Lab 01 – Nix Commands

Raymond Ng: JQG999
IS 3033-CY1 – Summer 2022
June 12 2022

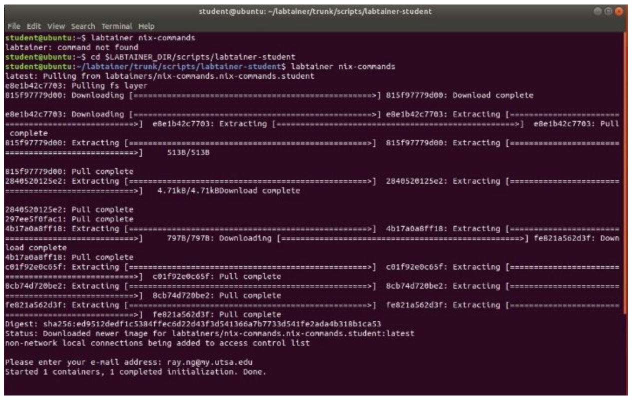
INTRODUCTION

The purpose of this lab is to allow basic users to become familiar with commonly used commands executed on Unix command-line. Moreover, it allows aspiring students majoring in computer science enclaves to safely experiment in a controlled environment via virtual machine.

PROCEDURE

Follow the setup and tasks outlined in this lab and detail your process. At a minimum, provide explanations and screenshots where requested in the lab instructions. You are encouraged to provide additional insights. Include relevant screenshots that *support your written explanations and observations.* In other words, any screenshots will enhance your narrative, not serve as stand-alone documentation.

Getting Started

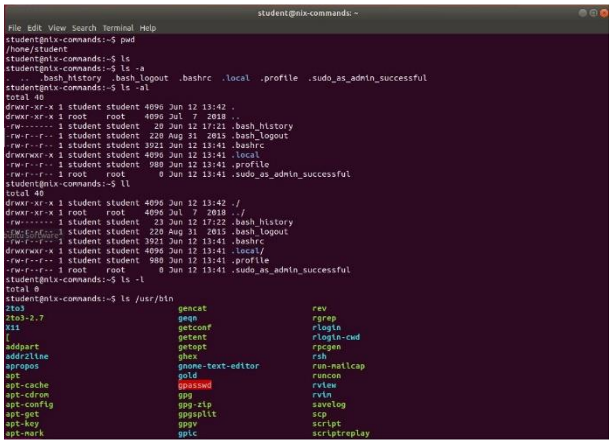


```
student@ubuntu:~$ labtainer nix-commands
labtainer: command not found
student@ubuntu:~$ cd $LABTAINER_DIR/scripts/labtainer-student
student@ubuntu:~/labtainer/trunk/scripts/labtainer-student$ labtainer nix-commands
latest: Pulling from labtainers/nix-commands.nix-commands.student
e8e1b42c7703: Pulling fs layer
815f97779d00: Downloading [==================================================>] 815f97779d00: Download complete

e8e1b42c7703: Downloading [==================================================>] e8e1b42c7703: Extracting [======================
============================>]  e8e1b42c7703: Extracting [==================================================>]  e8e1b42c7703: Pull
 complete
815f97779d00: Extracting [==================================================>]  815f97779d00: Extracting [======================
============================>]     513B/513B

815f97779d00: Pull complete
2840520125e2: Extracting [==================================================>]  2840520125e2: Extracting [======================
============================>]   4.71kB/4.71kBDownload complete

2840520125e2: Pull complete
297ee5f0fac1: Pull complete
4b17a0a8ff18: Extracting [==================================================>]  4b17a0a8ff18: Extracting [======================
============================>]     797B/797B: Downloading [==================================================>] fe821a562d3f: Down
load complete
4b17a0a8ff18: Pull complete
c01f92e0c65f: Extracting [==================================================>]  c01f92e0c65f: Extracting [======================
============================>]  c01f92e0c65f: Pull complete
8cb74d720be2: Extracting [==================================================>]  8cb74d720be2: Extracting [======================
============================>]  8cb74d720be2: Pull complete
fe821a562d3f: Extracting [==================================================>]  fe821a562d3f: Extracting [======================
============================>]  fe821a562d3f: Pull complete
Digest: sha256:ed9512dedf1c5384ffec6d22d43f3d541366a7b7733d541fe2ada4b318b1ca53
Status: Downloaded newer image for labtainers/nix-commands.nix-commands.student:latest
non-network local connections being added to access control list

Please enter your e-mail address: ray.ng@my.utsa.edu
Started 1 containers, 1 completed initialization. Done.
```

At first, when I was concerned that I did not execute the initial step properly because when I typed in `labtainer nix-commands` nothing was loading. I wondered if I had not downloaded the correct Labtainer for VM Ware. Approximately 7-10 seconds later, I saw the processes appear and by observing the multiple action/processes taking place—`pulling`, `extracting`—it made sense why I would observe a delay. Further, I knew I completed the step correctly when no errors appeared.

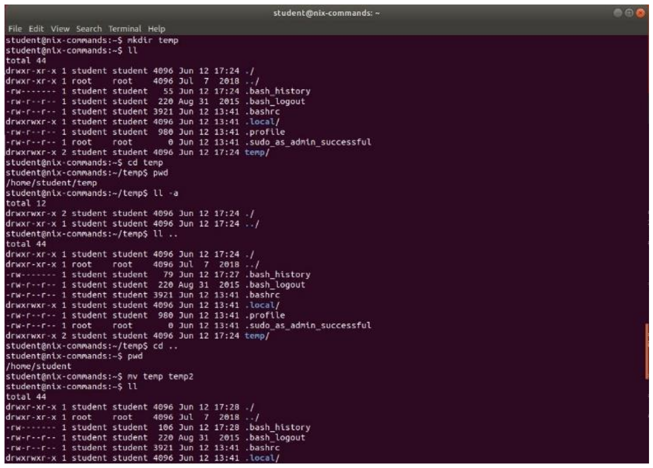
Basic Commands

I was glad that this lab began with an overview of basic commands in Unix like `pwd` (present working directory), `ls` (list), `ls -al` (list all), `ls -al` (list long), and `mk dir` (make directory) commands. I honestly had not used the command line since I took IS 1403. Good refresher. As you can see to right, all of the `ls` commands and the associated arguments listed all of the content found in the home directory (or the current directory I'm working in).



```
student@nix-commands:~$ pwd
/home/student
student@nix-commands:~$ ls
student@nix-commands:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .local  .profile  .sudo_as_admin_successful
student@nix-commands:~$ ls -al
total 40
drwxr-xr-x 1 student student 4096 Jun 12 13:42 .
drwxr-xr-x 1 root    root    4096 Jul  7  2018 ..
-rw------- 1 student student   20 Jun 12 17:21 .bash_history
-rw-r--r-- 1 student student  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 student student 3921 Jun 12 13:41 .bashrc
drwxrwxr-x 1 student student 4096 Jun 12 13:41 .local
-rw-r--r-- 1 student student  980 Jun 12 13:41 .profile
-rw-r--r-- 1 root    root       0 Jun 12 13:41 .sudo_as_admin_successful
student@nix-commands:~$ ll
total 40
drwxr-xr-x 1 student student 4096 Jun 12 13:42 ./
drwxr-xr-x 1 root    root    4096 Jul  7  2018 ../
-rw------- 1 student student   23 Jun 12 17:22 .bash_history
-rw-r--r-- 1 student student  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 student student 3921 Jun 12 13:41 .bashrc
drwxrwxr-x 1 student student 4096 Jun 12 13:41 .local/
-rw-r--r-- 1 student student  980 Jun 12 13:41 .profile
-rw-r--r-- 1 root    root       0 Jun 12 13:41 .sudo_as_admin_successful
student@nix-commands:~$ ls -l
total 0
student@nix-commands:~$ ls /usr/bin
2to3                                gencat                      rev
2to3-2.7                            geqn                        rgrep
X11                                 getconf                     rlogin
[                                   getent                      rlogin-cwd
addpart                             getopt                      rpcgen
addr2line                           ghex                        rsh
apropos                             gnome-text-editor           run-mailcap
apt                                 gold                        runcon
apt-cache                           gpasswd                     rview
apt-cdrom                           gpg                         rvim
apt-config                          gpg-zip                     savelog
apt-get                             gpgsplit                    scp
apt-key                             gpgv                        script
apt-mark                            gpic                        scriptreplay
```

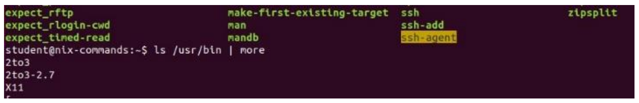
Here, we experimented with the `cd` (change directory), `mv` (move), and `cp` (copy) commands. I changed the current directory to *temp* and listed its contents. Then I listed the *dotdot* directory, which is a shortcut to the parent directory. Further, using the `cd` command, I changed the working directory to the parent directory of *temp*. Later, using mv, I moved (or renamed) *temp* to *temp2* and listed contents in the directory.



```
student@nix-commands:~$ mkdir temp
student@nix-commands:~$ ll
total 44
drwxr-xr-x 1 student student 4096 Jun 12 17:24 ./
drwxr-xr-x 1 root    root    4096 Jul  7  2018 ../
-rw------- 1 student student   55 Jun 12 17:24 .bash_history
-rw-r--r-- 1 student student  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 student student 3921 Jun 12 13:41 .bashrc
drwxrwxr-x 1 student student 4096 Jun 12 13:41 .local/
-rw-r--r-- 1 student student  980 Jun 12 13:41 .profile
-rw-r--r-- 1 root    root       0 Jun 12 13:41 .sudo_as_admin_successful
drwxrwxr-x 2 student student 4096 Jun 12 17:24 temp/
student@nix-commands:~$ cd temp
student@nix-commands:~/temp$ pwd
/home/student/temp
student@nix-commands:~/temp$ ll -a
total 12
drwxrwxr-x 2 student student 4096 Jun 12 17:24 ./
drwxr-xr-x 1 student student 4096 Jun 12 17:24 ../
student@nix-commands:~/temp$ ll ..
total 44
drwxr-xr-x 1 student student 4096 Jun 12 17:24 ./
drwxr-xr-x 1 root    root    4096 Jul  7  2018 ../
-rw------- 1 student student   79 Jun 12 17:27 .bash_history
-rw-r--r-- 1 student student  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 student student 3921 Jun 12 13:41 .bashrc
drwxrwxr-x 1 student student 4096 Jun 12 13:41 .local/
-rw-r--r-- 1 student student  980 Jun 12 13:41 .profile
-rw-r--r-- 1 root    root       0 Jun 12 13:41 .sudo_as_admin_successful
drwxrwxr-x 2 student student 4096 Jun 12 17:24 temp/
student@nix-commands:~/temp$ cd ..
student@nix-commands:~$ pwd
/home/student
student@nix-commands:~$ mv temp temp2
student@nix-commands:~$ ll
total 44
drwxr-xr-x 1 student student 4096 Jun 12 17:28 ./
drwxr-xr-x 1 root    root    4096 Jul  7  2018 ../
-rw------- 1 student student  106 Jun 12 17:28 .bash_history
-rw-r--r-- 1 student student  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 student student 3921 Jun 12 13:41 .bashrc
drwxrwxr-x 1 student student 4096 Jun 12 13:41 .local/
```

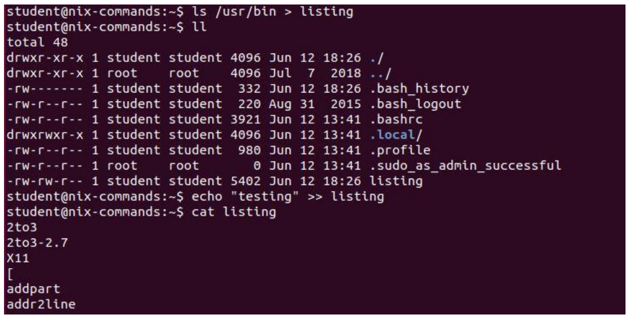
Pipes and Redirection

In this section I learned about the `|` (pipe) and `>` (redirection) commands.



```
expect_rftp                        make-first-existing-target  ssh               zipsplit
expect_rlogin-cwd                  man                         ssh-add
expect_timed-read                  mandb                       ssh-agent
student@nix-commands:~$ ls /usr/bin | more
2to3
2to3-2.7
X11
```

To the left, I'm using `|` to pipe `ls /usr/bin` to the `more` command which allows me to see the outputs one screen at a time.

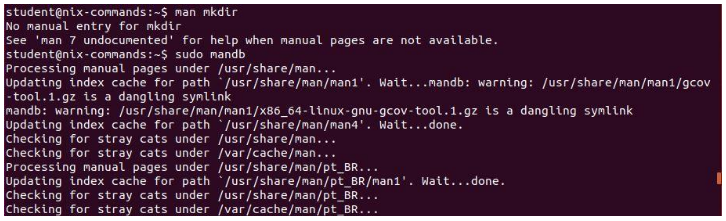


```
student@nix-commands:~$ ls /usr/bin > listing
student@nix-commands:~$ ll
total 48
drwxr-xr-x 1 student student 4096 Jun 12 18:26 ./
drwxr-xr-x 1 root    root    4096 Jul  7  2018 ../
-rw------- 1 student student  332 Jun 12 18:26 .bash_history
-rw-r--r-- 1 student student  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 student student 3921 Jun 12 13:41 .bashrc
drwxrwxr-x 1 student student 4096 Jun 12 13:41 .local/
-rw-r--r-- 1 student student  980 Jun 12 13:41 .profile
-rw-r--r-- 1 root    root       0 Jun 12 13:41 .sudo_as_admin_successful
-rw-rw-r-- 1 student student 5402 Jun 12 18:26 listing
student@nix-commands:~$ echo "testing" >> listing
student@nix-commands:~$ cat listing
2to3
2to3-2.7
X11
[
addpart
addr2line
```

Here, I'm using > to redirect the large output in `ls /usr/bin` into the file *listing*. Then I used `>>` to append the contents of *listing*. `cat listing` displayed all the contents of the text file *listing* and because I used `echo "testing"` to append to listing it displayed at the end of all the content within *listing*.

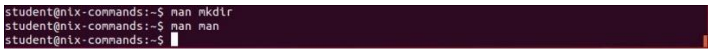
Help

Learning about the `man` (manual) command, I attempted to execute the `man mkdir` to get more information about the `mkdir` command. However, an error prompted. So, I did some research via *https://askubuntu.com/questions/927039/why-cant-i-find-any-manpages* and appears there are no manuals so I had to update the directory by using `sudo mandb`.



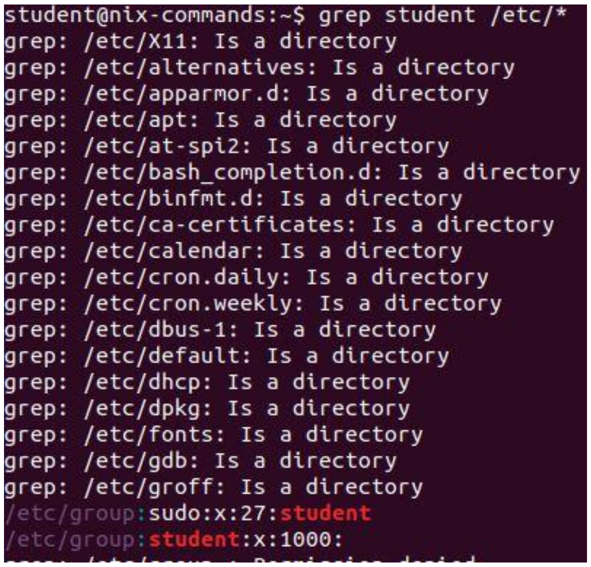
```
student@nix-commands:~$ man mkdir
No manual entry for mkdir
See 'man 7 undocumented' for help when manual pages are not available.
student@nix-commands:~$ sudo mandb
Processing manual pages under /usr/share/man...
Updating index cache for path `/usr/share/man/man1'. Wait...mandb: warning: /usr/share/man/man1/gcov
-tool.1.gz is a dangling symlink
mandb: warning: /usr/share/man/man1/x86_64-linux-gnu-gcov-tool.1.gz is a dangling symlink
Updating index cache for path `/usr/share/man/man4'. Wait...done.
Checking for stray cats under /usr/share/man...
Checking for stray cats under /var/cache/man...
Processing manual pages under /usr/share/man/pt_BR...
Updating index cache for path `/usr/share/man/pt_BR/man1'. Wait...done.
Checking for stray cats under /usr/share/man/pt_BR...
Checking for stray cats under /var/cache/man/pt_BR...
```

Update was successful. I was able to execute `man mkdir` and `man man` with no errors after the update.



```
student@nix-commands:~$ man mkdir
student@nix-commands:~$ man man
student@nix-commands:~$
```

Searching
In this section I experimented with search commands `grep` (global regular expression pattern) and `find`.



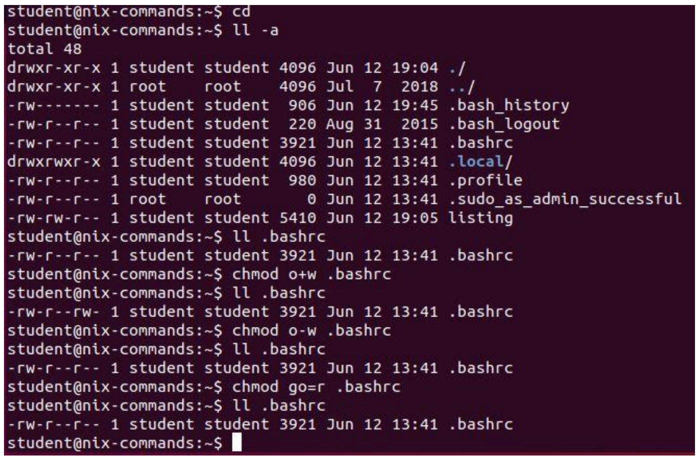
```
student@nix-commands:~$ grep student /etc/*
grep: /etc/X11: Is a directory
grep: /etc/alternatives: Is a directory
grep: /etc/apparmor.d: Is a directory
grep: /etc/apt: Is a directory
grep: /etc/at-spi2: Is a directory
grep: /etc/bash_completion.d: Is a directory
grep: /etc/binfmt.d: Is a directory
grep: /etc/ca-certificates: Is a directory
grep: /etc/calendar: Is a directory
grep: /etc/cron.daily: Is a directory
grep: /etc/cron.weekly: Is a directory
grep: /etc/dbus-1: Is a directory
grep: /etc/default: Is a directory
grep: /etc/dhcp: Is a directory
grep: /etc/dpkg: Is a directory
grep: /etc/fonts: Is a directory
grep: /etc/gdb: Is a directory
grep: /etc/groff: Is a directory
/etc/group:sudo:x:27:student
/etc/group:student:x:1000:
```

To the left, I executed `grep student /etc/*` where it search for the string "*student*" in all the files within the `/etc` directory.

In the same section, I learned to use `-s` to silence some of the errors reported from the output when executing `grep student /etc/*`. Further, using `sudo su` will allow user to gain root privilege to remedy permission problems when using the `find` command.

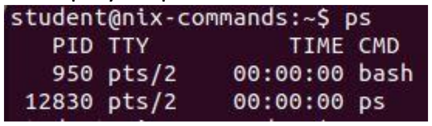
Access Control
In this section I learned about Discretionary Access Control (DAC), delineating owner-to-files and files-to-group.

To experiment, first I listed the contents in my home directory and observed the permissions on the far left side of each output as the relate to each object. First, we change the permissions of my *.bashrc* file so that anyone can write to it by using `chmod` (change mode) to `execute chmod o+w .bashrc`. `o+w` means add the *write* permission to *other*. Then I removed the permission, `chmod o-w .bashrc`. Executed multiple changes using `chmod ugo+rw .bashrc`. Lastly, I changed permissions so the group and other only have read access, `chmod go=r .bashrc`.



```
student@nix-commands:~$ cd
student@nix-commands:~$ ll -a
total 48
drwxr-xr-x 1 student student 4096 Jun 12 19:04 ./
drwxr-xr-x 1 root    root    4096 Jul  7  2018 ../
-rw------- 1 student student  906 Jun 12 19:45 .bash_history
-rw-r--r-- 1 student student  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 student student 3921 Jun 12 13:41 .bashrc
drwxrwxr-x 1 student student 4096 Jun 12 13:41 .local/
-rw-r--r-- 1 student student  980 Jun 12 13:41 .profile
-rw-r--r-- 1 root    root       0 Jun 12 13:41 .sudo_as_admin_successful
-rw-rw-r-- 1 student student 5410 Jun 12 19:05 listing
student@nix-commands:~$ ll .bashrc
-rw-r--r-- 1 student student 3921 Jun 12 13:41 .bashrc
student@nix-commands:~$ chmod o+w .bashrc
student@nix-commands:~$ ll .bashrc
-rw-r--rw- 1 student student 3921 Jun 12 13:41 .bashrc
student@nix-commands:~$ chmod o-w .bashrc
student@nix-commands:~$ ll .bashrc
-rw-r--r-- 1 student student 3921 Jun 12 13:41 .bashrc
student@nix-commands:~$ chmod go=r .bashrc
student@nix-commands:~$ ll .bashrc
-rw-r--r-- 1 student student 3921 Jun 12 13:41 .bashrc
student@nix-commands:~$ 
```

Process Management
Here, I learned about how to display process currently executing using `ps` (process status) and `ps ax` to display all process. Further learned to use `kill` to terminate a process.

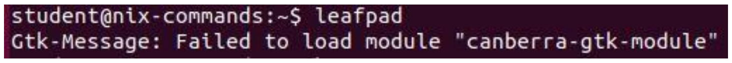


```
student@nix-commands:~$ ps
   PID TTY          TIME CMD
   950 pts/2    00:00:00 bash
 12830 pts/2    00:00:00 ps
```

To left is a screenshot of all the process that were executing at the time `ps` was executed.

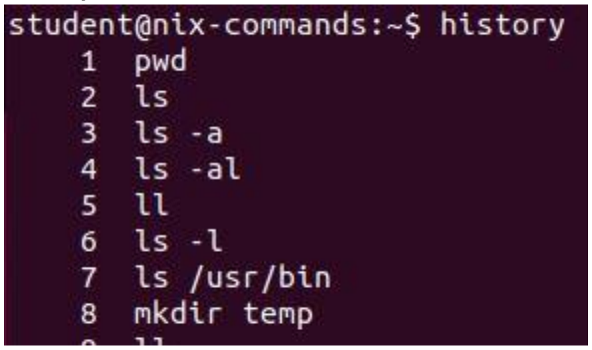
Editors

I encountered an issue with using `leafpad` editor. It's likely the module was not installed so I researched options to install via *https://askubuntu.com/questions/208431/failed-to-load-module-canberra-gtk-modul*e. I didn't want to install it because I wasn't sure if I needed it and I didn't want it taking up space. The strange part was that the editor did eventually load so I guess I didn't need to debug after all.



```
student@nix-commands:~$ leafpad
Gtk-Message: Failed to load module "canberra-gtk-module"
```

History

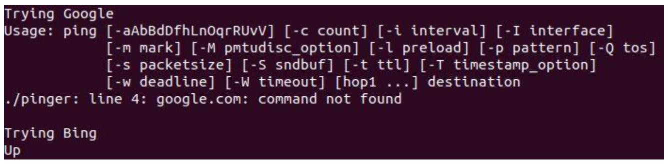


```
student@nix-commands:~$ history
    1  pwd
    2  ls
    3  ls -a
    4  ls -al
    5  ll
    6  ls -l
    7  ls /usr/bin
    8  mkdir temp
```

History command displayed all of the commands I have entered/executed as the student user during this lab.

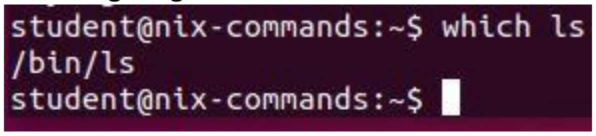
Shell Scripts

Here I experimented with the `ping` command. I'm glad the *leafpad* editor eventually worked so that I could write a script in the editor and save it in the home directory. To the right, you see that the output indicating the *pinger* file made from *leafpad* editor executed.



```
Trying Google
Usage: ping [-aAbBdDfhLnOqrRUvV] [-c count] [-i interval] [-I interface]
            [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
            [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
            [-w deadline] [-W timeout] [hop1 ...] destination
./pinger: line 4: google.com: command not found

Trying Bing
Up
```

Executing Programs



```
student@nix-commands:~$ which ls
/bin/ls
student@nix-commands:~$ 
```

Using the `which` command coupled with `ls`, `which ls`, I located the file that the shell will execute.

CONCLUSION

The goal of this lab was to experiment and become familiar with rudimentary Unix commands in controlled environment via virtual machine. Additionally, establish a general foundation for the student(s) to become more confident to execute succeeding labs.

As a self-identified novice user myself I thought this lab was extremely helpful and effective. It provided an overview, and refresher, to content taught in IS 1403 along with instructional guidance to safely—and confidently—execute commands on the command-line in a experimental domain, VM Ware.

In terms of difficulty, I thought it was moderately easy. However, there were a few instances where I had to conduct my own research to get a command to work (i.e. `man`). I'm sure we will be exploring other aspects of this topic as labs become more advanced during the progression of this semester.

References

Internet Resources

https://askubuntu.com/questions/208431/failed-to-load-module-canberra-gtk-module

https://askubuntu.com/questions/208431/failed-to-load-module-canberra-gtk-module

Collaboration

I did not collaborate with any of my peers (other students), nor did I seek expert guidance beyond what I could find on the internet to complete this lab.