

Lab 1

Deadline: Due in Lab in the week of Jan 22, 2024

Knowing the details of your programming environment is an essential ability. We will go over the installation and setup of the essential tools needed for C programming in this lab.

Requirements

When it comes to C language development, the differences between Windows, Linux, and macOS are particularly relevant. The need for cross-platform compatibility necessitates the development of code that can seamlessly run on different environments.

Tool Setup

In this course, you need to use essential tools for C language development. The main C compiler you have to use is GCC. For automating the build process, you should use the Make utility. As the preferred integrated development environment (IDE), use Visual Studio Code (VS Code).

Getting Started

1. Linux

- First of all, you have to make sure that you have the right version of Ubuntu. In order to do that, you should open a terminal in Ubuntu. Then, type the following command:

```
lsb_release -a
```

It should display Ubuntu 22.04.3 LTS. Make sure that, it is this version.

- Now, make sure that you have the x86-64 architecture by issuing the following command:

```
uname -m
```

It should display x86-64. Note that x86 is different from x86-64, so make sure that you have x86-64 Ubuntu installed.

- Now, we will install gcc, which is the default gcc compiler that we will be using throughout the course. Before we do that, let's update and upgrade your installed packages on your system. You will run the following commands:

```
sudo apt-get update  
sudo apt-get upgrade
```

These will upgrade your packages. *sudo* command in Linux basically performs actions as sudoer, which you can think as an administrator in other operating systems. You do not want to run and develop your code as sudoer as this is a malpractice unless there is a very specific reason to do so. But anything that modifies the system-related software such as your installed compiler, you want to perform it as sudoer. You will only do this if you want to change system related packages in this course.

- Now, let's install gcc and other essential tools for this course. We will install other things such as a debugger in the future, but for now, let's install essential tools by:

```
sudo apt-get install build-essential
```

2. Windows

- First, you have to make sure that you have the right version of Windows. It should be 64-bit Windows 10 version 20H2 or higher.
- Install WSL on windows, Installing WSL is now easier than ever. Search for Windows PowerShell in your Windows search bar, and then select Run as administrator. At the command prompt type:

```
wsl -install -d ubuntu
```

This will install both WSL and Ubuntu! Don't forget to restart your machine before continuing. Once Ubuntu has finished its initial setup you will need to create a username and password (this does not need to match your Windows user credentials).

- Now, search for Ubuntu app, it will open the terminal and follow the commands in Linux section above.
- For more info, <https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-10#1-overview>

3. Mac

- Open Terminal.
- Install Homebrew. Copy the command from here <https://brew.sh/> and paste in Terminal, hit enter. (You need to have an admin account).
- After Homebrew installs, type the following command in the terminal. It will install the gcc.

```
brew install gcc
```

It should install the gcc11.4.0

- To install make type the following command in Terminal.
- ```
brew install make
```
- To use make you need to add a path to .bash\_profile.
- ```
export PATH="/usr/local/opt/make/libexec/gnubin:$PATH"
```
- Note that, the default compiler in macOS is clang. But, you have to use gcc instead. To change default compiler to gcc use the following commands.

```
cd /usr/local/bin/
```

```
ls
```

```
ln -s gcc-11 gcc
```

- Logout and log back in. Open Terminal and type gcc or gcc -v.
- This video will guide you through the installation <https://www.youtube.com/watch?v=0z-fCNNqfEg>

4. Configuring VS code

- Download and install relevant VS code from here <https://code.visualstudio.com/Download>
Note that, VS code is available for Linux, Windows and macOS, install for the relevant operating system.
- Install the following extensions in VS code.
WSL extension by Microsoft for Windows only

C/C++ extension pack by Microsoft

- You can configure VS code from Welcome page. Select gcc as default compiler.
- You can create a folder in home/user/C and select as default workspace

5. Hello World

- Open VS code, create a New File hello.c in the default workspace directory e.g home/user/C

```
#include <stdio.h>

int main(){

    printf("Hello World!");
    return 0;
}
```

- Check the provided Makefile (no extension needed).

```
# Makefile

# Compiler and flags
CC = gcc
CFLAGS = -Wall -std=c11

# Source files
SRCS = hello.c

# Output
TARGET = hello.out

# Default target
all: $(TARGET)

# Rule to build the executable
$(TARGET): $(SRCS)
    $(CC) $(CFLAGS) $(SRCS) -o $(TARGET)

# Rule to run the program
run: $(TARGET)
    ./$(TARGET)

# Rule to clean generated files
clean:
    rm -f $(TARGET)
```

- Open terminal in VS code, make sure you are in the right directory where both hello.c and Makefile are saved.
- Run the following command in VS code Terminal.

```
make
```

It should compile the code without any errors.

```
make run
```

It should display the Hello World!

- Run the following command to delete the out file.

```
make clean
```

6. Restrictions

- You must have 64-bit architecture (if not sure, ask your lab instructor) and Linux based text editors i.e vim or emacs. Otherwise, you will actually have some features affecting grades in future labs.
- The Makefile is provided for the lab. You are not supposed to make any changes in the Makefile. The Makefile is supposed to work with hello.c file so, make sure to save the source code as hello.c
- I suggest you learn some basic Linux command by visiting: <https://www.hostinger.com/tutorials/linux-commands>

7. Grading

- For full marks this week:
(1 point) show your output.

8. Submission

- Nothing to submit for this lab.