Defence Research and Development Canada
Recherche et développement pour la défense Canada

Operating system hardware reconfiguration

A case study for Linux

R. Carbone
DRDC Valcartier

Defence R&D Canada – Valcartier
Technical Memorandum
DRDC Valcartier TM 2006-595
January 2007

Canada

Operating system hardware reconfiguration

A case study for Linux

R. Carbone
DRDC Valcartier

Defence R&D Canada – Valcartier

Technical Memorandum
DRDC Valcartier TM 2006-595
January 2007

Principal Author

Richard Carbone

Programmer Analyst

Approved by

Yves van Chestein

Head/Information Knowledge Management Section

Approved for release by

Gilles Berubé

Chief Scientist

Nobert Haché (DMSS)

Abstract

This memorandum explores how well a Linux system can withstand the test of time – specifically, whether it can withstand various hardware upgrades that would ensue over the coming years of its lifetime as the operating system for a command and control (C2) system. By understanding how Linux responds to changes in the hardware and how it can reconfigure itself, it allows for better understanding of the implications involved in maintaining a given operating system for the lifetime of the C2 system. Through several simple and repeatable experiments, it was determined that a given Linux operating system, if adequately maintained can successfully be used over the lifetime of the C2 system.

Résumé

Ce mémorandum étudie comment un système Linux résiste aux changements – particulièrement s'il peut résister aux changements périodiques de matériel sous-jacent qui permettrait d'assurer sa survie à titre de système d'opération d'un système de commandement et contrôle. En sachant comment Linux répond aux changements de matériel sous-jacent et comment il se reconfigure par lui-même, on peut comprendre les implications au niveau de sa maintenance pendant toute la durée de vie du système de commandement et contrôle. Suite à plusieurs petites expériences, on a déterminé qu'un système Linux donné, lorsqu'il est bien maintenu, peut être utilisé pour supporter un système de commandement et contrôle pendant toute sa durée de vie.

This page intentionally left blank.

Executive summary

Operating system hardware reconfiguration: A case study for Linux

Carbone, R.; DRDC Valcartier TM 2006-595; Defence R&D Canada – Valcartier; December 2006.

The Canadian Navy's Directorate of Maritime Ship Support (DMSS 8), under the auspice of the Halifax Modernized Command Control System (HMCCS) project requested that DRDC Valcartier perform an evaluation into how well the Linux operating system can withstand various hardware upgrades over the course of its lifetime as the new Halifax-class C2 operating system. More specifically, the Navy wished to learn if a Linux-based operating system can detect hardware changes and then reconfigure itself taking these changes into account. The HMCCS project seriously considered Linux as a potential new C2 operating system due to its flexibility and usability. The expected lifetime of the C2 system is expected to be between 15 to 25 years.

The hardware changes examined in this memorandum, although quantifiable as drastic hardware changes, are equally applicable to minor changes as well. If a drastic change is supported then surely a minor change is more readily supported by the operating system. A drastic hardware change occurs when most if not all of the hardware is no longer the same. This will happen, for example, if the operating system is migrated to a new system. Through empirical experimentation it has been determined that most (if not all) modern Linux-based operating systems can successfully undergo operating system hardware reconfigurations due to hardware changes where the operating system is reconfigured to support the new hardware. Furthermore, it can been reasonably concluded looking into the future that changes to the C2 system's hardware should be adequately supported by Linux-based operating systems in so long as the operating system itself (at least the kernel, drivers, and subsystems) are kept reasonably up to date. However, the various mechanisms and methods for keeping a Linux-based operating system up to date are not directly examined in this report.

Due to the lack of publicly available literature on the subject, experimentation was required to draw preliminary conclusions. While this memorandum can serve as a useful starting point, neither it nor its conclusions should be construed as definitive at this time. It would be well advised, rather, to understand the basic functionality of a given Linux-based operating system to determine if it is line with future objectives of a given project or requirement as not all operating systems are created equal.

This page intentionally left blank.

Sommaire

Operating system hardware reconfiguration: A case study for Linux

Carbone, R.; DRDC Valcartier TM 2006-595; R & D pour la défense Canada – Valcartier; December 2006.

Le Directeur - Soutien aux navires (DSN 8), sous la gouverne du projet Système de commandement et contrôle modernisé de la classe Halifax (SCCMH), a demandé que RDDC Valcartier accomplisse une évaluation de la capacité de Linux à se reconfigurer après une mise à niveau du matériel. Spécifiquement, la tâche consistait à évaluer si Linux pouvait être forcé à s'auto-reconfigurer et comment. Puisque le projet SCCMH envisage sérieusement la migration de ses systèmes de commandement et contrôle vers Linux, le système doit être capable de s'adapter et de survivre sur une période de 15 à 25 ans.

Les résultats observés suite aux changements de matériel de nature majeure s'appliquent aussi à des changements de nature plus mineure. Si un changement majeur est supporté, alors un changement mineur est sûrement supporté par le système d'opération. Un changement majeur est présent lorsque la plus grande partie, sinon la totalité du matériel n'est plus la même. Cela s'applique, par exemple, lorsque le système d'opération est porté sur une nouvelle plate-forme matérielle. À partir d'expériences empiriques, on a établi que la plupart (sinon la totalité) des systèmes d'opérations modernes basés sur Linux pouvaient se reconfigurer suite à un changement de matériel. En plus, il est raisonnable de conclure qu'un changement matériel d'un système de commandement et contrôle pourra être adéquatement supporté par un système d'opération basé sur Linux tant que le système d'opération lui-même (au moins le kernel, les pilotes et modules) est maintenu à jour. Par contre, les différentes techniques et méthodes nécessaires pour garder un système d'opération à jour ne sont pas étudiées dans ce mémorandum.

Puisqu'on manque de documentation disponible publiquement sur le sujet, une approche expérimentale était nécessaire pour en tirer ces conclusions. Bien que ce mémorandum puisse très bien servir de point de départ, aucune des ces conclusions ne devrait être considérée comme définitive pour le moment. Il serait plutôt conseillé de comprendre les fonctionnalités élémentaires d'un système d'opération particulier pour déterminer s'il respecte les objectifs ou besoins d'un projet donné, puisque les systèmes d'opérations ne se comportent pas tous de la même façon.

This page intentionally left blank.

Table of contents

List of tables

Acknowledgements

The author would like to thank Martin Salois for reviewing this report and translating the Abstract and Executive Summary during this report's first iteration. The author would also like to extend thanks to Marc Quintin for giving sound advice about the reader's requirements and for reviewing and translating the Abstract and Executive Summary during this report's second iteration.

1. Introduction

1.1 Objective

The main objective of this report is to determine if the Linux operating system is able undergo both an operating system hardware reconfiguration and a hardware migration. This will be determined through several empirical experiments using an older and more recent version of the Linux operating system. The experiments presented in this report aim to better understand if and how hardware reconfigurations can be used.

The secondary objective of this report is to determine if there are any general rules or observations that can be determined or extrapolated from these experiments. Understanding the actual mechanism(s) through which operating system hardware reconfiguration and hardware migration occur are not the subject of research in this report. This subject requires an in-depth knowledge of the Linux kernel and its configuration mechanisms in order to understand the actual responsible mechanism(s).

1.2 Background

The scope of this report is to provide technical information to the Canadian Navy which will aid it in determining which new operating system, if any, will replace the current C2 operating system (HP-UX) aboard the Halifax-class frigates. The frigates are currently undergoing modernization, including the operating system.

Originally, the mandate of DRDC Valcartier was to address the issue of various long-term support options that are available to the Navy should they decide to pursue the deployment of a Linux-based operating system aboard the frigates. This report is outlined in Report [1]. More specifically, with respect to Report [1], the Navy wished to understand how long-term operating system maintenance could be achieved. While this work was underway, the Navy's Directorate of Maritime Ship Support (DMSS 8), under the auspice of the Halifax Modernized Command Control System (HMCCS) project – the project responsible for the frigates' retrofitting – requested that DRDC Valcartier examine whether or not the Linux operating system could tolerate and handle changes in its underlying hardware. Put simply, the Navy wished to learn if the Linux operating system is capable of dealing with hardware changes and how well it can be expected to perform looking to the future.

Thus, this current study, examines the Linux operating system's ability to handle hardware changes, and although short-lived in nature, was designed as a preliminary examination of the subject. To date, there is little to no publicly available information on this subject. Due to the length of the time that a potential Linux operating system could be locked in for (or frozen) as the C2 operating system, a period of anywhere between 15 and 25 years, the Navy needed to ascertain the feasibility of not only maintaining the same operating system but whether it could handle periodic hardware upgrades that are certain to occur over the lifetime of the C2 system.

The key to understanding the purpose of this study is to understand the correlation between an operating system and its life expectancy. As time progresses, so does the hardware. It is only reasonable to expect that a C2 system will undergo periodic hardware changes. These changes could be minor or drastic in nature. However, in order for an operating system to be able to support hardware that has not yet come to market (or that even exists), it must be kept up to date so that it can deal with those changes when they arise. While it is not necessary to perform a complete operating system upgrade or update to achieve this, it is required, at the very least that the kernel, drivers, and underlying subsystems be kept up to date. The actual updating and upgrading mechanism(s) of the Linux operating system are not directly examined in this report as there are too many variables to be examined, and each Linux operating system is unique in its maintenance-related tasks. However, more information about system maintenance in general can be found in Report [2].

Periodically, it is also conceivable that full hardware upgrades (or complete overhauls) of the C2 system could occur, such that the operating system itself is altogether "migrated" or "copied over" to a new computer system. There are many reasons for doing this and are directly addressed in Section 2.2.3. To summarize, it can be said that the reason for preserving an existing operating system and migrating it over to another system would primarily be to avoid the necessity of reinstalling and reconfiguring the operating system, system configurations and preferences, applications, etc. Another important advantage to this is the more rapid redeployment of the C2 system as operating system reaccredidation and recertification will not be as lengthy or as difficult to complete.

1.3 Sources of information

While no useful publicly available sources of information could be found on the subject of operating system hardware reconfiguration and hardware migration, several internal documents, reports, publications, and draft publications were available and were instead used as the basis for this report. Some of these documents are informal in nature and thus have not been subjected to peer-review and are un-publishable. They are, in the order they appear in the References section:

1. Life-Cycle Support for Information Systems Based on Free and Open Source Software.

   Purpose: This report's objective is to ascertain the various life-cycle support options available to the Navy for the maintenance of a FOSS-based operating system such as Linux. This paper was presented to ICCRTS 2006.

2. Hardware Reconfiguration Methodology: A Linux case study.

   Purpose: The purpose of this document is to examine the methodology by which a Linux-based operating system should be updated, upgraded, patched, as well as how to backup and recover from a failed operating system hardware reconfiguration. At this time this paper is a draft Technical Memorandum that is due for completion December 2006.

3. Does Red Hat 5.0 Support Hardware Refreshes and can it Work on Modern x86 CPU's?

   Purpose: An internal document whose aim was to examine whether or not older versions of Red Hat Linux were able to undergo operating system hardware reconfigurations and whether it could work on more modern platforms and CPU's.

4. Can Linux be Easily Reconfigured from One Machine to the Next?

Purpose: An internal document examining the operating system hardware reconfiguration capabilities of Linux on different generations of x86-based CPU's and platforms.

5. A What to Avoid Guide in Doing your own In-House Migration.

Purpose: A short text on various issues to pay attention to and avoid when performing an operating system hardware reconfiguration, as well as a brief discussion of certain pitfalls and how to avoid them.

2. Technical background

2.1 Introduction

As has been stated in preceding sections of this report, the main objective of this study is to determine whether or not Linux-based operating systems are capable of performing what are known as operating system hardware reconfigurations and hardware migrations. To fully appreciate this, however, it is important to understand certain technical definitions and their background which are examined in the following subsections.

Some of the following subsections are simplifications only and are meant to be an introductory examination of various topics relating to long-term operating system maintenance. After reading the following subsections the complexities involved in long-term system maintenance should be clearer. A more thorough examination can be found in Report [2].

2.2 Operating systems

2.2.1 Definition

The term operating system has been given many varying definitions over the years. In this text, and in this specific context, the more classical definition of the term is maintained. An operating system can be big or small; it is not defined by its size. At a minimum, an operating system can be defined as "a collection of binary executable code and system configurations separated into individual computer files that controls the computer's hardware, operating system behaviour, and the user-based applications, tools, and utilities. An operating system is compromised of: 1) a kernel; 2) a shell or GUI for interacting with the kernel and launching applications, and 3) user-based applications." The kernel generally consists of itself as well as device drivers and a memory management facility. At its essence, the kernel is neither command-line based nor GUI-based; instead, it is an abstract layer of low-level software that controls the hardware and prepares it for performing useful work for the user. The user-input shell/GUI can be a command line-based shell as found with UNIX-based operating systems or a GUI as found with Windows-based operating systems. While they may appear to be different they perform the same basic tasks. User-based applications send signals and control requests to the kernel so that the hardware can process and run the application, perform calculations, display graphics, etc.

With Linux, it is important to understand the difference between what is meant by the Linux kernel and a Linux-based operating system. Linux, in of itself refers only to the GNU/Linux kernel which includes device drivers and various other subsystems such as a virtual memory management facility. A Linux-based operating system, on the other hand, at its most basic, is no more different than any other computer operating system. It too is compromised of a kernel, a shell (or GUI) for user-based input to interact with the kernel, and a collection of user-based applications. Thus, a Linux-based operating system is a grouping not only of the GNU/Linux kernel (and its subsystems), shell/GUI, but also an assortment of user-based tools, applications, utilities, documentation, source code, etc. Together, these various components form a stable and functional computer operating system.

Throughout this report the terms Linux (referring to the kernel) and Linux-based operating system are used somewhat loosely and interchangeably. However, it is not difficult to differentiate between the various meanings and implementations of the term Linux.

2.2.2 Dependencies

2.2.2.1 Library dependencies

Every operating system has dependencies. They are found throughout all operating systems, new and old alike. A dependency is said to occur when one piece of software, usually an application, requires something from another piece of software, usually a software library. The requesting application is seeking to fulfill a service or task that it itself cannot provide or perform; however, through its programming it knows where to look to fulfill its need(s). The requesting application thus makes a library call to a specific software library that is able to provide it a specific service. Almost all (if not all) the programs and applications in an operating system have library-based dependencies; and in general, the larger the program the more it is likely to have them. Developer's take advantage of these library services so that they can spend more time developing their applications and functionality rather than recreating existing services.

It is important to understand what a software library is. It is a compiled collection of services, specifically: subprograms, functions, API's, and subroutines that are stored in a binary executable format so that when a requesting application calls a specific piece of library code it can directly execute that code. To access a library service a library call is made; this call must state the library name and the service that is to be called. The developer must therefore know beforehand which library a given service resides within so as to correctly setup the library call from the requesting application (or library).

It is because many library services already exist that developers can spend more time developing newer and enhanced operating system components instead of wasting time redeveloping basic services into their software to provide the necessary foundations. However, if one library gets changed and some of its functionality is modified or altogether removed it could severely impact the system.

Building upon available library services will help to reduce overall operating system complexity through the reuse of code and services. In large Linux-based operating systems it is not uncommon to have many thousands of such dependencies spread across the entire system. This is due to the hundreds of applications and tools that are installed with the operating system. Furthermore, most if not all of the programs and applications will require functions and API's from any of the various software libraries distributed with the operating system. However, the bulk of those services will be provided by the core system's software libraries[1] .

---

[1] This is an essential system library that is absolutely required for the functioning of the system. Examples of this include the GNU C Library *libc.so* and *libstc.so*.

2.2.2.2 System calls

Another type of dependency is the system call [6]. A system call is an operating system interface and subroutine that provides a uniform method for being called and performing lower-level operating system-related tasks. These tasks can include operations such as deleting files, managing directories, opening hardware for I/O, communicating with the network, etc. It only makes sense for system calls to be placed in the kernel because it is here that control and management of the lower levels of the hardware and the operating system are to be found.

A system call consists of a low-level function or API that is provided by the operating system's kernel. The system calls are themselves based on various functions and API's that are provided by the Standard C Library and the GNU C Library that are compiled into the kernel and available as system calls via the Linux kernel. These C library functions are the basis for all system calls under Linux and UNIX-based operating systems.

By using system calls a programmer does not have recreate a function or API that is already provided by the operating system, and in all likelihood if he tried it would not be as well written or efficient as those found in the kernel. When a low-level task is required to be performed by an application, it places a system call that hands temporary control over to the kernel to execute the required task (or subroutine), and upon the kernel's completion of the task returns control back to the application.

2.2.2.3 Summary

It is easy to see how complex it can become managing dependencies (library and system calls). Sections 2.2.2.1 and 2.2.2.2 serve as a very preliminary introduction into the complexities and issues that can arise when dependencies are changed due to the modification of various operating system components [via upgrades, updates, and other means]. Changes to key library and system calls that are relied upon by the rest of the operating system can severely cripple both applications and the operating system alike. These issues must be kept in mind by those who manage and administrate operating systems.

The problems of dependencies are not unique to the Linux operating system and are found with all modern operating systems. Furthermore, today's modern operating systems consist of many thousands of such dependencies spread across the entire system. For a more detailed examination of the complexities involved in upgrades, updates, and general system maintenance, refer to Report [2].

2.2.3 The Linux kernel

The GNU Linux kernel is the heart of the Linux operating system, as any kernels is the heart of its respective operating system. The kernel is that piece of low-level software that takes care of and manages all system hardware including I/O devices and memory management. It provides that layer of abstraction necessary for the separation between higher-level applications and lower-level hardware. The kernel communicates with the computer system's hardware through device drivers. It manages memory through its virtual memory management facility. The kernel provides a uniform interface upon which to build the operating system and provides a system call facility so that many common low-level tasks and subroutines that are commonly requested by higher-level applications can be performed on their behalf rather than require the programmer to rewrite them from scratch. This frees the programmer to concentrate on higher-level application issues such as functionality.

Due to the openness of the Linux kernel and operating system it becomes possible to see how all the pieces fit together, layer upon layer, with the kernel as the basis for everything else. Without the kernel a great deal of extra programming would have to be incorporated into each program and library to allow for inter-process communications (IPC's), the management of system resources and hardware, etc. Processes, programs, and libraries interact with each other and with the computer's hardware and resources through IPC's and system calls (provided by the kernel). Thus, in understanding the dependencies placed upon the kernel by the rest of the operating system it is possible to comprehend how kernel changes can potentially affect the rest of the system.

2.3 Reconfigurations and migrations

2.3.1 Defining hardware reconfiguration

This term, mentioned throughout this report and reports [2, 3, 4, and 5], is equivalent to the terms operating system hardware reconfiguration, an operating system reconfiguration, a hardware reconfiguration, or simply reconfiguration. These terms can be defined as "the ability for an operating system to effectively deal with any changes to the underlying hardware and effectively perpetuate those changes to the appropriate software layers of the operating system such that the changes should remain as transparent as possible to the user."

Several things must be present for a successful reconfiguration to occur after the hardware has been changed. Firstly, the operating system must actually provide software-based support for hardware detection/redetection. Secondly, there must be a mechanism by which detection/redetection can be triggered, either automatically at system start-up or manually by the user at some point afterwards. Thirdly, the changes to the operating system must be effectuated in such a manner that they are transparent to the end user and do not generally require the user to manually modify system configuration files.

The majority of today's Linux distribution-based kernels are built "out of the box" to support a wide variety of hardware, devices, and computing platforms. Most of these operating systems also support various mechanisms for detecting hardware during their initial setup and installation phases. They also have redetection-based tools to detect post-installation changes to the hardware and make the necessary operating system changes to the various system configuration files. Some operating systems detect hardware changes automatically (sometimes referred to as dynamic reconfiguration) while others do not have this ability and must be run manually (this is referred to as static reconfiguration). In so long as the operating system supports either form (dynamic or static) of reconfiguration it can be said that the system does therefore support hardware reconfiguration. Whether it is dynamic or static is perhaps a reflection of the level of sophistication of the operating system itself. Therefore, for the remainder of this report both static and dynamic reconfigurations are treated as one in the same. Furthermore, it may be such that some enterprise versions of Linux ***may*** support hot-swapping and dynamic hardware reconfiguration; however, no evidence can be found for this existence. Hardware changes, other than PCMCIA changes are almost always made when the system is shutdown.

Generally, older Linux-based operating systems have more difficulty detecting hardware during installation, and almost never successfully detect post-installation hardware changes. Many of these older operating systems required many hours of manual modification to various system configuration files and the kernel to get the new hardware to work, assuming that the hardware actually was supported.

2.3.2 Defining hardware migration

A hardware migration, with respect to this report and reports [2, 3, 4, and 5] is often referred to as a migration. This is a term that is similar to an operating system hardware reconfiguration. However, they differ in that a migration is an operating system hardware reconfiguration that takes place only after the entire underlying computing platform has been replaced. In other words, the operating system is altogether transferred to another computer by various means. The operating system's system disk can be physically transferred to the new machine or the data can be copied over to the new machine's system disk through other means. An example of this would be moving a Linux-based operating system from a Pentium-class system to a Pentium IV-class system. Most, if not all the new system's hardware will be completely different from that of the original system. After the transfer is complete, the operating system hardware reconfiguration then takes place only once the power is applied and the new system and is allowed to boot up. Depending on the type of Linux operating system currently in use, the reconfiguration may be either dynamic or static in nature. Therefore, hardware migration and operating system hardware reconfiguration essentially refer to the same thing; they differ only in their perspective and scope concerning the changes made to the system.

2.3.3 Purpose of a hardware reconfiguration

The most common reason for performing a hardware reconfiguration is to save time and avoid having to reinstall either the same operating system or installing a newer one, as well as to avoid having to reconfigure the operating system and its configurations files. Furthermore, by performing a hardware reconfiguration on the existing operating system the security recertification and reaccredidation process will be much shorter. Thus, the existing installation can be leveraged and much unnecessary work and resources can be saved on.

The Navy has stated that it may not be possible for it to perform operating system upgrades[2] on the operating system due to restraints in security, certification, auditing, system management and maintenance. The Navy will likely opt to freeze or "lock out" the operating system with a given configuration for the entire lifetime of the C2 system which is likely to be a period of between 15 to 25 years. It is reasonable to assume, as the Navy has, that over such a long period of time periodic hardware upgrades will be made to the C2 environment. These upgrades may include minor hardware changes or be a complete overhaul of the C2 environment, as performed in this report's experiments.

[2] By this it is meant upgrading to the next subsequent version of the operating system provided by the maintainer of the distribution.

Thus, in order to maintain a system for a hardware reconfiguration it is necessary to maintain, at the very least, the kernel and it subsystems. The Navy has stated that operating system upgrades are out of the question due to the length of time it will take to recertify the operating system as fit for redeployment. Instead, by updating only the necessary components it will be possible to support newer hardware via hardware reconfigurations. As such, existing applications, preferences, configurations, and other settings can be maintained, as can the majority of the operating system dependencies, thereby forgoing the requirement for in-depth operating system recertification. Only software that has changed would need to be tested and recertified; namely the kernel. Since the kernel is much smaller to than the entire operating system, considering modern advancements in static analysis techniques, it will be easier to reassess the kernel's suitability for redeployment in a fraction the time it would take to recertify the entire operating system.

3. Methodology

3.1 Scope of experimentation

Having not been able to find any relevant reports, papers, or documentation on the subject of hardware reconfiguration, it became necessary to perform experiments to empirically confirm the hypotheses that were put forward in the following section. This report is not meant to be an all-inclusive guide to all possible incarnations of hardware reconfiguration. Rather, it should be seen as an introduction to this subject matter as it provides some preliminary conclusions. However, it should be emphasized that more experimentation with other Linux distributions and kernel versions are necessary to draw more specific and definitive conclusions.

This report, when combined with reports [1] and [2], will help to better inform the Navy and DMSS about the decisions that must be made concerning the maintenance and lifespan of the new C2 operating system if in fact it will be Linux-based.

3.2 Hypothesis

Thus, before going ahead with the experiments, the following hypotheses were predicted for Linux such that:

i) Newer and older Linux kernels alike will run on both newer and older hardware systems if they share the same basic architecture (i.e. x86-based systems).

ii) Red Hat 5 (RH5) will work but not support much of the hardware on today's modern systems.

iii) Fedora Core 4 (FC4) will work on older systems and support most if not all the hardware on both older and newer systems alike.

iv) An older kernel will run on newer systems that utilize backwards-compatible processors, although it will not support much if any of the newer system's hardware.

v) That if any device should be unsupported for RH5 then a supported device driver for kernel 2.0.x will be difficult if not impossible to find or make work.

vi) A successful operating system hardware reconfiguration or migration will only work with modern Linux distributions.

As with any hypothesis, it is important to await the final outcomes before coming to any final conclusion.

3.3 Configurations

The experiments are empirical in nature. They must be tested out in far more test cases than have been done here in this report in order to draw any definitive and absolute conclusions. The experiments performed are from different perspectives and examine how the Linux operating system and its kernel behaves when system hardware is changed. This is emphasized by performing operating system hardware reconfigurations and migrations on different platforms that are technologically generations apart.

At the beginning of each experiment there is a short introduction to better explain the goals and objectives of each experiment. Results for each experiment can be found at the end of their corresponding section. However, the Results analysis and Conclusion sections will summarize the results of all the experiments and attempt to draw any general rules or behaviours about the various Linux systems, respectively.

It is understood that different Linux distributions could yield different results. Thus, it is important that this be kept in mind while reading this report and that further experimentation with a variety of Linux operating systems could further solidify any conclusions.

3.3.1 The operating systems

The two operating systems that are used are Red Hat 5.0 (RH5) and Fedora Core 4 (FC4). RH5 was originally released December 1, 1997, while FC4 was originally released June 13, 2005. Fedora Core is considered the continuation of Red Hat's (RH) standard non-enterprise distribution[3] . These two operating systems are about 7.5 years apart, or 8 generations taking into account the iterations between RH5 and FC4. The operating systems are further apart than the actual hardware (see the following section). RH5 is no longer supported by RH and has been in end-of-life status for many years now. There have been no new recent kernel extensions for RH5 or kernel 2.0.x for many years.

It is understood that different Linux distributions, with their different tools and kernel settings and compiled-in features will differ significantly. However, in this report, only RH-based distributions are examined. It is unlikely that any other distribution, new or old would have significantly changed the various outcomes of this report. However, the Navy is invited to perform their own experiments and verify these claims.

The installation of RH5 is not very big, about 1 GB when all the packages are installed. At the time, RH5 was considered to be bleeding edge technology. A full installation of FC4 is slightly more than 6 GB in size.

3.3.2 The hardware

The specific hardware configurations for both test systems, a Pentium II and Pentium IV, can be found in Annex A and B, respectively.

[3] It was previously available for free download from RH when it was marketed as RH's entry-level operating system.

RH5 was released at around the same time that Pentium II processors (mid 1997 [7]) were making it to market; at this time the Pentium, 486, and Pentium Pro's were still very popular. The Pentium IV processor was released in late 2000. However, the specific processor used in the Pentium IV system throughout these experiments was released May 2003 [8]. This processor is a "hyper-threading" (HT) processor, a step above the standard Pentium IV processors that came before it. Therefore, there is about 6 years of difference between the two systems.

RH5 was compiled to run on all i386 compatible systems, including but not limited to the 386DX[4], 486, Pentium, and Pentium Pro systems. FC4 was compiled for Pentium-class systems only.

Undoubtedly, RH5 will not support USB, Firewire, or any other type of newer I/O hardware device. Furthermore, it is doubtful that it will even support Plug'n'Play.

3.4 Assumptions and potential problems

There are certain problems that can occur while placing an existing Linux installation onto a much newer machine.

In RH5, as with the majority of older operating systems, most of the newer hardware devices will not be supported. These can include Plug'n'Play devices, many newer PCI devices, video and network adapters, storage devices such as various SCSI, IDE, EIDE, ATAPI, USB, and Firewire devices. However, basic I/O will work. Basic I/O involves access to the system disk and other devices such as floppies and similarly supported hardware devices, including memory, virtual memory, and basic console input and output (terminal, keyboard, and mouse).

It is not the objective of this report to determine the facts about the dynamic reconfiguration capabilities of the Linux operating system. Instead, when performing the various experiments, dynamic reconfiguration will not be used. Although supported, in order to have a more consistent basis for performing the experiments, static reconfiguration is done by configuring the pre-migrated system to come up to *runlevel 1* before actually migrating it.

Runlevels are the UNIX equivalent of booting the Windows operating system into various levels of Safe Mode. The number and type of safe modes will vary according to the type of Windows operating system used. Under Linux and UNIX, there are many runlevels. These runlevels are configured one of two ways: modify the file /etc/inittab and specify the runlevel which will only take effect at the next system reboot or implicitly specify the runlevel at the command line using the *init* command (i.e. *init 3*). This command can only be run by the root user. Runlevels do not always have the same meaning under different UNIX-based operating systems. The following table will help to better understand the various Linux -based runlevels.

[4] The i386SX is actually a 2 16-bit pipelined processor, giving the appearance of 32-bits; however, it is still only a 16 bit processor.

Table 1. Runlevels and their significance

Level	Name	Significance
0	Halt	Halts (or shuts down) the operating system and hardware
1	Single-user mode	Root logins only; maintenance mode; network and services are disabled
2	Multi-user mode	Non-root logins allowed to the local system only; network and services are disabled
3	Full multi-user mode no GUI	Network and services online and available; non-root logins allowed both locally and from remote systems; X Windows disabled
4	Unused	Left empty as a user-configurable runlevel
5	Full multi-user mode with GUI	Network and services online and available; non-root logins allowed both locally and from remote systems; X Windows enabled
6	Reboot	Reboots the system

3.5 About the experiments

Three experiments have been performed for this report. Briefly, the experiments as performed in Section 4 are described below.

Experiment I investigates the capabilities of Linux-based migration and hardware reconfiguration using a recent version of the operating system on both a Pentium II and Pentium IV. Specifically, FC4 will be installed onto the Pentium IV and then migrated to the Pentium II, and then migrated back to the Pentium IV to verify that: 1) FC4 can be installed to a much older system; 2) migration/reconfiguration does occur, and 3) that it can occur multiple times without incident.

Experiment II examines the migration/reconfiguration capabilities of an older Linux operating system from an older platform to a newer one and then back again. Thus, using RH5 it will be installed onto the Pentium II and then migrated/reconfigured onto the Pentium IV and then back again to determine how well it responds to multiple occurrences of a migration/reconfiguration.

Experiment III focuses on RH5's ability to install onto much newer equipment, equipment it was designed for. Therefore, the experiment is to verify if it can be installed onto the Pentium IV without incident (the inverse of Experiment I). It is desired that this experiment will demonstrate if there are any potential issues involved with installing older versions of Linux onto newer platforms.

Experiment III should be of particular interest to the Navy since they have previously acknowledged their interest in possibly installing older operating systems to newer hardware. However, after the experiments have been conducted and thoroughly analyzed the Navy may be able to make more informed decisions about the deployment and exploitation of older operating systems with respect to newer hardware platforms.

After completion of the experiments, migration and hardware reconfiguration should be sufficiently understood such that preliminary conclusions can be drawn. An analysis of the results will attempt to determine any underlying trends present in Linux concerning past and future hardware support trends as well as trends for migrations and reconfigurations.

4. Experiments – migrating to newer and older hardware

4.1 Experiment I – FC4 – from the Pentium II to the Pentium IV and back again

4.1.1 Brief introduction

In this experiment, the goal is to empirically verify what will happen when FC4 is installed onto a Pentium II system and then migrated to a Pentium IV system. If the migration and ensuing operating system hardware reconfiguration are successful, a second migration and operating system hardware reconfiguration back to the Pentium II will be attempted.

4.1.2 Installation procedure

The Pentium II system (technical specifications in Annex A) was setup such that the hard drive is the primary master, the LS-120 the primary slave, and the CD-ROM drive the secondary master. The system was booted and from the BIOS, all disk devices were detected. The system was then rebooted and FC4 was then installed from the FC4 CD 1 as follows:

- boot CD-ROM kernel using the "linux text" kernel parameters
- select "English" installation"
- select "custom installation"
- select *Disk Druid* and create disk partitions as follows:

Table 2. Disk layout for /dev/hda under FC4

Partition	**Device**	**Size**	**Type**	**# Cylinders**
Swap	hda1	245 MB	swap	1-533
/	hda2	5935 MB	ext3	534-13396
Empty	none	6 MB	None	13397-13410

- select *GRUB*
- select "no Boot Loader options"
- select "no *GRUB* password"
- accept default *GRUB* options
- install *GRUB* to MBR
- accept "DHCP/activate on boot for eth0"
- set hostname to "testlnx"
- select "no firewall"
- confirm the above choice
- disable SELinux

- select time zone “America/Montreal”
- choose root password
- Select the following sets of packages, not withstanding the default packages:
    - X Window System
    - GNOME
    - KDE
    - Editors
    - Sound and Video
    - Graphics
    - Mail Server
    - DNS Name Server
    - FTP Server
    - Network Servers

The installation will then proceed and ask for the CD’s to be changed when necessary. The installation will require all 4 CD’s. When the installation is complete, the system will eject the CD for removal and reboot the system.

After the reboot the operating system is loaded. A little over a minute later the “Welcome” screen is presented. The following were used to finish the post-installation process:

- select “Next”
- select “Yes, I agree to the License Agreement,” then select “Next”
- select correct date and time, then select “Next”
- select the appropriate video resolution, then select “Next”
- do not create any users, then select “Next”
- do not install any additional CD’s, then select “Next”
- select “Next”

The system then finishes its boot cycle and obtains a DHCP address. To configure the DHCP client portion of the installation, technical support may be required from the appropriate IT support resource.

Two items should be noted. The first is that the installation program did not create any emergency boot diskette by default, something that is vitally important if the system cannot boot correctly. Secondly, FC4 does not suffer from any issues for booting past a disk’s 1024th cylinder even though, architecturally, the Pentium II does.

4.1.3 Testing the Pentium II

Once the system was completely booted up, from the command line, the command *Ntsysv* was used to enable the *SSH* and *HTTP* services. A small web site was then copied over to the system from another UNIX system using *NFS*.

From a Windows-based workstation, a *SSH* client was launched and was able to connect to the Pentium II. From the Pentium II, it too could connect to other local *SSH* servers. Using *Internet Explorer*, the Windows workstation was able to access the Pentium II’s web server. From another UNIX workstation on the network a *NFS* share was created and correctly mounted by the Pentium II. The Pentium II was then configured to share its own *NFS* data which was correctly mounted from the previous UNIX workstation. Then, from the Pentium II, using its web browser, data files were successfully downloaded from the Internet. Finally, the Pentium II system was configured to join the local UNIX *NIS* domain currently residing on a local Sun server. The system was then able to accept network logins from the *NIS* domain. From the Windows workstation, *Nmap* was able to successfully verify the services running on the Pentium II.

Therefore, networking and subsequent services were tested and ascertained to be fully functional.

Under X Windows, multimedia AVI and MPEG movies were played using *XMMS*. While there was no sound for the system (no sound card installed on the system), video playback was smooth.

Various CD's were mounted (no DVD player on this system), both from the command line and from within X Windows. CD automounting worked correctly. It was also possible to mount floppies and LS-120 DOS-formatted disks; however, they had to be mounted manually as the automount[5] service does not work for these devices. The */etc/fstab* file had the correct entries for mounting CD's drive and LS-120 devices (LS and 1.44 MB floppies). A USB disk drive was connected to the system and it was automatically detected and mounted. A file listing on the drive worked as well. A USB memory stick was then placed into an USB receptacle which was also automatically mounted and a file listing worked on it as well. All USB devices were then unmounted.

Disk I/O (floppy, CD, LS, hard disk), automounting, as well as additional I/O devices such as USB devices were tested and found to work. Video subsystems were also found to be working correctly. As far as could be determined, the system worked correctly without incident or strange random events occurring.

4.1.4 Hardware reconfiguration – from the Pentium II to the Pentium IV

The operating system recently installed onto the Pentium II underwent an operating system hardware reconfiguration to a much newer system, a Pentium IV (technical specifications in Annex B). In order to successfully perform and test for a successful migration, the following changes were made to the Pentium II prior to the migration:

- ensure that the system will come up into *runlevel 1* only (modify */etc/inittab*)
- disable all network services including *NIS* network authentication

Since the available disk drive bay on the target system (Pentium IV) will occupy the same bus position as on the Pentium II no modifications to the file */etc/fstab* were necessary for the kernel to boot[6].

---

[5] It is important to note that the automount service only works under X Windows.

[6] It is important to understand that the kernel is unable to correctly boot if the file */etc/fstab* does accurately reflect the partition layout. This is because the kernel may need to load certain additional non-built-in drivers and modules. Furthermore, the kernel may be dependent on certain key system libraries.

The system was powered down, the hard disk drive taken out of the Pentium II and inserted and fastened into the Pentium IV. The I/O and power cables were connected to the drive, the system's chassis was closed, and the system powered up. Then it was necessary to go into the BIOS to detect the new disk and make the appropriate adjustments. Upon exiting the BIOS, the system was rebooted, and in less than half a minute the *GRUB* boot-loader was waiting for input. None was given, the default options were assumed and the system started booting the operating system. The system entered single-user mode (*runlevel 1*). Running the command *Kudzu* caused the system to reconfigure the operating system's hardware tree and devices. No program output appeared and it took about 30 seconds to complete. The command ran a second time, again with no output. Switching to the directory */etc/sysconfig* and examining the file *hwconf*, it was evident that the old system's hardware settings had been replaced with those of the new system.

Kudzu is a utility that modifies many system files, including but not limited to */etc/fstab*, */etc/sysconfig/hwconf*. Various networking parameters and other configurations are also found in various locations under */etc/sysconfig*, as well as the *xorg.conf* X Windows configuration file.

Logging in as root at the console, the command *init 3* was issued to bring the system back into multi-user mode. All the services available earlier while the operating system was on the Pentium II came back online automatically. None of the services failed including but not limited to *SSH*, *HTTP*, and *NIS*.

4.1.5 Testing the Pentium IV

Once the system finished executing all the necessary *runlevel 3* scripts, system tests were performed.

Tests were made to determine that the UNIX workstation's *NFS* share could be mounted on the Pentium IV. A *NFS* share was created on the Pentium IV and shared out and successfully mounted by the UNIX workstation. Various *NIS* users were used for logging in to the Pentium IV, which worked correctly. *SSH* services could be accessed on other systems, and the *SSH* service on the Pentium IV could also be accessed from other systems. The web site on the Pentium IV was also accessible. Then, using the *init 5* command, the system was brought up into X Windows. Logging in as root, the display was preset to 800x600. This was then changed to 1152x864. Video (AVI and MPEG) files were played using *XMMS* and playback was smooth. The sound card worked when speakers were connected to the system.

Therefore, networking and subsequent services were tested and ascertained to be fully functional.

Floppy disk tests were initiated by mounting and unmounting them as well as reading and writing to them. Since floppies are not automatically mounted, they had to be mounted by hand from the command line. However, *Kudzu* did not automatically remove the LS-120 entry in */etc/fstab* and it had to be replaced with the appropriate 1.44 MB floppy entry.

Since the system came equipped with a CD-RW/DVD drive, tests were made to write to blank CD's. ISO images were built of the */etc* directory and burned to disk. The disk could then be automounted. DVD data disks and video disks were also automounted. Furthermore, DVD video disks were automatically played by *XMMS*.

A USB hard disk drive was connected to the system and was automatically mounted. A file listing worked correctly on it. Then it was unmounted and replaced with a USB memory stick which was also automatically mounted. It too could be file listed.

Disk I/O (floppy, CD, LS, hard disk), automounting, as well as additional I/O such as USB devices were tested and found to work. Video subsystems were found to be working correctly.

The system was then configured to come up to *runlevel 5* at the next subsequent reboot. *Runlevel 5* was achieved without issue.

As far as could be determined, the system worked correctly without incident or strange random events occurring.

4.1.6 A second hardware reconfiguration – from the Pentium IV back to the Pentium II

Once it was determined that FC4 was working correctly on the Pentium IV, the system was reconfigured to come back up at boot time to *runlevel 1*. None of the network programs or services were disabled since booting into *runlevel 1* does not load any of those services. The system disk was then removed from the Pentium IV and placed back into the Pentium II. Once inserted into the Pentium II, from the BIOS the disk was and the system was rebooted. Once the system came up to *runlevel 1*, at the command line in single-user mode, *Kudzu* was run twice. Then the hardware configuration files were checked to verify that the changes reflected the current system's hardware inventory. Those changes were verified and confirmation was made that the system had been brought back to its original form. Services like networking, *NIS*, and X Windows succeeded when the system was brought back up to *runlevel 5*. The system was then tested in the same manner as done in sections 4.1.3 and 4.1.5.

The system was properly reconfigured for X Windows and it was found to be working correctly. Networking, *SSH* access to and from the current system, as well as the system's web server worked. *NFS* was found to be working both for mounting remote shares and providing local shares for sharing out to the network. Access to floppies and LS-120 disks worked also; however, *Kudzu* did not automatically put back the appropriate entries into */etc/fstab* for the LS-120 drive; it had to be done manually. The CD-ROM was also found to work and could be automounted under X Windows. USB connectivity tests were then performed both on an USB hard disk drive and an USB memory stick. They were correctly automounted under X Windows and they could be file listed. Video (AVI and MPEG) files were played using *XMMS* and playback was smooth. Again, no sound was available as there was no sound card in the system.

Therefore, networking and subsequent services were tested and ascertained to be fully functional. Furthermore, disk I/O (floppy, CD, LS, hard disk), automounting, as well as additional I/O such as USB devices were tested and found to work. Video subsystems were found to be working correctly. As far as could be determined, the system worked correctly without incident or strange random events occurring.

4.1.7 Results

The new Linux kernel, version 2.6.11 which is currently used in FC4 appears to support both the majority of modern-day (circa 2004) and outdated PC hardware, including but not limited to motherboards and chipsets, sounds boards, video/display devices and adapters, mice/keyboards, networking components, and various disk devices. The newer kernel also has excellent support for USB and Firewire-based devices.

After verifying the compilation options for the FC4 kernel, it was found to have been compiled with built-in support for the full range of Pentium-based systems, new and old alike. However, support for non-Pentium systems was not compiled in. These systems include both 486 and 386 systems. However, if this support were required, it could be done by recompiling the kernel on a supported Pentium system with the appropriate processor (486/386) options enabled and then installing the newer kernel onto the older system.

Thus, an operating system hardware reconfiguration that would require non-Pentium systems can be done by rebuilding the kernel on a Pentium-class system and then migrating the operating system over to the non-Pentium system. Since it has been shown in this experiment that FC4 can go both ways via the reconfiguration/migration route, this option will work. Thus, due to the modular design of the Linux kernel specific hardware options that are required for functionality can be enabled or disabled as necessary and built into the kernel and its device drivers.

If the Linux operating system's kernel is periodically updated, continued hardware support for both older and newer systems can continued to be expected in the future. Thus, as time goes by, if a kernel and its required subsystems can be kept "reasonably" up to date, there is no reason why it will not support newer hardware. Furthermore, it may possibly provide improved hardware support for older systems and hardware. Keeping the kernel up to date will also improve the likelihood of hardware being correctly detected during a migration/reconfiguration. This is explored further in Experiment II.

The results that can be drawn from this experiment are:

1. Modern versions of Linux (and the kernel) can be expected to perform well when performing an operating system hardware reconfiguration/hardware migration to/from newer/older hardware;

2. Modern versions of Linux will support both recent and much older hardware with little to no manual configuration required;

3. The kernel will not by *default* support very old processors such as the 486 and 386; however, support can be compiled in.

4. It may be necessary to recompile the kernel to incorporate specific hardware support (i.e. previously unsupported adapter) present in the kernel source code but that is currently not enabled or compiled into the kernel.

5. Keeping a Linux distribution's kernel and required subsystems will lead to a successful operating system hardware reconfiguration/hardware migration for newer hardware that the original kernel was not meant to support.

6. Newer software can, at times, overcome architectural difficulties inherent in older technology. This is apparent with the case of the *GRUB* boot-loader where it was able to boot beyond the 1024th cylinder which is normally not possible on older computer systems.

4.2 Experiment II – RH5 – from the Pentium II to the Pentium IV and back again

4.2.1 Brief introduction

In this experiment, RH5 will be installed onto the Pentium II and its functionality verified. It is uncertain if all the hardware will be adequately supported on the Pentium II. At the very least it is expected that the basic I/O subsystem and console will be functional. Once installed and functioning the system will then undergo a migration and operating system hardware reconfiguration/hardware migration to the Pentium IV system. The system will then be tested to determine what is and is not functional. It is likely that RH5 can be migrated to the Pentium IV but unlikely that it will be able to successfully undergo an operating system hardware reconfiguration/hardware migration on the Pentium IV.

4.2.2 Installation procedure

The Pentium II system was configured such that the hard drive is the primary master, the LS-120 the primary slave, and the CD-ROM drive the secondary master (see Annex A for technical specifications). The devices were detected from the BIOS and the system was then rebooted. RH5 was installed from the RH5 CD-ROM as follows:

- Press “Enter” at the Linux boot-loader menu

The kernel begins booting and some seconds later, the system prompts for information about the display:

- “Press Yes for a colour monitor and No for a black & white monitor.” Select “Yes.”

The Welcome screen then appears:

- Press “Enter”
- Select “us” keyboard” and press “Enter”
- Select installation media as “Local CD-ROM”
- Select “Install” and press “Enter”
- Select the appropriate choice for the system attached SCSI adapters. None were selected. Press “Enter”

Note that, because RH5 only supports a maximum swap size partition of 127 MB, several swap partitions had to be created to have an adequate RAM-to-swap ratio.

Because RH5’s version of *LILO* has issues booting beyond the 1024^{th} cylinder on older PC’s, an appropriate partition layout had to be created to avoid this issue. Modern versions of *LILO*, however, do not suffer from this problem.

Using the *Disk Druid* tool, the following disk layout was created as follows:

Table 3. Disk layout for /dev/hda under RH5

Partition	**Device**	**Size**	**Type**
/	Hda1	472 MB	Ext2 filesystem
/usr	Hda5	1024 MB	Ext2 filesystem
/var	Hda6	1024 MB	Ext2 filesystem
/usr/lib	Hda7	1024 MB	Ext2 filesystem
swap	Hda7	127 MB	swap filesystem
swap	Hda8	127 MB	swap filesystem

Once the partitions were laid out, there was about 1.5 GB of free disk left at the end of the disk, which was left unused. The changes were saved to the partition table and the installation continued.

- Do not check for bad blocks
- Select all packages and install them

Once the installation starts, the filesystems are formatted and the software packages are then installed.

Upon completion of package installation, the installer found a PS/2 mouse that was configured as follows:

- Emulate a 3-button mouse
- Select the appropriate video card (in this case, the current adapter is an ATI Mach64 3D Rage II+ with internal RAMDAC); the installer fetches the appropriate driver from the CD-ROM
- Choose Custom and built a monitor configuration for the current monitor (RH5 does not auto-detect monitor or video devices)
  - Select 1280x1024 @ 76 Hz
  - Select 50-90 Hz for the monitor
  - Do not probe for the video devices as it usually causes the system to freeze
  - Select the appropriate amount of video memory
  - Select "No Clockchip Setting"
  - Select "24-bit" and "1024x768" resolution

Configure networking:

- Select "Yes" and press "Enter"
- Select the appropriate driver for the network card (in this case, a 3Com 3C59x (PCI))

However, card detection failed; the installation program could not detect it. All of the available 3Com drivers provided by the installation program were tried and they all failed. Networking could not be made to work, not even partially.

- Since networking did not work, select “Cancel” and press “Enter”
- Set the time zone to “Canada/Eastern”
- Do not configure any printers (local or otherwise)
- Provide the root password
- Install the boot-loader to the MBR
- Do not pass any options to *LILO*

When the installation program is complete the system is rebooted. The system then rebooted and everything came up correctly. The system, by default comes up to *runlevel 3* (X Windows will not start up automatically unless the system is in *runlevel 5*).

The system name defaults to “localhost.localdomain.”

The RH5 installation program did not create any emergency boot diskette by default, something that is vitally important if the system cannot boot correctly.

4.2.3 Testing the Pentium II

After installation, the system did boot up correctly and it was possible to log in to the system console as the root user.

X Windows was started up by running the startx command. However, X Windows did not start-up and resulted in errors that complained about video timings and resolutions. All possible combinations for the current video card/monitor were tried, using 8, 16, and 24-bit image depth with various monitor resolutions and timings, all of which failed. Finally, various video card memory sizes were also tried. This did nothing to resolve the matter.

It was then decided to attempt installing the standard SVGA X server. It was configured using an 8-bit driver, running at 800x600 @ 56 Hz, using an unlisted card, with 1 MB RAM, with a monitor frequency of between 50-70 Hz. This too changed nothing. Various memory sizes were also tried, again with various combinations for image depth, resolution, and timings, all of which failed. Finally, the same procedures were applied to the standard VGA X server, which also did not work. No further video card drivers for versions older than Red Hat 7.x could be found anywhere.

Therefore, it can be concluded that X Windows and video graphics could not be made to work under RH5. This is not uncommon with older versions of Linux due to poor X Windows video/monitor support.

Running the *dmesg* command and piping its output through the *more* command displayed system messages confirming that the kernel did detect a 3Com PCI adapter but that it was an unsupported version. After checking 3Com’s web site, it was found that the PCI version of the current network adapter was only supported under kernels 2.2.x and higher. Drivers were available for kernels 2.2.x and 2.4.x (http://support.3com.com/infodeli/tools/nic/linuxdownload.html). The current kernel version, 2.0.32, was unsupported by 3Com. Furthermore, after closely examining the details of the system messages about the network adapter and the PCI kernel source code file, *pci.h*, it was determined that at least some PCI technology (maybe all) was unsupported by RH5. The time frame of the release of RH5 was in line with the date when PCI technology was being brought to market.

While networking did not work, local network services did work on the localhost for services such as *HTTP* and *SSH*. It was possible to *SSH* the local system from a local *SSH* client. It was also possible to *Telnet* locally. Thus, local system-only networking was functional.

Both the hard drive and filesystem worked well even though it was formatted using the older Ext2 filesystem. The CD-ROM and LS-120 also worked correctly. However, the installer did not include any information about the LS-120 or floppy capabilities into the file */etc/fstab*. Instead, the device entry for a floppy disk drive, */dev/fd0*, had been added instead. Therefore, the appropriate LS-120 entries had to be entered manually into the file */etc/fstab* as follows:

```
/dev/hdb    /mnt/ls120    auto      defaults 0 0

/dev/hdb    /mnt/floppy   noauto    defaults 0 0
```

The mount points (*/mnt/floppy* and */mnt/ls120*) also had to be created manually on the root filesystem. After these two short steps, both LS-120 and floppy disks were found to work correctly. However, automounting was not a feature of this operating system, at least outside of X Windows, but since X Windows was not working, this too could not be confirmed. All disks had to be mounted manually, including the CD-ROM.

USB support was also not available under this kernel and the various USB devices connected to the system were never detected. Furthermore, the hardware tree did not provide any usable USB devices entries. The system was rebooted and the PS/2 mouse and keyboard were replaced with their USB equivalents. The system was not able to accept any I/O from either of these devices. They were removed, the system was rebooted and the PS/2 devices reinserted which restored console functionality. Therefore, USB support was not available.

It was concluded that the basic system was functional and stable. However, it was of limited use due to its lack of network and graphic-based capabilities.

4.2.4 Hardware reconfiguration – from the Pentium II to the Pentium IV

The system was set to come up to *runlevel 1* at the next boot. The system was shutdown and the disk was transferred to the same bus location on the Pentium IV (technical specifications in Annex B). The cables and other connections were reattached. The system was brought up to the BIOS and made to detect the new disk. Once detected, the changes were saved and the system rebooted.

However, it was unknown if a hardware migration would in fact take place because RH5 does not include either *linuxconf*[7] or *Kudzu*.

Normally, on the Pentium IV, a USB mouse and keyboard can be used, but these connections types were only attempted in the next section. For the time being, the required PS/2 devices were connected. The system was powered on and it began booting, but then the *LILO* boot-loader hung. Linux could not load the kernel. The top left-hand corner of the screen presented the following error:

LI

LILO started loading and then just hung. The RH5 CD-ROM was booted into "rescue mode" and was found to be too barebones to be of any real use. In order for the system to boot correctly, it was necessary for the boot-loader to load the kernel. To fix the problem, a DOS bootable diskette was created. Using MS-DOS 6.22, the disk was made bootable and the following files were copied to the diskette manually:

IO.SYS
MSDOS.SYS
HIMEM.SYS

From the RH5 CD-ROM (under *RH5CD\dosutils*) several files were also copied over to the diskette to provide a method of boot-loading the kernel. Under Linux, the kernel is generally self-contained within one single file. Device drivers are loaded only after the kernel itself loaded and provided the system with a means to accessing filesystems. The copied files were:

LOADLIN.EXE
VMLINUZ

The first file (*LOADLIN.EXE*) is a DOS -based boot-loader and the second (*VMLINUZ*) is the actual Linux kernel. It is the kernel used by default under RH5. A customized *CONFIG.SYS* had to be created as follows:

DEVICE=HIMEM.SYS
DOS=HIGH
DOS=UMB
FILES=40

High Memory Area access is required and this requires the driver *HIMEM.SYS* to be loaded, including the lines *DOS=HIGH* and *DOS=UMB* to be used. There was no need to create the *AUTOEXEC.BAT* file since no parameters or commands specifically needed to be executed from this file. After the diskette booted, running the following command would actually result in the booting of the kernel:

A:\> LOADLIN VMLINUZ root=/dev/hda1

[7] This is another Linux-based reconfiguration tool that was made available before *Kudzu* became available.

LOADLIN loads the kernel *VMLINUZ* into the High Memory Area (HMA). The option "root=/dev/hda1" instructs the Linux kernel where the drivers and other kernel modules reside, including system start-up scripts. The diskette kernel booted, and then loaded the necessary drivers and executed the appropriate system scripts. Shortly thereafter a login console was available. Logged in as the root user, the file */etc/lilo.conf* was modified and the option "Linear" was added to it. The file was then saved and the following command was run:

$ lilo -v

This command made the appropriate changes to the MBR by reloading the new parameters from the *LILO* configuration file. After the reinstallation of *LILO*, the following message was given:

Added Linux *

This meant that the reinstallation of *LILO* was successful. The system was then rebooted and it was able to boot correctly without the use of a DOS-based boot diskette and without any further intervention.

4.2.5 Testing the Pentium IV

After the system came up the root user was then able to log in to the system. The system was then configured for *runlevel 3*. It was not possible to get either networking or X Windows working any more than it had been possible while the operating system was on the Pentium II. The Intel-specific network adapter was not supported by RH5. Again, it was the same problem with X Windows as RH5 did not have the appropriate driver to manage the system's video adapter. Therefore, graphics and networking were completely inoperative.

The file */etc/fstab* had not been updated to reflect the change from a LS-120 device to a floppy drive. The change had to be done manually by removing the LS-120 entry and adding the appropriate floppy drive entry, device */dev/fd0*. Once done, floppy disks worked correctly. It was also possible to mount and list files on mounted CD's. However, DVD support was not available under RH5 and thus both DVD data and video discs were completely inaccessible. USB devices were also attempted, including memory keys, external disks, mouse and keyboard. None of the devices worked, even after rebooting the system with a USB mouse and keyboard connected to replace their PS/2 equivalents.

Thus it was determined that basic I/O worked, including hard drive, floppy drive, DVD drive (CD's only), as well as text-based display (console), mouse, and keyboard. Unfortunately, USB devices were completely inoperative.

Local networking continued to work. It was still possible to *SSH* and *Telnet* the localhost system only. Thus it can be concluded that basic localhost networking was functional.

As far as could be determined, the system worked correctly without incident or strange random events occurring. Nevertheless, the system was of limited use and the migration and operating system hardware reconfiguration had not been successful. For both to have been successful there should have been a redetection of hardware (either automatic or static) and a reconfiguration/migration to reflect this change, as well as the ability to use it.

4.2.6 A second hardware reconfiguration – from the Pentium IV back to the Pentium II

The Pentium IV was left at *runlevel 3* since there would be no discernable difference to the operating system whether it ran on either machine. The Pentium IV was then shutdown and the system disk was transferred back to the Pentium II. This was done to verify if the operating system would still be able to function after having been placed into a different system and then returned back to its original system. Since no system configuration changes had been made other than to */etc/fstab* everything should have continued working just as it already had.

The system disk was physically placed back into its original location within the Pentium II. The system was powered up, and from the BIOS the system disk was redetected. The system was then rebooted and Linux was allowed to boot up. Once booted it was possible to login in as root. The same tests as performed in sections 4.2.3 and 4.2.5 were performed and yielded the same results. However, since the Pentium II had a LS-120 drive and no floppy the contents of the file */etc/fstab* had to be reconfigured to reflect this.

The system appeared unchanged from the initial test performed in Section 4.2.3 and nothing about the system's behaviour warranted any expectations that the system had somehow changed its configuration or comportment during the unsuccessful migration and operating system hardware reconfiguration.

4.2.7 Results

From this experiment, the following findings can be drawn. The first is that older Linux kernels will boot and operate on processors that it was not specifically designed to run on in so long as the newer processors are backwards compatible. For example, while the code-set is not precisely the same between the 386 and Pentium IV processor, the Pentium IV is nonetheless backwards compatible with the 386 code-set. Thus, kernels compiled for 386/486/Pentium systems will run on more modern processors in so long as they remain backwards compatible. The results appear to indicate that older Linux kernels will at least support basic I/O and hardware. So long as PC's remain backwards compatible, an older Linux kernel will be able to boot and provide basic services.

The extent of an operating system's level of functionality will depend greatly on the amount of hardware support built into the kernel. In this experiment, since the RH5 kernel was at least 8 years out of date with FC4, it is only reasonable to expect that more modern peripheral devices such as the PCI bus, USB, Firewire, modern network cards, video adapters, etc., would not be supported.

Of equal importance, older Linux distributions do not have the ability to automatically detect new hardware and make the necessary changes to the operating system and configuration files to support the newer underlying hardware. It is therefore unreasonable to expect older Linux distributions to adapt to modern hardware and provide much functionality. It is possible, however, that some newer hardware components may be compatible with older existing drivers found in older releases of Linux.

Lastly, in performing migrations and reconfigurations of older operating systems to newer machines, there is always the risk that certain hardware components of the newer system may not be compatible with the operating system. Such was the case with *LILO*. Although in this case it was possible to reinstall *LILO* and fix it, with certain newer hardware it may not be possible and the system may be rendered inoperative. In such a case, it would be necessary to upgrade the kernel, drivers, and subsystems so that the required level of functionality and support are achieved.

4.3 Experiment III – Installing RH5 onto the Pentium IV

4.3.1 Brief introduction

The purpose of this experiment is to verify if RH5 can be installed on the Pentium IV. It has been ascertained (see Experiment I) that FC4 can be installed onto both the Pentium II and IV; however, it is unknown if RH5 can be installed onto a system that it was never designed to run on.

While it has been ascertained (see Experiment II) that older kernels will work on newer processors even if support is not directly compiled in, it is not known if an installation program will work on newer processors.

4.3.2 Installation procedure

The Pentium II system was configured such that the hard drive is the primary master, the LS-120 the primary slave, and the CD-ROM drive the secondary master (see Annex A for technical specifications). The devices were detected from the BIOS and the system was then rebooted. RH5 was installed from the RH5 CD-ROM as follows:

- Press "Enter" at the Linux boot-loader menu

The kernel begins booting and some seconds later, the system prompts for information about the display:

- "Press Yes for a colour monitor and No for a black & white monitor." Select "Yes."

The Welcome screen then appears:

- Press "Enter"
- Select "us" keyboard" and press "Enter"
- Select installation media as "Local CD-ROM"
- Select "Install" and press "Enter"
- Select the appropriate choice for the system attached SCSI adapters. None were selected. Press "Enter"

Note that, because RH5 only supports a maximum swap size partition of 127 MB, several swap partitions had to be created to have an adequate RAM-to-swap ratio.

Because RH5's version of *LILO* has issues booting beyond the 1024th cylinder on older PC's, an appropriate partition layout had to be created to avoid this issue. Modern versions of *LILO*, however, do not suffer from this problem.

Using the *Disk Druid* tool, the following disk layout was created as follows:

Table 4. Disk layout for /dev/hda under RH5

Partition	**Device**	**Size**	**Type**
/	Hda1	472 MB	Ext2 filesystem
/usr	Hda5	1024 MB	Ext2 filesystem
/var	Hda6	1024 MB	Ext2 filesystem
/usr/lib	Hda7	1024 MB	Ext2 filesystem
swap	Hda7	127 MB	swap filesystem
swap	Hda8	127 MB	swap filesystem

Once the partitions were laid out, there was about 1.5 GB of free disk left at the end of the disk, which was left unused. Save the changes to the partition table and continue the installation:

- Do not check for bad blocks
- Select all packages and install them

At this point, the filesystems were formatted and the first package was to be installed onto the system when the following message appeared on the console:

RPM install of setup failed:

package setup-1.9.1-1 is for a different architecture

Pressing the "Enter" key proceeded on to the following package, but the same message kept appearing for each ensuing package. With each following package the error message continued to appear. It slowly became clear that not a single package had been copied over to the disk. The installation program failed due to the architectural differences detected between the installation program's preconfigured list of acceptable target processors and the current system's unsupported processor.

Therefore, the installation was a complete failure and while it is known from Experiment II that RH5 will work and boot on the Pentium IV, it cannot be directly installed to that platform.

4.3.3 Testing the Pentium IV

Unfortunately, there was nothing to test because the installation was a complete failure. Nevertheless, the installation of RH5 was attempted several more times on the Pentium IV, always with the same results. Even making changes to the partitions and their sizes did nothing to resolve the problem.

4.3.4 Results

The results that can be drawn from this experiment are that while older Linux kernels can function on newer hardware and processors that they were not designed for, their installation programs may not have this capability or flexibility. Such was the case with this experiment. The installation program was totally incapable of installing the operating system to the newer platform. While there are methods of going around this, it defeats the purpose by doing so. Interestingly enough, the installation kernel on the RH5 CD-ROM was compiled for with support for the following processors:

- i386
- i486
- Pentium
- Pentium Pro

However, the installation program failed to correctly install the operating system components due to the fact the program had been hard-coded for specific processor sets instead of supporting more generic x86-based processors.

Although this experiment was conducted on RH5, it is currently unknown if other operating systems of similar age would behave in the same manner. It is quite conceivable that some of them may, although this cannot be verified at this time. Nevertheless, this is an inevitably in operating system installation program technology that has been uncovered. Installation programs therefore appear to be far more limited than the kernels they run under, and depending on how the installation program was written, it may not support more modern processors.

Therefore, the Navy, which has already clearly stated its interest in maintaining an older operating system, should seriously consider the advantages that can be leveraged by utilizing migration and reconfiguration. Thus, rather than attempt to install an operating system to a hardware platform that it may not support, by migrating it and performing an operating system hardware reconfiguration, the chances for success are very high if the kernel, drivers, and subsystems have been kept periodically up to date. This will enable the Navy to keep the same operating system with the same applications and settings, configurations, preferences, as well as other non-hardware operating system settings and configurations. Leveraging this can save a great deal of time which is particularly critical when deploying a C2 system that must be recertified by security/administrative personnel.

5. Results analysis

While analyzing the various results from the three experiments conducted, some general observations and possibly applicable trends are made here are as follows:

1. While a successful operating system hardware reconfiguration and hardware migration can be expected to/from newer/older hardware with more recent Linux distributions, there are two points that should be duly noted:

    a. In migrating/reconfiguring from a more recent system to an older system with an older operating system that is not currently supported because the compiled in processor options are not there, it will be necessary to recompile and reinstall the kernel with the appropriate processor-specific options on the newer system before migrating/reconfiguring to the older system.

    b. In migrating/reconfiguring from an older system to a newer one, in so long as the kernel, drivers, and required subsystems are adequately kept up to date the operating system will be successfully transferred to the newer platform.

2. It is possible to overcome certain architectural deficiencies in hardware through the use of more modern software. An exhaustive list is not available at this time. However, the case of the *GRUB* boot-loader is a good example to show that while the older hardware cannot normally boot beyond the 1024th cylinder, with the proper software, this issue can be circumvented.

3. The Linux kernel should normally be able to boot the operating system and provide basic I/O services even for processors and hardware that it was not originally designed to run on, provided that the newer system is backwards compatible. However, the amount of functionality will depend on the age and the amount of hardware support built into the kernel vis-à-vis the newer hardware platform itself.

4. Older Linux distributions do not generally have the ability to perform hardware reconfigurations or migrations and thus will only provide limited hardware support. In order to provide greater hardware support and ensure a successful migration and hardware reconfiguration, it will be necessary to update the kernel and subsystems. However, how this is done will vary from distribution to distribution, as will the current level of hardware support.

5. When performing migrations and hardware reconfigurations, there may be software in the operating system that may not be compatible with a specific hardware platform or component. Therefore, the risk, while minimal, is always there that incompatible software may have to be recompiled or updated to adequately support the new hardware. In some cases alternate software may have to be used to resolve the matter.

6. It is possible that significantly older distributions that are many revisions out of date may not install correctly onto newer hardware platforms. In such a case, it is necessary to install it to a supported platform where the system can be updated from (as required) in

order to support the newer hardware platform, and only then proceed with a migration and hardware reconfiguration.

6. Conclusion

Thus, it can be concluded that while Linux is a very flexible and robust operating system, it cannot be all things at all times to everyone, and as such it is not always possible to support hardware that did not exist when a given distribution and kernel were originally released. It is possible, however, to ensure compatibility with modern hardware by keeping the kernel and its subsystems up to date.

It is the recommendation of this report, that in order for the Navy to be able to maintain the same operating system throughout the lifetime of the C2 system, the kernel and its subsystems must be periodically updated so that a hardware migration or operating system hardware reconfiguration will be successful. A migration and hardware reconfiguration offer real advantages over performing a new and full operating system installation, including the reconfiguration of configuration files and system settings, as well as the security recertification and reaccredidation of the operating system on the newer hardware platform. This recommendation is equally applicable to not only the Navy but throughout DND/CF as well.

It cannot be overstressed that past capabilities of the Linux operating system may not be the same in the future. It is conceivable, as with any operating system that at some point in the future certain processor or platform support will be altogether removed from the kernel source code. For the Navy and the HMCCS project, hardware that it procures for the new C2 system will have been recent at the time of procurement, but that looking back some years later it no longer will be, and as such, the possibility that a future version of the kernel that no long supports that platform and hardware is a distinct possibility. However, by keeping hardware current through periodic hardware upgrades and overhauls, maintaining the kernel and subsystems and using hardware migration and reconfiguration, this issue can be mitigated.

Furthermore, the past is never a guarantee of future performance. As with any computer system and operating system, *caveat emptor*.

References

[1] Charpentier, Robert, Richard Carbone. Life-Cycle Support for Information Systems Based on Free and Open Source Software. Technical Paper for 11th ICCRTS. DRDC Valcartier. June 2006. http://www.dodccrp.org/11th_ICCRTS/abstracts/136.pdf.

[2] Carbone, Richard. Hardware Reconfiguration Methodology: A Linux case study. Technical Memorandum (Draft). DRDC Valcartier. Scheduled for completion December 2006.

[3] Carbone, Richard. Does Red Hat 5.0 Support Hardware Refreshes and can it Work on Modern x86 CPU's? Internal Report. DRDC Valcartier. November 2005.

[4] Carbone, Richard. Can Linux be Easily Reconfigured from One Machine to the Next?" Internal Report. DRDC Valcartier. October 2005.

[5] Carbone, Richard. A What to do Avoid Guide in doing your own In-House Migration. Internal Report. DRDC Valcartier. November 2005.

[6] Wikipedia. System call. Online encyclopaedia. Wikimedia Foundation Inc. October 2006. http://en.wikipedia.org/wiki/System_call.

[7] Wikipedia. Pentium II. Online encyclopaedia. Wikimedia Foundation Inc. November 2006. http://en.wikipedia.org/wiki/Pentium_II.

[8] Wikipedia. List of Intel Pentium 4 processors. Online encyclopaedia. Wikimedia Foundation Inc. November 2006. http://en.wikipedia.org/wiki/List_of_Intel_Pentium_4_microprocessors#Pentium_4.

Annex A Pentium II hardware specifications

A.1 Pentium II

The Pentium II-class system is a no-name brand computer purchased by DRDC Valcartier many years ago. The system is now outdated and was originally labelled for surplus. Its constituent makeup is described below.

A.2 Hardware components

The clone Pentium II system is specifically made up of the following components:

a) Microsoft two-button PS/2 wheel mouse

b) Microsoft compatible 104-key PS/2 keyboard

c) ViewSonic G90fb 19” monitor

d) Pentium II 300 MHz

e) 128 MB SDRAM

f) Fujitsu MPD306AT 6.5 GB IDE disk (primary master)

g) LS-120 (primary slave)

h) 28 X Panasonic CD (secondary master)

i) 3Com 3C905B TXNM XL PCI 10/100 network card

j) ATI 3D Rage Pro 2x AGP video card

k) Asus P2L97 motherboard

A.3 Linux hardware configuration file

The Linux hardware configuration file, as defined by */etc/sysconfig/hwconf*, is compiled by the system during installation. *Kudzu* has detected the following hardware in the Pentium II system:

class: OTHER

bus: PCI

detached: 0

driver: shpchp

desc: "Intel Corporation 440LX/EX - 82443LX/EX AGP bridge"

vendorId: 8086

deviceId: 7181

subVendorId: 0000

subDeviceId: 0000

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1

pcifn: 0

-

class: OTHER

bus: PCI

detached: 0

driver: unknown

desc: "Intel Corporation 440LX/EX - 82443LX/EX Host bridge"

vendorId: 8086

deviceId: 7180

subVendorId: 0000

subDeviceId: 0000

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 0

pcifn: 0

-

class: OTHER

bus: PCI

detached: 0

driver: i2c-piix4

desc: "Intel Corporation 82371AB/EB/MB PIIX4 ACPI"

vendorId: 8086

deviceId: 7113

subVendorId: 0000

subDeviceId: 0000

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 4

pcifn: 3

-

class: OTHER

bus: PCI

detached: 0

driver: unknown

desc: "Intel Corporation 82371AB/EB/MB PIIX4 ISA"

vendorId: 8086

deviceId: 7110

subVendorId: 0000

subDeviceId: 0000

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 4

pcifn: 0

-

class: NETWORK

bus: PCI

detached: 0

device: eth0

driver: 3c59x

desc: "3Com Corporation 3c905B 100BaseTX [Cyclone]"

network.hwaddr: 00:01:02:3D:68:BC

vendorId: 10b7

deviceId: 9055

subVendorId: 10b7

subDeviceId: 9055

pciType: 1

pcidom: 0

pcibus: 0

pcidev: b

pcifn: 0

-

class: MOUSE

bus: PSAUX

detached: 0

device: input/mice

driver: generic3ps/2

desc: "ImPS/2 Generic Wheel Mouse"

-

class: CDROM

bus: IDE

detached: 0

device: hdc

driver: ignore

desc: "MATSHITA CR-585"

-

class: VIDEO

bus: PCI

detached: 0

driver: Card:ATI Mach64

desc: "ATI Technologies Inc 3D Rage Pro AGP 1X/2X"

vendorId: 1002

deviceId: 4742

subVendorId: 1002

subDeviceId: 4742

pciType: 1

pcidom: 0

pcibus: 1

pcidev: 0

pcifn: 0

-

class: FLOPPY

bus: IDE

detached: 0

device: hdb

driver: ignore

desc: "LS-120 VER5 00 UHD Floppy"

physical: 0/0/0

logical: 963/8/32

-

class: HD

bus: IDE

detached: 0

device: hda

driver: ignore

desc: "FUJITSU MPD3064AT"

physical: 13410/15/63

logical: 13410/15/63

-

class: KEYBOARD

bus: PSAUX

detached: 0

driver: ignore

desc: "AT Translated Set 2 keyboard"

-

class: USB

bus: PCI

detached: 0

driver: uhci-hcd

desc: "Intel Corporation 82371AB/EB/MB PIIX4 USB"

vendorId: 8086

deviceId: 7112

subVendorId: 0000

subDeviceId: 0000

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 4

pcifn: 2

-

class: IDE

bus: PCI

detached: 0

driver: unknown

desc: "Intel Corporation 82371AB/EB/MB PIIX4 IDE"

vendorId: 8086

deviceId: 7111

subVendorId: 0000

subDeviceId: 0000

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 4

pcifn: 1

Annex B Pentium IV hardware specifications

B.1 Pentium IV

The Pentium IV-class system is a Dell computer system purchased in Spring 2003 by DRDC Valcartier. The system's hardware components are described below.

B.2 Hardware components

The Pentium IV system is a Dell OptiPlex Precision 360 desktop that is made up of the following constituent components:

a) Dell two-button USB wheel mouse

b) Dell Windows USB keyboard

c) ViewSonic E790 19" monitor

d) Pentium IV 2.4 GHz

e) 512 MB SDRAM

f) Fujitsu MPD306AT 6.5 GB IDE disk (primary master)

g) Samsung CD-RW/DVD SM-348B (secondary master)

h) Intel Corporation 82540EM Gigabit Ethernet Controller (integrated)

i) NVIDIA Quadro 4 128 MB 8x AGP

j) Dell OptiPlex 360 motherboard

k) Intel Corporation 82801EB/ER (ICH5/ICH5R) AC'97 Audio Controller (integrated)

B.3 Linux hardware configuration file

The Linux hardware configuration file, as defined by */etc/sysconfig/hwconf*, is compiled by the system during installation. *Kudzu* has detected the following hardware in the Pentium IV system:

class: OTHER

bus: PCI

detached: 0

driver: hw_random

desc: "Intel Corporation 82801 PCI Bridge"

vendorId: 8086

deviceId: 244e

subVendorId: 0000

subDeviceId: 0000

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1e

pcifn: 0

-

class: OTHER

bus: PCI

detached: 0

driver: i8xx_tco

desc: "Intel Corporation 82801EB/ER (ICH5/ICH5R) LPC Interface Bridge"

vendorId: 8086

deviceId: 24d0

subVendorId: 0000

subDeviceId: 0000

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1f

pcifn: 0

-

class: OTHER

bus: PCI

detached: 0

driver: i2c-i801

desc: "Intel Corporation 82801EB/ER (ICH5/ICH5R) SMBus Controller"

vendorId: 8086

deviceId: 24d3

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1f

pcifn: 3

-

class: OTHER

bus: PCI

detached: 0

driver: shpchp

desc: "Intel Corporation 82875P Processor to AGP Controller"

vendorId: 8086

deviceId: 2579

subVendorId: 0000

subDeviceId: 0000

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1

pcifn: 0

-

class: OTHER

bus: PCI

detached: 0

driver: unknown

desc: "Intel Corporation 82875P/E7210 Memory Controller Hub"

vendorId: 8086

deviceId: 2578

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 0

pcifn: 0

-

class: NETWORK

bus: PCI

detached: 0

device: eth0

driver: e1000

desc: "Intel Corporation 82540EM Gigabit Ethernet Controller"

network.hwaddr: 00:07:E9:54:E1:5F

vendorId: 8086

deviceId: 100e

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 2

pcidev: c

pcifn: 0

-

class: MOUSE

bus: PSAUX

detached: 0

device: input/mice

driver: generic3ps/2

desc: "PS/2 Generic Mouse"

-

class: AUDIO

bus: PCI

detached: 0

driver: snd-intel8x0

desc: "Intel Corporation 82801EB/ER (ICH5/ICH5R) AC'97 Audio Controller"

vendorId: 8086

deviceId: 24d5

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1f

pcifn: 5

-

class: CDROM

bus: IDE

detached: 0

device: hdc

driver: ignore

desc: "SAMSUNG CDRW/DVD SM-348B"

-

class: VIDEO

bus: PCI

detached: 0

driver: Card:NVIDIA Quadro 4 (generic)

desc: "nVidia Corporation NV18GL [Quadro4 NVS AGP 8x]"

vendorId: 10de

deviceId: 018a

subVendorId: 10de

subDeviceId: 0190

pciType: 1

pcidom: 0

pcibus: 1

pcidev: 0

pcifn: 0

-

class: HD

bus: IDE

detached: 0

device: hda

driver: ignore

desc: "FUJITSU MPD3064AT"

physical: 13410/15/63

logical: 13410/15/63

-

class: USB

bus: PCI

detached: 0

driver: uhci-hcd

desc: "Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI #3"

vendorId: 8086

deviceId: 24d7

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1d

pcifn: 2

-

class: USB

bus: PCI

detached: 0

driver: uhci-hcd

desc: "Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #1"

vendorId: 8086

deviceId: 24d2

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1d

pcifn: 0

-

class: USB

bus: PCI

detached: 0

driver: uhci-hcd

desc: "Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #2"

vendorId: 8086

deviceId: 24d4

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1d

pcifn: 1

-

class: USB

bus: PCI

detached: 0

driver: uhci-hcd

desc: "Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #4"

vendorId: 8086

deviceId: 24de

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1d

pcifn: 3

-

class: USB

bus: PCI

detached: 0

driver: ehci-hcd

desc: "Intel Corporation 82801EB/ER (ICH5/ICH5R) USB2 EHCI Controller"

vendorId: 8086

deviceId: 24dd

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1d

pcifn: 7

-

class: IDE

bus: PCI

detached: 0

driver: ata_piix

desc: "Intel Corporation 82801EB (ICH5) SATA Controller"

vendorId: 8086

deviceId: 24d1

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1f

pcifn: 2

-

class: IDE

bus: PCI

detached: 0

driver: unknown

desc: "Intel Corporation 82801EB/ER (ICH5/ICH5R) IDE Controller"

vendorId: 8086

deviceId: 24db

subVendorId: 1028

subDeviceId: 0156

pciType: 1

pcidom: 0

pcibus: 0

pcidev: 1f

pcifn: 1

List of symbols/abbreviations/acronyms/initialisms

API	Application Programming Interface
BIOS	Basic Input Output System
Caveat emptor	(*Latin*) Buyer beware
CD-ROM	Compact Disc-Read Only Memory
C2	Command & Control
DHCP	Dynamic Host Control Protocol
DMSS	Directorate of Maritime Ship Support
DND/CF	Department of National Defence/Canadian Forces
DRDC	Defence Research Development Canada
DVD	Digital Video Disc
FC4	Fedora Core 4
GB	Gigabyte
GRUB	GRand Unified Bootloader
GUI	Graphical User Interface
HMCCS	Halifax Modernized Command Control System
HT	Hyper-Threading
Hz	Hertz
I/O	Input/Output
IDE	Integrated Device Electronics
IPC	Inter-Process Communication
LILO	Linux Loader
MB	Megabyte
MBR	Master Boot Record
PCI	Peripheral Component Interconnect
PCMCIA	Peripheral Component MicroChannel Interconnect Architecture
PS/2	Personal System/2
RH	Red Hat
RH5	Red Hat 5
RHEL	Red Hat Enterprise Linux
SCSI	Small Computer System Interface
USB	Universal Serial Bus

Distribution list

Document No.: DRDC Valcartier TM 2006-595

LIST PART 1: Internal Distribution by Centre:

3 Document Library
1 Richard Carbone (author)
1 François Lemieux
1 Robert Charpentier
1 Michel Lizotte
1 Guy Turcotte
1 Yves van Chestein

9 TOTAL LIST PART 1

LIST PART 2: External Distribution by DRDKIM

DRDC Corporate HQ

1 Klaus Kollenberg (DSTC4ISR 3)
1 Donna Wood (DSTC4ISR 4)
1 Directorate R & D – Knowledge and Information Management

NDHQ (101 Colonel By, Ottawa, K1A 0K2)

1 Directorate of Maritime Support Systems
Attn: Mr. Norbert Haché
1 Directorate Information Management Strategic Planning 3
Attn: Maj. J.Perry Mellway
1 Directorate Information Management Strategic Planning 2-2
Attn: Maj. P.C. Kvas
1 Directorate of Aerospace Engineering Support 5
Attn: Sylvain Fleurant
1 Directorate Information Management Requirements 4-6
Attn: Mark Daniels
1 Directorate Distributed Computing Engineering and Integration 3-5
Attn: Capt. Karine Pellerin
1 Directorate Information Management Security 2-3-2-2
Attn: Paul Lamoureux

10 TOTAL LIST PART 2

19 TOTAL COPIES REQUIRED

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1\. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

DRDC Valcartier

2\. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

Unclassified

3\. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C, R or U) in parentheses after the title.)

(U) Operating system hardware reconfiguration: A case study for Linux

4\. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Carbone, R.

5\. DATE OF PUBLICATION (Month and year of publication of document.)

January 2007

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

58

6b. NO. OF REFS (Total cited in document.)

8

7\. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Memorandum

8\. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

DMSS HMCCS

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

1430JT 15AV34

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Valcartier TM 2006-595

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11\. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

(X) Unlimited distribution
(   ) Defence departments and defence contractors; further distribution only as approved
(   ) Defence departments and Canadian defence contractors; further distribution only as approved
(   ) Government departments and agencies; further distribution only as approved
(   ) Defence departments; further distribution only as approved
(   ) Other (please specify):

12\. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

(U) This memorandum explores how well a Linux system can withstand the test of time – specifically, whether it can withstand various hardware upgrades that would ensue over the coming years of its lifetime as the operating system for a command and control (C2) system. By understanding how Linux responds to changes in the hardware and how it can reconfigure itself, it allows for better understanding of the implications involved in maintaining a given operating system for the lifetime of the C2 system. Through several simple and repeatable experiments, it was determined that a given Linux operating system, if adequately maintained can successfully be used over the lifetime of the C2 system.

(U) Ce mémorandum étudie comment un système Linux résiste aux changements – particulièrement s’il peut résister aux changements périodiques de matériel sous-jacent qui permettrait d’assurer sa survie à titre de système d’opération d’un système de commandement et contrôle. En sachant comment Linux répond aux changements de matériel sous-jacent et comment il se reconfigure par lui-même, on peut comprendre les implications au niveau de sa maintenance pendant toute la durée de vie du système de commandement et contrôle. Suite à plusieurs petites expériences, on a déterminé qu’un système Linux donné, lorsqu’il est bien maintenu, peut être utilisé pour supporter un système de commandement et contrôle pendant toute sa durée de vie.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Linux, open source, reconfiguration, hardware reconfiguration, operating system hardware reconfiguration, migration, hardware migration, Red Hat, RH5, Fedora Core, FC4, Pentium II, Pentium IV, operating system updates, operating system upgrades

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE R&D DÉFENSE

WWW.drdc-rddc.gc.ca