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: Sriram Mohan

Campus ID: smohan6

Panther #: 002535667

PART 1

1.

S. no	grep	fgrep	egrep
1.	grep is a command used to search for basic regular expressions.	fgrep is a command used to search for fixed regular expressions.	egrep is a command used to search for extended regular expressions
2.	Eg:- grep '[a-z]*([a-z])' homework_instructions.txt	Eg:- fgrep 'topic(s)' homework_instructions.txt	Eg:- egrep '[a-z]*\([a-z]\)' homework_instructions.txt

All the above examples can return "topic(s)" String as an output.

2. sed utility can be used to compress and decompress files in Shell.

   You can compress files f1, f2, … , and fn by writing it to file new.txt using the following commands:

   ```
   $ sed ' <expression>/x'  f1 f2 f3 … fn > new.txt
   ```

   , where x is a label such as d, g, or p.

   Eg:-

   ```
   $ sed '1,5 g' addressOfScalar.c addressOfArray.c getMostFreqChar.c > n
   ew.txt
   ```

3. awk can be used to split a line into separate fields using a delimiter. By default, lines are delimited by a space. But you can define your own delimiter using the -F or –field-separator option:

   ```
   $ sed -F <delimiter> <command>
   ```

   Eg:-

   ```
   $ awk -F , {print $1 " to " $2}
   ```

   for any document with commas separating 2 words on a line.

4. The sort utility is used to sort a document in ascending or descending order based on a number of fields.

The number of fields for sorting depends on the delimiter and number of words in the document. The sort utility has many options such as -r (descending), -M (month), -n (numeric), -g (general numeric), -h (human numeric), etc.

$ sort <tc> <-r> +P1 -P2 … ±Pn <-Mnhg> <filename>

, where c is a delimiter and P1,..,Pn are the fields.

$ sort -r +0 -1 -n homework_instructions.txt

will sort the homework submission instructions in descending order.

PART IIa

5. Hello World!!!

6. This question does not specify which file to apply this command to. So, I am just going to attach screenshots for each command when tested on homework_instructions.txt which contains the submission instructions.

   -- 1 <= NF { print $5 }

```
[smohan6@gsuad.gsu.edu@snowball homeworks]$ awk -F: -f exec.awk homework_instructions.txt
```

   -- NR >= 1 && NR >= 5 { print $1 }

```
[smohan6@gsuad.gsu.edu@snowball homeworks]$ awk -F: -f exec.awk homework_instructions.txt
4.      Keep this page 1 intact on all your submissions. If this submissions instructions page is missing i
n 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 yo
ur 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.
```

   1,5 { print $0 }

```
[smohan6@gsuad.gsu.edu@snowball homeworks]$ awk -F: -f exec.awk homework_instructions.txt
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 i
n 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 i
n 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 yo
ur 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.
```

   {print $1 }

```
[smohan6@gsuad.gsu.edu@snowball homeworks]$ awk -F: -f exec.awk homework_instructions.txt
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 i
n 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 i
n 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 yo
ur 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.
```

7. good

8. $ awk -F: '{print NR " " $1 "+"}'

9. Deleting the first 5 lines of file:
   $ sed '1,5 d' file > x | mv x file

   Deleting the last 5 lines of file:
   $ sort -r +0 -1 file|sed '1,5 d' file > x | mv x file
   , where x is the new file name.

PART IIb

9. This awk command is printing out the first word of each line of float that contains expressions ending with '-ing', preceded by its line number and a colon. Thus only the first word of lines 1,3, and 4 will be printed.

   The following is the output of the given command:
   1:Wish
   3:When
   4:Now

10. The given awk script prints only the lines in float with line number greater than 2 and lesser than 4. Thus, it only printed out line 3, preceded by its line number and a colon.

    The following is the output of the given command:
    3:When everything seemed so clear.

11. The given awk script prints out the first word of a line, followed by a comma, followed by the n^{th} word of the line (where n is the line number), which just so happens to be last word of that line. Of course it also prints out "Start to scan file" and "END- " with the file name.

    The following is the output of the given command:
    Start to scan file
    Wish,is
    strong,,days
    When,clear.
    Now,all...
    END- float

12. The given sed command substitutes all whitespaces of float with tabs and prints each line of float with nothing but tabs. In other words, it sort of tabulates the entire float document.

    The following is the output of the given command:
    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...

13. The given set of commands is meant to do the following:
    - List all the files of .awk extension.
    - Print all the lines in such files that contain the keyword "BEGIN"
    - Run the previous awk script as a shell script.

The following is the output of the given command:
BEGIN { print "Start to scan file" }

14. The given list of commands is meant to do the following:
    - Create a directory called test, with two sub-directories test1 and test2.
    - Create a text file known as testt.txt in the test file.
    - Change the directory to that test directory.
    - Do the following:
        - List the files in the test directory.
        - Out of all those files, find the ones that have '^d' in their file permissions.
        - Create .bak files for each of them by using the cp command inside an awk command.
        - Run the previous awk script as a shell script.

    The following is the output of the given command:
    #on listing the files using ls
    test1 test1.bak test2 test2.bak testt.txt

PART III

Step 1: I log into my account on the server.

```
C:\Users\srimo>ssh smohan6@snowball.cs.gsu.edu
smohan6@snowball.cs.gsu.edu's password:
Last login: Sat Nov 27 17:56:15 2021 from c-76-17-78-22.hsd1.ga.comcast.net
   +
   |   GSU Computer Science
   |   Instructional Server
   |   SNOWBALL.cs.gsu.edu
   +
```

Step 2: Next, I do the following.

- I create directories each type of file (excluding folders on my home page).
- I search for all files of that type, using the ls and grep commands, with their respective extensions.
- I copy these files into these directories and append them with "_copy" and their respective extensions, using an awk turned shell script command.
- On creating those directories, I sort their files chronologically by month.
- I repeat this process with every type of file.

```
[smohan6@gsuad.gsu.edu@snowball ~]$ mkdir txtfiles | ls | grep '\.txt' | awk '{print "cp -r  ~/" $NF " ~/tx
tfiles/" $NF "_copy.txt"}' | sh | sort +0 -1 -M
[smohan6@gsuad.gsu.edu@snowball ~]$ cd txtfiles
[smohan6@gsuad.gsu.edu@snowball txtfiles]$ ls
mandatabase.txt_copy.txt  newList.txt_copy.txt    Result.txt_copy.txt
myexamfile.txt_copy.txt   phoneBook.txt_copy.txt
```

```
[smohan6@gsuad.gsu.edu@snowball ~]$ mkdir shfiles | ls | grep '\.sh' | awk '{print "cp -r  ~/" $NF " ~/shfi
les/" $NF "_copy.sh"}' | sh | sort +0 -1 -M
[smohan6@gsuad.gsu.edu@snowball ~]$ cd shfiles
[smohan6@gsuad.gsu.edu@snowball shfiles]$ ls
calc.sh_copy.sh        foo.sh_copy.sh    helpme.sh_copy.sh     simple.sh_copy.sh
checkError.sh_copy.sh  hello.sh_copy.sh  PhoneBook.sh_copy.sh  statementCount1.sh_copy.sh
```

```
[smohan6@gsuad.gsu.edu@snowball ~]$ mkdir cfiles | ls | grep '\.c' | awk '{print "cp -r  ~/" $NF " ~/cfiles
/" $NF "_copy.c"}' | sh | sort +0 -1 -M
[smohan6@gsuad.gsu.edu@snowball ~]$ cd cfiles
[smohan6@gsuad.gsu.edu@snowball cfiles]$ ls
findStr.c_copy.c  foo.class_copy.c  hello.c_copy.c  myName.c_copy.c
```

```
[smohan6@gsuad.gsu.edu@snowball ~]$ mkdir outfiles | ls | grep '\.out' | awk '{print "cp -r  ~/" $NF " ~/ou
tfiles/" $NF "_copy.out"}' | sh | sort +0 -1 -M
[smohan6@gsuad.gsu.edu@snowball ~]$ cd outfiles
[smohan6@gsuad.gsu.edu@snowball outfiles]$ ls
a.out_copy.out
```

```
[smohan6@gsuad.gsu.edu@snowball ~]$ mkdir pdffiles | ls | grep '\.pdf' | awk '{print "cp -r  ~/" $NF " ~/pd
ffiles/" $NF "_copy.pdf"}' | sh | sort +0 -1 -M
[smohan6@gsuad.gsu.edu@snowball ~]$ cd pdffiles
[smohan6@gsuad.gsu.edu@snowball pdffiles]$ ls
544c18_a73343e7e2dc42cf8ecc7e342684c029.pdf_copy.pdf
```

```
[smohan6@gsuad.gsu.edu@snowball ~]$ mkdir javafiles | ls | grep '\.java' | awk '{print "cp -r  ~/" $NF " ~/
javafiles/" $NF "_copy.java"}' | sh | sort +0 -1 -M
[smohan6@gsuad.gsu.edu@snowball ~]$ cd javafiles
[smohan6@gsuad.gsu.edu@snowball javafiles]$ ls
foo.java_copy.java
```

```
[smohan6@gsuad.gsu.edu@snowball ~]$ mkdir classfiles | ls | grep '\.class' | awk '{print "cp -r  ~/" $NF "
~/classfiles/" $NF "_copy.class"}' | sh | sort +0 -1 -M
[smohan6@gsuad.gsu.edu@snowball ~]$ cd classfiles
[smohan6@gsuad.gsu.edu@snowball classfiles]$ ls
foo.class_copy.class
```

Step 3: Next, I redirect myself to the home directory. I then list all files, use grep to filter the folders with the expression '.*files'. Then I create an awk script to print a command that takes these files and creates .tar files for each of them.

```
[smohan6@gsuad.gsu.edu@snowball ~]$ ls | grep '.*files' | awk '{print "tar -cvf " $NF ".tar " $NF}' | sh
cfiles/
cfiles/myName.c_copy.c
cfiles/findStr.c_copy.c
cfiles/hello.c_copy.c
cfiles/foo.class_copy.c
classfiles/
classfiles/foo.class_copy.class
javafiles/
javafiles/foo.java_copy.java
outfiles/
outfiles/a.out_copy.out
outfiles/pdffiles/
pdffiles/
pdffiles/544c18_a73343e7e2dc42cf8ecc7e342684c029.pdf_copy.pdf
shfiles/
shfiles/PhoneBook.sh_copy.sh
shfiles/hello.sh_copy.sh
shfiles/statementCount1.sh_copy.sh
shfiles/helpme.sh_copy.sh
shfiles/checkError.sh_copy.sh
shfiles/calc.sh_copy.sh
shfiles/simple.sh_copy.sh
shfiles/foo.sh_copy.sh
txtfiles/
txtfiles/newList.txt_copy.txt
txtfiles/phoneBook.txt_copy.txt
txtfiles/Result.txt_copy.txt
txtfiles/mandatabase.txt_copy.txt
txtfiles/myexamfile.txt_copy.txt
[smohan6@gsuad.gsu.edu@snowball ~]$ ls
544c18_a73343e7e2dc42cf8ecc7e342684c029.pdf  foo.java       Lab9             phoneBook.txt
a.out                                        foo.sh         ls               public
calc.sh                                      hello          mandatabase.txt  publicRegular
cfiles                                       hello.c        midterm          publicSubmission
cfiles.tar                                   hello.sh       myexamfile.txt   Result
checkError.sh                                helpme.sh      myName.c         Result.txt
classfiles                                   homeworks      newList.txt      shfiles
classfiles.tar                               javafiles      outfiles         shfiles.tar
csc3320                                      javafiles.tar  outfiles.tar     simple.sh
findStr                                      Lab3           pdffiles         statementCount1.sh
findStr.c                                    Lab4           pdffiles.tar     txtfiles
foo.class                                    Lab8           PhoneBook.sh     txtfiles.tar
```

Step 4: Finally, I create one last tar file for all the other tar files by using an ls-grep statement that filters all the files of extension .tar nested in a tar command. I call it tar.tar and examine its contents.

```
[smohan6@gsuad.gsu.edu@snowball ~]$ tar -tvf tar.tar
-rw-rw-r-- smohan6@gsuad.gsu.edu/smohan6@gsuad.gsu.edu 10240 2021-11-28 03:08 cfiles.tar
-rw-rw-r-- smohan6@gsuad.gsu.edu/smohan6@gsuad.gsu.edu 10240 2021-11-28 03:08 classfiles.tar
-rw-rw-r-- smohan6@gsuad.gsu.edu/smohan6@gsuad.gsu.edu 10240 2021-11-28 03:08 javafiles.tar
-rw-rw-r-- smohan6@gsuad.gsu.edu/smohan6@gsuad.gsu.edu 20480 2021-11-28 03:08 outfiles.tar
-rw-rw-r-- smohan6@gsuad.gsu.edu/smohan6@gsuad.gsu.edu 174080 2021-11-28 03:08 pdffiles.tar
-rw-rw-r-- smohan6@gsuad.gsu.edu/smohan6@gsuad.gsu.edu  20480 2021-11-28 03:08 shfiles.tar
-rw-rw-r-- smohan6@gsuad.gsu.edu/smohan6@gsuad.gsu.edu  51200 2021-11-28 03:08 txtfiles.tar
```

```
[smohan6@gsuad.gsu.edu@snowball ~]$ tar -cvf tar.tar $(ls | grep '.*\.tar')
cfiles.tar
classfiles.tar
javafiles.tar
outfiles.tar
pdffiles.tar
shfiles.tar
txtfiles.tar
[smohan6@gsuad.gsu.edu@snowball ~]$ ls
544c18_a73343e7e2dc42cf8ecc7e342684c029.pdf  foo.sh          mandatabase.txt  publicSubmission
a.out                                        hello           midterm          Result
calc.sh                                      hello.c         myexamfile.txt   Result.txt
cfiles                                       hello.sh        myName.c         shfiles
cfiles.tar                                   helpme.sh       newList.txt      shfiles.tar
checkError.sh                                homeworks       outfiles         simple.sh
classfiles                                   javafiles       outfiles.tar     statementCount1.sh
classfiles.tar                               javafiles.tar   pdffiles         tar.tar
csc3320                                      Lab3            pdffiles.tar     txtfiles
findStr                                      Lab4            PhoneBook.sh     txtfiles.tar
findStr.c                                    Lab8            phoneBook.txt
foo.class                                    Lab9            public
foo.java                                     ls              publicRegular
```