**Useful tips!**

* The **tab** key can be used to autocomplete a word
  + Example:
  + Say you have a directory with three subdirectories: testFolder, textFolder, and someFolder
  + If you type “cd te” and then press the **tab** key, it’ll list the subdirectories testFolder and textFolder as the options, but won’t autocomplete because it doesn’t know which directory you want to cd into
  + If you type “cd tes” and then press the **tab** key, the terminal will automatically fill your line as “cd testFolder”. This is especially useful if you have a file or directory with a very long name and you can use tab to save some typing pains!
* The **up and down arrows** can be used to scroll through and retrieve earlier commands, which can then be repeated by hitting enter.
* The **Ctrl+C** keystroke can be used to stop running programs that have crashed or otherwise gotten stuck. **Yes, you still need to use Ctrl+C on Mac instead of CMD + C.**
* The ***&*** can be used at the end of a command to divorce GUI windows from the terminal. This is useful when running programs like text editors, so that you can edit files and still use the same terminal to issue new commands. *Note: this is only usable on Linux OS, not Macs.*
* **“clear”** clears your terminal

*Quick Guide of Linux Commands:*

Note: These are Linux commands. Most are usable in Mac’s terminal (Open Finder → Terminal). Windows Command Prompt (cmd.exe) uses a different set of commands. If, at any time, you’d like to use Linux commands on Windows, Cygwin is a good option to download.

| **Commands** | **Usage Summary** |
| --- | --- |
| ***ls*** | List files in a directory |
| ***cp*** | Copy files |
| ***mv*** | Move files |
| ***rm*** | Remove file |
| ***cd*** | Change directory |
| ***pwd*** | Print working directory |
| ***mkdir*** | Make directory |
| ***rmdir / rm*** | Remove directory |

\*\* NOTE the absence of an ‘undo’ command…\*\*

*More In-Depth Reference for Each Command*

**Note:** This is far from being an exhaustive list of commands! If you are curious, feel free to look up commands on your own and try them out! Here’s a list of Linux Bash commands as a starting point*:* <https://ss64.com/bash/>. Depending on your operating system and shell, some of these commands may or may not work.

| *pwd* → print working directory  The *pwd* command prints the currently active (working) directory, also known as the pathname. This is our current location in the directory tree structure. If no directory is specified, all file commands work on files in this directory. |
| --- |
| *cd* → change directory  The *cd* command is used to traverse the tree-like file system structure, changing the currently active (working) directory to a different directory.  **Note: the angle brackets “< >” are not part of the command**   * *cd <directory\_name>* to take you from the working directory to the subdirectory named “*directory\_name”* * *cd .* to keep us at the current directory; note that there is a space between *cd* and dot(*.*) * *cd ..*  to take you one directory up in the hierarchy (directory tree) * *cd ~* to take us back to our home directory |
| *mkdir* → make directory  The *mkdir* command creates an empty directory.   * *mkdir <newdirectory>* creates a new empty directory called *newdirectory* in the working directory   **Note: when using mkdir to make directories, you CANNOT have spaces. The same applies to using ANY commands that deal with files and directories with spaces!**  **IF you want to have a directory with a space (or the backslash character!), make sure you use double quotes around the directory name. See the following wrong and right ways.**  Wrong way: mkdir ENGR 151 will make two separate directories. One called ENGR and one called 151.  Right way: mkdir “ENGR 151” will make one directory called ENGR 151.  Alternative right way: mkdir ENGR\ 151 will make one directory called ENGR 151. |
| *cp* → copy file  The *cp* command copies a file from a source to a destination.   * *cp <source\_file> <destination\_file>* copies the source file *source* to *destination*, which can either be a file or a directory. Creates a new file, *destination*, if one does not exist already. * Examples:   *cp hw1.cpp hw2.cpp* copies the file named *hw1.cpp* to a new file named  *hw2.cpp*  *cp hw1.cpp Private* copies the file named *hw1.cpp* to the directory *Private*. (The original file *hw1.cpp* is in the current directory, and the copy of that file is in the subdirectory *Private*) |
| *mv* → move file  The *mv* command is used to move or rename files. It works in a similar manner to the *cp* command, but the source (original) file is removed (no longer exists). A good analogy is the “cut” command on a Windows or Mac. |
| *touch* → make file (there are also other commands like *cat*; feel free to Google the many ways to create a file!)   * touch test.txt creates a blank text file called test * cat > test.txt creates a blank text file called test AND you can directly write into it. Then use Ctrl + C to exit out of it. |
| *cat* → view a file’s contents   * cat test.txt displays the contents inside the text file test |
| *nano* → ONE of the many ways to open a file and edit it from terminal   * nano test.txt opens the test.txt file and you can type in it to edit its contents |
| *rm* → remove files  The *rm* command is used to remove (delete) files from the computer.   * *rm <filename>* removes the file named *filename* in the working directory. |
| *rmdir* → remove directory  The rmdir command removes (deletes) an empty directory. **If the directory is not empty, the contents of the directory must first be removed using *rm***.   * *rmdir <mydirectory>* removes the **empty directory** *mydirectory* |
| *ls* → list files in a directory  The *ls* command lists all the files in a directory. If no directory is specified, it lists files in the working directory i.e. the directory that you are currently in.   * *ls <subdirectory\_name>* to list the files in a subdirectory of my directory * *ls <subdirectory\_name>/<subdirectory\_name>* to list the files in a subdirectory of a subdirectory of the working directory * *ls ~/<subdirectory\_name>* to list the files in a subdirectory of your home directory; You can use this to look in any folder in your home directory, no matter where you are currently located (aka it is not relative to your current location) |