

Operating Systems – COC 3071L

SE 5th A – Fall 2025

Objective

The purpose of this assignment is to:

1. Configure **Ubuntu** inside **WSL2 (Windows Subsystem for Linux v2)**.
2. Install and configure **Git** in Ubuntu.
3. Generate and set up **SSH keys** to connect with GitHub.
4. Install the **C development environment** in Ubuntu.
5. Write a **Hello World** program in **C**.

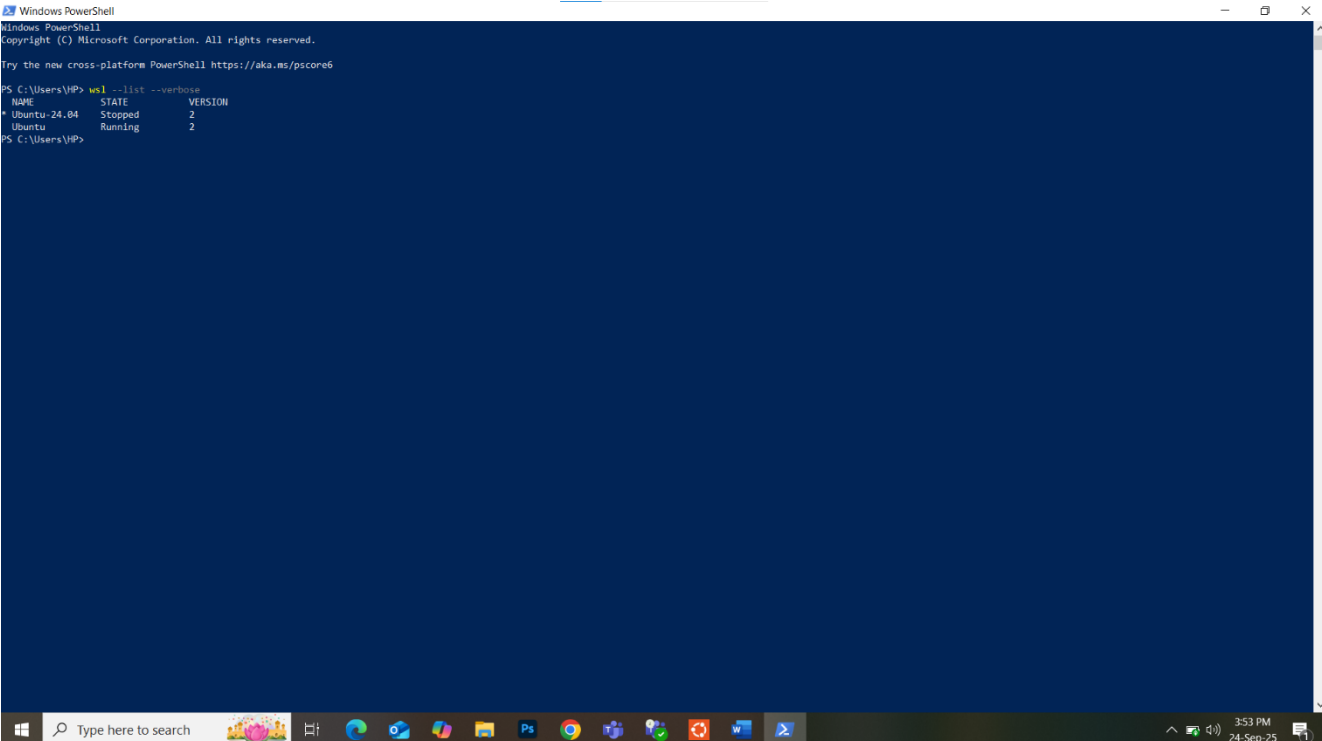
Part A: WSL2 & Ubuntu Setup

1. Verify WSL2 and Ubuntu installation

- Verify installation by running the following command in powershell:

```
wsl --list --verbose
```

Submit a screenshot showing Ubuntu installed and running on WSL2.

A screenshot of a Windows PowerShell terminal window. The title bar reads 'Windows PowerShell'. The terminal text shows the command 'wsl --list --verbose' being executed, which returns a table of installed WSL instances. The table has three columns: NAME, STATE, and VERSION. It lists 'Ubuntu-24.04' as 'Stopped' and 'Ubuntu' as 'Running', both with version '2'. The PowerShell prompt is 'PS C:\Users\HP>'. The Windows taskbar is visible at the bottom with various application icons and a system clock showing 3:53 PM on 24-Sep-25.

NAME	STATE	VERSION
Ubuntu-24.04	Stopped	2
Ubuntu	Running	2

2. Update Ubuntu environment

- Run the following command in Ubuntu:

```
sudo apt update && sudo apt upgrade -y
```

```
fatima1153@DESKTOP-RMTBLS: ~  
Generation complete.  
Setting up libdpkg-perl (1.22.6ubuntu6.5) ...  
Setting up libc-dev-bin (2.39-0ubuntu8.6) ...  
Setting up libc-devtools (2.39-0ubuntu8.6) ...  
Setting up dpkg-dev (1.22.6ubuntu6.5) ...  
Setting up libc6-dev:amd64 (2.39-0ubuntu8.6) ...  
Processing triggers for man-db (2.12.0-4build2) ...  
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...  
fatima1153@DESKTOP-RMTBLS:~$ sudo apt upgrade && sudo apt upgrade -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
The following packages were automatically installed and are no longer required:  
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libllvm17t64 libxcb-dri2-0  
Use 'sudo apt autoremove' to remove them.  
The following packages have been kept back:  
  libgl1-amber-dri libglapi-mesa  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
The following packages were automatically installed and are no longer required:  
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libllvm17t64 libxcb-dri2-0  
Use 'sudo apt autoremove' to remove them.  
The following packages have been kept back:  
  libgl1-amber-dri libglapi-mesa  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
fatima1153@DESKTOP-RMTBLS:~$
```

Part B: Git & GitHub SSH Setup

1. Configure Git

- Set your name and email:

```
git config --global user.name "Your Name"  
git config --global user.email "your@email.com"
```

- Show your config:

```
git config --list
```

```
fatima1153@DESKTOP-RMTBLS: ~  
Setting up libc-dev-bin (2.39-0ubuntu8.6) ...  
Setting up libc-devtools (2.39-0ubuntu8.6) ...  
Setting up dpkg-dev (1.22.6ubuntu6.5) ...  
Setting up lib66-dev:amd64 (2.39-0ubuntu8.6) ...  
Processing triggers for man-db (2.12.0-4build2) ...  
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...  
fatima1153@DESKTOP-RMTBLS:~$ sudo apt upgrade && sudo apt upgrade -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
The following packages were automatically installed and are no longer required:  
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libllvm17t64 libxcb-dri2-0  
Use 'sudo apt autoremove' to remove them.  
The following packages have been kept back:  
  libgl1-amber-dri libglapi-mesa  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
The following packages were automatically installed and are no longer required:  
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libllvm17t64 libxcb-dri2-0  
Use 'sudo apt autoremove' to remove them.  
The following packages have been kept back:  
  libgl1-amber-dri libglapi-mesa  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
fatima1153@DESKTOP-RMTBLS:~$ git config --global user.name "FatimaBashir"  
fatima1153@DESKTOP-RMTBLS:~$ git config --global user.email "fatihere75@gmail.com"  
fatima1153@DESKTOP-RMTBLS:~$ git config --list  
user.name=FatimaBashir  
user.email=fatihere75@gmail.com  
fatima1153@DESKTOP-RMTBLS:~$
```

Submit a screenshot.

2. Generate SSH Keys

- Run:

```
ssh-keygen -t ed25519
```

- ◆ Copy the public key:

```
cat ~/.ssh/id_ed25519.pub
```

- Add this key to your GitHub account under **Settings** → **SSH and GPG keys**.

```
fatima1153@DESKTOP-RMTBLBS:~$ cd ~/.ssh/
-bash: cd: ~/.ssh/: No such file or directory
fatima1153@DESKTOP-RMTBLBS:~$ cd ~/.ssh
-bash: cd: /home/fatima1153/.ssh: No such file or directory
fatima1153@DESKTOP-RMTBLBS:~$ gir config --list
Command 'gir' not found, but can be installed with:
sudo apt install gir-rust-code-generator
fatima1153@DESKTOP-RMTBLBS:~$ ssh-keygen -t ed25519 -C "fatihere75@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/fatima1153/.ssh/id_ed25519):
Created directory '/home/fatima1153/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/fatima1153/.ssh/id_ed25519
Your public key has been saved in /home/fatima1153/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:Av41Xed615RbH3KiwQ0wgmMrzgv7KURtXxRuUI fatihere75@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|...E|
|.. ...|
|+ o + o |
|o= = =|
| =B o .s|
|*B.O.o|
|B*+ ..|
|* + .|
|== ...|
+----[SHA256]-----+
fatima1153@DESKTOP-RMTBLBS:~$
```

3. Test Connection

```
ssh -T git@github.com
```

- ◆ Submit a screenshot showing successful authentication.

```
fatima1153@DESKTOP-RMTBL8S: ~/ssh
fatima1153@DESKTOP-RMTBL8S:~$ gir config --list
Command 'gir' not found, but can be installed with:
sudo apt install gir-rust-code-generator
fatima1153@DESKTOP-RMTBL8S:~$ ssh-keygen -t ed25519 -C "fatihere75@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/fatima1153/.ssh/id_ed25519):
Created directory '/home/fatima1153/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/fatima1153/.ssh/id_ed25519
Your public key has been saved in /home/fatima1153/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:Av41Xg6x415RBWh3Kiw6Qr9wGm+rzgV7KU-tXXFrUII fatihere75@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|    . E          |
|..  . .          |
|x O+  O .        |
|d=.  .  O        |
|==B O . . $      |
|*+B O . . O      |
|.B+* . .         |
|* . .           |
|=. .           |
|=. .           |
+---[SHA256]-----+
fatima1153@DESKTOP-RMTBL8S:~$ cd ~/.ssh/
fatima1153@DESKTOP-RMTBL8S:~/.ssh$ ls
id_ed25519  id_ed25519.pub
fatima1153@DESKTOP-RMTBL8S:~/.ssh$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIP380/2lUCeR/OM4in4OvwQRRH/haDL0Fi8DMp38RyIf fatihere75@gmail.com
fatima1153@DESKTOP-RMTBL8S:~/.ssh$ ssh -T git@github.com
The authenticity of host 'github.com (20.207.73.82)' can't be established.
ED25519 key fingerprint is SHA256:d0Y9wvW6TujJhbpZisF/zLDA0zPMSVHdkr4UVC0QU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi FatimaBashir511! You've successfully authenticated, but GitHub does not provide shell access.
fatima1153@DESKTOP-RMTBL8S:~/.ssh$
```

Part C: C Programming Environment & Practice

Step 1: Install Build Tools

Before writing C programs, install the **build-essential** package which contains `gcc` , `g++` , and other tools required for compiling.

Run:

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

Verify installation by checking the version of `gcc` :

```
gcc --version
```

- Submit a screenshot of successful installation and version output.

```
fatima1153@DESKTOP-RMTBL8S: ~  
fatima1153@DESKTOP-RMTBL8S:~$ sudo apt install build-essential  
[sudo] password for fatima1153:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
build-essential is already the newest version (12.10ubuntu1).  
The following packages were automatically installed and are no longer required:  
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libllvm17t64 libxcb-dri2-0  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
fatima1153@DESKTOP-RMTBL8S:~$ gcc --version  
gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0  
Copyright (C) 2023 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.  
fatima1153@DESKTOP-RMTBL8S:~$
```

Step 2: How to run a C Program

1. First write a C program in a file with `.c` extension.
2. Compile the file using `gcc filename.c -o filename.out`
3. Execute it using `./filename.out`

```
fatima1153@DESKTOP-RMTBL8S: ~  
GNU nano 7.2  
#include <stdio.h>  
  
int main(){  
    printf("Hello, Fatima!\n");  
    return 0;  
}
```

Breakdown

- ♦ `gcc`
 - ♦ This is the GNU Compiler Collection command.

- ♦ It compiles C (and other languages like C++) programs into machine code that can be executed by the computer.
- ♦ filename.c
 - ♦ This is the source code file you wrote in C.
 - ♦ Example: hello.c contains your C program.
- ♦ -o filename.out
 - ♦ The option -o means “output.”
 - ♦ By default, gcc creates an executable file named a.out if you don't specify anything.
 - ♦ With -o, you can choose the name of the output executable.
 - ♦ In this case, the compiled file will be named filename.out.

Step 3: Write a C Program

Write a simple C program of your choice. It can be a **Hello World** program or any other.

Submission: For the program, submit:

- ♦ The C source code (.c file).
- ♦ Screenshot of execution (./program)

```
fatima1153@DESKTOP-RMTBLS: ~
fatima1153@DESKTOP-RMTBLS:~$ sudo apt install build-essential
[sudo] password for fatima1153:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.10ubuntu1).
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libllvm17t64 libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
fatima1153@DESKTOP-RMTBLS:~$ gcc --version
gcc (Ubuntu 13.3.0-6ubuntu2-24.04) 13.3.0
Copyright (C) 2023 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.

fatima1153@DESKTOP-RMTBLS:~$ nano hello.c
fatima1153@DESKTOP-RMTBLS:~$ gcc hello.c -o hello
hello.c:1:10: fatal error: studio: No such file or directory
   1 | #include <studio>
     |          ^~~~~~
compilation terminated.
fatima1153@DESKTOP-RMTBLS:~$ nano hello.c
fatima1153@DESKTOP-RMTBLS:~$ gcc hello.c -o hello
fatima1153@DESKTOP-RMTBLS:~$ ./hello
Hello, Fatima!
fatima1153@DESKTOP-RMTBLS:~$
```

Deadline

- ♦ Submit before **12:00 AM, 25 September, 2025.**