Lab 02: Environment Setup Report

**Introduction:**

This lab is for the setup of a development environment for C code and assembly code development. In this lab, we download a virtualization software called VirtualBox and add two operating systems, Ubuntu 22.04 and Windows 11 Developer Evaluation image.

**Procedure:**

The Host OS for this lab and rest of the labs will be Windows 11 running on 16 core Intel i7-13700HX with 16GB of RAM and 1TB SSD storage. The host PC will be sufficient to run the virtual machines with no lag.

1. We download and install Oracle VirtualBox software which will handle all the virtual machines. Then we download and add Ubuntu and Windows 11 virtual machine images to VirtualBox.

A screenshot of a computer

Description automatically generated

1. Once we verify that the virtual machines are up and running, we install the required tools for each machine for building and debugging C and assembly code.]
2. For Ubuntu, we install gcc and gdb for building and debugging C and assembly code.

A screenshot of a computer

Description automatically generated

1. For Windows 11 Developer Evaluation image, we have Visual Studio 2022 pre installed in it. We download and install Windows SDK in it.

A screenshot of a computer

Description automatically generated

1. In order to build and debug, we have to set the configuration from the configuration manager to support Win32 as shown here.

A screenshot of a computer program

Description automatically generated

Then we’ll have options to switch between 32 bit and 64 bit while building and debugging C code.

1. We can build and debug C code in 32 bit application as shown below

A screen shot of a computer

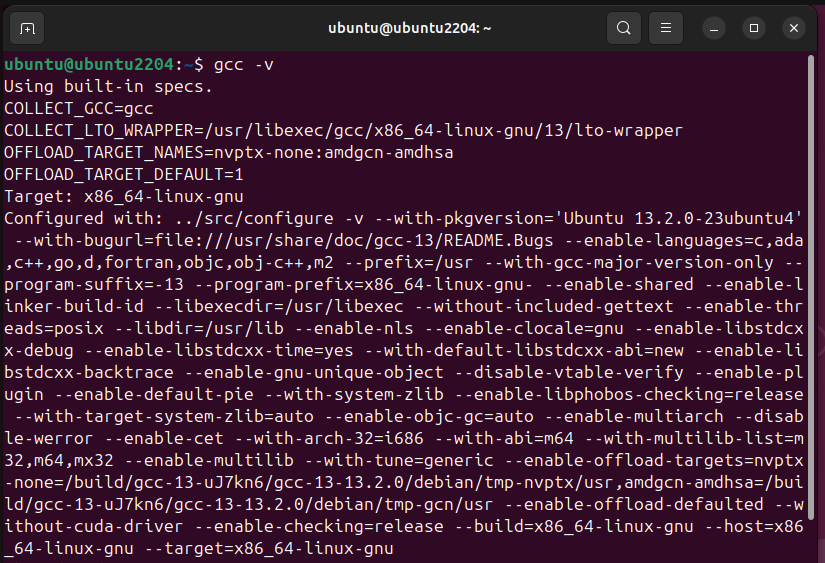
Description automatically generated

1. We can build and debug 64-bit C application as shown below. The only thing to change here is to switch to 64 bits.

A screen shot of a computer

Description automatically generated

1. For Ubuntu, we use gcc to build and run C application and gdb to debug them. By default, gcc compiles C applications in 64 bit configuration as we can see below:



1. To build a C application in 32 bit configuration, we need to have the necessary libraries in gcc. We can do that by installing them with sudo apt-get install gcc-multilib.

A screenshot of a computer program

Description automatically generated

1. For testing, we run a sample file called sample.c in gcc. We can do that by typing gcc -m32 sample.c -o output. This will build and compile the sample.c file and create an executable called output. We can run the executable to find the output as shown below.

A screenshot of a computer screen

Description automatically generated

1. To build and run the file in 64 bit configuration, we can change the mode to 64 bit by using the command gcc -m64 sample.c -o output. The output executable will be a 64 bit application.

A screenshot of a computer program

Description automatically generated

1. We can use gdb to debug the code by typing gdb ./output.

A screenshot of a computer screen

Description automatically generated

**Conclusion:**

We have setup Windows and Linux platforms to build and debug native C applications with 32-bit and 64-bit support. This will help us to complete labs for the rest of the sessions.