Programming for Bioinformatics I BIOL 7200

Week 1 Exercise

August 24, 2021

The goal of these exercises is to get you used to working with some basic UNIX commands and their options. Try and think about what each command does, and where it can be used. Reading the man page for each of these is **recommended**.

We're going to cover the following commands:

Command	Explanation
ls	List files in your current directory
man	Manual entry for a command
vi	A text editor
cp	Copy a file or directory
rm	Remove a file
mv	Move a file or directory
cat, >, >>, |	Concatenate, redirect, append, and pipe
head, tail, more, less	Exploring the contents of a file
mkdir	Make directory
rmdir	Remove directory
echo	Print something
export	Set an environmental variable
wc	Count the number of lines/words/bytes in a file

Instructions for submission

- Run Linux on your system. This can be done as a virtual machine as described **here**
- Download *ex1.bed* from Canvas and place it in ~/class/ex1/ on your Linux system.
- Prepare two solution sheets for this exercise - one for submission and another for your own reference. An assignment will be created on Canvas called "Exercise-1" **Only submit the submission sheet on Canvas**. The solution sheet you create for your reference should help you going forward - make it as detailed/brief as you'd like for your own learning style. For the submission sheet, copy the question and write the correct answers below the question.
- Name your submission sheet: gtusername.txt, gtusername.pdf, or gtusername.docx

Grading Rubric

This assignment will be graded out of 100.

- 20 points for correctly naming and submitting your sheets (20)
- Questions 1-4 carry 5 points each (20)
- Questions 5-10 carry 10 points each (60)

Exercises

1. Working with `ls` in the home directory
    1. List all the files in `/usr/bin`
    2. Use the `man` command with `ls`
    3. List the size and permissions of **all** the files in `/usr/bin`
    4. List the files that start with ‘a’ in `/usr/bin`
    5. What are the `.` and `..` files/directories?
    6. List all the file with extension `.py` in `/usr/bin`

2. Creating and editing with `vi`
    1. Use `vi` to open a file called `file1.txt` for editing
    2. Write `This is some text` in the file
    3. Save and close the file with Esc `:wq` Enter/Return

3. Copying and removing files
    1. Use `cp` to copy `file1.txt` to `file1_copy.txt`
    2. Use `rm` to remove `file1.txt`
    3. Use `ls` to see the contents of the current directory

4. Using `cat` and redirecting output
    1. Use `cat` to display the contents of `file1_copy.txt`
    2. Redirect the output of the `cat` command to another file called `file1_contents.txt` using `>`
    3. Use `cat` to display the contents of `file1_contents.txt`
    4. Redirect the contents of `file1_copy.txt` once again into `file1_contents.txt` but use `>>` instead
    5. Use `cat` to display the contents of `file1_contents.txt`
    6. Repeat parts *2.* and *3.* How is the output different?

5. Looking at what’s in a file
    1. Navigate to `~/class/ex1`

    2. Use `head` and `tail` to see the first and last 10 lines of the file, respectively
    3. Use the `head` command to see the top 50 lines of the file
    4. Use the `tail` command to see the last 25 lines of the file
    5. Use `more` on `ex1.bed`
    6. Use `less` on `ex1.bed`

6. Making and removing directories
    1. Create a directory called `myDir` using `mkdir`
    2. Remove the directory using `rmdir`
    3. Repeat step *1.*
    4. Copy `file1_contents.txt` in `myDir`
    5. Try to remove the directory using `rmdir`
    6. Remove the directory using `rm`
    7. Create the following directory structure using a single `mkdir` command. Do not use semicolon `;` , ampersand `&` or other command connectors for this purpose
       dir1
       l---- dir2
       hint: dir2 is inside dir1

7. Making the prompt prettier
    1. Use `echo` to see the current `PS1` environmental variable; Look up what the `PS1` variable does
    2. Use `ls` to display all the hidden files in your home directory
    3. Use `emacs`/`vi` to open your `.bash_profile`
    4. Add the following line to it:
       ```
       export PS1="\[\033[38;5;10m\]\u\[$(tput sgr0)\]@\[$(tput sgr0)\]\[\033[38;5;13m\]\h\[$(tput sgr0)\]:\[$(tput sgr0)\]\[\033[38;5;14m\]\w\[$(tput sgr0)\]\\$ \[$(tput sgr0)\]"
       ```
    5. Make sure you spaces & new lines don’t get interchanged. There is no new line in the above command.
    6. Save the file and exit. Start a new terminal. Colors!

8. Counting characters
    1. Navigate to `~/class/ex1`
    2. Count the number of characters in `ex1.bed` using `wc`
    3. Count the number of lines in `ex1.bed` using `wc`

9. Redirecting different streams
    1. Run this command:
       ```
       perl -e 'foreach(1..100){print $_."\n"; print STDERR ($_ / 2)."\n"}'
       ```

2. Run the command and redirect only the standard out to `myOut.txt`
3. Run the command and redirect the standard error to `myErr.txt`
4. Run the command and redirect the standard output to `myOut.txt` and standard error to `myErr.txt`
5. Run the command and redirect both the standard output and standard error to `mySeq.txt` in a single command

10. Piping data
    1. Run this command:
       `seq 0 .5 100 > longSeq.txt`
    2. Using the pipe with `head` and `tail`, get the 50th line of `longSeq.txt`
    3. Print everything but the top 13 lines
    4. Print everything but the last 13 lines
    5. Count the number of characters in lines 45-50 (inclusive)