Lab 02 – Navigating the Terminal

Raymond Ng: JQG999
IS 1003 Spring 2021
February 25, 2021

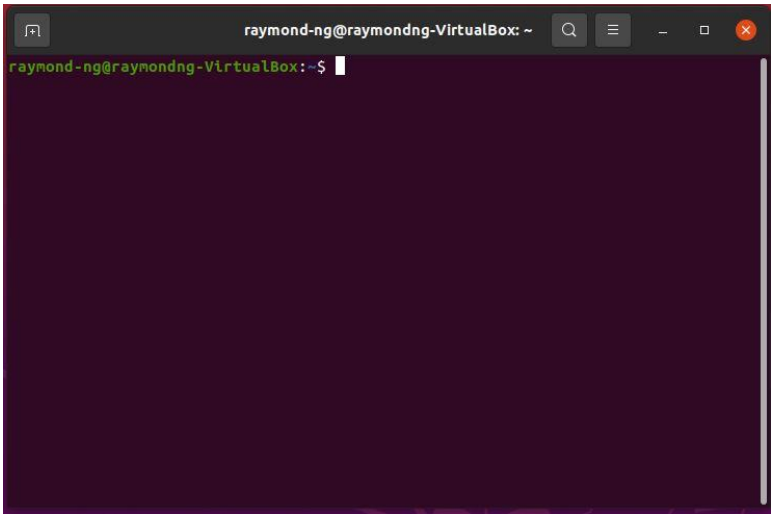
INTRODUCTION

The purpose of this lab is to provide students with the opportunity to experiment and become more familiarized with using the Command Line Interface (CLI). In this lab, students will be introduced to an array of rudimentary commands/arguments to navigate the CLI, specifically in Unix, as it applies to foundational knowledge of the cybersecurity field.

PROCESS

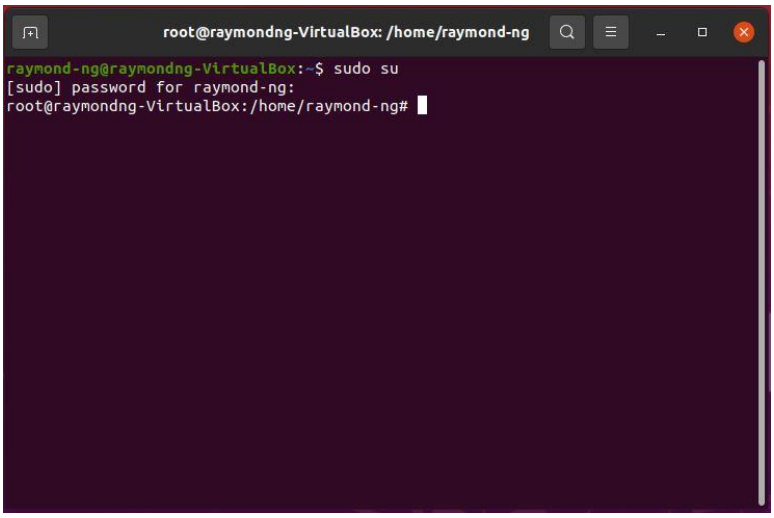
Setting Up

Before I began the lab I read through the reference material found on linuxcommand.org just to provide myself with some background knowledge before I executed each step of this lab. Especially to be more familiarized with terminology like “the Shell”. (Shotts, 2021)

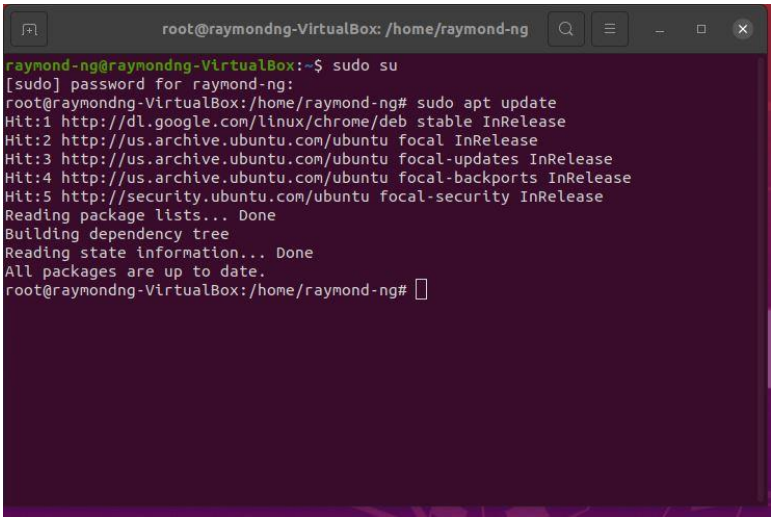


Revved up my Ubuntu browser via VM installed and setup from Lab01 and opened the terminal using the keys ‘CTRL’ + ‘ALT’ + ‘t’.

Part I: Administrative Privileges

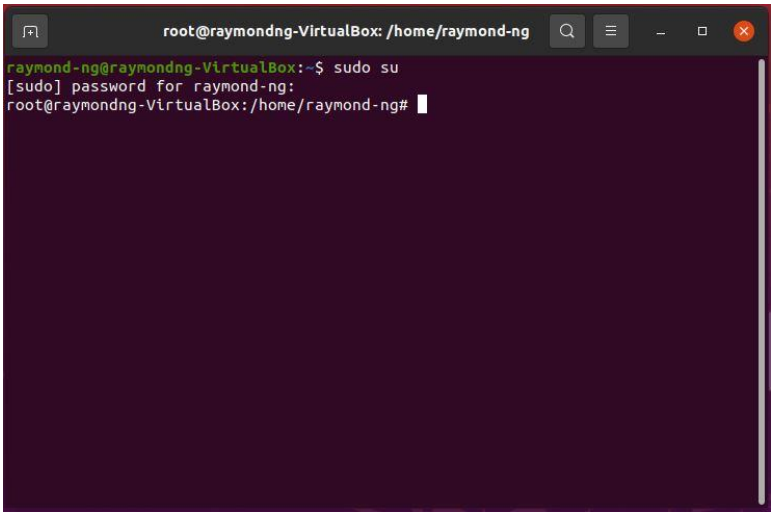


Per the instructions of Lab02, I entered the `sudo su` (switch user and do this command) command to gain root privilege. After the `sudo su` command was entered, I was prompted to enter the user password I created for my Ubuntu user profile. No observable issues to report on this step.

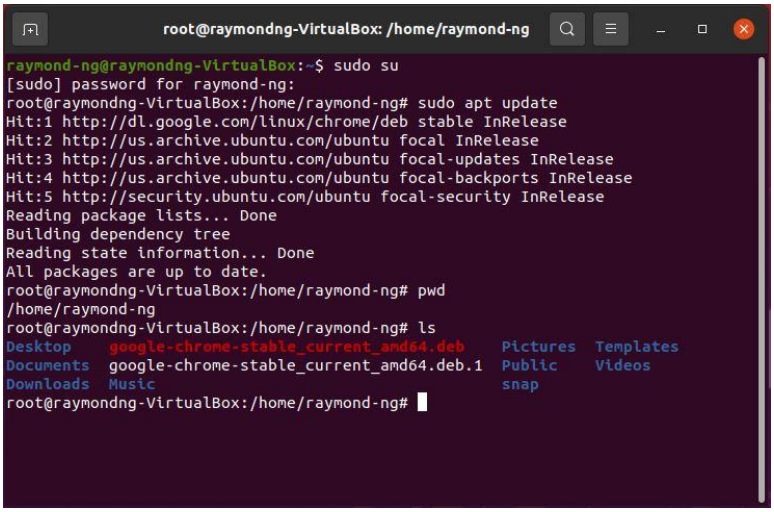


I updated the machine with the `sudo apt update` command. It appears Guest Additions are already set up so no further steps were required. No observable issue to report on this step.

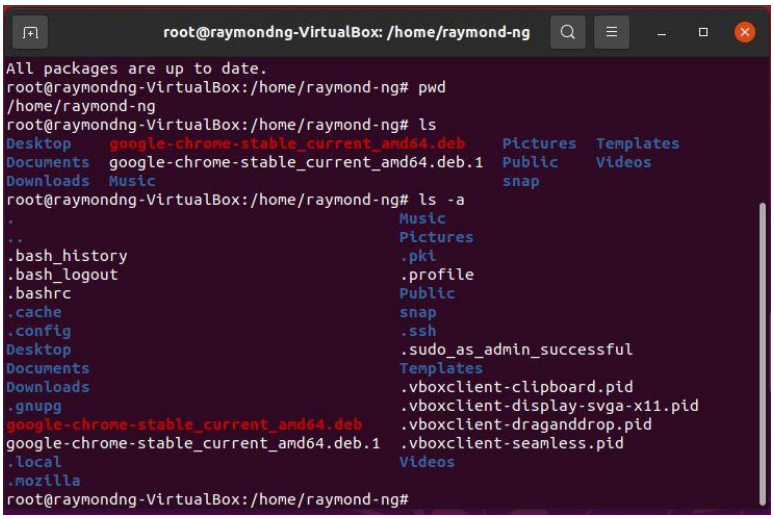
Part II: Looking at Your Directories



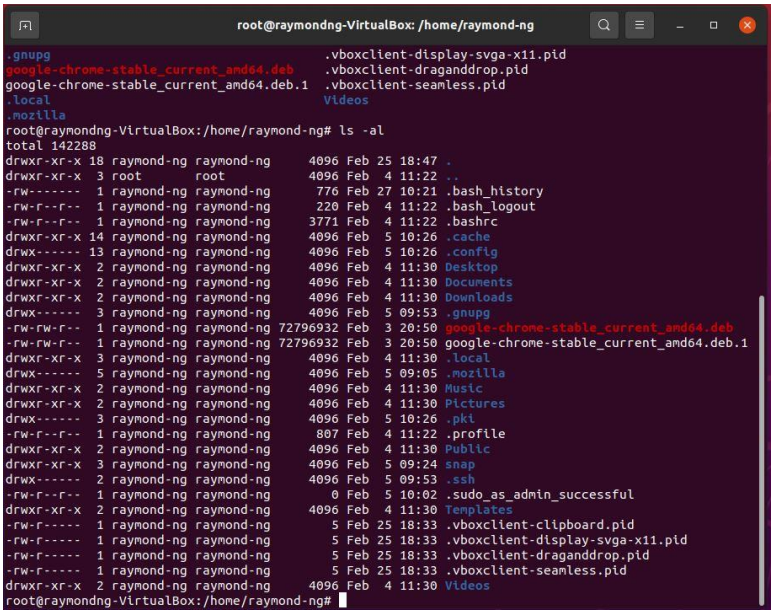
Typed the `pwd` (present working directory) command to locate my home directory which was located at `/home/raymond-ng`.



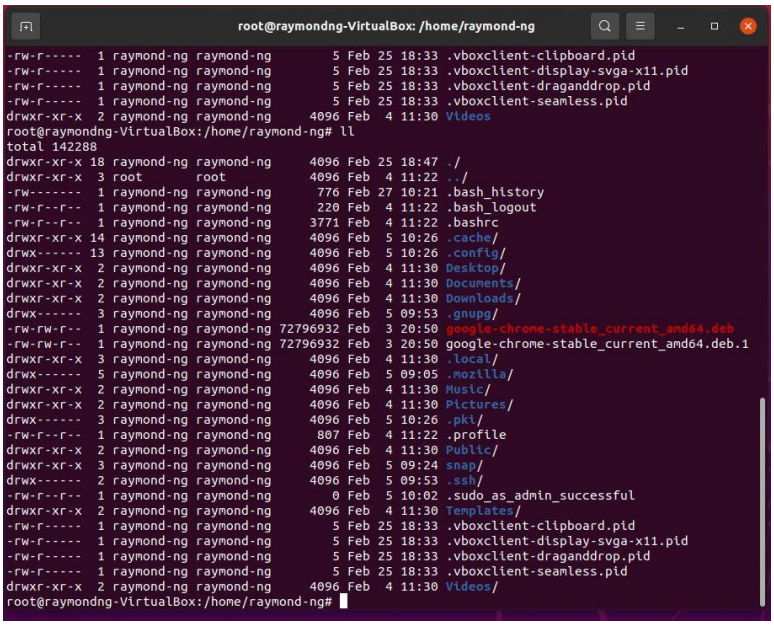
In order to list the contents of my home directory I used the `ls` (list) command. I noticed “google-chrome-stable_current_amd64.deb” was listed in the directory. I had previously downloaded Google Chrome browser from the previous lab, Lab01, using the command line.



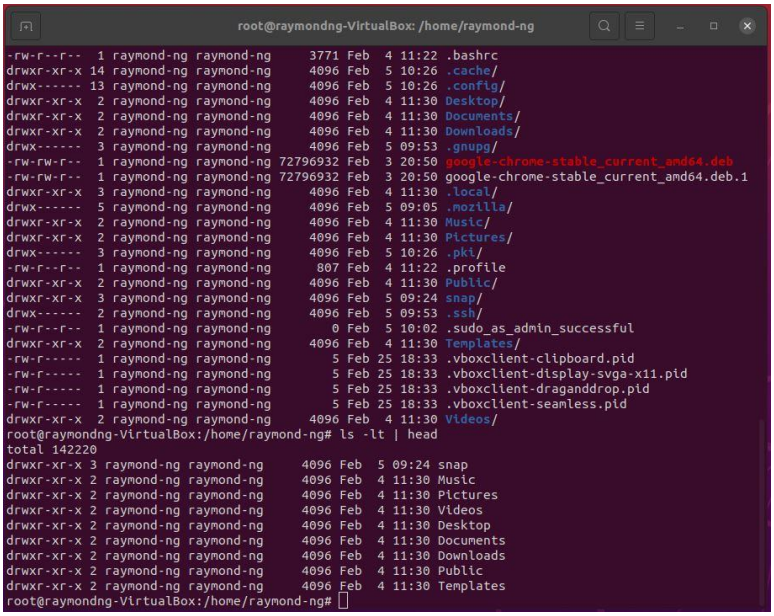
In this step I was instructed to use the `ls -al` (list all) command to list all objects in the directory. All of the objects, including hidden files and directories were depicted.



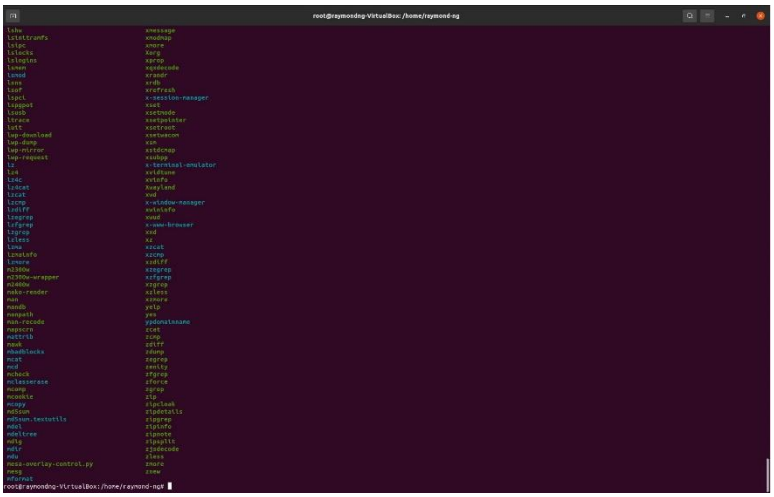
To view a bigger picture of the contents of the current directory I used the `ls -al` (list long) command as instructed. All the contents in this directory were not only listed, but owner and size of each object was depicted as well.



Per the block of instructions, I typed the shortcut `ll` command (shortcut command to `ls -l` command) and it depicted the same results.

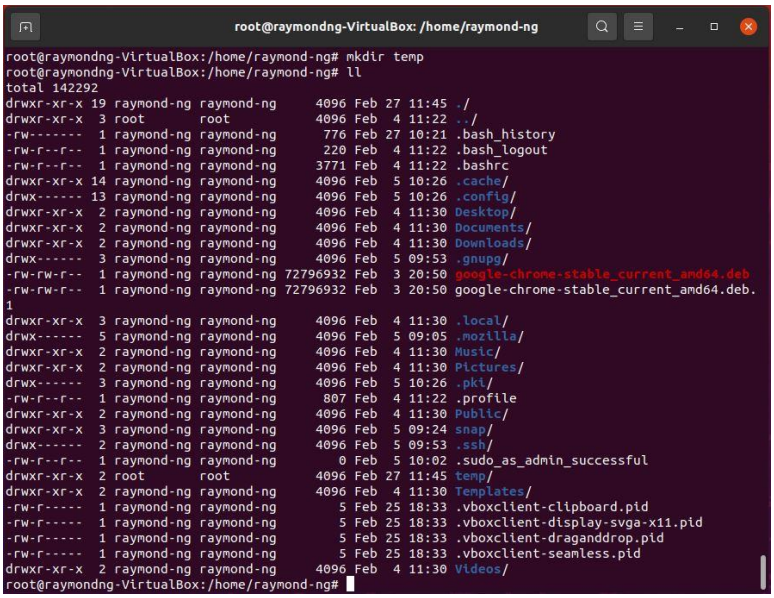


Here I was instructed to use the `ls -lt | head` (head command) to display the 10 newest files on the current directory.

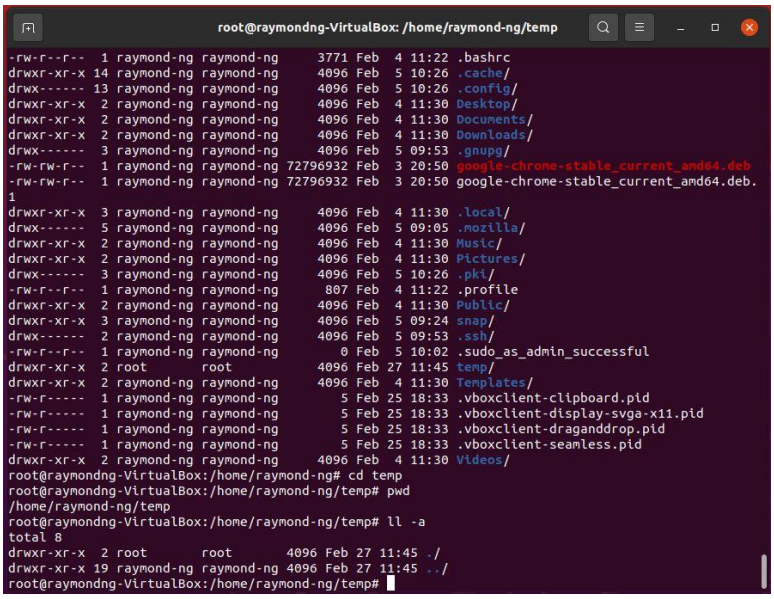


For the last step of Part I, I used the `ls /usr/bin` command to list the contents of the `bin` (binary) directory. The command yielded an overwhelmingly large list. The list was so long that I had to switch to a maximized window and still could not depict the entire list on the command terminal window.

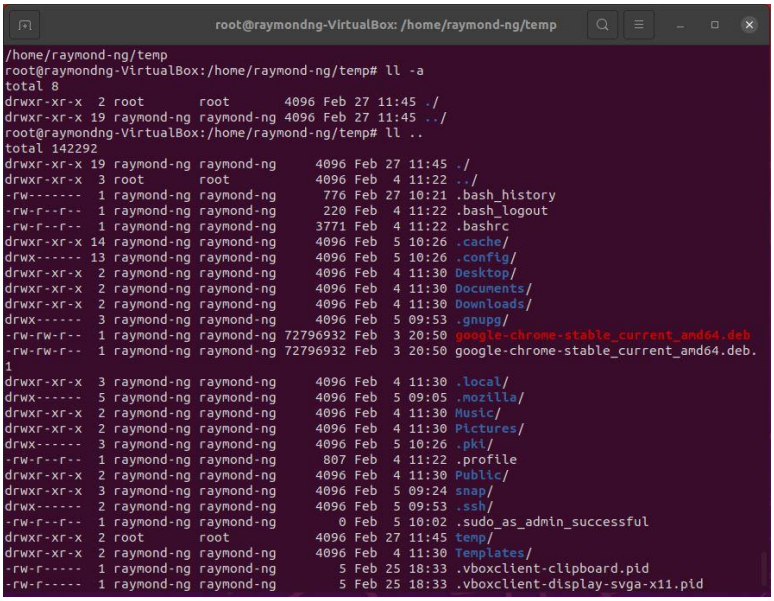
Part III: Creating and Navigating Directories



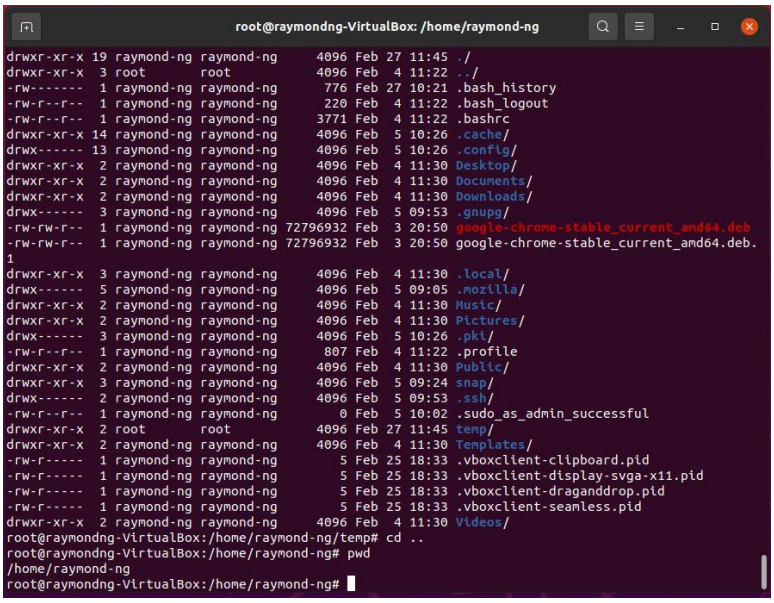
In Part III, the first step was to create a directory in my home directory by using the `mkdir` command. No observable issues. The results yielded from this command look the same as when I used the `ll` command from Part II of this lab.



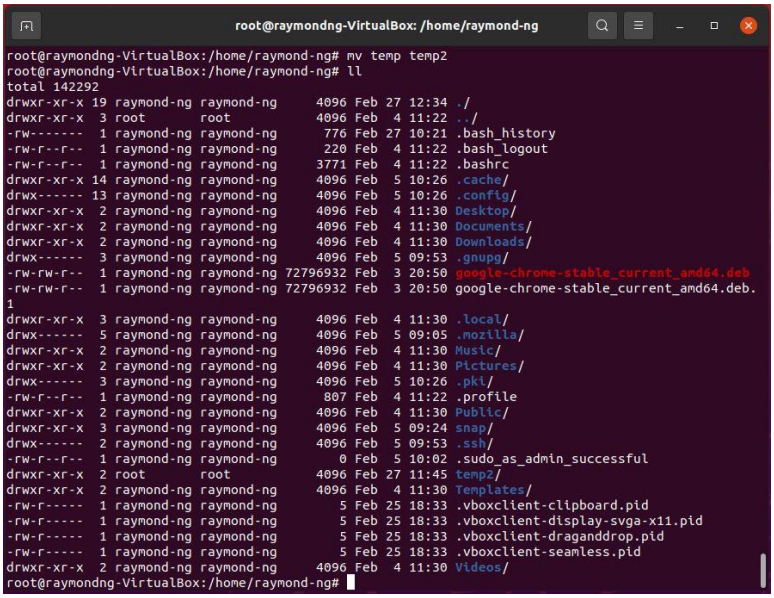
Here I changed the directory using the `cd` command and listed all the contents of the new directory with the `ll -a` command. At least two directories were depicted in this step.



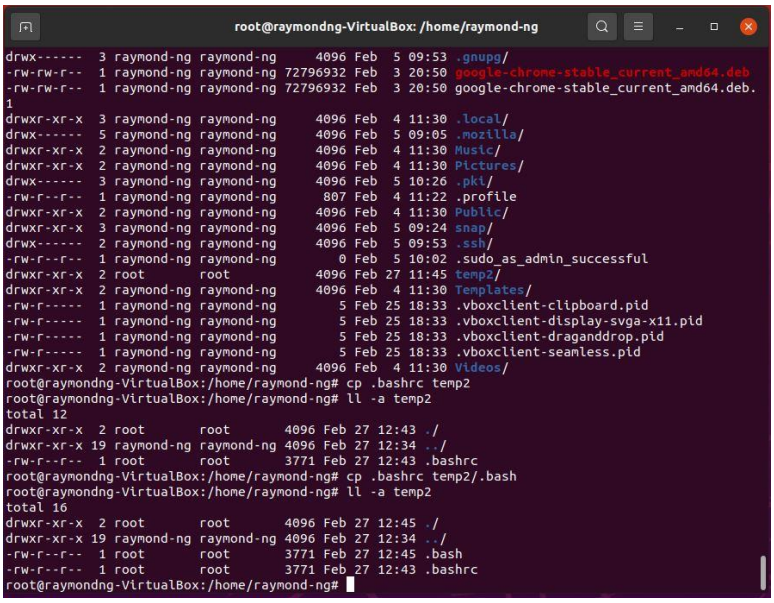
In this step the dotdot directory argument was executed. Here I used the `ll ..` command to list the contents of the parent directory.



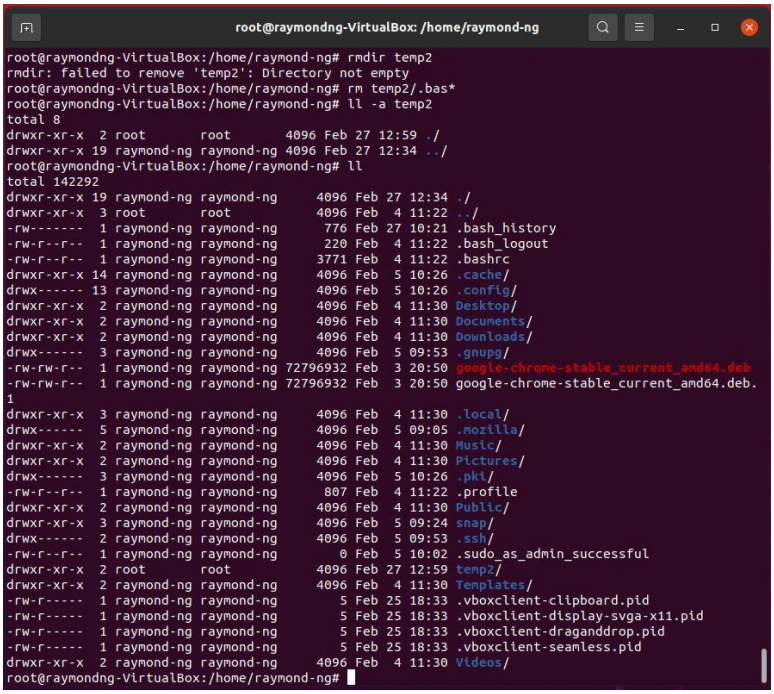
Continuing with the use of the dotdot directory argument, here I used the `cd ..` command to change my working directory to a parent directory and verified it with the `pwd` command.



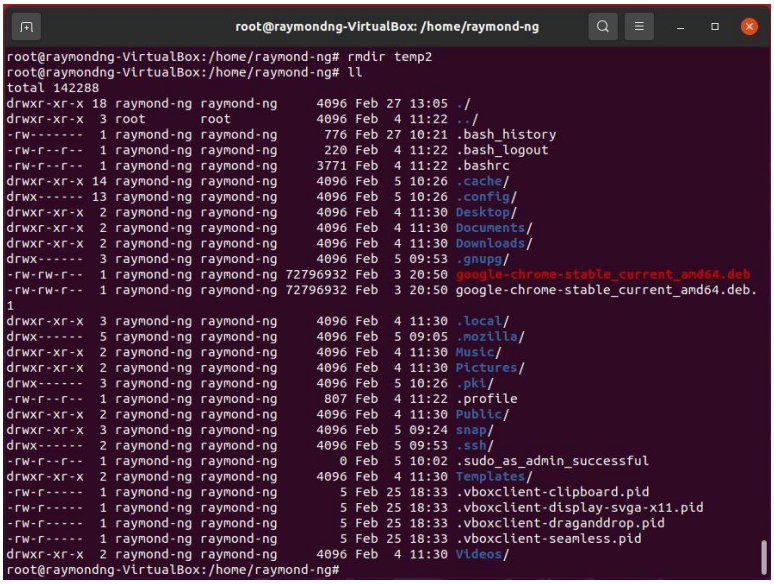
In this step I renamed the temp directory to `temp2` by using the `mv temp temp2` command and listing it with `ll` command. I chose the same nomenclature for my temp directory as the one indicated in the lab instructions because I thought for instructional purposes it would be easier.



In this step I used the `cp` command to copy one of the files in my temp2 directory, specifically `cp .bashrc temp2`. I verified the listing using `ll -a temp2` command. The hidden file was later copied again and renamed using `cp .bashrc temp2/.bash`. The name of the temp file was renamed to `.bash` and it was verified again using `ll -a temp2`.

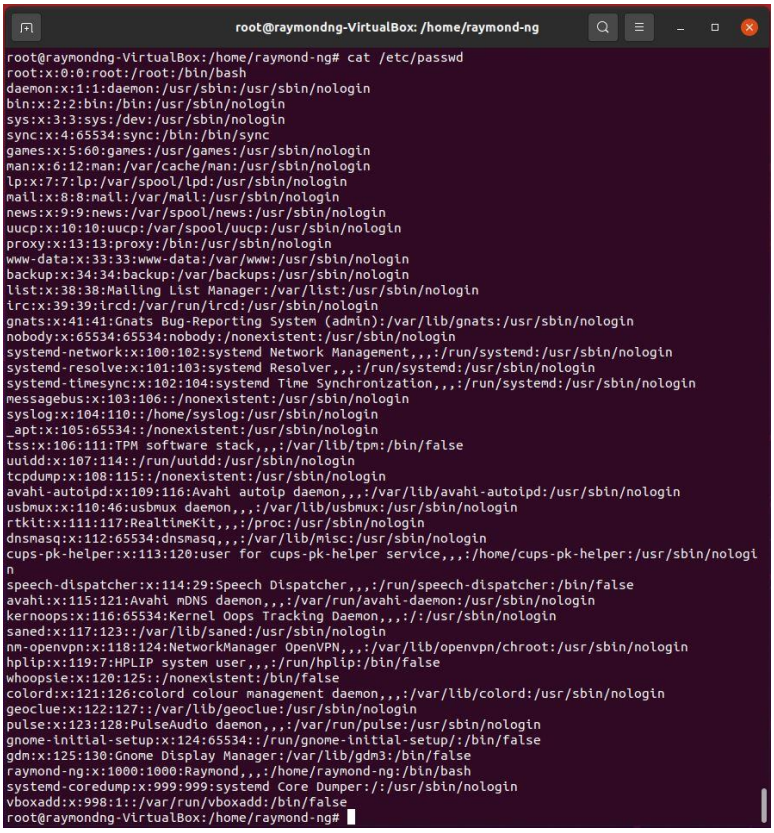


Lastly, in Part III, I followed the instructions to use the `rmdir` command to delete the renamed directory. The first attempt failed because the files still existed in the `temp2` directory. In order to remove the files in the `temp2` directory I used the `rm temp2/.bas*` command. Verified my work using `ll -a temp2` command.

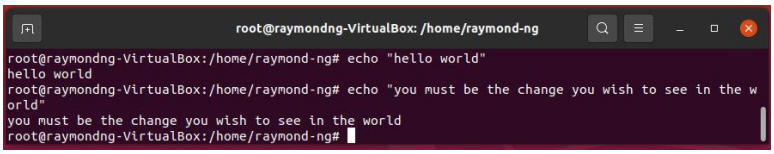


After deleting the files from the `temp2` directory I could finally remove the directory using the `rmdir temp2` command and I used `ll` command to verify my work.

Part IV: Looking at Text

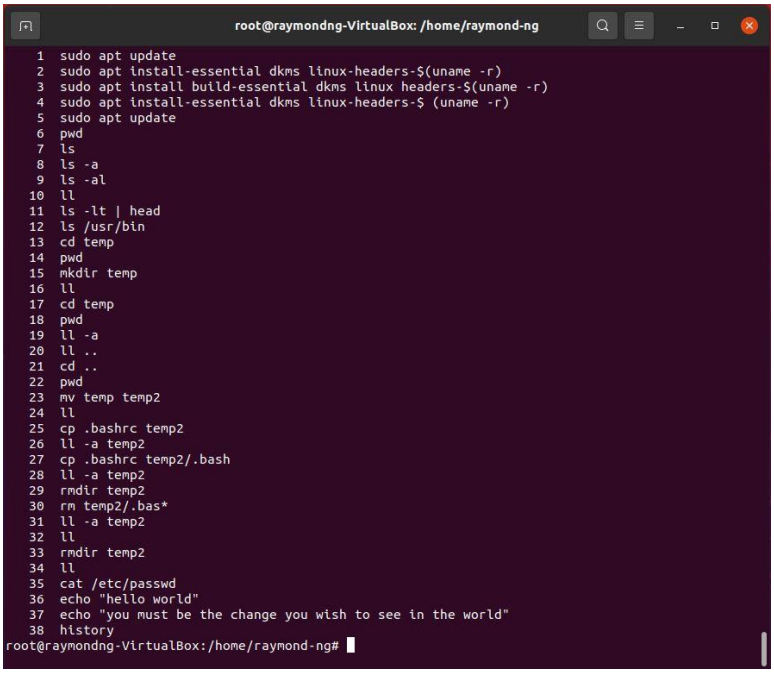


In this step I explored the use of the `cat` (concatenate) command to display the contents of text file to the screen. Here we are displayed the contents of the password file in the *etc* directory by using `cat /etc/passwd` command.



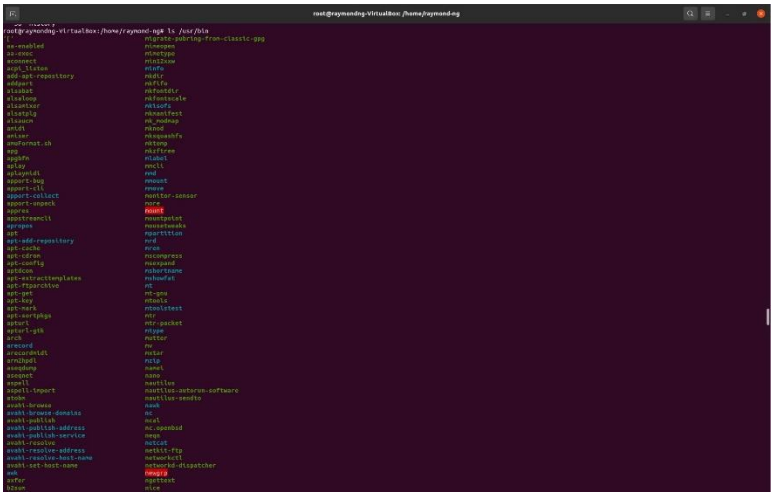
Lastly, Part IV, I explored the use of the `echo` command. First I wanted to try out the `echo "hello world"` command per the instructions just to see what would happen in the command terminal window. After I typed in a famous Ghandi quote `echo "you must be the change you wish to see in the world"`.

Part V: Useful Shortcuts

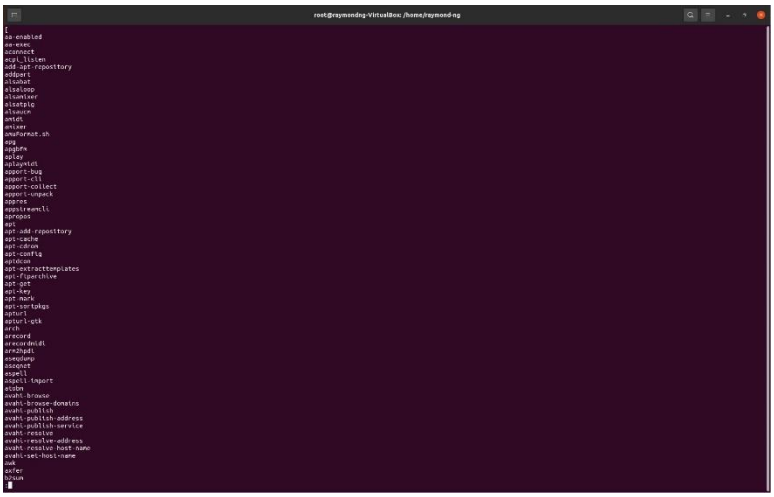


In Part V, I explored the use of the `history` command and additional shortcuts to include using the 'Up' and 'Down' arrow to reuse previous commands and the TAB key to auto-fill commands. Everything executed as they should per the instructions. No observable issues were found in Part V.

Part VI: Pipes



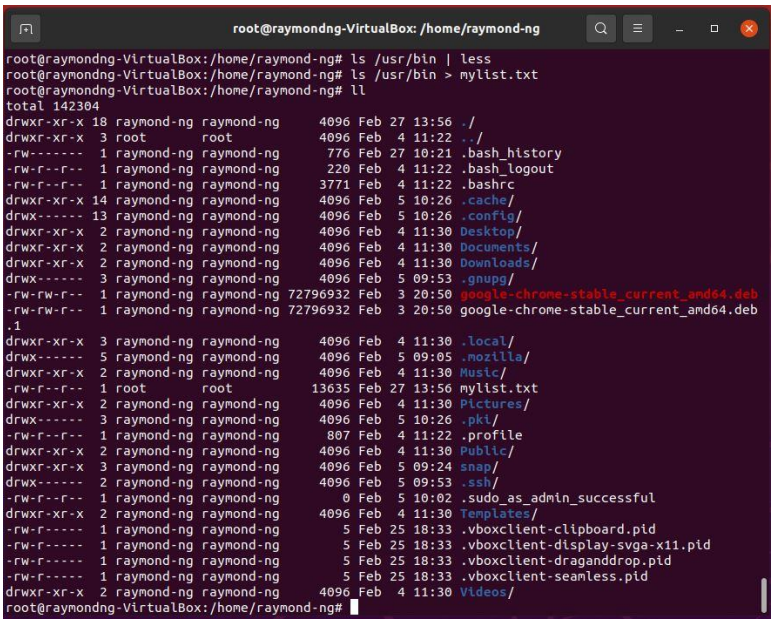
In Part VI, I learned how to slow down a command output by "piping" it to another command. Per the instructions I typed the following command into the terminal: `ls /usr/bin`. This was the same screen I saw in a previous step.



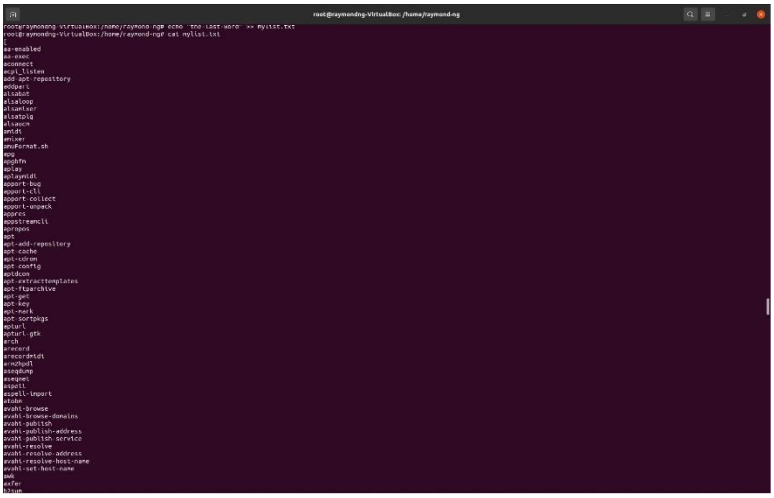
On this step I executed used the pipe symbol to execute `ls /usr/bin | less` command. I was able to navigate the text file using the '`Page Up`' and '`Page Down`' keys on my keyboard.

Pressing "`q`" on the keyboard exited out of the *less* screen.

Part VII: Redirection

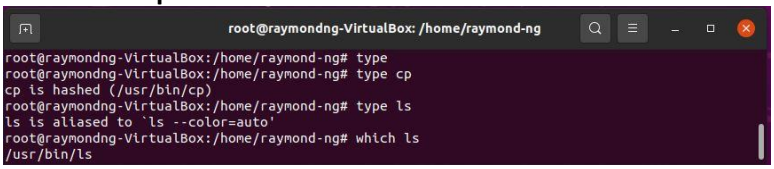


In Part VII, I learned how to use redirection to manage a lot of output from a command by using the "`>`" redirection symbol. In this screen I used `ls /usr/bin > mylist.txt` command to redirect the directory listing into a file named `mylist.txt`. Used the `ll` command to run list and verify.

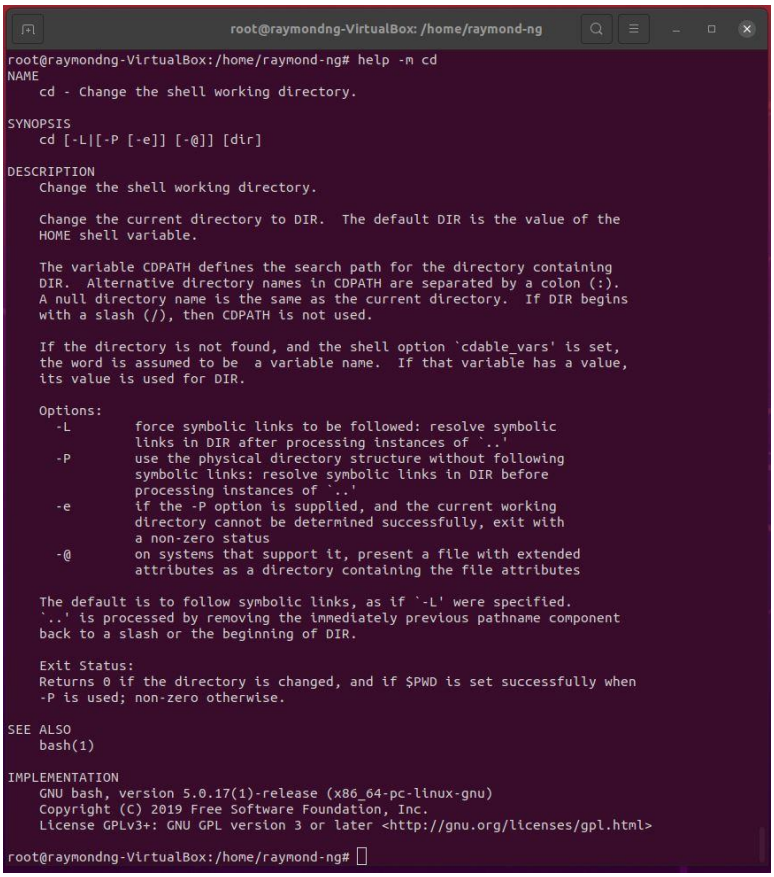


Per the instructions, I entered `echo "the-last-word" >> mylist.txt` command to append the `mylist.txt` file. Next used `cat mylist.txt` command to display the output, which listed the `/usr/bin` directory followed by the string I indicated.

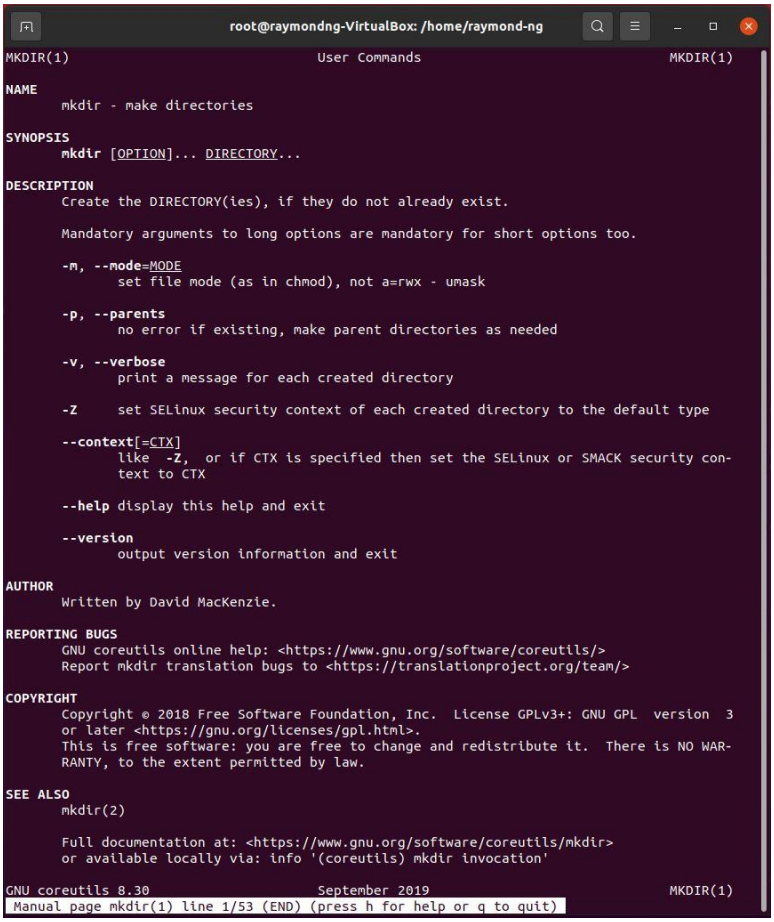
Part VIII: Help



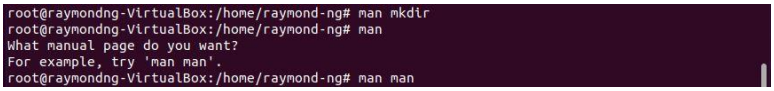
In this screen, I experimented with the *type* and *which* commands per the instructions provided. `type`, `type cp`, `type ls`, and `which ls` commands were used here.



On this screen I experimented with the `help` command. I typed in `help -m cd`, per the instructions.



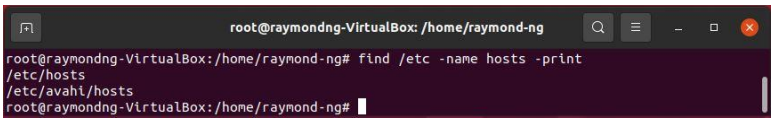
Following along with the prescribed instructions, I typed `man mkdir` command into the terminal to get more information regarding the `mkdir` command.



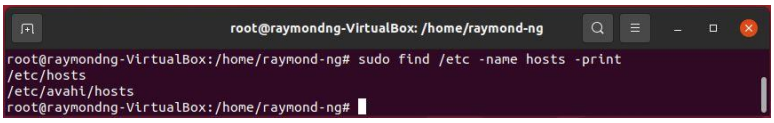
I wanted to get information about the `man` command itself so I typed into the terminal and the results yielded a question `What manual page do you want?` so I typed the example provided `man man`.

This screen appeared after the `man man` command was entered. It provided instructional guidance to the use of the *man* command.

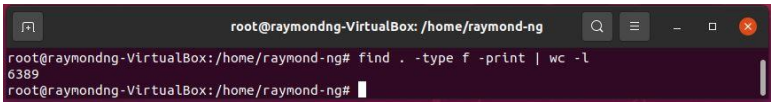
Part IX: Finding Files



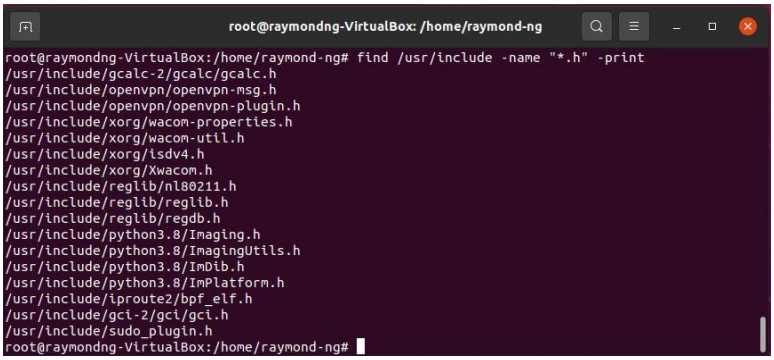
Here I used `find /etc -name hosts -print` command to locate a file with a known name.



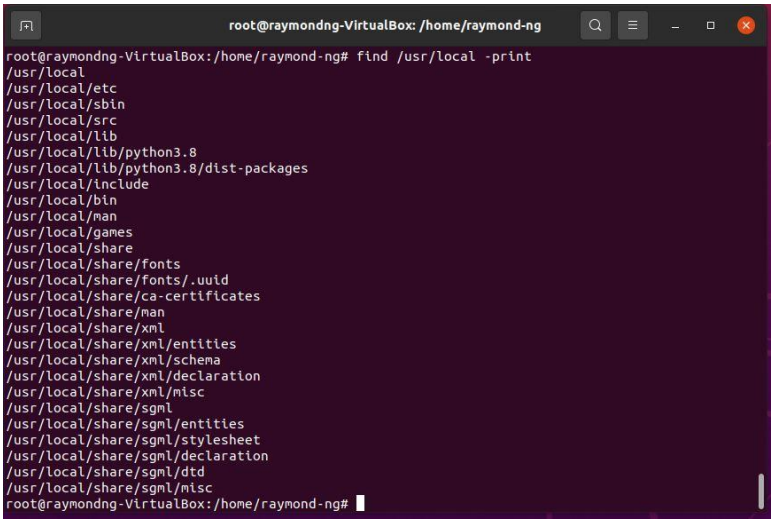
Executed the last `find` command with root privileges by using `sudo find /etc -name hosts -print` command.



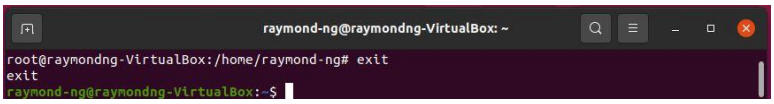
The command `find . -type f -print | wc -l` was executed to display the total number of files in the current working directory and all of its subdirectories.



The command `find /usr/include -name “*.h” -print` was executed to find all the files ending with “`.h`”.

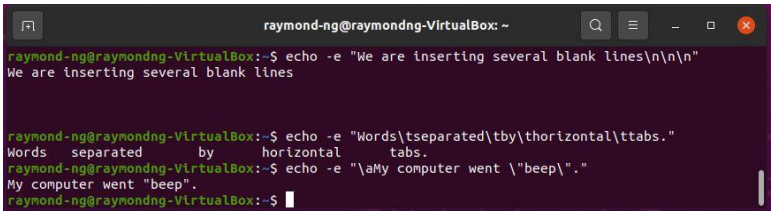


The command `find /usr/local -print` was executed to display all the files and directory names in the hierarchy starting with “`/usr/local`”.



The `exit` command was used to return the privilege of regular user.

Part X: Escape Sequences

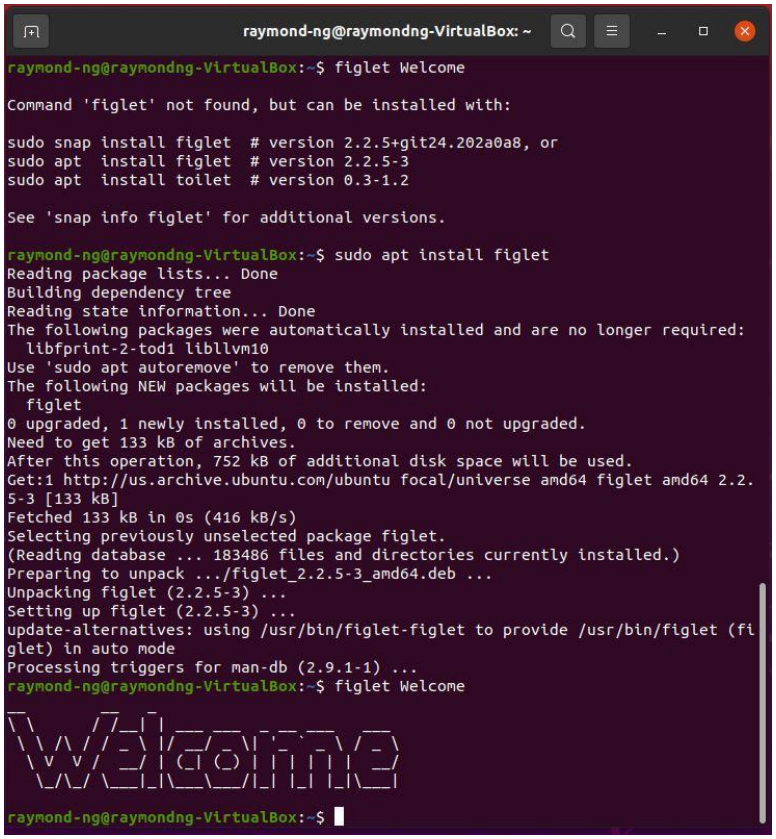


In Part XI I experimented with escape sequences, specifically the backlash escape characters: `\n` `\t` `\a`.

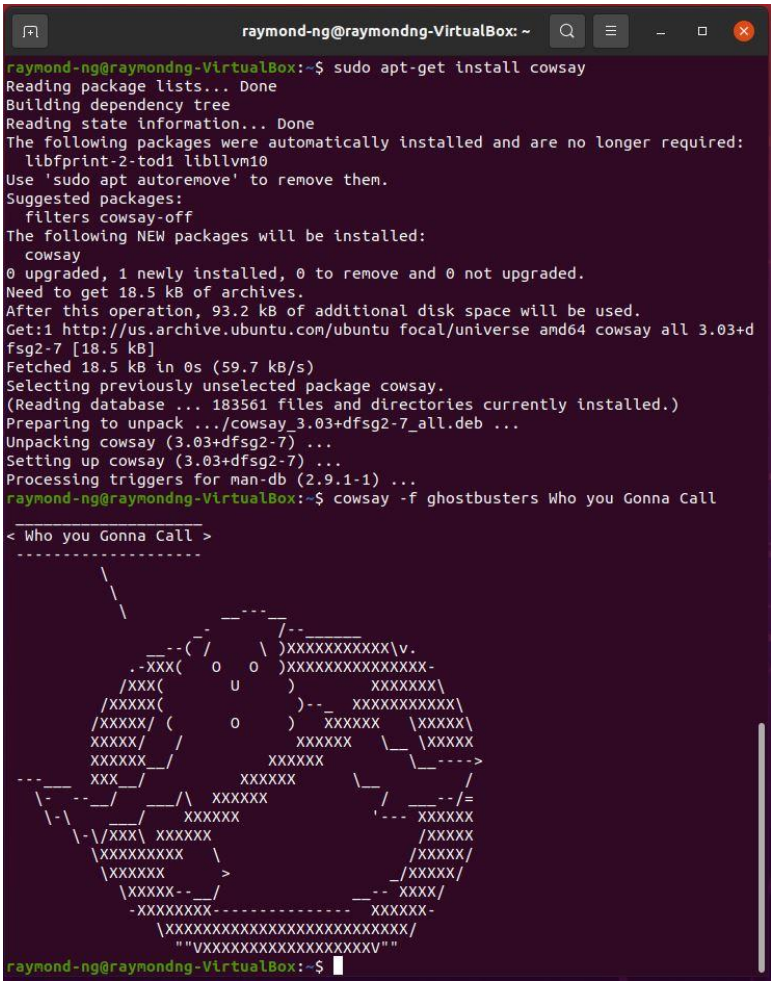
The following escape sequences were used in conjunctions with the echo command:

`Echo -e “We are inserting several blank lines\n\n\n`
`Echo -e “Words\tseparated\tby\thorizontal\ttabs.”`
`Echo -e “\aMy computer went \”beep\”.”`

Bonus: Command Line Art



For one of the extra credit options I attempted the figlet – draw banners option. First I had to install the `figlet` command first so I entered `sudo apt install figlet` command. Then entered `figlet Welcome` and it depicted a "*Welcome*" banner. (Moon, 2020)



I ghostbuster graphic looked appealing so I installed the `cowsay` command by entering `sudo apt-get install cowsay`. Next, I entered `cowsay -f ghostbusters Who you Gonna Call` into the terminal. (Moon, 2020)

CONCLUSION

In conclusion the lab was executed successfully with no issues. At no point did I encounter any error messages. I experimented and attempted to familiarize myself with every command-line argument, argument or option that was prescribed in the block of instructions. Additionally, the lab illuminated the important of learning to operate the CLI as it allows a greater deal of flexibility of use than the Graphic User Interface (GUI). One might argue the GUI does not have the same level of functionality and granular control as the CLI. A CLI can be used to easily do things that might be difficult or even impossible to do with a GUI, like interpreting problem sets in the cyber security domain. Further the simplicity of this lab allowed a novice user like myself to navigate the CLI a lot more confidently. At this

point I'm not really sure what I would like to explore more of specifically in this lab, but I'm sure that will change as we go through each module and perform future lab.

REFERENCES

Internet Resources

Moon, S. (2020, July 23). *20 Amusing Linux Commands to Haeve Fun with the Terminal*. Retrieved from BinaryTides.com: https://www.binarytides.com/linux-fun-commands/

Shotts, W. (2021). *Learning the Shell*. Retrieved from LinuxCommand.org: http://linuxcommand.org/lc3_learning_the_shell.php

Collaboration

No collaboration was involved in this lab. I did frequently tune into the chat on *Slack* to see if other students encountered issues that I should be on the lookout for; however, I completed this lab with none of the issues presented by the other students via *Slack*.