

Steven Tang
CSE 460
Winter 2017

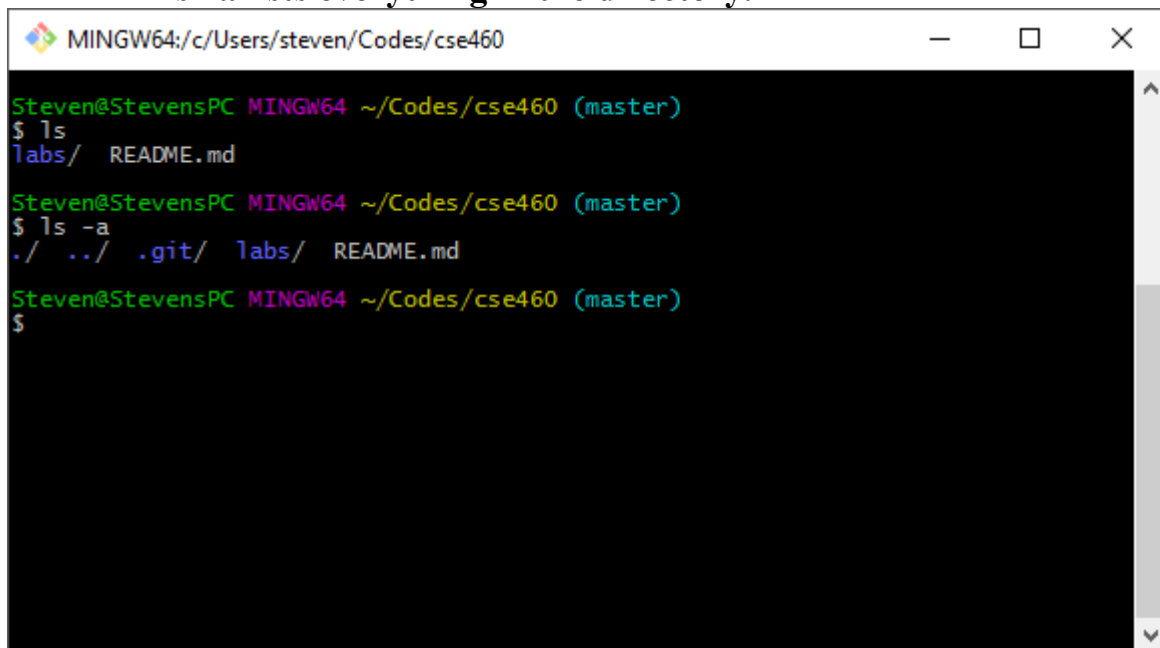
Lab 1

Objective:

In this lab, I am to give 2 examples of Unix commands: ls, cp, rm, mv, cd, mkdir, and vi. After giving examples of each command, I need to create a c++ file that would display a message that takes a string as the input and print it back out to the user.

LS EXAMPLE:

- **Basic ls list all visible folders and files in the directory**
- **Ls -a lists everything in the directory.**



```
MINGW64:/c/Users/steven/Codes/cse460

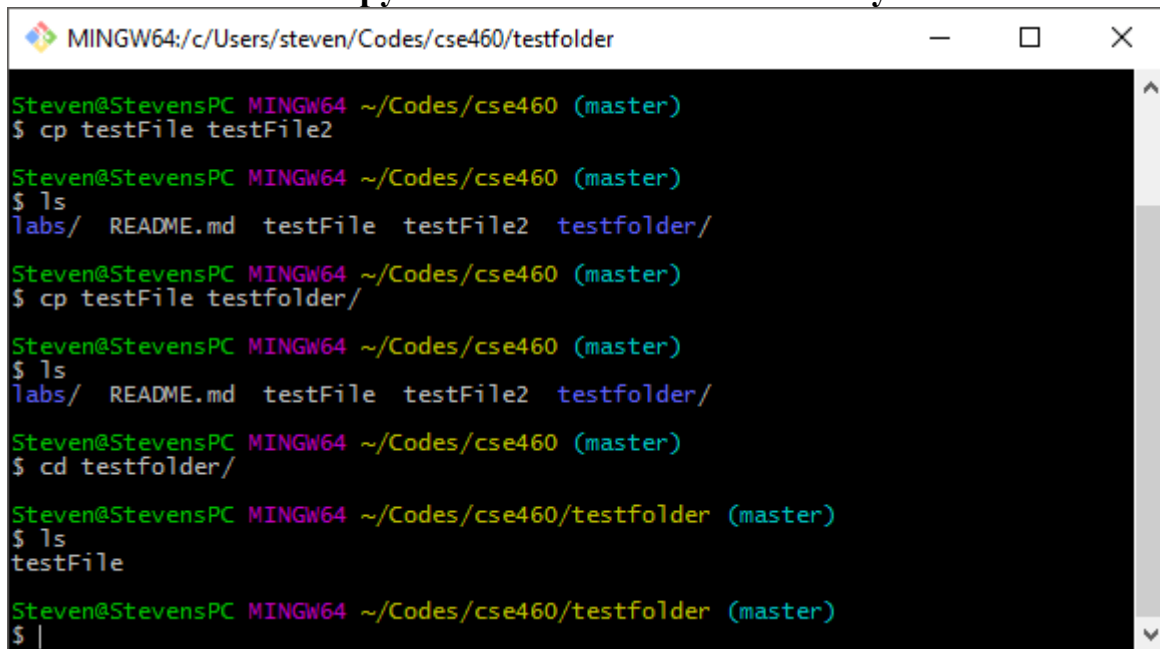
Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls
labs/  README.md

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls -a
./  ../  .git/  labs/  README.md

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$
```

CP EXAMPLE

- Cp can copy a file and rename that copied file
- It can also copy a file into a different directory.



```
MINGW64:/c/Users/steven/Codes/cse460/testfolder

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ cp testFile testFile2

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls
labs/  README.md  testFile  testFile2  testfolder/

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ cp testFile testfolder/

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls
labs/  README.md  testFile  testFile2  testfolder/

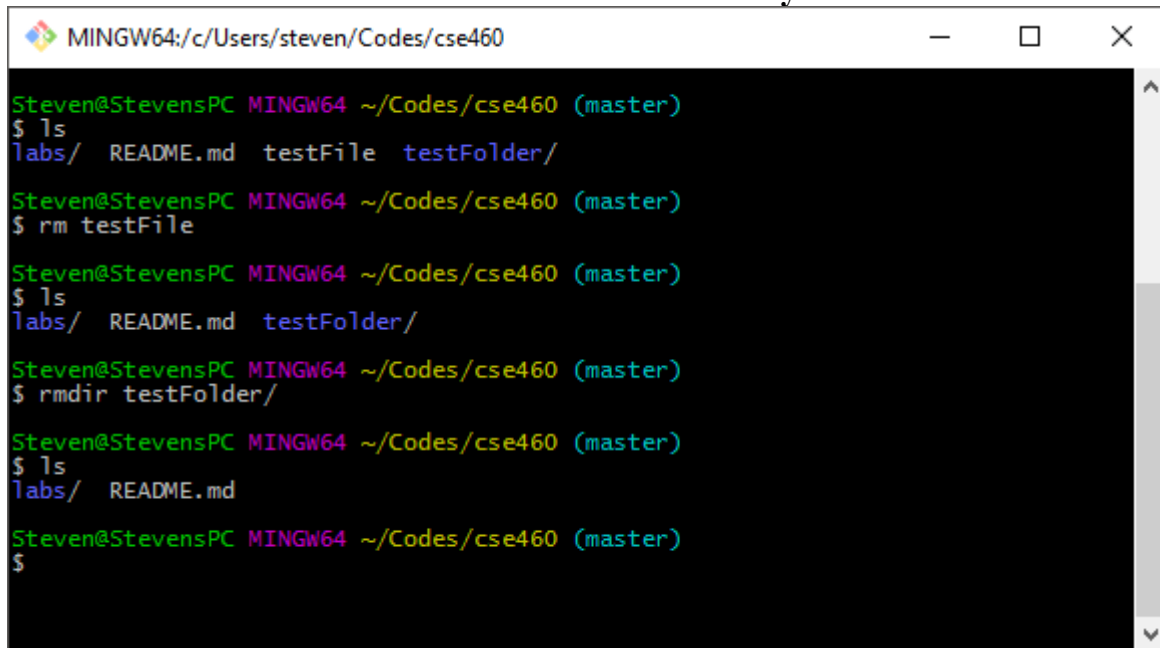
Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ cd testfolder/

Steven@StevensPC MINGW64 ~/Codes/cse460/testfolder (master)
$ ls
testFile

Steven@StevensPC MINGW64 ~/Codes/cse460/testfolder (master)
$ |
```

RM EXAMPLE

- **Rm** can remove a file
- **Rmdir** can remove an **EMPTY** directory

A terminal window titled 'MINGW64:/c/Users/steven/Codes/cse460' with standard window controls. The terminal shows a series of commands and their outputs. First, 'ls' is run, showing 'labs/', 'README.md', 'testFile', and 'testFolder/'. Then, 'rm testFile' is run. Next, 'ls' is run again, showing 'labs/', 'README.md', and 'testFolder/'. Then, 'rmdir testFolder/' is run. Finally, 'ls' is run a third time, showing only 'labs/' and 'README.md'. The prompt is '\$' throughout.

```
Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls
labs/  README.md  testFile  testFolder/

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ rm testFile

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls
labs/  README.md  testFolder/

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ rmdir testFolder/

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls
labs/  README.md

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$
```

MV EXAMPLE

- **Mv** can move a file to a different directory
- **Mv** can also **RENAME** a file

```
MINGW64:/c/Users/steven/Codes/cse460/testFolder

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ mv testFile testFolder/

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls
labs/  README.md  testFolder/

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ cd testFolder/

Steven@StevensPC MINGW64 ~/Codes/cse460/testFolder (master)
$ ls
testFile

Steven@StevensPC MINGW64 ~/Codes/cse460/testFolder (master)
$ mv testFile testFile1

Steven@StevensPC MINGW64 ~/Codes/cse460/testFolder (master)
$ ls
testFile1

Steven@StevensPC MINGW64 ~/Codes/cse460/testFolder (master)
$
```

CD EXAMPLE

- Cd changes directory
- Cd ~ changes to home directory

```
MINGW64:/c/Users/steven

Steven@StevensPC MINGW64 ~/Codes
$ ls
cse201/  cse313/  cse330/  Java/
cse202/  cse322/  cse460/  'Webpage Portfolio'/

Steven@StevensPC MINGW64 ~/Codes
$ cd cse460

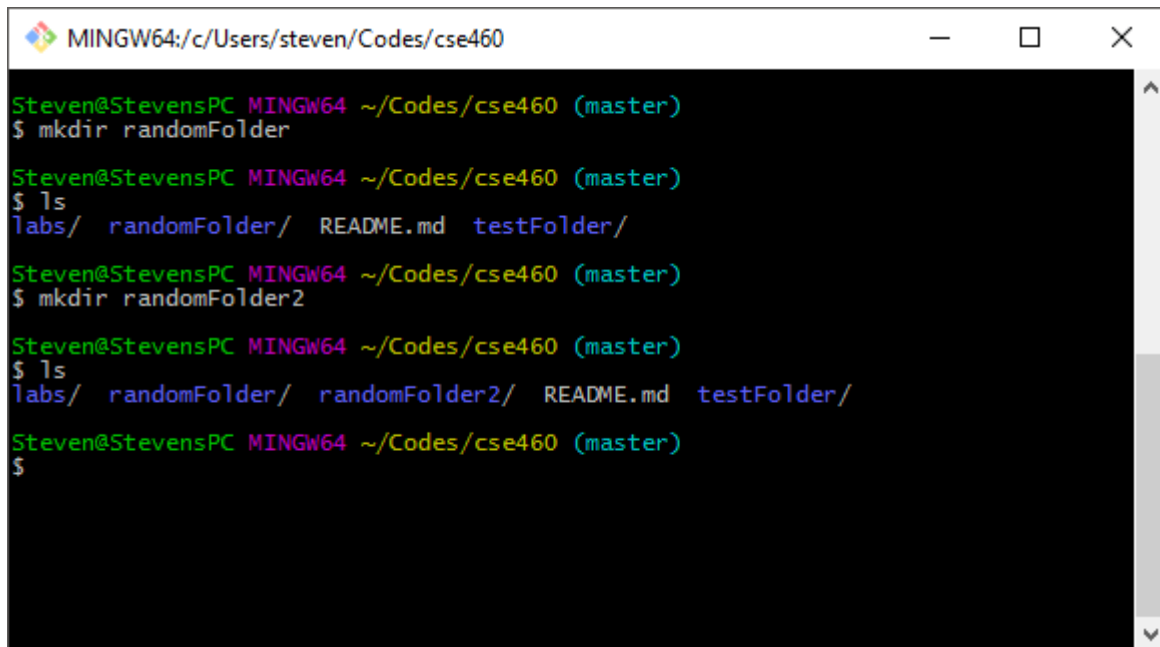
Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls
labs/  README.md  testFolder/

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ cd ~

Steven@StevensPC MINGW64 ~
$ |
```

MKDIR EXAMPLE

- Mkdir creates an empty directory



```
MINGW64:/c/Users/steven/Codes/cse460
Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ mkdir randomFolder

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls
labs/  randomFolder/  README.md  testFolder/

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ mkdir randomFolder2

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$ ls
labs/  randomFolder/  randomFolder2/  README.md  testFolder/

Steven@StevensPC MINGW64 ~/Codes/cse460 (master)
$
```

VI EXAMPLE

- Vi is a text editor for UNIX


```

3.
4. using namespace std;
5.
6. int main(){
7.     displayMessage();
8.     cout << "\n";
9. }

```

MESSAGE.CPP

```

1. #include <iostream>
2. #include "function.h"
3.
4. using namespace std;
5.
6. void displayMessage(){
7.     string message;
8.     getline(cin,message);
9.     cout << "\n" << message;
10. }

```

MAKEFILE

```

1. message: main.o message.o
2.     g++ -o message message.o main.o
3. message.o: message.cpp
4.     g++ -c message.cpp
5. main.o: main.cpp
6.     g++ -c main.cpp
7. clean: message.o main.o
8.     rm message message.o main.o

```

OUTPUT (Used script to record terminal)

```

1. Script started on Tue Jan 10 18:18:23 2017
2. [?1034hbash-3.2$ make[K[K[K[Kmake
3. g++ -c main.cpp
4. g++ -c message.cpp
5. g++ -o message message.o main.o
6. bash-3.2$ l[Kls
7. cd_example.png      main.cpp              message.cpp           rm_example.png
8. cp_example.png      main.o                message.o             typescript
9. function.h          makefile              mkdir_example.png    vi_example.png
10. ls_example.png      [31mmessage[39;49m[0m      mv_example.png
11. bash-3.2$ ./message
12. CSUSB CSE 460 the Beautiful!
13.
14. CSUSB CSE 460 the Beautiful!
15. bash-3.2$ make clean
16. rm message message.o main.o
17. bash-3.2$ scr[K[K[Kexit
18. exit
19.
20. Script done on Tue Jan 10 18:18:58 2017

```

Evaluation

I have given multiple examples of each command that was required and correctly developed a program that would take an input from the user and output it to the screen. I also correctly created a makefile that would compile most of the files for the user. In the makefile, I also made it an option to have it clean up all the .o files that were not used anymore.

Score: 20/20