Programming for Bioinformatics
BIOL 8803 B
August 17th, 2015

The goal of these exercises is to get you used to working with some of the basic UNIX commands and their options.

Try and think after each command, what did I just do?

We don't have time in class, but read the `man` pages for all of these commands, you may learn something

Today, we're going to get through the basics of these commands/concepts

`ls` – list files in your current directory

`man` - manual

`emacs` – a text editor

`cp` – copy a file or directory

`rm` – remove a file

`mv` – move a file or directory

`cat` >, >> and | - concatenate, redirect, append and pipe

`head`, `tail` and `more` – look at what's in a file

`mkdir` – make directory

`rmdir` – remove directory

`echo` – print something

`exit` – leave a shell or remote connection

`wc` – count the number of words in a file

Instructions for submission:

- Install VirtualBox on your system and add the virtual machine provided in the instruction manual on T-square.
- Please download the ex1.bed file from T-square and place it in ~/class/ex1/ in your VM
- **This exercise has an associated quiz to it.** The quiz can be found on the T-square under **Quizzes** tab.
- **You will be required to submit a brief solution sheet like the one shown in the lecture** in addition to the quiz on T-square under the **Exercises** tab.
- Again – the solution sheet will be brief i.e., **at most** 5 pages long if needed be.

1.) Working with `ls`

a.) List all of the files in `/usr/bin`

b.) Use the `man` command with `ls`

c.) List the size of all of the files in `/usr/bin`

d.) List the permissions of those files

e.) Lit all of the files that start with 'a', what are the . and .. files/directories?

f.) List all files of the type .pl in /usr/bin

2.) Creating and editing a file with `emacs`

a.) Use `emacs` to open a file by the name `abc.txt` for editing

b.) Write something in the file

c.) Save and close the file with `ctrl-x ctrl-s` and `ctrl-x ctrl-c`

3.) Copying and removing a file

a.) Use cp to create a copy of the file you just created

b.)Use `rm` to remove the copy

4.) `cat` and redirecting output

a.) Use `cat` to display the contents of the file you created

b.) Redirect the output to another file using >

c.) `cat` the first file to the second file again, but use >>

d.) Use `cat` to display the contents of the second file

5.) Looking at what’s in a file

a.) Navigate to ~/class/ex1/, Use the `head` and `tail` commands on ex1.bed

b.) Use the `more` command on your ex1.bed

c.) Try using different sizes with the `head` and `tail` commands

d.) Use the `less` command on your ex1.bed

6.) Making and removing directories

a.) Create a directory with `mkdir`

b.) Remove the directory using `rmdir`

c.) Create another directory

d.) Stick a file in the directory using `cp` or `mv`

e.) Try to remove the directory

f.) Remove the directory for real using `rm`

g.) Create the directory structure 1/2/3/4 using a **single** mkdir command.
This is not the right answer: mkdir 1; mkdir 2; mkdir 3; mkdir 4.

7.) Making the prompt prettier

a.) Use `echo` to see the current `PS1` environmental variable

b.) `PS1` is an environmental variable; it tells the command line how to display the prompt

c.) Use `ls` to display all of the hidden files in your home directory

d.) Use `emacs` to open your `.bash_profile` file

e.) Add the line ‘`export PS1="\e[36m\u@\h:\e[32m\W\$ \e[m"`’. What does this line mean?

f.) Save the file and open up a new terminal. Colors!

8.) Count the number of characters

a.) Navigate to ~/class/ex1/

b.) Count the character in the file ex1.bed using `wc`

c.) Count the number of lines in the file ex1.bed using `wc`

9.) Redirecting different streams

a.) Run this command:

```
      perl -e 'foreach(1..100){print $_."\n"; print STDERR ($_ / 2)."\n"}'
```

b.) Redirect only the standard output to out.txt

c.) Redirect only the standard error to err.txt

d.) Redirect standard output to out.txt and standard error to err.txt

e.) Redirect both standard output and standard error to seq.txt

10.) Piping data – Key concept (if you learn one thing here, let it be this)

a.) Run this command:

```
      perl -e 'foreach(1..100){print $_."\n".($_ / 2)."\n"}' > seq.txt
```

b.) Browse the output with the `head`, `tail` and `more` commands.

c.) Display different amounts of data with the `head` and `tail` commands.

d.) Using the `head` and `tail` commands, along with ‘`|`’, get the 50th line of the file.

e.) Print everything but the top 13 lines

f.) Print everything but the last 13 lines

g.) Count the number of characters between lines 45-50 (inclusive)

Additional exercises:

1.) Go through the emacs tutorial. Many of the same controls for editing text in emacs can be used to edit text at the command line, e.g. `ctrl-a` or `ctrl-k.` Emacs is a very powerful editor and knowing it will serve you well. The meta key defaults to escape in Mac OS X, change it under preferences.

2.) Change the colors in your prompt setting to something you like.

3.) Get a webpage up by requesting access through OIT