Lab 1: Virtual Machine Setup, Introduction to C programming, Github and Linux

Student Name: Herbert Vernon anak Verghese

Github Repository for Lab 1: https://github.com/itsmeherbert/Lab-1---VM-Setup.git

Matric No: 2022659828

Group: CS2554C

# Linux Virtual Machine Setup

Virtual machine is an emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer. Ubuntu is a free and open-source Linux distribution based on Debian. Ubuntu is officially released in three editions: Desktop, Server, and Core. All the editions can run on the computer alone, or in a virtual machine. You will be using Ubuntu server for this lab.

## Installing a Hypervisor – Virtual Box or VMware Workstation Player

Oracle VirtualBox is a hypervisor which allows you to emulate an operating system on your own PC and use it like it's running on real hardware. The emulated host running on hypervisor is called as virtual machine.

VirtualBox is Free but Proprietary software by Oracle Corporation; you can download it from here <https://www.virtualbox.org/wiki/Downloads>

After finish downloading, install and run the hypervisor on your machine.

## Install Ubuntu Virtual Machine

You can install Ubuntu 18.04.3 LTS or later from either bootable image (.iso) or virtual machine image (.ova). You can download both types from here https://ubuntu.com/download/server

If you install via bootable image (.iso), create new virtual machine in your hypervisor and boot the image. Follow the installation instruction.

If you install via VM image (.ova), you can import it into the hypervisor, it may takes a while. When done, run the installed Ubuntu virtual machine.

## Make sure VM has internet connectivity

Please make sure you configure your virtual machine network adapter as NAT (network address translation). You can test internet connectivity by opening a terminal on Ubuntu and try to ping google or cloud flare DNS. **ping 8.8.8.8** / **ping 1.1.1.1**

## Run and Explore Ubuntu Linux

After finish with installation, it’s time to explore Ubuntu and software / tools that came with it, please answer this question.

Please provide the output and screenshot when execute “**uname –a**” in terminal. Please explain the output.

Output:

Explaination: To show the version of Ubuntu.

## Access your Ubuntu Linux remotely via SSH protocol

Network administrator usually access their linux sever via SSH protocol. Before you can access your linux via SSH, you need to install SSH server on your Ubuntu virtual machine. You can follow this tutorial [https://linuxize.com/post/how-to-enable-ssh-on-](https://linuxize.com/post/how-to-enable-ssh-on-ubuntu-18-04/) [ubuntu-18-04/](https://linuxize.com/post/how-to-enable-ssh-on-ubuntu-18-04/)

After you have enabled SSH on your Ubuntu machine, you need to install SSH client on your window machine. You can download ssh client from here <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Once you have installed ssh client on your window machine, you can ssh to your Ubuntu using putty by using your Ubuntu IP address and username. You can check your Ubuntu machine IP address by using this command on Ubuntu terminal **#ifconfig.**

Please provide screenshot that you able to access Ubuntu machine on Putty using SSH protocol.

# Introduction to Version Control and GIT

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows. You will be using git for Lab work and project through this course

## Installing GIT and exploring GIT

You can use this tutorial to install git in Linux <https://www.atlassian.com/git/tutorials/install-git>

This is a complete tutorial in using GIT <https://www.atlassian.com/git/tutorials>

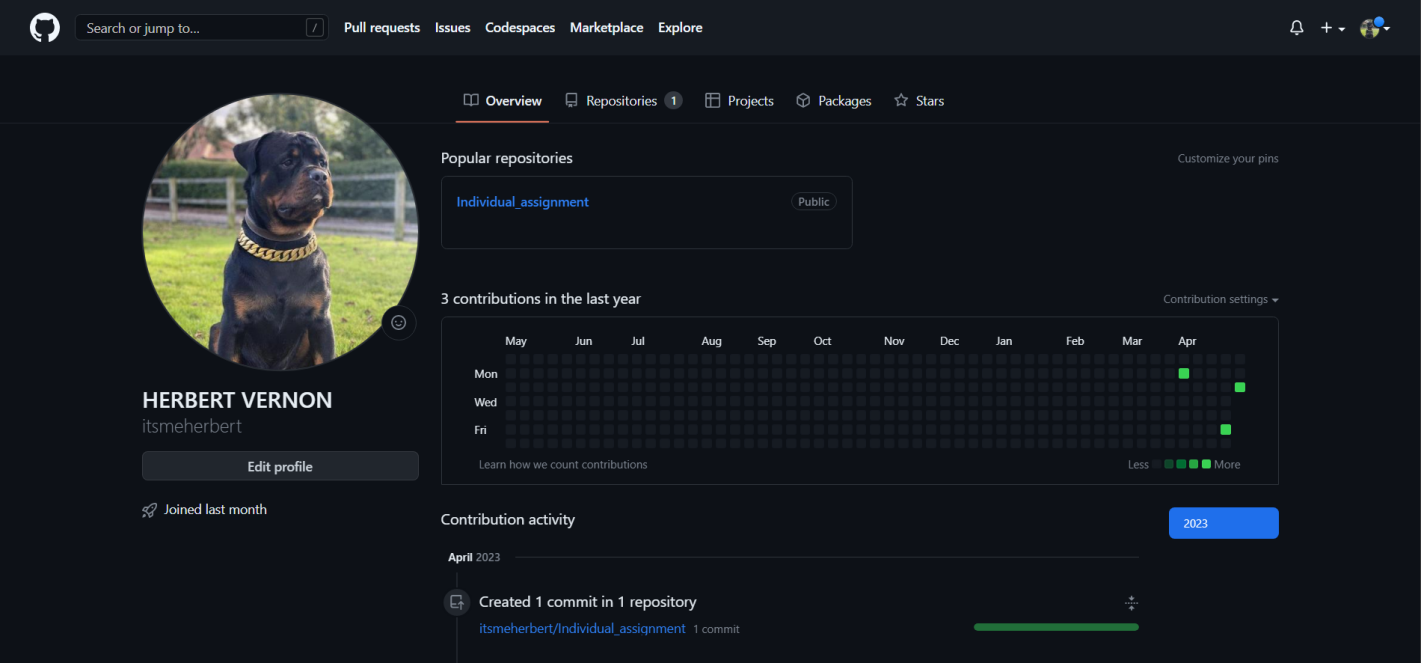
Please provide screenshot that you have install GIT in your Linux



## Signing up your Github Account

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. You'll create your own Hello World repository and learn GitHub's Pull Request workflow, a popular way to create and review code.

Please sign-up a git account (<https://github.com/>) and provide screenshot of your github profile.



# Introduction to C and Linux

For the lab you are required to have a distribution of Linux in order for you to compile the program as well as checked whether your program have run correctly. Please create a github repository for this lab exercise and push your code to your github repository.

After this lab, students should be able to:

* Create, Edit, Save a file using Linux
* Compile a C program
* Check for open port and running program

## Hello World Program in C

* + - 1. Open up a terminal.
      2. Create a new file by using the following command:

touch helloworld.c

* + - 1. Edit the file by using nano text editor. nano helloworld.c
      2. Type in the following program segment:

#include <stdio.h> int main(void) {

/\* This is my first program in C \*/

printf(“Hello World!”); printf(“I Love C”); return (0);

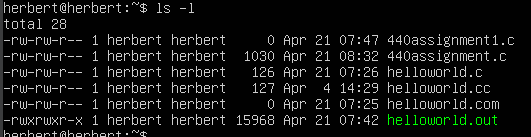
}

* + - 1. Save the program by pressing ctrl O. When prompted for file name, press ENTER.
      2. Exit from nano text editor by pressing ctrl X
      3. The file should have been saved inside your home directory, to verify run the following command:

ls –l

You should see the file listed along with other files inside the directory.

Please provide screenshot of file inside the directory.

g

## Compiling a C Program

In linux, the popular compiler are known as GNU C Compiler or gcc in short. For this course, we are going

to use gcc to compile our C program.

* + - 1. Open up a terminal.
      2. Go to your home directory or directory where you have saved your helloworld.c. To go to your

home directory type *cd* or *cd /home/your\_username.*

* + - 1. Compile the program by using the following command:

gcc helloworld.c

* + - 1. If you get an error, edit the program again using nano and recompile. If you get no error ( after

running the above command, you are returned to the prompt) then congratulations you have

succesfully compile your first C program.

* + - 1. A succesful compilation will create an object file called a.out. Check whether the file exist by

running:

ls –l

You should see a file called a.out

* + - 1. To see the output of the program, you can run the file by running the following command:

./a.out

Please provide screenshot of the program output



## Assigning Names to the output file

One problem with having the output file name as a.out is that once we compile a new program, we are

unable to run our previously compiled program without recompiling the program. In order to overcome

this program, we could give a name to the output file during compilation.

* + - 1. Open up a terminal
      2. Go to your home directory or directory where you have saved your helloworld.c. To go to your

home directory type *cd* or *cd /home/your\_username.*

* + - 1. Compile the program by using the following command:

gcc –o helloworld.out helloworld.c

the option –o let us specify the name of the output file.

* + - 1. A succesful compilation will create an object file called helloworld.out. Check whether the file

exist by running:

ls –l

You should see a file called helloworld.out instead of a.out.

* + - 1. To see the output of the program, you can run the file by running the following command:

./helloword.out

Please provide screenshot of the program output.



## Getting a variable from user

* + - 1. Open up a terminal.
      2. Create a file called userinput.c by using nano text editor. nano userinput.c
      3. Type in the following program segment:

#include <stdio.h> int main(void) {

/\* This is my second program in C \*/ int age;

printf(“Hi, how old are you? > ”); scanf(“%d”, &age);

printf(“You are %d years old”, age);

return (0);

}

* + - 1. Save the program by pressing ctrl O. When prompted for file name, press ENTER.
      2. Exit from nano text editor by pressing ctrl X
      3. The file should have been saved inside your home directory, to verify run the following command:

ls –l

You should see the file listed along with other files inside the directory.

* + - 1. Compile the program by using the following command: gcc –o userinpit.out userinput.c
      2. If you get an error, edit the program again using nano and recompile.

Please provide screenshot of the program output.



## Checking for open port

There are times when we are required to check list of port open at our PC, to do this we can utilize the

*netstat* command:

* + - 1. Open up a terminal.
      2. Run the following command:

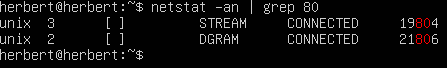
netstat –an | grep <port no.>

For example if you are interested in checking for port 80, you would run the command as: netstat –an | grep 80

* + - 1. If there is a result, it shows you that there are currently port 80 opened in your computer or

you are accessing port 80 at the remote location.

Please provide screenshot of the netstat output.



## Checking for a running program

At times, we wanted to know if a program is running (for example a child program) then we can use the

*ps* command to check.

* + - 1. Open up a terminal
      2. Run the following command:

ps waux | grep <command>

replace command with the name of the program that you are interested in.

* + - 1. If you get a result, it shows that the program is still running

Please provide screenshot of the ps output

