Lab3 Brandon Fowler

1) Clearly explain the difference between which, whereis, grep, and find.

Which finds a file in your current path. Whereis is useful for finding binary files and manuals. Find will find any file on the system by searching recursivley, starting with a user specified starting directory. Grep searches an input file for patterns of regular expressions.

2) Issue the find command looking for the file named ld starting at the root directory.
a. Assuming you are not logged in as root, you should get a list of errors as well as where the file was found. Capture the output and include it in your submission – you do not need to include all the permission errors just a few to get the idea but do include where the file was found.
b. Repeat the command (again not as root) – illustrating a method of eliminating the error messages and printing only what was found.

```
brandonf@cslinux:~$ find / -name ld
find: `/proc/27803/fd': Permission denied
find: `/proc/27803/fdinfo': Permission denied
find: `/proc/27803/ns': Permission denied
find: `/proc/28017/task/28017/fd': Permission denied
find: `/proc/28017/task/28017/fdinfo': Permission denied
find: `/proc/28017/task/28017/ns': Permission denied
find: `/proc/28017/fd': Permission denied
find: `/proc/28017/fdinfo': Permission denied
find: `/proc/28017/ns': Permission denied
find: `/proc/28018/task/28018/fd': Permission denied
find: `/proc/28018/task/28018/fdinfo': Permission denied
find: `/proc/28018/task/28018/ns': Permission denied
find: `/proc/28018/fd': Permission denied
find: `/proc/28018/fdinfo': Permission denied
find: `/proc/28018/ns': Permission denied
find: `/opt/quest/lib/x86_64-linux-gnu': Permission denied
find: `/opt/quest/emul/ia32-linux/lib': Permission denied
find: `/opt/quest/libexec/vgp/xlators/lib': Permission denied
find: `/opt/quest/libexec/vgp/xlators/User': Permission denied
find: `/var/lib/sudo': Permission denied
find: `/var/lib/samba/usershares': Permission denied
find: `/var/log/samba': Permission denied
find: `/var/opt/quest/vas/authcache': Permission denied
find: `/var/opt/quest/vgp/gpt': Permission denied
find: `/var/opt/quest/vgp/exts': Permission denied
find: `/var/opt/quest/vgp/cache_user': Permission denied
find: `/var/opt/quest/vgp/cache': Permission denied
find: `/var/cache/ldconfig': Permission denied
find: `/var/spool/cron/atjobs': Permission denied
find: `/var/spool/cron/atspool': Permission denied
find: `/var/spool/cron/crontabs': Permission denied
/usr/lib/compat-ld/ld
/usr/lib/gold-ld/ld
/usr/bin/ld
/usr/share/doc/binutils/ld

brandonf@cslinux:~$ find / -name ld 2</dev/null
```

```
/usr/lib/compat-ld/ld
/usr/lib/gold-ld/ld
/usr/bin/ld
/usr/share/doc/binutils/ld
```

3) In class we talked about the '-name' option for the find command.
a. Explain how to use the size option.
b. Issue and capture the results of the find command in your home directory that display all files that are greater than 1K. Do not search for more than 3 subfolders. Do not display error messages.

Use the find command and use -size instead of -name, then type in a size to look for. The size options finds files with the size specified in amount of blocks.

```
brandonf@cslinux:~$ find ~ -size +1 2</dev/null
/home/EASTERN/brandonf
/home/EASTERN/brandonf/.bashrc
/home/EASTERN/brandonf/.profile
/home/EASTERN/brandonf/.cache
```

4) Use a text editor on the remote machine to create a file named frost.poem that contains the following text:

a. Use the grep command, capture both the command and the output, to finds all lines, including the line number, that end with a comma.
b. Use the grep command, capture both the command and the output, to finds all lines, including the line number, containing the word as.
c. Use the grep command, capture both the command and the output, to finds all lines, including the line number that starts with the word and (case DOES NOT matter).
d. Use the grep command, capture both the command and the output, to finds all lines, including the line number that starts with the word and (case DOES matter).

```
brandonf@cslinux:~$ grep -n , frost.poem
2:Two roads diverged in a yellow wood,
4:and be one traveler, long I stood
7:Then took the other, as just as fair,
9:Because it was grassy and wanted wear,
11:Had worn them really about the same,

5:And looked down one as far as I could
7:Then took the other, as just as fair,
9:Because it was grassy and wanted wear,
10:Though as for that the passing there

brandonf@cslinux:~$ grep -i -n and frost.poem
3:And sorry I could not travel both
4:and be one traveler, long I stood
5:And looked down one as far as I could
8:And having perhaps the better claim
9:Because it was grassy and wanted wear,

brandonf@cslinux:~$ grep -i -n and frost.poem
3:And sorry I could not travel both
```

```
4:and be one traveler, long I stood
5:And looked down one as far as I could
8:And having perhaps the better claim
9:Because it was grassy and wanted wear,
```

5) Capture, creating a directory named lab3.
a. Capture placing a copy of frost.poem in the directory lab3. There should be one copy of frost.poem in your home directory and one in lab3.
b. Within your home directory, capture the grep command and its output that will recursively find all instances of the word I (case DOES matter) in all files that end with .poem.

```
brandonf@cslinux:~$ mkdir lab3
brandonf@cslinux:~$ cp frost.poem lab3

brandonf@cslinux:~$ grep -i -r i *.poem
Two roads diverged in a yellow wood,
And sorry I could not travel both
and be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;
Then took the other, as just as fair,
And having perhaps the better claim
Because it was grassy and wanted wear,
Though as for that the passing there
brandonf@cslinux:~$ grep -r -i i *.poem
Two roads diverged in a yellow wood,
And sorry I could not travel both
and be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;
Then took the other, as just as fair,
And having perhaps the better claim
Because it was grassy and wanted wear,
Though as for that the passing there
```

6) Execute man source.
a. Did the manual page display?
b. What command do you need to issue to get source to display?
c. Capture the results of that command.
d. Where could have you used this in relation to lab 1?

```
brandonf@cslinux:~$ man source
No manual entry for source
No the manual page did not display.
```

I needed to use the help command to get source to display.

```
brandonf@cslinux:~$ help source
source: source filename [arguments]
    Execute commands from a file in the current shell.
      Read and execute commands from FILENAME in the current shell.  The entries
in $PATH are used to find the directory containing FILENAME.If any ARGUMENTS are
supplied, they become the positional parameters when FILENAME is executed.
      Exit Status:
    Returns the status of the last command executed in FILENAME; fails if
FILENAME cannot be read.
```

I no longer have the write up for lab1, and the assignmnet is locked so I can't

download it. I do not know how this relates to lab1.

7) Using a text editor create a file named myScript that contains the following:

./
a. Execute the script with ./myScript and capture the output.
b. Execute and capture the command that will change the permissions on myScript to be user executable without changing any other permissions.
c. Execute the script with ./myScript and capture the output.

```
brandonf@cslinux:~$ ./myScript
-bash: ./myScript: Permission denied

brandonf@cslinux:~$ chmod 744 myScript

brandonf@cslinux:~$ ./myScript
/bin/bash: string=Hello World echo $string : No such file or directory
```

8) Using a text editor create a file named secondScript that contains the following:
#!/bin/ksh string=Hello World print $string
a. Execute the script with ./secondScript and capture the output.
b. Execute and capture the command that will change the permissions on secondScript to be user executable without changing any other permissions.
c. Execute the script with ./secondScript and capture the output.
d. What does the #! mean?
e. In problem 7 what shell did the code execute in?
f. In problem 8 what shell did the code execute in?

```
brandonf@cslinux:~$ ./secondScript
-bash: ./secondScript: Permission denied

brandonf@cslinux:~$ chmod 744 secondScript

brandonf@cslinux:~$ ./secondScript
/bin/ksh: line 1: World: not found
```

#! is the start of a script.

In problem 7 the code executed in the born again shell

In problem 8 the code executed in the korn shell

9) Using the man page for env
a. Describe (in your own words not with captures from the man page) the output of env command with no arguments.
b. Capture a command other than pwd that will show your current working directory.
c. Describe what you would have to do to make this change permanent for all future sessions.

The env command with no arguments will simply display information on the current environment.

```
brandonf@cslinux:~$ echo ~
```

/home/EASTERN/brandonf

Modify the bashrc file

10) What is the difference between a shell variable and an environment variable in the bash shell?

Shell variables store temporary information that usually changes fruequently, and reset each new session, whereas environment variables are more permanent, and are cosistent in different sessions.

11) Define what a process is and what a job is, clearly explain how jobs differ from processes.

A process is simple a program that is running. A job is a type of process that is constrained to a shell.

12) Give the grep command that will start in your home directory and show the file names and line numbers containing the term "stdio" in all .c files in the home directory and all directories below the home.

```
brandonf@cslinux:~$ grep -r -n -H stdio *.c
grep: *.c: No such file or directory
```

13) Consider the following command ls " al | more.

a. How many processes are created with that command?
b. What exactly does "|" do in this command?

Three processes are created with this command.

In this command the | sends the output from ls -al to the more command.

14) Using the man page for ps
a. Issue and capture the ps command with the appropriate options to allow listing of all processes in the system.
b. Using the output from part A, what was the first process started and by whom was it started?
c. What was the first non-root process that was started?
d. What was the last process started and by whom?

```
brandonf@cslinux:~$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1  29532  2580 ?        Ss   Mar25   0:02 /sbin/init
root         2  0.0  0.0      0     0 ?        S    Mar25   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    Mar25   0:26 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S    Mar25   0:00 [kworker/u:0]
root         6  0.0  0.0      0     0 ?        S    Mar25   0:00 [migration/0]
root         7  0.0  0.0      0     0 ?        S    Mar25   0:05 [watchdog/0]
root         8  0.0  0.0      0     0 ?        S    Mar25   0:02 [migration/1]
root        10  0.0  0.0      0     0 ?        S    Mar25   0:08 [ksoftirqd/1]
root        12  0.0  0.0      0     0 ?        S    Mar25   0:06 [watchdog/1]
```

```
root        13  0.0  0.0      0     0 ?        S<   Mar25   0:00 [cpuset]
root        14  0.0  0.0      0     0 ?        S<   Mar25   0:00 [khelper]
root        15  0.0  0.0      0     0 ?        S    Mar25   0:00 [kdevtmpfs]
root        16  0.0  0.0      0     0 ?        S<   Mar25   0:00 [netns]
root        17  0.0  0.0      0     0 ?        S    Mar25   0:21 [kworker/u:1]
root        18  0.0  0.0      0     0 ?        S    Mar25   0:33 [sync_supers]
root        19  0.0  0.0      0     0 ?        S    Mar25   0:00 [bdi-default]
root        20  0.0  0.0      0     0 ?        S<   Mar25   0:00 [kintegrityd]
root        21  0.0  0.0      0     0 ?        S<   Mar25   0:00 [kblockd]
root        22  0.0  0.0      0     0 ?        S<   Mar25   0:00 [ata_sff]
root        23  0.0  0.0      0     0 ?        S    Mar25   0:00 [khubd]
root        24  0.0  0.0      0     0 ?        S<   Mar25   0:00 [md]
root        25  0.0  0.0      0     0 ?        S    Mar25   0:00 [khungtaskd]
root        26  0.0  0.0      0     0 ?        S    Mar25   0:02 [kswapd0]
root        27  0.0  0.0      0     0 ?        SN   Mar25   0:00 [ksmd]
root        28  0.0  0.0      0     0 ?        SN   Mar25   0:00 [khugepaged]
root        29  0.0  0.0      0     0 ?        S    Mar25   0:00 [fsnotify_mark]
root        30  0.0  0.0      0     0 ?        S    Mar25   0:00 [ecryptfs-kthr]
root        31  0.0  0.0      0     0 ?        S<   Mar25   0:00 [crypto]
root        39  0.0  0.0      0     0 ?        S<   Mar25   0:00 [kthrotld]
root        40  0.0  0.0      0     0 ?        S    Mar25   0:00 [scsi_eh_0]
root        41  0.0  0.0      0     0 ?        S    Mar25   0:00 [scsi_eh_1]
root        62  0.0  0.0      0     0 ?        S<   Mar25   0:00 [devfreq_wq]
root       165  0.0  0.0      0     0 ?        S<   Mar25   0:00 [mpt_poll_0]
root       166  0.0  0.0      0     0 ?        S<   Mar25   0:00 [mpt/0]
root       210  0.0  0.0      0     0 ?        S    Mar25   0:00 [scsi_eh_2]
root       225  0.0  0.0      0     0 ?        S    Mar25   0:00 [jbd2/sda1-8]
root       226  0.0  0.0      0     0 ?        S<   Mar25   0:00 [ext4-dio-unwr]
root       296  0.0  0.0      0     0 ?        S    Mar25   0:11 [flush-8:0]
root       307  0.0  0.0  17232   636 ?        S    Mar25   0:00 upstart-udev-br
root       309  0.0  0.0  15436  1448 ?        Ss   Mar25   0:00 /sbin/udevd --d
root       434  0.0  0.0  15432  1116 ?        S    Mar25   0:00 /sbin/udevd --d
root       435  0.0  0.0  15432  1112 ?        S    Mar25   0:00 /sbin/udevd --d
root       458  0.0  0.0      0     0 ?        S<   Mar25   0:00 [kpsmoused]
root       459  0.0  0.0      0     0 ?        S<   Mar25   0:00 [ttm_swap]
root       488  0.0  0.0  15188   376 ?        S    Mar25   0:00 upstart-socket-
root       518  0.0  0.0  19200   984 ?        Ss   Mar25   0:01 rpcbind -w
root       676  0.0  0.0      0     0 ?        S<   Mar25   0:00 [reiserfs]
root       701  0.0  0.0      0     0 ?        S<   Mar25   0:00 [rpciod]
root       716  0.0  0.1  45776  2796 ?        Ss   Mar25   0:00 /usr/sbin/sshd
root       718  0.0  0.0      0     0 ?        S<   Mar25   0:00 [nfsiod]
root       727  0.0  0.2 127404  4268 ?        Ss   Mar25   0:04 smbd -F
syslog     744  0.0  0.0 243132  1476 ?        Sl   Mar25   0:28 rsyslogd -c5
statd      745  0.0  0.0  21504  1272 ?        Ss   Mar25   0:00 rpc.statd -L
root       746  0.0  0.0  19208   580 ?        Ss   Mar25   0:00 rpc.idmapd
102        751  0.0  0.0  26924  1272 ?        Ss   Mar25   0:00 dbus-daemon --s
avahi      771  0.0  0.0  35360  2032 ?        S    Mar25   0:04 avahi-daemon: r
avahi      772  0.0  0.0  35200   540 ?        S    Mar25   0:00 avahi-daemon: c
root       795  0.0  0.1  97576  2380 ?        Ss   Mar25   1:07 nmbd -D
root       824  0.0  0.0   9452   832 tty4     Ss+  Mar25   0:00 /sbin/getty -8
root       830  0.0  0.0   9452   832 tty5     Ss+  Mar25   0:00 /sbin/getty -8
root       846  0.0  0.0   9452   836 tty2     Ss+  Mar25   0:00 /sbin/getty -8
root       847  0.0  0.0   9452   836 tty3     Ss+  Mar25   0:00 /sbin/getty -8
root       849  0.0  0.0   9452   832 tty6     Ss+  Mar25   0:00 /sbin/getty -8
root       854  0.0  0.0   4328   648 ?        Ss   Mar25   0:00 acpid -c /etc/a
root       857  0.0  0.0  12780   920 ?        Ss   Mar25   0:01 cron
root       861  0.0  0.0 442420  1980 ?        Ssl  Mar25   9:54 /usr/sbin/autom
daemon     866  0.0  0.0  10576   344 ?        Ss   Mar25   0:00 atd
root       867  0.0  0.0  15980   664 ?        Ss   Mar25   2:06 /usr/sbin/irqba
root       868  0.0  0.1 111388  3780 ?        Ss   Mar25   0:00 /usr/sbin/winbi
whoopsie   888  0.0  0.1 183320  3516 ?        Ssl  Mar25   0:04 whoopsie
root       907  0.0  0.1 111592  3612 ?        S    Mar25   0:01 /usr/sbin/winbi
```

```
root       912  0.0  0.0 104848  1812 ?        S    Mar25   0:00 /usr/sbin/winbi
root       913  0.0  0.1 111328  3156 ?        S    Mar25   0:00 /usr/sbin/winbi
root       914  0.0  0.0 104848  1716 ?        S    Mar25   0:00 /usr/sbin/winbi
root       917  0.0  0.4  43020  9472 ?        Ss   Mar25   3:58 /opt/quest/sbin
daemon     925  0.0  1.8  68520 37328 ?        S    Mar25   1:44 /opt/quest/sbin
root       928  0.0  0.0 127508  1772 ?        S    Mar25   0:00 smbd -F
daemon     953  0.0  1.2  57380 26148 ?        S    Mar25   7:16 /opt/quest/sbin
daemon     954  0.0  0.9  52608 19764 ?        S    Mar25  12:19 /opt/quest/sbin
daemon     955  0.0  0.9  52608 19680 ?        S    Mar25   0:11 /opt/quest/sbin
root      1006  0.0  0.0   9452   828 tty1     Ss+  Mar25   0:00 /sbin/getty -8
root      2308  0.0  0.0      0     0 ?        S    22:25   0:00 [kworker/1:2]
root      2663  0.0  0.0      0     0 ?        S    22:30   0:00 [kworker/1:0]
jhutton5  2977  0.0  0.0   9384   920 pts/14   T    22:34   0:00 grep --color=au
root      5136  0.0  0.0      0     0 ?        S    22:59   0:00 [flush-0:21]
root      5507  0.0  0.0      0     0 ?        S    23:04   0:00 [kworker/0:1]
root      6059  0.0  0.2  95424  5260 ?        SLs  23:12   0:00 sshd: brandonf
brandonf  6107  0.0  0.1  95424  2436 ?        S    23:12   0:00 sshd: brandonf@
brandonf  6108  0.8  0.4  31080  9584 pts/2    Ss   23:12   0:00 -bash
brandonf  6263  0.0  0.0  23244  1568 pts/2    R+   23:13   0:00 ps aux
root      6605  0.0  0.0      0     0 ?        S    Apr08   0:28 [cifsd]
root     10680  0.2  0.0      0     0 ?        S    18:28   0:36 [kworker/1:1]
root     13618  0.0  0.0      0     0 ?        S<   06:41   0:00 [xfs_mru_cache]
root     13619  0.0  0.0      0     0 ?        S<   06:41   0:00 [xfslogd]
root     13620  0.0  0.0      0     0 ?        S<   06:41   0:00 [xfsdatad]
root     13621  0.0  0.0      0     0 ?        S<   06:41   0:00 [xfsconvertd]
root     13624  0.0  0.0      0     0 ?        S    06:41   0:00 [jfsIO]
root     13625  0.0  0.0      0     0 ?        S    06:41   0:00 [jfsCommit]
root     13626  0.0  0.0      0     0 ?        S    06:41   0:00 [jfsCommit]
root     13627  0.0  0.0      0     0 ?        S    06:41   0:00 [jfsSync]
root     19138  0.0  0.2  93360  5228 ?        SLs  20:06   0:00 sshd: twalker [
twalker  19167  0.0  0.1  93360  2432 ?        S    20:06   0:00 sshd: twalker@p
twalker  19168  0.0  0.4  31076  9588 pts/1    Ss+  20:06   0:00 -bash
root     23156  0.0  0.2  95424  5264 ?        SLs  20:39   0:00 sshd: jhutton5
jhutton5 23207  0.4  0.2  97444  4440 ?        S    20:39   0:37 sshd: jhutton5@
jhutton5 23208  0.0  0.4  31080  9620 pts/14   Ss+  20:39   0:00 -bash
twalker  24043  0.0  0.0   9384   928 pts/1    T    20:42   0:00 grep --color=au
twalker  24044  0.0  0.0   9384   936 pts/1    T    20:42   0:00 grep --color=au
root     26045  0.0  0.2  95424  5264 ?        SLs  20:57   0:00 sshd: jaimew [p
jaimew   26105  0.0  0.1  95424  2428 ?        S    20:57   0:00 sshd: jaimew@pt
jaimew   26106  0.0  0.4  31156  9716 pts/0    Ss+  20:57   0:01 -bash
root     28672  0.0  0.2  95420  5252 ?        SLs  21:25   0:00 sshd: bdaschel
bdaschel 28707  0.0  0.1  95420  2296 ?        S    21:25   0:00 sshd: bdaschel@
bdaschel 28708  0.0  0.0  17992  1280 ?        Ss   21:25   0:00 /usr/lib/openss
root     32281  0.0  0.0      0     0 ?        S    21:58   0:01 [kworker/0:2]
```

The first process started was init, started by the shell upon creating the current session.

rsyslogd -c5 was the first not root program started, it was started by syslog.

ps aux was the last process, and it was started by me.

15) In a single terminal window capture the command to start firefox, gedit, gcalctool and gnomine as background jobs:
a. What are the job numbers of the above?
b. What are the process ID numbers of the above?
c. Capture the command and output to bring gedit to the foreground.
d. Capture the command(s) to send gedit to the background.
e. Capture the command(s) to kill gcalctoolone using its job number.
f. Capture the command(s) to kill gnomine using its process number.
g. Can CTRL C be used to kill any job? Why or why not?

```
huntersike@ubuntu:~$ gnomine &
[1] 3684
huntersike@ubuntu:~$ gcalctool &
[2] 3704
huntersike@ubuntu:~$ firefox &
[3] 3708
huntersike@ubuntu:~$ gedit &
[4] 3760
```

Process IDs are gnomine: 3684, gcalctool: 3704, firefox: 3708, gedit: 3760

```
huntersike@ubuntu:~$ fg %4

huntersike@ubuntu:~$ bg %4

huntersike@ubuntu:~$ kill %2

huntersike@ubuntu:~$ kill 3684
```

No CTRL C cannot be used to kill any jobs. Some jobs are protected by the system or can ignore kill commands.

16) Using a text editor create a file named hello.c that contains:

```
#include <stdio.h>
int main()
{
printf("Hello\n");
return 0;
}// end main
```

a. Capture the command that compiles hello.c and leaves the executable program in a file named
hello.
b. Execute hello and capture its output.
c. Copy the contents of hello.c into a file named hello2 (no .c suffix)
d. Try to compile hello2 into a file named hello_2 and capture the output.
e. Explain the output from Part D.

```
huntersike@ubuntu:~$ gcc hello.c -o hello.c

huntersike@ubuntu:~$ ./hello.c
Hello

huntersike@ubuntu:~$ gcc hello2 -o hello_2
hello2: file not recognized: File format not recognized
collect2: error: ld returned 1 exit status
```

The gcc command was unable to compile hello2 because it is not a .c file and is therefore not recognized by the compiler.

17) Using a text editor create the following C programmed named almostEndless.c

```
#include <stdio.h>
int main()
{
int x = 0;
while(x < 20000000)
{
printf("..");
fflush(stdout);
sleep(3);
```

```
}// end while
return 0;
}// end main
```

a. Compile your program with gcc almostEndless.c
b. Start your program with ./a.out
c. With your program running, describe the commands you would use (without using ctrl-c) to
terminate that program, from the same terminal window in which it was started
d. Execute and capture the commands, using process notation, to terminate a.out
e. Restart your program with ./a.out
f. Execute and capture the commands, using job notation, to terminate a.out

To kill this program with out CTRL C I would use CTRL Z to stop it, then kill it with the kill command.

```
[4]+  Stopped                 ./a.out
huntersike@ubuntu:~$ ps
  PID TTY          TIME CMD
 3583 pts/1    00:00:00 bash
 3708 pts/1    00:00:05 firefox
 4070 pts/1    00:00:00 a.out
 4071 pts/1    00:00:00 ps
huntersike@ubuntu:~$ kill -9 4070

[4]+  Stopped                 ./a.out
huntersike@ubuntu:~$ kill %4
[4]+  Terminated
```