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: Alex Siegel

Campus ID: asiegel11

Panther #: 002-41-1802

PART 1 (2.5 points each): 10pts

1. What are the differences among ***grep***, ***egrep*** *and* ***fgrep***? Describe using an example.
    a. grep, egrep, and fgrep are all used to search a file for a particular string. However, grep only supports basic regex expressions while egrep supports extended regex expressions. fgrep is used to search for fixed character strings. Egrep and fgrep are essentially the same as grep -E and grep -F respectively.

2. 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.
    a. The grep command can be used to compress and decompress files as well as compress multiple files into a single file. One example is the command “200,300 s/A/a/g’ f1 f2 f3 >new

3. 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.
    a. The awk breaks a line into multiple fields. The default separator is a tab/space, but a separator can be defined by using the -F option. One example of the command is awk -F: ‘{ print NF,$1 }’ /etc/passwd

4. What does the ***sort*** command do? What are the different possible fields? Explain using an example.
    a. The sort command is used to sort a file in ascending or descending order. The command sort -tk -r +0 -2 -b -f -M

test.txt has many different fields. The -t field sets the seperator to k instead of blank. The -r sorts the file descending instead of ascending. The +0 is the starting position and the -2 is the ending position. The -b ignores leading blanks, the -f command ignores case, the -M command sorts the file by month and the -n sorts the file by numeric sort.

Part IIa (5 points each): 25pts

5. What is the output of the following sequence of bash commands: **echo 'Hello World' | sed 's/$/!!!/g'**
    a. Hello World!
6. What is the output for each of these awk script commands?

    -- 1 <= NF { print $5 }: Prints the 5th field of every non-blank line
    -- NR >= 1 && NR >= 5 { print $1 }: Prints the first field on every line after the 4th line (not including the 4th)
    -- 1,5 { print $0 } – Prints every line
    -- {print $1 } – Prints the first field of every line

7. What is the output of the following command line:
    echo good | sed '/Good/d'
    a. The output is “good” since the sed command is case-sensitive

8. Which **awk** script outputs all the lines where a plus sign + appears at the end of line?
    a. /.*\+$/ {print $0}

9. What is the command to delete only the first 5 lines in a file "foo"? Which command deletes only the last 5 lines?
    a. sed ‘1,5 d’ foo
    b. tail -n5 foo | sed ‘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

a) 1: Wish
   3: When
   5: Now

10. As the next command following question 9,

$ awk -f h1.awk float

a. 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**

a. Wish, days
   ,
   When,clear
   ,
   Now,all
   ,
   END- float

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

a. Wish I was floating in blue
across the sky, my imagi
nation 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)*

a. 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

a. sh: line 1: cp-rtest1test1.bak: command not found
sh: line 2: cp-rtest2test2.bak: command not found

Part III Programming: 15pts

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

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

```
[asiegel11@gsuad.gsu.edu@snowball ~]$ ./HW2.sh
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls
awkfiles  HW2.sh  shfiles        TestFile3.awk  Testfile.sh
cfiles    Lab3    Testfile2.txt  Testfile4.c    txtfiles
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls txtfiles
Testfile2_copy.txt
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls awkfiles
TestFile3_copy.awk
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls cfiles
Testfile4_copy.c
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls shfiles
HW2_copy.sh  Testfile_copy.sh
[asiegel11@gsuad.gsu.edu@snowball ~]$
```

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).

```
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls
HW2.sh  Lab3  Testfile2.txt  TestFile3.awk  Testfile4.c  Testfile.sh
[asiegel11@gsuad.gsu.edu@snowball ~]$ ./HW2.sh
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls
awkfiles  HW2.sh  shfiles        TestFile3.awk  Testfile.sh
cfiles    Lab3    Testfile2.txt  Testfile4.c    txtfiles
[asiegel11@gsuad.gsu.edu@snowball ~]$
```

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

```
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls shfiles
HW2_copy.sh  Testfile_copy.sh
[asiegel11@gsuad.gsu.edu@snowball ~]$ cat >F.sh
Testffff
[asiegel11@gsuad.gsu.edu@snowball ~]$
[asiegel11@gsuad.gsu.edu@snowball ~]$ ./HW2.sh
Testffff




        #Adds each file to its respective folder
#!/bin/bash
        cp "$f" "$d"
# Creates the folder to store each file in
done
        d="$s/$f"
        ext="${f##*.}"
        fi
for f in $(ls -p | grep -v /); do
        if [ ! -d $s ]
                mkdir $s
        mv "$d" "${d%.${ext}}_copy.${ext}"
        s="${ext}files"
        sort -M $f
        #Sorts each file
        then
Bean
Lol
This is a test
awk
TEst3
c
test4
bean
hi
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls shfiles
F_copy.sh  HW2_copy.sh  Testfile_copy.sh
```

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

```
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls
awkfiles  awkfiles.tar  cfiles  cfiles.tar  F.sh  HW2.sh  Lab3  shfiles  shfiles.tar  tarfiles  tarfiles.tar  Testfile2.txt  TestFile3.awk  Testfile4.c  Testfile.sh  txtfiles  txtfiles.tar
[asiegel11@gsuad.gsu.edu@snowball ~]$
```

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

```
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls *.tar | tar -cvf allFiles.tar *.tar
awkfiles.tar
cfiles.tar
shfiles.tar
tarfiles.tar
txtfiles.tar
[asiegel11@gsuad.gsu.edu@snowball ~]$ ls
allFiles.tar  awkfiles.tar  cfiles.tar  HW2.sh  shfiles      tarfiles      Testfile2.txt  Testfile4.c  txtfiles
awkfiles      cfiles        F.sh        Lab3    shfiles.tar  tarfiles.tar  TestFile3.awk  Testfile.sh  txtfiles.tar
```

```
#!/bin/bash

# Creates the folder to store each file in
for f in $(ls -p | grep -v /); do
        ext="${f##*.}"
        s="${ext}files"
        if [ ! -d $s ]
        then
                mkdir $s
        fi

        #Adds each file to its respective folder
        d="$s/$f"
        cp "$f" "$d"
        mv "$d" "${d%.${ext}}_copy.${ext}"

        #Sorts each file
        sort -M $f
done

# Creates a tar file
# Loops through all directories
for d in $(ls -d */ | egrep '(files/)$'); do

        #Create the tar file
        tar -cvf "${d%/}.tar" $d
done

# Creates a tar file of the other tar files
ls *.tar | tar -cvf allFiles.tar *.tar
~
~
```

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