



European  
University Cyprus

LAUREATE INTERNATIONAL UNIVERSITIES

School of Sciences | Department of Computer Science and Engineering

---

## **CSC326 – Programming in Unix-like environments**

### **Software installation and usage guide**

**By Andreas Grondoudis**

#### **Table of contents**

Table of contents.....	1
Introduction .....	3
Obtaining the software .....	4
What software will be used.....	4
Downloading the software .....	4
Removable storage .....	4
Installing the software .....	5
Installing Virtualbox .....	5
Creating the virtual machine.....	5
Installing Ubuntu (on the virtual machine) .....	13
Installing <g++> .....	18
Using <gedit> and <g++>.....	21
Creating the first program.....	21
Compiling and running.....	21
Programming tips .....	23

*This pages has been left empty intentionally*

## Introduction

The aim of this document is to (briefly) present some simple instructions on obtaining, installing and using the software tools that are required for this course. I believe that in order to do well in the course you should have the software up-and-running in the shortest amount of time possible.

It is also my opinion that the following should hold true.

### **Constant availability (R1)**

The university provides the software already installed and configured in the relevant laboratory. However students (and instructors) tend to require the software to be installed on their own home/office machine for constant availability.

### **Minimum impact to existing configuration (R2)**

Nowadays most home machines are already installed and configured with some version of Microsoft Windows. Actually installing the Ubuntu operating system is not difficult but it certainly requires changing your system configuration to support the new operating system. I would like this to be avoided if possible

### **Minimum configuration of new OS (R3)**

When Ubuntu is installed in a 'clean' partition it usually does not require any other configurations. However, sometimes, certain configurations are required and some users might find them confusing and/or difficult to implement. I would like this to be avoided also.

Lab 106 already has everything installed and configured. If you plan to work in the lab, that is fine. However, for the reasons mentioned above I am going to be using the following configuration on my office machine:

- Keep whatever version of windows exist on the machine
- Install Oracle Virtual Box to support virtual machines running under Windows
- Install Ubuntu as a virtual machine running under Virtual Box.

## Obtaining the software

### ***What software will be used***

The software to be used in the course includes:

- Whatever version of windows you (already) have running on your machine
- Oracle Virtual Box v 3.2.8
- Ubuntu v10.04 LTS – Be advised: you must get an ISO image (see below)
- <g++> compiler (available in Ubuntu)
- <gedit> editor (available in Ubuntu)

### ***Downloading the software***

Virtualbox can be downloaded from <http://www.virtualbox.org>. Go to the download section and make sure you select the correct version for your platform.

Ubuntu can be downloaded from <http://www.ubuntu.com>. Go to the **Desktop edition** find the link for **Get Ubuntu Desktop Edition** and then select **Alternative downloads**. Once on that page you can click on any of the mirrors to get to links for ISO images.

#### Why get the Ubuntu as an ISO?

Because virtual box will be able to use it to install ubuntu (see the installation of ubuntu later on)

### ***Removable storage***

If your machine is using an Intel (i386-base) processor, then you are in luck. I already have the above software downloaded. Bring me some removable storage and you can have it. Note:

- the Ubuntu ISO is 685 MB and is for i386-platforms
- the VirtualBox is around 80 MB and is for i386 and amd64 platforms

## Installing the software

In this section you are presented with instructions on how to install:

- Virtualbox
- Ubuntu
- <g++> (the c++ compiler)

### Installing Virtualbox

Virtualbox is a standard windows application so all you have to do is click on setup.exe and follow the wizard-based prompts. Nothing out of the ordinary there; after a few clicks you should have a shortcut on your desktop that looks like the one below



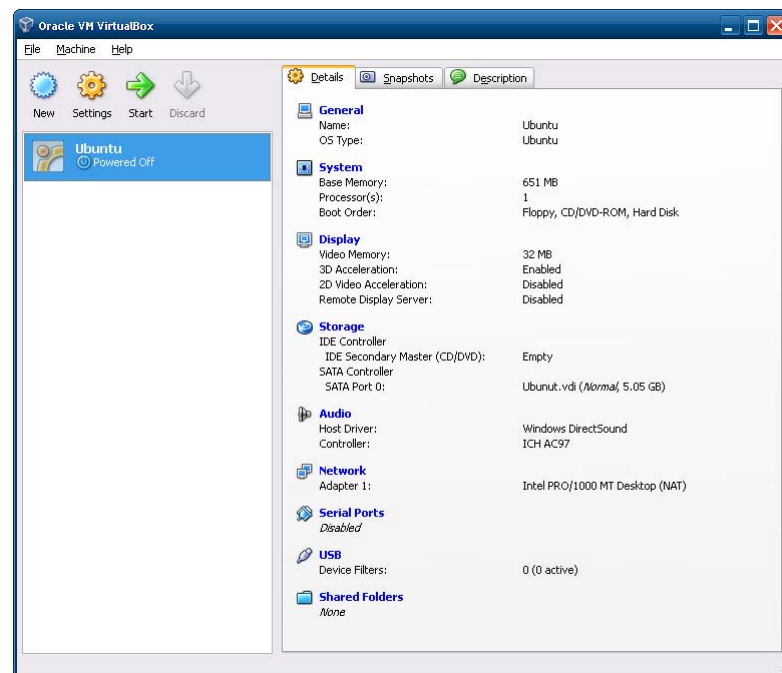
We will install Ubuntu under Virtualbox. The description that follows assumes that:

- you have already installed Virtualbox
- you have at least 5 gigabytes of disk space available
- you have obtained the relevant Ubuntu ISO image and know where it is in your file system structure.
  - o Mine was in z:\Data images\ubuntu-10.04.1-desktop-i386.iso
- You are operating with Administrator (or at least power user) privileges

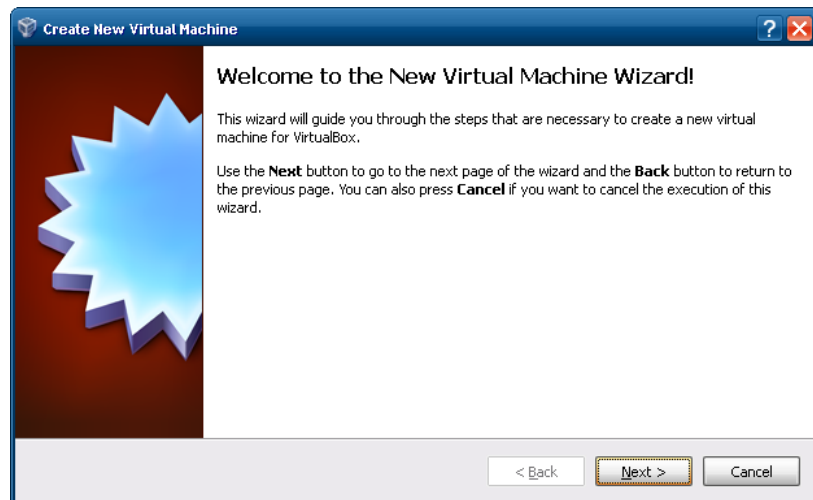
### Creating the virtual machine

- 1 Start up Virtualbox.

As you can see I already have my original Ubuntu installation but I will do another one for this demonstration

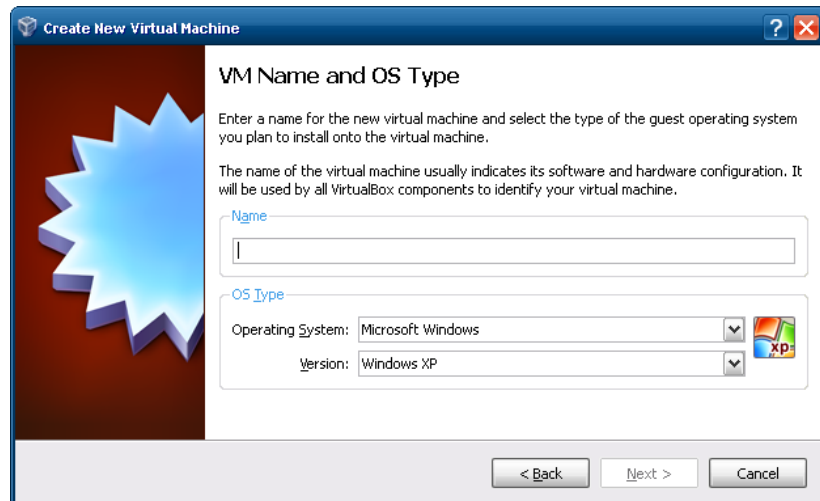


- 2 Click the <New> button on the top left of the Virtualbox window.
- The wizard starts up and you going to set up the new virtual machine.



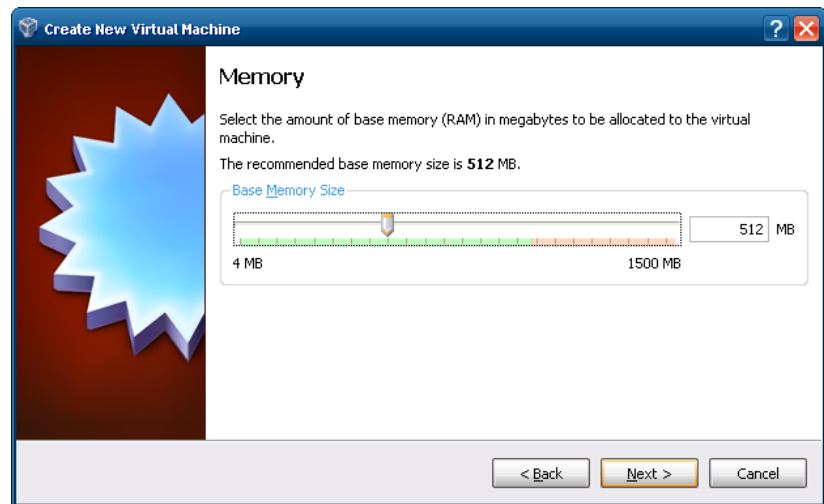
Your action: Click next

- 3 Virtualbox is now asking you to name the virtual machine and supply information on to what OS it will use.



Your action: Enter <Ubuntu> as the name and select <Linux> at the operating system dropdown and <Ubuntu> as the version. Click next when finished

- 4 Virtualbox is now asking how much RAM should this new system use from your machine's available RAM.



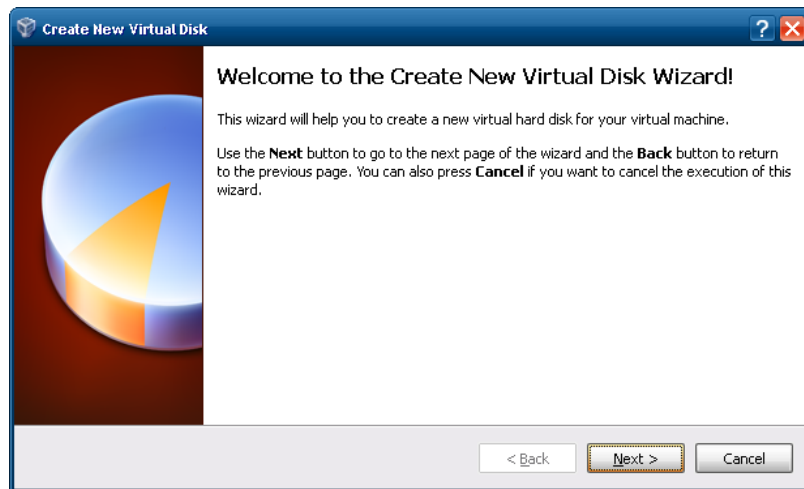
Your action: Depending on the RAM you have installed select something appropriate. I have 2GB RAM on my machine and I left the setting as it is (512MB). Click next

- 5 Virtualbox is now asking the location of the hard disk to use for the virtual machine.



Your action: Leave the selections as they stand. The <Boot hard disk> should be ticked and the <create new hard disk> must be selected. Click next

- 6 The virtual hard disk wizard starts up and you will select which space to use



Your action: Click next

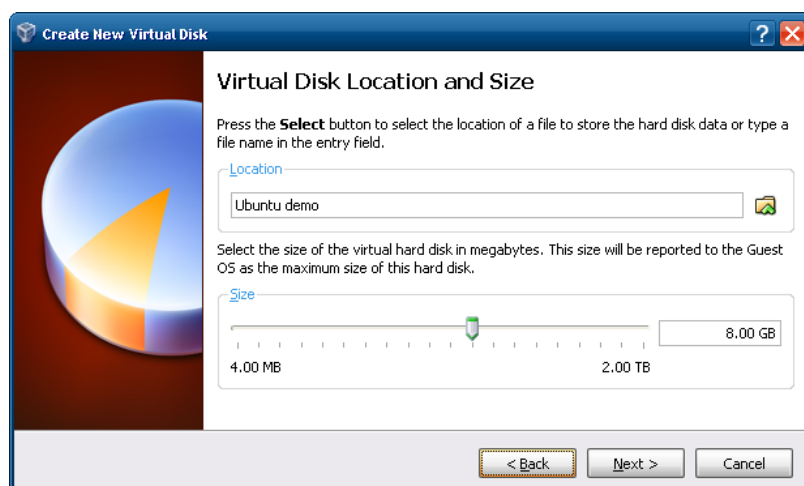
- 7 The wizard asks about the type of storage (dynamic or fixed)



Your action: Select <fixed-size storage> and click next

- 8 The wizard now asks where and how much space to take up.

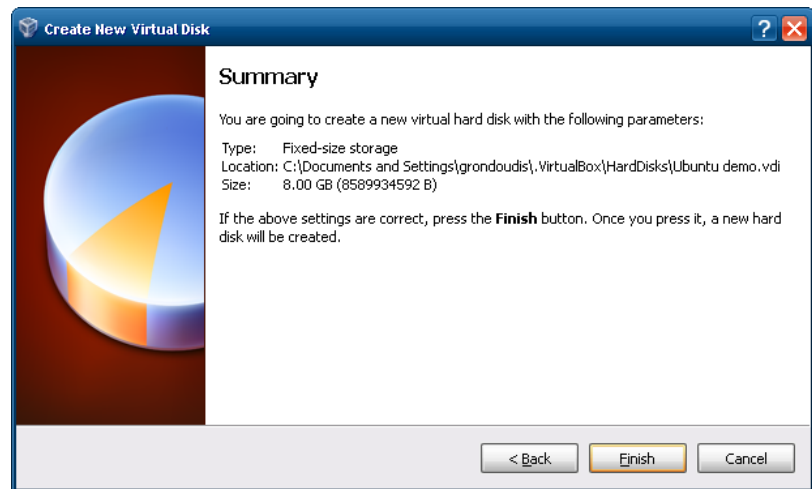
Usually the virtual disks are created under the Virtualbox directory by just having one file. The location is the file and you can select the size at the bottom



Your action: leave the settings as are. The name of the virtual disk will be the name of the system you entered (Ubuntu) and the recommended size is 8 gigabyte. Click next

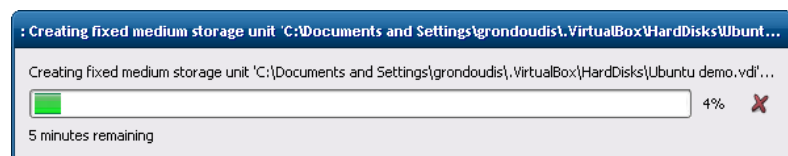


- 9 The wizard confirms the action and you are finished with the creation of the virtual hard disk



Your action: read the details and ensure that you know what is happening. Click finish to create the disk.

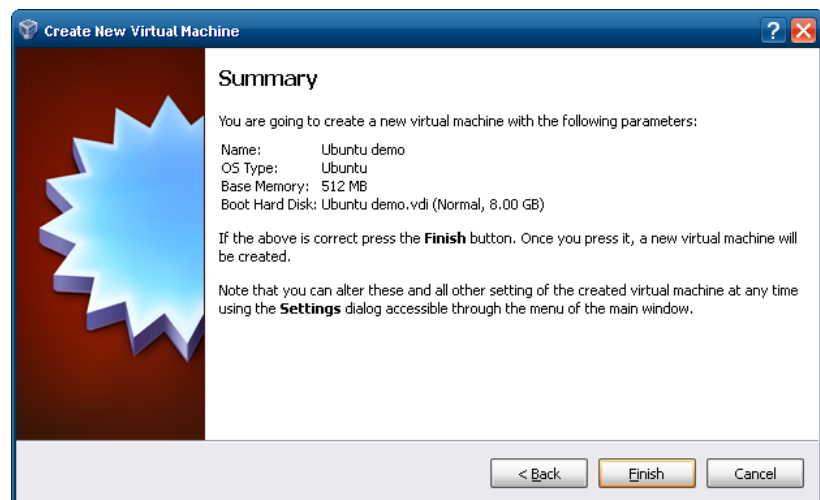
- 10 The virtual hard disk is created



Your action: Nothing just wait for this to be finished

You will then be taken back to resume the virtual machine setup

- 11 Virtualbox is now confirming that it is ready to create the virtual machine



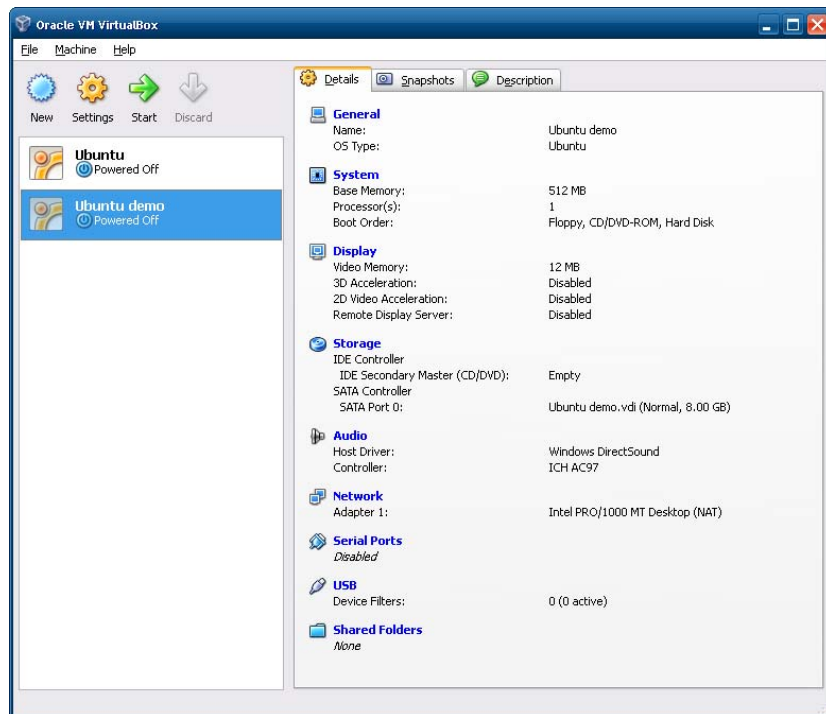
Your action: Click finish.

12

The new machine is created.

As you can see a new entry has been entered below my original Ubuntu.

The **Ubuntu demo** machine is now ready to be installed



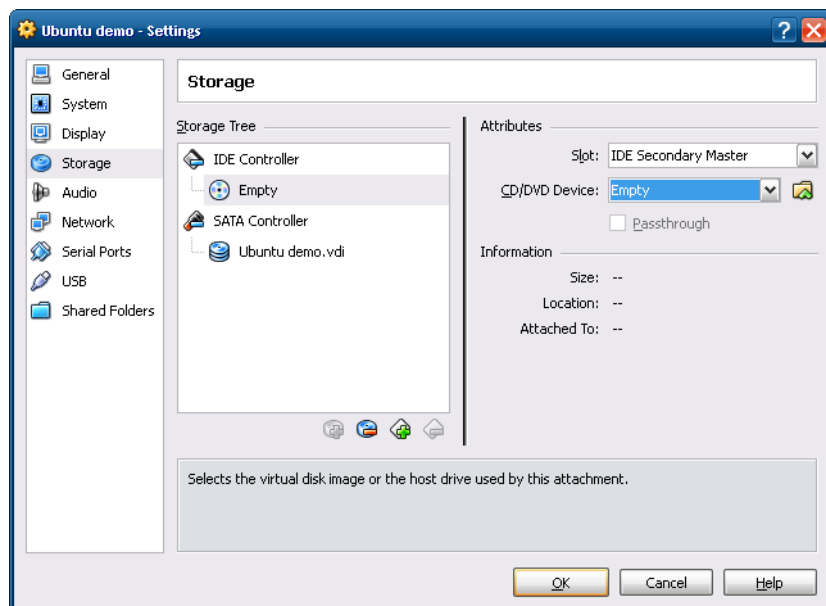
Your action: Click the yellow <Settings> button

13

The settings dialog appears.

Here you can configure the setting of the virtual machine.

What we want to do now is load the image in the CD/DVD drive and start the Ubuntu installation

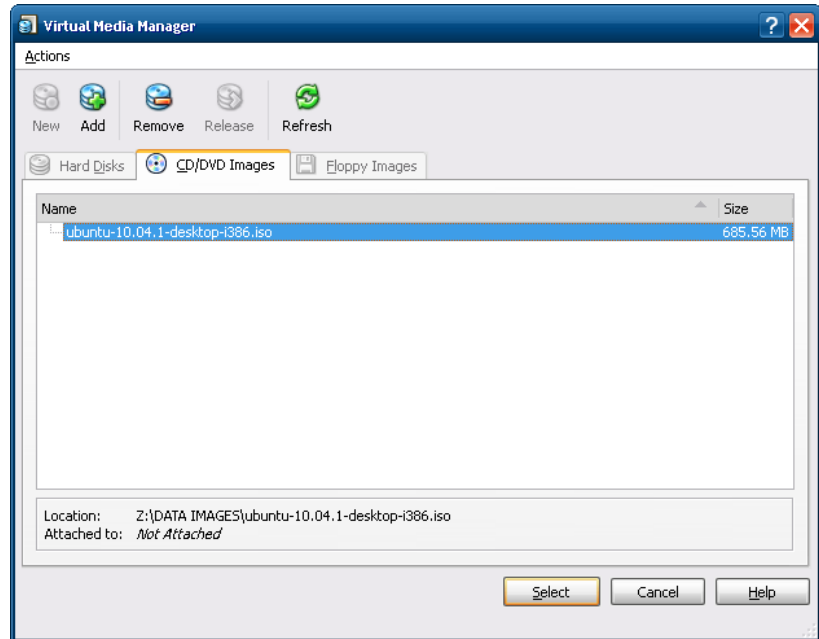


Your action: Select <Storage> from the listing on the left. The <Storage tree> appears and as you can see there are two items. 1) at the bottom the SATA controller (the hard disk) named Ubuntu demo and 2) the IDE controller, the CD/DVD drive which currently is indicated as empty.

Go to <Attributes> and the CD/DVD device and click the little icon to the right of the drop down

- 14 Virtual box will now ask for the CD/DVD image to load so that it will appear to be inserted in the CD/DVD of the virtual machine.

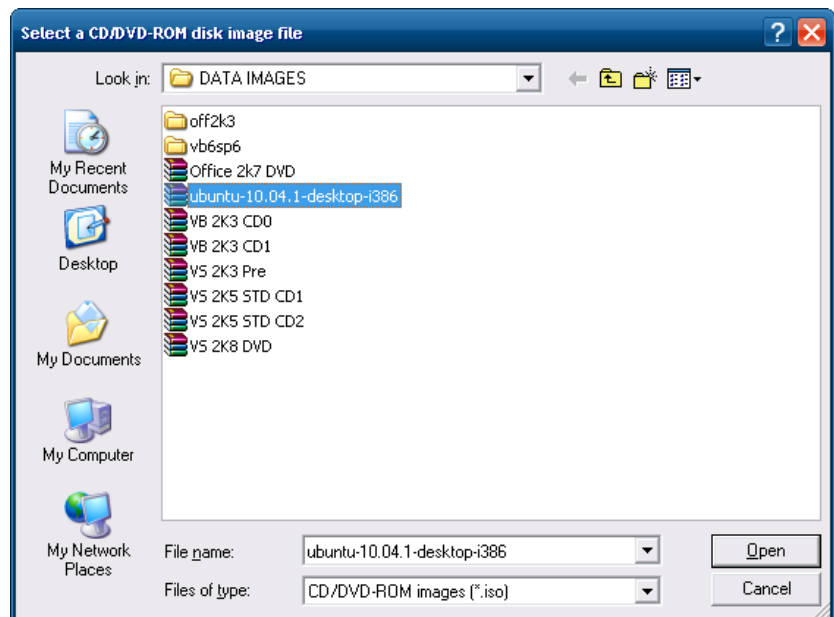
The virtual media manager will have (at the bottom) a listing of previously used media but you can add a new



My listing already includes the image file and I can select it; but I am going to demo how to add a new one.

Your action: Click the <Add> button at the top

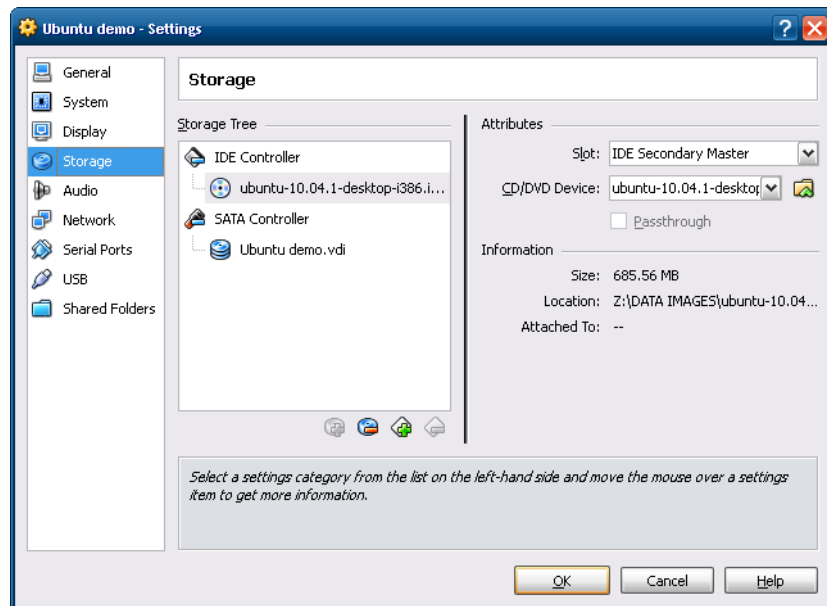
- 15 The dialog opens up and you are asked to select an image



Your action: find the Ubuntu (ISO) image file in your file system; select it; and click open.

The entry will now be entered in the media list; select it and click <Select>

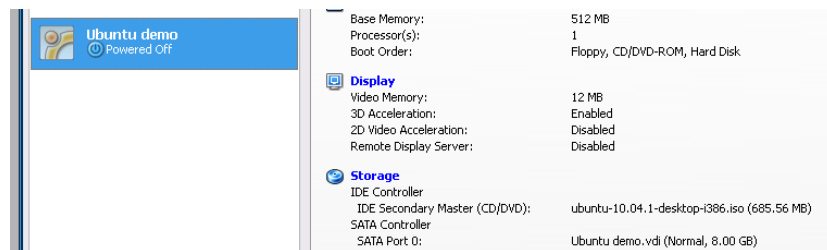
- 16 You are taken back to the settings dialog and as you can see the CD/DVD device dropdown has now changed.



Your action: Click OK to apply changes and load the image

- 17 In Virtualbox with the <Ubuntu> virtual machine selected.

The CD/DVD contents have now changed.

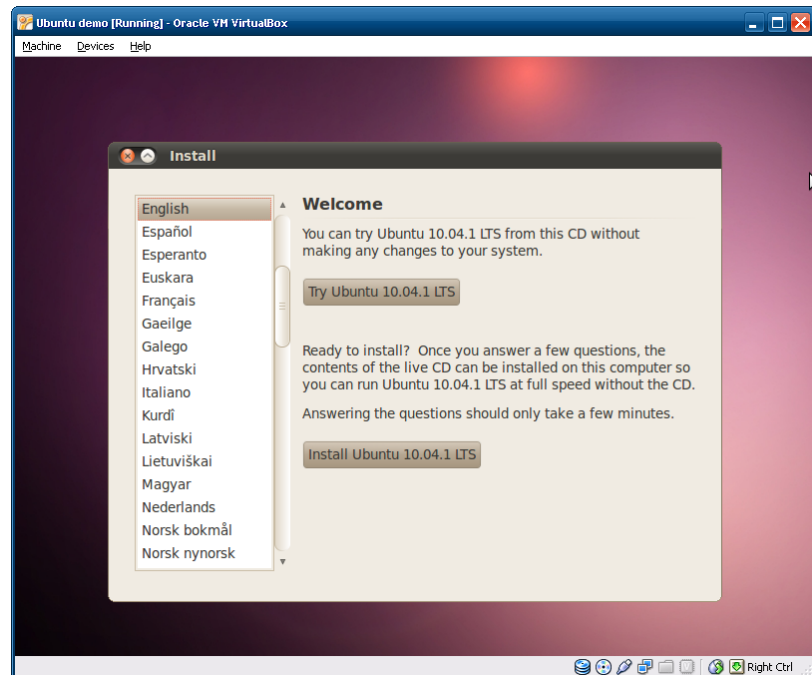


This is it with the virtual machine. Next, we start the machine and install and configure Ubuntu

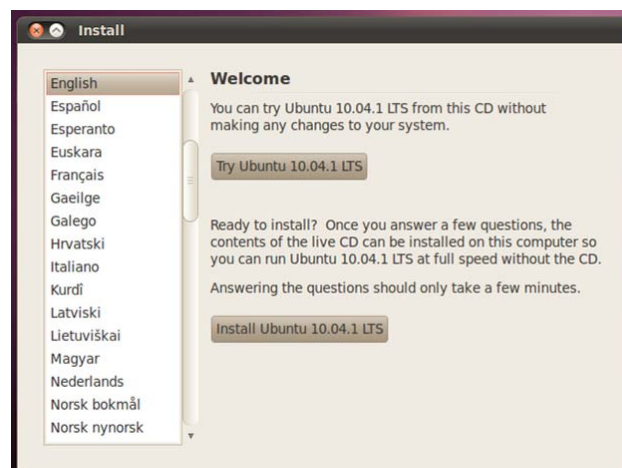
Your action: Click the <Start> green arrow button to start (bootup) the virtual machine.

## Installing Ubuntu (on the virtual machine)

- 1 A new window will open up and a couple of minutes later you will see something like the capture below. The virtual machine has started, the ISO image has been loaded and the installation of Ubuntu has started.

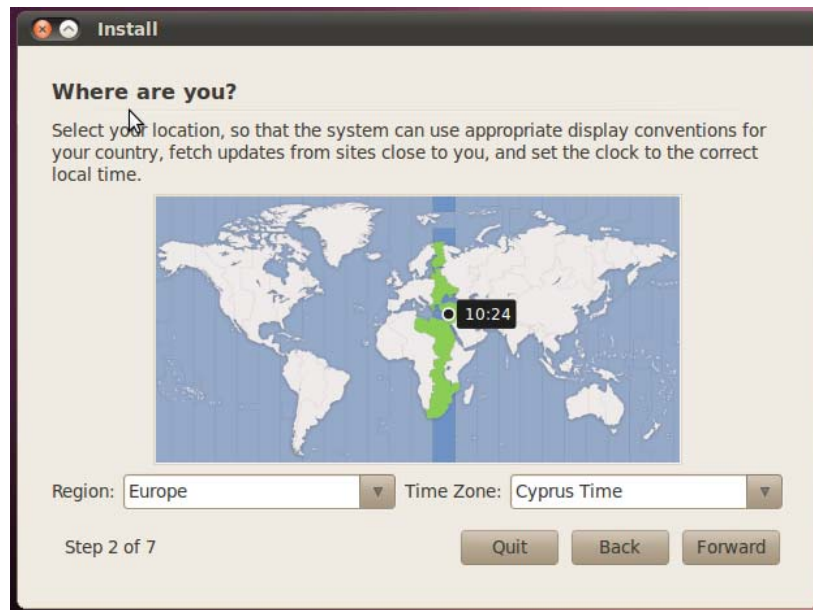


- 2 Select the language



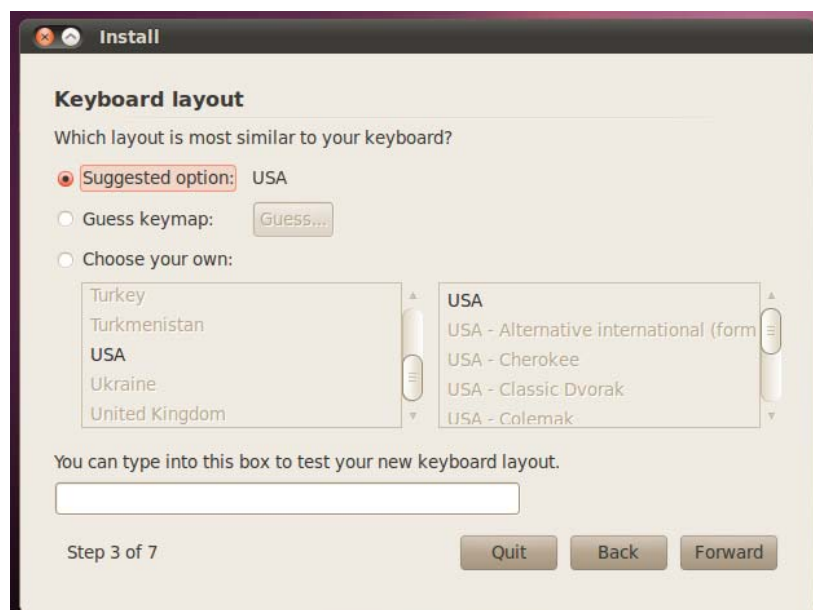
Click the Install Ubuntu button at the bottom

- 3 Select your location to set up the time zone for your machine



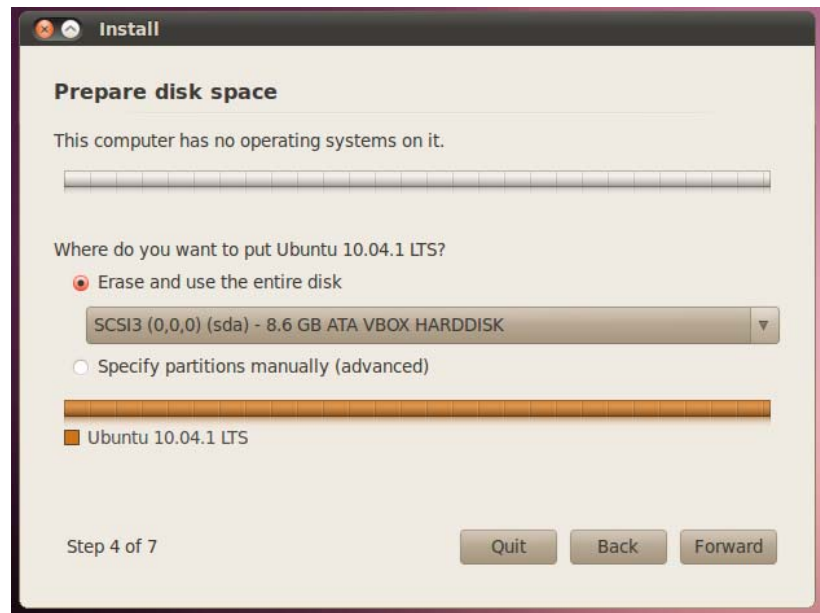
Click <Forward>

- 4 Select the keyboard layout for the machine



Click <Forward>

- 5 Set up the disk to install (leave settings as they are (Erase and use the entire disk))



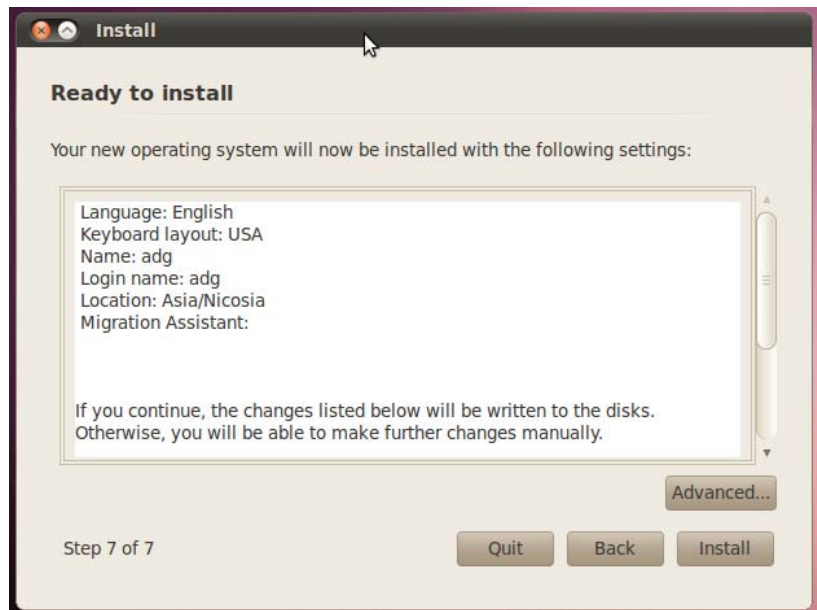
Click <Forward>

- 6 Set up your credentials for the machine



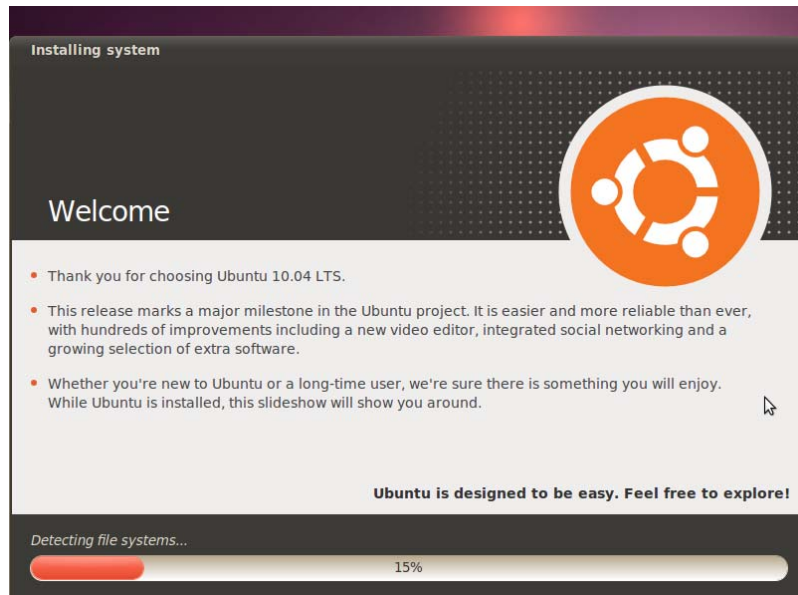
Click <Forward>

- 7 Ubuntu setup will confirm the details and prepare to begin installation



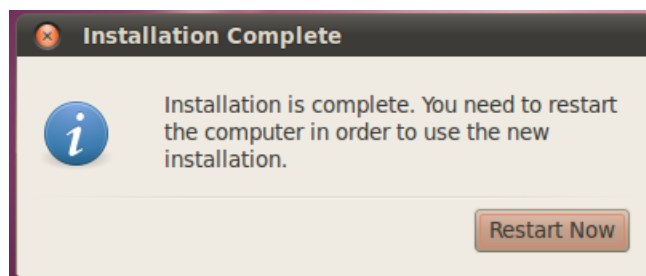
Click <Install>

- 8 Ubuntu carries on and installs



Your action: just wait (it does not take that long)

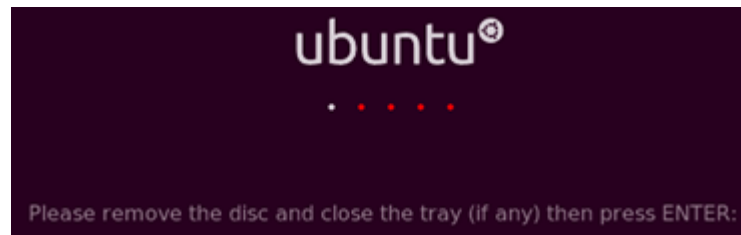
- 9 In less than 15 minutes the installation is complete and you can restart your new system



Click <Restart now>

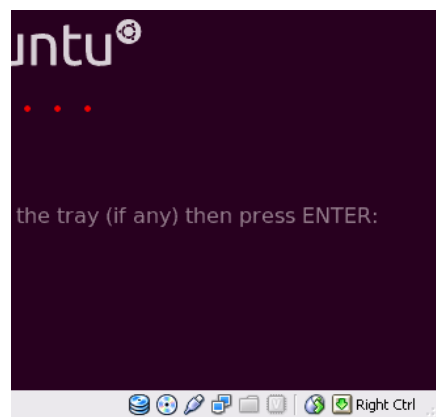


- 10 The disk is detected and Ubuntu notifies you

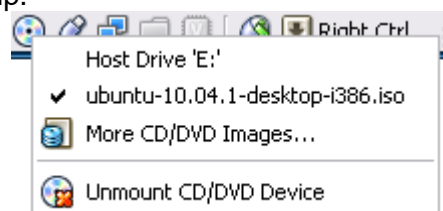


You must unmount the CD/DVD image

- 11 In the Virtual machine window that is running Ubuntu you will notice some icons at the bottom right



Your action: right click on the second icon from the left (the CD) and select <Unmount CD/DVD device> from the menu that pops up.

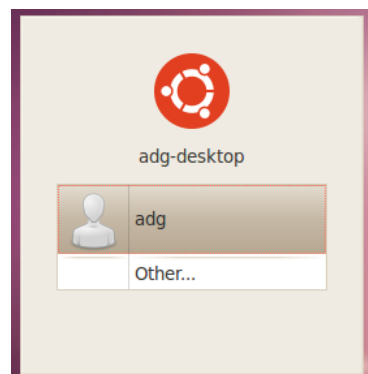


- 12 With the CD/DVD unmounted



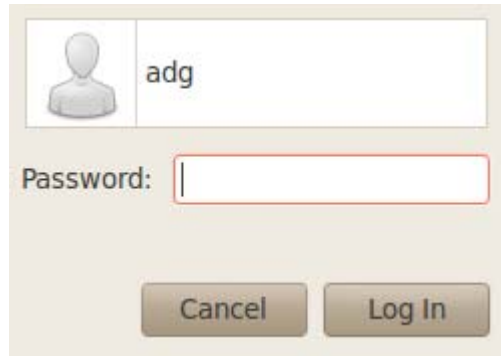
Your action: Press enter with the focus being on the Virtual Machine window

- 13 Ubuntu restarts and the select user screen appears



Your action: click on your username

## 14 Login

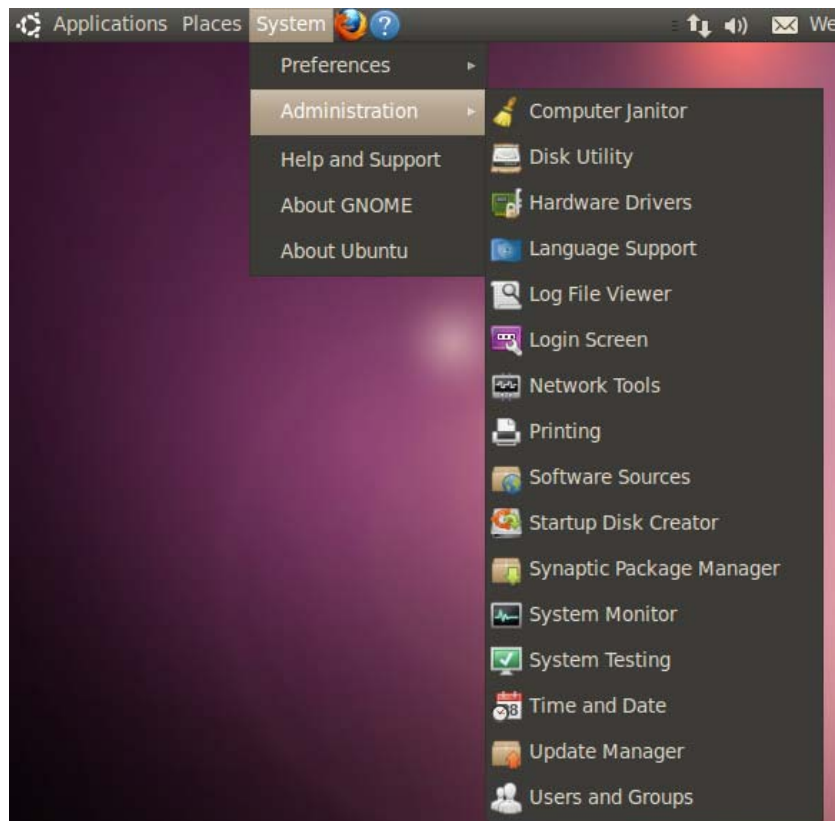


Your action enter the password you previously selected and click login to start using your new Ubuntu installation

## Installing <g++>

The g++ compiler is not installed with the default configuration of Ubuntu. This is why we have to configure our Ubuntu installation to include g++. The following describes how

- 1 Drop down the <System> menu.  
  
Go the the <Administration> submenu

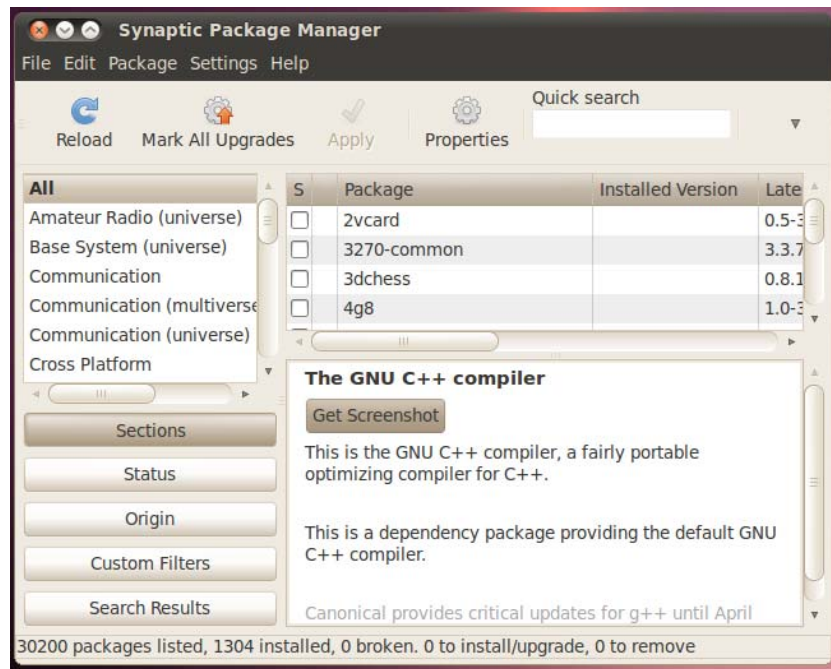


Your action: Click on <Synaptic Package Manager>.

Note: you will be asked for your password

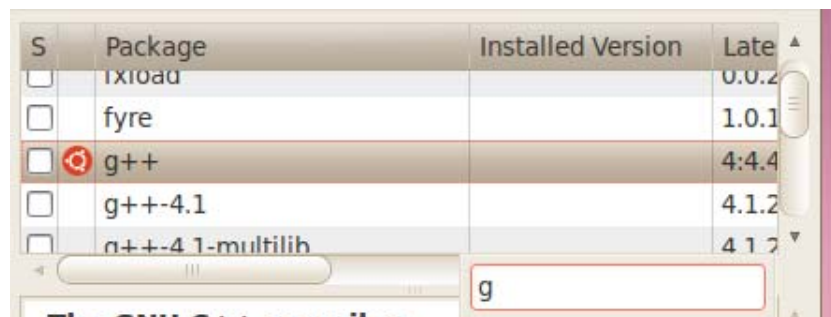
- 2 The window appears and you must now select the g++ compiler.

Notice the listing on the top right; the one starting with <2vcard>



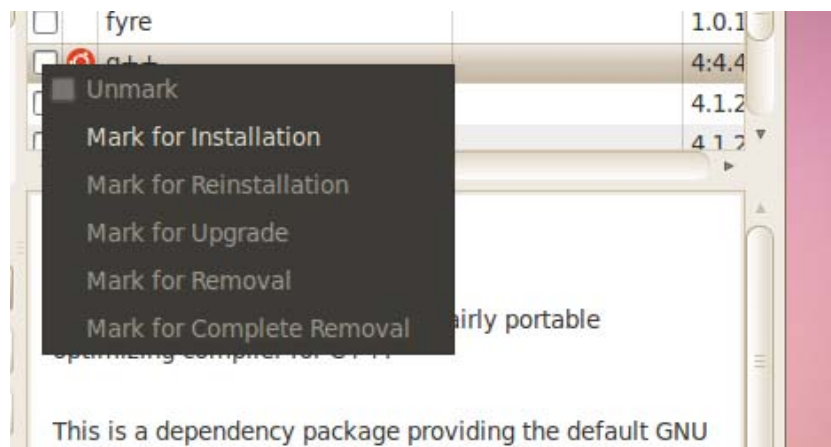
Your action: Click on <2vcard> and on your keyboard press the letter <g>

- 3 The list updates to your 'selection'



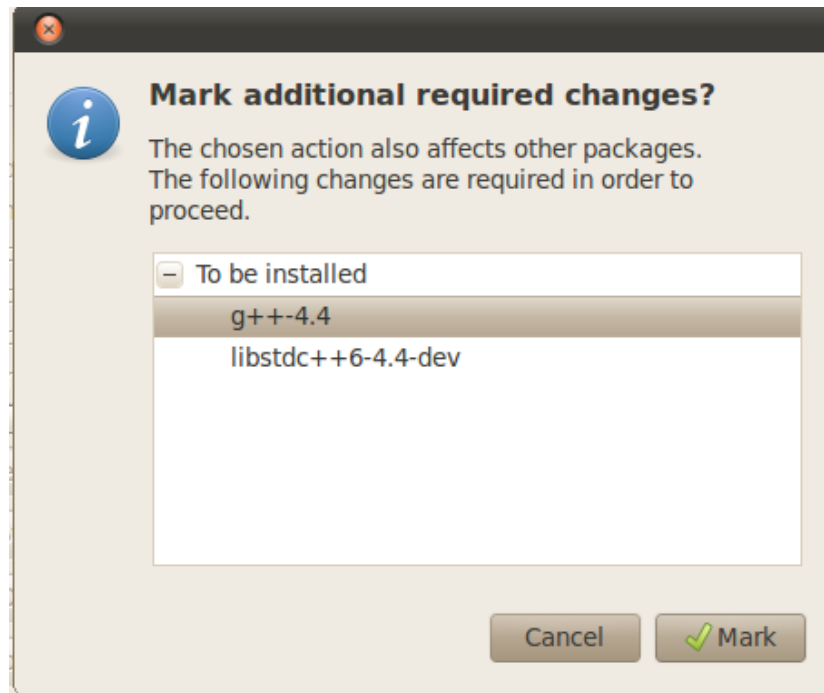
Your action: Click on g++

- 4 The pop up appears



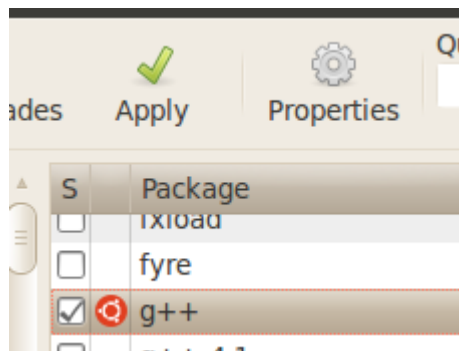
Your action: Select <Mark for Installation>

- 5 Ubuntu confirms



Your action: Click Mark

- 6 The package manager updates. Just apply the changes now.

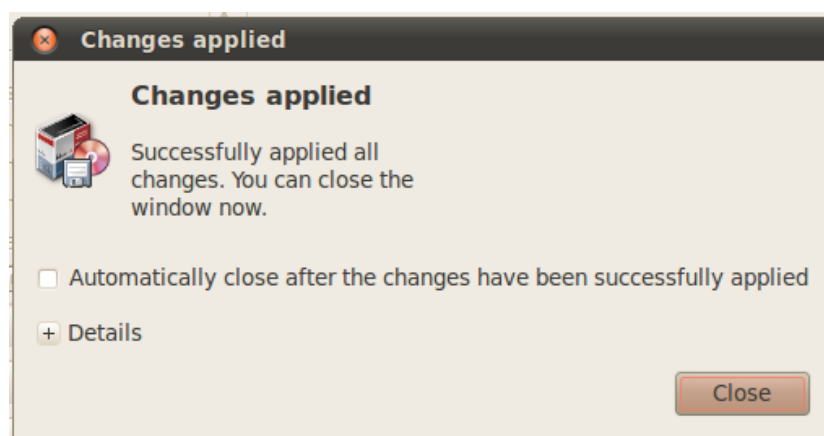


Your action: click the <Apply> button to install

- 7 The manager will download and install the package for you.

All done

You are now ready to starting programming in your Ubuntu installation.



Your action: just click close

## Using <gedit> and <g++>

In this brief section I'll let you know how to write your programs inside Ubuntu using <gedit> and how to then compile and run them using <g++>

Here is the typical first program for C++. Never mind what everything means (for now). What you need to know is that this program will output onto the screen the message "Hello World".

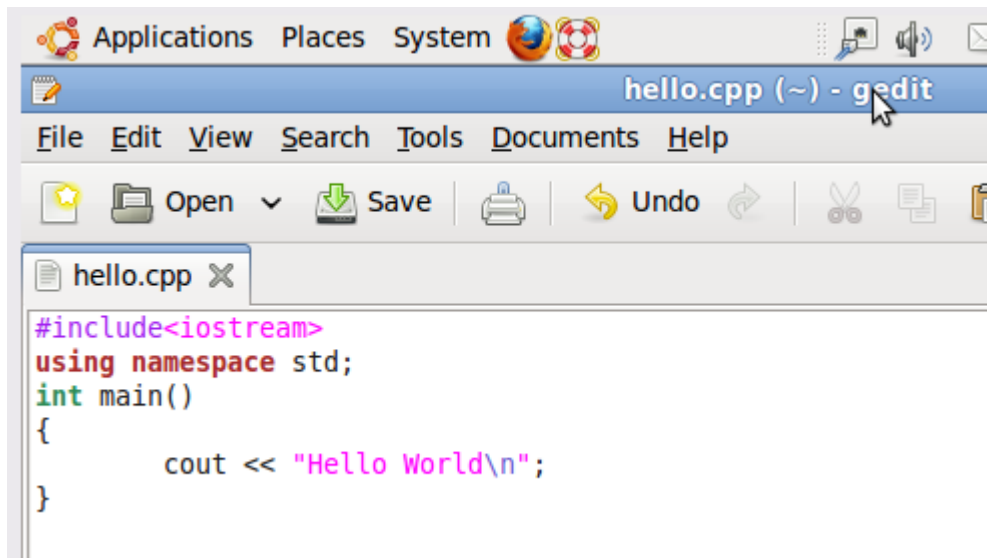
```
#include<iostream>
using namespace std;

void main()
{
    cout << "Hello World\n";
}
```

### ***Creating the first program***

Boot up your Ubuntu and login. Once up and running, pull down the <Applications> menu and go to the <Accessories> submenu. Select the <gedit Text Editor> item. The application will start and a new document will automatically be created.

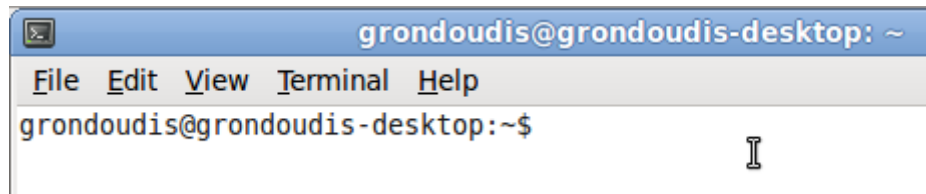
Enter the above text in there and save the file as <hello.cpp> (without the < and >). You should now have something like the capture below



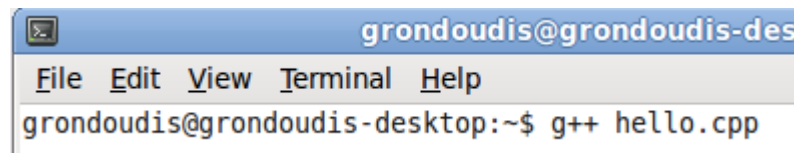
This file is now ready and it should be in your home directory.

### ***Compiling and running***

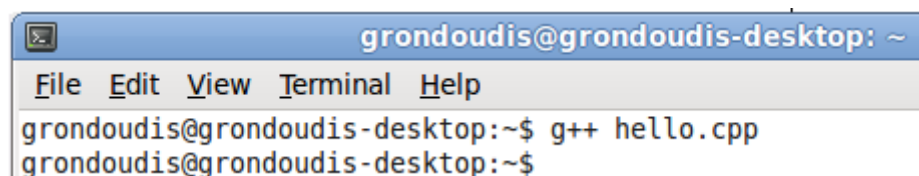
Having created the program we can now compile and run it. Pull down the <Applications> menu and go to the <Accessories> submenu. Select the <Terminal> item. An Ubuntu command prompt will open up and you are presented with something like the capture below:

A terminal window titled "grondoudis@grondoudis-desktop: ~" with a menu bar containing "File", "Edit", "View", "Terminal", and "Help". The command prompt shows "grondoudis@grondoudis-desktop:~\$" with a cursor at the end.

On the command prompt write `g++ hello.cpp` as shown below

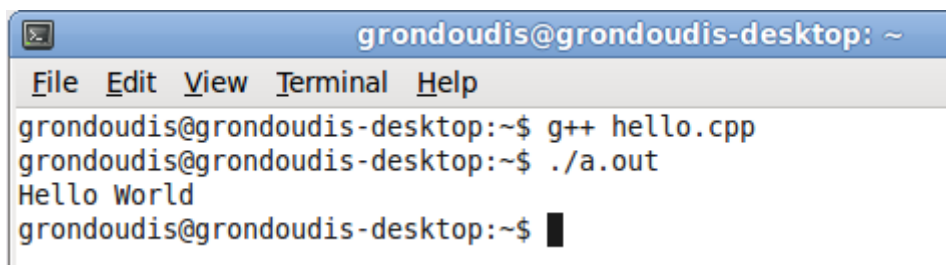
A terminal window titled "grondoudis@grondoudis-des" with a menu bar containing "File", "Edit", "View", "Terminal", and "Help". The command prompt shows "grondoudis@grondoudis-desktop:~\$ g++ hello.cpp" with a cursor at the end.

Press enter and (provided no typing mistakes were made) you should have the prompt return (just like below)

A terminal window titled "grondoudis@grondoudis-desktop: ~" with a menu bar containing "File", "Edit", "View", "Terminal", and "Help". The command prompt shows "grondoudis@grondoudis-desktop:~\$ g++ hello.cpp" followed by a new line with "grondoudis@grondoudis-desktop:~\$".

This means that the compilation was successful and the executable version of your program is ready. Ubuntu (like linux, unix and zenix) create the executable in the form of a file called a.out. All you have to do is execute that file.

The capture below shows the execution command as well as the expected output.

A terminal window titled "grondoudis@grondoudis-desktop: ~" with a menu bar containing "File", "Edit", "View", "Terminal", and "Help". The command prompt shows "grondoudis@grondoudis-desktop:~\$ g++ hello.cpp", followed by "grondoudis@grondoudis-desktop:~\$ ./a.out", and then the output "Hello World". The prompt returns to "grondoudis@grondoudis-desktop:~\$".

Note that I didn't just write `a.out` but rather `./a.out` and this has been done in order to let the system know that I would like to run the program in my home directory.

That is it. You have just edited, compiled and run your first C++ program in Ubuntu.

## Programming tips

1. You have to be patient and persistent. Programs can take a while to complete (compile and run). Don't give up
2. The compiler has nothing against you, as do the keyboard, monitor and case. They are just doing their job.
3. C++ is case sensitive. This means that 'String' is not the same as 'string' and 'cout' is not the same as 'Cout'
4. The compiler is very rarely wrong. If it is reporting a mistake then it means that you do have a mistake somewhere (see point 6 below). In time you will learn to recognize the text for reporting a mistake and you will know what needs correcting.
5. When you do get 10 or even 100 scrolls worth of mistakes don't despair. Usually mistakes at the beginning of the program can cause many mistakes later on. What you should do is scroll-back at the top of the mistakes window and try and correct the first few.
6. Quite often mistakes are reported on lines that when you see them no mistake exists. This, 99% of the times, means that there is mistake somewhere above in the program. If the code is in a loop somewhere; look at the loop first; then look and the method that the loop was in, and so on. Try and double-check related code (close and sometimes far) to the mistake reporting line.
7. When copying the same code in 2 or 3 places and you need to make minor adjustments; don't copy everywhere first and then go and do the adjustments. Make it so that you: make a copy-make the adjustment; make the next copy-make the adjustment and so on. This way, you will not forget to make an adjustment.
8. When starting a method or loop or anything that uses curly brackets ( { and } that is) don't write just the opening brace and then start writing the code inside hoping to remember the closing bracket. Instead, write both the opening and closing brackets at the same time and then take your cursor back in the middle of the two to complete the code. This way, anything you write in there (no matter how many lines it is) will be inside the braces.