Lab Manual

Operating Systems

Lecturers: Padraig Moran, Jacqueline Humphries

Contents

[Week 1 – Installing Ubuntu 3](#_Toc504751168)

[Week 2 - 4](#_Toc504751169)

# Week 1 – Installing Ubuntu

This week I learned that the Ubuntu Operating System is a Linux-based. I learned that Ubuntu is an open-source operating system, that not only provides the ‘Desktop’ operating system, but also a ‘Server’ operating system, which is also open-source.

I discovered that there are two versions of the ‘Desktop’ Operating System available for Ubuntu; the Long Term Support and then the latest development build. The Long Term Support (LTS) is a stable version, whereas the other is the unstable version.

I learned that VirtualBox is a Type-2 Hypervisor, or Virtual Machine Monitor (VMM). This is a piece of software that allows us to create, and manage Virtual Machines. VirtualBox is also the only, professional solution that is freely available to the general public.

I discovered that in a type-2 hypervisor, there is a host, and a client. The host will be the operating system being used to run the hypervisor, and the client will be the operating system(s) on the virtual machine(s) installed on the hypervisor. In our case the Host OS was Windows 10, and the client was Ubuntu 16.04.03 LTS (Long-term Support)

I learned how to create a virtual machine in VirtualBox. In this Lab’s case it was the Ubuntu operating system that I installed onto the virtual machine. Following the Lab notes, I learned how to allocate memory, primary and secondary, to the virtual machine upon the install.

I ascertained how to change the amount of memory being allocated, or other things such as: change the amount of processors to use, change the boot order to the virtual machine, adding secondary storage like a HDD, or after it had been installed; by shutting the VM down, and opening it’s settings.

I learned how to install the VirtualBox Guest Additions onto the Ubuntu client, which installs drivers and system applications that optimize it for better performance and usability.

# Week 2 – Using VI Text Editor

This week I learned about the vi editor, which is a file editor built into most UNIX systems. This means that becoming fluent in using it will allow you to work on many different flavors of the UNIX systems. This includes server system which don’t use a Graphical User Interface (GUI), but instead rely solely on a Command-Line Interface (CLI) as the GUI isn’t necessary and takes too much space.

In the lab, I discovered that the vi editor can create files that can be written to simply by calling the vi <new filename> command into the terminal in Ubuntu. This creates a new file with whatever name it is supplied with if the file does not currently exist in the directory. If the file does exist, it will open that file for editing.

The vi editor has two different modes, the first of which is the Insert mode. This mode allows us to enter text into the file. To get to the Insert mode, I learned that you need to press the ‘i’ key, then once you’ve entered the text and finished inserting, you exit the insert mode by hitting ESC.

If we are not in insert mode, then we are in command mode. This mode, I discovered, allows us to enter in commands to edit the document. To edit the document, all that’s needed is a few commands, which can be entered in command mode.

The commands I learned are outlined in the lab instructions; moving around the document, searches, changing words, replacing letters, deleteing letters, deleting multiple lines, copying and moving lines, adding lines, and joining lines. Then I also discovered how to save and quit the file, and subsequently, the vi editor.

Going on my own initiative I discovered how to quit the editor without making any changes to the file by using :q!