Learning to use BASH(**Bourne**-**Again SHell)**

Pwd

Print Working Directory

Open

Open /users/tabishshahmohsin/desktop/coding

Open coding (if coding is in the opened directory.)

Tab

After writing the first few letters of the name of a directory or a file, can press tab to expand.

Ls

List

Used to list the items in the opened directory.

Can use ls directory/file.extension to list files of the directory without opening it in the terminal.

Cd

Change directory

Cd /directory opens the directory in the terminal.

Cd .. goes to the parent directory ( dot means the current directory and .. means the parent directory. )

Cd ~ changes the directory to /users/tabishshahmoshin (main directory).

Clear or Command + k

Clears the previous codes.

Touch

Touch file.extension creates a file in the opened directory.

Touch /directory/file.extension creates a file in the directory.

Nano

Used to edit a file

Arrow keys

Used to navigate to previous codes.

History

Helps you to look at previous codes even after they are cleared.

Mkdir

Make directory

Mkdir directory\_name makes a directory within the opened directory.

Rm

Remove

Rm filename.extension to remove/delete a file.

Rm /directory doesn’t work as it expects an option, Rm -r /directory works to delete the file or directory without opening it in the terminal.

Rm deletes everything permanently.

Rmdir

rmdir directory, but this will only delete an empty directory.

Cp

Copy files

cp source\_file destination

Mv

Move files.

rename Files.

mv [source\_file\_name(s)] [Destination\_file\_name]

code

writing code simply opens vs code.

How to run and perform code on terminal:

code hello.c

make hello or gcc hello

./hello

The first command, code hello.c creates a file and allows us to type instructions for this program. The second command, make hello, compiles the file from our instructions in C and creates an executable file called hello. The last command, ./hello, runs the program called hello. Make hello makes the file hello while gcc makes the file with name a.

Clang

clang array.c : compiles it into a.out

./a.out : Runs a.out

clang -o asdf array.c : compiles array.c into asdf

clang -o hello hello.c -1cs50 : To include libraries which don’t come with c like cs50.

clang is the same as gcc but a little better and user-friendly.

Making command-line arguments yourself in C:

#include<stdio.h>

int main(int argc, char \*argv[]){

printf("Hello, %s\n", argv[1]);

return 0;

}

* argc:
  + argument count
  + How many words did the human type at the prompt.
* argv
  + argument vector
  + It’s a list of command line arguments
* C will figure out the typed command and hand it to you with its length.
* Using “make greet” new then “./greet Tabish” will print “Hello, Tabish”
* If used “argv[0]” would have printed “Hello, ./greet”, userful to know which command did the user run in order to troubleshoot.
* If out of index, “(null)” will get printed
* If I run ./greet Tabish Shah then “./greet” is argv[0], “Tabish” is argv[1], “Shah” is argv[2] and the value of argc=3.

Exit Status

All programs which don’t run till infinity, as you see the prompt again. Exit status is a return value from the program itself which is by default zero.

echo $?

This arcane command shows the return value of the last program.

Using this 2 times consecutively will always return 0.