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: Paul Ofremu Jr.

Campus ID: pofremu1

Panther #: 002513676

PART 1

1. grep, egrep, and fgrep are all commands to filter text. grep uses basic regular expressions to match patterns. grep. egrep uses extended regular expressions to match patterns. fgrep only matches with fixed strings. (Ex: **grep** ‘sw.*ng’ filename ; **egrep** ‘s.+w’ filename ; **fgrep** ‘swing’ filename)

2. The utility tar is used to compress and decompress files. Ex: **tar** nameOfCompressedFile [list of files or dir]

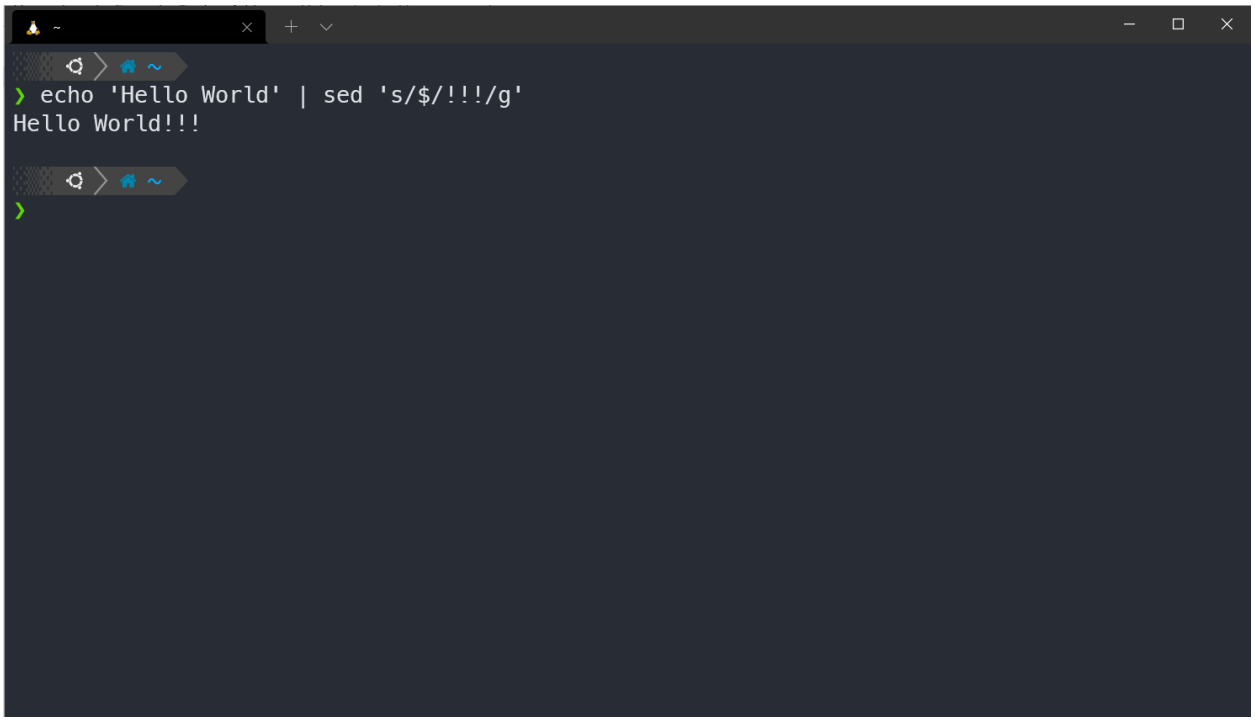
3. The commands **awk** and **cut** can be used to break a line into multiple fields. The default separator is whitespace. To define a separator manually set the FS variable to a new separator $FS=”;” Ex: awk ‘{ FS = “,” } ; The default separator for cut is TAB. To change it **cut** -d(--**new separator**)

4. The sort command sorts the content of a file line by line. The different fields are numerically, reverse, column number, month. Ex:

cat textfile.txt	=>	**sort** textfile.txt
John		Abby
Mary		John
Abby		Mary

Part IIa (5 points each): 25pts

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



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

   -- 1 <= NF { print $5 }
   If there is 1 or more fields, print the 5th field of each line
   -- NR >= 1 && NR >= 5 { print $1 }
   Print the 1st field of the 5th and greater rows
   -- 1,5 { print $0 }

   -- {print $1 }
   Print the first field of each line

7. What is the output of the following command line:
   echo good | sed '/Good/d'

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

   awk “/.*\+$/” foo

9. 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' foo
   - $ sed '$(($(wc -l < foo)-5)),$ d' foo

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

This command matches lines with any string that contains any amount of any characters followed by “ing” and prints the line number with the first word in that line.

```
❯ awk '/.*ing/ {print NR  ":"  $1}' float
1:Wish
3:When
5:Now
```

10. As the next command following question 9,

$ awk -f h1.awk float

This command matches with line numbers greater than 2 and less than 4 and prints the line number followed by the whole line.

```
❯ awk -f h1.awk float
3:When everything seemed so clear.
```

11.

$ **cat h2.awk**

```
BEGIN { print      "Start to scan file" }
{print  $1       ","  $NF}
END {print      "END-" , FILENAME }
```

$ awk -f h2.awk float

This awk file prints the first and last word of each line separated by a comma

```
❯ awk -f h2.awk float
Start to scan file
Wish,days
,
When,clear.
,
Now,all...
END- float
```

12. sed 's/\s/\t/g' float

This command replaces all spaces with tabs.

```
❯ sed  's/\s/\t/g'   float
Wish    I       was     floating        in      blue    across  the     sky,    my      imagination  i
s       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)*

This command goes through each .awk file and prints lines that have “BEGIN” as the first word

```
❯ ls *.awk| awk '{print "grep --color 'BEGIN' " $1 }' |sh
BEGIN {print "Start to scan file"}
```

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

This command matches with all directories within the test folder and makes a backup of the directories

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

❯ ls
test1  test1.bak  test2  test2.bak  testt.txt
```

Part III Programming: 15pts

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

Files:

```
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ ls
c_prog1.c  c_prog2.c  script1.sh  script2.sh  textfile1.txt  textfile2.txt
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$
```

a. A copy of each file in that folder must be made. Append the string "_copy" to the name of the file

```
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ ls -l | grep "^-" | awk '{print "cp " $NF " _copy"$NF""}' | sh
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ ls
_copyc_prog1.c  _copyscript1.sh  _copytextfile1.txt  c_prog1.c  script1.sh  textfile1.txt
_copyc_prog2.c  _copyscript2.sh  _copytextfile2.txt  c_prog2.c  script2.sh  textfile2.txt
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$
```

b. 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).

```
copiedfiles=$(ls -l | grep "^_copy" | awk '{print $NF}')
extensions=$(ls | sed 's/^.*\.//' | sort -u)

for ext in $extensions
do
    mkdir "$ext"files
    for file in copiedfiles
    do
        mv *.$ext "$ext"files
    done
done
```

```
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ ../movefiles.sh
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ ls -l
total 12
drwxrwxr-x. 2 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 4096 Oct  8 12:17 cfiles
drwxrwxr-x. 2 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 4096 Oct  8 12:17 shfiles
drwxrwxr-x. 2 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 4096 Oct  8 12:17 txtfiles
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$
```

c. The files in each directory must be sorted in chronological order of months.

```
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ ls -l | grep "^d" | awk '{print "ls -l", $NF "| sort"}' | sh
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:20 c_prog1.c
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:20 c_prog2.c
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:25 _copyc_prog1.c
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:25 _copyc_prog2.c
total 0
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:20 script1.sh
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:20 script2.sh
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:25 _copyscript1.sh
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:25 _copyscript2.sh
total 0
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:20 textfile1.txt
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:20 textfile2.txt
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:25 _copytextfile1.txt
-rw-rw-r--. 1 pofremu1@gsuad.gsu.edu pofremu1@gsuad.gsu.edu 0 Oct  8 11:25 _copytextfile2.txt
total 0
```

d. An archive file (.tar) of each directory must be made. The .tar files must be sorted by name in ascending order.

```
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ ls -l | grep "^d" | awk '{print "tar -cf " ""$NF".tar " $NF}' |
sort | sh
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ ls
cfiles  cfiles.tar  shfiles  shfiles.tar  txtfiles  txtfiles.tar
```

e. An archive file of all the .tar archive files must be made and be available in your home directory.

```
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ mkdir tarfiles
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ ls *.tar | awk '{print "mv " $NF " tarfiles"}' | sh
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ tar -cf tarfiles.tar tarfiles
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ mv tarfiles.tar ~/tarfiles.tar
[pofremu1@gsuad.gsu.edu@snowball homeworks2]$ cd ~
[pofremu1@gsuad.gsu.edu@snowball ~]$ ls
Foo.class  Lab6          copyfiles.sh  foo.sh     homeworks2       myName    pun.c
Lab3       Result        csc3220       hello      mandatabase.txt  myName.c  simple.sh
Lab4       a.out         foo.class     hello.c    midterm          public    tarfiles.tar
Lab4test   awk_practice  foo.java      homeworks  movefiles.sh     pun       test.txt
```

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