CS2211a Assignment 1 Vivian Lam

Q1(a)

```
obelix.gaul.csd.uwo.ca[42]% ls -a ?????????????????
activation-client*  gnome-search-tool*  gnome-text-editor@
gnome-font-viewer*  gnome-sys-suspend*  gst-xmllaunch-0.8*
```

%%%

Q1(b)

```
obelix.gaul.csd.uwo.ca[5]% ls -a ?z*
7z*            bzegrep*       bzmore*        gzfgrep*       gznew*
7za*           bzfgrep*       gzcat*         gzforce*       tzselect*
7zr*           bzgrep*        gzcmp*         gzgrep*
bzcat*         bzip2*         gzdiff*        gzip*
bzcmp*         bzip2recover*  gzegrep*       gzless*
bzdiff*        bzless*        gzexe*         gzmore*
```

%%%

Q1(c)

```
obelix.gaul.csd.uwo.ca[7]%  ls -a *[ij]
fmli*              gtkdoc-scanobj*   native2ascii@     vi*
gtkdoc-scangobj*   idlj@             tsoljdslabel-ui*  xdtosj@
```

%%%

Q1(d)

```
obelix.gaul.csd.uwo.ca[61]% ls -a -d [A-Z]*
CC@               ControlPanel@   HtmlConverter@
CCadmin@          DBMirror.pl*    X11/
```

%%%

Q1(e)

```
obelix.gaul.csd.uwo.ca[65]% ls -a -d [A-CE-KL-Z]*
CC@               CCadmin@        ControlPanel@   HtmlConverter@  X11/
```

$$$

Q2(a)

obelix.gaul.csd.uwo.ca[11]% mkdir public_html

%%
Q2(b)

obelix.gaul.csd.uwo.ca[12]% ls -ld public_html
drwxr-xr-x   2 vlam54   vlam54   2 Sep 24 14:03 public_html/

%%
Q2(c)

obelix.gaul.csd.uwo.ca[13]% cd public_html

%%
Q2(d)

obelix.gaul.csd.uwo.ca[17]% touch abc

%%
Q2(e)

obelix.gaul.csd.uwo.ca[19]% cd ..

%%
Q2(f)

obelix.gaul.csd.uwo.ca[26]% chmod 300 public_html

%%
Q2(g)

You can see information about **abc** when executing **ls public_html/abc,** but not when executing **ls public_html** because the permission of **public_html** was changed to be 300. This means the user can only write and execute, not read.

obelix.gaul.csd.uwo.ca[28]% ls public_html/abc
public_html/abc
obelix.gaul.csd.uwo.ca[29]% ls public_html
public_html: Permission denied
obelix.gaul.csd.uwo.ca[30]%

$$

Q3(a)

```
obelix.gaul.csd.uwo.ca[30]% ls -r -d .*[r][c]
.twmrc*   .tcshrc*  .mwmrc*   .dmrc*    .cshrc*
```

%%

Q3(b)

```
obelix.gaul.csd.uwo.ca[68]% finger vlam54
Login name: vlam54                      In real life: Vivian Amie Lam
Directory: /gaul/s1/student/2015/vlam54 Shell: /local/tcsh
On since Sep 25 14:03:10 on pts/6 from nexus-9.wireless.uwo.ca
No unread mail
No Plan.
```

%%

Q3(c)

```
obelix.gaul.csd.uwo.ca[99]% echo 'I dont really have any plans, except
for finishing this assignment I guess' >.plan
```

%%

Q3(d)

```
obelix.gaul.csd.uwo.ca[100]% chmod 744 .plan
```

%%

Q3(e)

```
obelix.gaul.csd.uwo.ca[102]% finger vlam54
Login name: vlam54                      In real life: Vivian Amie Lam
Directory: /gaul/s1/student/2015/vlam54 Shell: /local/tcsh
On since Sep 25 14:03:10 on pts/6 from nexus-9.wireless.uwo.ca
No unread mail
Plan:
I dont really have any plans, except for finishing this assignment I
guess
```

%%

Q3(f)

```
obelix.gaul.csd.uwo.ca[103]% chmod 600 .plan
```

%%

Q3(g)

```
obelix.gaul.csd.uwo.ca[104]% finger vlam54
Login name: vlam54                      In real life: Vivian Amie Lam
Directory: /gaul/s1/student/2015/vlam54 Shell: /local/tcsh
On since Sep 25 14:03:10 on pts/6 from nexus-9.wireless.uwo.ca
No unread mail
No Plan.
```

%%

Q3(h)

Yes there is a difference between the (b), (e), and (g) outputs. This is because at each step we edited files, specifically the **.plan** file. In step (b) there was no **.plan** file so the output displayed **No Plan.** By (e) a **.plan** file was created and the permission of the file was set to be **744**, allowing the user to read, write and execute and group and other users permission to read. This permission setting allows it to display the contents of the file**: Plan: I dont really have any plans, except for finishing this assignment I guess .** However, by step (g) the permission of **.plan** file was changed to **600** which allows the user to read and write, but not execute and it does not allow groups or other users to read, write or execute. Since we cannot execute the file the terminal displays **No Plan.**

$$

Q4(a)

obelix.gaul.csd.uwo.ca[16]% mkdir Working-Area

%%
Q4(b)

obelix.gaul.csd.uwo.ca[25]% mkdir Dir1
obelix.gaul.csd.uwo.ca[26]% cat > File1
obelix.gaul.csd.uwo.ca[27]% ^D

%%
Q4(c)

obelix.gaul.csd.uwo.ca[39]% cd Dir1
obelix.gaul.csd.uwo.ca[40]% mkdir Dir3 Dir4

%%
Q4(d)

obelix.gaul.csd.uwo.ca[42]% cd Dir3
obelix.gaul.csd.uwo.ca[43]% cat > File3
obelix.gaul.csd.uwo.ca[44]% ^D

%%
Q4(e)

obelix.gaul.csd.uwo.ca[48]% cd ..
obelix.gaul.csd.uwo.ca[50]% cd Dir4
obelix.gaul.csd.uwo.ca[51]% touch File4 File5 File6

%%
Q4(f)

obelix.gaul.csd.uwo.ca[53]% cd ..
obelix.gaul.csd.uwo.ca[55]% cd ..
obelix.gaul.csd.uwo.ca[73]% ln -s /student/vlam54/Working-Area/Dir1/Dir4 Dir2

%%
Q4(g)

obelix.gaul.csd.uwo.ca[21]% chmod 700 Working-Area

%%
Q4(h)

obelix.gaul.csd.uwo.ca[24]% cd Working-Area/Dir1
obelix.gaul.csd.uwo.ca[26]% chmod 750 Dir3

%%
Q4(i)

```
obelix.gaul.csd.uwo.ca[28]% cd Dir3
obelix.gaul.csd.uwo.ca[29]% chmod 755 File3

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Q4(j)

```
obelix.gaul.csd.uwo.ca[31]% cd /student/vlam54/Working-Area/Dir1/Dir4
obelix.gaul.csd.uwo.ca[32]% chmod 511 File5

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

CS2211a Assignment 1 Vivian Lam

Q5(a)

```
obelix.gaul.csd.uwo.ca[35]% cd $HOME
obelix.gaul.csd.uwo.ca[36]% cat > letter.txt
01
02
03
04
05
06
07
08
09
10
11
12
obelix.gaul.csd.uwo.ca[37]% ^D
```

%%%

Q5(b)

```
obelix.gaul.csd.uwo.ca[37]% cat letter.txt
01
02
03
04
05
06
07
08
09
10
11
12
```

%%%

Q5(c)

The command **tail -3 ~/letter.txt** shows the last three lines of the file **letter.txt** while the command **tail +3 ~/letter.txt** shows all the lines starting from line 3 to the end of the file.

```
obelix.gaul.csd.uwo.ca[38]% tail -3 ~/letter.txt
10
11
12
obelix.gaul.csd.uwo.ca[39]% tail +3 ~/letter.txt
03
04
05
06
07
```

08
09
10
11
12

%%

Q5(d)

The command **head -3 ~/letter.txt** shows the first three lines of the file **letter.txt** while the command **head +3 ~/letter.txt** shows the first ten lines, which is the default number for the head command. This happened because the shell interpreted the **+3** as a file or directory name and thus displayed the message **+3: No such file or directory.** The shell continued to use the head command on letter.txt and by default showed the first ten lines. So, the command we inputted was interpreted by the shell as "display the first ten lines of the two files: +3 and letter.txt"

```
obelix.gaul.csd.uwo.ca[40]% head -3 ~/letter.txt
01
02
03
obelix.gaul.csd.uwo.ca[41]% head +3 ~/letter.txt
+3: No such file or directory
==> /gaul/s1/student/2015/vlam54/letter.txt <==
01
02
03
04
05
06
07
08
09
10
```

%%

Q5(e)

The command **who | tee ~/letter2.txt | wc -l** displays how many people are currently on Gaul. To be specific, the command **who** displays the information of those who are currently on Gaul. This is then followed by a pipe **|** to connect this command with the command **tee ~/letter2.txt**, making the standard output of one command the input of another. The command **tee ~/letter2.txt** replicates the input from **who** into a file in the home directory called **letter2.txt** where each person from **who** is on a new line in **letter2.txt**. And finally, the command **wc -l** prints the number of newline counts from **letter2.txt**.

```
obelix.gaul.csd.uwo.ca[46]% who | tee ~/letter2.txt | wc -l
      17
```

%%
Q5(f)

```
obelix.gaul.csd.uwo.ca[32]% cal 11 1955
   November 1955
 S  M Tu  W Th  F  S
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

%%
Q5(g)

cat < letter.txt The shell opens the file while in **cat letter.txt** cat opens the file.

```
obelix.gaul.csd.uwo.ca[33]% cat letter.txt
01
02
03
04
05
06
07
08
09
10
11
12
obelix.gaul.csd.uwo.ca[34]% cat < letter.txt
01
02
03
04
05
06
07
08
09
10
11
12
```

%%
Q5(h)

The command **echo cat** makes the shell display the words **cat** on the terminal screen. On the other hand, the command **cat echo** tries to open a file called **echo**.

```
obelix.gaul.csd.uwo.ca[11]% echo cat
cat
obelix.gaul.csd.uwo.ca[12]% cat echo
cat: cannot open echo

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

CS2211a Assignment 1 Vivian Lam

Q6(a)

The command **cp -r** must be used to copy all the files under the **courses/** directory because the **-r** will recursively copy everything in the **courses/** directory. I also included the **/*** in **~vlam54/courses/*** to ensure it copies anything under this source directory. And lastly the . at the end of the statement means to put all the copied files into the current directory, which is the home director in this case.

obelix.gaul.csd.uwo.ca[57]% cp -r ~vlam54/courses/* .

%%

Q6(b)

obelix.gaul.csd.uwo.ca[64]% cp -r ~vlam54/courses/* ~vlam54

%%

Q6(c)

obelix.gaul.csd.uwo.ca[75]% chmod -R 700 ~vlam54/courses/*

$$

CS2211a Assignment 1 Vivian Lam

dQ7(a)

In unix absolute pathnames string together all the names and separates them with slashes, while relative pathnames gives the address relative to the current working directory. For example, if the current working directory was **/student/vlam54/Working-Area/** and we wanted to change to a subdirectory within **Working-Area**, **Dir1**, then

A command using an absolute pathname would be:
obelix.gaul.csd.uwo.ca[12]% cd /student/vlam54/Working-Area/Dir1
and a command using a relative pathname would be:
obelix.gaul.csd.uwo.ca[13]% cd Dir1

%%%
Q7(b)

The **./** directory is the current working directory

%%%
Q7(c)

The **../** directory is the parent directory

%%%
Q7(d)

The **~/** directory is the home directory

%%%
Q7(e)

The command **rm abc_dir** failed to remove **abc_dir** because there are still files in the **abc_dir** directory. The command said it is not empty even though no files are displayed because there are hidden files in that directory. The command **ls abc_dir** does not display hidden files, but the command **ls -a abc_dir** should show that there are still files in the **abc_dir** directory, they are just hidden (start with a dot).

%%%
Q7(f)

Three possible reasons that the file did not copy, despite existing are: No permission to read, No permission to write, or it exists in another directory and the command specified the wrong pathname to the file.

%%%
Q7(g)

The command **tty** displays the absolute pathname of the terminal device. The command **cp ~/.login /dev/pts/1** is trying to copy the **.login** file

to the terminal. A terminal only accepts inputs and displays the output to the screen and so it prints the contents of the **.login** file.

```
obelix.gaul.csd.uwo.ca[110]% tty
/dev/pts/1
obelix.gaul.csd.uwo.ca[115]% cp ~/.login /dev/pts/1
#
#  WGUI is twm or mwm
#

if (!($?HOSTTYPE)) then
  set HOSTTYPE = `uname -m`
endif

switch ($HOSTTYPE)
  case SunOS:
  case sun4:
  case sun4m:
  case sun4u:
  case i86pc:
    eval `tset -I -s -Q`
    #
    # LD_LIBRARY_PATH determines where to look for sharable libraries
at run time
    setenv LD_LIBRARY_PATH \

/usr/lib:/opt/SUNWspro/lib:/usr/openwin/lib:/usr/local/lib:/usr/dt/lib
:/usr/local/qt/lib:/usr/local/mysql/lib/mysql
    setenv LM_LICENSE_FILE /gaul/packages/max2work/y/license.dat
    #
    # The Window Manager
    setenv WGUI /usr/openwin/bin/twm
    #
    # Open Look Window Manager
    #   setenv WGUI "/usr/openwin/bin/olwm -follow"

    #
    # The QT libraries
    #
    # setenv QTDIR /usr/local/qt

breaksw
#
 case Linux:
 case i386:
 case i486:
 case i586:
 case i686:
 case i686:
 case i386-linux:
 case i486-linux:
 case i586-linux:
```

```
 case i686-linux:
 case x86_64-linux:
 case x86_64:
     unsetenv WGUI
     unset WGUI
 breaksw
#
 case sun3:
    setenv WGUI /usr2/lib/X11R4/bin/twm
 breaksw
#
 default:
   echo "** .login: Unknown Host Type **"
 breaksw

endsw
```

%%
Q7(h)

If **umask** shows the value **000** it means the owner, a group, and other users all have permission to read, write and execute the file with that **umask** value. From a security point of view this isn't secure as anyone can edit the file and taper with the file. If **umask** shows the value **002** it means the owner and a group have permission to read, write and execute while other users can only read an execute that particular file. From a security point of view this still unsecure as anyone can view the file, but only a select group can edit it.

$$