Lab 1 - Description

(Setting up your development environment)

Lab Overview:

For this lab, we will be setting up our Linux development environment for systems programming in C. Make sure to follow all steps outlined in the "tasks" subsection of this lab description. Be sure to ask questions if anything is not immediately clear.

Core Tasks:

1. Set up your virtual machine (VM) and installing Linux Mint.
2. Install development tools (GDB, Valgrind)
3. Hello World in C. (developing, compiling, running, and testing code)


Task Details:

Be sure to follow these instructions while setting up your VM. We will be grading your work on a VM running this exact setup, so if these instructions are not followed then there is a significant chance that your submissions will not compile or run when we grade them.

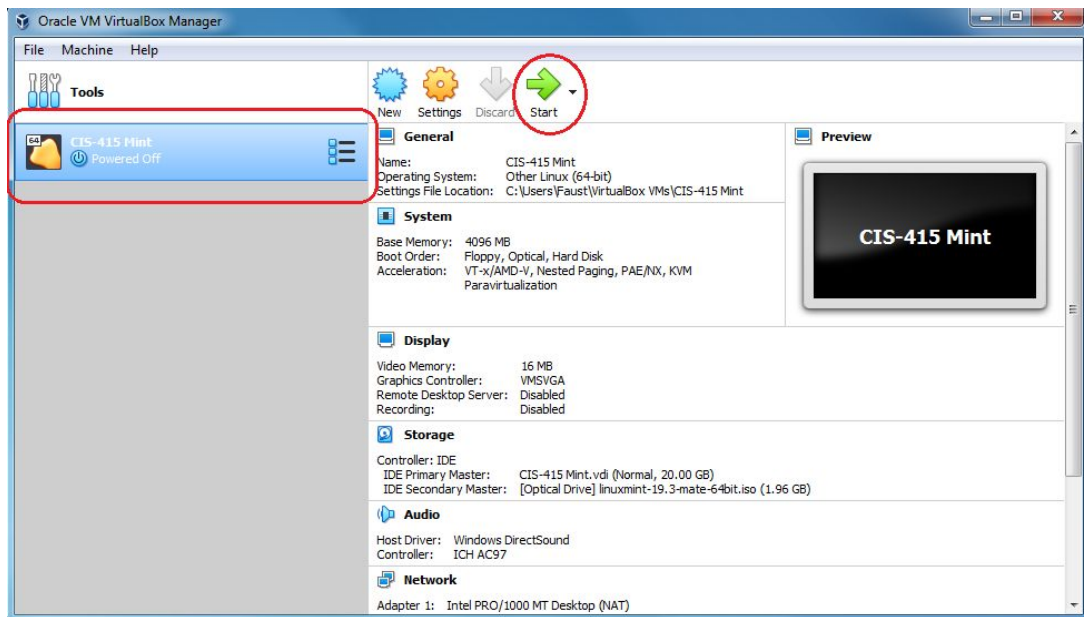**Set up your virtual machine (VM).**
1. Download and install Oracle Virtual Box.
    a. Link (https://www.virtualbox.org/)
2. Download the Linux Mint (MATE) ISO
    a. If your computer is less than 15 years old it is most likely running a 64-bit system. We have provided both the 32-bit and 64-bit versions on Canvas under the files tab.

Once you have the ISO downloaded and virtual box installed, open VirtualBox. Click on "New", then enter the name "CIS-415 Mint", select the amount of ram to use (we used 4GB of ram It's best to use a good amount so the VM runs smoothly), check the "create virtual disk now" button, then click "create". You may get a popup asking for which type of disk image to create; select
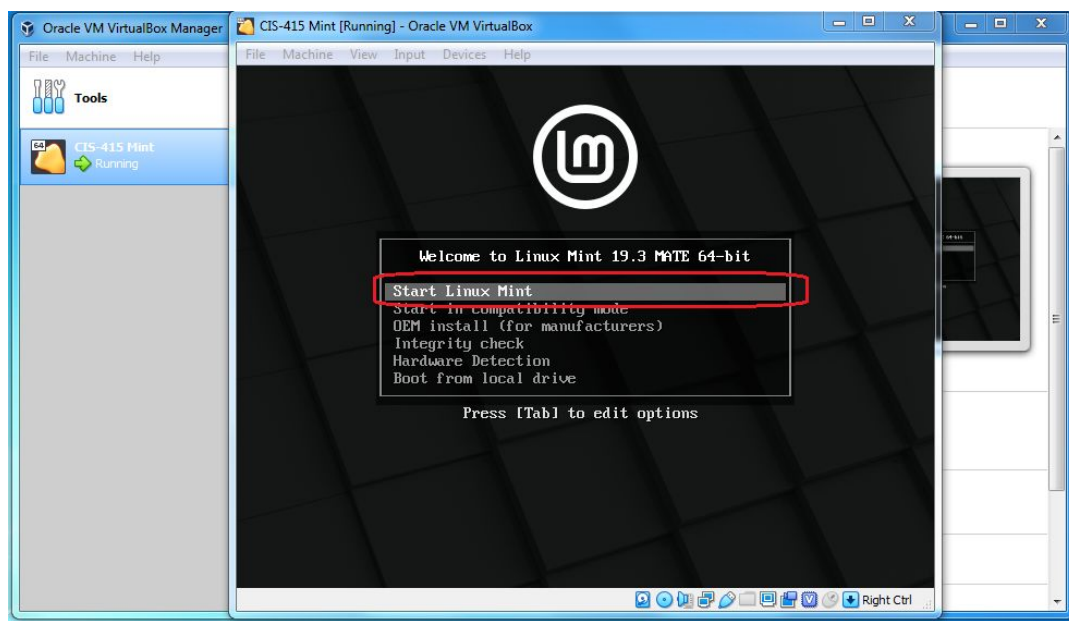
"Virtual Disk Image". (Mint requires at least 10.7G of storage space to install, so make sure you give the VM enough memory)
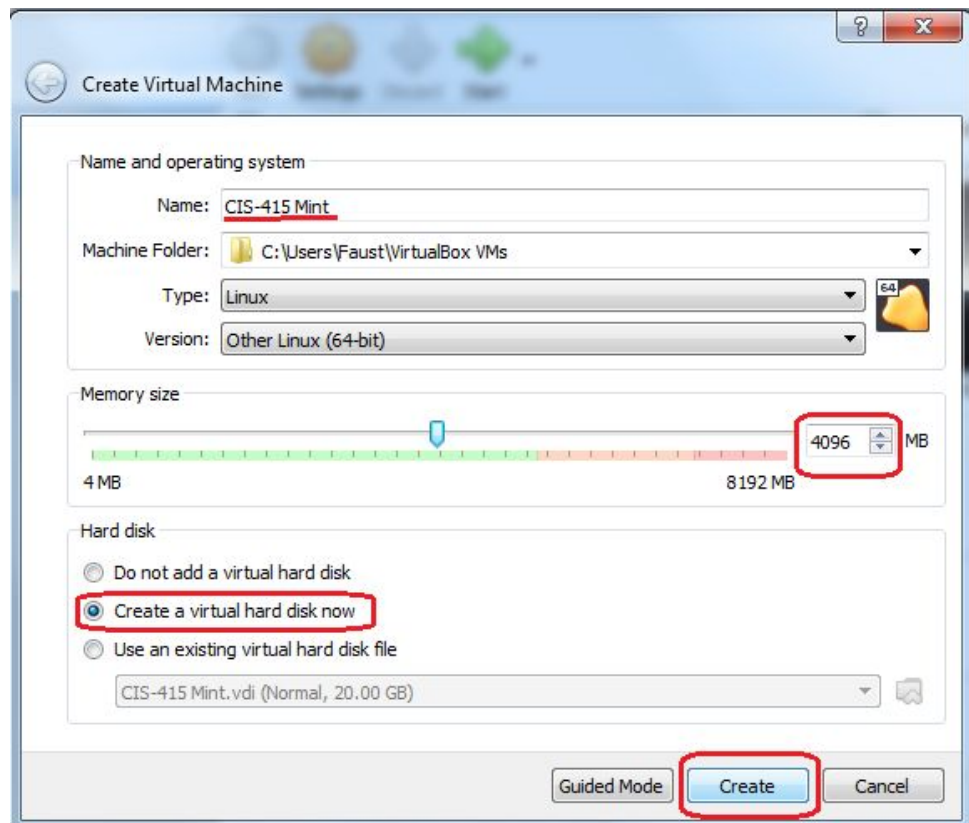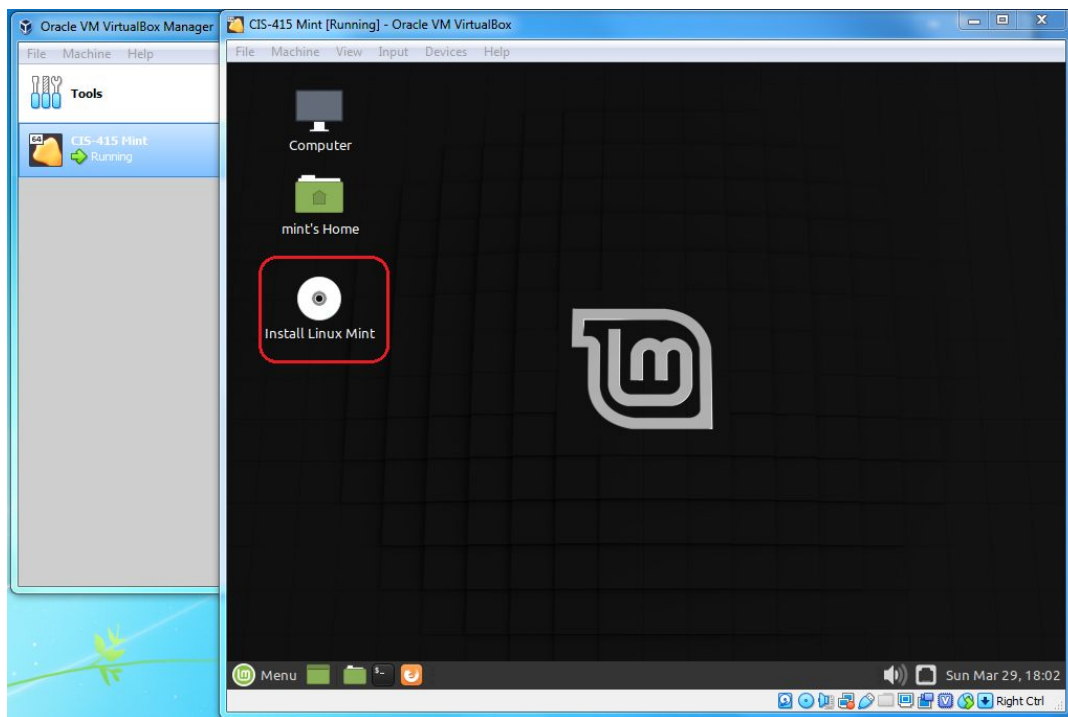

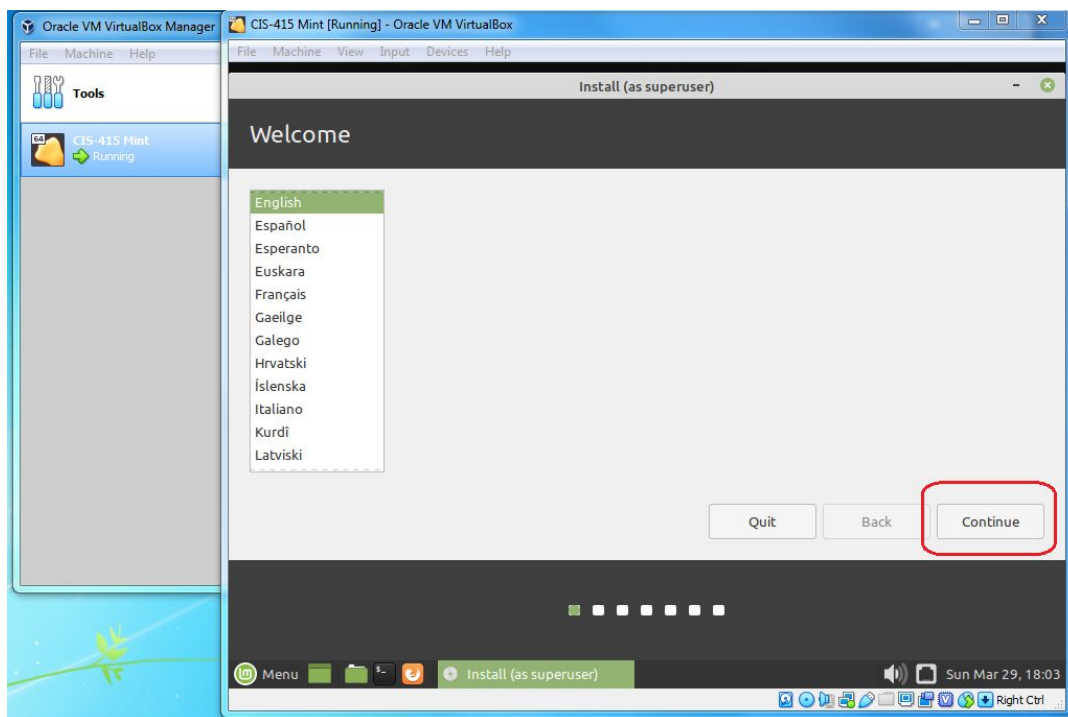Once your VM is created, click on it, then click on start.

This will start a window with your virtual machine in it. Upon loading the VM will show a bootloader for Mint. Press <enter> to start Mint in testing mode.
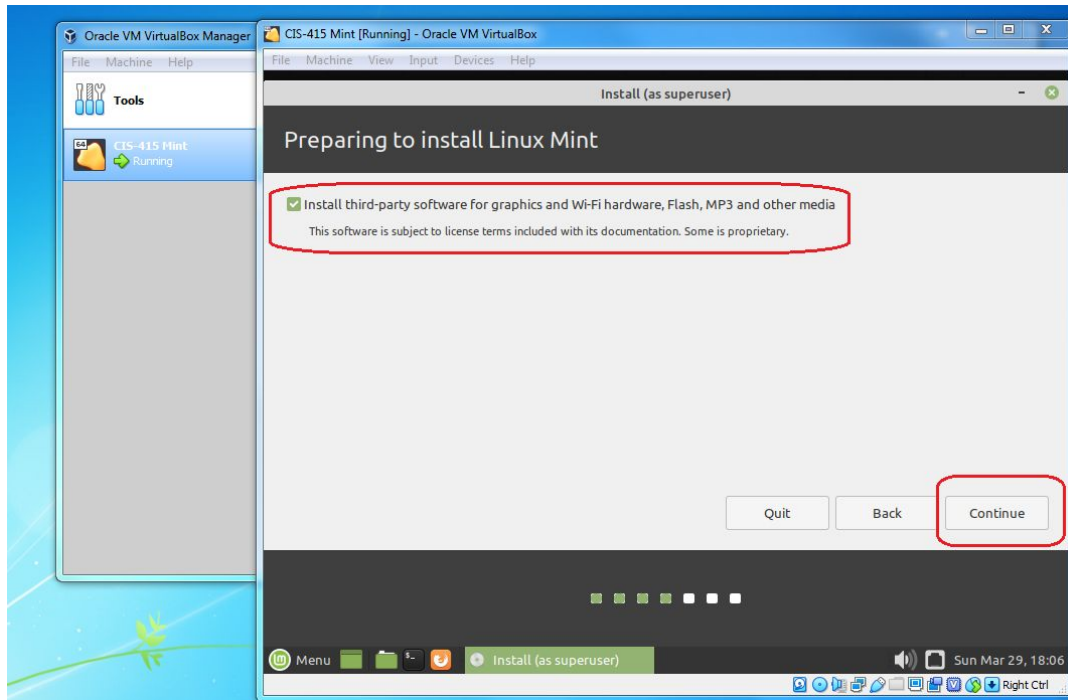

On the desktop of the Mint VM is an application to install Mint on the VM. double click on it to run it.
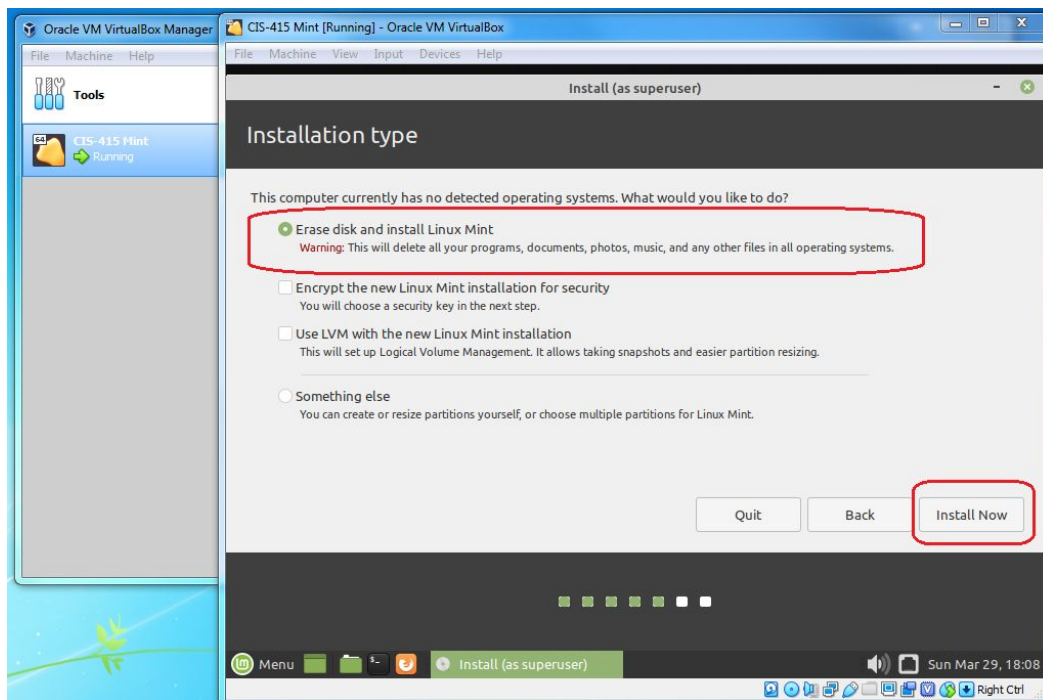
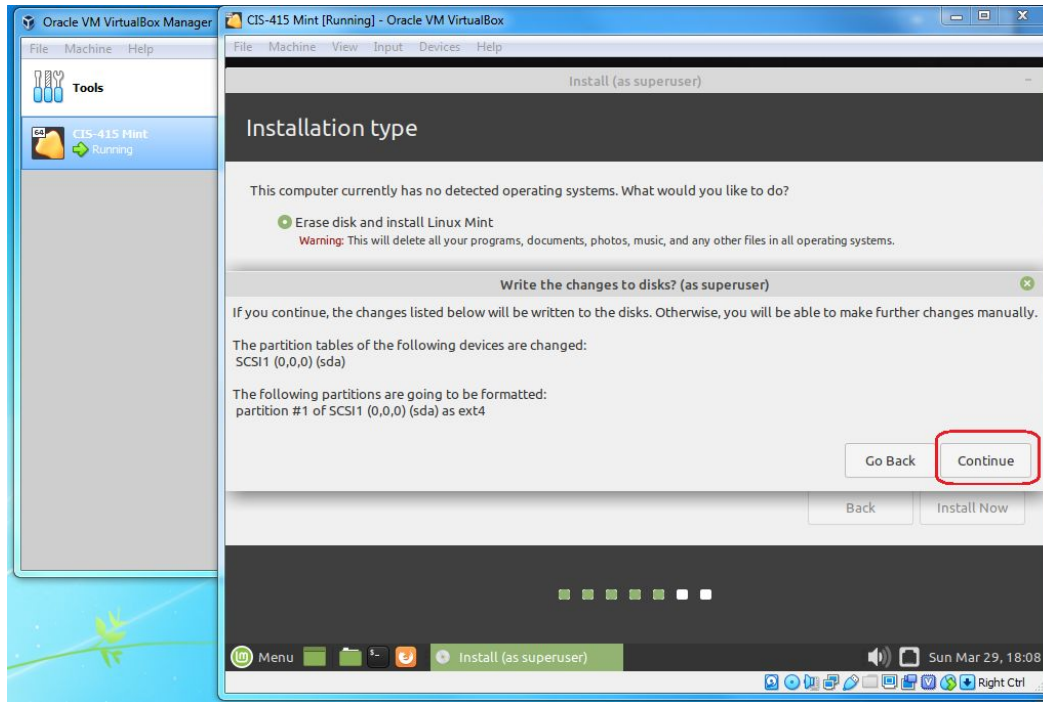Select a language then click "Continue".

Click on the option to install additional software, then click "Continue".
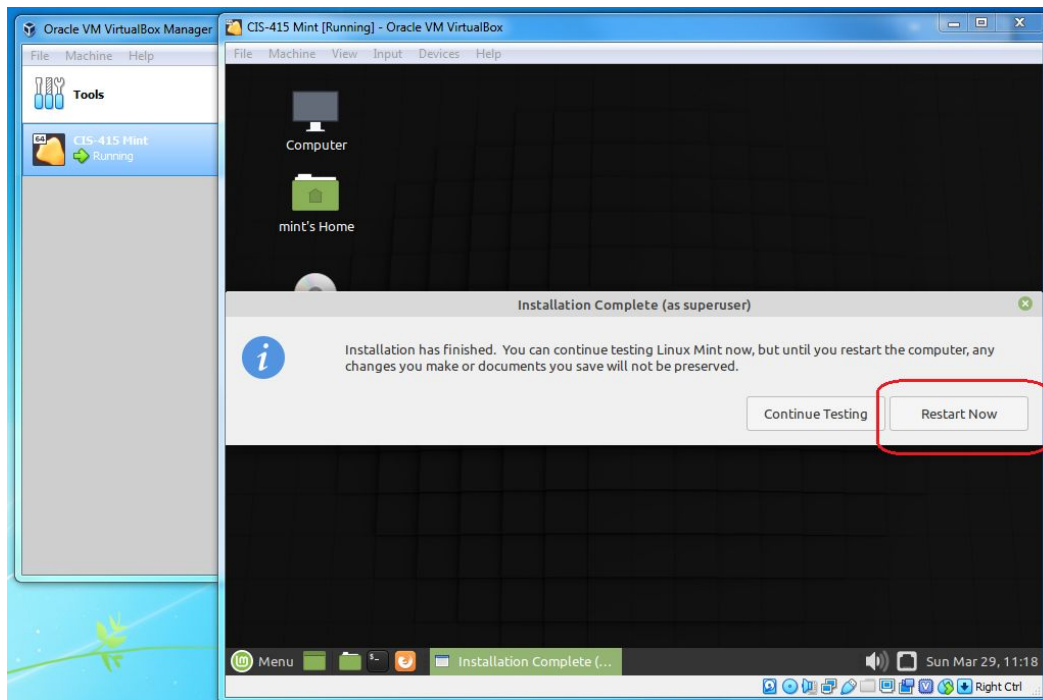
Click the first option then click "Install Now".



If you get the following popup click "Continue".

Follow the rest of the prompts. Once the installation is completed, click on "Restart Now"

This concludes setting up the VM. The special thing about Mint is that it comes with almost everything we need for this class right out of the box.

Setting up our development environment

To set up our development environment there are three basic things we need. In this class, all of our programming assignments will take place in C '89. As such we need a C compiler. Luckily for us, Mint comes with the latest version of GCC. The second thing we need is an IDE to write programs in. In this case, Mint comes with the Atom IDE. Simply click the green button on the bottom left-hand corner and type "atom" then press the enter key. You can also left-click it and select "add to desktop" for more convenient use.
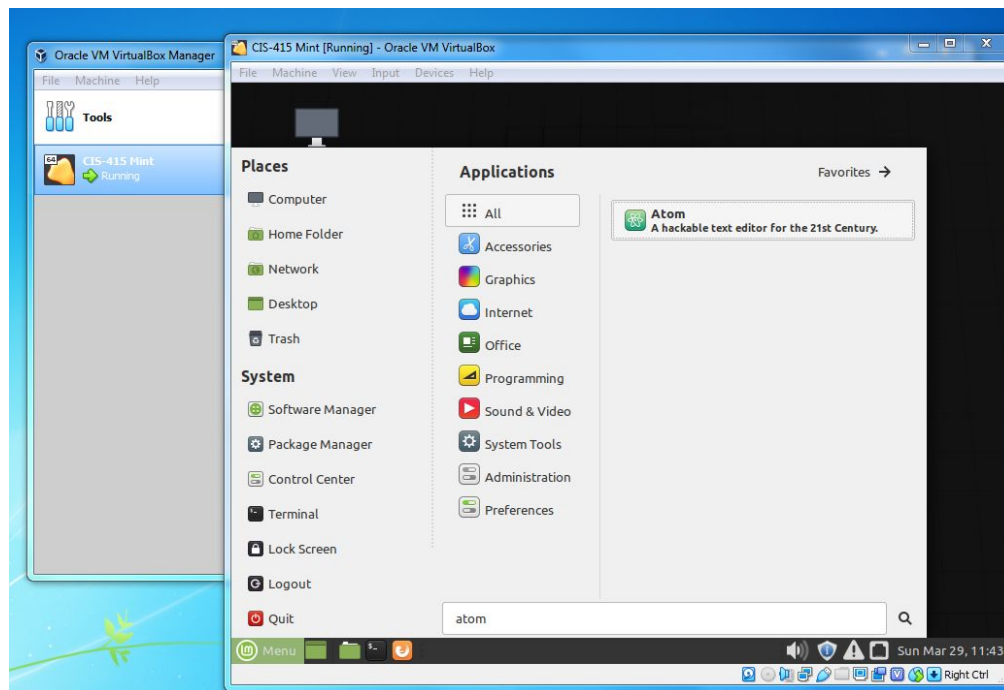
Finally, we need to install GDB and Valgrind. These can be installed by opening a terminal and typing "**sudo apt install valgrind**" and "**sudo apt install gdb**".

From here, you need to create the following file structure:

1. A folder named: cis-415
2. Inside the cis-415 folders:
    a. Create a folder named: labs
    b. Create a folder named: projects
3. Inside the labs folder:
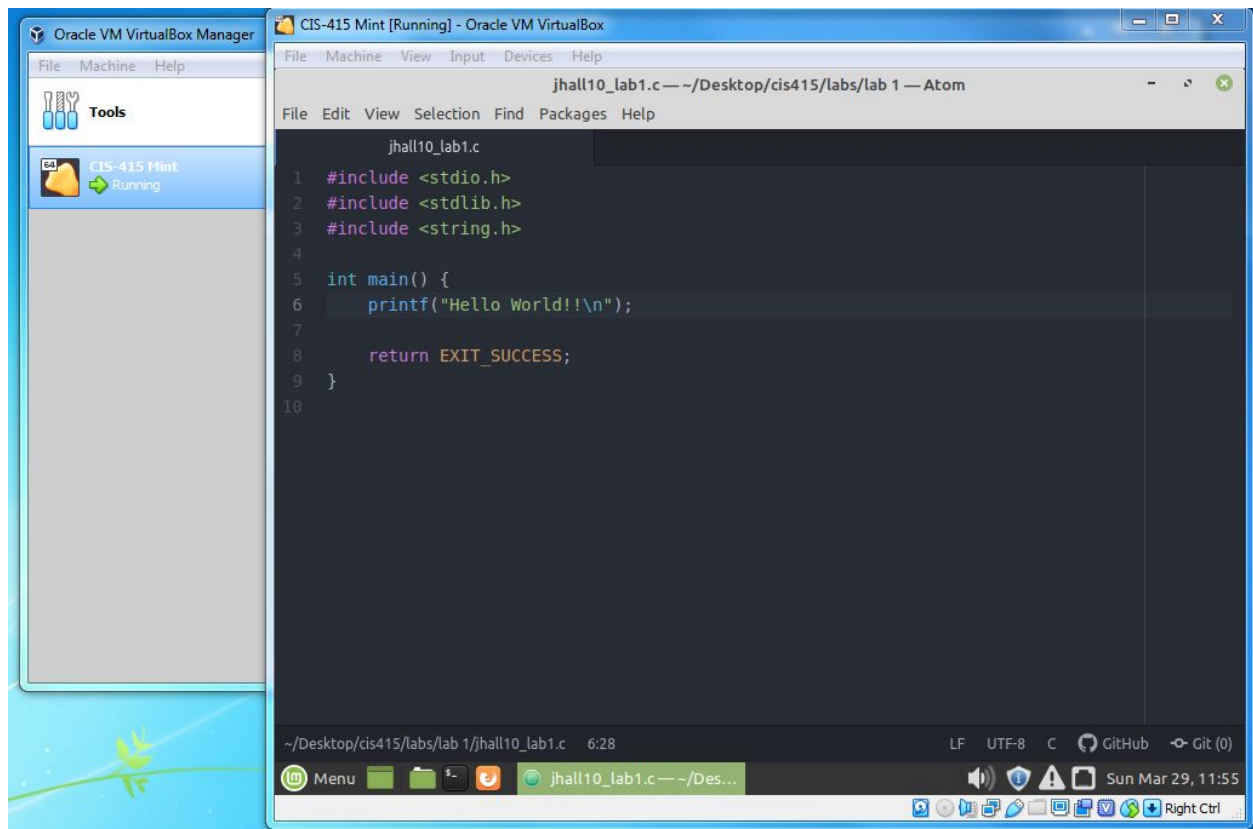   a. Create a folder named: lab1

This file structure is the schema we will be following all semester. In order to submit a lab assignment or a project, you will need to do the following:

1. Write a main.c file.
2. Compile and run it in the VM.
3. Take a full screenshot of the program running on its own.
4. Run Valgrind and redirect its output to a log.txt file.
5. Place all of these into the appropriate lab folder.

6. Tar the folder and submit it on canvas before the due date. (late submissions of the lab will not be accepted)

Hello World in C. (developing, compiling, running, and testing code)

In this section, we will describe the typical dev cycle for the labs and projects using lab1. First, you will need to navigate to/create your lab folder. From here you will create a new file named "main.c". This file will contain your program main. Then open atom, click open file, navigate the appropriate file and open it. For this lab, we will be writing a simple "hello world" program that will just print some content to the console. The program is as follows:

Once this program is done, do the following:
1. Open a terminal and navigate to the lab folder. Type: "**gcc main.c"** to compile your code.
2. Run your code and take screenshots as necessary.
3. Type: "**valgrind --leak-check=full --tool=memcheck ./a.out > log.txt 2>&1** "
4. Tar the lab folder and submit it onto Canvas.

Valgrind can help you spot memory leaks in your code. As a general rule any time you allocate memory you must free it. Points will be deducted in both the labs and the project for memory leaks so it is important that you learn how to use and read Valgrind's output. See (https://valgrind.org/) for more details.

Submission Requirements:

In order to receive any credit for a lab, completion of the labs' core tasks must be demonstrated. To do so, take screenshots as directed and include them in your tar file along with the Valgrind log showing no memory leaks. Only *.tar.gz submissions will be accepted in canvas.

The following items must be in your **lab1.tar.gz:**

1. main.c
2. Screenshots of main.c running.
3. log.txt