**Lab Module 10**

Create a directory called ***module10.*** All work will be done inside of this unless otherwise stated.

1. Create (using touch) a file called **file1.txt** inside of this new directory.

Use chmod to change **file1.txt** permission so that YOU can read, write and execute the file, the group can read it and all others can do nothing to it, not even read it. How did you do this?

chmod 740 file1.txt

2. Use the touch command to create an empty file in the ***module10*** directory called **file2.txt**. Use chmod to change **file2.txt** so you can read and write to it, the group can read, write and execute to it and all others can read it. How did you do it?

chmod 674 file2.txt

3. Go back to your home directory and use chmod to change the directory ***module10*** ‘s permissions to be read and writable to everyone, including yourself and grouup not executable. How did you do this?

chmod 606 module10

Try to change into ***module10***. Were you successful? Why or why not?

No, permission denied, because it allows reading and writing but not access (executing)

Do a long listing of the content of ***module10*** with the following command: *ls -l module10.*

Were you successful? Explain what happened (don't give me a screenshot, just explain)

Semi - got the names of the files inside the directory but was denied seeing permissions or file sizes

In your home directory touch a file called **test**. Use copy to copy it into ***module10*** directory.

What message did you get? Why?

mv: accessing `module10/test': Permission denied. because my group does not have permission to enter.

Change the permissions of the ***module10*** directory back to 755. Repeat step 7. Did it work now?

Yes.

So, what permissions do you need to copy a file into a directory?

WX (Write, Execute)

What permissions do you need to change into a directory?

X (execute)

4. Use the ls –l command to tell me what are the permissions on /etc/shadow file? How did you do this? Who can read it? Who owns it?

root owns it, root and group can read it

5. From your ***module10*** directory

Type: *umask*

What do you see?

0002

Create an empty file called **umask1.txt** in your ***module10*** directory. Using ls -l , what are the default permissions with the umask you have now?

Read and Write for user and group, read for all

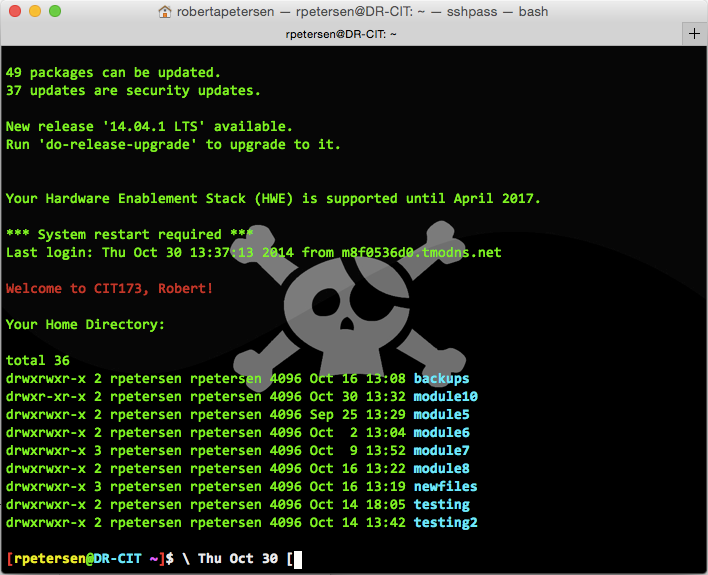
Change your umask to 0024

Create an empty file called *umask2.txt* in your *module10* directory. What are the default permissions with the umask you have now?

Read and Write for user, read for group, write for all

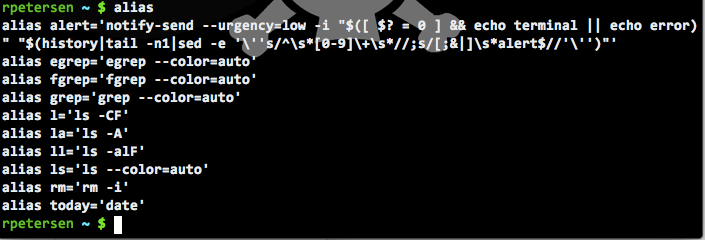
Change your umask back to what it was.

6. Use google to find out how to change your prompt to something cool. I would like to see color and your login name at the very minimum. Add one other element to it. Take a screenshot of it and place it here.



7. Create an alias that when you type “today” it will run the date command.

8. Create an alias called “rm” that when you type it, it will prompt you to confirm the removal (you may need to use the man pages for the rm command).

Type: *alias* and insert a screenshot here

9. In your module10 directory create a file using the echo called **link1** that has the word hello in it.

Use cat to confirm you did this correctly.

Type the following command to create a softlink to a file called **link2** in your ***module10*** directory:

*ln -s link1 link2*

Type: ls –li and insert a screenshot here

Type *echo goodbye >> link2*

Use cat to make sure it is in both files

Use the remove command to remove link1

Read link2 using cat. Are you successful? Why or why not?

Nope, because link2 was only a link to link1, which now doesn’t exist.

Use ls to see if it is there. Is it?

Yes, but Red, because broken/bad.

Remove link2 using remove command

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Let’s do this again in your module10 directory

Type: *echo hello again > link1*

Type the following command to create a hardlink to a file called **link2**

*ln link1 link2*

Type: *ls –li* and insert a screenshot here

Type *echo goodbye again >> link2*

Use cat to make sure it is in both files

Use the remove command to remove link1

Read link2 using cat. Are you successful? Why or why not?

Yes, because it was a hardlink with same node number. Same data.

Use ls to see if it is there. Is it?

Yes.

Remove link2 using remove command

Go back to your home directory and do the following:

Type: *ln module10 links*

What error message did you get?

ln: `module10': hard link not allowed for directory

Type: *ln –s module10 links*

You now have a directory called links that has the contents of module10 in it.

Type*: cd links*

Do a listing and insert a screenshot here

10. What do you conclude about this experiment and the difference between symbolic and hard links?

Symbolic links are typically better ;-) Symbolic can link to directories, or other file systems. hard links are just same inode with a different title?