Now, let’s see how we can execute a C program in Linux.

As mentioned earlier, I can connect to a Linux computer managed by Purdue E C N.

.

After logging in, I click the right mouse button and start a Terminal.

The al as command shows the files or directories already there.

If I add minus F after al as, then the directories are shown with ending slashes.

I can use the R M D I R command to remove the directories I will not use.

To remove a file, use the R M command. To remove a directory, use the R M D I R command.

We can also see the files and directories by clicking the Home icon.

The clear command can hide the past commands in the terminal.

I will use the M K D I R command to create a directory called E C E two six four.

Then, I use the C D command to enter this directory.

Let’s create a directory called Lectures and inside this directory another directory 01.

You can use the P W D command to see where you are in the directory. P W D means print working directory.

Let’s create a program called prog01.c. You can use your preferred text editor. I will use e macs for this example.

When you start e macs, it is divided into two parts. The lower part shows some information about e macs. You can select “Never show it again” and the lower part will not appear when you use e macs the next time.

E macs has a good feature called syntax highlighting. Syntax highlighting makes computer programs easier to read.

This is a simple program. It prints ten numbers 0, 1, 2, to 9.

I am going to show you how to execute this program.

C programs have three formats. The first format is the text format. This format is readable by humans, but computers cannot understand this format. The second format is the executable format. The executable format is readable by computer but not readable by humans.

There is also an intermediate format called object format. This is also readable by machines, not humans.

Converting the text format to the object format needs a process called compilation.

Converting the object format to the executable format needs a process called linking.

In this class, we use G C C for these processes. G C C is a compiler and linker.

Compilation can be done by adding minus C after G C C.

.

What are the advantages of these different formats?

By separating the text formats from the executable formats, it is possible to have the same programs written in the text formats to run on different types of machines. Different types of machines may have different hardware or different operating systems. The same programs can run on these machines after the text formats are compiled and linked. Obviously, different types of machines need different compilers and linkers.

After compilers and linkers have been created, many many programs can become available to different types of machines.

There are other advantages of using the three formats. We will talk about the other advantages in a later lecture.

In this example, we use G C C to convert P R O G zero one dot C.

.

By default, the name of the executable is called A. dot out.

As you can see a new file is created by G C C.

.

In Linux, the file command can tell us information about a file.

File P R O G zero one dot C tells us that this is a text file for a C program.

File A. dot OUT tells us this is an executable file for the 64 bit X eighty six architecture and Linux version two dot six dot three two.

Do not worry about the other pieces of information.

To execute this program, type dot slash and then A. dot out.

The reason of adding dot slash is to ensure that you are running the program in the current directory.

It is possible that you may have several files all called A. dot out in different directories.

If you do not add dot slash, it is possible that you run a program in a different directory.

If you do not like the name A. dot out, you can add minus o to specify the name of the executable. Please notice that this is the lower case o.

In this example, the executable is called my P R O G zero one.

By convention, in Linux, executable files have no extension. Do not add dot E. X. E.

.

Here are two commonly made mistakes.

First, many students like to call their executable files “test”. T E S T is a Linux command. When you name your executable T E S T, do you run your own program or the Linux command? If you remember to add dot slash in front, you will run your own program. If you forget to add dot slash, you will run the Linux command. This is too confusing. It is better to avoid this problem simply by not using T E S T as your executable.

Another common mistake is to put the text file after minus o. If you do that, your text file will be erased.