

Warm-Up Project

COP5614 – Operating System

1. Summary

This project is giving you a taste about how to do the simple multi-threads programming. You need to do it in user space using a widely-used threads programming interface, POSIX Threads (Pthreads). You should implement this in Linux, which supports Pthreads as part of the GNU C library.

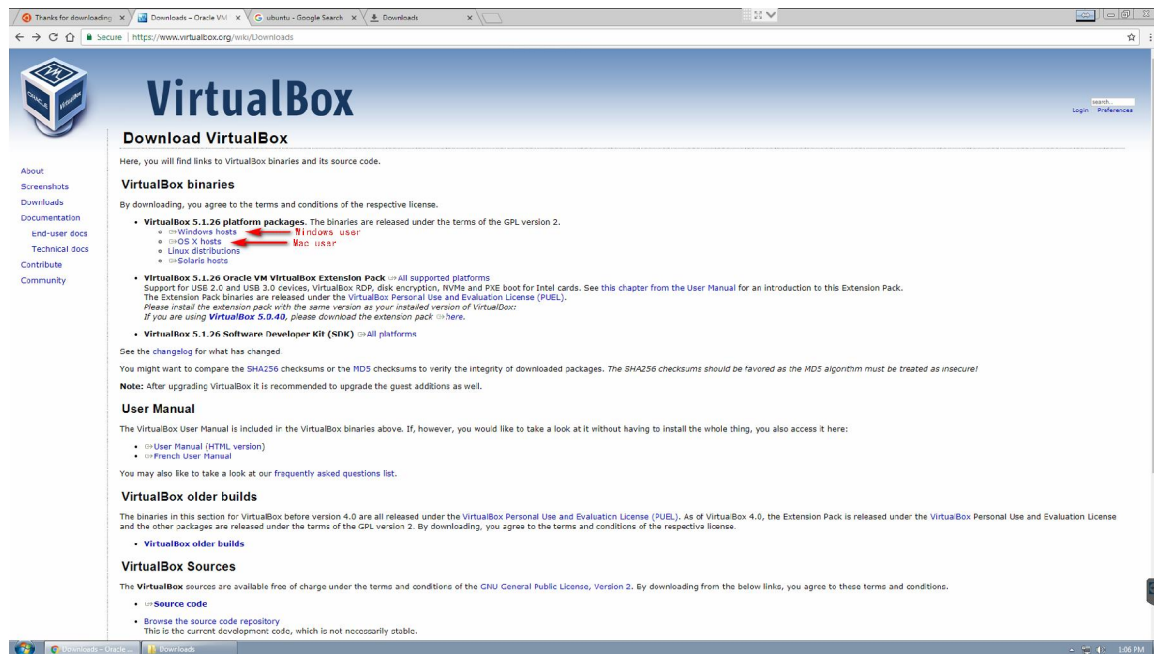
2. Environment Set Up

It's recommendable to use Ubuntu as operating system to accomplish this project. Ubuntu is an open source operating system software for computers. It is one of the distribution systems of Linux, and is based on the Debian architecture. For those who don't have Ubuntu installed in their computers, it's doable to use virtual machine and install the virtual Ubuntu in your Local operating system.

For virtual machine, it's recommendable to use VirtualBox:

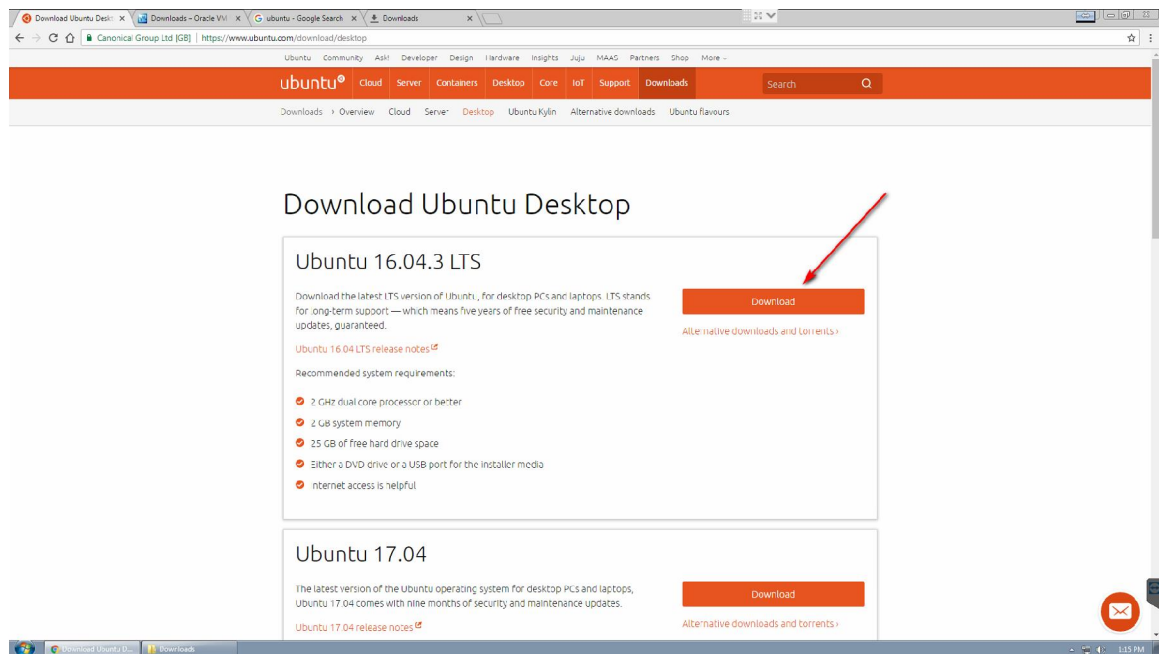
<https://www.virtualbox.org/wiki/Downloads>

If you have Windows operating system installed with your computer, try to download "Windows hosts"; if you use Mac, try to download "OS X hosts".



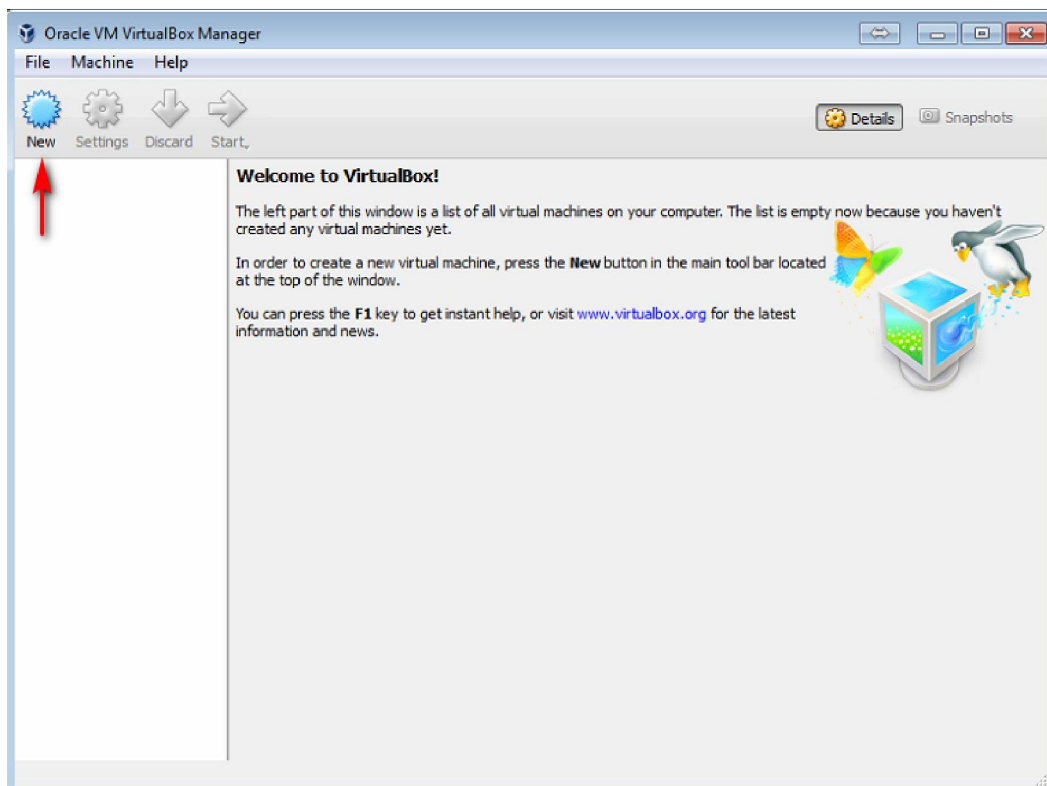
After you download and install VirtualBox in your computer, next step is to download the .iso file of Ubuntu from the official website:

<https://www.ubuntu.com/download/desktop>

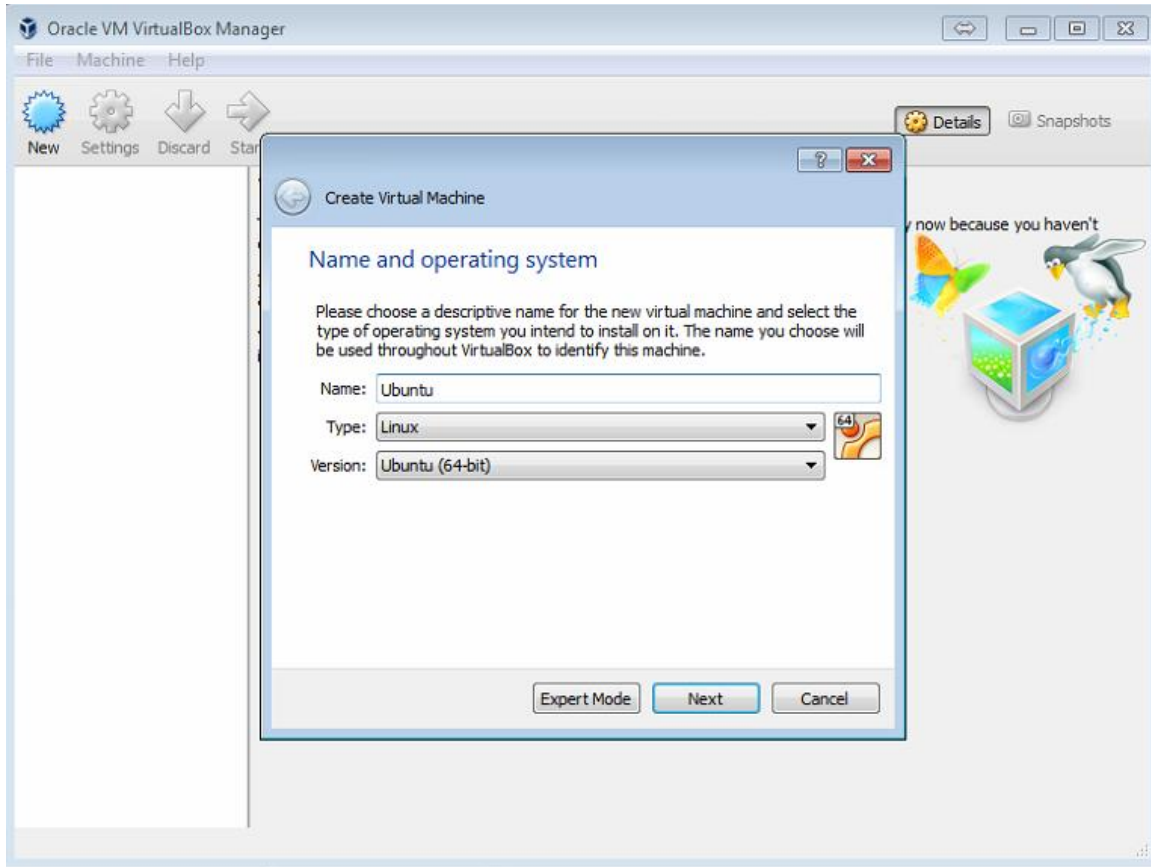


When you finish downloading the .iso file of Ubuntu, then it's time to install a virtual Ubuntu in your virtual machine.

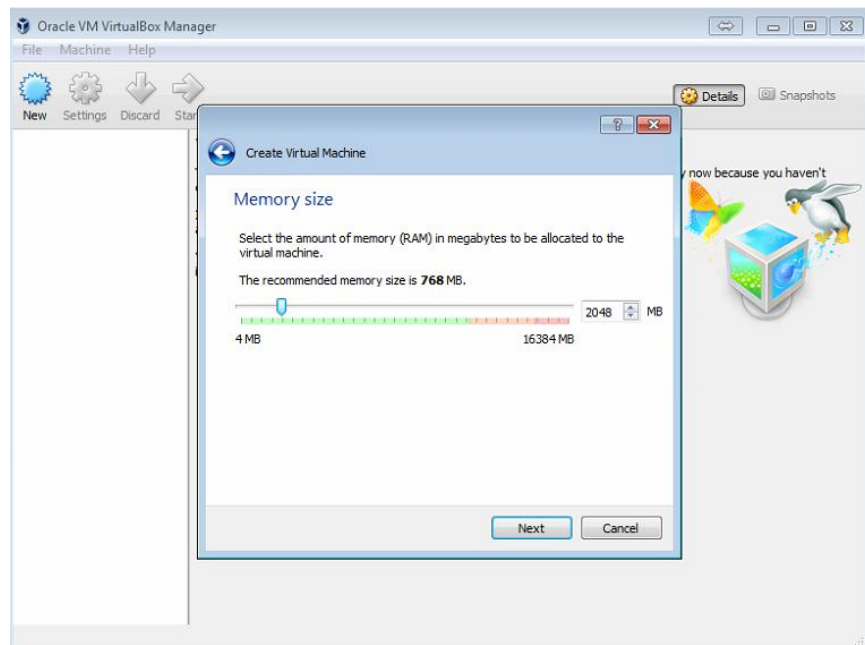
To do this, first you need to open VirtualBox, then click "New":



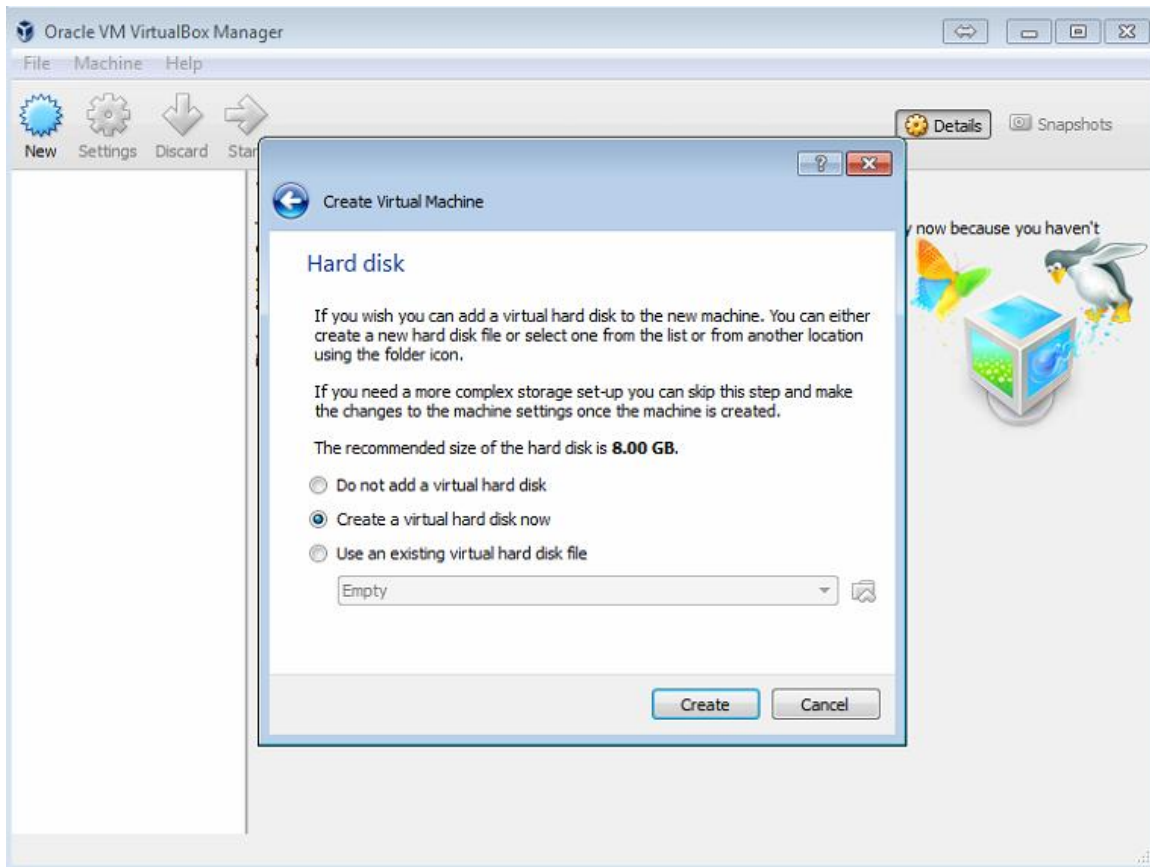
In the configuration window, you need to set a name for your virtual Ubuntu, select “Linux” for “Type”, and select “Ubuntu(64-bit)” for “Version”:



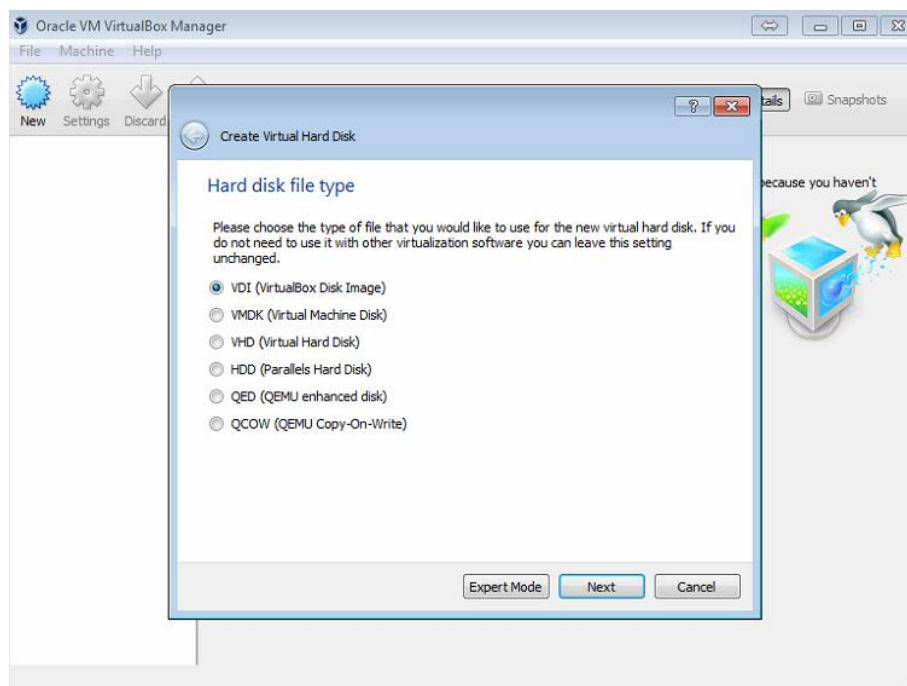
Select the amount of memory for your virtual Ubuntu. Here we recommend 2GB:



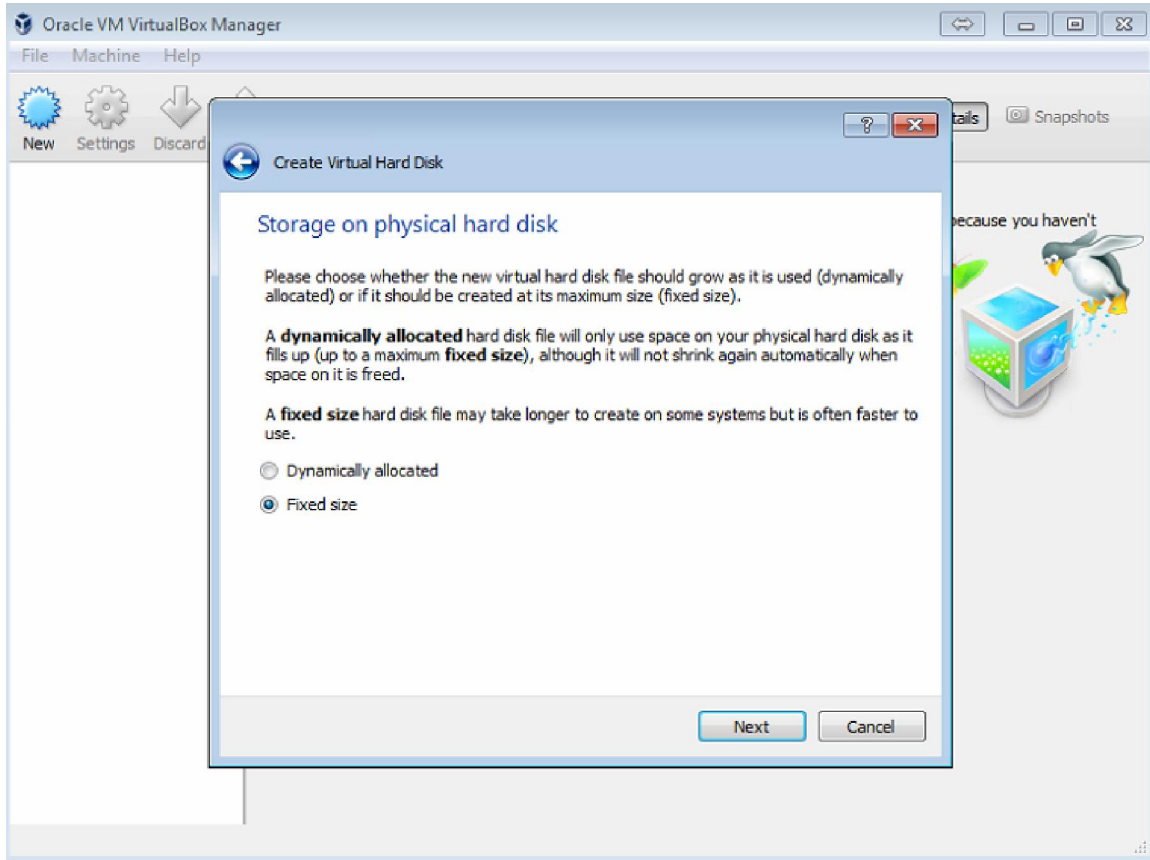
In this window, select “Create a virtual hard disk now”, then click “Create”:



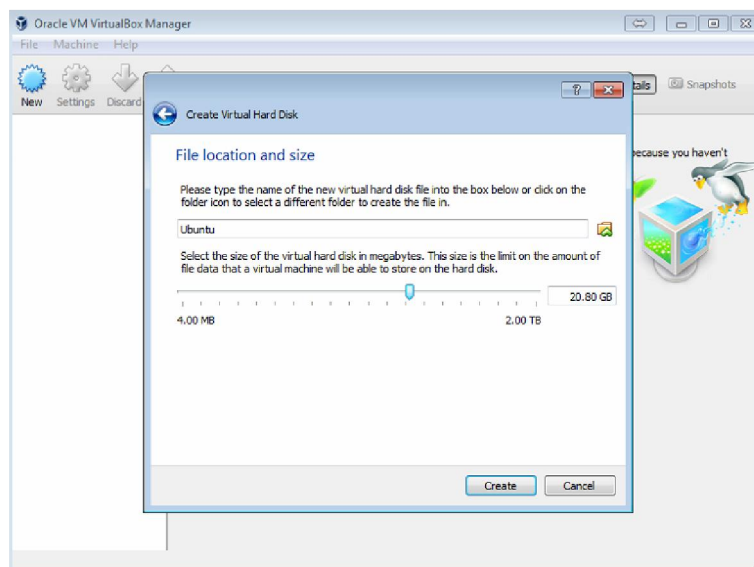
Click “Next” in this configuration window:



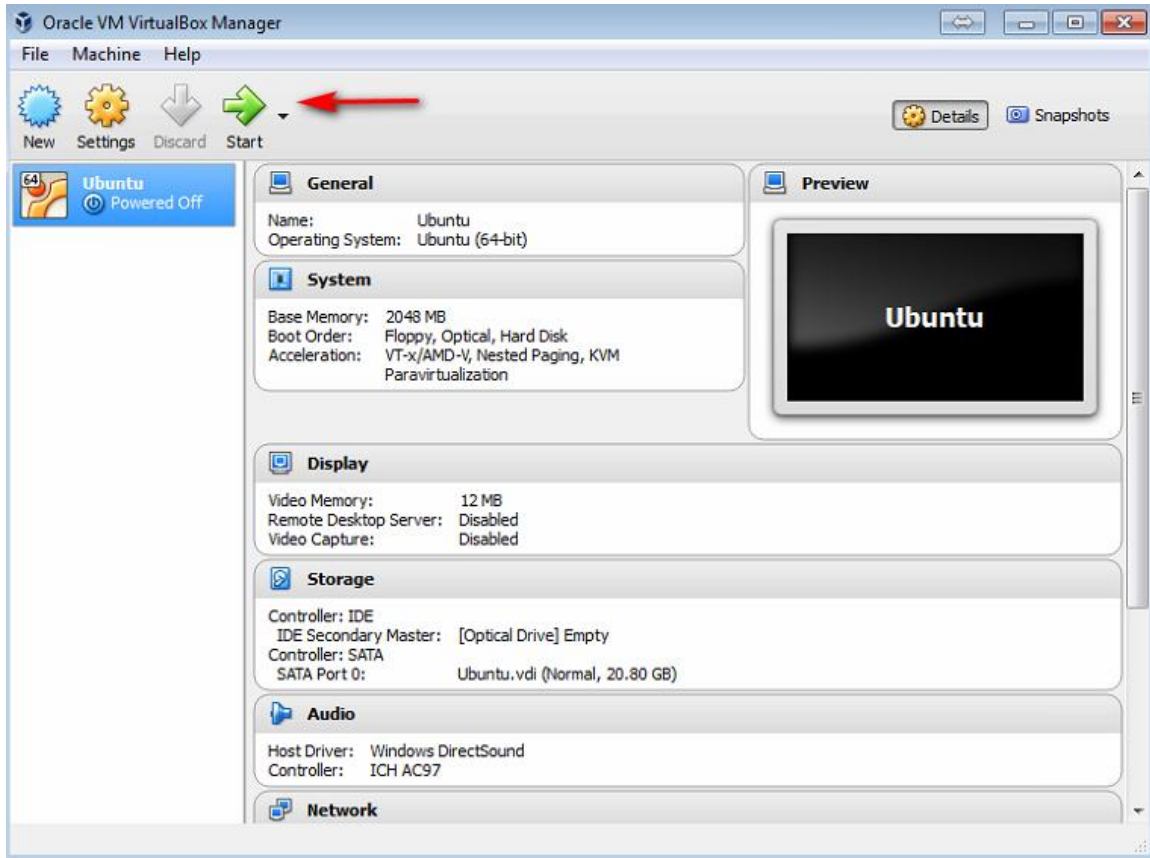
In this configuration window, you can select “Dynamically allocated”, which can make your allocated storage of the virtual Ubuntu grows if it fills up (up to a size that you will configure in the next step); or you can select “Fixed size” and then configure a size of storage for your virtual Ubuntu:



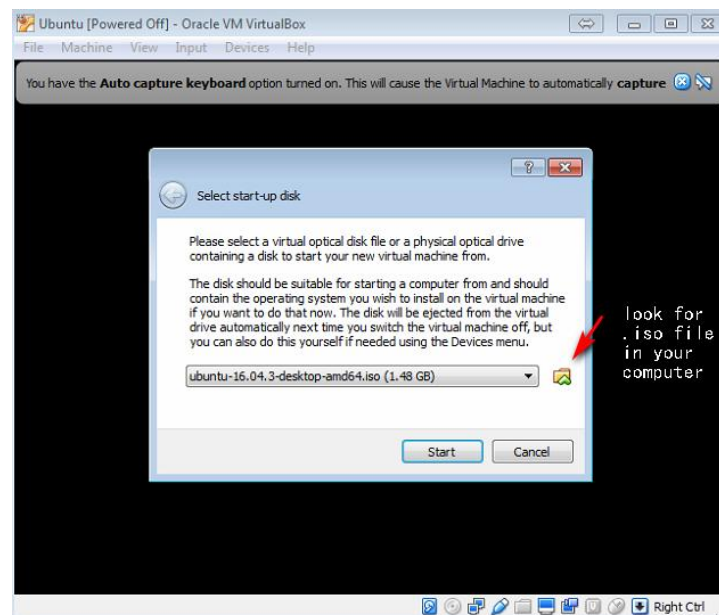
Set a size for your virtual Ubuntu. Here we set 20GB, then “Create”:



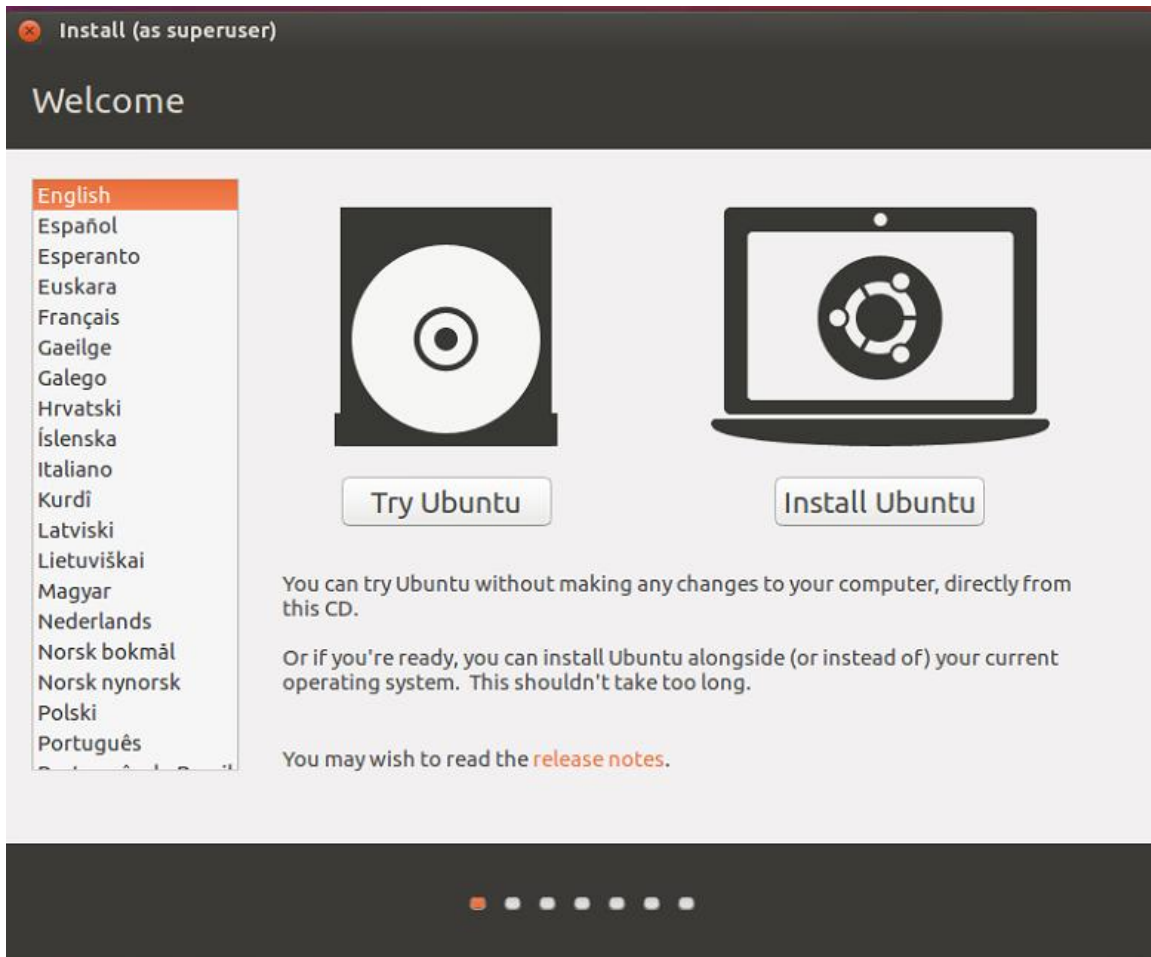
At this point, we have finished the configuration. Now you can see your virtual Ubuntu has been created in the left menu. Next you need to click “Start”:



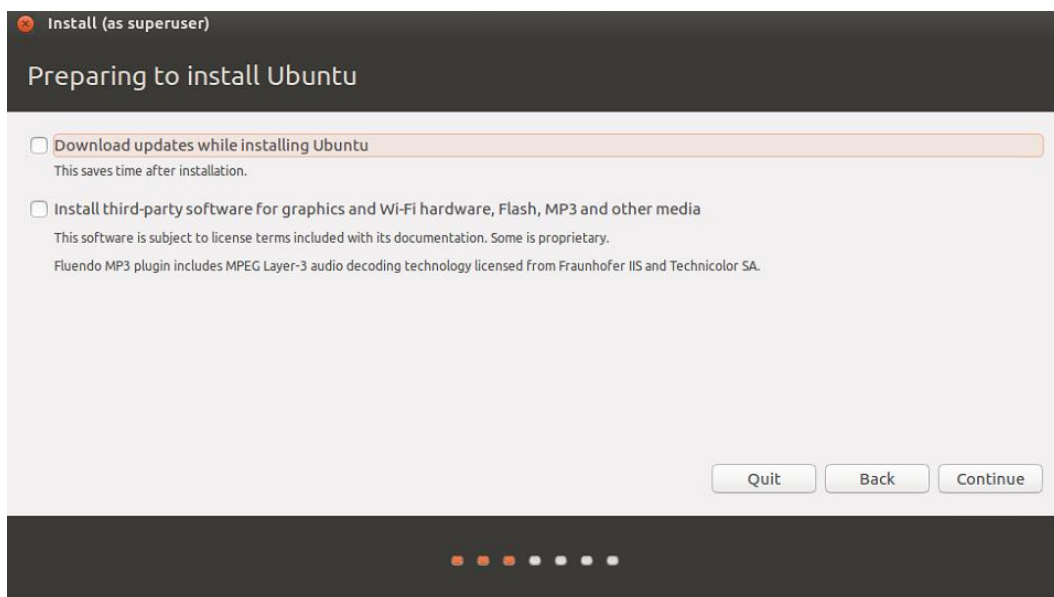
After starting, VirtualBox will ask you to “select start-up disk”, which means you need to select the .iso file of Ubuntu that you just download, then click “Start”:



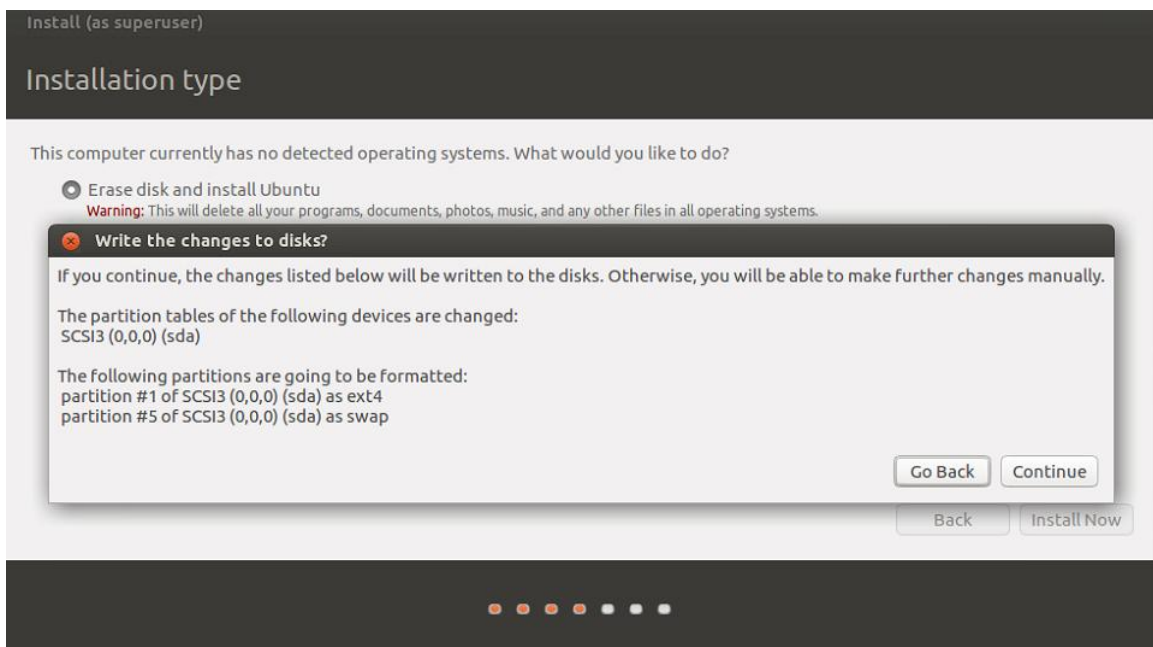
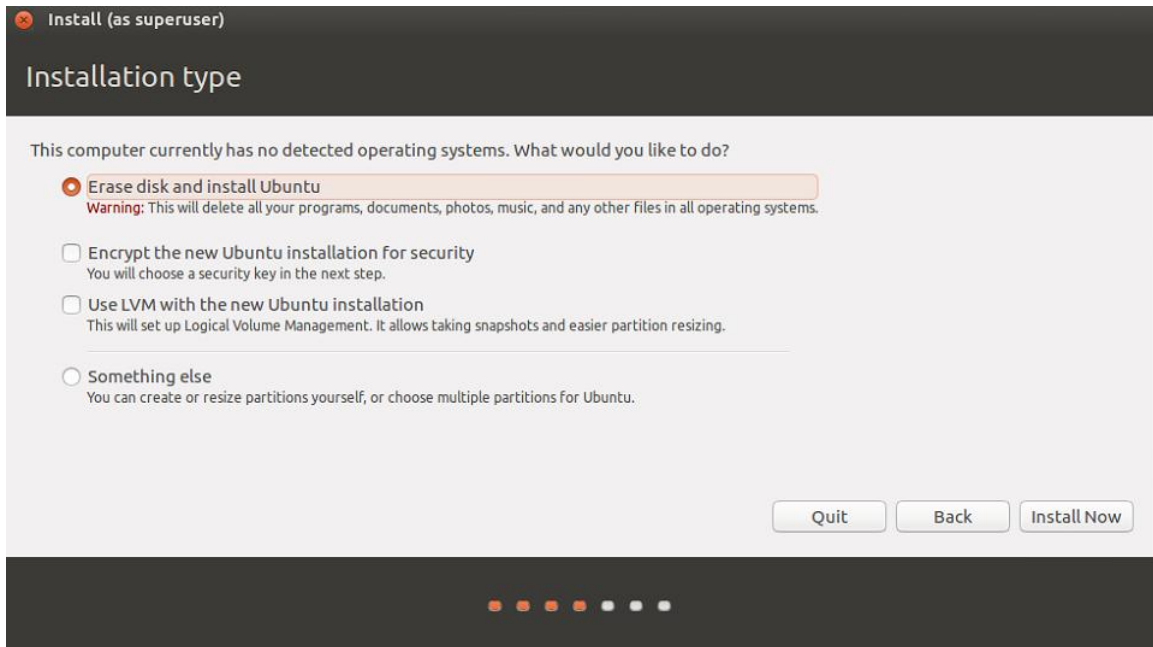
Now it's time to install Ubuntu in VirtualBox. Click "Install Ubuntu":



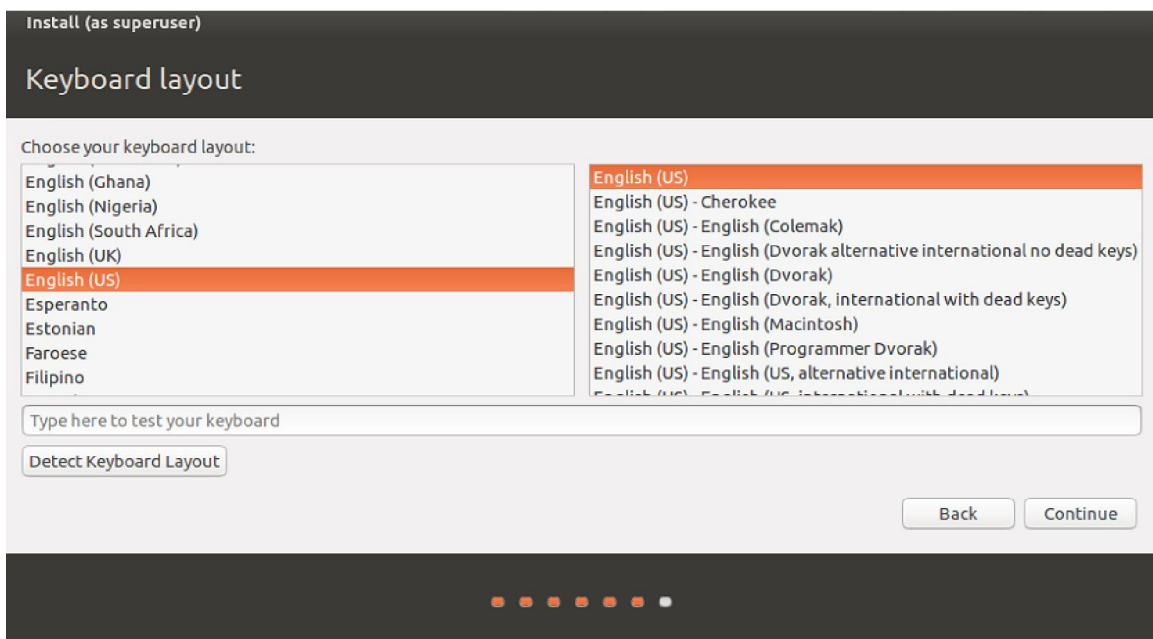
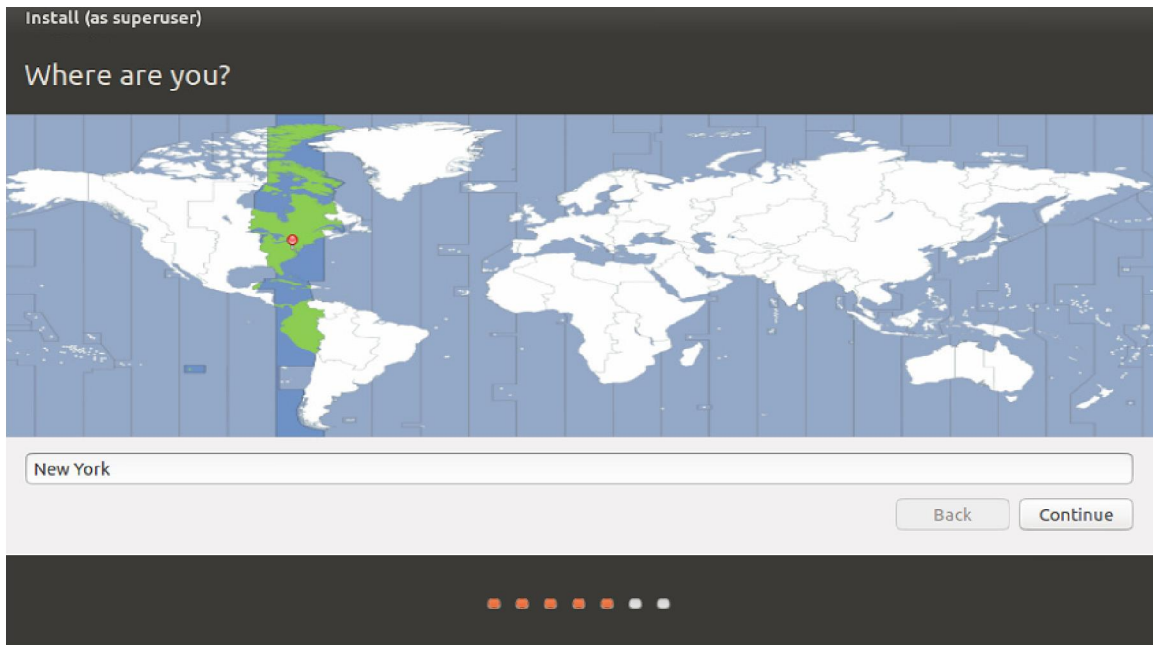
Click "Continue":



Click “Install Now” and then “Continue”:



Click “Continue” for both of them:



Set up your name, computer's name, username and password for your virtual Ubuntu, then "Continue":

Install (as superuser)

Who are you?

Your name: ✓

Your computer's name: ✓
The name it uses when it talks to other computers.

Pick a username: ✓

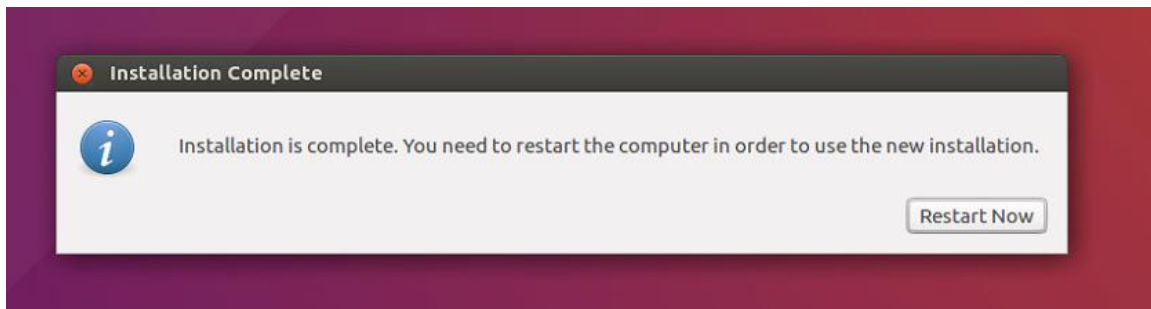
Choose a password: Weak password

Confirm your password: ✓

☐ Log in automatically
☒ Require my password to log in
☐ Encrypt my home folder

• • • • • • • •

After finishing installation, click “Restart Now”:



If you get a message “Please remove the medium and press ENTER”, just simply do so or shut down your virtual machine and start it again.

Now you get your virtual Ubuntu installed, next you can continue working on this project with it.

3. Description

In this project, you will be working with the "threads" subsystem of Linux. This is the part of Linux that supports multiple concurrent activities within the kernel. You will write a simple program that creates multiple threads. The purpose of this project is for you to get some experience using the threads primitives provided by Pthreads [1].

First, you need to write a program (i.e. program_name.c) using the Pthread library that forks 5 threads each executes the SimpleThread function below. All the threads modify a shared variable SharedVariable by adding 1 to it and display its value.

```

int SharedVariable = 0;

void *SimpleThread(void *threadid)
{
    long tid;
    tid = (long)threadid;
    int val;

    val = SharedVariable;
    printf("thread #%ld sees value %d\n", tid, val);
    SharedVariable = val + 1;
    pthread_exit(NULL);
}

```

Compile and run your program in Terminal and screenshot your program outputs in your project report.

Please check the reference to see how to get it done. (especially the part “**Creating and Terminating Threads**”)

4. Deliverables

- (1) A project report, which describes how we can compile and run your code, as well as your program outputs;
- (2) You source code.

5. Submission Requirements

You need to strictly follow the instructions listed below:

- 1) Submit a .zip file that contains all files;
- 2) The submission should include only your source code and project report. Do not submit your binary code.
- 3) Your code must be able to compile; otherwise, you will receive a grade of zero.
- 4) Provide **sufficient comments** in your code to help the TA understand your code. This is important for you to get at least partial credit in case your submitted code does not work properly.

6. Policies

- 1) Late submissions will not be graded.
- 2) You must work individually on this project. We will use anti-plagiarism tools to detect violations of this policy.

7. Resources

The Pthreads tutorials at

<https://computing.llnl.gov/tutorials/pthreads>

and

http://pages.cs.wisc.edu/~travitch/pthreads_primer.html are good references to learn Pthreads programming.

7. References

[1] POSIX Threads Programming : <https://computing.llnl.gov/tutorials/pthreads/>

[2] Pthreads Primer: http://pages.cs.wisc.edu/~travitch/pthreads_primer.html

[3] POSIX thread (pthread) libraries: <http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html>