

Regis University

ePublications at Regis University

Regis University Student Publications
(comprehensive collection)

Regis University Student Publications

Summer 2006

Unix Remote Access Via Ibm S/390 System

Selena Pena
Regis University

Follow this and additional works at: <https://epublications.regis.edu/theses>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Pena, Selena, "Unix Remote Access Via Ibm S/390 System" (2006). *Regis University Student Publications (comprehensive collection)*. 346.

<https://epublications.regis.edu/theses/346>

This Thesis - Open Access is brought to you for free and open access by the Regis University Student Publications at ePublications at Regis University. It has been accepted for inclusion in Regis University Student Publications (comprehensive collection) by an authorized administrator of ePublications at Regis University. For more information, please contact epublications@regis.edu.

Regis University
School for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

Use of the materials available in the Regis University Thesis Collection ("Collection") is limited and restricted to those users who agree to comply with the following terms of use. Regis University reserves the right to deny access to the Collection to any person who violates these terms of use or who seeks to or does alter, avoid or supersede the functional conditions, restrictions and limitations of the Collection.

The site may be used only for lawful purposes. The user is solely responsible for knowing and adhering to any and all applicable laws, rules, and regulations relating or pertaining to use of the Collection.

All content in this Collection is owned by and subject to the exclusive control of Regis University and the authors of the materials. It is available only for research purposes and may not be used in violation of copyright laws or for unlawful purposes. The materials may not be downloaded in whole or in part without permission of the copyright holder or as otherwise authorized in the "fair use" standards of the U.S. copyright laws and regulations.

REGIS UNIVERSITY

"UNIX Remote Access via IBM S/390 System"

PROFESSIONAL PROJECT THESIS

BY:

Selene Peña
SEAD 2005B
MSCIT Program

School for Professional Studies
Regis University
Denver, CO

February 27, 2006

Acknowledgments

I want to thank God for giving me the strength and motivation to complete this project and my studies at Regis University.

I would like to thank my husband, Omar Peña, for his tremendous support, love, and encouragement. Thank you, Omar, for helping with everything.

Thank you, Dr. James Lupo for all of your help, guidance, and advice. Thank you for always pointing me in the right direction, for taking time to answer all of my questions, and for always being so helpful. Your insight truly helped throughout this project.

Thank you, Mr. Likarish for the opportunity to participate and learn in the SEAD practicum. Especially for the chance of serving in the Administrative Lead position for the SEAD 2005B practicum. It was a terrific learning experience.

Thanks to all of the Regis students that I had the pleasure of meeting and working with.

Abstract

Business Problem

The MSCIT graduate program at Regis University offers UNIX and Information Assurance courses for all graduate students. In addition, the program gives students the opportunity to participate in the Systems Engineering and Application Development Practicum (SEAD) to complete their professional project/thesis. Due to the limited number of UNIX systems available, the graduate and SEAD students have a limited environment to work on.

The UNIX and Information Assurance classes share the Acadunix UNIX server to complete lab work and homework assignments. Some students have limited root access via “sudo.” Students cannot practice remote commands and can't use networking tools because another UNIX server is not available. This limits the environment and students' learning. There has been interest in having multiple UNIX boxes by some facilitators and students. In addition, having only one server is a risk, with the threat of data loss and lack of availability. If the system crashes or fails, students and faculty will be unable to complete their work and continue with the course productively. On the other hand, adding additional systems to the network increases the possibility of a system breach. Account management on all systems could become an issue, but it is a possible solution. A single sign-on solution could be an option as well.

Technical Solution

The proposed solution includes providing MSCIT graduate and SEAD Practicum students, access to multiple UNIX machines by using the virtual machines hosted on an IBM 390 system, while maintaining security and providing a method for maintaining student accounts. The IBM 390 virtual machines, as provided, have an applied security policy that prohibits access to outside networks. The virtual machines can only communicate with each other, and there is no method to manage accounts on the virtual machines.

Business Case

MSCIT Students will be able to have a more realistic learning environment by being able to access other systems remotely to test systems administration commands and security tools. This paper also offers suggestions for the use of the virtual machines on the IBM S/390 in a lab environment, and suggestions for future developments of this project.

Table of Contents

PROFESSIONAL PROJECT THESIS.....	1
CERTIFICATE OF AUTHORSHIP.....	2
ADVISOR APPROVAL PAGE.....	3
PROJECT PAPER REVISION HISTORY.....	4
ACKNOWLEDGMENTS	5
ABSTRACT.....	6
TABLE OF CONTENTS.....	7
LIST OF TABLES	8
LIST OF FIGURES / EXHIBITS.....	8
CHAPTER ONE: INTRODUCTION	9
PROBLEM STATEMENT	9
PROJECT RELEVANCE TO THE ACADEMIC RESEARCH NETWORK.....	9
PROJECT SCOPE AND LIMITATIONS	12
SUMMARY	13
CHAPTER TWO: REVIEW OF LITERATURE AND BACKGROUND.....	14
OVERVIEW	14
BACKGROUND.....	14
PROJECT STATUS	19
PROJECT CONTRIBUTION	21
SUMMARY	22
CHAPTER THREE: METHODOLOGY.....	23
RESEARCH METHODS	23
DELIVERABLES AND EXPECTED RESULTS	27
PROJECT TASK PLAN.....	28
SUMMARY.....	28
CHAPTER FOUR: RESULTS AND CONCLUSION.....	30
PROJECT DISCUSSION.....	30
EVALUATION OF PROJECT GOALS	31
ANALYSIS RESULTS.....	33
SUMMARY	42
CHAPTER FIVE: LESSONS LEARNED.....	44
THE PROJECT EXPERIENCE.....	44
NEXT STAGE OF EVOLUTION.....	45
CONCLUSION.....	45
CHAPTER SIX: BACK MATTER	47
REFERENCES	47
DEFINITION OF TERMS	48
APPENDIX A QUESTIONNAIRE & RESULTS COMMENTS	49
APPENDIX B REMOTE ACCESS AND ACCOUNT SETUP PROCESSES	53
APPENDIX C ACCOUNT SETUP AND REMOVAL SCRIPTS.....	61

APPENDIX D ACCOUNT SETUP TEST PLANS.....	66
APPENDIX E PROJECT CHECKLISTS	68
APPENDIX F STUDENT AND ADMINISTRATOR LAB MANUALS	72

List of Tables

Table 1 Installed Tools	20
-------------------------------	----

List of Figures / Exhibits

Figure 1 Systems Development Life Cycle.....	24
Figure 2 Project Task Plan	28
Figure 3 High Level Inputs & Outputs.....	29
Figure 4 Revised Project Task Plan	31
Figure 5 Need multiple Unix systems	34
Figure 6 Benefit multiple Unix systems.....	35

Chapter One: Introduction

Problem statement

Regis MSCIT educational programs have limited resources to work on UNIX or UNIX related projects. The goal of this project is to make additional UNIX systems available by using virtual machines in order to provide students with a lab environment in which they could work on projects, assignments or simply for testing and practice. Regis University has been given access to the virtual machines that are running on the IBM S/390 system. The proposed solution includes providing access to these virtual environments, while maintaining security and providing a method for managing student accounts.

Project relevance to the Academic Research Network

Currently, MSCIT students only have access to one UNIX server, named Acadunix. Other UNIX boxes exist, but are not available for use by Regis graduate students. Personal experience as a MSCIT graduate student provided a first hand look at the issue of lacking the adequate environment to allow students to thoroughly work on assignments and gain experience in a networked UNIX server environment. Specifically, because there is only one UNIX system available, students are not able to use basic network utilities such as FTP, rsh, rlogin, ssh and scp. Also, basic network and systems administration hands-on learning was limited. For example, getting experience with NFS networked file system is not possible when using a stand alone box. Hands-on UNIX network security work was also limited because any change made on the Acadunix system could negatively affect the other users. For example, using SSL caused performance issues, making the system very slow for everyone trying to use it. In addition, having only one server is a risk, with the threat of data loss. If the system crashes or fails,

students and faculty will be unable to complete their work and continue with the course productively. Moreover, the number of projects dealing with UNIX administration and security are limited for the Systems Engineering and Application Development (SEAD) Practicum students.

This year, the SEAD practicum was granted access to the virtual machines running on an IBM S/390 system by Colorado State University. This presents an opportunity to provide multiple UNIX machines for students to work on, and provide other testing and development benefits.

One issue is having more than one Unix box and giving access to more people increases the possibility of compromise and means having to manage separate accounts on each additional system. Another issue is that the virtual machines on the IBM S/390 do not have connectivity to the outside world, and the virtual machines can only communicate with each other via their private network. The virtual machines can only be accessed by ssh, and no outbound connectivity is allowed. Access is allowed via an ssh client (i.e. Putty for Windows) or via ssh from the Acadunix box. Once logged into one of the virtual machines, access to any of the other virtual machines is possible by using their local IP address and the normal SSH port. In order to be able to login to the other virtual machines, the user must have an account setup on each of the virtual machines (VMs).

An approach to providing user access to the virtual machines will have to be determined by analysis, research and testing. Below are the current possible approaches.

Using generic accounts

Students can be given access to all virtual machines by using generic accounts with the proper

permissions and environment setup. Since ssh is already setup on the virtual machines, and the assumption is that students can have access to at least one of the virtual machines; then they can login to all others as long as their generic accounts are setup. The passwords would need to be changed at the start of every new term.

Using customized student accounts

Students can be given access to all virtual machines by using their own account and username. This means setting up all student accounts on each virtual machine. Since ssh is already setup on the virtual machines, public and private keys can be used for password-less authentication, if needed for administration tasks. This approach would require way to manage creation & deletion of accounts at the start of every new term.

Single sign-on solution

Account management by using NIS, Kerberos or LDAP. This approach will require research and testing.

Lab Environment

Since no outbound access or communication is allowed, these VMs could possibly provide a substantial lab environment. These VMs could essentially be used as crash and burn boxes for projects, testing, student assignments and troubleshooting break/fix issues. If any one of the virtual machines cannot be fixed, the simple solution would be to re-image the machine without affecting any production environment or any other VM. Re-imaging of the VMs is done by the Colorado State University (CSU) faculty and graduate students.

Root Access

Students can use Sudo to obtain root access. The sudo utility allows a user to run commands as another user. It logs all commands that are run and it is great for user access and change control. Sudo is already installed on the VMs.

Project Scope and Limitations

The project scope is everything and anyone that is relevant to this project. The items that are in scope include:

- MSCIT Graduate Facilitators
- MSCIT Graduate Students
- UNIX and Information Assurance course work and projects
- SEAD Practicum projects

The items that will not be considered for this project are:

- Undergraduate Programs
- Non-UNIX and/or non-Security related courses

The project restrains include:

- Little knowledge and experience with the IBM S/390 system. Since the author does not have extensive knowledge of the IBM S/390 functionalities, a portion of the research for this project will be dedicated solely to increasing this knowledge.
- Junior Unix administration skills. Although the author has systems administration skills,

more advanced skills could help this project advance quicker and/or develop more technical and elaborate solutions.

- Limited time frame to complete research and project. The usual time allotted for completion is six months. Due to additional SEAD practicum responsibilities, additional time may be required to complete this project.
- Other people may be using the virtual machines. The virtual machines are only on lease from CSU, and other faculty or students may also have access.
- Have to rely on CSU faculty to reset VM instances. Regis University does not own the IBM S/390 system. Additional time and communication with CSU faculty is required for resetting, upgrading or rebooting the VM instances.
- Unknown time granted on the VMs. Again, since the VMs are only on lease to Regis University, the VMs may or may not be available after the completion of this project.

Summary

In summary, limited resources to work on Unix or Unix related projects highlight the need for additional UNIX systems for MSCIT educational programs. Since access to the IBM S/390 system was granted to the Regis SEAD Practicum, there is an opportunity to use these virtual machines as a lab environment for the UNIX and Information Assurance classes, as well as the SEAD practicum members. This project will attempt to determine the best solution for giving students access to all virtual machines, while maintaining a secure and manageable environment, and a providing a useful lab environment.

Chapter Two: Review of Literature and Background

Overview

During the last few years, academic and commercial researchers have been drawn to the benefits that virtual machines can bring to an academic environment and to a professional commercial environment. "Using the ability of z/VM to host multiple virtual servers running a variety of guest operating systems, mainframes such as IBM's S/390 and zSeries™ are now being used to create virtual server farms" (Turk and Bausch, 2005). With Linux used as the guest operating system, other open-source software is available to run on this virtual server. This chapter presents the literature research that is relevant to this project, and is divided into sections. The first section discusses the definition of a virtual machine, provides a brief history of the IBM S/390 system, gives an explanation of Linux and how it operates on the virtual machines that are running on the IBM S/390, and looks at the use of virtual machines in an academic environment. The next section gives a summary of what is known and unknown about this project. Specifically, it covers how the S/390 VM is setup, lists installed tools, explains the current access method, and presents what is not known. The last section states the contribution this project will make to the field.

Background

What is a virtual machine?

First, a virtual machine is defined as "a self-contained operating environment," which allows for sharing of power and resources from a mainframe and operates as an independent

system (Virtual Machine). In other words, the virtual machine is not the actual machine, but behaves exactly like the actual machine. Virtual servers use shared resources, such as memory, processing, communication, I/O, and networking resources. Thus, Turk and Bausch point out that virtual servers are superior to physical server farms in terms of the efficient use of resources. Because of their ability to share resources, virtual servers help solve several problems. For instance, virtual machines allow computers to share the same hardware among many programs by partitioning the hardware. Also, virtual machines allow software to be portable between various operating systems (Kohlbrenner, E., Morris D., & Morris, B., 1999). On computational platforms such as the S/390 and the z/Series, the hardware supports logical partitioning of the machine into several distinct and isolated regions that operate independently of one another (Turk and Bausch, 2005).

History of the IBM S/390 System

Next, we discuss the history of the S/390. The S/390 is IBM's large server, or mainframe, line of computer systems with a history that stretches back to the System/360 of the early 1960s. "IBM first developed an implementation of its virtual machine around 1965 after IBM researchers came up with the idea of using a virtual machine for its experiment" (Kohlbrenner, E., et al., 1999). This machine is known as the IBM System 370 (S/370) and the IBM System 390 (S/390). These machines are all referred to as IBM's VM lineage and they use the IBM VM/ESA operating system. IBM's VM/ESA operating system can control multiple virtual machines on an IBM S/390 system (Virtual Machine definition). This enabled users to create and manage multiple copies of different Virtual Machines on a single computer (Kohlbrenner, E., et al., 1999).

Kohlbrenner, E., et al. state that:

This implementation of virtual machines has proven to be quite powerful, as it is a machine that is still being sold today. The idea of virtual machines has since been used to solve many other computing problems. Problems such as computer program portability and operating system design. (1999).

The IBM S/390 used in this project is running the Linux operating system on its virtual machines.

Linux on the Virtual Machines of the IBM S/390

Linux is an operating system that resembles the Unix operating system, which includes a graphical user interface (GUI), an X window system, TCP/IP and the Emacs editor. Linux was initially developed by Linus Torvalds as a graduate student at the University of Helsinki in Finland. Linux itself is the core of the operating system, known as the kernel. The kernel handles the basic functions of the operating system such as memory allocation, process allocation, and device input and output (Kernel definition). "Linux is part of the Open Source Initiative. The open source ideal is that software can be freely read, redistributed and changed" (Kreuter, 2000). Because Linux is open source, people have been developing operating systems based on the Linux kernel. These are called the "Linux distributions" (Linux). The most common Linux distributions include Red Hat, Caldera, and SuSE. "Noncommercial distributions of Linux such as Debian are less well known and haven't reached the same scale of popularity" (Shah, 2001, p. 4). "Linux comes in versions for all the major microprocessor platforms including the Intel, PowerPC, Sparc, and Alpha platforms" (Linux definition).

Linux on S/390 has been generally available since early 2000 from either IBM developerworks or from third party organizations, such as SuSE, RedHat and TurboLinux (Kreuter, 2000).

The IBM web page “Fast Facts” and Turk and Bausch explain different ways to run Linux on S/390 and zSeries. These methods are listed as follows:

- Linux can run natively. This means Linux can run on the entire machine, with no other operating system.
- Linux can run in a logical partition (LPAR). The zSeries hardware can be logically partitioned into a maximum of 15 separate LPARs (or up to 30, depending on the hardware). In this configuration, no virtual machines are created; Linux is running natively, and not as a guest operating system on a virtual machine.
- Linux can run in z/VM Guest Support, where Linux can run as a virtual machine using z/VM. A customer running z/VM can have hundreds of Linux systems running on a single zSeries. This is the current configuration of the S/390 system used for this project.

Virtual Machines in an Academic environment

In their paper, Turk and Bausch describe their experience at Colorado State University running hundreds of virtual Linux servers on an IBM S/390 mainframe with the z/VM operating system. They focus on three types of problems that were encountered: security issues,

performance issues and administration issues related to managing Linux instances. The security issues included account (and password) maintenance, being careful not to have identical ssh keys when building clones of virtual machines, and taking action to avoid security exposure due to the delay in getting software updates. The performance issues included cron jobs and CPU-intensive jobs running at the same time across a large number of (cloned) virtual machines. Finally, Turk and Bausch develop their own script as a tool for automating administration tasks such as updating software and accounts on a large number of virtual Linux servers. This paper directly relates to the use of the S/390 in an academic environment.

Moreover, Stockman states that instruction in the area of computer networking (specifically systems administration) can be cumbersome and ineffective; and is almost always is an expensive prospect when it comes to instructional and lab facilities (2003). **Stockman** points out problems that arise both in the classroom and the lab when trying to recreate a true computing environment. To help solve these problems, two solutions are offered in **the** paper. The solutions are virtual machines and remoting technology, such as a Virtual Private Network (VPN). "Virtual machines allow for displaying multiple operating systems interacting on a single screen, while remoting provides access to the physical machine on which those virtual machines reside" (Stockman, 2003).

In addition, in his article, Kreuter says that an ideal setting for Linux on VM is in the academic community, where hundreds of Linux machines can be easily created, deleted, or modified from a single point of control (2000). "Students can be assigned a Linux machine during a single semester and at the conclusion of the course the Linux/VM can be deleted within seconds" (Kreuter, 2000).

Project status

James Lupo, a Regis University affiliate faculty member, provided much of the initial information of the current state of the virtual machines for this project. There are seven (7) virtual machines that are available for use by Regis University, and are managed by the Colorado State University (CSU) faculty. They are accessible by only the root user via ssh from any Unix or Windows system. The virtual machines were setup by CSU; therefore they already have several tools and utilities installed. This section covers the current state of the virtual machines in more detail.

Virtual Machine Setup & Operating System

The seven (7) virtual machines are setup in a private network.

System Name: linux390.colostate.edu

Operating System: SuSE Linux Enterprise Server (SLES) 8 distribution

Linux kernel: 2.4.21

IP Address: 204.146.24.13

Local IP Addresses (private network) for the seven virtual machines:

10.10.102.157
10.10.102.158
10.10.102.159
10.10.102.160
10.10.102.161
10.10.102.162
10.10.102.163

Installed Tools/Utilities

Tool/Utility	Function
SSH, SCP	OpenSSH. SSH client (remote login program), SCP (secure copy program)
Sudo	Execute a command as another user
Perl	Dynamic web serving language
Apache	Web serving
cp, mv, rm, chmod, cat	File access utilities
ls, cd, mkdir, rmdir	Directory access utilities
dd, tar, gzip, gunzip	Archiving/compressing utilities
mount, umount, mkswap, swapon, fsck, finger, who, vi	System utilities
ifconfig, route, netstat, ping, traceroute	Network utilities
Bind	Domain Name Server
Regina	REXX language
Imap	Internet Email Server
OpenNLDAP	Internet Directory Server

Table 1 Installed Tools

Access Method

The virtual machines can only be accessed by SSH, but no outbound connectivity is allowed. They can be accessed by logging in via SSH from Acadunix, any Unix/Linux system, or Windows system. The port must be specified with the '-p' option to ssh, and by adding 5000 to the last octet of the IP address. Only the external entry point does port mapping. For example, to login to 10.10.102.162 from a Unix system, you must type:

```
ssh -p 5162 root@linux390.colostate.edu
```

From a Windows system, the Putty ssh client can be used to start an ssh session from the dialog box or from the command line:

```
putty.exe -ssh -P 5158 root@linux390.colostate.edu
```

The virtual machines can only communicate with each other via their private network. Once you are on one of the virtual machines, you can access any of the others by using their local IP address and the normal SSH port. For example,

```
ssh root@10.10.102.160
```

As mentioned earlier, only the entry point of the virtual machines does port mapping. Port mapping is the process where packets arriving to a particular IP address/port can be translated and then redirected to a different IP/port. Port Mapping is only necessary for incoming connections, not returning traffic (Kerio Technologies).

Unknowns

At this time, the in-depth details of the configuration and setup of the virtual machines is not known. The requirements for a lab environment are also not known. Further hands-on investigation and analysis to gather more information will be done during the research portion of this project. In addition, the virtual machines have been leased to Regis University for an undefined time period. So the future use of the virtual machines may or may not be possible.

Project contribution

This project will contribute additional knowledge in implementing and using virtual machines as a solution for a lab environment for academic use. Specifically, this project can contribute information and knowledge about using the IBM S/390 as a solution for having multiple Unix-like systems. Additionally, this paper will provide information on using the VMs in an academic lab environment and may provide ideas for future work.

Summary

To summarize, the virtual machine has existed since the 1960s, and has facilitated the sharing of resources in a computing environment, such as hardware, memory, power and networking resources. Over time, this has helped solve other problems such as computer program portability and operating system design. With the development of the Linux operating system by Linus Torvalds and the Open Source Initiative, Linux now runs on the IBM S/390 mainframe. The z/VM operating system on the S/390 allows for hundreds of virtual machines to run on a single server. With this capability, those in the academic community have taken notice and recommended using virtual machines in an academic environment. This project will intend to make the seven S/390 virtual machines available to Regis University students, and to contribute additional knowledge about using virtual machines in an academic environment.

Chapter Three: Methodology

Research methods

The research methods to be used for this project are experimental and problem solving. Hands-on analysis and development of a solution will be done. The System Development Life Cycle (SDLC) will be used to divide the project into separate phases. Many SDLC models have been developed due to the need for flexibility for different types of projects. The most common SDLC models include the waterfall method, the fountain method, the spiral method, build and fix, rapid prototyping and the incremental model (Kay, 2002). For this project, the SDLC will be adapted to the needs and requirements of the project. The project will be broken down into the following SDLC phases.

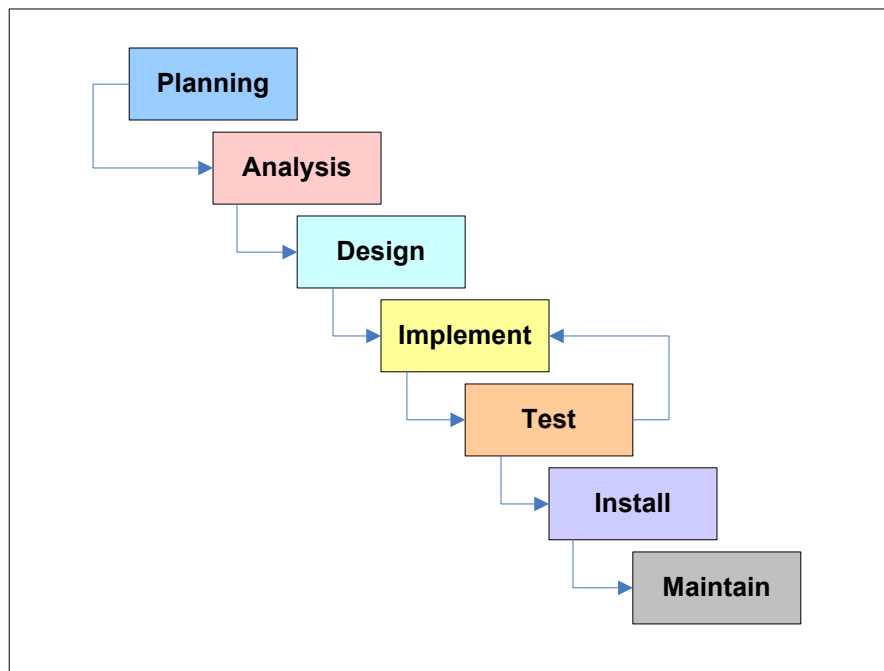


Figure 1 Systems Development Life Cycle
Selene Peña, 2005

Phase1 - Planning

In the Planning phase of the project, a high level view of the project will be established and the goals of the project will be determined. The project proposal will be used as the basis of the project and will be further developed into each one of the SDLC phases. UNIX and Information Assurance instructors will be contacted to inform them of this project and ask for participation. Then, a questionnaire will be developed and emailed to the instructors in order to determine the requirements and/or needs of the virtual lab environment.

Phase 2 – Analysis

The Virtual Lab Requirements Questionnaire will be emailed to previously contacted instructors. The purpose of this questionnaire is to obtain information about the requirements of a virtual lab for the UNIX and Information Assurance classes at Regis University. The completed questionnaires and responses will be analyzed and used for developing a solution.

In addition, the virtual machines running on the IBM S/390 system will be evaluated to investigate the unknowns about the current setup, and understand how VPNs work on the IBM 390. Another task is to find the strengths and limitations of the virtual machines, as well as understanding how network services are configured and controlled.

Most importantly, this phase consists of determining the best method to give students access to virtual machines and how to manage student accounts. All research findings will be documented in a project journal. The analysis results will be presented in chapter 4 of this paper.

Phase 3 - Design

The solution for remote access by students will be designed, including creating an access procedure, process diagrams, and documentation. Also, a process for setting up accounts on the virtual machines will be created. This will include designing a script and writing pseudo code for the script. A shell script, called `account.sh`, was developed by James Lupo to create accounts on Acadunix. The `account.sh` script will be used as a guideline for the script that will be developed in this project. In order to test the functionality of the account setup process and the script, a test plan will be designed and used during the testing (Phase 5) phase of this project.

Furthermore, two lab manuals will be created. The first lab manual is to be used by an administrator to setup student accounts and related tasks. The second manual is to be used by students to login to the virtual machines and perform exercises related to system administration.

The sample exercises in the student manual will cover:

- Logging In and Out
- Using the Command Line
- File Ownership
- File Permissions
- Viewing & Managing Processes
- Using Remote Access
- Using the Manual Pages

Phase 4 – Implement

At this phase, the design of the solution will be implemented by converting the script pseudo code into working code and by documenting the remote access process. The tools available on the virtual machines to complete this include Perl and Shell programming languages. All other process documentation and lab manuals will be revised to complete final drafts.

Phase 5 - Test

Next, the account setup script will be tested by following the test plan, and errors/problems will be fixed at this phase. If possible, students will be asked to test the student lab manual for functionality and readability. The communication to virtual machines from Acadunix will be verified as well. As indicated in Figure 1 SDLC, implementation and testing will follow a loop until satisfactory test results are reached. This phase will help determine if the project requirements were met.

Phase 6 – Install

The final working script will be put into the production environment of the virtual machines and the final drafts of all documentation will be published in Chapter 6 of this paper.

Phase 7 - Maintain

The final phase of this project will evaluate the solution to validate if the project met the initial project expectations and check if any part of the project could have been done differently. Lessons learned and suggestions for future work on this project will be given and documented in Chapter 5 of this paper.

Deliverables and Expected Results

The deliverables of this project include:

- Solution for remote access
- Account creation process with script
- Analysis results
- Testing results
- Requirements met evaluation (with test plans)
- Project paper
- Lab Manual
- Power Point Presentation

The success of the results for each phase will be measured by evaluating task completion for each phase. The success of the deliverables will be measured by determining if the requirements and goals were met. The analysis results data will be presented in the most appropriate format including summaries, tables, charts, and/or figures. The script, procedures and lab manuals will be published as appendices in this paper. A completed draft of this paper will be submitted for advisor approval, followed by a Power Point Presentation of the project.

The expected results include learning the requirements of a virtual lab environment for

Unix/Information Assurance classes, and providing a solution to give students access to all seven virtual machines. Writing a script will be required for the solution. Due to lack of advanced scripting skills, and the allotted time frame, a working script may not be completed on time. The alternative is to propose the access a method and pseudo code for future work.

Project Task Plan

The Gantt Chart in Figure 2 Project Task Plan provides a timeline for each phase of the project.

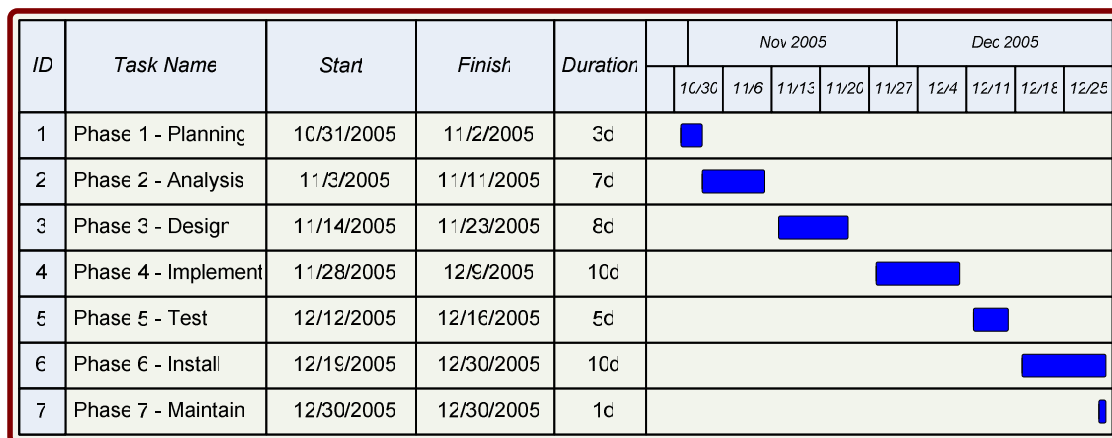


Figure 2 Project Task Plan
Selene Peña, 2005

Summary

In summary, experimental and problem solving methods will be used, as well as hands on approach for execution and completion of this project. The project will be sectioned into phases by utilizing the Systems Development Life Cycle (SDLC). Different parts of other SDLC models may be considered for specific procedures required for this project. The goal is to accomplish the set of deliverables and reach the expected results. If the goals are met and the tasks are completed, then the deliverable has been accomplished. Below is a high level diagram

of the inputs and outputs process for this project.

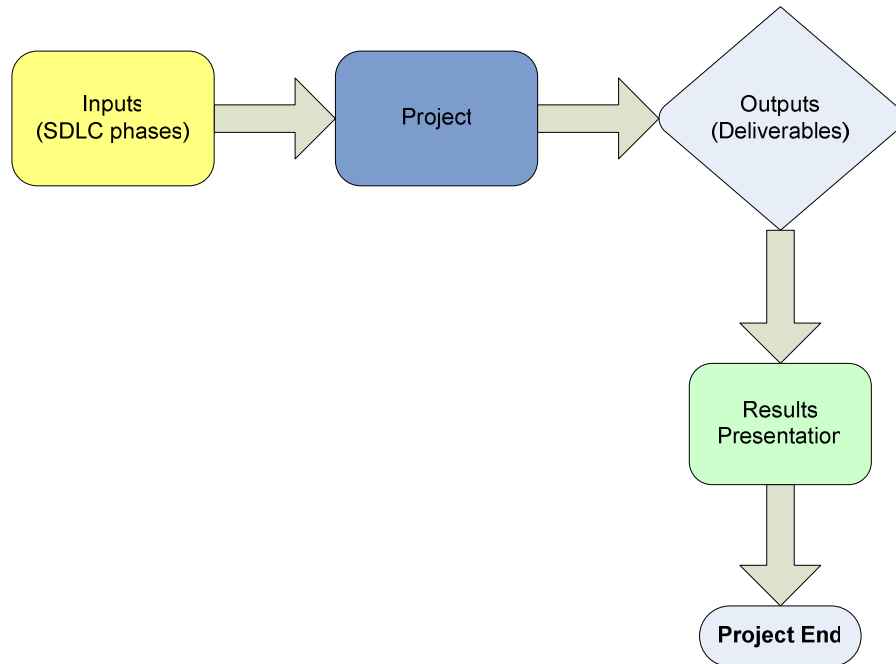


Figure 3 High Level Inputs & Outputs
Selene Peña, 2005

Chapter Four: Results and Conclusion

Project Discussion

Interest in the project began when the topic area was announced at an NLP practicum meeting. (Note: The NLP is now called the SEAD Practicum). I had been looking for a project to work on that would be challenging and interesting, and that could be done as part of the practicum. As an MSCIT student, I saw this as an opportunity for students to be able to work on other systems, and not be limited to use only one system. I also saw this as an opportunity to obtain more experience with a different flavor of Linux (SuSe) and thought it would be an excellent way to improve my systems administration skills. I had experienced the frustration of not being able to do more in my Regis UNIX courses, due to the availability of only one system for labs and homework. Discussions with faculty members provided plenty of information regarding the virtual machines that are running on the IBM S/390, which gave a nice start to the project.

The project was managed according to the guidelines provided by Mr. Dan Likarish and by using the Systems Development Life Cycle (SDLC) to divide the project into manageable phases. An SDLC completion checklist was developed to keep track of each task that was completed for each phase of the project. A virtual machine (VM) evaluation checklist was also created and used during Phase 2 of the project. A journal of project notes, questions, results, and reminders was kept to track project status as well.

Phase 2 (Analysis) of the project was a significant event because it was the official start of the project, in which a questionnaire was emailed to instructors and the responses were analyzed. Completion of this task gave helpful insight to the needs and preferences of the instructors for their particular courses. The next significant milestone was the completion of the

production version of the account setup script, which successfully allows an administrator to remotely create user accounts and home directories on all seven virtual machines.

Phase 4 (Implement), Phase 5 (Test), and Phase 6 (Install) of the project required more time than originally allocated. The original project task plan was changed to provide an additional six weeks for the final phases. Figure 4 shows the revised project task plan.

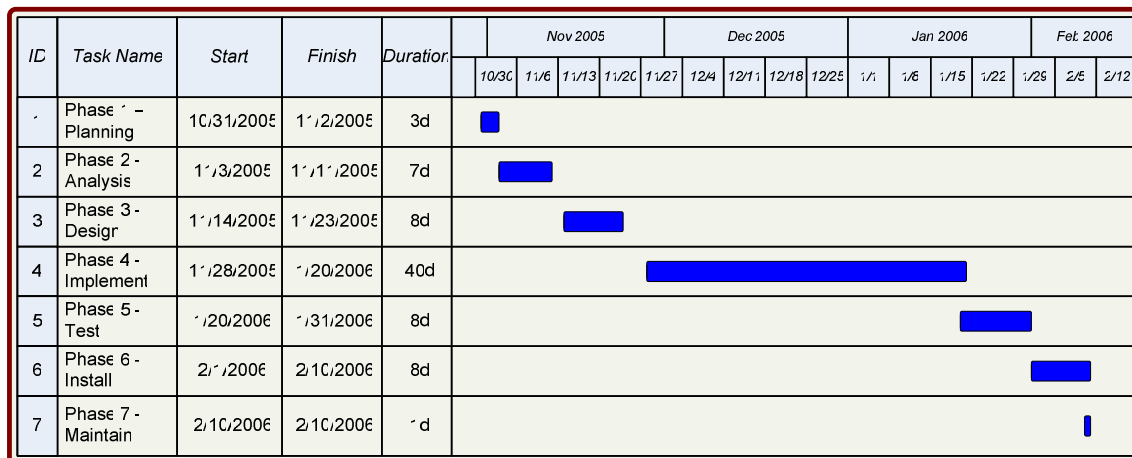


Figure 4 Revised Project Task Plan
Selene Peña, 2006

Due to the delay in the Test phase, the test of the student lab manual was done by the author and not by actual students. The goal was to create a lab that could be tested by students during their classroom lab time and provide feedback. This task will be suggested for future work of this project.

Evaluation of project goals

The goals for this project were met with all of the deliverables accomplished. To understand the IBM S/390, research was conducted and an evaluation of the virtual machines was done. In order to determine if the S/390 could be useful to MSCIT courses, a questionnaire

was distributed to selected Regis instructors. Their responses were compiled and analyzed, which provided information on what instructors thought the requirements of a virtual lab should be, if the S/390 virtual machines were available for them to use in their classes. This background knowledge and information provided a good basis for designing a solution.

A solution for remote access was chosen from the proposed approaches. The solution was to provide access by creating customized student accounts and a process with instructions on how to login remotely to the virtual machines. This solution included designing and creating a script that could create customized student accounts on all seven virtual machines. The account setup script was written in Perl and tested on the virtual machines by following the test plans. Once the script successfully completed every item in the script test plan, the process for creating accounts was developed. The account creation process was also tested with a test plan to ensure it provided correct instructions of how to prepare the environment for new accounts, and how to use the account creation script. The account creation process included several preliminary steps in order to be able to execute the account creation script and successfully create accounts on all seven virtual machines. The test plan for this process also required that the preliminary steps were all completed correctly and before any further testing could be done. Analysis and testing results were documented in this paper. The administrator lab manual includes the remote access process and the account creation process. Two additional deliverables were completed, which were not in the original list of deliverables. An extra script was created to help remove/delete accounts, and an extra lab manual was created for students to login to the virtual machines and work through a few suggested exercises, which includes using remote commands and other utilities.

These are the deliverables that were accomplished for this project:

- Solution for remote access
- Account creation process and script
- Account removal script
- Analysis results
- Testing results
- Requirements met evaluation (with test plans)
- Project paper
- Administrator and Student Lab Manual
- Power Point Presentation

Planning, research, and guidance from James Lupo provided a great deal of helpful information and insight to begin the project. My previous experience with account administration for Acadunix provided excellent insight and background for this project. For example, prior knowledge of the format of the student roster helped during the design of the account setup script. The roster file had to be processed first, before the script could read the contents.

Serving as the Administrative Lead for the SEAD 2005B practicum provided excellent project management experience, but also limited the amount of time to be dedicated to this project. The additional position responsibilities took precedence during the implementation and testing phases of the project. Therefore, the project plan was extended to allow more time for the project completion.

Analysis results

The analysis results are presented here in two sections. First, the results of the questionnaire are presented. Second, the results of each phase of the project (SDLC) are presented, including the results for testing the access procedure and script solutions.

Questionnaire results

Seven graduate instructors received the "Regis University Virtual Lab Requirements Questionnaire", which can be found in Appendix A. Only five out of seven instructors responded. The analysis is based on the five responses that were received.

Eighty percent of instructors agreed that there is a need for multiple UNIX/Linux servers for UNIX and Information Assurance classes. Twenty percent (one instructor) strongly disagreed, and zero percent disagreed and/or had a neutral opinion. Figure 5 below represents these responses.

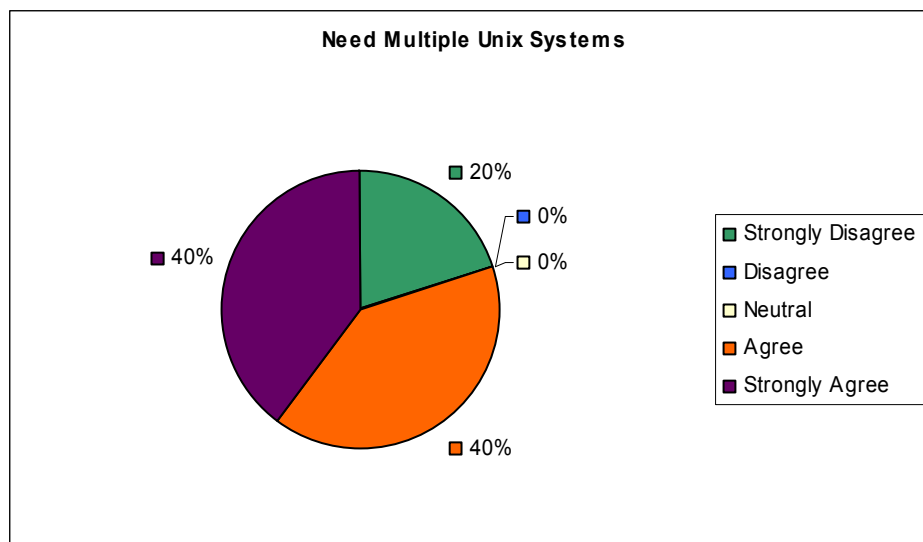


Figure 5 Need multiple Unix systems

Eighty percent of instructors that responded agreed that students would benefit from using more than one UNIX/Linux server to do their work. Twenty percent (one instructor) strongly disagreed. None of the instructors disagreed or had a neutral opinion about this

question. Figure 6 below represents these responses.

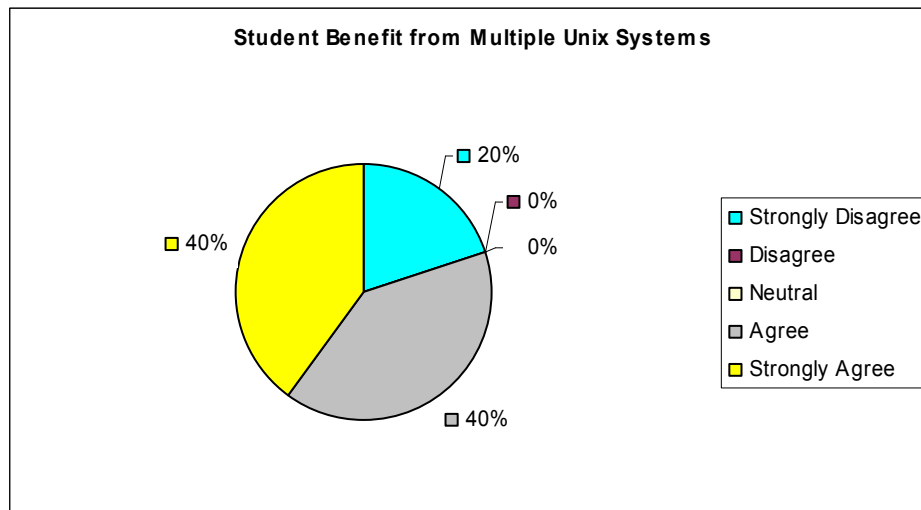


Figure 6 Benefit multiple Unix systems

Sixty percent of instructors thought that access to remote systems was not essential for students to learn the course objectives. Forty percent firmly thought that access to remote systems would be important for learning course objectives, and expressed that a system without a network is not a true system. Although the majority of instructors expressed remote system access was not vital, several said that it would be valuable to be able to demonstrate remote administration tasks such as remote logins, network security, and remote web and database applications. They also coincided that by having several machines available; students can have a more "real-life" learning experience.

The next question asked was, "If the virtual machines running were accessible, would you consider using them for your class?" One hundred percent of instructors answered "yes". Instructors were asked to explain how they would incorporate the virtual machines into their classes if they answered "yes". Below is a summary of how the virtual machines would be used.

- For the networking and security portions of the UNIX class

- To create networked mirrored servers
- To build firewalls and remote DB/Web applications
- To use remote shell and SSH commands
- To have students detect vulnerabilities
- Assign lab exercises to observe existing conditions and see differences in conditions of the appliance.
- For troubleshooting
- For file transfer and cross mounted file systems
- To see logical interfaces, run tools and scan traffic

Next, instructors were presented with a list of tools and utilities that are currently installed on the virtual machines. They were asked to mark the tools they were already using in their classes, and list any additional required tools. One hundred percent said they needed gawk, perl, sudo, and sed from the installed tools. They also said they needed basic system utilities such as ssh, scp, mount, vi, tcl/tk, nfs, finger, who, rwho, ksh, sh, awk, rsh, rcp, ftp, and telnet. Also php, cgi, terminal server, and security tools were listed as additional utilities needed. The basic system utilities mentioned are already available on the virtual machines.

Finally, instructors were asked to list any specific functionality required on the virtual machines for class instruction or student learning, and if their students would require root or super user capabilities. Sixty percent of the instructors said their students would need root access via sudo. One of the responses said that having the ability to configure tools and services and turn them on and off would be required. The need for accounts on several machines was expressed, in order for students to be able to use all virtual machines. Since there is no outside access to the Internet, where most open source security tools can be downloaded from, any rpm

or tar file would have to be copied from another system that has communication to the outside world such as Acadunix or a PC.

SDLC Phase Results

All steps of the planning phase (phase 1) of the project were completed on time. In the analysis phase (phase 2), the responses to the questionnaire were analyzed, an assessment of the virtual machines on the IBM S/390 was done, and the best method for giving students access to the VMs was chosen. The analysis of the questionnaire was presented in the previous section, *Questionnaire Results*. The assessment of the virtual machines was done by following the "VM Evaluation Checklist". After evaluating the virtual machines, it was discovered that some of the system utilities are disabled. The daemons are not running for telnet, rsh, and ftp; and they are disabled in /etc/inetd.conf. Therefore, the telnet, rwho, rsh, rcp, and ftp utilities are not available for use on the virtual machines. The secure shell (ssh) and secure copy (scp) tools can be used for connecting to the VMs and for transferring files.

The VM assessment also revealed the strengths and limitations of virtual machines. The strengths included being able to share I/O, use less hardware and increase availability (uptime) for students. The limitations included not being able to connect to outside networks (only within their private network); therefore, it is not possible to download rpm's and other binaries directly to the VMs.

The pros and cons of possible access methods were compared in order to help choose the most appropriate and manageable method to meet the requirements. It was determined that the best method for giving students access to the virtual machines is by creating individual accounts for each student and giving root access to those who require it via sudo, which allows restriction

of usage. Using the customized accounts approach was desirable because it is a more secure method of tracking user activity, and the username can be customized to match the student Regis ID. Also, this is the method that has been successfully used for account creation on Acadunix.

During the design phase (phase 3) of the project, a remote access process and diagram was created and documented. An account creation process for setting up new accounts on the virtual machines was also created, and is incorporated into the administrator lab manual. The account creation script design and pseudo code were developed during this phase and implemented in phase 4 of the project. The test plan to check the functionality of this process and script was developed as well. Although not specified as a deliverable, an account removal script was developed during this phase as well. The last deliverable for this phase was also completed; two lab manuals were created. The Administrator Lab Manual and the Student Lab Manual can be found in Appendix F.

At the implementation phase (phase 4) of the project, the pseudo code for the account creation and account removal scripts was converted into working Perl code scripts. The scripts reside on the master virtual machine. All documentation created in the design phase was revised and sections were improved throughout the duration of the project.

During the testing phase (phase 5), access to the virtual machines from Acadunix was verified. As stated at the beginning of the project, access to the virtual machines can only be done via SSH from a PC or UNIX system. Testing for this phase was divided into two parts: 1) testing for the account setup script, and 2) testing of the process to create accounts.

Account Setup Script Test Results

The acct_setup.pl script resides in the administrator directory, /home/regisadm/Documents, on the master virtual machine 10.10.102.157. An old sample roster file with bogus Regis IDs was used for this test. The test plan used to test the functionality of the script is located in Appendix D. The acct_setup.pl script was first tested on the master virtual machine to only create one account at a time. This test was successful. Next, each item in the test plan was tested for success or failure. Any failures were corrected and the script was re-tested until all items tested generated successful results. Below are the final test results:

A checkmark (✓) means the test was successful.

An x (✗) means the test failed.

Test Items

- ✓ Read in the 'roster' file from within the /home/regisadm/Documents directory.
- ✓ Remove extra characters from the input roster file to prepare for processing.
- ✓ Extract the student RegisID, name, and course information from the processes roster file.
- ✓ Create a "students" group in /etc/group. Set as the default group for new accounts.
- ✓ Create the student username, from the first 5 letters of the student last name and the last 3 digits of the student RegisID.
- ✓ Define variables for all fields of the password entry: UID, GID, GECOS, HOMEDIR, and SHELL.
- ✓ Create a backup copy of the existing password file.
- ✓ Create a new account for each student in the roster list.
- ✓ Create a new home directory for each student with correct permissions (755) and ownership (username:student).
- ✓ Set a default password for each account.
- ✓ Delete any temporary files or settings used during execution.
- ✓ Check if username already exists.

- ✓ Setup each user environment (.profile, permissions, umask).
- ✓ Create multiple accounts at a time.

Although all items tested were successful, there were two unexpected results. First, some accounts were created with shorter UIDs. The student Regis ID consists of seven digits. If the last four digits of the student Regis ID started with zeros, the script did not include the zeros when creating the user ID (UID). For example, for a Regis ID with last four digits of 0041, the script created a UID of 41. Another example, for Regis ID with last four digits of 0662, the UID created was 662. Second, some usernames have a period in it. Since the script takes the first five letters of the student last name to create the username, any last names with less than five letters have a period at the end of the last name in the roster file. The script includes the period in the username. For example, if the roster file contains last names Hutt and Khan, the usernames created are hutt.1234 and khan.1234 respectively. There needs to be a line in the script that checks if the last name is less than five characters, then only use the characters before the period. The period does not affect the functionality of the account, but it is not consistent with the naming convention.

Account Setup Process Test Results

Preliminary work was required before the testing of the account setup process could be started. The preliminary work for the administration of accounts across all seven virtual machines included designating one virtual machine to be the master server where the original account setup and removal scripts can reside, as well as the output of the scripts. The VM instance designated as the master is 10.10.102.157. In addition, an administrator account, regisadm, and home directory, /home/regisadm, were created to be used by the Regis University

administrator to create accounts on each virtual machine. The administrator can use the /home/registadm directory to process account creation files and execute scripts, and is only accessible to the registadm and root users. The registadm administrator account is setup with full sudo access. This was all setup on all seven virtual machines.

In order to automate administration tasks related to account creation/removal on all seven virtual machines, SSH encryption public/private key pairs were generated on the master VM, to allow for connections from the master VM to the rest of the VMs without being prompted for a password. The master VM public key was copied to each one of the 'client' virtual machines. To ensure the SSH Encryption Key pair setup worked correctly, tests were done by copying files from the master VM 10.10.102.157 to each of the client VMs 10.10.102.xxx.

After all preliminary work was complete, all was tested for functionality by making sure each item on the test plan was successful. Below are the final test results:

A checkmark (✓) means the test was successful.
An x (✗) means the test failed.

Preliminary Work

- ✓ Designate one virtual machine to be the server where the original account setup and removal scripts reside as a form of revision and access control.
- ✓ Create an administrator account and group to be used when creating accounts on each virtual machine.
- ✓ Set the password for the new administrator account.
- ✓ Setup the new administrator account with sudo access.
- ✓ Create directory to work in and process account creation, only accessible to the administrator and root user on all VM instances. Set permissions to 700..
- ✓ Setup ssh encryption private/public keys on all hosts for the new administrator account in order to be able to login and copy files between all VMs without being prompted for the password.

- ✓ Test to make sure the SSH Encryption Key pair setup works by copying files from 10.10.102.157 to 10.10.102.xxx.

Functionality

- ✓ Login to all seven VMs with the new administrator account to test that it is working properly.
- ✓ Copy the acct_setup.pl script, acct_delete.pl script, and the roster (list of students) to all 'client' VMs. Then execute the acct_setup.pl script remotely on each VM instance from the master VM. Student accounts should be created on each VM locally.
- ✓ Login with new student accounts and default password to all VM instances to ensure the new accounts work.

For the installation phase (phase 6), the operational versions of the account setup and remove scripts were saved on the master virtual machine (10.10.102.157) in the /home/regisadm/Documents directory. Final versions of all documentation were published as appendices in chapter 6 of this paper. For the final phase, maintenance, time was spent evaluating the solution to validate if the project met the initial project expectations and verify if anything would have been done differently in the project. During this phase, the final two chapters of the project paper were written, which included putting all of the back matter in order, and reviewing the lessons learned and suggestions for future work on this project.

Summary

In summary, the questionnaire results suggest that there is definite interest in having more that one system on which students can work on and gain a more real-life experience of a production environment. Although most instructors agreed that using other remote systems is not crucial, they also expressed that being able to demonstrate remote administration and

network security would be helpful for their classes. During the evaluation of the virtual machines, much information was obtained about them. For instance, some of the remote administration utilities such as rsh, rcp, and ftp are not available, while the tools such as ssh and scp are available. Although this might be a drawback for a lab environment, secure remote administration is still possible and most of the tools required by instructors for their classes are available. Moreover, although this set of virtual machines have no outside connectivity to other networks, the ability to share resources, use less hardware, and have their own private network are still a plus when it comes to availability of systems for Regis students. As far as choosing an approach for providing access to the virtual machines, creating customized accounts was considered the best method for this in order to keep track of users and to make it easier to identify individual users with a unique user ID and username. The design and implementation of the solution took the most time to complete, with the challenge of designing methods and scripts, and then converting the designs into concrete solutions. The desired deliverables and goals were accomplished for those phases. The testing phase was a very important, since any failed tests meant returning to the implementation phase to fix any errors in the scripts or making improvements to the processes and documentation. All testing resulted in success, which permitted completion of this project within the revised project plan timeframe.

Chapter Five: Lessons Learned

The Project Experience

This project has given me the opportunity to view the use and purpose of virtual machines in a different way. Having used other virtual machine products and systems, I had an idea of their function, but in this project I learned more about how virtual machines are setup in the IBM S/390, how they work, the disadvantages and advantages of using virtual machines, and how they can be used in an academic environment. Specifically I learned more about port mapping and how a private network is setup. I also learned more about SuSe Linux; I had to do extra reading and research to learn the differences between SuSe and Red Hat Linux. Developing and writing Perl scripts helped me expand my programming skills and helped increase my understanding of the Perl scripting language. Furthermore, I was able to use analysis and design concepts. For instance, analysis of the current situation was done, the need/problem was identified, and a solution was designed. Developing test plans and using them were learning tasks as well. In addition, as part of the SEAD Practicum, I had the opportunity to serve as the Administrative Lead for the practicum. This provided experience of project management, coordination, documentation and training. I also learned more about using Citrix, how to manage accounts by using Active Directory and how to manage applications from the Active Directory Command Console. Being part of the practicum has given me a chance to be in a lead position, which is very beneficial for the development of my career.

The project met initial expectations and the results were more than satisfactory. A working script was not expected to be completed and an account removal script was not one of the original deliverables. The account setup script was completed, met all of the test objectives and is functional. During the course of the project, a way to clean up testing of account creation

was needed, so an account deletion script was created. I decided to include this script as part of the project deliverables. As expected, the completion of each phase of the project did not necessarily go in chronological order. I found myself working on the design, implementation and test phases all at the same time. For example, after the initial implementation of the account setup script, I performed testing, then revised the script and did more testing. Writing the paper was also a continual work on different chapters during different phases of the project. Ideas and journal entries were inserted into pertinent parts of the paper.

Next Stage of Evolution

The next stage of this project would continue with the maintenance phase, in which improvements could be made. This phase would include:

- Revision and improvement of the account setup and remove scripts.
- Further development of an account management plan.
- Further development of security controls.
- Installing, testing, and running security tools.
- Further testing and development of student lab manual.
- Take action on extra comments made by facilitators regarding the need for multiple UNIX systems (Appendix A).
- Further development of this solution to provide a virtual lab environment for students.

Conclusion

Overall, the project met the goal of providing more than one UNIX system for students to work on. Although the time allotted on the IBM S/390 is uncertain, this project provides an

initial start for granting access and managing users on these virtual machines. This project can be developed further to have more of an impact on the availability of resources for Regis University students. Specifically, the scripts produced in this project can be greatly improved, more tools can be developed, several student labs and exercises can be developed, and everything mentioned in the *Next Stage of Evolution* could be done. I strongly believe that Regis could benefit from having these virtual machines and expand the learning experience for future students.

Chapter Six: Back Matter

References

- Courtney, S. (2001, Aug 29). Colorado State University, IBM in \$21.6M National Technology Hub Partnership. *Linux Today*. Retrieved July 25, 2005, from http://linuxtoday.com/news_story.php3?ltsn=2001-08-29-001-20-PR-DP-HE
- Fast Facts. (n.d.). Retrieved October 1, 2005, from <http://www-03.ibm.com/servers/eserver/zseries/os/linux/facts.html>
- Glossary of Terms. (n.d.). Keiro Technologies. Retrieved August 28, 2005, from <http://www.kerio.com/manual/wrp/en/>
- Kay, R. (2002, May 14). Quick Study: System Development Life Cycle. *Computer World*. Retrieved October 25, 2005, from <http://www.computerworld.com/developmenttopics/development/story/0,10801,71151,00.html>
- Kernel definition. (n.d.). Retrieved on October 1, 2005, from <http://linux.about.com/cs/linux101/g/kernel.htm>
- Kohlbrenner, E., Morris, D., & Morris, B. (1999). *The Core of Information Technology Introduction to Virtual Machines*. Retrieved July 25, 2005, from <http://www.cne.gmu.edu/itcore/virtualmachine/>
- Kreuter, D. (2000). *The Evolution of Linux on the S/390 Platform. Technical Support, August*. Retrieved on August 29, 2005, from <http://www.naspa.com/PDF/2000/0800%20TS%20PDFs/T0008004.pdf>
- Linux (n.d.). Retrieved October 1, 2005, from http://linux.about.com/od/glossary/l/bldef_linux.htm
- Linux definition. (n.d.). Retrieved October 1, 2005, from http://whatis.techtarget.com/definition/0,289893,sid9_gci212482,00.html
- Linux Users and Sudo. (n.d.). Retrieved December 4, 2005, from http://www.siliconvalleyccie.com/linux-hn/addusers.htm#_Toc92808556
- Perl Crypt (n.d.). Retrieved January 28, 2006, from http://www.perlscriptsjavascripts.com/tutorials/howto/encrypt_unix_passwords.html
- S/390 definition. (n.d.). Retrieved on October 1, 2005, from http://search390.techtarget.com/sDefinition/0,,sid10_gci214587,00.html

Shah, S. (2001). *Linux Administration A Beginner's Guide (2nd)*. Berkely, CA: McGraw-Hill Companies.

Stockman, M. (2003). Creating Remotely Accessible “Virtual Networks” on a Single PC to Teach Computer Networking and Operating Systems. Association for Computing Machinery, Special Interest Group for Information Technology Education (ACM/SIGITE) Conference for Information Technology Curriculum IV Proceedings and the ACM Digital Library.

Turk, D., Bausch, J. (2005). Virtual Linux servers under z/VM: Security, performance, and administration issues. *IBM Systems Journal*, 44, 341-351.

Virtual Machine. (n.d.). Retrieved on August 28, 2005, from
http://www.webopedia.com/TERM/v/virtual_machine.html

Virtual machine definition. (n.d.). Retrieved on July 25, 2005, from
http://whatis.techtarget.com/definition/0,,sid9_gci213305,00.html

Definition of terms

Term	Definition
GUI	Graphical User Interface. The interface between the computer application and the user, where the user clicks on a visual screen that has icons, windows and menus, by using a pointing device, such as a mouse. GUI is pronounced like "gooey".
Port Mapping	Port mapping is the process where packets arriving to a particular IP address/port can be translated and redirected to a different IP/port. This functionality is a way to create a persistent passage through Network Address Translation (NAT).
NAT	Network Address Translation. Also known as IP masquerade, it is a process of translating the source header of IP packets so they will be routable across wide area networks.
Virtual machine (VM)	A self-contained operating environment, which allows for sharing of power and resources from a mainframe and operates as an independent system.
Kernel (Linux)	“vmlinuz” is the core of the Red Hat Linux operating system. The kernel handles the basic functions of the operating system such as memory allocation, process allocation, and device input and output.
NLP	Networking Lab Practicum
SEAD	Systems Engineering and Application Development Practicum. Formerly known as NLP.
MSCIT	Master of Science in Computer Information Technology

Appendix A Questionnaire & Results Comments

Virtual Lab Requirements Questionnaire

Project: Unix remote access via IBM S/390 Virtual Machines

Selene Peña

November 3, 2005

The purpose of this questionnaire is to obtain information about the requirements of a virtual lab for the UNIX and Information Assurance classes at Regis University. This questionnaire will be used as a starting point and is part of my thesis/professional project for the Regis MSCIT Masters degree program.

The Regis Systems Engineering and Application Development (SEAD) Practicum was granted the use of an IBM S/390 server, which hosts seven functional SuSE Linux virtual machines. The goal of my project is to make these multiple Linux systems available for students to use during their lab time, to work on assignments and projects, or simply for testing and practice. Your responses will help determine the needs of a lab environment. This in turn, will be used to develop and design a solution for providing a substantial lab environment for students in your classes. Student access and account management solutions will also be developed as part of this project. Please take a few minutes to help, by filling out this questionnaire and returning it to selenepena@comcast.net by Sunday, November 13, 2005.

1. Which class do you teach? _____

2. For each statement, please indicate your answer by marking with an "x".

There is a need for multiple UNIX/Linux servers for UNIX and Information Assurance classes.

<i>Strongly Disagree</i>	<i>Disagree</i>	<i>Neutral</i>	<i>Agree</i>	<i>Strongly Agree</i>
------------------------------	-----------------	----------------	--------------	---------------------------

(1)___	(2)___	(3)___	(4)___	(5)___
--------	--------	--------	--------	--------

Students would benefit from using more than one UNIX/Linux server to do their work.

<i>Strongly Disagree</i>	<i>Disagree</i>	<i>Neutral</i>	<i>Agree</i>	<i>Strongly Agree</i>
------------------------------	-----------------	----------------	--------------	---------------------------

(1)___	(2)___	(3)___	(4)___	(5)___
--------	--------	--------	--------	--------

3. Is access to remote systems essential for students to learn the course objectives?

☐ Yes
☐ No

If yes, please explain why or give examples.

4. If the virtual machines running were accessible, would you consider using them for your class?

☐ Yes
☐ No

If yes, how would you use them or incorporate them into your class?

5. Bellow is a list of tools and applications installed on the virtual machines. Please mark the tools and applications you currently use, and list any additional required tools in the space provided below.

Installed Tools	
<input type="checkbox"/> apache	<input type="checkbox"/> mysql-shared
<input type="checkbox"/> bind	<input type="checkbox"/> net-tools
<input type="checkbox"/> cracklib	<input type="checkbox"/> openssl
<input type="checkbox"/> gawk	<input type="checkbox"/> pam
<input type="checkbox"/> gcc	<input type="checkbox"/> perl
<input type="checkbox"/> glib	<input type="checkbox"/> sed
<input type="checkbox"/> glibc	<input type="checkbox"/> sendmail
<input type="checkbox"/> gzip	<input type="checkbox"/> sudo
<input type="checkbox"/> iptables	<input type="checkbox"/> tar
<input type="checkbox"/> mailx	<input type="checkbox"/> Imap
<input type="checkbox"/> make	<input type="checkbox"/> OpenLDAP

Additional tools:

6. Please list any specific functionality required on the virtual machines for class instruction or student learning.

7. Will your students require root or super user capabilities?

☐ Yes

☐ No

If no, what security level is needed on the virtual machines for students?

☐ Low (i.e. read access to other users' directories and files)

☐ Medium (i.e. read access to only user directories and files in the same class)

☐ High (i.e. restricted read access permissions to all non-system directories and files)

☐ Other _____

8. Are there any specific disk space requirements?

☐ Yes

☐ No

If yes, please explain what the requirements are and why they are needed.

9. May I contact you regarding your responses if more information is needed?

☐ Yes

☐ No

Thank you for your participation!!

(Please include any comments in this blank area)

Questionnaire Results Additional Comments

- Ensure there is adequate maintenance coverage at the physical location of the devices. Getting kicked out of the device and modifying the configuration in a way that the device is rendered inoperable can happen regularly. On-line students will be logging in all hours of the day because of geographical extremes so I'd suggest 24/7 if that's within the "budget" boundary. Someone needs to be on site to reload an old configuration, reset passwords or reboot systems.
- The lab exercises need to be constantly kept up to date. Configuration control needs to constantly be monitored and the lab exercises tested/quality controlled regularly.

- It would be a good idea to monitor activities performed on the machines and what keep an eye on what tools are being used and where. Doesn't have to be an IDS, but it would be a good idea for someone to review logs and get an idea of what's happening on the machines internally and externally.
- You might want to set up a web page so that students and instructors can see the status of lab tools, devices and just provide a general idea of what tools/devices are active and accessible; it might even list software versions and provide an "infrastructure map" of the extent of the resources.

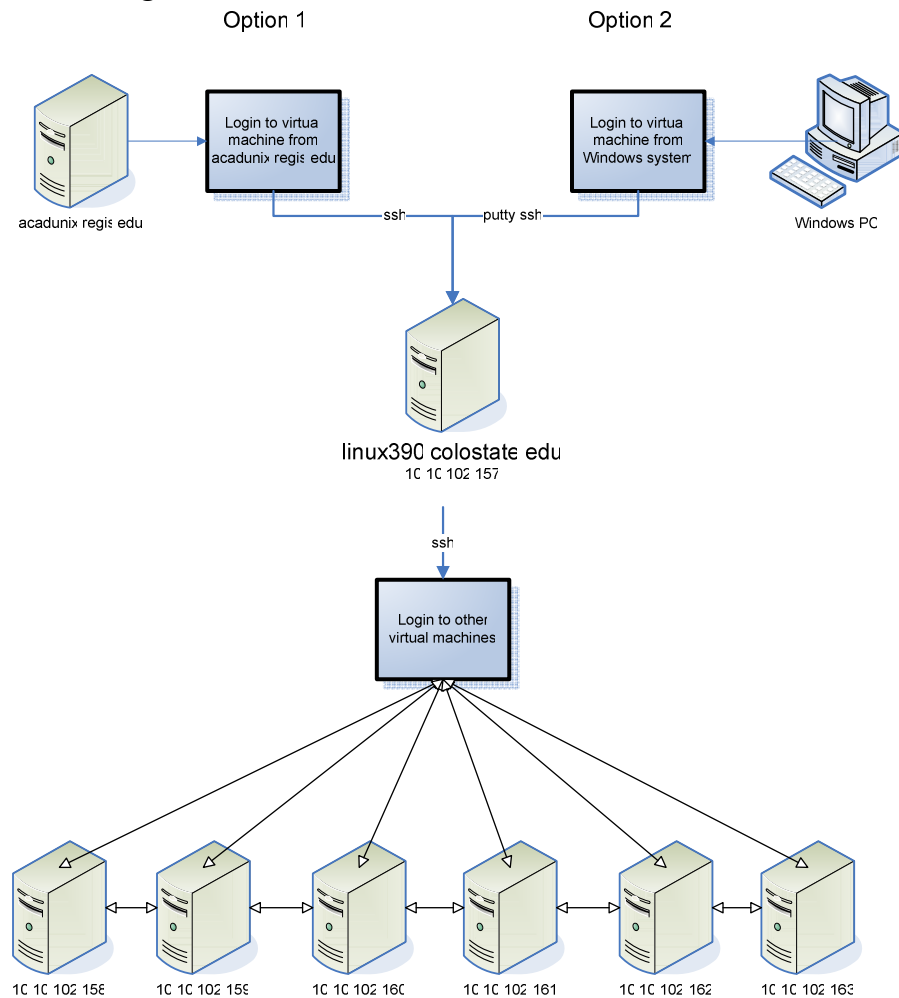
Appendix B Remote Access and Account Setup Processes

As part of the solution, a remote access process was developed to provide students/users instructions on how to login to the virtual machines on the IBM S/390 system via different methods. Before any students can login to the virtual machines, they must be given access by the administrator. The account setup process was developed to take the administrator through the steps that need to be completed to setup user accounts for a typical set of Regis University of students.

Remote Access Process

Connecting to the Linux Virtual Machines IBM S/390

High Level Process Diagram



Detailed Process Steps

These are the seven virtual machines available on the IBM S/390.

10.10.102.157
10.10.102.158
10.10.102.159
10.10.102.160
10.10.102.161
10.10.102.162
10.10.102.163

To login, start with option 1 or option 2 below.

Option 1:

1. Login to acadunix.regis.edu. From acadunix, login to any one of the virtual machines via ssh with the assigned user account and password. The port must be specified with the `-p` option to ssh.

```
ssh -p 5157 user@linux390.colostate.edu
```

Option 2:

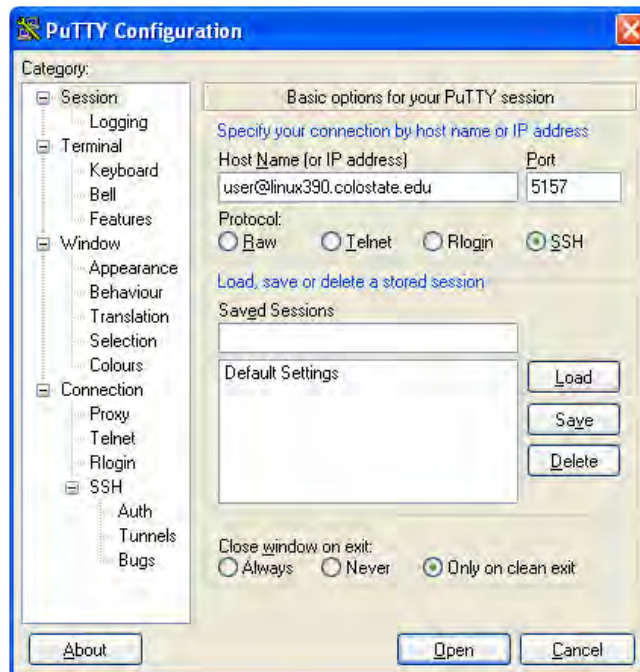
1. Use putty to login to the virtual machines directly from a Windows system with assigned user account and password.

From the command line, enter the following command:

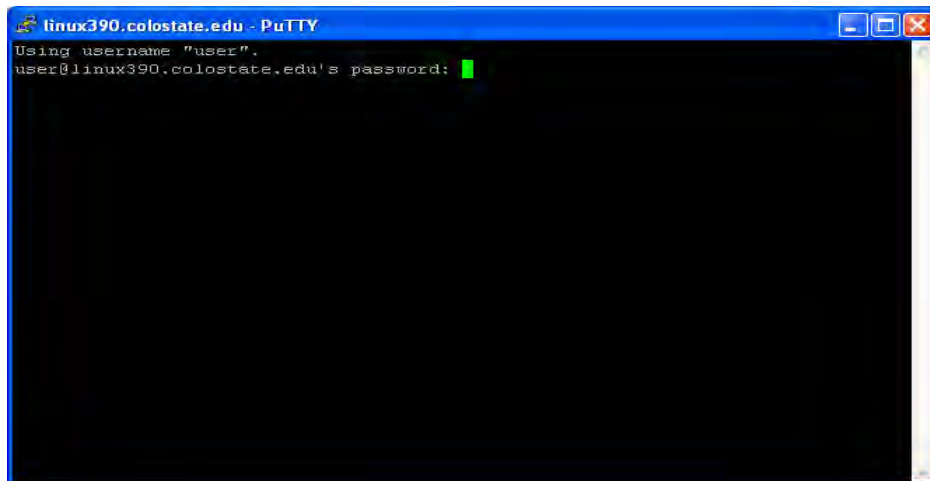
```
putty.exe -ssh -P 5157 user@linux390.colostate.edu
```

From the putty dialog box, enter the host name, including the username you will login with, and enter the port number in the designated field.

```
user@linux390.colostate.edu
```



When prompted, enter your password.



2. Once logged into one virtual machine, the user can login to all the other virtual machines from there.

The virtual machines can only communicate with each other via their private network; they cannot communicate with other external systems. To login to virtual machine 10.10.102.159, for example, enter the following command:

```
ssh user@10.10.102.159
```

3. User accounts and home directories were setup by the administrator with a default password. The user must change his/her password on each of the virtual machines by logging into each VM and executing the “passwd” command.

For example:

`passwd`

Changing password for user.

Old Password:

New password:

Re-enter new password:

Password changed

A secure password can be selected by following these suggestions:

- must be 6 to 8 characters long
- must have a combination of numbers, lower-case letters, capital letters, and symbols (i.e. !, \$, *, etc.)

Account Setup Process

1. Obtain the student roster from Regis faculty, which lists students that need accounts to be setup. If the file in Word (.doc) format, copy and paste the contents into a text (.txt) file. Save the text file as "roster".

2. Transfer the "roster.txt" file to the 10.10.102.157 virtual machine (VM). This VM is designated as the master VM, where all account management tasks are centralized. "regisadm" is the Regis University administrator account, which has root privileges via the sudo utility.

You may use a secure copy or secure ftp client to transfer the roster.txt file to the master VM. If you do not have one of these clients, follow these instructions to download the PSFTP (secure ftp) client to your PC.

a. Go to the PuTTY download page via this URL:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

b. Scroll down to the Binaries section, under "*The latest release version*" section. Right click on "psftp.exe," and chose "Save Target As".

Save "psftp.exe" to your Windows desktop. Now you should have a PSFTP icon on your desktop.

c. PSFTP is ready to be used.

Note: If you do not have an SSH client on your PC, follow the same steps above to download "putty.exe".

After downloading the secure ftp client, continue here to transfer the roster.txt file.

- Open a command prompt on your PC.
- Change directories into your desktop (where you saved psftp.exe)
- Enter the following commands:

```
psftp -P 5157 regisadm@linux390.colostate.edu
Using username "regisadm".
regisadm@linux390.colostate.edu's password: <enter the password>
Remote working directory is /home/regisadm
psftp> cd Documents
Remote directory is now /home/regisadm/Documents
psftp> put roster.txt
local:roster.txt => remote:/home/regisadm/Documents/roster.txt
psftp> quit
```

The roster.txt file has been transferred to the /home/regisadm/Documents directory on the master VM 10.10.102.157.

3. Login to 10.10.102.157 with the regisadm administrator account. (Please see the "Remote Access Process" published in chapter 6 of this paper for more details).

From your PC command prompt enter:

```
putty.exe -ssh -P 5157 regisadm@linux390.colostate.edu
```

Enter the password. You will be in the /home/regisadm directory as soon as you login. Go to the Documents directory.

```
cd Documents
```

4. Now you are ready to create student accounts. From the /home/regisadm/Documents directory, execute the acct_setup.pl script and save the output to another file such as results157, in which 157 represents the 10.10.102.157 virtual machine.

```
./acct_setup.pl > results157
```

The acct_setup.pl script will read the roster.txt file that you copied in the previous steps.

The results157 file will contain a list of the accounts that were created. Here is an example:

schwa035, Schwarzman. Stuart A, 03F8W2_CS_465_C70

burro513, Burroughs. Michael J, 03F8W2_CS_465_C70

avena061, Avena. Ramon D, 03F8W2_CS_465_C70

COMPLETE

Script executed on Sun Jan 29 19:46:00 MST 2006

5. Now, create accounts on the rest of the virtual machines.

Virtual Machines:

10.10.102.158

10.10.102.159

10.10.102.160

10.10.102.161

10.10.102.162

10.10.102.163

Use secure copy to copy the roster file and acct_setup.pl script to each one of the virtual machines. For example:

```
scp acct_setup.pl roster.txt 10.10.102.158:/home/regisadm/Documents
acct_setup.pl                  100% 3230   0.0KB/s  00:00
roster.txt                     100% 250    0.0KB/s  00:00
```

```
scp acct_setup.pl roster.txt 10.10.102.159:/home/regisadm/Documents
acct_setup.pl                  100% 3230   0.0KB/s  00:00
roster.txt                     100% 250    0.0KB/s  00:00
```

etc...

Execute the account setup script on each virtual machine. Save the output of the script to a "results" file for each virtual machine in order to keep track of where the accounts were created. For example, from 10.10.102.157 enter:

```
ssh 10.10.102.158 sudo /home/regisadm/Documents/acct_setup.pl > results158
```

```
ssh 10.10.102.159 sudo /home/regisadm/Documents/acct_setup.pl > results159
```

etc...

The "results" files will be saved locally on the master VM. You should have seven results files.

```
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:44 results157
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:44 results158
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:45 results159
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:45 results160
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:45 results161
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:46 results162
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:46 results163
```

If the user account already exists, and if the group already exists, the script output will show:

groupadd: group student exists

useradd: user schwa035 exists
schwa035, Schwarzman. Stuart A, 03F8W2_CS_465_C70

useradd: user burro513 exists
burro513, Burroughs. Michael J, 03F8W2_CS_465_C70

useradd: user avena061 exists
avena061, Avena. Ramon D, 03F8W2_CS_465_C70

COMPLETE

Script executed on Sat Jan 28 14:06:28 MST 2006

Test a few of the accounts to make sure you can login with the new student accounts on each virtual machine . Use the default password. For example,

`putty.exe -ssh -P 5158 user123@linux390.colostate.edu`

OR

`ssh user123@10.10.102.158` (from another virtual machine)

6. Delete the files you copied to each of the virtual machines. This will ensure that you are always using a recent version of the setup script and the most current roster.

```
ssh 10.10.102.158 rm /home/regisadm/Documents/acct_setup.pl  
ssh 10.10.102.158 rm /home/regisadm/Documents/roster.txt
```

If you prefer, you can login to each virtual machine to remove files locally.

```
ssh 10.10.102.158
```

7. Email the "results" file to the requestor of the accounts. The results file will contain the username, student name and the course information.

Appendix C Account Setup and Removal Scripts

These set of scripts were developed for providing access to the virtual machines on the IBM S/390 system. The account setup script can be used to create new accounts and home directories for students. This script reads the “roster” file, which provides a list with student information. The account removal script can be used to remove student accounts and home directories. This script reads the "delete_roster" file, which contains the usernames of students whose accounts should be removed. For instructions on how to use these scripts, refer to Appendix B Remote Access and Account Setup Processes.

Account Setup Script

```
#!/usr/bin/perl
#
# Regis University
# SEAD 2005B, Professional Project, 2005-2006
# Author: Selene Pena
# Script name: acct_setup.pl
# Script start: Dec. 5, 2005
# Script complete: Jan. 29, 2006
# Usage: /home/regisadm/Documents/acct_setup.pl > results
# Purpose: This script is to be used to create new accounts and home directories for students.
```

```
#####
#Define variables
#####
```

```
my $infile = `cat /home/regisadm/Documents/roster.txt`;
my $regisid;
my $studentname;
my $course;
my $newfile = "newfile";
my $last;
my $first;
my $pwfile = "/etc/passwd";
```

```
#-----
#This script reads in the roster file, which contains the list of students.
#The current format looks like this:
#
#    "1234567","Michaelis. Jeffrey C","03F8W2_MSC_612_M72",
#
```

```

#
#The format of the roster file must be fixed.
#-----
#
#Strip out the extra quotation marks and commas from the student roster file.
$infile =~ s/"//g;
$infile =~ s/,,$//gm;
$infile =~ s/,/:/g;
#print "$infile\n";

#Save the stripped out roster file to a temporary new file.
open(NEWFILE, ">$newfile");
print NEWFILE "$infile";
close NEWFILE;

#Save the new file to an array.
my @newinfile = `cat newfile`;
#print "@newinfile\n";

#Create a new student group in /etc/group, with group ID of 1000.
`/usr/sbin/groupadd -g 1000 student`;

#Now extract the RegisID, Student Name, and Course information from the new file.
foreach $line (@newinfile){
($regisid,$studentname,$course) = split(":", $line);

#Break down the student name into first and last names.
#The current format for $studentname is: "Lastname. Firstname MiddleInitial"
($last,$first) = (split " ", $studentname);
#print "$last\n";

#Create the user login from the first 5 letters of the student last name and
#the last 3 digits of the student RegisID. The RegisID is 7 digits long.
my $fivelast = substr($last, 0, 5);
my $three = substr($regisid, 4, 7);
#print "$fivelast\n";
#print "$three\n";

#Create the username and make everything lower case.
my $username = lc("$fivelast$three");

#-----
#Now that the standard usernames are created,
#go ahead and create the new user accounts.
#-----

```

```

#The UID is the last four digits of the student RegisID.
my $uid = substr($regisid, 3, 7);

#The GID is 1000 "student" group in /etc/group. All student accounts will default to this group.
my $gid = "1000";

#The GECOS field is the student name and a description.
my $gecos = "$first $last Regis University";

#This is the default HOME directory.
my $homedir = "/home/$username";

#The SHELL is /bin/bash
my $default_shell = "/bin/bash";

#The PASSWORD is a temporary default password. Each student must change this password for
their own account.
my $pass = "grKQn3VBMkqic";

#-----
#Use the previously defined variables to form the password entry.
#Password entry requires the following fields in this format:
#
#   account:password:UID:GID:GECOS:directory:shell
#
#-----
#
#Create the accounts for each student in the roster.

#First make a backup copy of the password file.
`cp $pwfile /etc/passwd.bak`;

#Add the user accounts.
`/usr/sbin/useradd -p $pass -d $homedir -m -g $gid -u $uid -s $default_shell $username`;
system("/usr/bin/chfn -f \"$gecos\" $username > /dev/null 2>&1");

# Print a message to confirm completion and list the accounts that were created.
print "$username, $studentname, $course\n";

}

my $date = `date`;
print "COMPLETE\n";
print "Script executed on $date\n";

```

```
#Delete any temporary files.
unlink $newfile;
```

```
#####
#End of acct_setup.pl script.
#####
```

Account Removal Script

```
#!/usr/bin/perl
#
# Regis University
# SEAD 2005B, Professional Project, 2005-2006
# Author: Selene Pena
# Script name: acct_remove.pl
# Script start: Jan. 16, 2006
# Script complete: Jan. 16, 2006
# Usage: /home/regisadm/Documents/acct_remove.pl > results
# Purpose: This script is to be used to remove student accounts and home directories.
#         The script reads the "delete_roster" file, which contains the usernames of
#         students whose accounts should be removed.
```

```
#####
#Define variables      #
#####
```

```
my $username;
my @infile = `cat /home/regisadm/Documents/delete_roster.txt`;
my $pwfile = "/etc/passwd";
```

```
#-----
#First make a backup copy of the password file. Then delete account and home directory.
#-----
```

```
`cp $pwfile /etc/passwd.bak`;
```

```
foreach $username (@infile) {
```

```
print "Deleting account $username...";
```

```
`/usr/sbin/userdel -r $username`;  
print "$username removed\n";  
  
}
```

```
#####  
#End of script      #  
#####
```


Appendix D Account Setup Test Plans

In order to test the functionality of the account setup process and the script, test plans were designed and used during the testing phase of this project. The account creation process was tested with the Account Setup Process Test Plan to ensure it provided correct instructions of how to prepare the environment for new accounts, and how to use the account creation script. The account creation process included several preliminary steps in order to be able to execute the account creation script and successfully create accounts on all seven virtual machines. The Account Setup Script Test Plan was used to test the functionality of the script. Each item in the test plan was tested for success or failure. Any failures were corrected and the script was re-tested until all items tested generated successful results.

Account Setup Process Test Plan

Preliminary Work:

- ❑ Designate one virtual machine to be the server where the original account setup and removal scripts reside as a form of revision and access control.
- ❑ Create an administrator account and group to be used when creating accounts on each virtual machine.
- ❑ Set the password for the new administrator account.
- ❑ Setup the new account with sudo access.
- ❑ Create directory to work in and process account creation, only accessible to the administrator and root user on all VM instances. Set permissions to 700..
- ❑ Setup ssh encryption private/public keys on all hosts for the new administrator account in order to be able to login and copy files between all VMs without being prompted for the password.
- ❑ Test to make sure the SSH Encryption Key pair setup works by copying files from 10.10.102.157 to 10.10.102.xxx.

Functionality:

- ❑ Login to all seven VMs with the new administrator account to test that it is working properly.
- ❑ Copy the acct_setup.pl script, acct_delete.pl script, and the roster (list of students) to all 'client' VMs. Then execute the acct_setup.pl script remotely on each VM instance from the master VM. Student accounts should be created on each VM locally.
- ❑ Login with new student accounts and default password to all VM instances to ensure the accounts work.

Account Setup Script Test Plan

Test Items:

- ❑ Read in the 'roster' file from within the /home/regisadm/Documents directory.
- ❑ Remove extra characters from the input roster file to prepare for processing.
- ❑ Extract the student RegisID, name, and course information from the processes roster file.
- ❑ Create a "students" group in /etc/group. Set as the default group for new accounts.
- ❑ Create the user login/username, from the first 5 letters of the student last name and the last 3 digits of the student RegisID.
- ❑ Define variables for all fields of the password entry: UID, GID, GECOS, HOMEDIR, and SHELL.
- ❑ Create a backup copy of the existing password file.
- ❑ Create a new account for each student in the roster list.
- ❑ Create a new home directory for each student with correct permissions (755) and ownership (username:student).
- ❑ Set a default password for each account.
- ❑ Delete any temporary files or settings used during execution.
- ❑ Check if username already exists.
- ❑ Setup each user environment (.profile, permissions, umask).
- ❑ Create multiple accounts at a time.

Appendix E Project Checklists

Checklists were used to uphold the goals of the project. The two documented checklists are the SDLC Completion Checklist and the Virtual Machine (VM) Evaluation Checklist. The SDLC Completion Checklist was used to keep a high level view of the tasks to be completed at each phase of the project. Items on this checklist were marked with a checkmark when they were completed, or circled to indicate further work was required. The VM Evaluation Checklist was used to determine the configuration and setup of the virtual machines running on the IBM S/390 system. The information obtained from each of the virtual machines was documented in the author's project journal.

SDLC Completion Checklist

Phase1 – Planning

- Establish a high level view of the project and determine the goals of the project.
- Contact Unix and Information Assurance instructors to inform them of my project and ask for participation.
- Prepare a questionnaire to Regis instructors in order to determine the needs of the lab environment.

Phase 2 – Analysis

- Evaluate the virtual machines on the IBM S/390 and investigate the unknowns.
- Understand how VPNs work on the IBM 390 virtual machines.
- Find the strengths and limitations of the virtual machines.
- Understand how network services are configured and controlled.
- Determine best method to give students access to virtual machines and how to manage student accounts.
- Analyze questionnaire responses.
- Create process for giving users root access via sudo.
- Keep a log and document research findings.

Phase 3 – Design

- Design the solution for remote access.
- Create process diagrams and documentation.
- Create pseudo code for any scripts needed.
- Setup communication to virtual machines from acadunix.
- Create sample lab to be used for testing by students.
- Design test plans.

Phase 4 – Implement

- Implement the design of the solution.
- Convert pseudo code into working code.
- Finalize process documentation.

Phase 5 – Test

- Test the solution and fix any errors/problems.
- Have students test solution by working through sample lab.

Phase 6 – Install

- Put solution into production.
- Create the environment.
- Final problem solution and documentation.

Phase 7 – Maintain

- Evaluate new environment and solution.
- Give suggestions on maintenance for future administrators.
- Provide lab manual.

VM Evaluation Checklist

1. Network setup and General information.

- ☐ View network status by checking the following files.
 - /etc/nsswitch.conf
 - /etc/resolv.conf
 - /etc/host.conf
 - /etc/hosts
 - /etc/services
 - /etc/networks
 - /etc/protocols
 - /etc/exports
- ☐ Check the network status with the netstat command.
- ☐ View interface configuration with the ifconfig command.
- ☐ How is data transferred between VMs?
- ☐ Check for the capacity of the file systems (df).
- ☐ Review the information in the /proc file system:
 - /proc/cpuinfo
 - /proc/version
 - /proc/dasd/devices
 - /proc/cpuinfo
 - /proc/meminfo

2. Configuration files setup.

- Check start-up control/system configuration:
 - /etc/inittab
 - /etc/rc.d/*
 - /etc/fstab
 - /etc/syslog.conf
 - /etc/modules.conf

3. Security setup.

- Check the configuration of these files:
 - /etc/passwd
 - /etc/group
 - /etc/shadow
 - /etc/securetty
 - /etc/hosts.allow
 - /etc/hosts.deny
- File permissions (umask)
- Profile setup
 - x/.profile or x/.bashrc
 - /etc/profile or /etc/bashrc
- SSH
- System monitoring (any tools installed?)
 - Security tools (syslog, monitor bandwidth with MRTG, COPS, TripWire, SATAN)

4. SMTP (mail) and external services.

- Check the following files:
 - /etc/inetd.conf
 - /etc/sendmail.cf
 - /etc/httpd/*
 - /etc/exports

5. System Services.

- Check the following files:
 - /etc/inittab
 - inetd processes
 - /etc/inetd.conf
 - /etc/syslog.conf
- Check for running daemons:
 - syslog

- ssh
 - rsh
 - telnet
 - ftp
 - cron
- Other system utilities:
- crontab
 - ssh, scp, mount, vi, tcl/tk, nfs, finger, who, rwho, ksh, sh, awk, rsh, rcp, ftp, and telnet php, cgi, terminal server, security tools
 - chkconfig
 - NFS
 - NIS
 - Samba (smbd, nmbd)

Appendix F Student and Administrator Lab Manuals

These lab manuals were developed to aid in using the virtual machines on the IBM S/390 system, and to provide account management instructions for the administrator. The first manual is to be used by students to login to the virtual machines and perform exercises related to system administration. The second lab manual is to be used by an administrator to setup student accounts and related tasks.

Student Lab Manual

Regis University

Student Lab Manual

Virtual Machines on the IBM S/390

Contents

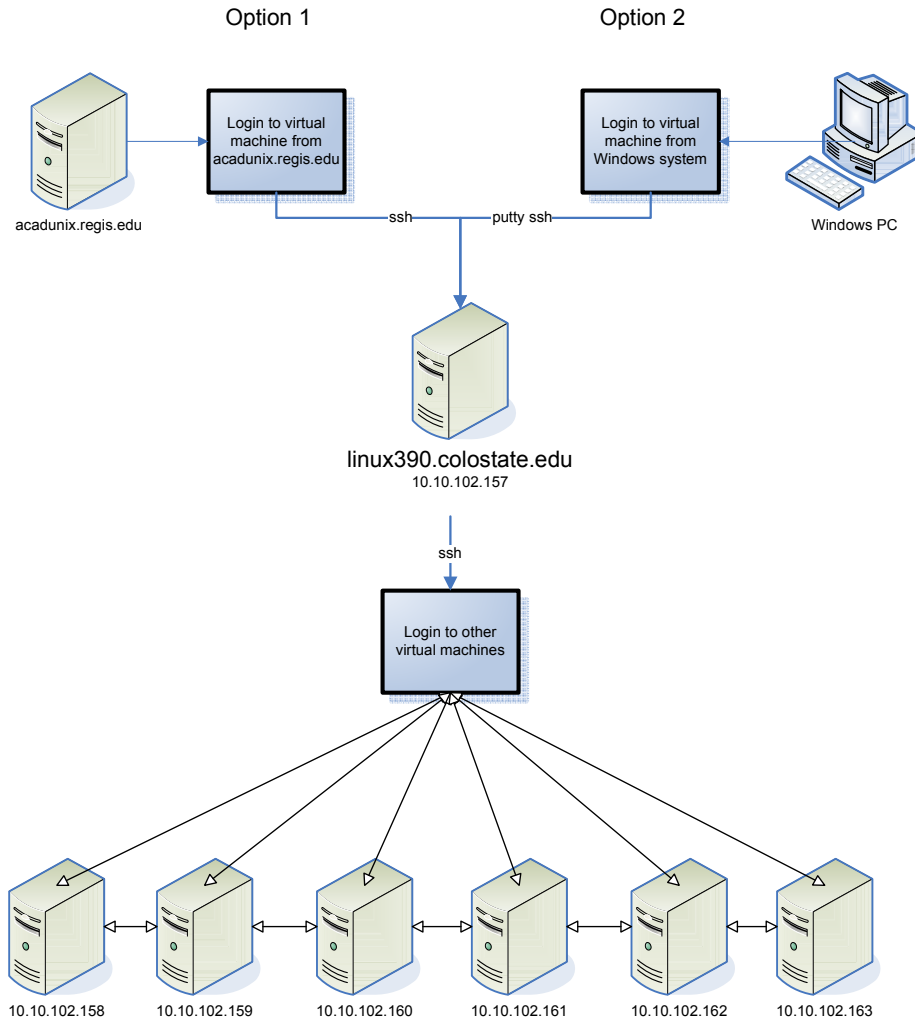
Connecting to the IBM S/390

Exercises

Exercise Answers

Connecting to the IBM S/390

High Level Process Diagram



Detailed Process Steps

These are the seven virtual machines available on the IBM S/390.

10.10.102.157
10.10.102.158
10.10.102.159
10.10.102.160
10.10.102.161
10.10.102.162
10.10.102.163

To login, start with option 1 or option 2 below.

Option 1:

2. Login to acadunix.regis.edu. From acadunix, login to any one of the virtual machines via ssh with the assigned user account and password. The port must be specified with the `-p` option to ssh.

```
ssh -p 5157 user@linux390.colostate.edu
```

Option 2:

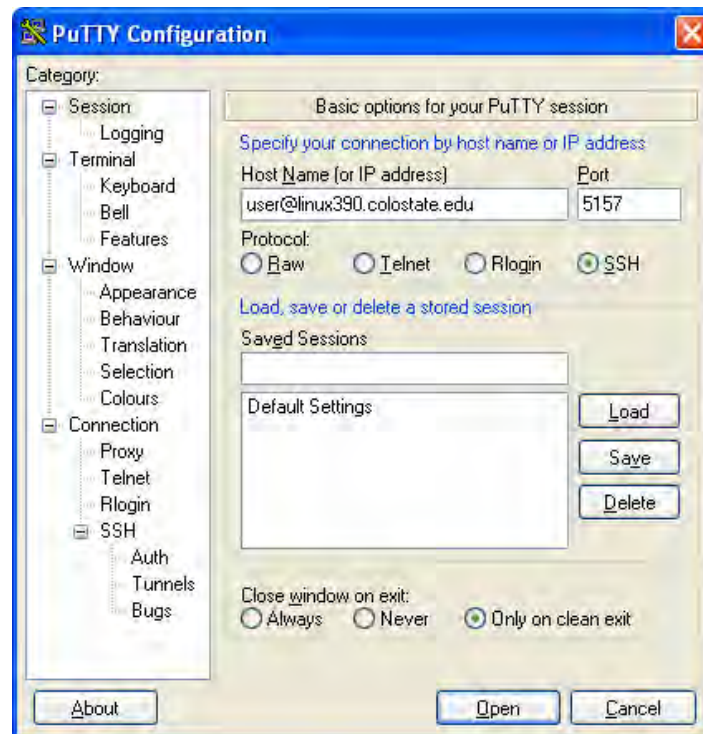
2. Use putty to login to the virtual machines directly from a Windows system with assigned user account and password.

From the command line, enter the following command:

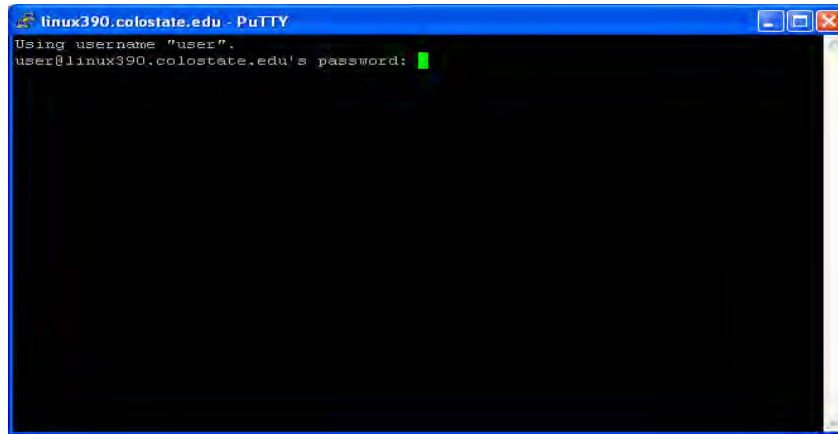
```
putty.exe -ssh -P 5157 user@linux390.colostate.edu
```

From the putty dialog box, enter the host name, including the username you will login with, and enter the port number in the designated field.

```
user@linux390.colostate.edu
```



When prompted, enter your password.



2. Once logged into one virtual machine, the user can login to all the other virtual machines from there.

The virtual machines can only communicate with each other via their private network; they cannot communicate with other external systems. To login to virtual machine 10.10.102.159, for example, enter the following command:

```
ssh user@10.10.102.159
```

3. User accounts and home directories were setup by the administrator with a default password. The user must change his/her password on each of the virtual machines by logging into each VM and executing the “passwd” command.

For example:

```
passwd
```

Changing password for user.

Old Password:

New password:

Re-enter new password:

Password changed

A secure password can be selected by following these suggestions:

- must be 6 to 8 characters long
- must have a combination of numbers, lower-case letters, capital letters, and symbols (i.e. !, \$, *, etc.)

Exercises

The exercises in this manual are example tasks that cover:

- Logging In and Out
- Using the Command Line
- File Ownership
- File Permissions
- Viewing & Managing Processes
- Using Remote Access
- Using the Manual Pages

1. List all files in your home directory.
2. List all processes that are running the virtual machine you are currently logged into.
3. View the processes that are taking up the most CPU.
4. Use the kill command to stop a process.
5. Create a file in your home directory and name it "myfile". Use the vi editor to enter some text into the file, and then save the file.
6. List the files in the directory to see the ownership and permissions of the "myfile" file you created in the previous step. Who is the owner of the file and what are the permissions set to?
7. Copy the "myfile" file from the current system to your home directory on a remote system by using the scp command. Then, do a listing of your home directory on the remote system to see that your file was copied.
8. Copy the "myfile" file from the current system to the /tmp directory on a remote system by using the scp command. Then, do a listing of the /tmp directory on the remote machine to see that your file was copied.
9. Login to a remote system from the local system that you are logged into, with your username. You may login to any of the seven virtual machines. You will notice the prompt will show the hostname of the remote system that you logged into.

For example, if you login to VM 10.10.102.158, the prompt will look like this:

```
penas715@ls102158:~>
```

Type exit at the command prompt to exit the remote system and return to your local system.

10. Use the man page to find out how to use the netstat command.

Exercise Answers

1. Any three of these commands may be used:

```
ls -l ~  
ll ~  
ll $HOME
```

2. To view all of the processes that are running, use the ps command.

```
ps -ef
```

where the e option shows the environment, and the f option formats the output to show parent and child processes.

3. Use the top command to display the top CPU processes. The top command allows you to see processor activity in real time. It displays a listing of the most CPU intensive tasks on the system.

Type "top" on the command line:

```
penas715@ls102157:~> top
```

```
top - 14:47:44 up 26 days, 1:21, 2 users, load average: 0.00, 0.00, 0.00  
Tasks: 28 total, 1 running, 27 sleeping, 0 stopped, 0 zombie  
Cpu(s): 0.0% user, 0.0% system, 0.0% nice, 100.0% idle  
Mem: 61372k total, 58144k used, 3228k free, 10808k buffers  
Swap: 143896k total, 0k used, 143896k free, 13092k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	15	0	244	244	208	S	0.0	0.4	0:00.35	init
2	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration_CPU0
3	root	25	0	0	0	0	S	0.0	0.0	0:00.00	kmcheck
4	root	15	0	0	0	0	S	0.0	0.0	0:00.00	keventd
5	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd_CPU0
6	root	15	0	0	0	0	S	0.0	0.0	0:00.48	kswapd
7	root	25	0	0	0	0	S	0.0	0.0	0:00.00	bdflood
8	root	15	0	0	0	0	S	0.0	0.0	0:01.87	kupdated
9	root	25	0	0	0	0	S	0.0	0.0	0:00.00	kinoded
10	root	25	0	0	0	0	S	0.0	0.0	0:00.00	mdrecoveryd
25	root	15	0	0	0	0	S	0.0	0.0	0:00.12	kreiserfsd

.
. .
.

4. The kill command can be used to control running processes and send a variety of signals to processes. The syntax is:

`kill -signal pid`

where pid is the process ID

where -signal is one of these signals:

9 (kills the process by force)

15 (to terminate the process gracefully)

If no signal is passed, kill will use the TERM signal by default.

5. The touch and vi commands can be used.

`touch myfile` or `vi myfile`

`vi myfile`

i (to enter the insert mode of the vi editor in order to enter text)

:wq! (to save the changes made and quit the vi editor; the command prompt is returned)

6. Use the `ls -l` command.

Example:

```
penas715@ls102157:~> ls -l
```

```
total 13
```

```
drwxr-xr-x  2 penas715 users      520 2006-01-23 21:28 Documents
```

```
-rw-r--r--  1 penas715 users      45 2006-02-11 14:11 myfile
```

```
drwxr-xr-x  2 penas715 users       80 2005-09-15 06:27 public_html
```

command: ll or `ls -al`

owner of file is: penas715

permissions of file: 744

owner (read, write)

group (read)

other (read)

7. Use these commands:

```
scp myfile 10.10.102.158:~  
ssh 10.10.102.158 ls -l ~
```

Examples:

```
penas715@ls102157:~> scp myfile 10.10.102.158:~  
penas715@10.10.102.158's password:  
myfile                                100% 45   0.0KB/s  00:00
```

```
penas715@ls102157:~> ssh 10.10.102.158 ls -l ~  
penas715@10.10.102.158's password:  
total 4  
drwxr-xr-x  2 penas715 users      152 Oct  2 18:46 Documents  
-rw-r--r--  1 penas715 users      45 Feb 11 14:13 myfile  
drwxr-xr-x  2 penas715 users      80 Sep 11 22:51 public_html
```

8. Use these commands:

```
scp myfile 10.10.102.158:/tmp  
ssh 10.10.102.158 ls -l /tmp
```

Examples:

```
penas715@ls102157:~> scp myfile 10.10.102.158:/tmp  
penas715@10.10.102.158's password:  
myfile                                100% 45   0.0KB/s  00:00
```

```
penas715@ls102157:~> ssh 10.10.102.158 ls -l /tmp  
penas715@10.10.102.158's password:  
total 4  
-rw-r--r--  1 penas715 users      45 Feb 11 14:16 myfile
```

9. Use these commands:

```
ssh penas715@10.10.102.158  
exit
```

Examples:

```
penas715@ls102157:~> ssh penas715@10.10.102.158  
penas715@10.10.102.158's password:
```

Last login: Tue Jan 24 21:25:58 2006 from c-67-161-204-32.hsd1.co.comcast.net
penas715@ls102158:~>

penas715@ls102158:~/Documents> exit
logout
Connection to 10.10.102.158 closed.

10. There is a man page for every command and important system configuration file. The syntax is simply:

`man netstat`

You can use the following keys when you are viewing a man page:

Up arrow (↑)

Down arrow (↓)

q (quits man and returns you to the command prompt)

/string (used to search a man page. Press the enter key and the cursor will advance to the first match for the *string* you entered on the page. Press the n key to put the cursor on the next match of the *string*.)

Administrator Lab Manual

Regis University

Administrator Lab Manual

Virtual Machines on the IBM S/390

Contents

How to Login

Creating Student Accounts

Setting the Default Password for Student Accounts

Deleting Student Accounts

Downloading Putty SSH client & PSFTP secure FTP client

How to grant Sudo access

Additional Information

How to Login

These are the seven virtual machines available on the IBM S/390.

10.10.102.157

10.10.102.158
10.10.102.159
10.10.102.160
10.10.102.161
10.10.102.162
10.10.102.163

To login, start with option 1 or option 2 below.

Option 1:

3. Login to acadunix.regis.edu. From acadunix, login to any one of the virtual machines via ssh with the assigned user account and password. The port must be specified with the `-p` option to ssh.

```
ssh -p 5157 user@linux390.colostate.edu
```

Option 2:

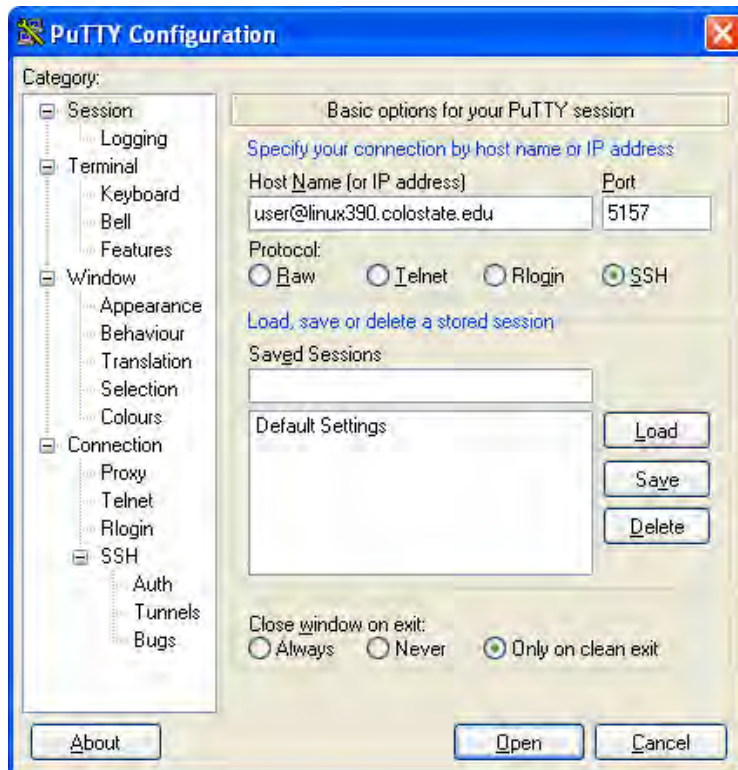
3. Use the putty ssh client to login to the virtual machines directly from a Windows system with assigned user account and password.

From the command line, enter the following command:

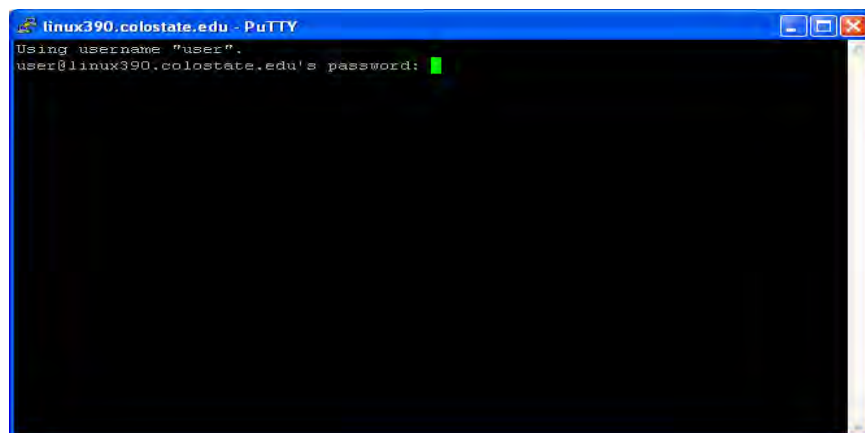
```
putty.exe -ssh -P 5157 user@linux390.colostate.edu
```

From the putty dialog box, enter the host name, including the username you will login with, and enter the port number in the designated field.

```
user@linux390.colostate.edu
```

When prompted, enter your password.



2. Once logged into one virtual machine, you can login to all the other virtual machines from there.

The virtual machines can only communicate with each other via their private network; they cannot communicate with other external systems. To login to virtual machine 10.10.102.159, for example, enter the following command:

```
ssh user@10.10.102.159
```

Creating Student Accounts

1. Obtain the student roster from Regis faculty, which lists students that need accounts to be setup. If the file in Word (.doc) format, copy and paste the contents into a text (.txt) file. Save the text file as "roster".
2. Transfer the "roster.txt" file to the 10.10.102.157 virtual machine (VM). This VM is designated as the master VM, where all account management tasks are centralized. "regisadm" is the Regis University administrator account, which has root privileges via the sudo utility.

You may use a secure copy or secure ftp client to transfer the roster.txt file to the master VM. If you do not have one of these clients, follow the "Downloading Putty SSH client & PSFTP secure FTP client" instructions in this manual.

After downloading the secure ftp client, continue here to transfer the roster.txt file.

- Open a command prompt on your PC.
- Change directories into your desktop (where you saved psftp.exe)
- Enter the following commands (in blue):

```
psftp -P 5157 regisadm@linux390.colostate.edu  
Using username "regisadm".
```

```
regisadm@linux390.colostate.edu's password: <enter the password>  
Remote working directory is /home/regisadm
```

```
psftp> cd Documents  
Remote directory is now /home/regisadm/Documents
```

```
psftp> put roster.txt  
local:roster.txt => remote:/home/regisadm/Documents/roster.txt  
psftp> quit
```

The roster.txt file has been transferred to the /home/regisadm/Documents directory on the master VM 10.10.102.157.

3. Login to 10.10.102.157 with the regisadm administrator account. (See the "How to Login" section for more details).

From your PC command prompt enter:

```
putty.exe -ssh -P 5157 regisadm@linux390.colostate.edu
```

Enter the administrator password. You will be in the /home/regisadm directory as soon as you login. Go to the Documents directory.

```
cd Documents
```

4. Now you are ready to create student accounts. From the /home/regisadm/Documents directory, execute the acct_setup.pl script and save the output to another file such as results157, in which 157 represents the 10.10.102.157 virtual machine.

```
./acct_setup.pl > results157
```

The acct_setup.pl script will read the roster.txt file that you copied in the previous steps.

The results157 file will contain a list of the accounts that were created. Here is an example:

```
schwa035, Schwarzman. Stuart A, 03F8W2_CS_465_C70
```

```
burro513, Burroughs. Michael J, 03F8W2_CS_465_C70
```

```
avena061, Avena. Ramon D, 03F8W2_CS_465_C70
```

```
COMPLETE
```

```
Script executed on Sun Jan 29 19:46:00 MST 2006
```

5. Now, create accounts on the rest of the virtual machines.

Virtual Machines:

```
10.10.102.158
```

```
10.10.102.159
```

```
10.10.102.160
```

```
10.10.102.161
```

```
10.10.102.162
```

```
10.10.102.163
```

Use secure copy to copy the roster file and acct_setup.pl script to each one of the virtual machines. For example:

```
scp acct_setup.pl roster.txt 10.10.102.158:/home/regisadm/Documents
acct_setup.pl                100% 3230   0.0KB/s  00:00
roster.txt                   100% 250    0.0KB/s  00:00
```

```
scp acct_setup.pl roster.txt 10.10.102.159:/home/regisadm/Documents
acct_setup.pl                  100% 3230   0.0KB/s  00:00
roster.txt                     100% 250    0.0KB/s  00:00
```

etc...

Execute the account setup script on each virtual machine. Save the output of the script to a "results" file for each virtual machine in order to keep track of where the accounts were created. For example, from 10.10.102.157 enter:

```
ssh 10.10.102.158 sudo /home/regisadm/Documents/acct_setup.pl > results158
```

```
ssh 10.10.102.159 sudo /home/regisadm/Documents/acct_setup.pl > results159
```

etc...

The "results" files will be saved locally on the master VM. You should have seven results files.

```
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:44 results157
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:44 results158
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:45 results159
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:45 results160
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:45 results161
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:46 results162
-rw-r--r-- 1 regisadm regis      205 2006-01-29 19:46 results163
```

If the user account already exists, and if the group already exists, the script output will show:

```
groupadd: group student exists
```

```
useradd: user schwa035 exists
schwa035, Schwarzman. Stuart A, 03F8W2_CS_465_C70
```

```
useradd: user burro513 exists
burro513, Burroughs. Michael J, 03F8W2_CS_465_C70
```

```
useradd: user avena061 exists
avena061, Avena. Ramon D, 03F8W2_CS_465_C70
```

COMPLETE

Script executed on Sat Jan 28 14:06:28 MST 2006

Test a few of the accounts to make sure you can login with the new student accounts on each virtual machine . Use the default password. For example,

```
putty.exe -ssh -P 5158 user123@linux390.colostate.edu
```

OR

```
ssh user123@10.10.102.158 (from another virtual machine)
```

Note: To set the default password, go section "Setting the Default Password for Student Accounts".

6. Delete the files you copied to each of the virtual machines. This will ensure that you are always using a recent version of the setup script and the most current roster.

```
ssh 10.10.102.158 rm /home/regisadm/Documents/acct_setup.pl  
ssh 10.10.102.158 rm /home/regisadm/Documents/roster.txt
```

If you prefer, you can login to each virtual machine to remove files locally.

```
ssh 10.10.102.158
```

7. Email the "results" file to the requestor of the accounts. The results file will contain the username, student name and the course information.

Setting the Default Password for Student Accounts

The acct_setup.pl script sets a default password for all new student accounts. The default password can be changed by executing the "pass_encrypt.pl" script.

The pass_encrypt.pl script uses the crypt() perl function to encrypt a password.

1. From the /home/regisadm/Documents directory, edit the pass_encrypt.pl script. Change the first and second arguments (line) of the script.

```
#!/usr/bin/perl  
#  
# Regis University
```

```
# SEAD 2005B, Professional Project, 2005-2006
# Author: Selene Pena
# Script name: pass_encrypt.pl
# Usage: /home/regisadm/Documents/pass_encrypt.pl
# Purpose: This script can be used to encrypt the default password for student accounts.
```

```
$VAR{pass} = "password";
$VAR{salt} = "xy";
$VAR{pass} = crypt($VAR{pass}, $VAR{salt});
print $VAR{pass};
print "\n";
```

#end of script.

The first is the password you want to encrypt, and the second argument is the salt.

The salt is any two characters which are to be used as part of the encryption's algorithm, after the password. It's stored by Unix as the first two characters of the encrypted string for later comparison.

2. Save the changes you made to the pass_encrypt.pl script and execute it. The output is the encrypted password.

```
#!/pass_encrypt.pl
```

3. After encrypting the password, insert the new encrypted password in the acct_setup.pl script. Edit and save the script. Modify the \$pass variable.

>>excerpt from acct_setup.pl script

```
#The PASSWORD is a temporary default password. Each student must change #this password
for their own account.
my $pass = "grKQn3VBMkqic";
```

Deleting Student Accounts

When the student accounts and home directories are no longer needed at the end of the term, they can be deleted.

1. Obtain a list of the user accounts, including usernames, from the instructor of the course that the accounts are no longer needed for.

2. Create a text file called "delete_roster" and enter only the usernames of the account list you received. For example:

```
schwa234
burro987
avena423
smith537
barne967
```

3. Transfer the delete_roster.txt file to the master VM 10.10.102.157 by using PSFTP to securely FTP the file from your PC.

```
psftp -P 5157 regisadm@linux390.colostate.edu
Using username "regisadm".
```

```
regisadm@linux390.colostate.edu's password: <enter the password>
Remote working directory is /home/regisadm
```

```
psftp> cd Documents
Remote directory is now /home/regisadm/Documents
```

```
psftp> put delete_roster.txt
local:roster.txt => remote:/home/regisadm/Documents/roster.txt
```

```
psftp> quit
```

4. Login to the master VM 10.10.102.157 and execute the acct_remove.pl script.

```
regisadm@ls102157:~> pwd
/home/regisadm
regisadm@ls102157:~> cd Documents
regisadm@ls102157:~/Documents> ./acct_remove.pl
```

The output of the script will show which accounts were removed. For example:

```
Deleting account schwa035
...
schwa035
  removed
Deleting account burro513
...
burro513
  removed
```

Deleting account avena061

...
avena061
removed

5. To delete accounts on any of the virtual machines, copy the delete_roster.txt and acct_remove.pl script to all VMs and then execute the script remotely.

For example, copy the files to VM 10.10.102.159:

```
regisadm@ls102157:~/Documents> scp acct_remove.pl delete_roster.txt \
10.10.102.159:/home/regisadm/Documents
  acct_remove.pl                100% 784   0.0KB/s  00:00
  delete_roster                 100%  45   0.0KB/s  00:00
```

Next, execute the script remotely and save the output to a local file:

```
regisadm@ls102157:~/Documents> ssh 10.10.102.159 sudo
/home/regisadm/Documents/acct_remove.pl > deleted158
```

Remove the files you copied from each virtual machine.

```
regisadm@ls102157:~/Documents> ssh 10.10.102.158 rm ~/Documents/acct_setup.pl
regisadm@ls102157:~/Documents> ssh 10.10.102.158 rm ~/Documents/roster.txt
```

Downloading Putty SSH client & Psftp secure FTP client

To download the PSFTP (secure ftp) client to your PC:

1. Go to the PuTTY download page via this URL:
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
2. Scroll down to the Binaries section, under "*The latest release version*" section. Right click on "psftp.exe," and chose "Save Target As".

Save "psftp.exe" to your Windows desktop. Now you should have a PSFTP icon on your desktop.

3. PSFTP is ready to be used.

To download the SSH client on your PC, follow the same steps above to download "putty.exe".

How to grant Sudo access

The sudo utility allows a user to run commands as another user such as root. Root privileges should only be given to students when it is approved and requested by the instructor. Sudo logs all commands that are executed, which is helpful for tracking user activity.

1. Edit the /etc/sudoers file with the visudo command:

```
#/usr/sbin/visudo
```

2. Enter the accounts that require root capabilities.

For example, to give sudo access to all students in the "student" group (in /etc/group) insert this entry:

```
%student ALL=(ALL) ALL
```

To grant sudo access to an individual user, insert this entry:

```
username ALL=(ALL) ALL
```

To grant sudo access to an individual user, include NOPASSWD:

```
username ALL=(ALL) NOPASSWD: ALL
```

3. All commands run as sudo are logged in the log file /var/log/messages. These logs should be reviewed frequently. You can grep for all sudo commands run by any user:

```
#cat /var/log/messages | grep sudo
```

4. For more information on restricting access for users, view the sudo man page:

```
#man sudo
```

Additional Information

The IBM S/390 system is managed by Colorado State University faculty and graduate students. To obtain the root password or to request a rebuild of any of the instances provided to Regis University, contact Dan Likarish at dlikaris@regis.edu. Dan Likarish is the primary point of contact for communication with CSU faculty.