

Lab Tutorial (C Programming)

Assignments that require C programming **MUST** compile in Linux. You are free to do them however you wish as long as they compile in Linux. This tutorial will talk about different methods to compile the code using Windows, Linux, or Mac.

For the C Code:

As was pointed out, C is a prerequisite for this course. However, the assignments are straight forward and do not require a lot of experience in C. If you have taken an intro course to C or C++ as an undergraduate, you should be fine. For a quick refresher, check these 2 websites:

- <https://www.learn-c.org/>
- <https://www.sololearn.com/> (This one is Question/Answer style and has android/ios app!)

The main topics you should know are:

1. Variables and Types
2. Arrays
3. Conditions
4. Strings (mainly `printf` command)
5. For loops
6. While loops
7. Functions
8. Pointers (Last assignment only)
9. Structures (Last assignment only)

Please make sure to include **comments** in your code (Points will be deducted for uncommented code).

For Uploading your Assignment to Canvas:

Most assignments will require you to submit one file. If the assignment states to submit more than one file, do **NOT** zip the files together. Instead add each file separately during submission as shown in the figure below.

File Upload

Text Entry

Google Drive

Upload a file, or choose a file you've already uploaded.

File:

Choose File

No file chosen

←

✕

Source file 1

Choose File

No file chosen

←

✕

Source file 2

+ Add Another File

←

Add more files

Comments...

Cancel

Submit Assignment

Compiling directly on your personal PC (Recommended):

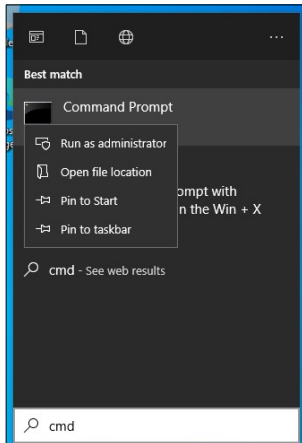
For **Linux and Mac**, you can compile C code directly out of the box in most cases.

For **Windows 10 and 11** follow the directions below to install **Windows Subsystem for Linux**, if it is not installed already.

Note: You must be running Windows 10 version 2004 and higher (Build 19041 and higher) or Windows 11. If you are on Windows 10 and the below instructions do not work, please update Windows to the latest version, or verify that you do not already have WSL installed.

The instructions below were adapted from <https://docs.microsoft.com/en-us/windows/wsl/install>.
If they do not work, please visit the link above for more information.

1. Search 'cmd' then right click "Command Prompt" and choose "Run as administrator"



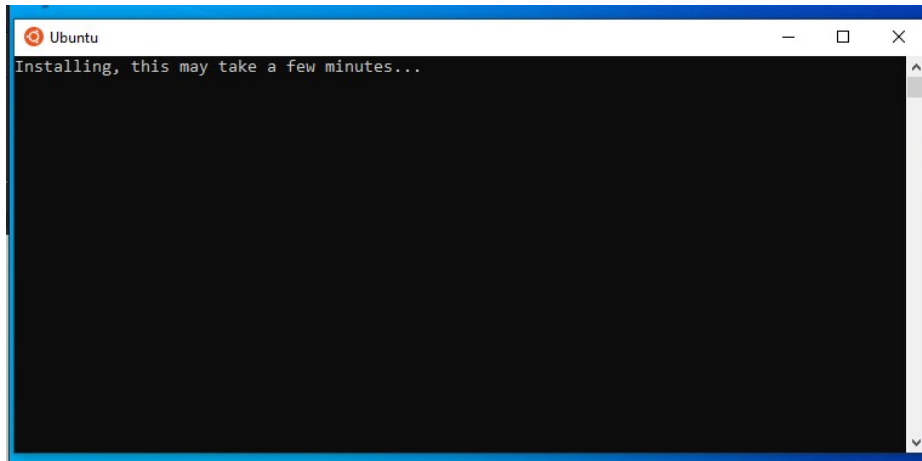
2. After it opens, run the command "wsl --install" then wait for the installation to finish

```
Administrator: Command Prompt - wsl --install
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.

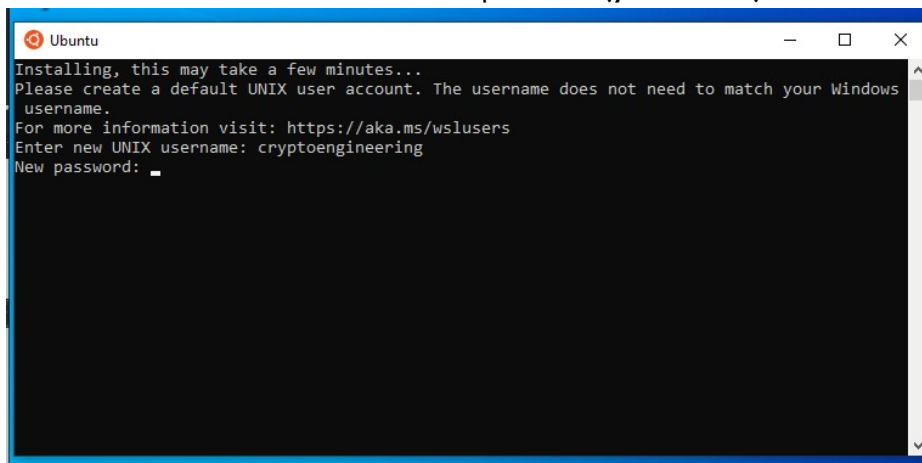
C:\Windows\system32>wsl --install
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
Downloading: WSL Kernel
Installing: WSL Kernel
WSL Kernel has been installed.
Downloading: Ubuntu
[===== 5.8% ]
```

3. Restart your system as instructed

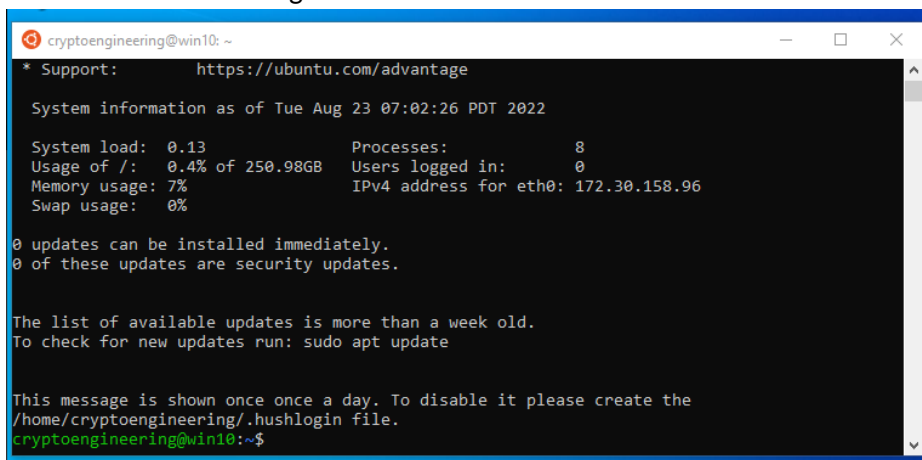
4. Wait for the next installation to finish



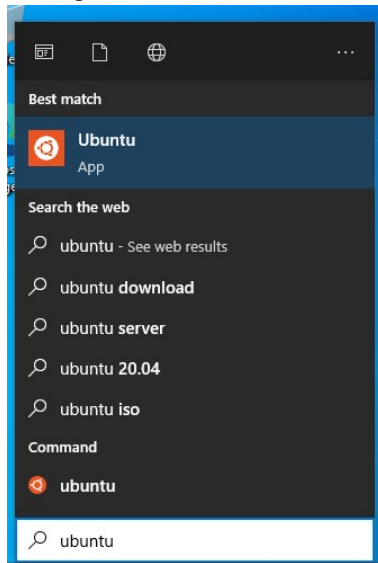
5. Choose a new Ubuntu username and password (your choice)



6. You will then be brought to the Ubuntu shell



7. To get back to the Ubuntu shell later, open the Start menu then search for and run Ubuntu



Note: Your entry for “Ubuntu” may be slightly different than what is shown here, that is ok.

After opening the Linux bash shell, type the following 3 commands (the last one in case you want to use a debugger):

- `sudo apt update`
- `sudo apt install build-essential`
- `sudo apt install gdb`

You are now ready to compile your C code in Linux. Note that your C: drive is in `/mnt/c`. Your documents folder will be in `/mnt/c/user/USERNAME/documents` where USERNAME is your PC username.

For Windows 10, Linux and Mac: To compile your code from command line (Go to **Visual Studio Code and Debugging** section to use a GUI instead):

1. Go to the folder where the C code you want to compile is in (Use `ls` to print all the files and folders in your current folder, `cd folder1` to go into a folder named **folder1**, `cd ..` to go out of the current folder).
2. Type `gcc q1.c` to compile q1.c.
3. Type `./a.out` to run your code.

For students that have older version of Windows, you can get a free student copy of Windows 10 at Microsoft's Azure website. If that is not an option, please let us know.


Visual Studio Code and Debugger:

Visual Studio Code is a free and open source editor by Microsoft for Windows, Linux, and Mac. It is lightweight and provides syntax highlighting and debugging capabilities for C. It is also one of the few options to debug C code for Linux on Windows 10 directly. You are free to use any text editor/IDE but it is highly recommended to use Visual Studio Code.

In **Mac and Linux**, after installing Visual Studio Code,

1. Click on **Extensions tab** (fifth option in the left bar)
2. Search for **C/C++** by Microsoft and install it
3. Go back to the first tab in the left bar. Open a project folder and start coding!

In **Windows 10/11**, after getting the Linux bash shell (WSL) and installing Visual Studio Code

1. Click on **Extensions tab** (fifth option in the left bar)
2. Search for **Remote Development** by Microsoft and install it
3. In the bottom left corner, there is going to be a green button
4. Click on it and select **Remote-WSL: New Window**. A new window will open. The button in the bottom left corner should say WSL similar this:

5. Go to the **Extensions tab** again
6. Search for **C/C++** by Microsoft and install it on WSL.
7. Go back to the first tab in the left bar. Open a project folder and start coding! Note that C: drive is in **/mnt/c**. Your documents folder will be in **/mnt/c/user/USERNAME/documents** where USERNAME is your PC username.

In **Windows 10/11, Mac and Linux**, once done coding,

1. Click on the **debug** tab (third option in the left bar).
2. Click on the green play button OR the dropdown menu and Add Configuration.
3. Choose C++ (GDB/LLDB)
4. Choose gcc build and debug active file
5. A file called launch.json will popup. Leave the default configuration and close it.
6. Click on the green play button from step 2. An error popup will show (if you don't see it skip to the final step)
7. Choose Configure task.
8. Choose gcc build and debug active file
9. A file called task.json will popup. Leave the default configuration and close it.
10. You can now click on the green play button from step 2 and launch the debugger (make sure the C file you want to debug is in the active window.)

You can setup breakpoints by clicking to the left of the line numbers. This tutorial will not discuss how to use the debugger as it is outside of the scope of the course. If you need help using Visual Studio Code or the debugger, **Google** is your friend.