Akintomide Akinsola

CNA LABORATORY ENHANCEMENT BY VIRTUALISATION

Bachelor's thesis

CENTRIA UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Information Technology

June 2015

ABSTRACT

Unit	Date	Author/s
Kokkola-Pietarsaari	June 2015	Akintomide Akinsola

Degree programme
Information Technology

Name of thesis
CNA LABORATORY ENHANCEMENT BY VIRTUALISATION

Instructor	Pages
	28 + 3

Supervisor

The role of the Cisco Networking Academy of Centria University of Applied Sciences in the Media and Communication Technology specialisation is an essential one. An improvement in the infrastructure of the CNA laboratory directly leads to an improvement in the quality of education received in the laboratory. This thesis work described the creation of an alternative arrangement using Linux Ubuntu as a supplementary option in the studying of networking subjects in the CNA laboratory of the university.

Linux Ubuntu is a free software available to download and with some adjustments and modifications it can be made to function just as properly as a Windows operating system in the experience of learning networking. The process of creating and deploying a customised Ubuntu image deployed via the Virtual Machine, was discussed in this thesis work. Linux is a UNIX-like software, the knowledge of which is valuable to students as they study to become professionals. An introduction to several applications such as Minicom, Wireshark, and Nmap was also discussed in this thesis. A simple laboratory experiment was designed to test the performance and functioning of the newly created system. Expertise in more than one operating system expands the horizon of future students of the UAS, an opportunity older students did not have, and capacitates them to become proficient engineers.

Key words
Cisco, distribution, Linux, image, operating system, Ubuntu, Unix, Virtual Machine Monitor, virtualisation, VMware, Windows.

TABLE OF CONTENTS

ABBREVIATIONS

CCNA	Cisco Certified Networking Associate
CNA	Cisco Networking Academy
OS	Operating System
IP	Internet Protocol
NIC	Network Interface Card
NAT	Network Address Translation
VMM	Virtual Machine Monitor
LAN	Local Area Network
UCK	Ubuntu Customisation Kit
UAS	University of Applied Sciences
Nmap	Network Mapper
CPU	Central Processing Unit
ssh	Secure Shell

1 INTRODUCTION

Centria University of Applied Sciences is a citadel of learning, with the main campus situated in Kokkola; Central Ostrobothnia, four hours from the Finnish capital city of Helsinki. A UAS that offers students the opportunity to study any of three major degree programmes namely; Bachelor of Management in International Business, Bachelor of Engineering in Chemistry and Technology, Bachelor of Engineering in Information Technology at the campus on Talonpojankatu 2, Kokkola. All these are full-fledged degree programmes that offer advanced courses in the fields highlighted above for thoroughbred graduates.

The Degree Programme in Information Technology offers two broad specialisations namely; Software Engineering, and Media and Communication Technology. The Media and Communication Technology specialisation is optimised to provide competence in the understanding, design and operation of computer networks, from small-scale to medium-scale. To this end, the Cisco Networking Academy is an integral aspect of the degree programme. It is of necessity to have a Cisco Networking Academy laboratory to successfully be considered competent to offer Cisco courses and the university is equipped with a standard laboratory.

The degree programme in information technology is designed, amongst other things, to provide students with the full knowledge and skill necessary to partake in the Cisco CCNA certification examination. The academic module of the CCNA bundle is offered currently by the university. CCNA is a second-tier Cisco career certification required by industries to indicate the competence of a holder in the design, configuration, operation, and troubleshooting of a medium-sized routed and switched networks. This also includes the verification and implementation of connections to remote sites on a WAN. The preparation for this important fundamental certification and competence for network engineers is a requirement for students who wish to specialise in media and communication technology at the university so as gain proficiency in this field.

The use of standard Cisco-issued or Cisco-certified equipment is necessary to achieve adeptness in CCNA subjects, and the required laboratory equipment for the programme is defined by Cisco Systems. For this reason, schools and academies that offer the CCNA curriculum must provide their students with access to equipment installed preferably in their own local laboratory. Centria has fulfilled this requirement – all the customary equipment are available in the CNA laboratory on the school premises. In the CNA laboratory, there are routers, switches, and personal computers and all necessary support equipment installed to facilitate the appropriate teaching and learning of CCNA subjects. These equipment provide students with the opportunity to acquire hands-on skill with activities related to networking. Besides the online materials available for CNA courses, having sufficient experience and expertise in using proper equipment is necessary. The laboratory is designed to provide students with the chance to operate real Cisco devices and develop essential practical skills that are useful. The laboratory has 19 personal computers installed to cater for 18 students and a teacher. The lab recently had more support equipment installed to facilitate better operation and functioning.

During the period of research and study at the UAS, especially in the CNA laboratory, prevailing inadequacies of the present system were noticed. The system of learning the CNA course, as the NITS12K group experienced, employed, almost entirely, the use of Microsoft-based software. Ranging from the operating system, Windows 7, to the installed packages used for learning, such as Teraterm and PacketTracer, all run on the operating system. This situation presents a gap in the teaching and understanding of networking across many platforms. Student competence after undergoing the CNA laboratory module is limited to Windows-based packages. If there was an alternative system or a separate operating system, the group was not expressly exposed to it for unclear reasons.

In the age of information technology, it is recommended to have more than one operating system on a personal computer station. This offers the chance to become adept at approaching learning from multiple perspectives which in turn offers a richer knowledge base for participating students. The desktop computers in the CNA laboratory have a Virtual

Machine running a UNIX-like Linux operating system installed that can also be used to in teaching the CCNA course. However, the current system used in the CNA module does not mandate the use of this operating system, neither does it expose the importance of it – so much that many students do not in fact know that this Virtual Machine is available on the desktop computers.

The teaching of CNA modules involves giving instruction and offering assistance to students whenever and in any capacity that such is required. Offering assistance is an essential aspect of teaching. Presently, the teacher is required to be present at the terminal of the student needing technical assistance to offer such. This is not optimal. The UNIX-like system can be configured to also present the teacher with the option of assisting students directly from his desktop station without the need to approach each assistance-needing individual.

As presented earlier in this paper, the need for a viable and equally advanced alternative to the present Windows-based teaching system in the CNA laboratory is of importance. This research presents the process of installing a Linux operating system for the teaching of the CNA modules at the university. This will bring an improvement in the overall competence of students as they learn these modules.

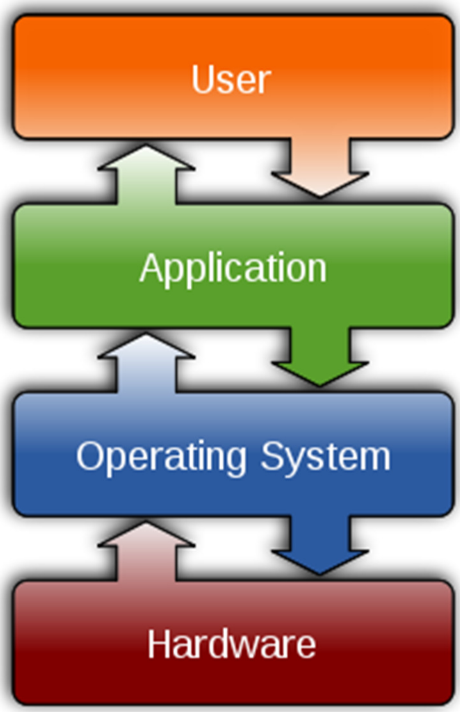
The possibility of offering teaching and learning assistance from a single designated desktop computer is required for the teacher. The possibility to ssh, troubleshoot, offer correction and monitor student activity via the Virtual Machine on the student's host is significant to the building of competence in the students. This new design with the system can present these options. This paper presents the configuration for the prospect of presenting new options for the students as they learn.

2 THEORETICAL FRAMEWORK

An operating system is a computer program that controls the execution of application programs, acts as an interface between applications and computer hardware, and provides common services for computer programs. Essentially therefore, an operating system is a computer program that governs, monitors and assists other computer programs in fulfilling their tasks by means of being the interface between these programs and computer hardware resources. (Stallings, 2005.)

2.1 Operating Systems

Typically, computer programs do not have express access to the hardware resources they require to carry out their designated functions, by design. Resources such as the memory, the processor and others are designedly out of reach of standard software. This lacuna is filled by the operating system as it connects the hardware wherewithal available on the host station to the need of the guest application as necessitated. It goes without saying that nowadays, without the operating system, the computer application is without any usefulness. From the perspective of the user, the operating system has the inherent objectives of convenience, efficiency and the ability to evolve. All these objectives serve to guild the designers of the operating system in their job to make operating systems even more efficient, useful and enduring. (Stallings, 2005.). Graph 1 shows the strategic location of the operating system in a typical computer system.



Graph 1. Basic computer system (Gesher, 2009.)

2.2 Operating Systems in Review

Before the advent of operating systems, programmers and computer users interacted directly with computer hardware resources. In that period, computer consisted mainly of moving mechanical parts and users interacted with these parts by means of other moving parts such as toggle switches and display lights. The main output device was the printer. This proved problematic and inefficacious as a way to utilise the new device as many bottlenecks such as scheduling troubles and setup time became rampant and exhausting. Furthermore, the cost of manufacturing a computer at that time was prohibitive, therefore, rapid progress needed to be recorded to make the computer a useful and ubiquitous device. Progress was recorded indeed – from simple batch systems to multiprogrammed batch systems, to time-sharing systems, rapid progress was made in the advancement of not only computer design but also computer architecture. This in turn led to increases efficiency and general usefulness of the computer system. (Stallings, 2005.)

2.3 Types of Operating Systems

Over the years, computer systems have evolved through strict processes of redesign and reinvention; from primitive, single-user single-task computers to multi-user multi-task personal computers. The evolution led to newer ways of conceptualising computers and their internal modes of operation. This was the beginning of the incubation of the idea of the operating system. The operating system introduced unprecedented ease to the functioning of computer systems. Such was the effectiveness that this mode of operation became widespread in the construction of computers thenceforth. The adoption of this style of computer architecture led in turn to the rapid development of the operating system itself; springing forth different types of it according to the separate methods of solving the same problem. Today, the most popular operating systems according to their names include but are not limited to: Windows operating system range; developed by Microsoft, Macintosh operating system range; developed by Apple Inc. and the UNIX operating system range; which is an open source operating system. The most important operating system of the UNIX abstraction is the Linux operating system. (Abzug, 2004.)

2.4 UNIX as an Alternative

Central to the goal of this thesis is the Linux operating system; which is, as stated earlier, an open source operating system. The UNIX operating system is the brainchild of the inventors at Bell laboratories and its very first instance became operational in 1970. The history of the UNIX is well known in computer academic circles so, it is needless therefore to go through it step by step. More specific to the theme of this thesis is the UNIX operating system itself and its possible deployment as an alternative to the Windows operating system for usage in the teaching of Cisco courses. Currently, the Windows operating system is being utilised in the CNA laboratory at the university such that every course in the laboratory is completed using this operating system and this is the case for the NITS12K class. If there are alternatives on the VMware present in the computers, the class did not experience this and therefore did not utilise it. This essentially limits the expertise of the students to Windows-based applications and software. An introduction of a different yet versatile operating system has the potential to deepen the proficiency of the students and expand their horizon. A UNIX or UNIX-like operating system will be an adequate alternative to the Windows platform for implementation in the CNA laboratory. (Stallings, 2005.)

2.5 Linux

Linux, an operating system, developed by Linus Torvalds fits the profile of an alternative, equally viable operating system. Linux is a free operating system software developed and maintained by enthusiasts all around the world as against the other types of operating systems mentioned earlier that are created and sold at a gain. The source code for Linux is freely available for download and it has given users the right to modify and adapt the software to their specifications. It has also mandated users to share whatever modifications they effect on the software with other users so as to keep the development of the software progressive, a plan/setup known as ‘copyleft’. Linux is renowned for its stability and efficiency in the utilisation of resources. The software also provides the choice between using a graphical user interface (GUI) and text-based user interface. The graphical user interface is less efficient given that resources have to be employed in the display more than absolutely necessary. Au contraire, the text-based user interface is very efficient in resource management allowing only functionality and fewer resources are dedicated to the user interface. Linux is likely to increase in importance as the years go by. (Abzug, 2004.)

2.6 Benefit of Alternative Operating Systems

The world is speedily changing as technology changes at an unprecedented rate. The more these advancements take foothold in the field of computer science and information technology, the more exigent the necessity of the understanding of the fundamentals governing the field. For students who desire to venture into the field of computer technology therefore, a vast foundational knowledge is of the essence. Nowadays, the knowledge and expertise in more than one operating system increases the chance of success as a networks professional significantly. Consequently, it is important for entrants into the field, particularly students, to be adroit at as many platforms as possible as they traverse the discipline in their peregrination to being inveterate.

3 VIRTUALISATION

In recent years, as technology has advanced rapidly, more sophisticated systems are required to keep up with everyday demands of the users of technology – especially internet technology. The suppliers of this facility therefore are obligated to provide a strong backbone in order to stay profitable, relevant and ahead of competition. For communication purposes today, companies have email servers, Web servers, FTP servers and so forth. This is a novel necessity to keep up with the demands of present day technology. For company A, all these servers may be run on separate machines in the same equipment rack, connected together by high-speed network. (Tanenbaum et al, 2015)

3.1 Introduction to Virtualisation

This setup is advantageous for company A for several reasons, amongst which are: the inability of one computer to physically handle the load of operating several systems and also the reliability is enhanced by using more than one computer; it may not be professionally recommended to install all the systems a company uses on one computer because the failure of that one computer leads to a total collapse of all the services of the company. Keeping different services on separate machines also serves as a security measure – even if an intruder attacks one machine, not all services are compromised as a result. This is also known as sandboxing. (Tanenbaum et al, 2015)

Furthermore, an organisation may wish to run their separate systems on different operating systems for professional reasons – as multiple operating systems can serve the same purposes and different operating systems are perceived as being better for different functions. This is also a reason company A may choose to run many machines, each performing seperate functions. This is an effective design, albeit an expensive one, in terms of resources necessary to setup and maintain such a facility. (Tanenbaum et al, 2015)

3.2 Virtual Machine Monitor

A viable solution to running multiple operating systems effectively and relatively cheaply is by the employment of the Virtual Machine Monitor, also known as a hypervisor. A Virtual Machine Monitor creates an avenue for multiple virtual computers to work on the same physical hardware by creating the environment necessary for a computer to function properly virtually. This is known as virtualisation. (Kivity, Kamay, Laor, Lublin & Liguori, 2007.)

By definition, virtualisation is a design structure that enables one computer to accommodate several virtual computers, each possibly running a different operating system. This structure has the advantage of being able to replicate the configuration of many physical computers running on multiple hardware on one computer running on a single and localised hardware setup. The failure of one virtual machine has no effect on the others, just like the setup discussed earlier in this chapter. However, it has the demerit of exposing the entire structure to vulnerability should something go wrong with the physical computer itself. This is not troublesome in real terms because it has been discovered that computer failure is not fundamentally due to hardware malfunctioning but poorly designed and implemented software, especially the operating system. In these systems, the Virtual Machine Monitor is the software running in the topmost privilege mode, so, this reduces the concerns about failure significantly because it is not a full operating system and therefore it is less prone to problems that arise with full-fledged operating systems. (Kivity et al, 2007.)

3.3 Virtualisation and Cloud Computing

Cloud computing has opened a brand new echelon of computing operations, making the process of hosting and utilising computer hardware capabilities less complicated. Cloud computing, in theory, refers to applications and services that run on a distributed network using virtualised resources and that are accessed by common internet protocols and networking standards. In other words, cloud computing is system of using hardware resources remotely via the internet to perform functions locally. Cloud computing has introduced a frontier shift in the technology of virtualisation. (Sosinsky, 2011.)

Before cloud computing and in many cases, those who required computing services would own these systems in their entirety. This was not only an expensive endeavour, it was also a technically and managerially complicated one. The growth of the internet in reach and robustness has facilitated the movement of services onto distant systems, which can then be accessed just as quickly and efficiently as though they were not remotely located. This is of advantage, especially to small and medium scale enterprises that do not possess the resources to purchase, install and maintain their own computing facilities locally. By implementing abstraction and virtualisation, cloud computing increases the availability of computer hardware resources to anyone who needs it. Nowadays, using virtualised machines in the cloud to run applications and even save and secure data is a common practice in the field of computer and internet technology. (Sosinsky, 2011.)

3.4 Support for Virtualisation

In 1960, IBM started experimenting with two independently developed Virtual Machine Monitors. One was named SIMMON and the other CP-40. Afterwards, CP-40, which was a research project, was re-implemented as CP-67 to form the central program of CP/CMS, a Virtual Machine operating system for IBM System/360 model 67. It was re-implemented and released as VM/370 for System/370 series in 1972. Whereas the hardware technology improved considerably, the fundamental architecture did not change for ease of backward compatibility. All these systems, including the z-series released in 2000, supported virtualisation before it became widespread with the x86, a group of backward compatible instruction set architectures based on Intel 80-series CPUs. In 1974, Gerald Popek and Robert Goldberg published ‘Formal Requirements for Virtualisable Third Generation Architecture’ in which they enumerated the precise requirements a computer architecture must fulfil to support virtualisation effectively. (Popek & Goldberg, 1974). In 1990, researchers developed a new Virtual Machine Monitor called Disco which eventually became VMware – a virtualisation apparatus which offers both type 1 and type 2 Virtual Machine Monitors, offered by VMware Inc. (Bugnion et al., 1997). In 1999, VMware presented its first virtualisation solution for x86. (Tanenbaum et al, 2015.)

3.5 Prerequisite for Virtualisation

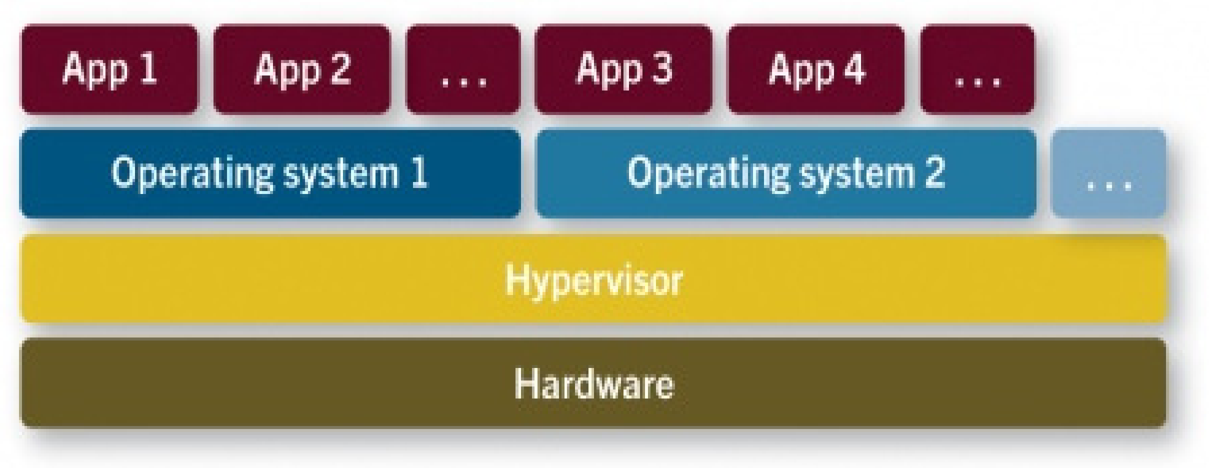
For a system based entirely on virtual machines, it is of utmost necessity that the virtual machine mimics a real machine completely. The necessity arises with the problem of installing and operating software. This mimicry must be effective and efficient in all respects. Accordingly, the virtual machine must perform satisfactorily in three aspects. The Virtual Machine Monitor must have complete hegemony over all system resources as a matter of safety. The operation of a software program on a virtual machine must also be similar in all aspects to the same software program working directly on real hardware resources, as a matter of fidelity. Software program must work without assistance by the Virtual Machine Monitor; this ensures that the efficiency of the system is maximised. (Tanenbaum et al, 2015.)

3.6 Types of Virtual Machine Monitors

In practise, there are two types of Virtual Machine Monitors and an introduction to both of them will be given in this work. They are characterised on the basis of technical design and architecture, which influences the mode of operation of the Virtual Machine Monitor. These influences cut across many paradigms, including the accessibility of system's hardware resources, safety, and the control of input and output devices amongst others. (Tanenbaum et al, 2015.)

3.6.1. Type 1 Virtual Machine Monitor

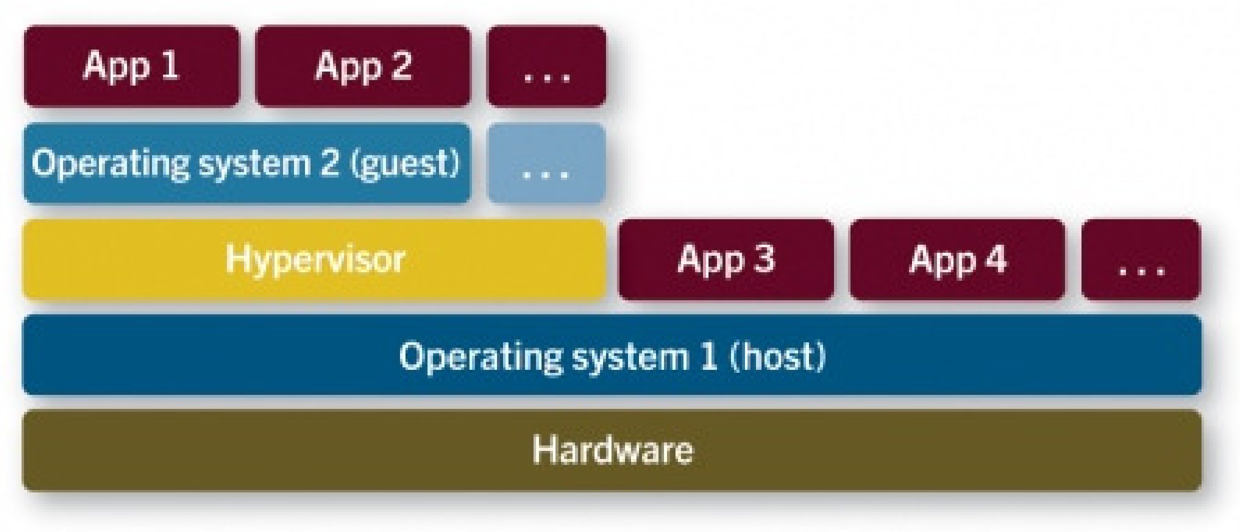
The distinction between the two types of Virtual Machine Monitors was observed by Goldberg (1972). Type 1 Virtual Machine Monitor acts like a typical operating system in many ways. This is because it is the only software program working in the highest privilege mode on the computer hardware resources. It works to provide assistance for multiple instances of the real hardware called the virtual machines which is essentially identical to what a genuine operating system does. In other words, it acts like an operating system while not being one. (Tanenbaum et al, 2015.) Graph 2 below explains the design of the type 1 Virtual Machine Monitor.



Graph 2. Type 1 Virtual Machine Monitor (Microsoft 2015).

3.6.2 Type 2 Virtual Machine Monitor

Contrary to type 1 Virtual Machine Monitor, type 2 Virtual Machine Monitor requires an operating system to function. The operating system, such as Windows or Linux, needs to already be installed on the hardware before the type 2 Virtual Machine Monitor is installed on it. This type of Virtual Machine Monitor requires the background operating system to allocate and schedule resources – typical of most computer programs. The operating system working on the hardware directly is called the host operating system. The VMware Workstation was the first instance of a type 2 Virtual Machine Monitor. (Bugnion et al., 2012). Graph 3 shows the structure of the type 2 Virtual Machine Monitor. (Tanenbaum et al, 2015.)



Graph 3. Type 2 Virtual Machine Monitor (Microsoft 2015).

3.7 VMware Workstation

VMware Workstation was the first virtualisation product for 32bit x86 computers. The introduction of VMware Workstation has had an undeniable impression on the field of computer science. For the technical details of the VMware, a perusal of the technical article published by Bugnion et al., 2012, is necessary. Important to the idea of the VMware was the thought that a system primarily built to run a particular operating system can be expanded to work with another operating system that is of a different type. This helps to solve the limitations of present systems such as application interoperability, operating system migration, reliability, and security. Additionally, the cohabitation of more than one operating system would be made feasible by the VMware. The VMware faced tough challenges in coming to the x86 operating system environment. This was the case because, nowadays, the job of building and manufacturing computer systems has been divided into at least four different units. A typical computer has four distinct groups of components, manufactured or assembled by separate companies namely: processors, made by Intel or AMD; operating system, made by Microsoft, Apple or the Linux Foundation; I/O devices and peripherals, made by other companies; and the hardware itself, assembled by HP, Lenovo and Acer, amongst others. As a direct consequence, VMware faced a structural challenge which it only could overcome by combining well-known virtualisation solutions, solutions from other domains, and new solutions into a new result. (Tanenbaum, 2015.)

3.8 The Evolution of VMware Workstation

Since the introduction of VMware Virtual Machine Monitor, the frontier of technology has been expanded. The hosted architecture is still used today for state-of-the-art interactive Virtual Machine Monitors such as VMware Workstation, VMware Player, and VMware Fusion (Barr et al., 2010). The ability to clearly separate the host operating system and the virtual machine context is the foundational mechanism of VMware's hosted solutions. The emergence of hardware support for virtualisation has helped improve VMware's guest operating system-centric strategy. The most current versions emphasise on performance optimisation for guest operating systems. (VMware, 2011.)

4 UBUNTU OPERATING SYSTEM DISTRIBUTION

An operating system comprising of a compilation of software on top of a Linux kernel is called a Linux distribution or 'distro'. A package manager is also usually compiled with a Linux distribution. A package manager is a set of tools that automates the procedure of the configuration, installation, upgrading and removal of software from a particular operating system. A typical Linux distribution comprises a Linux kernel, GNU tools and libraries, additional software, documentation, X window system, a window manager and a desktop environment. Most of the included software is free and open-source software made available both as compiled binaries and in source code form, allowing modifications to the original software. Almost all Linux distributions are open source but a notable exception is the Android. The Android has no command-line interface and software made for typical Linux distributions. (Corbet, 2011.)

The Linux Ubuntu, as stated in section 2.5 of the chapter three of this thesis, is a free software. The notion of free software causes confusions amongst novices in the computer science discipline. In the field of computer science and technology, the concept of free can be explained in two distinct ways, using the following analogies, the two ways may be clearly understood. The explanation given next sheds light on the correct conceptualisation of the manner in which the Linux software is free.

4.1 The Free Beer Concept

The action of getting a beer for free may mean that the person who does the actual drinking of the beer does not have to or need to pay the price for the beer but someone else does. In any case, the beer does not come without a cost, the cost just is not charged to the person who drinks it, it is charged to someone else – there is an exchange of money involved. In this analogy, the beer is a parallel for a software package. (Byfield, 2014.)

4.2 The Free Speech Concept

In many climes, human beings are entitled to free speech as a right. No one pays for it. No resources, either financial or otherwise is exchanged to allow for free speech, it is just bestowed as a human right. In this analogy, the parallel to software is speech. Free software is given, with all the privileges, and no exchange is required. (Byfield, 2014.)

4.3 What is Free Software

Free software is the type of software that a user can install, copy, distribute, study, modify and improve upon. Free software gives users the freedom to install and run the software as desired and for any specific purpose. Additionally, it gives the freedom to study how the software works and to modify it to suit desired requirements, which means the user has access to the source code as a precondition. The freedom to redistribute widely as desired to others, and the freedom to modify the software and to share those modifications to the public so that others can also profit from it is also included. Ubuntu is a free software in this way and therefore suitable for the purpose of this thesis. (Byfield, 2014.)

4.4 Licencing

Ubuntu software collection is preinstalled with thousands of programs that are sourced and created by many individuals, groups and/or organisations. These programs individually are under separate licenses according to the licencing policy in use by the Ubuntu organisation. Licencing policy describes the procedure utilised in deciding whether or not a piece of software created will be included in the Ubuntu operating system by default. The four main components of the Ubuntu licencing policy are: main, restricted, universe and multiverse. (Ubuntu, 2015.)

4.4.1 'Main' Component Licencing Policy

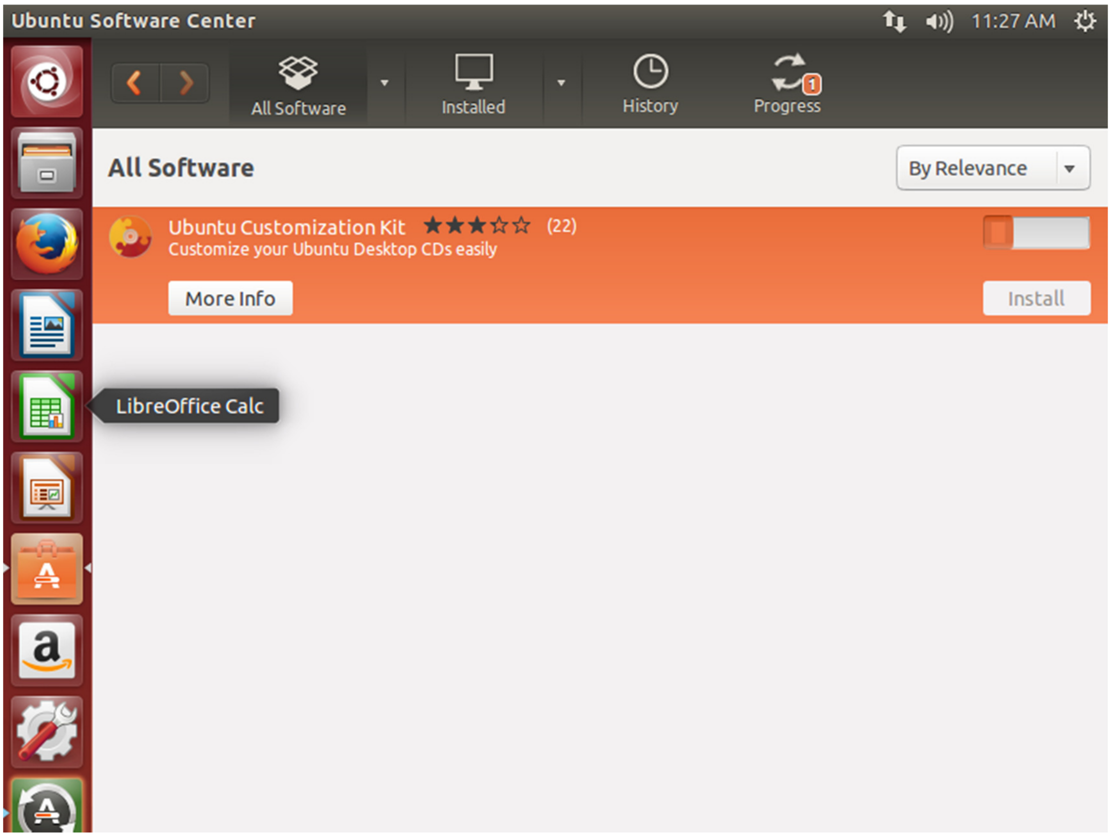
All application software available under this policy must include the source code. This component has a strict and non-negotiable requirement that application software included in it must include the full source code. Additionally, all application software under this policy must allow modification and distribution of modified copies under the same licence. Having the source code alone does not come with the freedom of having the right to modify it to suit a specific purpose. Without this permission to modify the source code, Ubuntu community cannot support, translate and improve software or fix bugs. (Ubuntu, 2015.)

4.4.2 'Main' and 'Restricted' Component Licencing Policy

All application software belonging to this category must allow redistribution of software exactly as downloaded or modified. Also, all software must not require royalty payments or any other fee for redistribution of modifications. All these rights must also be allowed to pass on along with the application software. The software must also be devoid of discrimination against any group, person, and field of human endeavour. Additionally, all application software must be distributed under a licence specific to Ubuntu and not separately. Furthermore, the software application must not contaminate, destroy or otherwise cause harm to any other software licences. All these are the requirements for the 'main' and 'restricted' component licencing policy. (Ubuntu, 2015)

4.5 Creating a Customised Ubuntu Distribution

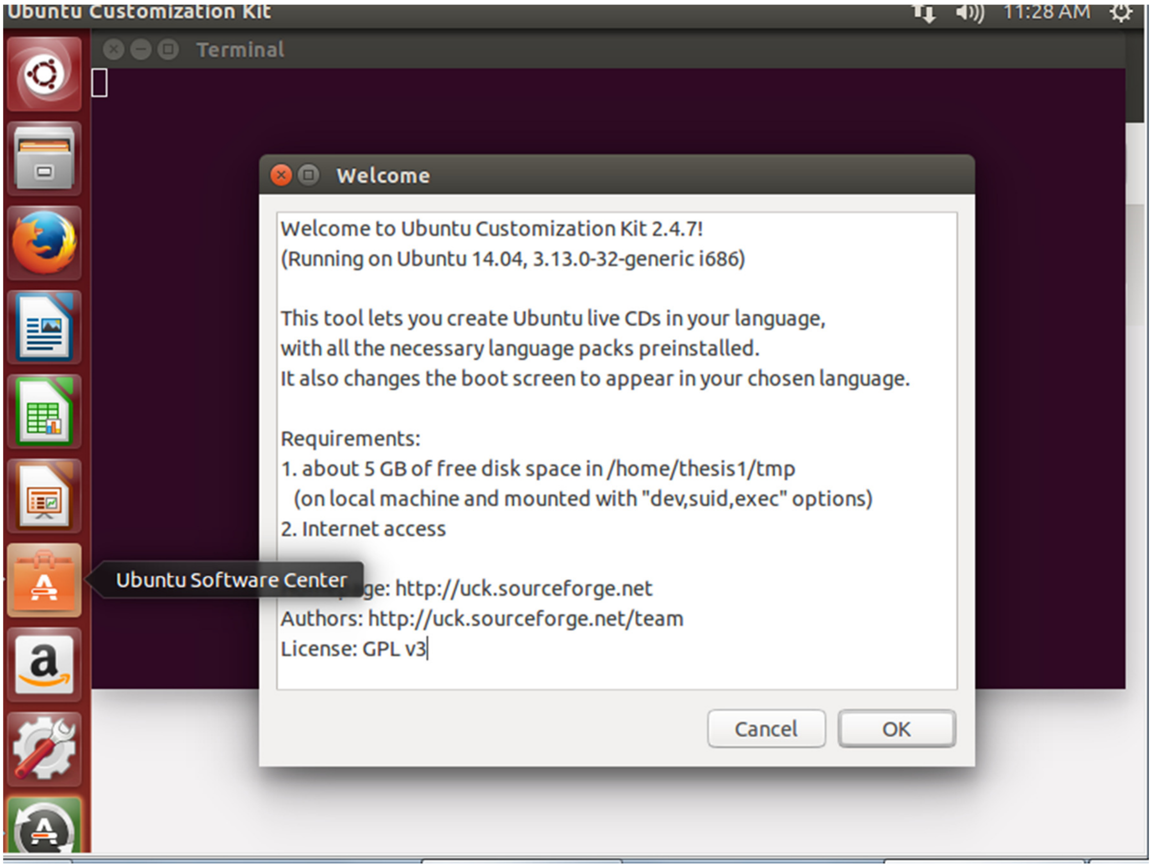
For this thesis work, an Ubuntu LiveCD was necessary and it was created using the Ubuntu Customisation Kit (UCK). When the Ubuntu operating system is needed for a specific purpose and function, it needs to be modified and customised to suit that specific purpose. This is one of the fundamental differences between Linux and other operating systems – the capacity to add and remove unneeded applications thereby streamlining the operating system to suit the desired purpose. The creation of the customised Ubuntu distribution utilised for this thesis will be explained in detail subsequently. Graph 4 shows the Ubuntu UCK while being installed from the Ubuntu Software Centre.



Graph 4. Ubuntu Customisation Kit installation

4.6 Ubuntu Customisation Kit

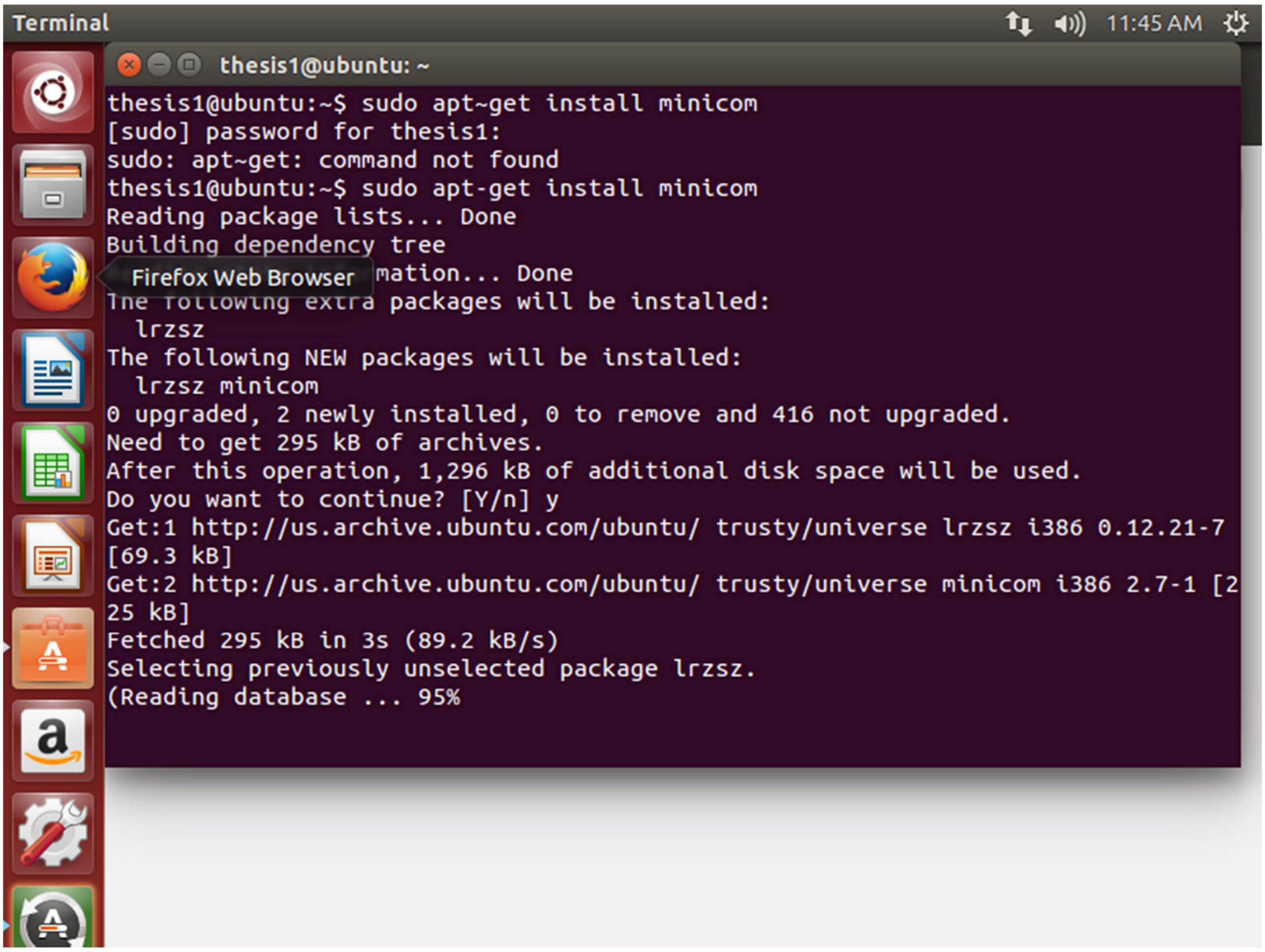
To test the thesis, a new and specialised Ubuntu distribution was needed. The easiest way to create a new Ubuntu distribution was by using the Ubuntu Customisation Kit (UCK). The necessity of the customisation of the Ubuntu distribution arose from the realisation that only a few of the preinstalled applications in the original, freely downloaded software are required for the purpose of this work. Additionally, many applications required were not preinstalled in the software and had to be included. The Ubuntu Customisation Kit is a freely downloadable software from the Ubuntu Software Centre. After the customisation, the Ubuntu operating system becomes an Ubuntu distribution that can be shared with others as needed. Graph 5 shows the downloading of the UCK for this thesis and the start page for creating the Ubuntu LiveCD using the UCK.



Graph 5. Creating the LiveCD

4.7 Minicom

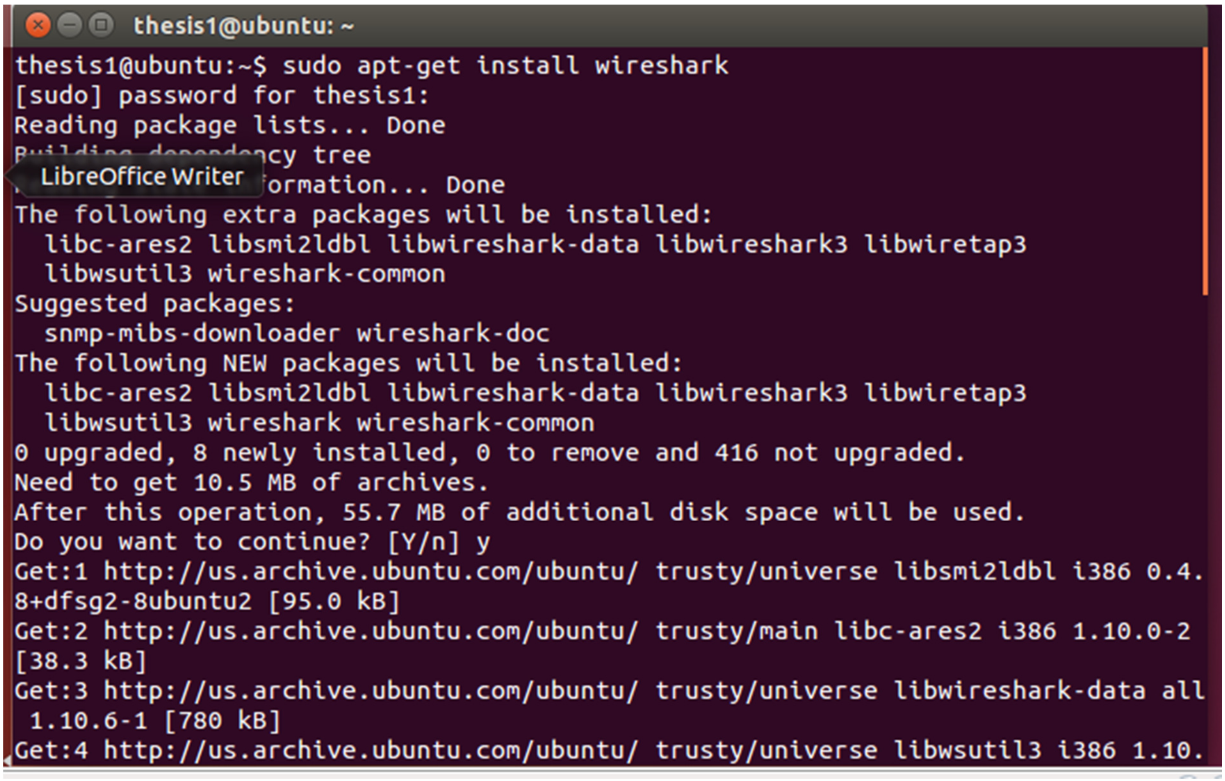
Minicom is a text-based terminal emulation program designed for UNIX operating systems by Miquel van Smoorenburg. Minicom is a menu-driven communications program and a free software. It has features that include, but are not limited to dialling directory and auto-redial and support for UUCP-style lock files on serial devices. It can be used to configure routers and other network devices. In the CNA laboratory, students need to configure devices and equipment such as routers, modems and switches as they study and learn. Minicom is a terminal emulation program that assists in the fulfilment of this goal. To install Minicom, the command shown in Graph 6 was used. The tool ‘apt-get’ is a tool included in the package manager supplied with the Debian package ‘apt’. APT retrieves, configures and installs the dependencies automatically. Hence, ‘apt-get’ is a tool used to update, upgrade, install and remove applications from an Ubuntu distro. (Linux, 2015.)



Graph 6. Installing Minicom

4.8 Wireshark

A packet analyser is a computer program that analyses networks by capturing moving packets and printing the analyses in a way that the computer networks’ administrator can read and understand. Wireshark is a type of packet analyser. One of the reasons Wireshark is successful is that it includes special effects such as colour coding that allows for a clearer understanding of printed results of inspected packets in network traffic. It is useful for student in the laboratory so as to enable them comprehend what transpires within a network. (Wireshark, 2015.). Graph 7 shows Wireshark being installed in the test Ubuntu.



Graph 7. Installing Wireshark

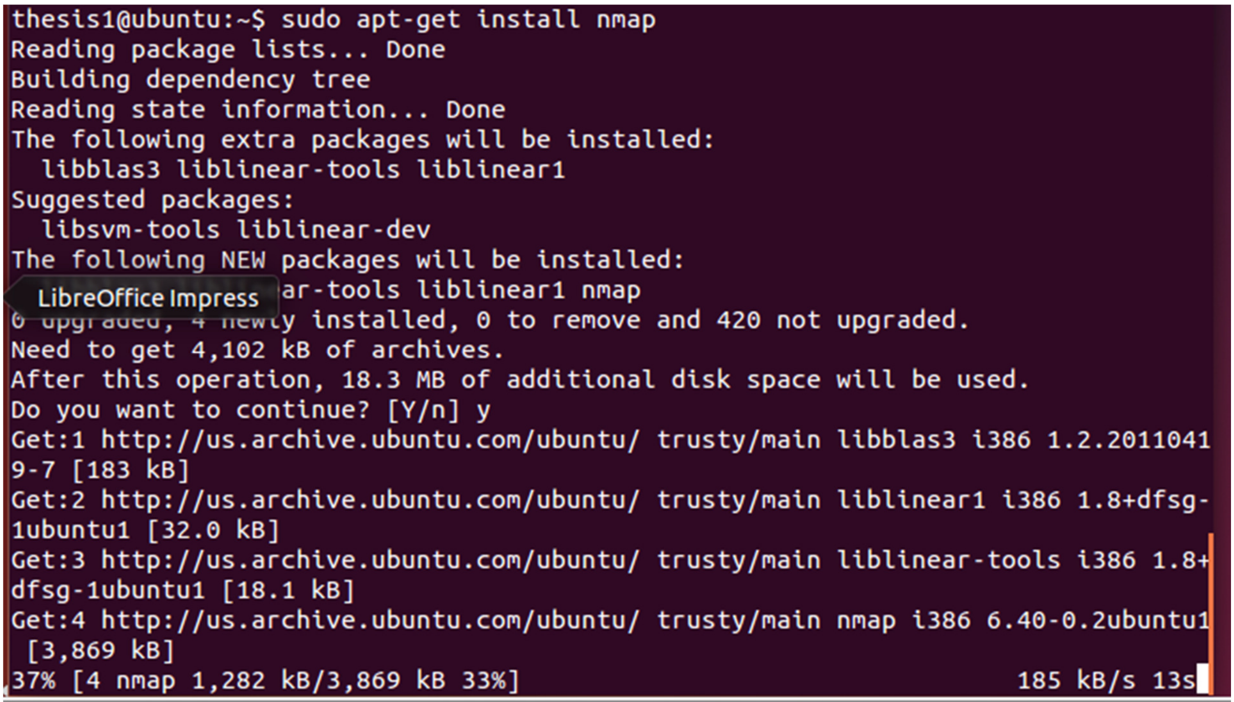
4.9 Secure Shell

Secure Shell is a protocol belonging to the application layer that allows for communication between two hosts on any network configuration in a secure manner. Commonly, any communication between two ports connected via a public network is precarious in that the packets are susceptible to interruption and discrete analyses by any intruder. In Secure Shell, this is prevented by the use of public key cryptography to validate the connection between the initialising port and the receiving port. In this scenario, confidentiality and integrity of data is guaranteed by the encryption of the connection. (Wembao, 2003.)

For students, it is necessary to learn and acquire a sense of the importance of information/data security. For this reason, the knowledge and usage of Secure Shell is an important part of their training. As against Telnet that communicates data in plain text exposing it to danger and risks, Secure Shell is encrypted. It has replaced Telnet in networking circles on the basis of this encryption. (Wembao, 2003.)

4.10 Network Mapper (Nmap)

This is an application for investigating networks and appraising security features. It examines IP packets in an effective way to review what hosts and end devices are connected to the network, what applications and services are interacting within the host and across the network, what operating systems are installed on the end devices, and it also analyses what kinds of firewalls are in use and their status. It is useful for administrator services such as registering the content of the network, monitoring hosts and network services, and scheduling updates. This tool is very important to the purpose of this thesis work because the teacher can use it to monitor network devices as the students study the CNA courses. Graph 8 shows the Nmap being installed in the test Ubuntu. (Linux, 2015.)



Graph 8. Installing Nmap

5. DEPLOYMENT AND TESTING

The whole process of creating the Ubuntu specialised operating system (also known as Ubuntu distribution or Ubuntu Distro) culminated in the transferring of the customised Ubuntu image files to the computer systems in the laboratory. The end and final process of this thesis process was the proper testing of the Ubuntu distribution setup, as installed on the hosts in the laboratory to ascertain the functioning of the new Ubuntu distribution in the appropriate conditions. All previous activity in the Ubuntu package had been on a remote computer so as to have avoided any corruption of the laboratory devices that might have occurred during the creation procedure.

5.1 Testing Parameters

For the purpose of this thesis, it was not only important to have created the distro, it was also important to test the final product in the laboratory for which it was developed. Strict testing scenarios had to be designed. These scenarios had to be designed to mimic proposed functionality to the best degree of eventual relevance. The testing was divided into two, for the two proposed users – the teacher and the student.

5.1.1 Parameters for the Student

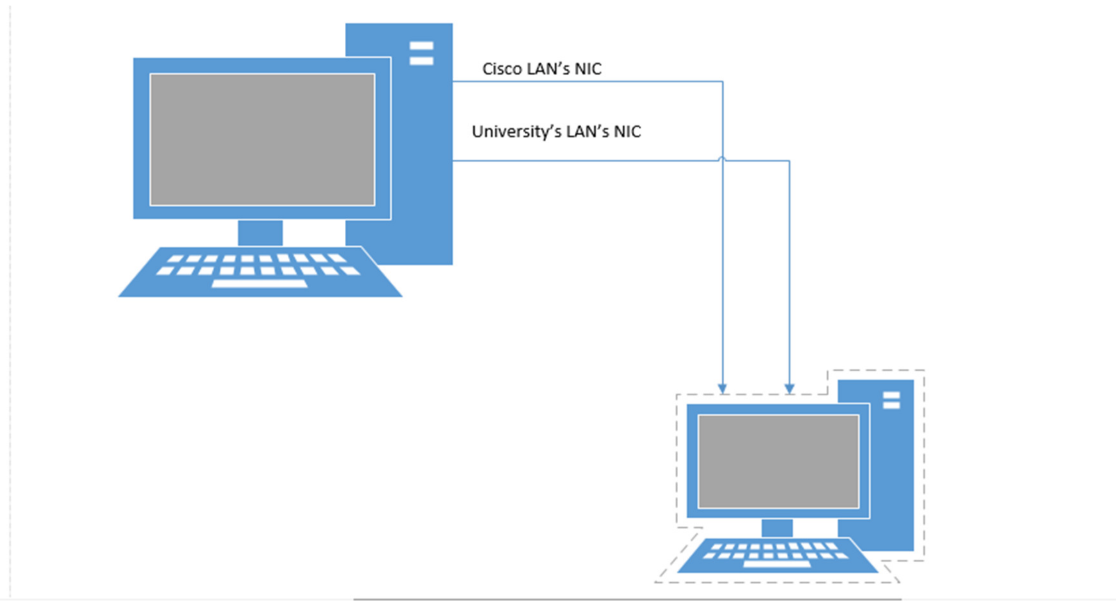
Before the system could be declared functional and ready for use, the student's host computer must ping the teacher's computer and vice versa. The student's host computer, in the necessary states, must configure Cisco devices as well. It is important that the student's system passes these tests.

5.1.2 Parameters for the Teacher

The functionality required for the teacher was the ability to connect via Secure Shell (ssh) to the student's host computer, to the students virtual machine and while in the virtual machine, telnet to the Cisco devices in the group configuration in the right conditions. Additionally, the teacher must run Nmap test on student's virtual machine while connected via ssh. If the teacher's host computer could accomplish this, then the operating system would be declared read for use. This is an important aspect of teaching the courses in the laboratory. Not only does it demonstrate to the students the capabilities of networks, it also introduces them to the possibilities of remote controlling devices and hosts.

5.2 Testing

To test the performance of the distribution under laboratory conditions, the image files were required to be transferred from the remote computer where the system was created to the computers in the laboratory. Subsequently, the new Ubuntu distribution had to be installed on the Virtual Machine Monitor. For this thesis work, the Ubuntu distribution was moved to the computers in the CNA laboratory and subsequently installed on the VMware already available on the host computers in the laboratory. This new Ubuntu distribution was installed on two computers; one for the teacher and one for the student. In the laboratory, all the machines have two Network Interface Cards (NICs); one for the university's Local Area Network (LAN) and the other one for the CNA laboratory as illustrated by Graph 9.



Graph 9. Two NICs on the Current System

After the installation, another network adapter was created for the VMware in addition to the one already available. The first network adapter was bridged and the physical address connection state was replicated. The second network adapter was placed on NAT, which enabled it to share the host's IP address.

5.2.1 Testing the Student's System.

As earlier stated in section 5.1.1 of this chapter, the student must be able to configure CNA devices using appropriate configurations before the system can be declared functional. The serial port application used was Minicom and its port configuration was according to Graph 10. After this setup, the configuration of the Cisco devices was as shown in the scenario depicted by Graph 11. The exact scripts may be found in the appendix 1 attached to the end of this thesis.

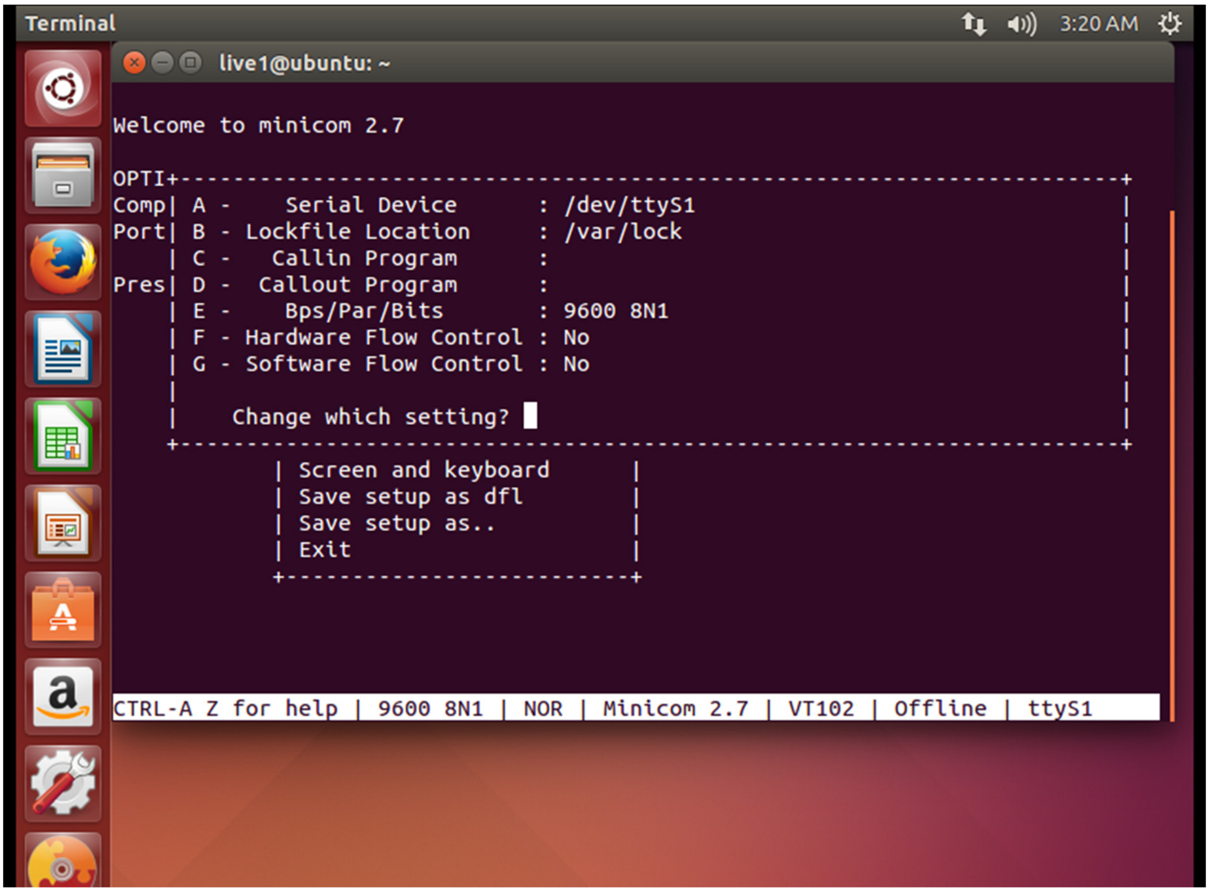
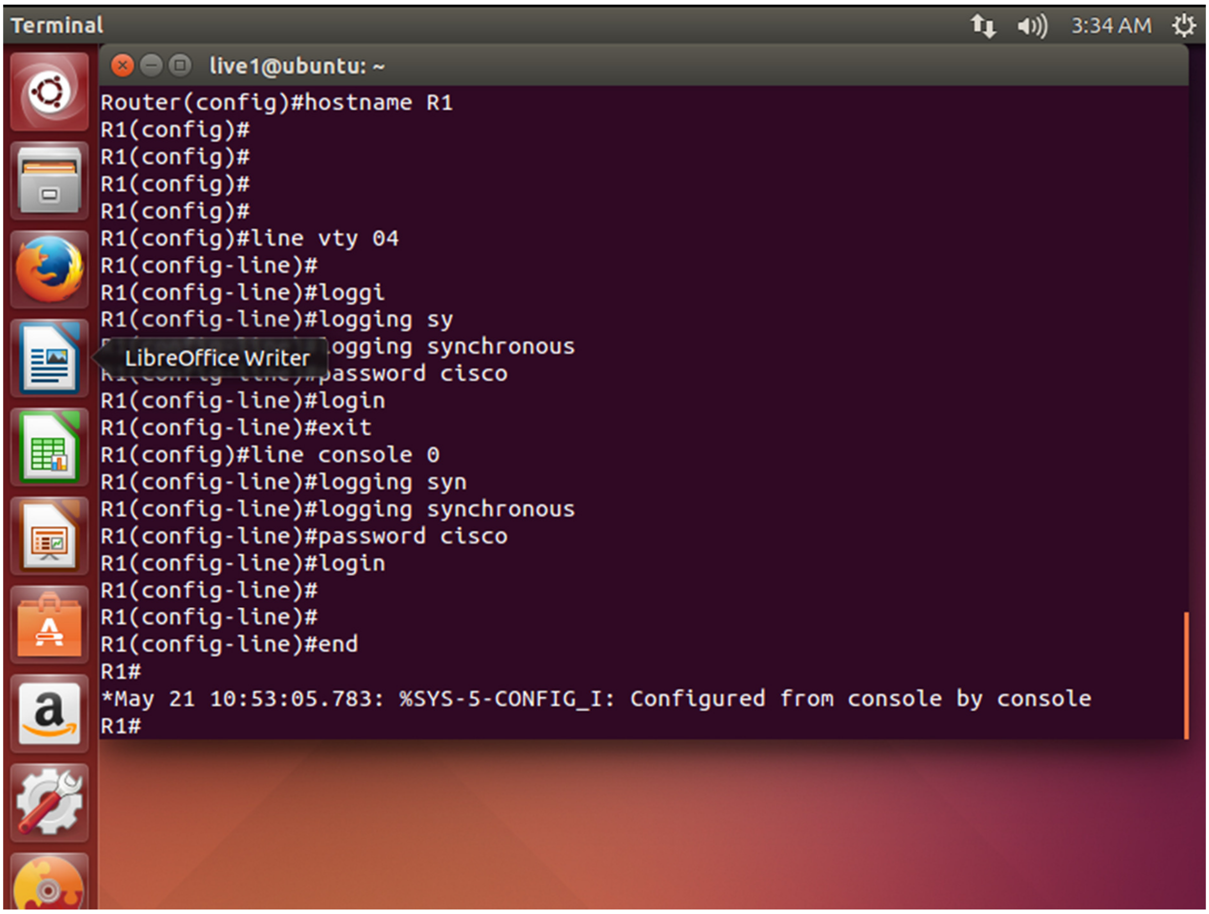


Fig 10. Port configuration on Minicom



Graph 11. Testing the Student's System

5.2.2 Testing the Teacher's System

The setup is that the teacher can ping and connect via ssh to the student's computer to provide assistance as it may be necessary during the laboratory classes. During the testing, pinging between the teacher's computer and the student's computer was successful. However, due to the present configuration in the school's network, an ssh connection could not be established between the two computers. Running an ssh connection on the school's networks is prohibited. Although the connection may be otherwise established using the CNA network if it is configured properly, but this is beyond the scope of this thesis work. The reason for which the teacher needs to establish an ssh connection with the student's host computer, in the first instance, is to assist the student in establishing the proper CNA laboratory network configuration, if such assistance is required. However, it was discovered during the course of this thesis work that the student who can properly configure group connections giving the appropriate scenario may not require the teacher’s remote assistance at all.

6. CONCLUSION AND RECOMMENDATION

As this thesis work fully implemented a secondary system using an Ubuntu distribution to the present Windows cofiguration, other students who will make use of the laboratory henceforth have the potential to explore and approach the CNA courses from a two distinct perspectives – the importance of this to their academic development cannot the overemphasised. Hitherto, the choice of a separate approach was summarily unavailable, that situation has been rectified. From now on, it will be possible, even advisable, to configure routers and switches, surf the internet, and do other studies related activities with a UNIX-like operating system and not rely solely on the Windows platform. However, this is not an end in itself. It is propose that more Linux-related subjects should be introduced and included in the UAS curriculum so that students who have identifiable or specific interest in this operating system can explore that interest. This is an alternative that was previously unavailable at Centria University of Applied Sciences. To bring about a change to this restrictive system was the reason for undergoing this project – to give future students a better chance at being seasoned professionals. This will also have positive effect on the quality of education, public perception, and ranking of the university.

The ssh connection between the teacher's computer and the student's computer, using the UAS' LAN, as explained in chapter 5 of this thesis, is currently prohibited by the university's network configuration. This led to the failure recorded while attempting to implement a telnet connection during the course of this thesis work. It is recommended, for future objectives, that the UAS' network arrangement be revised and reconfigured to enable ssh communication between computers with the appropriate permissions.

REFERENCES

Abzug, C., 2004. Linux Operating System. Hoboken, NJ. John Wiley and Sons Inc.

Adodo, S., 2011. CNA Laboratory Enrichment by Virtualisation. Available: http://theseus.fi/bitstream/handle/10024/29410/Segun%20Adodo.pdf?sequence=1 Accessed 10.11.2014

Bhor H., Rote U., Shinde U., 2013. Operating System. Available: https://www.coursehero.com/file/8273719/OS-PDF/ Accessed 8.5.2015

Centria University of Applied Sciences. 2015. Guide for Thesis Writers. Finland, Kokkola: Centria University of Applied Sciences.

Cisco Networking Study Module 2015. Available: https://www.netacad.com Accessed 01.05.2015.

Corbet, J., 2011. Android, Forking, and Control. Available: http://lwn.net/Articles/446297/ Accessed: 25.04.2015

Kivity, A., Kamay, Y., Laor, D., Lublin, U., Liguori, A., 2007. Proceeding of Linux Symposium. Available: http://www.linux-kvm.com/sites/default/files/kivity-Reprint.pdf Accessed 7.5.2015.

Microsoft Documentation. Available: http://social.technet.microsoft.com/wiki/contents/articles/261.understanding-and-comparing-the-hyper-v-v1-0-and-v2-0-architecture.aspx Accessed 05.04.2015

Minicom 2015. Available: https://help.ubuntu.com/community/Minicom Accessed 04.04.2015

Tanenbaum, A., Bos, H., 2015. Modern Operating Systems. Essex, England. Pearson Educational Limited.

Sosinsky, B., Cloud Computing. Hoboken NJ. John Wiley and Sons Inc.

Stallings, W., 2005. The Unix Operating System. New Jersey, NJ. Prentice Hall Inc.

VMware Workstation 2011. Available http://www.vmware.com Accessed 11.04.2015

Wembao, M. 2003. Modern Cryptography: Theory and Practise. New Jersey, NJ: Prentice Hall.

Gesher, A., 2009. Like An Operating System for Data Analysis. Available https://www.palantir.com/2009/11/palantir-like-an-operating-system-for-data-analysis/ Accessed: 17.05.2015

APPENDIX 1

```
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
enable secret 5 $1$mERr$9cTjUIEqNGurQiFU.ZeCi1
!
!
!
!
!
!
!
!
!
!
!
```

```
!
!
!
!
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet1/0
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial2/0
 ip address 192.168.3.1 255.255.255.0
 clock rate 64000
!
interface Serial3/0
 no ip address
 shutdown
!
```

```
interface FastEthernet4/0
 no ip address
 shutdown
!
interface FastEthernet5/0
 no ip address
 shutdown
!
ip classless
!
!
!
banner motd ^C
This is a test device and setup ^C
!
!
!
!
line con 0
 password cisco
 logging synchronous
 login
!
```

```
line aux 0
!
line vty 0 4
 password cisco
 logging synchronous
 login
line vty 5 15
 password cisco
 logging synchronous
 login
!
!
!
end
```