CS2100 Computer Organization Lab 0:
(Week 3)

This introductory lab is to get acquainted with your Lab TA and set up your environment for future labs. Everyone attending will receive full marks.

To prepare, you'll need:

1. **A C compiler** (GCC)
2. **A debugger tool** (GDB/LLDB)

Instructions in Lab 1 assume you're using GCC and GDB. If you're not already using Ubuntu on your MacBook or Windows laptop, you'll need to set up the Ubuntu environment first.

1A) Setup For MacBook Users

MacBooks with Apple CPUs no longer support gdb (so sad). But we can fix it by setting up a virtual Ubuntu environment using a tool called **lima**.

Step 1: Set up lima.

Open a terminal in your MacBook, then type the following at the command prompt:

```
$ brew install lima
```

After lima is installed successfully, start it up:

```
$ limactl start
```

Step 2: Give lima permission to access your files.

```
$ open ~/.lima/default/lima.yaml
```

This will open up the MacOS text editor. Using Command-F (coz the file is big and cryptic), look for the line `- location: "~"`. There are other location lines so you must find this particular one. A few lines down, you will see `writable: null`. Change it to `writable: true`. After you are done, save the file and exit the editor.

Step 3: Restart the lima daemon.

```
$ limactl stop default
$ limactl start default
```

Step 4: Enter Ubuntu environment via lima

Simple. Type at the terminal:

```
$ lima
```

You should see a change in the prompt indicating that you are in Ubuntu. You can confirm this by typing the following at the prompt:

```
$ uname -a
```

You should see "ubuntu" somewhere.

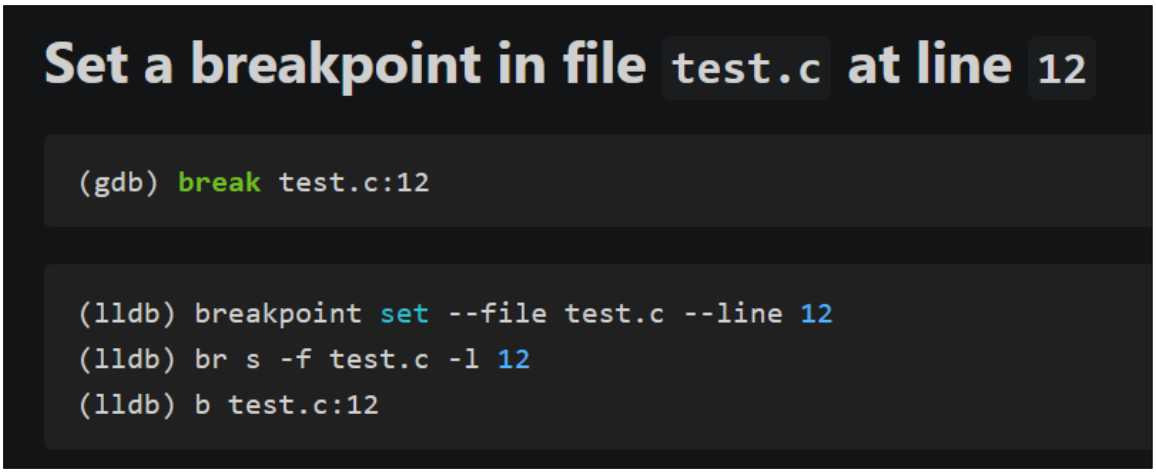
1A) An alternative for MacOS users - LLDB

For MacOS on Apple Silicon (M1, M2, etc) devices you can also use LLDB instead, the default debugger on Mac.

Both GDB and LLDB are great debugger tools, and the majority of GDB and LLDB commands are the same, just with different names. The lab sheet contains several GDB commands, you will need to find the equivalent command on LLDB.

To do so, refer to this map from GDB commands to LLDB commands:
https://lldb.llvm.org/use/map.html#breakpoint-commands

Eg. Setting breakpoint at a specific line:



The above is a screenshot from the GDB-LLDB command map (above link). Here, instead of the gdb command (**break test.c:12**), we use instead the equivalent lldb command (**b test.c:12)**.

If you are more interested in how LLDB works, you can find an LLDB tutorial here:
https://lldb.llvm.org/use/tutorial.html

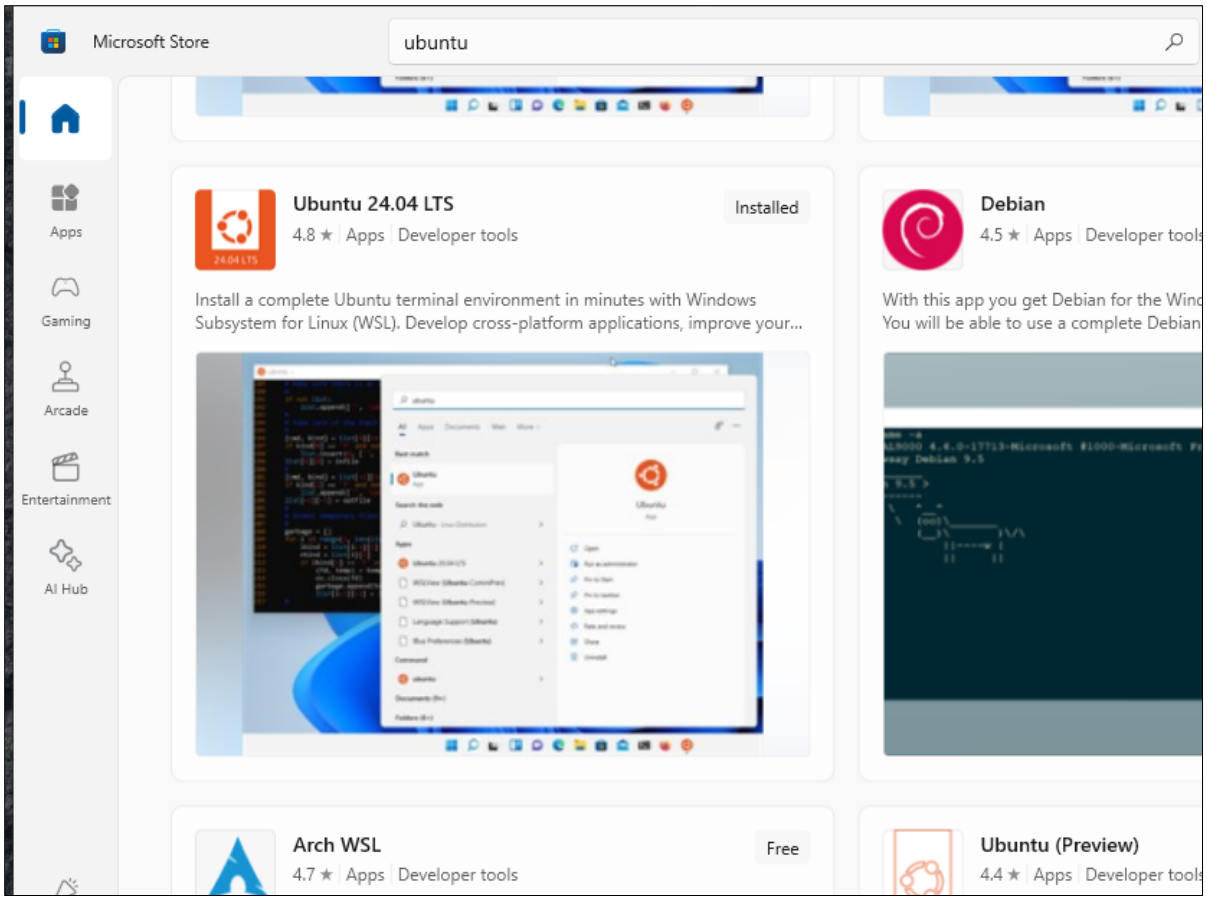
1B) Setup For Windows users

Step 1: Install the Windows Subsystem for Linux.

Follow Microsoft's own instructions found here:

https://learn.microsoft.com/en-us/windows/wsl/install

Step 2: Install Ubuntu

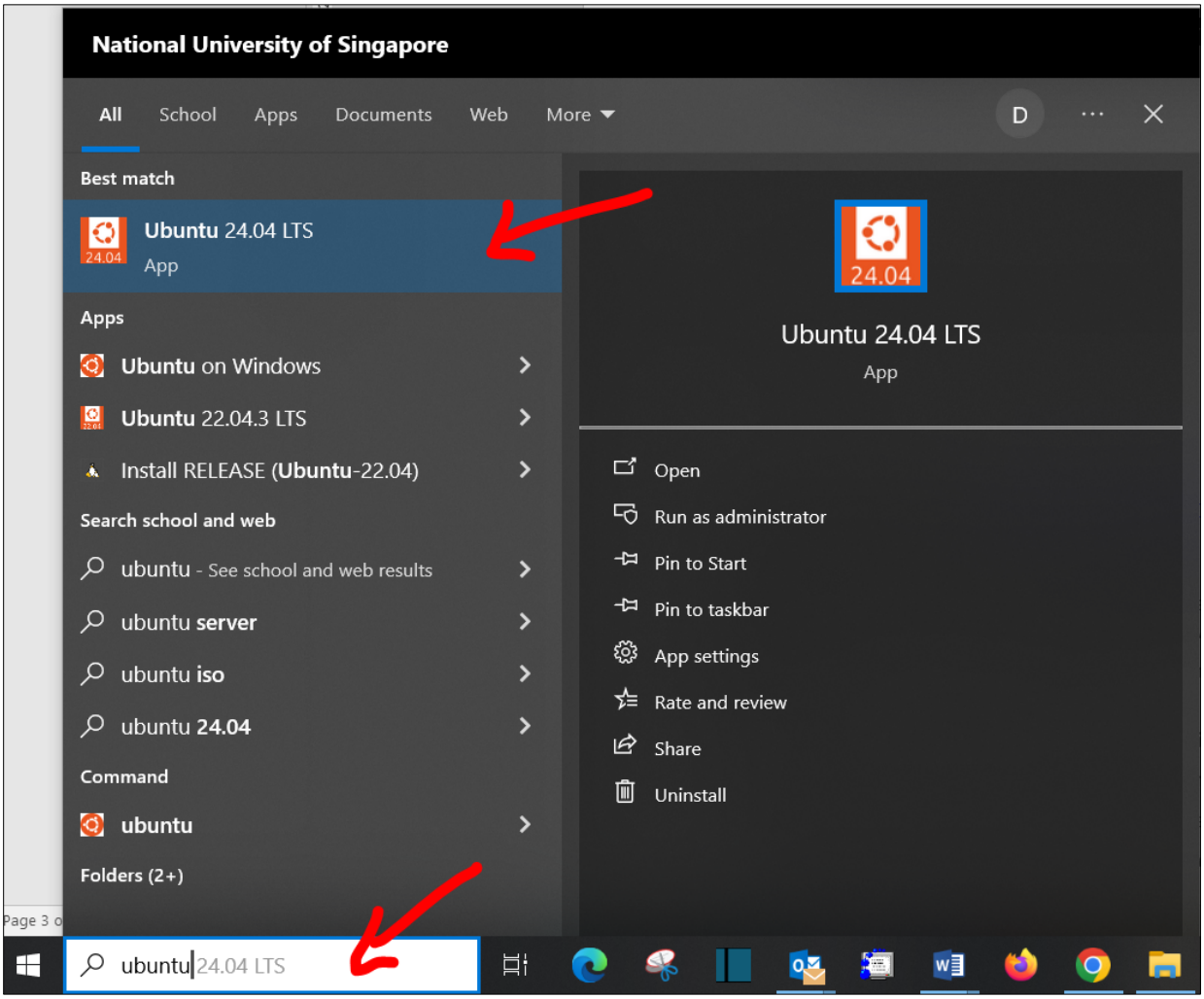


Head over to the **Microsoft Store** on your laptop and search for "Ubuntu." Install **"Ubuntu 24.04 LTS"** (other versions will work for our labs, but you might as well grab the latest).

Since I've already installed it on my machine, it shows "Installed." On yours, it should say "Free". Just click "Free" to start the installation.

Step 3: Fire up Ubuntu

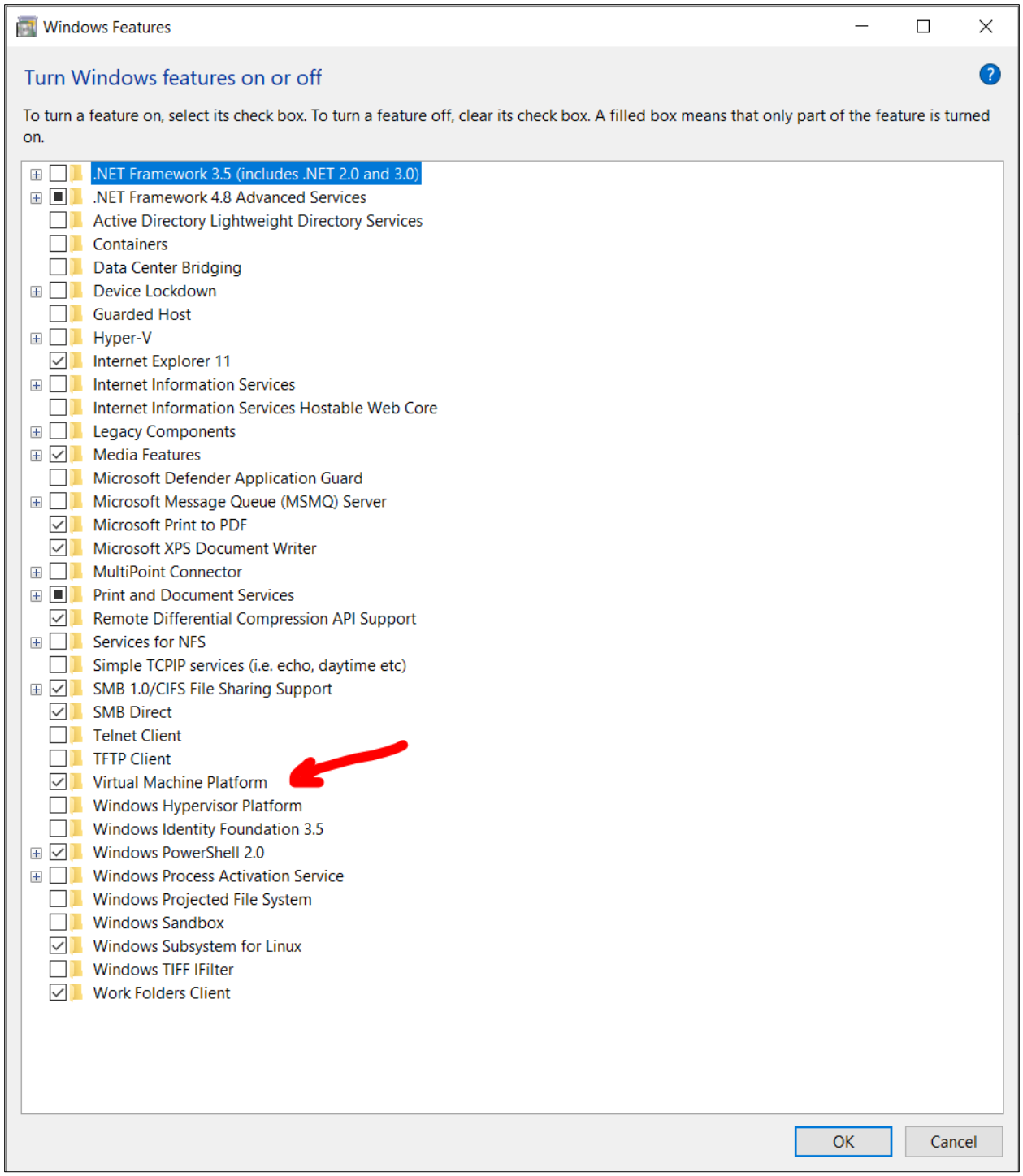
This should be straightforward:



An Ubuntu terminal window should open up.

If you encounter any issues, try the following steps:

- In *"Turn Windows features on or off"*, ensure that *"Virtual Machine Platform"* is turned **on** and *"Windows Hypervisor Platform"* is turned **off**.
- Reboot your system and check the BIOS settings to confirm that "*virtual machine support"* is enabled (it's usually on by default, but it's worth checking).


Windows Features
Turn Windows features on or off
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.
.NET Framework 3.5 (includes .NET 2.0 and 3.0)
.NET Framework 4.8 Advanced Services
Active Directory Lightweight Directory Services
Containers
Data Center Bridging
Device Lockdown
Guarded Host
Hyper-V
Internet Explorer 11
Internet Information Services
Internet Information Services Hostable Web Core
Legacy Components
Media Features
Microsoft Defender Application Guard
Microsoft Message Queue (MSMQ) Server
Microsoft Print to PDF
Microsoft XPS Document Writer
MultiPoint Connector
Print and Document Services
Remote Differential Compression API Support
Services for NFS
Simple TCPIP services (i.e. echo, daytime etc)
SMB 1.0/CIFS File Sharing Support
SMB Direct
Telnet Client
TFTP Client
Virtual Machine Platform
Windows Hypervisor Platform
Windows Identity Foundation 3.5
Windows PowerShell 2.0
Windows Process Activation Service
Windows Projected File System
Windows Sandbox
Windows Subsystem for Linux
Windows TIFF IFilter
Work Folders Client
OK
Cancel

2) Installing GCC and GDB

Once we made it this far, we should have a Ubuntu window opened. In the Ubuntu terminal, type

```
$ sudo apt install gcc gdb
```

Indicate "yes" to its requests by hitting the return (since "yes" is the default.)

If you get any error while executing the above, you may try the following first and then try to install the gcc and gdb:

```
$ sudo apt-get update
```

3) Learning vi

And we're all set! Next up is using the Linux editor, **vi**, to write your program. **vi** is a powerful editor, but it can take some getting used to. It has two main modes:

- **Command mode:** What you type is interpreted as a command.
- **Insert mode:** What you type is inserted as text.

Not sure which mode you're in? Since there are only two modes, you'll toggle between them. When in doubt, just press **ESC** (you can press it multiple times, though once is enough) to return to command mode. To switch to insert mode from command mode, press **"i"** (lowercase, it's case-sensitive). To go back to command mode, just hit **ESC** again.

To begin, type the following at the Ubuntu prompt:

```
$ vi myfirst-prog.c
```

Step 1: Enter insert mode by pressing "i". Then type in your first program:

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    printf("Hello World!\n");
}
```

Step 2: Enter command mode by pressing ESC.

Step 3: Save the file and exit by typing ":wq" in command mode. Done! That's it for the lab.

There are many guides for learning more about the vi editor and its shortcuts. Here is one:
https://www.redhat.com/sysadmin/introduction-vi-editor