1. What is a Graphical User Interface (GUI) and how does that differ from the Command Line Interface (CLI)?
   * GUI is a visual way to navigate ones computer and is the most common way people use their computers.always user to do repetitive tasks quicker and with less mistakes.lamguage based. CLI is the computers programs that allows the used to run programs and navigate through files.
2. What does the shell do?
   * It is a program that enables users to type commands and move through a computers files and directores. There are different kinds of shells (bash, sh, etc.). It requires the user to learn the vocab of the language.
3. What is the command used for listing things in a directory?
   * ls
4. What is the command used for changing directories?
   * cd
5. What command would you use to get your current working directory?
   * pwd
6. How do you get the manual for these commands?
   * Man command
7. What does the shell prompt look like?
   * %
8. How would you list things in a directory in chronological order?
9. Name two ways to get to the ‘home’ or ‘root’ directory.
   * cd
   * ~
10. What is the difference between an absolute path and a relative path?
    * A relative path is relative to what file/directory user is currently in
    * An absolute path specifies a location from the root of the file system.
11. What are the two relative path directories we talked about and what do they mean?
12. Why shouldn’t you put spaces in filenames?
    * When including filenames with a space in an argument the shell would read the single filename as two arguments
13. Name a way to have multiple readable words in a filename without spaces?
    * \_
    * .
    * -
    * Camel case
    * Use quotations around file name that has spaces
14. What is the program nano, what does it do? Do you use nano? If not what do you use?
    * Nano is a text editor.
15. \* is a form of a wildcard. What does it mean?
    * Matches 0 or more characters
16. The following code would match what? ls \*.txt
    * All files ending in .txt in working directory
17. Why do we need to be careful with the mv command?
    * You can overwrite files
18. Why do we need to be careful with the rm command?
    * Removing is forever. There is no trash bin to rm files out of luck if you want them back
19. What is the difference between > and >>?
    * >  tells the shell to **redirect** the command’s output to a file instead of printing it to the screen.
    * >> Send output of a command to a file, appending to existing files versus overwriting (potentially losing info) when using >
20. What does head do?
    * Display the first few lines of text of a file can specify how many lines using -n 1 (one line) -n 20 (20 lines)
21. What is the purpose of | (pipe)?
    * tells the shell that we want to use the output of the command on the left as the input to the command on the right.
22. Interpret the following command:

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

stream content of file in terminal showing unique vales and select the first five and save to new file

1. What is a loop and when would you use it?
   * Mini program that allow user to repeat commands for each item in a list. Automates repetitive work thus time saver and reduces potential mistakes.
2. In a directory with the following files. What would the following loop do?

$ for filename in \*.txt; do

* + cat ${filename} >>fruits.txt
  + done
    1. this would concatenate the file content that is a .txt file and append it to a file named fruits.txt

1. What is a shell script?
   * A small program.
   * Commands saved in a file
   * Executable in shell
2. What are the benefits of writing and running a script over typing the code in?
   * Saves times
   * Less likely to make mistakes
3. When looking at a script what does # mean? Why would you use one?
   * # is how we comment in a script. Compute ignores it. Allows author to explain what script is doing to others and future self
4. What does grep stand for and what does it do?
   * ‘grep’ is a contraction of ‘global/regular expression/print’
   * finds and prints lines in files that match a pattern.
   * It is case sensitive
   * Does not have to form the complete word ‘the’ would include ‘thesis’
     1. -w for word boundaries
   * For a phrase enclose phrase in double quotations
5. What does find do?
   * Finds files in directories
   * Combined with grep often
   * $( ) shell starts here and then takes commands out side parenthetical
6. Open the shell and change directories to the desktop and list the contents, paste your code here.

(base) erickarkay@ERICKAs-MBP ~ % cd Desktop

(base) erickarkay@ERICKAs-MBP Desktop % ls

Biol792

EECB\_Fall\_2021.pdf

GRAD778\_04\_2020

Julies\_Class

R bootcamp

Relocated Items.nosync

Remind me how pollinatorers decipher the amount of nectar present in a flower.docx

Screen Shot 2021-01-24 at 2.54.04 PM.png

Screen Shot 2021-08-20 at 1.45.19 PM.png

Screen Shot 2021-08-26 at 3.17.18 PM.png

Screen Shot 2021-09-13 at 5.06.23 PM.png

Screen Shot 2021-09-14 at 1.51.54 PM.png

Screen Shot 2021-10-08 at 7.53.59 AM.png

Screen Shot 2021-10-18 at 5.19.00 PM.png

Screen Shot 2021-10-18 at 8.01.07 AM.png

Screen Shot 2021-10-19 at 8.30.58 AM.png

Screen Shot 2021-10-21 at 4.58.21 PM.png

Screen Shot 2021-10-21 at 5.05.46 PM.png

Screen Shot 2021-10-24 at 5.24.29 PM.png

Screen Shot 2021-10-24 at 5.30.03 PM.png

Screen Shot 2021-10-24 at 5.30.29 PM.png

Screen Shot 2021-10-24 at 5.37.51 PM.png

Screen Shot 2021-10-24 at 5.38.04 PM.png

Screen Shot 2021-10-29 at 9.06.22 AM.png

Screen Shot 2021-11-01 at 3.24.10 PM.png

Screen Shot 2021-11-12 at 7.01.35 AM.png

Screen Shot 2021-12-04 at 12.16.26 PM.png

Screen Shot 2021-12-07 at 11.43.01 AM.png

Screen Shot 2021-12-07 at 11.49.27 AM.png

Screen Shot 2021-12-07 at 11.49.32 AM.png

Screen Shot 2021-12-07 at 11.56.20 AM.png

Screen Shot 2021-12-07 at 11.59.16 AM.png

Screen Shot 2021-12-07 at 11.59.30 AM.png

Screen Shot 2021-12-07 at 12.11.55 PM.png

Screen Shot 2021-12-07 at 12.13.26 PM.png

Screen Shot 2021-12-07 at 12.14.16 PM.png

Screen Shot 2021-12-07 at 12.17.48 PM.png

Screen Shot 2021-12-07 at 12.19.05 PM.png

Screen Shot 2021-12-07 at 12.21.12 PM.png

Screen Shot 2021-12-07 at 12.24.50 PM.png

Screen Shot 2021-12-07 at 12.27.06 PM.png

Screen Shot 2021-12-07 at 12.34.16 PM.png

Screen Shot 2021-12-07 at 9.06.43 AM.png

Screen Shot 2021-12-07 at 9.08.33 AM.png

Screen Shot 2021-12-07 at 9.21.58 AM.png

Screen Shot 2021-12-07 at 9.24.54 AM.png

Screen Shot 2021-12-20 at 12.10.48 PM.png

Screen Shot 2022-01-17 at 4.50.36 PM.png

Screen Shot 2022-01-20 at 3.40.54 PM.png

Shared copy

Tania Israel form .docx

books

dirsource

dissertation

golden girls.pdf

house:dog:chores

mandolin

manuscript reviews

1. Text, letter

   Description automatically generatedDraw 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.
2. 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) erickarkay@ERICKAs-MBP Desktop % mkdir Julie\_HW\_wk1

(base) erickarkay@ERICKAs-MBP Desktop % cd Julie\_HW\_wk1

(base) erickarkay@ERICKAs-MBP Julie\_HW\_wk1 % touch example1.txt

(base) erickarkay@ERICKAs-MBP Julie\_HW\_wk1 % touch example2.txt

Find . -name “\*.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) erickarkay@ERICKAs-MBP Julie\_HW\_wk1 %

cp /Users/erickarkay/Downloads/Hutia\_DNA.fasta html .

(base) erickarkay@ERICKAs-MBP Julie\_HW\_wk1 % ls

(base) erickarkay@ERICKAs-MBP Julie\_HW\_wk1 %

wc -l $(find . -name "\*.fasta")

(base) erickarkay@ERICKAs-MBP Julie\_HW\_wk1 %

grep -c ">" $(find . -name "\*.fasta")

1191002 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) erickarkay@ERICKAs-MBP Julie\_HW\_wk1 % grep -v ">" Hutia\_DNA.fasta > MyOutputFile.fasta

(base) erickarkay@ERICKAs-MBP Julie\_HW\_wk1 % less MyOutputFile.fasta

(base) erickarkay@ERICKAs-MBP Julie\_HW\_wk1 % head -n 100 MyOutputFile.fasta |sort |uniq | wc -l

99 unique lines

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.

Usage: (base) erickarkay@ERICKAs-MBP Julie\_HW\_wk1 % bash /Users/erickarkay/Desktop/Julie\_HW\_wk1 grep\_txt.sh Julie\_HW\_wk1 NewDir

Bash.sh:

mkdir -v "$2"

for file in "$1"

do

find -name "\*.txt" "$1" | cp $file >> "$2"

echo "$file"

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.

script name is taxa\_list.py

script:

#!/usr/bin/env python

##############################

#author: Ericka Kay

#usage bash 'find\_taxa.py' list of files to run program on (I did not specify only \*.fasta files though)

#usage example: bash taxa\_list.py \*.fasta | wc-l

#wc -l will output number of files processed instead of names

for datafile in "$@"

do

echo $datafile

bash find\_taxa.py $datafile taxa-$datafile

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?

bash-3.2$ grep -c "GAGA" Hutia\_DNA.fasta

304806

Julie’s answer to #7 in class

#usage: sh run\_find\_taxa.sh

#sends files in list (\*.fasta) to run the

For file \*.fasta

Do

Python3 find\_taxa.py $file

Done