Blaine Mason

Lab-01

Dr.Park

Task 1:

1.) Write down the absolute path to the directory named include on the far left of the figure.

```
/usr/include
```

Task 2:

1.) Write down the relative path from the directory /dev to the file /bin/ls.

```
../bin/ls
```

Task 3:

2.) Write out your answers to the following questions

a.) What are the permissions on the file?

```
(base) ➜  ~ ls -l foo
-rw-rw-r-- 1 blaine blaine 0 Feb  4 12:51 foo
```

- read write for users, read write for groups, and just read for others.

b.) Who owns the file?

```
(base) ➜  ~ ls -l foo
-rw-rw-r-- 1 blaine blaine 0 Feb  4 12:51 foo
```

- "blaine" owns the file

c.) What group is associated with the file?

```
(base) ➜  ~ ls -lg foo
-rw-rw-r-- 1 blaine 0 Feb  4 12:51 foo
```

- the "blaine" group is associated with this file

d.) Are you in the group? (Invoke groups to find out).

- From the answer below we can see that I am in fact a member of the "blaine" group

e.) Write down the names of all the groups you are in.

```
(base) ➜  ~ cat /etc/group
root:x:0: daemon:x:1: bin:x:2: sys:x:3: adm:x:4:syslog,blaine
tty:x:5:syslog disk:x:6: lp:x:7: mail:x:8: news:x:9: uucp:x:10: man:x:12:
proxy:x:13: kmem:x:15:
dialout:x:20: fax:x:21: voice:x:22: cdrom:x:24:blaine floppy:x:25:
tape:x:26: sudo:x:27:blaine audio:x:29:pulse dip:x:30:blaine www-data:x:33:
backup:x:34: operator:x:37: list:x:38: irc:x:39: src:x:40: gnats:x:41:
shadow:x:42: utmp:x:43: video:x:44: sasl:x:45: plugdev:x:46:blaine
staff:x:50: games:x:60: users:x:100: nogroup:x:65534:
systemd-journal:x:101: systemd-network:x:102: systemd-resolve:x:103:
systemd-timesync:x:104: crontab:x:105: messagebus:x:106: input:x:107:
kvm:x:108: render:x:109: syslog:x:110: tss:x:111: bluetooth:x:112: ssl-
cert:x:113: uuidd:x:114: tcpdump:x:115: avahi-autoipd:x:116: rtkit:x:117:
ssh:x:118: netdev:x:119: lpadmin:x:120:blaine
avahi:x:121: scanner:x:122:saned saned:x:123: nm-openvpn:x:124:
whoopsie:x:125: colord:x:126: geoclue:x:127: pulse:x:128: pulse-
access:x:129: gdm:x:130: lxd:x:131:blaine
blaine:x:1000: sambashare:x:132:blaine systemd-coredump:x:999:
lightdm:x:133: nopasswdlogin:x:134:
```

3.) Change the permissions on ~/foo so owner has only execute permission, group has only write permission, and all others have both read and execute permission. Write out the command(s) you used to do this.

```
(base) ➜  ~ chmod u-rw foo
(base) ➜  ~ chmod u+x foo
```

```
(base) ➜  ~ chmod g-r foo
```

```
(base) ➜  ~ chmod o+x foo
```

```
(base) ➜  ~ ls -l foo
---x-w-r-x 1 blaine blaine 0 Feb  4 13:23 foo
```

4.) Attempt to delete the file. Write down the command you used and what happened.

```
(base) ➜  ~ rm foo
rm: remove write-protected regular empty file 'foo'?
```

5.) Change permissions on ~/foo so you can delete it, then do so.

```
(base) ➜  ~ chmod u+w foo
(base) ➜  ~ rm foo
```

Task 4:

2.) There are many options available for ls but the most important ones are a,l, and R. Write down the meanings of each of these options.

```
-a
    do not ignore entries starting with .

-l
    use a long listing format

-R
    list subdirectories recursively
```

3.) Move to your home directory and invoke ls with no options. Now invoke ls -l. Now invoke ls -a. Write down how the output of ls differs from the output of ls -a.

- ***ls*** lists the directories.
- ***ls -a*** lists the all the dotfiles, files, and directories.

4.) Different entities may have the same name. For example time as a user command differs from time as a system call. Write down the NAME

information for both of these time entities.

```
(base) ➜  ~ man 1 time
NAME
       time - run programs and summarize system resource usage
```

```
(base) ➜  ~ man 2 time
NAME
       time - get time in seconds
```

5.) Of course, man itself is a user command, so it should have a man page. Write down the NAME information for the man page on man.

```
(base) ➜  ~ man man
NAME
       man - an interface to the system reference manuals
```

Task 5:

2.) Write down the result of invoking the command pwd.

```
(base) ➜  ~ pwd
/home/blaine
```

3.) Walk one level up the file system hierarchy using the cd command. Write down the exact command you invoked.

```
(base) ➜  ~ cd ..
```

4.) Write down the result of invoking the command pwd.

```
(base) ➜  ~ pwd
/home
```

5.) Write down the permissions on your home directory.

```
(base) ➜  /home ls -l
drwxr-xr-x 35 blaine blaine 4096 Feb  4 13:54 blaine
```

6.) Get back to your home directory.

```
(base) ➜  /home cd
(base) ➜  ~
```

Task 6:

1.) Write down the file type of /bin/bash.

```
(base) ➜  ~ file /bin/bash
/bin/bash: ELF 64-bit LSB shared object
```

2.) Create an empty file named foo. Write down the file type of this file.

```
(base) ➜  ~ file foo
foo: empty
```

3.) Write down the file type of the file ~/.bash_history (yes, there is a dot before the file name. It indicates that this is a "hidden" file, not displayed with ls by default.)

```
(base) ➜  ~ file ~/.bash_history
/home/blaine/.bash_history: ASCII text
```

4.) Write down the file type of /usr/local.

```
(base) ➜  ~ file /usr/local
/usr/local: directory
```

Task 7:

4.) The head command allows you to see the top part of a file. You may specify the number of lines you want, or default to ten lines. Use head

to display the first 5 lines of ~/.bash_history. Write down the exact command you used.

```
(base) ➜  ~ head -n 5 ~/.bash_history
sudo apt install plank
sudo apt-get install plank
sudo apt remove gnome-shell-extension-ubuntu-dock
restart now
shutdown now
```

5.) 5. The tail command works like head, except that it shows the last lines of a file. Use tail to display the last 10 lines of ~/.bash_history. Write down the exact command you used.

```
(base) ➜  ~ tail -n 10 ~/.bash_history
exit
chsh -s /usr/bin/zsh blaine
exit
nvim test.tex
exit
ls
PS1='PEXPE\[\]CT_PROMPT>' PS2='PEXPE\[\]CT_PROMPT_' PROMPT_COMMAND=''
export TERM=dumb PAGER=cat
jupyter-notebook
exit
```

Task 8:

1.) Copy the file ~/.bash_history to the file ~/dotbashhistory. Write down the exact command you used.

```
(base) ➜  ~ cp ~/.bash_history ~/dotbashhistory
```

2.) Write down the file and directory permissions necessary to successfully copy a file in general.

- source directory: xr
- source file: r
- destination directory: xw
- destination file: w

3.) Invoke the exact command again in an attempt to copy the "source" to the now-existing "destination." Write down how the system handled

this.

```
(base) ➜  ~ cp ~/.bash_history ~/dotbashhistory
```

- no output

4.) Compare the permissions and date on the original file and the copy. Write down if and how they differ.

```
(base) ➜  ~ ls -l ~/.bash_history
-rw------- 1 blaine blaine 4062 Dec 11 13:02 /home/blaine/.bash_history
```

```
(base) ➜  ~ ls -l ~/dotbashhistory
-rw------- 1 blaine blaine 4062 Feb  4 14:18 /home/blaine/dotbashhistory
```

- No difference between permissions, but they have different dates.

5.) Write down the cp command you would use to insure that the copy has the same permissions and date as the original.

```
(base) ➜  ~ cp ~/.bash_history ~/dotbashhistory --preserve
```

Task 9:

1.) Rename your ~/dotbashhistory as ~/dotbashhistory.old. Write down the exact command you used.

```
(base) ➜  ~ mv ~/dotbashhistory ~/dotbashhistory.old
```

Task 10:

1.) Remove ~/dotbashhistory.old. Write down the exact command you used.

```
(base) ➜  ~ rm ~/dotbashhistory.old
```

Task 11:

1.) Create the directory ~/Foo. Write down the exact command you used.

```
(base) ➜  ~ mkdir ~/Foo
```

2.) Write down the permissions of this new directory.

```
drwxrwxr-x  2 blaine blaine      4096 Feb  4 14:27 Foo
```

- user: read write executable
- group: read write executable
- other: read executable

3.) Create a directory named Bar, within the Foo directory. Write down the exact command you used.

```
(base) ➜  ~ mkdir ~/Foo/Bar/
```

Task 12:

1.) Write down what happens when you invoke rmdir ~/Foo.

```
(base) ➜  ~ rmdir ~/Foo
rmdir: failed to remove '/home/blaine/Foo': Directory not empty
```

2.) Write down a sequence of rmdir commands that would allow removal of the ~/Foo directory and its subdirectory Bar. Don't invoke the commands, just write down what you would do.

```
(base) ➜  ~ rmdir ~/Foo/Bar
(base) ➜  ~ rmdir ~/Foo
```

3.) Invoke a single rm command to remove the ~/Foo directory and all its contents (check the man page). Write down the exact command you

used. Hint: it involves two flags, one to force the operation, the other to descend the directory recursively.

```
(base) ➜  ~ rm -rf ~/Foo
```