Michael Smith

CSE 460

Lab 1

20 points total

1) Basic Linux commands

Ls – lists all files in a directory

user@user-virtual-machine ~ $ ls

Desktop Documents Downloads Music Pictures Public Templates Videos

Ls – a: the –a will list all the files in the current directory, including any hidden files

user@user-virtual-machine ~ $ ls -a

.   .dbus   .ICEauthority .sudo_as_admin_successful ..   Desktop
.linuxmint   Templates .bash_history .dmrc   .local   Videos .bash_logout
Documents   .mozilla   .Xauthority .bashrc   Downloads   Music
.xsession-errors .cache   .fontconfig Pictures   .xsession-errors.old
.cinnamon   .gconf   .pki .codeblocks .gksu.lock .profile .config
.gnome   Public

Cp – copies files or directories

user@user-virtual-machine ~/Desktop/cse460/lab1 $ ls

test.txt

user@user-virtual-machine ~/Desktop/cse460/lab1 $ cp test.txt copy.txt

user@user-virtual-machine ~/Desktop/cse460/lab1 $ ls

copy.txt test.txt

Also can copy the source to another locations

user@user-virtual-machine ~/Desktop/cse460/lab1 $ cp copy.txt New

user@user-virtual-machine ~/Desktop/cse460/lab1 $ cd New

user@user-virtual-machine ~/Desktop/cse460/lab1/New $ ls

Copy.txt

Rm – remove files

user@user-virtual-machine ~/Desktop/cse460/lab1/New $ ls

copy.txt

user@user-virtual-machine ~/Desktop/cse460/lab1/New $ rm copy.txt

user@user-virtual-machine ~/Desktop/cse460/lab1/New $ ls

user@user-virtual-machine ~/Desktop/cse460/lab1/New $

The –d modifier will remove a directory

user@user-virtual-machine ~/Desktop/cse460/lab1 $ ls

copy.txt New test.txt

user@user-virtual-machine ~/Desktop/cse460/lab1 $ rm -d New

user@user-virtual-machine ~/Desktop/cse460/lab1 $ ls

copy.txt test.txt

user@user-virtual-machine ~/Desktop/cse460/lab1 $

Mv – moves files or directories

user@user-virtual-machine ~/Desktop/cse460/lab1 $ ls

copy.txt Lab1 test.txt

user@user-virtual-machine ~/Desktop/cse460/lab1 $ mv test.txt Lab1

user@user-virtual-machine ~/Desktop/cse460/lab1 $ cd Lab1

user@user-virtual-machine ~/Desktop/cse460/lab1/Lab1 $ ls

Test.txt

Moving a directory

user@user-virtual-machine ~/Desktop/cse460/lab1 $ ls

copy.txt Lab1 Moved

user@user-virtual-machine ~/Desktop/cse460/lab1 $ mv Lab1 Moved

user@user-virtual-machine ~/Desktop/cse460/lab1 $ cd Moved

user@user-virtual-machine ~/Desktop/cse460/lab1/Moved $ ls

Lab1

Mkdir – will make directories within the current locations

user@user-virtual-machine ~/Desktop/cse460/lab1/Moved $ ls

Lab1

user@user-virtual-machine ~/Desktop/cse460/lab1/Moved $ mkdir CSE460

user@user-virtual-machine ~/Desktop/cse460/lab1/Moved $ ls

CSE460 Lab1

user@user-virtual-machine ~/Desktop/cse460/lab1/Moved $ ls

CSE460 Lab1

user@user-virtual-machine ~/Desktop/cse460/lab1/Moved $ mkdir LinuxCommands

user@user-virtual-machine ~/Desktop/cse460/lab1/Moved $ ls

CSE460 Lab1 LinuxCommands

Cd – will change directory

user@user-virtual-machine ~/Desktop/cse460/lab1 $ ls

copy.txt Moved

user@user-virtual-machine ~/Desktop/cse460/lab1 $ cd Moved

user@user-virtual-machine ~/Desktop/cse460/lab1/Moved $ ls

CSE460 Lab1 LinuxCommands

The .. Is a hidden file in each folder allowing cd to move backward easier.

user@user-virtual-machine ~/Desktop/cse460/lab1/Moved $ ls

CSE460 Lab1 LinuxCommands

user@user-virtual-machine ~/Desktop/cse460/lab1/Moved $ cd ..

user@user-virtual-machine ~/Desktop/cse460/lab1 $ ls

copy.txt Moved

Vi – a built-in text editor. Vi will let the user create a new file or edit an existing one. Does require being saved before exiting

user@user-virtual-machine ~/Desktop/cse460/lab1 $ vi vitest

user@user-virtual-machine ~/Desktop/cse460/lab1 $ ls

copy.txt Moved vitest

Not using the :wq command before exiting vi

user@user-virtual-machine ~/Desktop/cse460/lab1 $ vi test2

user@user-virtual-machine ~/Desktop/cse460/lab1 $ ls

copy.txt Moved vitest

2) Simple C++ Program

Main.cpp

```cpp
#include <iostream>
#include "display_message.h" //function header
int main()
{
    display_message();
    std::cout << std::endl;
    return 0;
}
```

Display_message.h

```cpp
void display_message();
```

Display_message.cpp

```cpp
#include <iostream>
#include <string>
#include "display_message.h"
using namespace std;
void display_message(){
    string message;
    getline(cin, message);
    cout << message;
}
```

3) Makefile

```makefile
HEADERS = display_message.h
default: message
display_message.o: display_message.cpp $(HEADERS)
    gcc -c display_message.cpp -o display_message.o
main.o: main.cpp $(HEADERS)
    gcc -c main.cpp -o main.o
message: main.o display_message.o
```

```
    g++ main.o display_message.o -o message
 clean:
   -rm -f main.o
   -rm -f display_message.o
```

4) Using the makefile and executable

      user@user-virtual-machine ~/Desktop/cse460/Lab1 $ ls

      bin   display_message.h Lab1.depend main.o   obj

      display_message.cpp Lab1.cbp   main.cpp   makefile

      user@user-virtual-machine ~/Desktop/cse460/Lab1 $ make

      gcc -c display_message.cpp -o display_message.o

      g++ main.o display_message.o -o message

      user@user-virtual-machine ~/Desktop/cse460/Lab1 $ ./message

      CSE 460 is beautiful

      CSE 460 is beautiful

Points: 20/20

I was successful in explaining all the commands with two examples of each. As well as created the program to take in a user message and display it back. Made a makefile to create a executable and successfully ran that executable.