Name: Zakeriya Muhumed

Shell Commands (20 points)

The first two commands to become familiar with are the commands used to get information about commands. The two commands are `man` (short for "user's manual") and `info`. To find out about the `ls` command, you can type "`man ls`" or "`info ls`". The `info` command tends to give more detailed information, so try `man` first and if you don't find what you're looking for there, try `info`. Once you find what you're looking for, you can type `q` to quit from either utility.

What do the following commands do? Give a brief description. (Use the `man` pages or just experiment to find out)

1) `man` ***Manual page***

2) `cd` ***Changes the current directory***

3) `ls` ***list the content of the directory*** (try '`ls -tral`')

4) `rm` ***remove delete file or directory***

5) `mkdir` ***Create a directory***

6) `rmdir` ***remove empty directory***

7) `diff` ***Compare two text files & display differences***

8) `echo` ***Display message or value on terminal screen***

9) `printenv` ***Display value of a environment variable***

10) `chmod` ***Changes the file mode bits of each given file***

11) `mv` ***Move a file or directory***

12) `cp` ***Copy a file or directory***

13) `cat` ***Display content of files on screen***

14) `less` ***View content of text file***

15) *more* ***Display the text file, allowing scroll thur***

16) `id` ***Display user & group id for current username***

17) `whoami` ***Display username*** (compare to '`who am i`')

18) `who` ***Display info abt user currently log in***

19) `w` ***Display information about logged in users***

20) `history` ***Display a list of most recent commands***

21) `grep` ***Search for all occurrences***

22) `uname` ***Display current operating systems***

23) `logout` ***Terminate current session & close window***

24) `exit` ***Exit the shell in the terminal*** (what do you see?)

25) `pwd` ***Display currently working directory***

26)	`clear`	***Erase all text and output***
27)	`wc`	***Word count***
28)	`seq`	***Display sequences of number***
29)	`ln`	***Create link between files or directory***
30)	`cal`	***Display a calendar for specified months & years*** (try `'cal 1752'`) Does anything look strange about September? If it was September 2nd, 1752 and I said you have an exam on September 14th, how would you feel about that?

Write down the command and options for doing the following (use `man` to help find answers)

1. List all files, including "hidden" files. To search for `ignore` within the `man` page for `ls`, type the following `'/ignore'` and press return.

    a. ***ls -a***

2. List all files, including their sizes and timestamps.

    a. ***ls -lh***

3. List all files, including their sizes and timestamps sorted so that the newest file is listed last.

    a. ***ls -altr***

4. Delete all files in a directory **and** in all subdirectories of that directory

    a. ***rm -rf path/directory***

5. Copy all files in a directory **and** all subdirectories to a new location:

    a. ***cp -r***

The `pwd` command (Print Working Directory or Present Working Directory) command shows what directory you are currently "in". Use this command and write down your current directory.

/u/muhumed

Make sure you are in your "home" directory (type `cd` and press enter). Typing just `'cd'` followed by return is like Dorothy clicking her heels together and saying "There's no place like home." Use the `pwd` command to see that you are in your "home" directory. This is your **home directory**.

The `mkdir` (make directory) is used to create a new directory. Use this command to create a directory called "cs201" in your home directory.

The `cd` (Change Directory) command is used to change your current directory (`cd cs201`). Use this command to change to your `cs201` directory. Use `pwd` to make sure the `cd` command worked as expected. Create another directory called "`Lab0`" within the `cs201` directory.

What happens when you type `cd` without any parameters?

-**Return to home**

Files have an associated protection (or mode) that limits who can do what with the files. Use the following command to create a file in your `Lab0` directory:

```
echo "stuff" > my.file
```

The > symbol means redirect the output from the previous command (in this case `echo`) into the file name that follows (in this case `my.file`).

Add some more text into `my.file` by using this:

Yes, that is 2 greater than symbols.

```
echo "more stuff" >> my.file
```

The >> symbols means redirect and **append** the output from the previous command (in this case `echo`) into the file name that follows (in this case `my.file`).

Show the contents of the file in your terminal:

```
cat my.file
```

Use the `chmod` command to change the mode of the file so that you have full access, people in your group can read the file, and no one else can do anything with it.

Copy a file from my home directory into your `Lab1` directory. To do this you should enter the command:

```
cp ~rchaney/file.txt .
```

Yes, that is a dot at the end of the command. It is required.

The ~ (a tilde) character is a reference to a home directory, in this case my home directory. If you use the ~ alone, without a user log name following it, it means **your** home directory.

What does the dot at the end of line mean in the above command?

-Current Directory

Create a symbolic link

In your home directory, create a symbolic link to the files under my home directory for the CS201 class:

```
ln -s ~rchaney/Classes/cs201 ./rjchaney
```

This creates a symbolic link called rchaney in your home directory to the files I have under my home directory for the CS201 class.

For example, you can ‘`cd rchaney/Labs`’ to change directory to where files for the programming labs will be located.