**Regis University CC&IS**

**CS465 Unix**

**Lab Homework #2**

**MATTHEW HARTIGAN**

**23-Mar-2019**

Read through the following 30 steps and then complete them on a Unix system. Fill in your answers on this document. Points that will be given for each correct answer are listed beside each question.

WARNING: Case matters in Unix, so the case must be correct in the answers you give.

See the end of the document for submission instructions.

Notes:

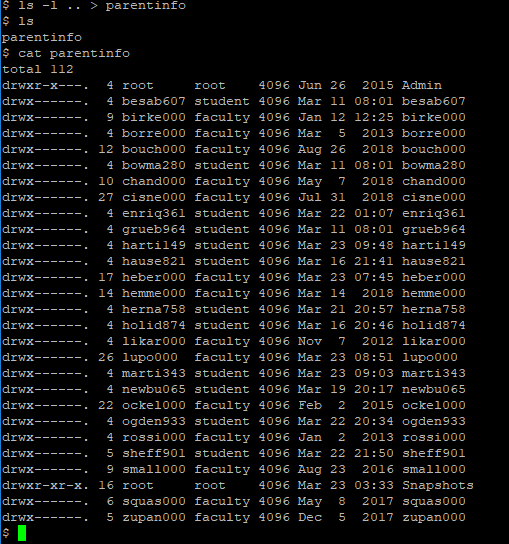
a) When asked to "**explain**" the output, explain it in your own words – do **not** just submit the command output!

b) When asked "**what command you used**", list the **entire** command, including any arguments and options you used.

1. From your home directory, capture a **long listing** of your **parent** directory to a file, using the command:    $ ls -l .. > parentinfo

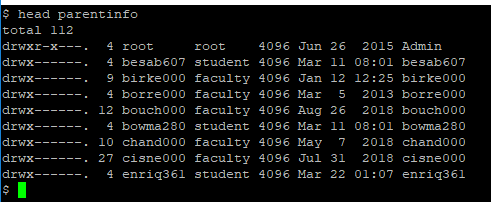
a) (1 point) What is saved in the file? (supply a complete copy of the ***contents*** of the file **parentinfo** here)

The “ls -l ..” portion of this command would normally output the long form contents of the parent directory (“..”) to the stdout (terminal screen). By adding the “> parentinfo” to the command, it redirects the output to a file named parentinfo and saves it in the current working directory. The contents of the file is the long form listing of every user profile that is defined for this particular directory hierarchy (which looks like our class list).



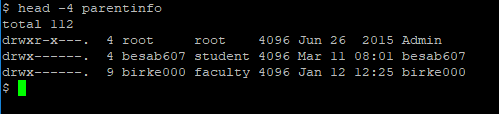
b) (2 points) Display the beginning of the the **parentinfo** file with the **head** command. How many lines are displayed (be sure to count blank lines)?

10 lines are displayed (i.e. the default number of lines).



c) (2 points) Modify the **head** command to display only the first 4 lines of the file **parentinfo**. What command did you use?

head -4 parentinfo



1. (5 points total - 1 per correct answer) Capture the contents of the manual page about the **cp** command in the file **cpinfo**, using the command:    $ man cp > cpinfo  
   Note: You do **not** need to supply a listing of this file.  
     
   Use the **more** command to view the **cpinfo** file.  
   The "More" prompt appears at the bottom of the page.  
   1. Try pressing the **Enter** key TWICE – explain what happens?

Pressing enter twice advances us down the contents of the file.

“—More—(33%)” at the bottom of the screen updates to “—More—(34%)”

* 1. Try pressing the **space bar** – explain what happens?

Pressing spacebar also advances us down the contents of the file, but more quickly than pressing enter did.

“—More—(34%)” at the bottom of the screen updates to “—More—(72%)”

* 1. Type the letter **b** – explain what happens?

Pressing b moves us back to the point we were at in the file previously. In other words, we moved back a page from “—More—(72%)” to “—More—(34%)”

* 1. What does the percentage at the bottom of the screen represent?

The percentage of the file that you have viewed (i.e. the amount that is ‘above’ your cursor).

* 1. Try typing the letter **q** – what happens?   
     You are returned back to the command prompt.

1. Using **vi**, create a file called **text1**.

Type the following three lines into the file (and nothing else).

This is line one

Line two is here

What is on line three?

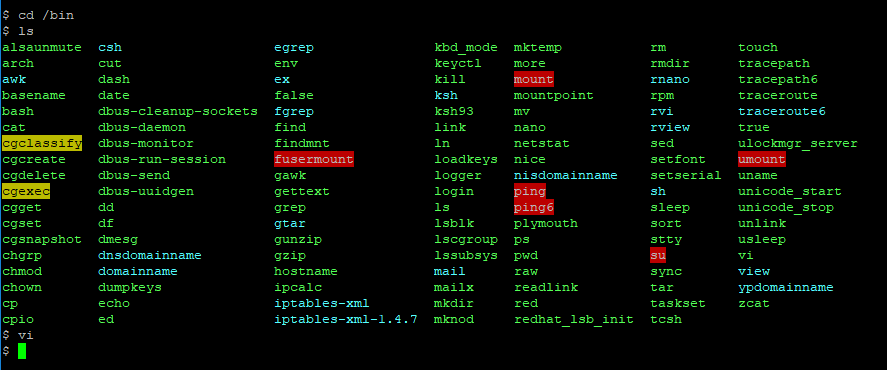
Save the file and exit **vi**.

1. (2 points) What command did you use to enter **vi**?

vi text1

However, this command technically opened vim instead of vi. I took a look in the /bin directory as shown below, and found the following article explaining how, although vi is listed in /bin, it is a symlink to vim.

<https://superuser.com/questions/852177/why-does-the-vi-command-open-vim-editor>

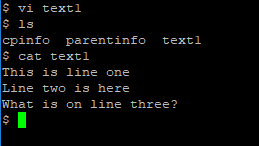


1. (2 points) What command(s) did you use to save the file and exit vi?

ZZ

1. (2 points) Without re-opening the file in **vi**, display file **text1** that you created above. What command did you use?

cat text1



1. (5 points total) Using **vi**, edit the first line in file **text1** to read:  
    Line One  
   After editing, re-save the file and exit **vi**.

NOTE: You should **not** just delete or replace the entire first line and re-type it. Instead use vi commands to edit each part of the old first line to become the new first line shown. In other words, you must ***delete*** the first two words and ***capitalize*** the first letter of the last two words.

List ALL the **vi** commands and movement keys you used to move around and to change each item on the line, in the order you used them.

vi text1<enter>

dw

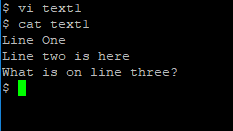
dw

rL

/one<enter>

rO

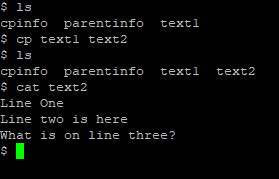
ZZ



1. From the command prompt (i.e. **not** from within vi), copy the file **text1** to a file named **text2**.

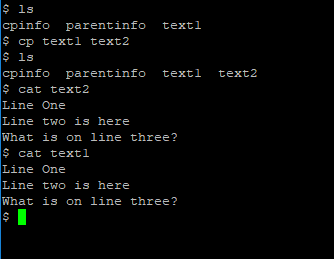
a) (2 points) What command did you use?

cp text1 text2



b) (1 point) Does file **text1** still exist? Why?

Yes. It still exists because we copied the file and its contents to a second file named text2 but did not delete the file text1 or its contents.



1. Using **vi**, add the following line to file **text2 as the last line in the file:**  
     
      Where for art Line Four?  
     
   (3 points) List ALL the **vi** commands you used to move around in the file and to add the line.

L

$

a

<enter>

Where for art Line Four?  
<esc>

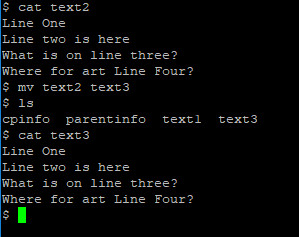
ZZ

1. Display file **text2** to see that the change made above is there. Then rename file **text2** to be file **text3**.

a) (2 points) What command did you use?

cat text2

mv text2 text3

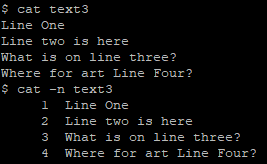


b) (1 point) Does file **text2** still exist? Why?

Technically text2 still exists because we just renamed it text3 (i.e. we didn’t delete it).

1. (2 points) Use the **cat** command to display file **text3**. Now use the **-n** option with the **cat** command to display file **text3**. How is the second display different from the first?

“cat text3” displays the file contents to the standard output. “cat -n text3” uses the -n option to include a line number for each line of the file contents that is displayed to standard output while using the cat command.



1. (12 points total – 1 point per answer) Open the **text3** file for editing, using **vi**. Then do the following:
   1. Use ONE **vi** command to **move to the last line** in the file. What command did you use?

L

* 1. Use ONE **vi** command, **delete the first word on the last line.** What command did you use?  
     dw
  2. Use ONE **vi** command to **move to the end** of the last line. What command did you use?

$

* 1. Use ONE **vi** command to replace the **?** with a **!**. What command did you use?

r!

* 1. Use ONE **vi** command to move to the top of the screen. What command did you use?

H

* 1. Use ONE **vi** command to delete the **entire first line**. What command did you use?

dd

* 1. Use ONE **vi** command to **undo** the last change you made (i.e. to undelete the first line). What command did you use?

u

* 1. Use ONE **vi** command to **move** to the next **'w'** character in the file.

/w<enter>

* 1. Use ONE **vi** command to copy the current line. What command did you use?

yy

* 1. Use ONE **vi** command to move to the third line. What command did you use?

3G

* 1. Use ONE **vi** command to paste (put) the copied line before the cursor (i.e. before line 3). What command did you use?

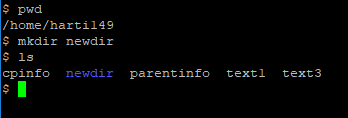
P

* 1. Use ONE **vi** command to exit vi, **discarding all changes** you made during this editing session.

:q!<enter>

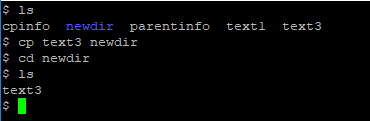
1. (2 points) Create a subdirectory called **newdir**, directly under your home directory. What command did you use?

mkdir newdir



1. (2 points) From your home directory, copy file **text3** into subdirectory **newdir**. What command did you use?

cp text3 newdir

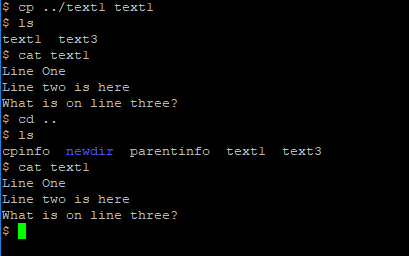


1. (2 points) Move to the subdirectory **newdir**. What command did you use?

cd newdir

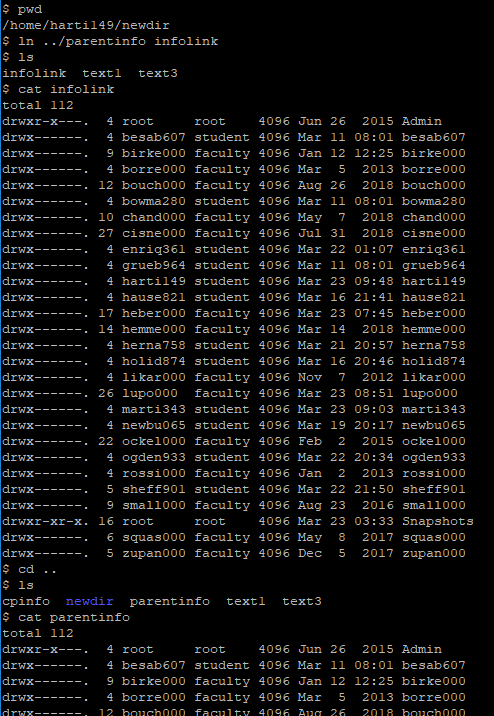
1. (2 points) From within subdirectory **newdir**, copy the file **text1** from your home directory into the new subdirectory **newdir**. What command did you use?

cp ../text1 text1



1. (2 points) Create a link called **infolink** in the subdirectory **newdir** that links you to the file **parentinfo** in your home directory. What command did you use?

ln ../parentinfo infolink



1. Display a listing of all files, including hidden ones, in the subdirectory **newdir**. What is the output? (submit an exact listing of your output here)

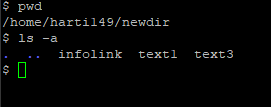
.

..

infolink

text1

text3



1. Using **vi**, add the following line to the end of file **infolink**, and resave the file:  
      This is the end.

(2 points) Then display the last 5 lines of the file.   
 What command did you use?

Commands / keyboard input leading up to displaying last 5 lines of edited infolink file:

vi infolink

L

$

a

<enter>

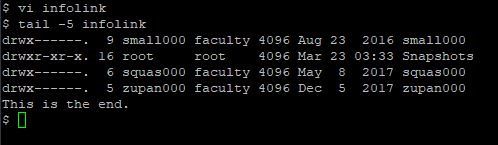
This is the end.

<esc>

ZZ

Final command:

tail -5 infolink

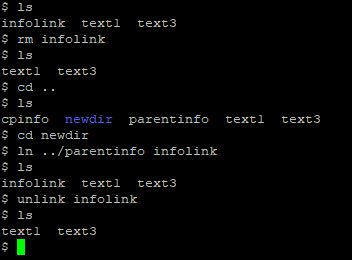


1. (2 points) Delete the **infolink** link. What command did you use?

rm infolink

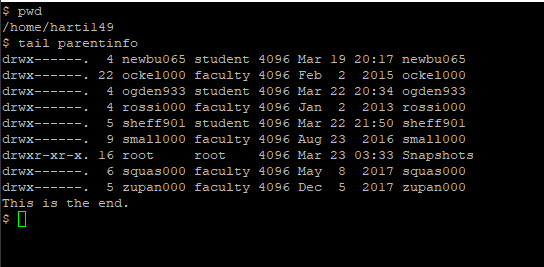
You can also use the following:

unlink infolink



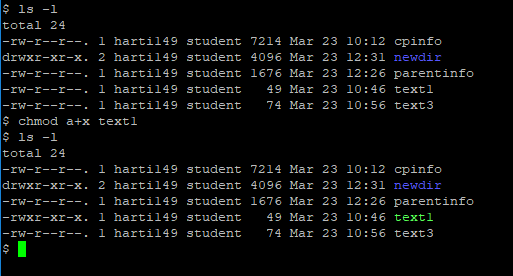
1. (2 points) Move back to your **home** directory. Use the **tail** command to display the end of the file **parentinfo**. Is the line you added via the link **infolink** there? Explain why the line is or is not there.

Yes, the line that we added via the link infolink is still there. This is because all links of a file point to the same file (in this case, our link, infolink pointed to the original file, parentinfo). When we opened infolink and added a line, we saved the addition before we exited. Therefore, even after we deleted the link in Task #18, the edit is still reflected in the original file, parentinfo.



1. Change the file permissions as indicated in the following three steps (display a long listing each time, to confirm that the file protections have been changed correctly).  
   1. (2 points) Change the file permission on file **text1** so that everyone on the system has execute access to it (do not change any other permission settings on the file). What command did you use?

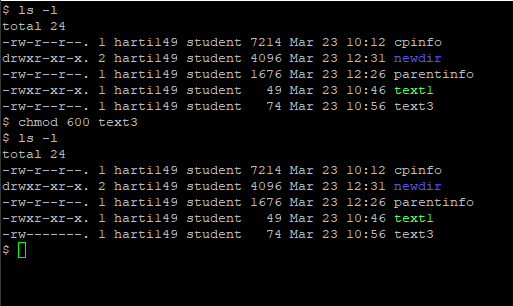
chmod a+x text1



NOTE: Leave the execute permission on this file until you have completed all of this homework.

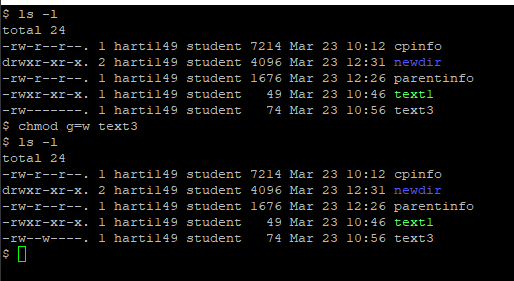
* 1. (2 points) Using an **octal permission setting**, change the file permission on file **text3** so that **only you** have read and write access to it, and nobody else has ANY access to it. What command did you use?

chmod 600 text3



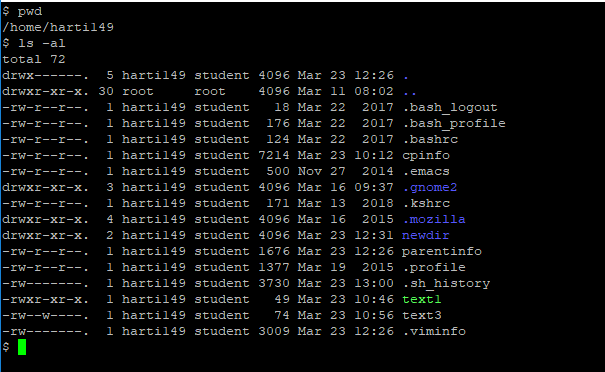
* 1. (2 points) Using a **symbolic string** (cluster selection), change the file permission on file **text3** so that **your group** has write permission to it (do not change any other permission settings on the file). What command did you use?

chmod g=w text3



1. (1 point) Do a ***long*** listing of all the files in your home directory (including hidden files), using **ls -al**. What was the **output**? (submit an exact listing of the output here)

See below:



1. Notice the access permissions assigned to each file (note that in Unix, directories are also files). Answer these questions:  
   1. (1 point) List the names of all the files that are considered **directories**.

.

..

.gnome2

.mozilla

newdir

* 1. (1 point) List the names of all the files that have **execute** permission.

.

..

.gnome2

.mozilla

newdir

text1

* 1. (1 point) What do the **read/write** permissions mean for a **regular file**?

For a regular file, read permission means that the file contents can be viewed or printed.

For a regular file, write permission means that the file contents can be changed or deleted.

Having read/write permissions means you can do both of these!

* 1. (1 point) Explain how **read** permission for a **directory** is this different than read permissions for regular files?

When you have read permission for a regular file, you are able to view the entire contents of the file. In contrast, when you have read permission for a directory, you are able to view the list of all the contents (files, subdirectories) in that directory. The difference lies in the fact that a file and a directory are fundamentally different to begin with.

* 1. (1 point) Explain how **write** permission for a **directory** is this different than write permissions for regular files?

When you have write permission for a regular file, you are able to change / delete the contents of the file. In contrast, when you have write permission for a directory, you can add or remove entries from the directory contents.

* 1. (2 points) Do you **own** all the files listed? If not, which ones don't you own? Who does?

No, I do not own the parent directory (“..”). This is the /home directory. This is owned by the user ‘root’.

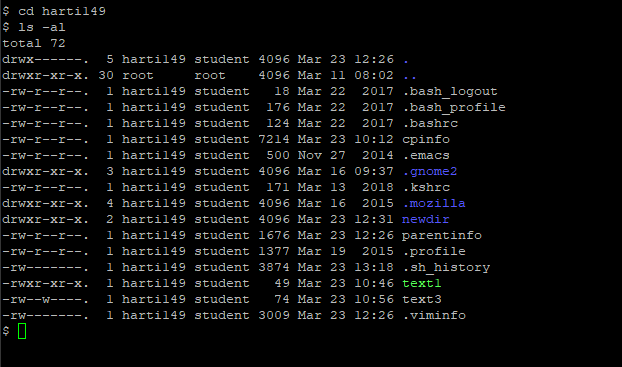
* 1. (1 point) What is the name of the **group** that is allowed access to the files in your home directory?

My home directory is /home/harti149

The ‘student’ group is allowed to access the files in it.

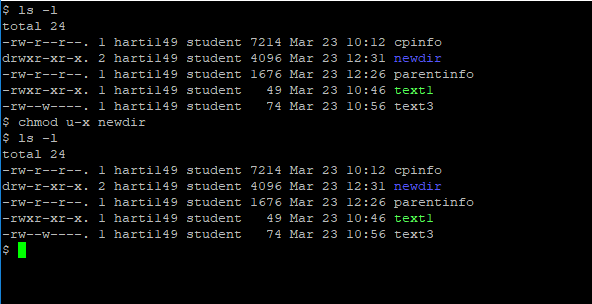
* 1. (2 points) Which default **permissions** does the group have for your regular files? For your directory files?

The student groups default permissions for my regular files is read only (r--). An example of this is the ls -l command output for cpinfo.



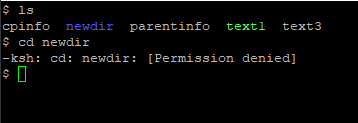
1. (2 points) Take away your execute permission from the **newdir** directory. What command did you use?

chmod u-x newdir



1. (2 points) Try to change directories to directory **newdir**. EXPLAIN what happens (do not just state what the output is – explain why you got this output). Re-grant execute permission on the **newdir** directory before going on to the next question.

In Task #23 we took away execute permission for the user (me) on the newdir directory. In Task #24 when we try to access the newdir directory, we are denied. This is because, for directories, the “x” permission allows the user/group/other to search and cd into the directory. When the “x” permission isn’t granted, you can’t.

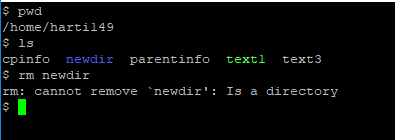


1. (2 points) Try to delete the subdirectory **newdir** using the command:  
      $ rm newdir  
   EXPLAIN what happens (do not just state what the output is – explain why you got this output).

From “man rm”:

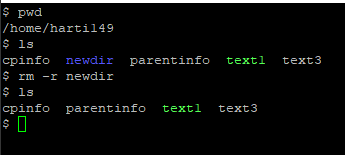


When we try to delete newdir using the rm command, we get an error message saying that we can’t because it is a directory. In other words, the default rm command cannot remove directories. You need to give it the ‘-r’ option to recursively delete all the directory’s contents and the directory itself.



1. (2 points) Figure out how to modify the **rm** command so that you can use it to delete the **newdir** subdirectory. What command and option did you use?

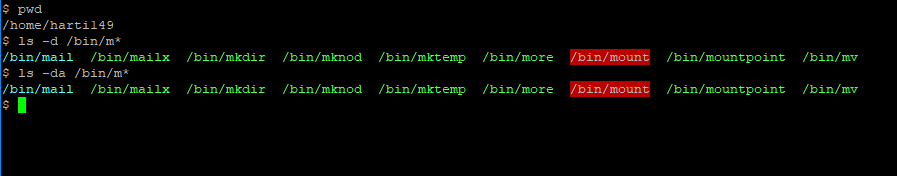
rm -r newdir



1. **Without changing your current working directory**, display a listing of the files/directories in the root's subdirectory **bin** that begin with "m".

a) (2 points) What command did you use?

ls -d /bin/m\*



b) (1 point) What files are listed?

mail

mailx

mkdir

mknod

mktemp

more

mount

mountpoint

mv  
(all with /bin/ prepended)

1. Change directories to the directory **/bin**. Use ONE command to list all the files in the directory that begin with "a", "d" or "w".   
   1. (1 point) What command did you use to change directories?

cd /bin

* 1. (2 points) What command did you use to list the files?

ls -d a\* d\* w\*

* 1. (1 point) What files are listed?

alsaunmute

arch

awk

dash

date

dbus-cleanup-sockets

dbus-daemon

dbus-monitor

dbus-run-session

dbus-send

dbus-uuidgen

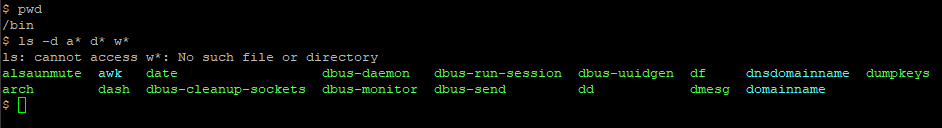
dd

df

dmesg

dnsdomainname

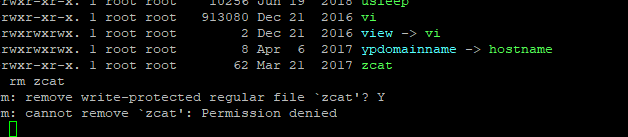
domainname



* 1. (2 points) Try to delete one of the files. Answer Y to any questions asked. Explain what happens. (do not just state what the output is – explain why you got this output)

After running an ‘ls -l’ command to view the permissions for each file in the directory, we can see that ‘root’ is the user and group for almost every file. This means that my username is classified as ‘other’. We can also see that the permissions for other are ‘r-x’ for almost every file. This means that I can read and execute, but not edit or delete most of the files.

I attempted to delete the ‘zcat’ file with the command ‘rm zcat’. I answered Y to the question that was thrown, but was denied because I don’t have write access as described above.

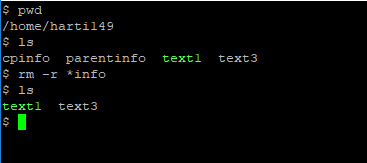


1. (2 points) Change directories back to your home directory, using the simplest command you can think of. What command did you use?

cd /home/harti149

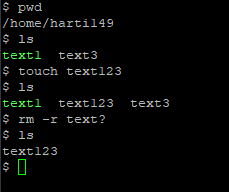
1. (2 points) Use ONE command with a wildcard to delete all files in your home directory that end with the word **info**. What command did you use?

rm -r \*info



1. (2 points) Use ONE command with a wildcard to delete all files in your home directory that start with the letters **text** and contain only one character following the letters **text**. What command did you use?

rm -r text?



**Submission**

This homework assignment is due by midnight Sunday (last day of Week 2).

Submit a filled in copy of this Word document to the **Homework Assn 2** drop box (located under the Dropbox tab in the online course).

Before submitting the Word file with your answers, you MUST rename it as follows:

### Lastname-hwk2.docx

For example:

### Smith-hwk2.docx