

Lab Assignment 0

Getting Familiar with Linux

COL 100

February 28, 2021

1 Setting up a Linux Environment

The programming aspect of the course requires you to be comfortable with working in Linux environments and to be able to compile and run C programs. We have listed a few ways to setup Linux environments on different operating systems.

1.1 Windows: Cygwin

Cygwin provides a Linux based terminal to run Linux commands on windows. To install Cygwin with a gcc compiler go to [this](#) website and follow step **1** to **4**. For step **5** where it says 'select required packages' you need to select gcc and make, see pics [here](#) for reference.

1.2 MAC

Linux and Mac both are UNIX based environments so you should be able to run most of the commands without any changes.

1.3 Android: Termux

If you do not have a PC then you can install **Termux**. It is an application which provides Linux based terminal for Android devices. You can install it from the playstore.

1.4 Online: JSLinux

This is a pretty good online Linux terminal emulator where you can run most of the Linux commands. You can even edit, compile and run C programs. It comes with gcc compiler pre-installed. Create an account to save work across sessions.. Link [here](#).

1.5 Other Methods

Though above methods are recommended as they are fairly straightforward but if you want a full experience of a Linux OS then you may try installing one using a virtual machine or you can even install a Linux OS on your system which will be completely separate from your Windows OS.

1.5.1 Setting up a virtual machine (VM Ware)

VM Ware is a virtualization software that allows to simulate any OS over any other OS. You can install VMware and then install Linux based OS (E.g. Ubuntu) in it. You can follow the tutorial [here](#).

1.5.2 Dual Booting Linux with Windows

If you have windows installed on your device then you can dual boot your system with another OS like Ubuntu, which is an open source OS, and available for free. Link to a tutorial is [here](#).

2 VPN and Remote Access (Optional)

Instruction will be communicated separately.

3 Command Line Interface(CLI)

Command Line Interface is text based interface which is another method to interact with your PC without using graphical elements like mouse pointer, icons, windows, etc.

3.1 Basic Linux Commands

1. Make a new directory(folder): **mkdir** dirname
2. Create/Open text file using vi : **vi** filename
3. Copy file: **cp** src dest
4. Copy folder: **cp -r** src dest
5. Rename: **mv** originalname newname
6. Delete a single file: **rm** filename
7. Delete a whole folder: **rm -r** foldername
8. Change working directory: **cd** path
9. Move to parent folder: **cd ..**

10. List contents of a folder: **ls**
11. List contents of a folder including hidden files: **ls -a**.
12. Print current directory: **pwd**.
13. Open manual for a command: **man** commandname

3.2 Using Commands to operate your system

Open a Terminal: Cygwin(Windows), Termux(Android), Terminal App in Mac, JSLinux(Online), etc.

1. Create a new directory with the name *COL100* (**mkdir** COL100).
2. Change the current directory to *COL100*. (**cd** COL100)
3. In this directory, create another folder, called as *Lab0*. (use **mkdir**)
4. Change the current directory to *Lab0*. (**cd** Lab0)
5. Create a text file *me.txt* using **vi** (or any text editor, see section 4.1) and write your name and entry number in it. Save and close it (See section 4.1.1 for saving and closing a file in vi)
6. Copy the contents of *me.txt* to *mycopy.txt* using **cp**
7. Open *mycopy.txt* using vi(or any text editor) and verify that it is a copy of *me.txt*. Close the file
8. Rename *mycopy.txt* as *stillme.txt* using **mv**
9. Check if the file has been renamed by listing the contents using **ls**
10. Move out of *COL100* using **cd ..** twice. (**TIP**: You can retrieve previous commands you typed using the **UP arrow key** without typing them again.
11. Copy the directory hierarchy *COL100* to *COL100copy* using **cp -r**
12. Check the contents of the folder *COL100copy* by going to the folder (**cd**) and then listing the contents(**ls**).
13. Go out of the folder (**cd ..**) and delete the whole directory COL100copy using **rm -r**
14. You can use **man** to learn about different commands

4 Running a simple C program

You need a text editor to edit files and, the gcc compiler to compile C programs.

4.1 Installing an editor

You need a text editor to create/edit text files. You can lookup and install any editor that you find suitable for your machine. Some common examples are Atom, SublimeText, notepad++, gedit(Linux), Xcode(MAC), etc.

4.1.1 Terminal Based Editor: vi

This editor does not have a GUI and runs in the terminal itself on any OS. It comes pre-installed in almost all of the UNIX based terminals. Create/Open a file using 'vi filename'. Enter editing mode by pressing **i**. Enter your text and exit using following steps:

1. Press **Esc** to get out of the editing mode
2. Press **:** (i.e. shift+;))
3. Enter **wq** which stands for write quit and press enter
(Write **q!** instead of **wq** to quit without saving changes.)

4.2 Checking if gcc is already installed

In your terminal type the following command:

```
gcc --version
```

If the output is something like the following then gcc is already installed and you can skip 4.3.

```
gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There
is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
```

Otherwise if the the output says something like the following then you need to install gcc on the system.

```
command not found: gcc
```

4.3 Installing gcc

Try the installation and verify it by following 4.2 again.

4.3.1 Linux based (VM Ware, Dualboot, etc.)

Execute the following commands on the terminal:

```
sudo apt update
sudo apt install build-essential
sudo apt-get install manpages-dev
```

4.3.2 MAC

Follow the tutorial [here](#).

4.3.3 Android(Termux)

Type the following command:

```
pkg install clang
```

4.4 Creating a C program

Go to the folder *Lab0* you created and create a C file named *hello.c* using vi or any text editor and write the following code into it.

Note: For submission replace 'Hello World' with your name and entry number. E.g.: "James Doe 2020XX10234\n"

```
#include <stdio.h>

void main(){
    printf("Hello World\n");
    return;
}
```

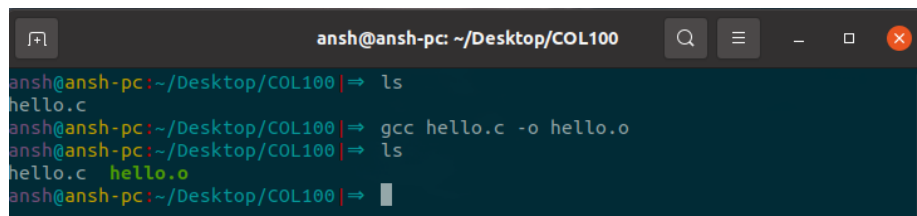
Save and close the file.

4.5 Compiling the program

Use the **gcc** compiler installed to compile the code as given below.

```
gcc hello.c -o hello.o
```

Notice, using `ls`, that a new file *hello.o* will be created in the directory as shown in the screenshot below.

A terminal window titled 'ansh@ansh-pc: ~/Desktop/COL100' showing the following commands and output:

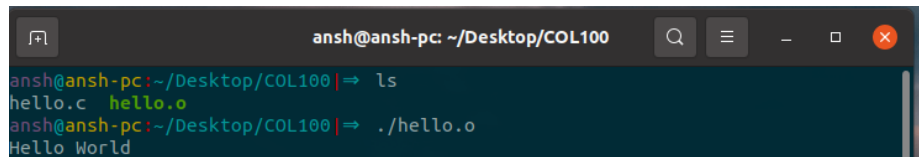
```
ansh@ansh-pc:~/Desktop/COL100|⇒ ls
hello.c
ansh@ansh-pc:~/Desktop/COL100|⇒ gcc hello.c -o hello.o
ansh@ansh-pc:~/Desktop/COL100|⇒ ls
hello.c  hello.o
ansh@ansh-pc:~/Desktop/COL100|⇒
```

4.6 Executing the program

In the same directory type the following command:

```
./hello.o
```

The program should output 'Hello World' on the terminal as shown here.

A screenshot of a terminal window with a dark background. The title bar at the top reads 'ansh@ansh-pc: ~/Desktop/COL100'. The terminal shows the following commands and output: the user runs 'ls' and sees 'hello.c' and 'hello.o'; then the user runs './hello.o' and the output 'Hello World' is displayed.

```
ansh@ansh-pc: ~/Desktop/COL100
ansh@ansh-pc:~/Desktop/COL100| => ls
hello.c  hello.o
ansh@ansh-pc:~/Desktop/COL100| => ./hello.o
Hello World
```

PS: You can use online IDEs(E.g. IDeone) to quickly edit, compile and run codes if you do not have anything setup on on your system. Note: they do not provide Linux environmnets and are just for editing and running code.

4.7 Submission

Now, having setup the environment and learned how to run C code. You are required to edit the above code, replacing "Hello World" with **your name** and **entry number** and submit the screenshot of output of the program as shown above.

Important: Your screenshot should clearly show the **whole terminal** with the **output** and the commands for **compiling** and **executing**.