Programming for Bioinformatics
BIOL 8803 B
August 24th, 2015

Today's commands/exercises will focus on these things:

1.) Gathering information about the system and its users
2.) Managing permissions & processes
3.) Finding command binaries, symbolic links and aliases
4.) Personalizing your environment

It's perfectly fine to write as solutions the commands that you used for executing whatever was asked.

Commands for today (don't worry; a lot of them are easy/closely related):

`chmod` – manage permissions

`who` and `finger` – look up information about users

`top` – look at the system usage

`free` – look at free RAM in the system

`df` and `du` – see how much space on the disk you're taking up

`ps` – look at running processes

`kill` – stop a running process

`fg`, `jobs`, and `&` – background and foreground processes

`which` – find the location of a commands binary

`ln` – create a symbolic link

`alias` – create your own command

`locate` – find a file

`echo` – print something

**Instructions for submission:**

- **This exercise has an associated quiz to it that can be found on the T-square**
- You will be required to submit a brief solution sheet like the one shown in the lecture as well as complete the quiz.
- Again – the solution sheet will be brief i.e., **at most** 5 pages long if needed be.

1.) Changing file permissions

a.) Create a temp file using your favorite editor – emacs, vi, vim, cat, whatever

b.) List the permissions of the file you just created

c.) Use `chmod` to make the file unwriteable

d.) Try to remove the file

e.) Make the file writeable again

f.) Change the permission of the file to the following now:

i) read and write for all users

ii) read + write for the owner and readable for everyone else

iii) only readable by the owner

g.) Change the owner to www-data (requires `sudo`) and see if you can still read the file using

h.) Create a directory, remove its executable permissions. Try opening the directory

2.) `who` and `finger`

a.) Use `who` to see yourself logged into the machine.

b.) Find all the logins on the machine.

c.) Use `finger` to find information about yourself on the Mac or your Linux VM or whatever you have.

3.) Monitoring system usage with `top` and `free`

a.) Use `top` to look at what's running on the computer.

b.) Find the CPU and memory usage of a program; how much memory is left?

c.) Order the processes by CPU, ascending and descending. Do the same for memory. Note for future: `top` varies quite a bit by system.

d.) If your system has the `free` command, use it to see how much memory is being used. What do the numbers mean?

4.) Looking at hard disk space

a.) Use `df` to find the free space on your machines hard disk.

b.) Use `du` to find the total size of some directory.  Make the output easy to read, e.g. '`315M`'

5.) Finding running processes

a.) Find all of the processes that you have running using `ps`.

b.) Find ALL processes that are running on the machine.

c.) Find all of the processes being run by the user '`root`' in user format

6.) Murder most foul

a.) Kill your login shell with the `kill` command.

b.) Open a new shell.

7.) Managing jobs

a.) Start editing a file with `emacs`, then put it in the background using `ctrl-z`.

b.) Find it using the `jobs` command.

c.) Restore it to the foreground using `fg`.

d.) Start a new job in the background using '`&`' after the command

e.) Move the new job to the foreground using `fg`.

8.) Finding the location of binaries, linking them and alias

a.) Find out where the `awk` command is located using `which`.  Find out if it is a link, or a real binary.

Follow this through till you have found the real binary.  There might be multiple iterations that you have to follow this through.

b.) If it is a link, where is the real file?

c.) Create a symbolic link to the `ls` command in your home directory, using the `ln` command, then execute the created link (`./<whatever you called it>`)

d.) Create an alias for '`ls -a`'.

9.) Hard vs Symbolic link

a.) Create a file abc.txt, put some random text in it (doesn't matter what, just remember bits and pieces of what you wrote there)

b.) Now create a symbolic and a hard link for abc.txt, name it sym.txt and hard.txt respectively

c.) Create another file def.txt, put some other random text in it.

d.) Now delete abc.txt.   Which of the two links still work?

e.) Rename def.txt to abc.txt.  Which of the two links still work? If both works, is there a difference between the two? If only one works, why is it so?