Final Project

Courtney Hagen

W0263284

OSYS1000

Table of Contents

[Introduction 3](#_Toc6362561)

[Part One 3](#_Toc6362562)

[Task One - Installation 3](#_Toc6362563)

[Task 2 – Perform System Updates 4](#_Toc6362564)

[Task 3 – Configure a Remote Client and Server through SSH 4](#_Toc6362565)

[Task 4 – Configure a Desktop Client 4](#_Toc6362566)

[Task 5 – Configure a Remote Desktop Client and Server through VNC 5](#_Toc6362567)

[Task 6 – Configure File Sharing through SMB 5](#_Toc6362568)

[Troubleshooting 8](#_Toc6362569)

[Part Two – Choice #2 – Configure a custom rescue thumb drive 9](#_Toc6362570)

[Creating the Drive 10](#_Toc6362571)

[Software Table 11](#_Toc6362572)

[References 12](#_Toc6362573)

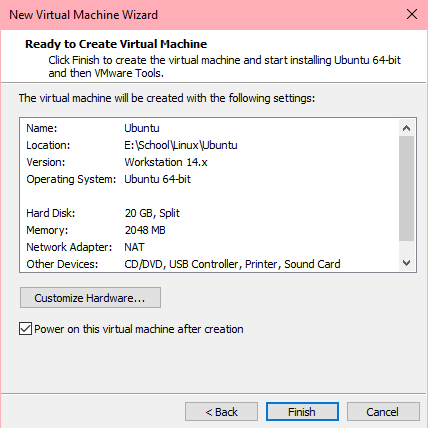
# Introduction

In this document will be a representation of the material I have learned this semester in OSYS1000. There will be two main parts, the first of which will be exploring remote access of a Linux distribution that I have not used thus far in the class. The second will be to make a rescue thumb drive.

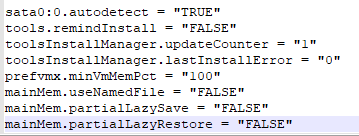
# Part One

## Task One - Installation

For the purposes of this assignment, I have decided to install Ubuntu.



This image demonstrates the specifications used for the virtual machine.



The image above demonstrates the lines added to the end of the .vmx file.

## Task 2 – Perform System Updates

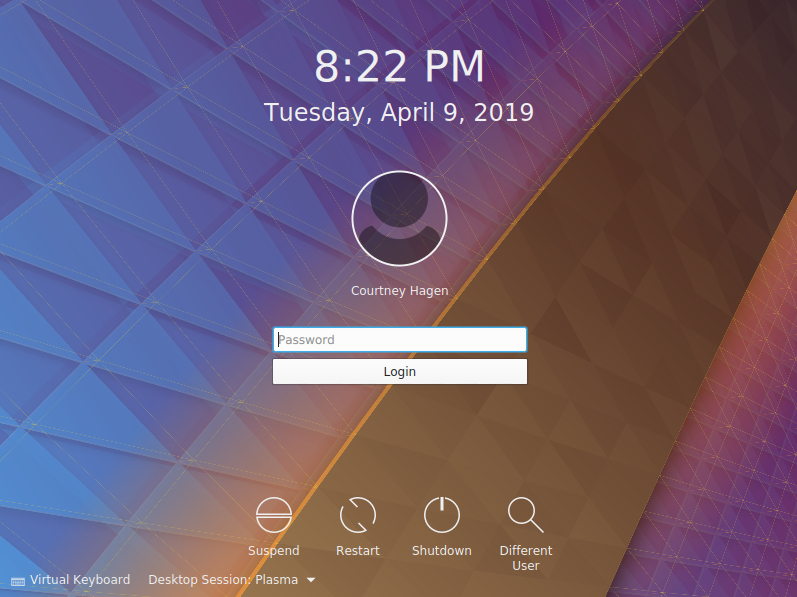
The command I used in the terminal to perform the updates was sudo apt update && sudo apt upgrade. To set the default runlevel to 3, I first created a file on the /etc/ directory called inittab using the command sudo touch inittab in the directory. After that, I added to the file in vi by doing sudo vi inittab, adding “:id:3:initdefault:” to the file, and using :wq! to override the read-only state of the file. (Sobell, 2008)

## Task 3 – Configure a Remote Client and Server through SSH

To install SSH on Ubuntu I used the command “sudo apt install openssh-server”. (How to Enable SSH on Ubuntu 18.04, 2018) To install it on Linux Mint I tried the same command and it worked. To configure the firewall, I used the command “sudo ufw allow ssh”. I also enabled ufw using the command “sudo ufw enable”.

## Task 4 – Configure a Desktop Client

I chose to install KDE on my Ubuntu VM. To do this I used the command “sudo apt install kubuntu-desktop”. (Wallen, 2011) To edit the runlevel, I went back in the same file I created before, and edited to contain “:id:5:initdefault”.

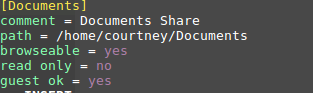


I then, as demonstrated above, was able to restart and have the VM boot directly into the KDE desktop.

## Task 5 – Configure a Remote Desktop Client and Server through VNC

To install VNC server on my Ubuntu VM, I used the command “sudo apt install vnc4server”. To start VNC, I typed vnc4server in the terminal. It prompted me to make a password and confirmed it. To configure the firewall, I used the command “sudo ufw allow 5901/tcp” as VNC listens on port 5901. I did this to both VMs. (Kumar, 2017) To set up the Linux Mint VM to use the VNC server, I used the command “sudo apt install vncviewer”. After this, it prompted me to put the IP of the VNC server from the Ubuntu VM. I did this, using a colon and the port number to connect to the VNC server.

## Task 6 – Configure File Sharing through SMB

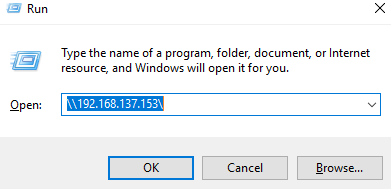
To install Samba on my Linux Mint VM, I used the command “sudo apt install samba”. I then used an elevated vi to modify the samba configuration file in /etc/samba/smb.conf. 

I then changed the permissions by using sudo chmod 0777 /home/courtney/Documents

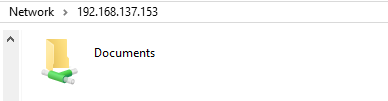
This is the text I added to the smb.conf file to create the share. I then restarted by executing the smbd file by typing it’s path sudo /etc/init.d/smbd restart in the terminal.

As far as the firewall goes, I used sudo ufw allow on ports 137, 138, 139, and 445.

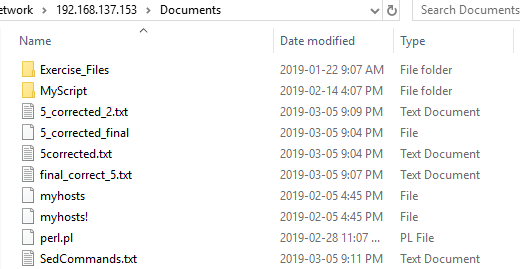
To access the new share, I typed “run” after pressing the start button, typing //(Linux IP)/ as show below. (Mejia, 2011)



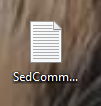
It then opened the Documents share as shown below.



I then, was able to copy my SedCommands.txt file from my share

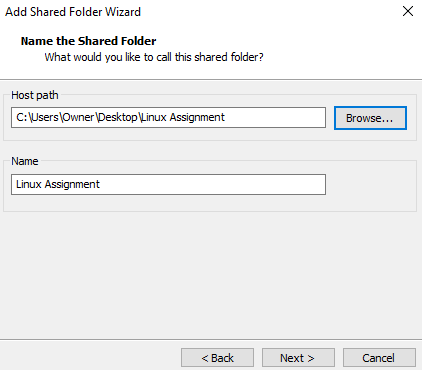


To my Windows desktop on my host.

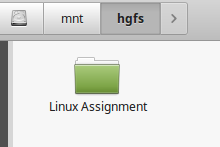


I then created a directory on my Windows desktop called “Linux Assignment” and made it a shared folder using Advanced Sharing.

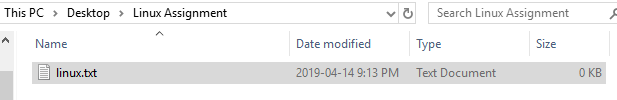
I then enabled sharing for the Linux Mint VM through VMware



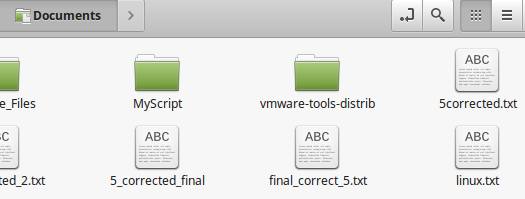
The share then showed up in the appropriate folder as shown below. (Holmes, 2017)



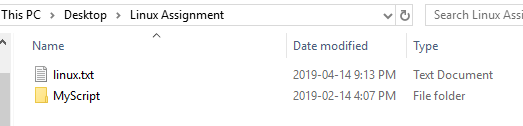
I created a linux.txt file on my Windows machine in the Linux Assignment share.



Which I then copied to the Documents folder on the Linux VM.



I then copied the MyScript folder to the Linux Assignment share from the Linux VM which can be seen on the Windows host.



# Troubleshooting

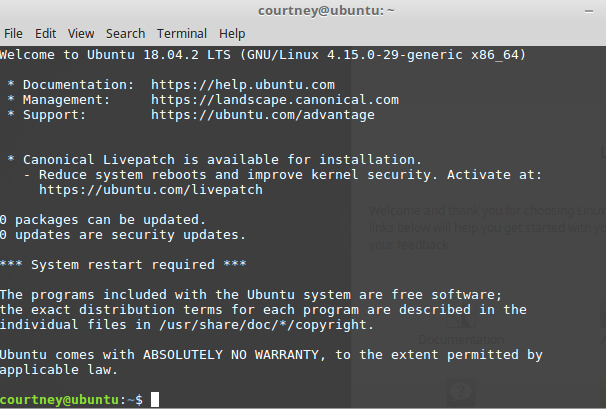
While trying to use the terminal to update, I got this error: Could not get lock /var/lib/dpkg/lock - open (11: Resource temporarily unavailable)

E: Unable to lock the administration directory (/var/lib/dpkg/), is another process using it?

I tried manually killing the processes using the kill command, but that didn’t work, and I didn’t want to remove the lock on it, so I opened the software update manager to see if it would stop the process, and it worked.

To set up the firewall to allow connection via SSH, I first used the command “sudo ufw default deny incoming”, and then “sudo ufw default allow outgoing”. Then, to allow SSH, I used “sudo ufw allow ssh”. Finally, to enable the configured firewall, I used “sudo ufw enable”. (Boucheron, 2018) I used the same commands on my Linux Mint VM.

To connect to my Ubuntu VM using my Linux Mint VM, I got the Ubuntu VM’s IP address by using the command “ip address” then on the Linux Mint VM, I used the command “ssh [courtney@192.168.199.132](mailto:courtney@192.168.199.132)” which was my IP address at the time for the Ubuntu VM. It prompted me to put in my Ubuntu VM’s password in, then I had access via SSH. (How To Use SSH to Connect to a Remote Server in Linux or Windows, 2019)



To do this from the Ubuntu VM to the Linux Mint VM, it would be the same commands, just with the username created in Linux Mint and the Linux Mint VM’s IP address.

# Part Two – Choice #2 – Configure a custom rescue thumb drive



## Creating the Drive

To create my custom rescue thumb drive, I decided to build with Arch Linux as it is very minimalistic and I could really build it from nothing and make it what I needed it for. To do this, I used two USBs, one that was formatted and one that was made bootable using Rufus with an Arch Linux ISO. I booted from this USB, which gives you a live version of Arch Linux to use to install on whatever partition is chosen. I partitioned my USB, using only one partition due to it being a rescue drive. I first used the lsblk command to decipher what the right disk was to create the partition, then I used the fdisk tool to create the primary partition. I then formatted it with the appropriate filesystem, that being ext4. I used the command mkfs.ext4 /dev/sdb to do this. I then mounted it using mount /dev/sdb /mnt, and made the home directory using mkdir /mnt/home. Before installing, I had to secure an internet connection. I did this both using wired and wireless connections. With a wired connection, it found the connection automatically, but with wireless I had to configure it. All I did was use the command wifi-menu and it provided a GUI to connect to the Wi-Fi easily.

Then, I finally installed the operating system to the USB by using the command pacstrap /mnt base base-devel. After this was done, I generated a fstab file to handle disk management on the new operating system by using the command genfstab -U /mnt >> /mnt/etc/fstab. I then used arch-chroot /mnt to make appropriate changes in the newly installed operating system, using the exit command when it wouldn’t allow me to do something in chroot. Next, I set up the locale, otherwise there would be issues with certain applications when the desktop environment is installed. I used locale-gen, echo LANG=en\_US.UTF-8 > /etc/locale.conf, and export LANG=en\_US.UTF-8 to do this. Next, I created a hostname by using the command echo CourtneysRescue > /etc/hostname and echo CourtneysRescue > /etc/hosts.

The next important step in insuring the drive is bootable is to install the bootloader. I did this by first downloading the packages using pacman -Sy grub, then installing it using pacman -S grub, (I attempted just to use the latter command, but it threw an error and told me to use the first one first.) I then used the command grub-install /dev/sdb –force, and finally, made a grub configuration file by using the command grub-mkconfig -o /boot/grub/grub.cfg. Lastly, I created a root password by using the passwd command.

Next, I decided to proceed with the installation of the Gnome desktop to the rescue drive. To do this, I first installed Xorg display server so the desktop would work. I did this using the pacman -S xorg xorg-server command. After this, I used the pacman -S gnome gnome-extra command to install the Gnome desktop. Lastly, to insure it would always boot to this desktop, I used the command systemctl enable gdm.service. To go into the desktop after the install, I used the command systemctly start gdm.service so I didn’t have to reboot. (Kumar, 2018)

Next, I decided to install ntfs-3g so that the rescue drive is compatible with NTFS. I did this using the command pacman -S ntfs-3g. For IDE, SATA, SCSI, and RAID support, I installed GParted using the same command, changing the package name only, as I did with all of the software I installed in the table below.

To modify the GRUB boot screen, I installed Grub Customizer. It came with a default theme called “starflight” which I used. I only modified the background image to include my welcome message, name, and W number. I used this same image modified with my name and the version of Linux I used as my desktop background.

## Software Table

|  |  |  |
| --- | --- | --- |
| Software | Function | Reason for inclusion |
| SSHFS | Used to mount remote file systems over SSH. (How To Use SSHFS to Mount Remote File Systems Over SSH, 2016) | Easier to modify files via SSH on the fly. (How To Use SSHFS to Mount Remote File Systems Over SSH, 2016) |
| FSarchiver | Creates compressed archive files of the contents of a file system. (Filesystem Archiver for Linux, n.d.) | Useful to make images on popular file systems. (Filesystem Archiver for Linux, n.d.) |
| nmap | Scans networks/hosts to discover security issues. (Nmap, n.d.) | Useful in determining security problems on a machine. (Nmap, n.d.) |
| Xinetd | Daemon sed for starting services based on requests over a network. (Negus, 2002) | Protection against port scanners and other network security issues while recovering a machine. (Negus, 2002) |
| Samba | Used for sharing between Linux and Windows machines. (Samba, n.d.) | Useful in an organization using both Linux and Windows machines. (Samba, n.d.) |
| Gparted | Used for managing disks. (Creating partitions, shrinking partitions, data rescue, etc.) (Gedak, n.d.) | Useful for disk management on a single device. |
| GRUB | Allows the control of the computer to be transferred to the operating system kernel. Also allows for multiple operating systems on the same system. (Dubbs, n.d.) | Simple and small bootloader to allow Arch Linux to boot from the USB to simplify use. |

# References

Boucheron, B. (2018, July 05). How To Set Up a Firewall with UFW on Ubuntu 18.04. Retrieved April 7, 2019, from <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-18-04>

Dubbs, B. (n.d.). GNU GRUB. Retrieved April 17, 2019, from https://www.gnu.org/software/grub/

Filesystem Archiver for Linux. (n.d.). Retrieved April 17, 2019, from <http://www.fsarchiver.org/>

Gedak, C. (n.d.). GParted. Retrieved April 17, 2019, from https://gparted.org/

Holmes, S. (2017, May 11). How to share folders between Windows and Ubuntu using VMware Player. Retrieved April 14, 2019, from https://theholmesoffice.com/how-to-share-folders-between-windows-and-ubuntu-using-vmware-player/

How to Enable SSH on Ubuntu 18.04. (2018, December 13). Retrieved April 7, 2019, from <https://linuxize.com/post/how-to-enable-ssh-on-ubuntu-18-04/>

How To Use SSHFS to Mount Remote File Systems Over SSH. (2016, November 09). Retrieved April 17, 2019, from https://www.digitalocean.com/community/tutorials/how-to-use-sshfs-to-mount-remote-file-systems-over-ssh

How To Use SSH to Connect to a Remote Server in Linux or Windows. (2019, April 06). Retrieved April 7, 2019, from <https://phoenixnap.com/kb/ssh-to-connect-to-remote-server-linux-or-windows#htoc-how-to-connect-via-ssh>

Installing SystemRescueCd on a USB stick. (n.d.). Retrieved April 14, 2019, from <http://www.system-rescue-cd.org/Installing-SystemRescueCd-on-a-USB-stick/>

Kumar, A. (2018, January 22). How to Install Arch Linux [Step by Step Guide]. Retrieved April 17, 2019, from https://itsfoss.com/install-arch-linux/

Kumar, S. (2017, September 04). Installation of VNC server on Ubuntu. Retrieved April 9, 2019, from <https://sunnykrgupta.github.io/installation-of-vnc-server-on-ubuntu.html>

Mejia, A. (2011, July 13). How to set up Samba in Ubuntu/Linux, and access it in Mac OS and Windows. Retrieved April 12, 2019, from <https://adrianmejia.com/blog/2011/07/12/how-to-set-up-samba-in-ubuntu-linux-and-access-it-in-mac-os-and-windows/>

Negus, C. (2002). "Running Network Services". *Red hat Linux 8 Bible*. Wiley Publishing Inc. p. 1007. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [0-7645-4968-5](https://en.wikipedia.org/wiki/Special:BookSources/0-7645-4968-5).

Sobell, M. (2008, February 08). Ubuntu's Upstart event-based init daemon. Retrieved April 7, 2019, from <https://www.linux.com/news/ubuntus-upstart-event-based-init-daemon>

Wallen, J. (2011, September 13). DIY: Replace Ubuntu Unity with a different desktop. Retrieved April 9, 2019, from https://www.techrepublic.com/blog/diy-it-guy/diy-replace-ubuntu-unity-with-a-different-desktop/