### Exercises

We start our shell scripting discussion this week. Most of the concepts in this exercise will be covered in the class. There are a couple of additional commands that are used in this exercise but will not be discussed in the class. Part of the homework will be understanding what these commands are and how to use them.

#### sort and uniq

1. More piping with sort and uniq (This is a repeat from a previous assignment, just a little refresher)
   1. Create the following file:

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

Text

Description automatically generated

* 1. sort the file numerically and output it to another file without using > and count the number of lines

**ANS: sort -n uniq.txt|tee result.txt|wc -l**

**The number of lines are 200.**

* 1. Pipe the result of the sort to uniq. What happened? How many lines are there?

ANS: **sort -n uniq.txt|uniq|wc -l**

**The number of lines are now 150 because uniq got rid of duplicate values in its viscinity.**

* 1. Pipe the result of the sort to uniq and discard all lines that appear more than once. I.e., I don’t want the lines (e.g., 1, 2, etc.) which occur more than once.

**ANS: sort -n uniq.txt|uniq -u**

* 1. Pipe the result of the sort to uniq and count the number of times each number appear

**ANS: sort -n uniq.txt|uniq -c**

**This ouputs two columns. The first column is the count of the number of times each entry occurs and the second is the entry itself.**

**Text

Description automatically generated**

#### Basic Shell scripting

1. Write shell scripts for the following:
   1. Write a shell loop that adds up all the numbers from 1-100

**ANS:**

Text

Description automatically generated

Text

Description automatically generated

* 1. Write a shell loop that prints each letter in the word Hello separately. This can be done using a for loop iterating through the letters H e l l o. This is a 5 line script; don’t make it harder than that.

ANS:

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

* 1. Have variable x increase from 0 to 100 and variable y decrease from 100 to 0 by 1 unit at a time and print "x and y are equal" when the variables are equal

ANS:

Text

Description automatically generated

Text

Description automatically generated

* 1. Repeat the above (c) but this time print the difference between x and y ($x-$y) when they are not equal and print "x and y are equal” when they are equal

ANS:

Text

Description automatically generated

Table

Description automatically generated

* 1. Write a getopts block with 3 options including one that requires an argument

ANS:

Graphical user interface, text, email

Description automatically generated

Text

Description automatically generated

* 1. Write a script utilizing case that asks the user the day of the week and prints the day number based on user input

Monday = 1; Tuesday = 2; ...; Sunday = 7

Example terminal output (lines that start with > are output from the script, lines that start with $ are user input):

./scriptName.sh

> Please enter a day:

$ Monday

> Monday is day number 1

#### ANS:

#### Text Description automatically generated

#### Text Description automatically generated

#### Mini challenge part 1

1. We are going to do an exercise for fun, but it will enforce shell concepts and will help you prepare for this week. To make things easier for you, I have provided you with stepwise instructions on how to proceed. You will add one element each time, making the loop a little more complicated as compared with the step before.
   1. Write a for loop to create 10 files by the name seq1.fasta, seq2.fasta, seq3.fasta, and so on. You can create a file using the touch command.
   2. Modify this loop to now do two more things:
      1. Delete the seq1.fasta, seq2.fasta, seq3.fasta, etc. if they exist
      2. Create new seq1.fasta, seq2.fasta, seq3.fasta, etc. files with a FASTA description line in it. The FASTA description lines needs to be like this: >seq1 for seq1.fasta, >seq2 for seq2.fasta, >seq3 for seq3.fasta, etc.
   3. Add another element to this loop: the loop now also adds a random DNA sequence along with the FASTA description line. The following command will give you a random DNA string of 50x10 letters: cat /dev/urandom | tr -dc 'ACGT' | fold -w 50 | head

**ANS:**

**#!/bin/bash**

**for ((i=1;i<11;i++));**

**do**

**touch seq$i.fasta**

**if [ $i -le '10' ]; then**

**rm -r seq$i.fasta**

**fi**

**echo ">seq$i" > seq$i.fasta**

**cat /dev/urandom | tr -dc 'ACGT' | fold -w 50 | head >> seq$i.fasta**

**done**

Text

Description automatically generated

#### Mini challenge part 2: Let’s try nesting the loops

1. Nesting loops is really having a loop within a loop. Instead of creating 10 single sequence files, as in part 1, we will create 10 multiple sequence FASTA files (aka, 10 multi-FASTA files) with 8 sequences in each.

The FASTA descriptors for file 1 will go like this: >seq1\_1, >seq1\_2, >seq1\_3, …; the FASTA descriptor for file2 will go like this: >seq2\_1, >seq2\_2, >seq2\_3, … ; etc.

**#!/bin/bash**

**for ((i=1;i<11;i++));**

**do**

**touch seq$i.fasta**

**if [ $i -le '10' ]; then**

**rm -r seq$i.fasta**

**fi**

**for ((j=1;j<9;j++));**

**do**

**echo ">seq${i}\_${j}" >> seq$i.fasta**

**cat /dev/urandom | tr -dc 'ACGT' | fold -w 50 | head >> seq$i.fasta**

**done**

**done**

Graphical user interface, text

Description automatically generated

#### Mini challenge part 3: Now add getopts to the beginning of the script

1. For the final part of this series of questions, I want you to add getopts to the script so that it takes take three options:

- Option n => will take a number as an input. This argument describes the number of files to be created (which until now was 10)

- Option m => will take a number as an input. This argument describes the number of sequence to be added in each file (which until now was 8)

- Option v => verbose mode, i.e., I want the script to print out every single action it is doing (e.g., what file is the script currently working on? what sequence number?). This is a flag, so no input is expected with it.

* Again, you should submit a script for the 3rd Mini Challenge as part of your submission on Canvas.

**ANS: Uploaded agogate8.sh on canvas**

Graphical user interface, text

Description automatically generated