

Thursday, January 27: In class assignment/homework

1. Set-up
 - a. We have a shared directory for the course.
 - i. Its path is **/mnt/ceph/ajones_csb**
 - ii. You will put your finished scripts in a folder there so we can grade them.
 - b. Create a symbolic link to the **ajones_csb** folder.
 - i. Make sure you are in your CSB directory (**cd ~/CSB**).
 - ii. Type **ln -s /mnt/ceph/ajones_csb**
 1. The first letter of this command is lower-case L.
 2. This creates a symbolic link, which is basically a shortcut, so you don't have to type the full path to access the shared folder.
 - c. Create a new directory in **ajones_csb**, with the name yourlastname_firstinitial.
 - i. For me, I would type **cd ajones_csb** (to cd to the shared directory)
 - ii. Then I would type **mkdir jones_a**
 - iii. Type **ls** to make sure that the directory is present in **ajones_csb**
 - iv. cd into this new directory and make another new directory named **scripts**
2. Where to go for help
 - a. The beginning of this tutorial is good: <https://linuxconfig.org/bash-scripting-tutorial-for-beginners>
 - b. Also check out: <https://ryantutorials.net/bash-scripting-tutorial/>.
 - c. Use Google to search the specific thing you want to do, such as "how to use a for loop in bash" or "variables in bash", etc.
3. Challenges
 - a. Challenge 1:
 - i. Write a script named **hello_world.sh** that outputs "Hello World" to the screen.
 - b. Challenge 2:
 - i. Write a script that accepts a name as an argument, and prints "Hello, Name", where the Name is replaced by the argument.
 - ii. Name this script **hello_there.sh**.
 - iii. So, for instance, if I run it like this: **hello_there.sh Frodo**, it should print "Hello, Frodo" to the screen.
 - c. Challenge 3:
 - i. Modify the script from Tuesday's class to split a fastq file into individual reads.
 - ii. Name this script **split_fastq_file.sh**.
 - iii. Remember that each fastq record is four lines: a header, the sequence, another header, and the quality scores for the sequence.
 - iv. You can test this script on "fastq_example.fastq" from Tuesday.
 - d. Challenge 4:
 - i. Modify the script from Tuesday's class to add ".fastq" to the files that are output by the script you wrote in Challenge 3.
 - ii. Name this script **add_fastq_extension.sh**.
 - e. Write a script that automatically creates a directory structure with readme files in each directory.

- i. The top directory should be named `rna_seq`.
 - ii. Within `rna_seq`, there should be directories named `genome`, `reads`, and `trimmed_reads`.
 - iii. Within `genome`, there should be directories named `CDS`, `whole_genome`, and `blast_databases`.
 - iv. Every folder should have a file named `readme.md`, which can be empty or contain some text – it's up to you.
 - v. Name this script **`create_directory_hierarchy.sh`**.
- f. Write a script that counts the number of records in a fasta file and outputs it to the screen.
 - i. Name this script **`count_fasta_records.sh`**
 - ii. Output: "This file contains XX fasta records."
- g. Write a script that converts a tab delimited text file into a comma-delimited file.
 - i. Name this script **`convert_tab_to_csv.sh`**
 - ii. Download the files in the folder "`newt_files.tar.gz`" from Canvas and use these to test the next several scripts. It is compressed, so you will have to uncompress it.
- h. Write a script that converts a tab delimited text file into comma-delimited format and renames it to have the extension "`.csv`".
 - i. You can just add the "`.csv`" to the end of each file.
 - ii. (Optional) If you want a challenge, replace the "`.txt`" with "`.csv`".
 - iii. Name this script **`convert_and_rename_tab_to_csv.sh`**
- i. Write a script that converts a folder of tab-delimited files into comma-delimited and renames them all to have a "`.csv`" extension.
 - i. In other words, do the same thing the previous script did but do it for all of the appropriate files in a folder.
 - ii. Name this script **`convert_all_tab_to_csv.sh`**
- j. Move all of your working scripts to the "`scripts`" folder you created in part 1.
 - i. The names must be **exact**.
 - ii. Do not upload scripts that do not function. That is, test your scripts before you upload them.
 - iii. Not all scripts will be tested and graded, but of those we test, a non-functional script will result in a 0.