shell extension for Chrome |
| **[ctrl] + [alt] + t** | • this will open up a new terminal window for you<br>• is super convenient on the computers in the lab |
| **[up arrow]** | • this will bring up the last command you used in the your terminal<br>  ○ if used multiple times will keep going up in your command history<br>• can be useful when using long/ repeated commands<br>• the down arrow will go back through commands in the other direction |
| **history** | • this will print out a list of your previous terminal commands<br>• can be useful if you are trying to remember a complex command you did previously but can’t quite remember |
| **clear** | • this will totally clear your terminal screen<br>• can be useful when have just run something really complex and would like a clean slate |
| **man** | • short for **man**ual<br>• **usage:** `man [something confusing]`<br>  ○ example: `man grep`<br>• will give you helpful usage information about certain commands/ system calls<br>  ○ is especially useful to look at the flags for commands |
| **echo** | • this will basically just print something to your terminal window<br>• **usage:** `echo “hello world”`<br>• is really useful in bash scripts<br>  ○ **NOTE**: bash scripts are a way to run a collection of terminal commands as a single command |
| **[ctrl] + c** | • this will stop whatever is currently running in your terminal<br>• can be especially useful when you accidently run a program with an infinite loop<br>  ○ Or any other long running program/ script that you want to stop |
| | |

| | |
|---|---|
| **[ctrl] + [shift] + c** | • this will let you copy something from you terminal<br>• useful if trying to Google what an error means |
| **[ctrl] + [shift] + v** | • this will let you paste into the terminal<br>• useful if just Googled a way to do something cool |
| **sudo** | • short for **s**uper **u**ser **do**<br>• will allow you to run commands you normally aren't allowed to<br>• **usage:** `sudo [command]`<br>  ○ **NOTE:** you cannot run sudo on the lab machines |
| **exit** | • will close out of the terminal window without having the hit the little x in the corner |

## Directories and Files

| | |
|---|---|
| **ls** | • short for **l**i**s**t<br>• lists all the files/ directories in the current directory<br>• you might also want to try the sl command on the lab machines |
| **ls -al** | • will list all the files in the current directory along with their permissions<br>• permissions:<br>  ○ read - can view the stuff<br>  ○ write - can edit the stuff<br>  ○ execute - can run (for scripts and such)<br>• 3 sets<br>  ○ (owner) (group) (anyone) |
| **pwd** | • short for **p**rint **w**orking **d**irectory<br>• any easy way to know where you are in the file hierarchy if you forget |
| **cd** | • short for **c**hange **d**irectory<br>• used to navigate between directories in your file structure<br>• **usage:** `cd [directory]`<br>  ○ can use "cd .." to go back up the directory structure<br>  ○ can also put in a full path instead of just a directory name<br>  ■ "/" at front of directory will be an absolute path from your root directory<br>  ■ no "/" at front of directory will be a relative path<br>  ■ "." just means current directory so ./hello.txt is the same as hello.txt |
| **[tab]** | • this will autocomplete whatever you are currently doing in the terminal<br>• ex: `cd Doc + [tab]` would autocomplete Doc to Documents without you having to type out the whole thing |
| **mkdir** | • short for **m**a**k**e **dir**ectory<br>• will make a new directory for you |

| | |
|---|---|
| | • **usage:** `mkdir [directory name]`<br>  ○ **NOTE:** can also use relative vs. absolute paths instead of just a directory name |
| **cp** | • short for **c**o**p**y<br>• a way to make a copy of something in a different directory<br>• **usage:** `cp [source/file name] [destination]`<br>  ○ again can use relative or absolute paths for the source and destination<br>  ○ **NOTE:** This copies to destinate and keeps the original in source as well |
| **scp** | • short for **s**ecure **c**o**p**y<br>• a way to copy files between computers<br>• **usage:** `scp [source] [destination]`<br>  ○ from other computer: `scp [host]:[source/file name] [destination on your computer]`<br>  ○ to other computer: `scp [source/file name] [host]:[destination on other computer]` |
| **mv** | • short for **m**o**v**e<br>• a way to actually move files/directories around on your computer<br>  ○ also an easy way to rename directories<br>• **usage:** `mv [source] [destination]`<br>  ○ as usual you can use either a relative or absolute path for the source and destination |
| **rm** | • short for **r**e**m**ove<br>• deletes a file<br>• **usage:** `rm [file name]`<br>• helpful things:<br>  ○ `rm -rf [directory name]`<br>  ▪ will delete a directory and everything inside it<br>  ▪ use with caution, if you don't give a destination for this it will delete EVERYTHING from your current directory down |
| **touch** | • will either create a new file or update the last modified date on a file to the current date<br>• **usage:** `touch [file]` |
| **cat** | • short for **cat**enate<br>• will print a file's contents to the terminal<br>• **usage:** `cat [file]` |
| **chmod** | • used to change permissions<br>• **usage:** `chmod [new settings] [file]`<br>• new setting options |

| | |
|---|---|
| | <table><tr><th>Reference</th><th>Operator</th><th>Mode</th></tr><tr><td>u - user<br>g - group<br>o - others<br>a - all (everybody)</td><td>add<br>remove<br>= set exactly</td><td>r - read<br>w - write<br>x - execute</td></tr></table> |
| **grep** | • a way to search through file(s)<br>• **usage:** `grep [search for] [file]`<br>  ○ can search for things using regex<br>• helpful flags:<br>  ○ -n lists the line number next to matches<br>  ○ -r search recursively<br>  ○ * instead of a file name will search the whole directory |
| **find** | • used to find out where a file lives in your file hierarchy<br>• **usage:** `find [path] -name [file]`<br>  ○ if path is not given then will search the current directory and every directory it contains |
| **diff** | • short for **diff**erence<br>• shows the difference between 2 files<br>• **usage:** `diff [file 1] [file 2]`<br>• helpful flags:<br>  ○ -b ignore white space diffs<br>  ○ -i ignore case<br>  ○ --side-by-side - see differences next to each other |

## Redirection Input/ Output

| | |
|---|---|
| **\|** | • will make the output from command on the left the input for the command on the right<br>• ex: `man hello \| grep "hello"`<br>  ○ will search for the word hello in the man pages for hello (this will actually work on the lab machines) |
| **> and >>** | • will redirect output on left into the file on the right<br>• single > will will replace the contents of the file with the given output and double >> will append to the file<br>• ex: `echo "hello" > hello.txt`<br>• ex: `cat [file1] [file2] > [file3]` |
| **<** | • will redirect thing on the right to be the input for the thing on the left<br>• ex: `ProgramTakesInAge < 12`<br>• is really good for testing projects that take in user input |

## Java Specific

| | |
|---|---|
| **java -version** | • will tell you what version of Java is currently installed on your machine<br>• will also tell you if java is not installed on your machine at all |
| **javac** | • used to compile a java program<br>• **usage:** `javac [file]`<br>  ○ must have the .java extension<br>• if successful will create a .class file with the same name as the original Java file |
| **java** | • used to run a compiled Java file<br>• **usage:** `java [name of .class file]`<br>  ○ ex: java Test<br>  ■ don't put .class at the end of the file name<br>  ■ this would have come from compiling a file called Test.java |

## UTCS Specific

| | |
|---|---|
| **lpq** | • short for **l**ist **p**rinter **q**ueue<br>  ○ or at least that is how I remember it<br>• will give you a list of pending jobs on a given printer<br>• **usage:** `lpq -P[printer name]`<br>  ○ no space between the -P and the printer's name<br>  ○ the Linux printer in the 3rd floor lab is lw301 |
| **lprm** | • can use to remove all your pending jobs from a printer's queue<br>• **usage:** `lprm -P[printer name] [CSID]`<br>• very useful if the printer is backed up/ not working and you need to run to class soon without worrying about wasting paper |
| **chkquota** | • short for **ch**eck **quota**<br>• each UTCS student only gets a certain amount of space so it is important to check how much you are using at any given time<br>• if you reach 100% you will stop being able to save files/ do certain things<br>• **NOTE:** caching on certain browsers can cause this quota to fill up quickly, to fix just clear your cache folder |
| **du -sk ~/* ~/.??* \| sort -n** | • will basically tell you which files are using the most memory<br>• super useful if your disk quota is at 100% and you need to figure out what to delete |

## Fun Stuff

| | |
|---|---|
| **cal** | • will give you a little ASCII calendar of the current month with the current day highlighted<br>• can be useful when you are having a tired moment and forget what your life looks like |
| **date** | • will give you the current date and time as a string<br>• again useful if you are having a tired moment and just need to know time still works properly |
| **yes** | • will print the same phrase repeatedly in your terminal until you hit `[ctrl] + c`<br>• **usage:** `yes [some words]` |
| **cowsay** | • will take a phrase and print a little ASCII art cow saying that phrase<br>• **usage:** `cowsay "[some words]"`<br>• can also pipe things into cowsay<br>  ○ you could have a cow tell you your grep output<br>• there are also many other animals you could do<br>  ○ for a list do cowsay -l<br>  ○ **usage for different animal:** `cowsay -f [animal file] [some words]`<br>  ■ ex: `cowsay -f dragon-and-cow "hello"` |
| **fortune** | • will print a random fortune to your screen<br>  ○ there are some fun flags for this like, you should man fortune to see what they are<br>• these are particularly fun to pipe into cowsay |
| **xeyes** | • will pop up a screen with little eyes on your screen that will follow your cursor around<br>• can be entertaining when you are super stressed/ tired<br>• **NOTE:** won't work over ssh |