Lab 2 Brandon Fowler

1. Clearly explain why programs should be placed in /bin or /usr/bin.

A program placed in /bin can be used by all users on the system. If a user wants a program installer just for that user it should be placed in /usr/bin

2. You are asked to use a program named mystery which you have never used before. Explain how
you would find information on the program and what it does. List all the ways (Google/Bing is
not an option)

I would use either the info command fowllowed by the program name or the whatis command collowed by the program name.

3. There are many other environment variables available to the user. One such variable is $HOME.In your current shell capture your current $HOME. Now change HOME to be /bin. Capture the
command. Issue a cd and pwd to illustrate the environment variable has changed. Close your
terminal and reopen it.

```
huntersike@ubuntu:~$ echo $HOME
/home/huntersike
huntersike@ubuntu:~$ HOME=/bin
huntersike@ubuntu:/home/huntersike$ cd ~
huntersike@ubuntu:~$ pwd
/bin
```

4. In class we discussed the use of the accent (back tick) for when it comes to executing the date command. Can you use the accent on the ls command? How would you use the accent on the ls
command? Capture the usage of the accent on the ls command.

Yes but you would have to use it within double quotes just like the date command.

```
huntersike@ubuntu:~$ "Files in directory are: `ls`"
Files in directory are: almost
cscd240
Desktop
Documents
Downloads
examples.desktop
lab1
Music
myTest
orig.lab1
Pictures
Public
Templates
```

5. Capture the output of the file command on the chmod executable. (Where does chmod live?)
Explain the information being displayed.

```
huntersike@ubuntu:~$ file /bin/chmod
```

```
/bin/chmod: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux 2.6.24,
BuildID[sha1]=0x00380ea140da52dede122ac46dc934d984456959, stripped
```

chmod lives in the bin directory. This information explains the type of executable that chmod is listing information such as version, OS, build, etc.

6. Capture the output of the stat command on the chmod executable. Explain the information being displayed.

```
huntersike@ubuntu:~$ stat /bin/chmod
  File: `/bin/chmod'
  Size: 51792           Blocks: 104        IO Block: 1024   regular file
Device: 700h/1792d      Inode: 62574       Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2013-04-06 12:11:30.000000000 -0700
Modify: 2012-10-01 09:07:45.000000000 -0700
Change: 2012-10-17 02:32:50.000000000 -0700
 Birth: -
```

That stat command displays information on chmod such as, size, memory used, access permissions, and when it was last changes, modified, and accessed.

7. Try and delete chmod. Did it delete why or why not?

```
huntersike@ubuntu:~$ rm /bin/chmod
rm: remove write-protected regular file `/bin/chmod'?
```

No it did not delete chmod, because chmod is a protected file.

8. Try and delete chmod and capture the output from standard error to a file named err.txt

```
huntersike@ubuntu:~$ rm /bin/chmod 2>err.txt
```

9. Capture the command to create test1, test2, test3, test33, stu1, stu2, stu22.

```
touch test1 test2 test3 test33 stu1 stu2 stu22
```

10. Using meta characters and a single ls command list all files named test.

```
huntersike@ubuntu:~$ ls test*
test1  test2  test3  test33
```

11. Using meta characters and a single ls command list only the files with the number 2 or 22 in them.

```
huntersike@ubuntu:~$ ls *2
stu2  stu22  test2
```

12. Using meta characters and a single ls command list only the files with a single 2 not 22 in them.

```
huntersike@ubuntu:~$ ls test2 stu2
stu2  test2
```

13. In your home directory, capture the ls command piped to more, and the output from more.

```
huntersike@ubuntu:~$ ls | more
almost
cscd240
Desktop
Documents
Downloads
err.txt
examples.desktop
lab1
Music
myTest
Pictures
Public
stu1
stu2
stu22
Templates
test1
test2
test3
test33
Videos
```

14. Issue the which command on ls. Was and where was the command found?

```
huntersike@ubuntu:~$ which ls
/bin/ls
```

Yes, the command was found in the bin directory.

15. Issue the which command on pthread.h. Was the command found? If it was not found why not?
How would you modify this.

No the command was not found, This means that the command is not in my path.
Using the find command may work better.

16. Using only octal values add executable access to test1, test2, test3.

```
huntersike@ubuntu:~$ chmod 775 test1 test2 test3
```

17. Using only alphanumeric characters remove read access from stu1 and stu2.

```
huntersike@ubuntu:~$ chmod ugo-r stu1 stu2
```

18. Execute help set

```
huntersike@ubuntu:~$ help set
```

19. Set the noclobber option on err.txt

```
huntersike@ubuntu:~$ set noclobber err.txt
```

20. Issue an ls –al redirected to err.txt. What was the output and why?

```
huntersike@ubuntu:~$ ls < err.txt
almost   Documents  examples.desktop  myTest    stu1   Templates  test3
cscd240  Downloads  lab1              Pictures  stu2   test1      test33
Desktop  err.txt    Music             Public    stu22  test2      Videos
```

It still output to the screen instead of err.txt. When noclobber is activated bash does not overwrite and existing file with redirect commands.

21. Remove the noclobber option from err.txt

```
huntersike@ubuntu:~$ set 0 noclobber err.txt
```

22. Explain the --help option for a program.

The help option only works for commands, not programs.

23. Explain double quotes, single quotes and the accent

Both double quotes and single quotes designate the line of characters between them a string. Accents can be used within double quotes to execute a command before the line is designated a string. The accents do not work inside single quotes. Everything inside single quotes will be saved as a string exactly how it was entered.