**CSc 3320: Systems Programming**

Fall 2021

Homework

# 2: Total points 100

Submission instructions:

1. Create a Google doc for each homework assignment submission.
2. Start your responses from page 2 of the document and copy these instructions on page 1.
3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing in your document TWO POINTS WILL BE DEDUCTED per submission.
4. Keep this page 1 intact on all your submissions. If this *submissions instructions* page is missing in your submission TWO POINTS WILL BE DEDUCTED per submission.
5. Each homework will typically have 2-3 PARTS, where each PART focuses on specific topic(s).
6. Start your responses to each PART on a new page.
7. If you are being asked to write code copy the code into a separate txt file and submit that as well.
8. If you are being asked to test code or run specific commands or scripts, provide the evidence of your outputs through a screenshot and copy the same into the document.
9. Upon completion, download a .PDF version of the document and submit the same.

Full Name: Adrian Rivas

Campus ID: arivas7

Panther #: 002-39-5009

**PART 1 (2.5 points each): 10pts**

1. What are the differences among ***grep****,* ***egrep*** *and* ***fgrep***? Describe using an example.

When using Grep you can use it to search for a specific term on any files you are currently connected too and a example would be how we did it on lab four we were searching for specific stuff in the file and to make it easier we used “grep CSC temp\_course.txt”

Egrep would be the same but the difference would be that egrep searches for a specified file line by line. An example “egrep ‘CSC 3|CSC 1’ temp\_course.txt”

Fgrep on the other hand is used to search for a fixed-character strings in a file. An example “fgrep “CSC.3” para

1. Which utility can be used to compress and decompress files? And how to compress multiple files into a single file? Please provide one example for it.

The utility that is used to compress and decompress files would be Tar. An example would be tar -czvf filename.gz

1. Which utility (or utilities) can break a line into multiple fields by defining a separator? What is the default separator? How to define a separator manually in the command line? Please provide one example for defining the separator for each utility.

The utility would be Awk. ‘\t’ and it is defined by a filed separator by typing “-F” and then to switch under the command line with “FS=”. An example would be awk ‘{print $3}’ FS=’\t’.

1. What does the ***sort*** command do? What are the different possible fields? Explain using an example.

Sort command sorts the contents of the file in alphabetic or numeric order.

Example $ cat > mix.txt

Abc

Apple

BALL

Abc

bat

$ sort mix.txt

Abs

BALL

Abc

apple

bat

**Part IIa (5 points each): 25pts**

1. What is the output of the following sequence of bash commands: **echo 'Hello World' | sed 's/$/!!!/g'**

**Hello World!!!**

1. What is the output for each of these awk script commands?

-- 1 <= NF { print $5 } It would print out all of the values in the 5th column

-- NR >= 1 && NR >= 5  { print $1 } We are printing the first column by using NR that displays the line number that is greater than 5.

-- 1,5 { print $0 } Everything in the file would be printed.

-- {print $1 } Prints out all the values in the first column.

1. What is the output of the following command line:

**echo good | sed** **'/Good/d'**

good

1. Which **awk** script outputs all the lines where a plus sign + appears at the end of line?

/\+${print $0}

1. What is the command to delete only the first 5 lines in a file "foo"? Which command deletes only the last 5 lines?

sed ‘1,5d’ file

**Part IIb (10pts each): 50pts**

 Describe the function (5pts) and output (5pts) of the following commands.

**9.**     **$ cat  float**

Wish I was floating in blue across the sky, my imagination is strong, And I often visit the days

When everything seemed so clear.

Now I wonder what I'm doing here at all...

**$ cat h1.awk**

**NR>2 && NR<4{print  NR  ":" $0**

**$ awk    '/.\*ing/ {print NR  ":"  $1}'  float**

We use the NR command to look for row 3 in the file, and then it will print the row number 3 with this “:”. Then the following command searches for the “ing” in the file then the row number 1 with “:” and also the first word in the line. Output: 3. When everything seemed so clear. 1. Wish. 3. When. 4. Now.

**10.** As the next command following question 9,

**$ awk  -f  h1.awk  float**

The command of awk it will read the file and apply it to the file h1.awk.

**11.**

|  |  |  |
| --- | --- | --- |
| $ **cat h2.awk** | | "Start to scan file" } |
| BEGIN { print | |
| {print  $1 | "," | $NF} |
| END {print | "END-" , FILENAME } | |

* **awk -f h2.awk float**

First command prints the first word of each line and afterwards prints the last word of each line and prints the file name at the end of the code.

Output: Wish, strong,

And, days

When,clear.

Now,all…

END- float

**12. sed  's/\s/\t/g'   float**

The command sed will change (s) spaces with tab (\t) of the file named (/g)

Output:

Wish I floating blue across the sky, my

Imagination is strong

And I often visit the days

When everything seemed so clear.

Now I wonder what I’m doing here at all…

**13.**

$ ls \*.awk| awk '{print "grep --color 'BEGIN' " $1 }' |sh *(Notes:* ***sh file*** *runs file as a shell script . $1 should be the output of  ‘* ls \*.awk ‘ in this case, not the 1st field *)*

In this command it has three parts, the first part will list all the files extensions in the awk, in the second part it will go the file and show the lines with the word BEGIN, then in the last part will make the file a shell script.

**14.**

$ mkdir  test test/test1  test/test2

$cat>test/testt.txt This is a test file ^D

* cd  test
* ls  -l **.** | grep '^d' | awk '{print "cp-r"  $NF  "" $NF ".bak"}' | sh

mkdir command will make directory called test with one called test 1 and test 2. Cat command will show what’s the text file in test directory. Cd command will change the directory to test. Then the last command will create a backup file in the extension into the directory and subdirectory and will print cp -r.

**Part III Programming: 15pts**

15. Sort all the files in your class working directory (or your home directory) as per the following requirements:

1. A copy of each file in that folder must be made. Append the string “\_copy” to the name of the file
2. The duplicate (copied) files must be in separate directories with each directory specifying the type of the file (e.g. txt files in directory named txtfiles, pdf files in directory named pdffiles etc).
3. The files in each directory must be sorted in chronological order of months.
4. An archive file (.tar) of each directory must be made. The .tar files must be sorted by name in ascending order.
5. An archive file of all the .tar archive files must be made and be available in your home directory.

As an output, show your screen shots for each step or a single screenshot that will cover the outputs from all the steps.

Graphical user interface

Description automatically generated