# Homework 1 - Linux Refresh

## Part I: Questions

1. What is a Graphical User Interface (GUI) and how does that differ from the Command Line Interface (CLI)?

**GUI stands for graphical user interface. This is the most common way people interact with their computers. It includes pull down menus and clicking to get things done. CLI stands for computer line interface. This involves using command lines to communicate with your computer.**

1. What does the shell do?

**The shell is a program that runs other programs. The shell is a program where users can type commands.**

1. What is the command used for listing things in a directory?

**ls**

1. What is the command used for changing directories?

**cd**

1. What command would you use to get your current working directory?

**pwd**

1. How do you get the manual for these commands?

**man**

1. What does the shell prompt look like?

**Usually, the shell prompt starts with a $**

1. How would you list things in a directory in chronological order?

**ls -r (oldest first) or ls -t (most recent first)**

1. Name two ways to get to the ‘home’ or ‘root’ directory.

**cd and cd ~**

1. What is the difference between an absolute path and a relative path?

**An absolute path contains the root element and the complete directory list to the location or file in which you are trying to get. A relative path changes as you move around and uses . and .. to get to where you’re trying to go.**

1. What are the two relative path directories we talked about and what do they mean?

**. and .. are the two relative path directories. The . means the current directory and .. means one directory up.**

1. Why shouldn’t you put spaces in filenames?

**The computer may confuse the space for meaning an argument or something else that is not a file.**

1. Name a way to have multiple readable words in a filename without spaces?

**Typing\_like\_this (underscores).**

1. What is the program nano, what does it do? Do you use nano? If not what do you use?

**Nano is a text editor that is built into linux. Yes, I used nano last semester.**

1. ``\*`` is a form of a wildcard. What does it mean?

**It essentially means ‘anything.’ For example, \*.txt means ‘anything that ends with .txt.’**

1. The following code would match what?

```ls \*.txt```

**Anything that ends in .txt**

1. Why do we need to be careful with the ```mv``` command?

**Because the mv command moves files to a different location or can save files under a different name.**

1. Why do we need to be careful with the ```rm``` command?

**The rm command will delete files.**

1. What is the difference between ```>``` and ```>>```?

**The >> will look for a named folder first, and the > will not.**

1. What does ```head``` do?

**Lists the first 10 lines in a file.**

1. What is the purpose of ```|``` (pipe)?

**The pipe essentially lists an order of actions. It takes the output of what’s written before it and inserts it as the input for what comes after.**

1. Interpret the following command:

```cat huge\_file.fasta | uniq | head -n 5 >proteins.fasta```

**Concatenate file, omit repeating lines, and place the first 5 lines into a file named proteins.fasta**

1. What is a loop and when would you use it?

**A loop performs a repeat of code a set number of times until a certain task you want to accomplish is complete. Loops are useful for working with high numbers of files or when working with slower programming languages.**

1. In a directory with the following files. What would the following loop do?

```apple.txt, banana.txt, orange.txt

$ for filename in \*.txt; do

> cat ${filename} >>fruits.txt

> done

**For every file ending in .txt, concatenate them into a file named fruits.txt, done. The >> prevents the loop from overwriting the folder because it will look for a folder by that name first.**

1. What is a shell script?

**A shell script is** **a text file that contains a sequence of commands for a UNIX-based operating system**.

1. What are the benefits of writing and running a script over typing the code in?

**You can save the code for use later on.**

1. When looking at a script what does ```#``` mean? Why would you use one?

**# starts a bash script. It can also permit for commenting and writing in noncoding elements.**

1. What does ```grep``` stand for and what does it do?

**grep stands for global regular expression print. It searches a file for a particular pattern of characters, and displays all lines that contain that pattern.**

1. What does ```find``` do?

**find can find any letters, numbers, or other characters in a document, web page, or file that you need.**

## Part II - Writing Code

As part of this exercise please open the shell and practice moving around into different files. Best practice is to try a little bit every day. Try to challenge yourself by not using the mouse. For each of the questions below type your code and the result from the shell prompt.

1. Open the shell and change directories to the desktop and list the contents, paste your code here.  
   **(base) ➜ ~ cd Desktop**

**(base) ➜ Desktop ls**

1. Draw the file structure from the root directory into one of the folders on the desktop. You can submit a pdf with a drawing. Select three locations and type the commands for changing directories to those locations use a combination of absolute and relative paths.

Users

ariellejewelpastore

Desktop

hw2

hw

misc

Waiver.pdf

Lease.pdf

**cd /Users/ariellejewelpastore/Desktop/hw2**

**cd ../misc**

**cd ../hw**

1. On your computer create a directory on your desktop. Create two files without opening them that have file extensions ```.txt```. Then use wildcards to list the files in that directory. Paste you code here.  
   **(base) ➜ Desktop mkdir hw**

**(base) ➜ Desktop cd hw**

**(base) ➜ hw touch first.txt**

**(base) ➜ hw touch second.txt**

**(base) ➜ hw ls \*.txt**

**first.txt second.txt**

1. Download the file [Hutia\_DNA.fasta](https://www.dropbox.com/s/07y610c8zt63nt5/Hutia\_DNA.fasta?dl=0) from the Github repo. Make a new directory on the desktop. Move the file into that directory. List the contents of the directory. How many lines are in this file? Fasta files are a type of data file that holds DNA sequences. They are all formatted the same. The first line starts with a > and a name followed by a line break, and then DNA sequence followed by a line break. The next line starts with an > and another name followed by a line break and the DNA sequence. With that in mind, how could you use linux commands to determine how many sequences are in this file? How many sequences are in this file?

**(base) ➜ hw2 wc -l Hutia\_DNA.fasta**

**2382004 Hutia\_DNA.fasta**

**(base) ➜ hw2 grep -c ">" Hutia\_DNA.fasta**

**1191002**

**There are 2,382,004 lines in this file. You can use ‘grep’ with the argument -c using the pattern “>” to determine the number of sequences. There are 1,191,002 sequences.**

1. Build a single line of code that would take the first 100 sequences of this file and put them in a new file called ‘MyOutputFile.fasta’. Use a combination of wc, sort and uniq in a single line of code to tell me how many unique lines are in this file. Paste your commands and the answer here.  
   **(base) ➜ hw2 head -n 100 Hutia\_DNA.fasta > MyOutputFile.fasta**

**(base) ➜ hw2 sort MyOutputFile.fasta | uniq | wc -l**

**100**

**There are 100 unique files.**

1. Write code that would create a loop to copy all the files in one directory ending in ```.txt``` to another directory. Paste the code here.   
   **for i in ~/Desktop/\*.txt,**

**do cp $i ~/Desktop/bio792 >> output.txt**

**done**

1. Write a bash script with in-line documentation (hint #) to show how to run a made up python program (script) called 'find\_taxa.py' on set of files ending in ‘.fasta’. Show in the script how to run the program on each of those files. What is the name of your script? Type the script here.

**nano find\_taxa.py**

**#!/usr/bin/python3**

**#############################################################**

**This was created by Arielle Pastore.**

**It will read all fasta files.**

**Usage: python3 find\_taxa.py**

**#############################################################**

**#For loop that will cat fasta files**

**for i in \*.fasta**

**do cat $i >> total\_fasta**

**done**

1. From the Hutia\_DNA.fasta file tell me how on how many lines do we find the pattern ‘GAGA’. What was the code used to find this?

**304,806 lines. I used the code:**

**(base) ➜ hw2 grep -c "GAGA" Hutia\_DNA.fasta**