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: Aleeza Iftikhar

Campus ID: aiftikhar2

Panther #: 002473603

PART 1 (2.5 points each): 10pts

1. What are the differences among ***grep***, ***egrep*** *and* ***fgrep***? Describe using an example.
    1. grep searches for strings and supports basic regexes only.
    2. egrep is extension of grep, which also supports extended regexes.
    3. fgrep searches for fixed/exact strings only.

    Example:
    - grep 's\(earch\|tring\)' file.txt searches for two words in file: search or string.
    - egrep 's(earch|tring)' file.txt does the same as above without escaping metacharcters.
    - fgrep 's(earch|tring)' file.txt looks for exact string i.e., s(earch|tring).

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.
    - gzip filename for compress
    - gzip -d filename or gunzip filename for decompress
    - To compress multiple filles into a single file, create a tar archive (.tar file) and then compress it with gzip (.tar.gz or .tgz). Example:

      tar -cvf - file1 file2 | gzip > combinedFile.tar.gz

      or tar -cvzf combinedFile.tar.gz file1 file2

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.
    - awk, sed, tr
    - spaces and tabs
    - awk -F, filename use comma as a separator.
    - sed ‘s/:/\n/g’ filename replace colon with new line.
    - cat filename | tr ‘:’ '\n' replace colon with new line.

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

    sort → sorts lines of text files (in ascending ASCII) and prints output.

    - Suppose file letter.txt contains following content:
      Moon
      Sun
      Aleeza

    sort letter.txt will not change original content of file instead it will print output in sorted manner as follows:
      Aleeza
      Moon
      Sun

    - There are many different field options in sort, for example:
        1) -r will print all things in above file in descending order.
        2) -f will ignore case while sorting
        3) -b will ignore leading blanks while sorting
        4) -n sort numerically
        5) -o write sorted output to a new file
        6) -u remove duplicates while sorting
        7) -M sort by month.

Part IIa (5 points each): 25pts

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

   Hello World!!!

```
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$ echo 'Hello World' | sed 's/$/!!!/g'
Hello World!!!
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$
```

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

   Testing commands on mountainList.txt

   -- 1 <= NF { print $5 }

   Print fields starting from fifth field (separator: space) in each line

```
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$  awk '1 <= NF { print $5 }' mountainLi
st.txt




point),4561,feet,Towns


County
point),4420,feet,Rabun

County

[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$
```

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

   Print first fields of each line only

```
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$  awk 'NR >= 1 && NR >= 5  { print $1 }
' mountainList.txt
Wolfpen
Counties
Blood
Tray
Grassy
Slaughter
Double
Coosa
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$
```

-- 1,5 { print $0 }

Print everything.

Note: A/c to textbook, "If *condition* is missing, *action* is performed on every line. If *action* is missing, then all matching lines are simply sent to standard output"

```
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$  awk '1,5 { print $0 }' mountainList.t
xt
    Brasstown Bald, (summit),4784,feet,Union County
Rabun Bald, (summit),4696,feet,Rabun County
Dick's Knob, (summit),4620,feet,Rabun County
                Hightower Bald, (summit),4568,feet,Towns County
Wolfpen Ridge, (ridge high point),4561,feet,Towns and Union
Counties
        Blood Mountain, (summit),4458,feet,Union County
Tray Mountain, (summit), 4430,feet,Towns County
        Grassy Ridge, (ridge high point),4420,feet,Rabun County
Slaughter Mountain, (summit),4338,feet,Union County
Double Spring Knob, (summit),4280,feet,Rabun County
Coosa Bald, (summit),4280,feet,Union County
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$
```

-- {print $1 }

Print first field of every line.

```
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$  awk '{print $1 }' mountainList.txt
Brasstown
Rabun
Dick's
Hightower
Wolfpen
Counties
Blood
Tray
Grassy
Slaughter
Double
Coosa
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$
```

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

   echo good | sed '/Good/d'

   good

```
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$ echo good | sed '/Good/d'
good
[aiftikhar2@gsuad.gsu.edu@snowball Lab4]$
```

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

   awk '/+$/' filename

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ cat hw2Rough.txt

Bucks Milwaukee     60 22 0.73+2+
Raptors Toronto     58 24+ 0.707
76ers Philadelphia 51 31 0.622
Celtics Boston      49 33 0.598
Pacers Indiana      48 34 0.585+
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ awk '/+$/' hw2Rough.txt
Bucks Milwaukee     60 22 0.73+2+
Pacers Indiana      48 34 0.585+
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

9. What is the command to delete only the first 5 lines in a file "foo"?
   Which command deletes only the last 5 lines?
   Original file: Total 13 lines (including dashes' line)

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ cat > hw2_rough.txt
line 1
line 2
line 3
line 4
line 5
line 6
line 7
line 8
line 9
line 10
line 11
line 12
        ------------------------------
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

Delete first 5 lines:
sed -i '1,5d' filename

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ sed -i '1,5d' hw2_rough.txt
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ cat hw2_rough.txt
line 6
line 7
line 8
line 9
line 10
line 11
line 12
        ------------------------------
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

Delete last 5 lines:
tac filename | sed '1,5 d' | tac

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ tac hw2_rough.txt | sed '1,5 d' | tac
line 6
line 7
line 8
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

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

print first field of every line with line numbers.

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ cat h1.awk
NR>2 && NR<4{print  NR  ":"     $0
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ awk    '/.*ing/ {print NR  ":"  $1}'  flo
at
1:Wish
2:When
3:Now
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

10. As the next command following question 9,

$ awk -f h1.awk float

File h1.awk had a command to print a full line between line 2 and line 4. -f allows to execute command from h1.awk.

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ awk -f h1.awk float
3:Now I wonder what I'm doing here at all...
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

11.

$ **cat h2.awk**

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

END {print     "END-" , FILENAME }
```

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

- **Print start to scan file,**
- **Then Print 'first word' followed by comma (,) followed by 'last word' of every line.**
- **Finally Print END- followed by filename (which is float in this case)**

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ cat h2.awk
BEGIN { print "Start to scan file" }
{print  $1      ","     $NF}
END {print      "END-" , FILENAME }
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ awk -f h2.awk float
Start to scan file
Wish,days
When,clear.
Now,all...
END- float
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

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

Replace all spaces characters with tab spaces in file.

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ sed 's/\s/\t/g' float
Wish    I       was     floating        in      blue    across  the     s
ky,     my      imagination     is      strong, And     I       often   v
isit    the     days
When    everything      seemed  so      clear.
Now     I       wonder  what    I'm     doing   here    at      all...
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

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)*

First, list all .awk files then search BEGIN and if found print that whole line with word ‘BEGIN’ in red color.

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ ls *.awk| awk '{print "grep --colo
r 'BEGIN' " $1 }' |sh
BEGIN { print "Start to scan file" }
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

14.

$ mkdir test test/test1 test/test2

Create directory test, then inside that directory create two sub-directories, test1 and test2.

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ mkdir test test/test1 test/test2
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ ls
a.out          foo.class  h2.awk    homeworks      Lab4      Result
checkError.sh  foo.java   hello     hw2_rough.txt  myName.c  simple.sh
csc3320        foo.sh     hello.c   hw2Rough.txt   practice  test
float          h1.awk     hello.sh  Lab3           public
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ cd test
[aiftikhar2@gsuad.gsu.edu@snowball test]$ ls
test1  test2
[aiftikhar2@gsuad.gsu.edu@snowball test]$
```

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

Create a file testt.txt inside test folder and write in it and exit.

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ cat > test/testt.txt
This is a test file [aiftikhar2@gsuad.gsu.edu@snowball ~]$ cd test
```

$ cd test

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

Open test folder and then make copies of directories in it with extension .bak

```
[aiftikhar2@gsuad.gsu.edu@snowball test]$ ls  -l . | grep '^d' | awk '{pr
int "cp -r "  $NF  " " $NF ".bak"}' | sh
[aiftikhar2@gsuad.gsu.edu@snowball test]$ ls
test1  test1.bak  test2  test2.bak  testt.txt
[aiftikhar2@gsuad.gsu.edu@snowball test]$
```

Part III Programming: 15pts

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

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ ls
a.out          float      h2.awk     hw2_rough.txt  myName.c    shfiles
awkfiles       foo.class  hello      hw2Rough.txt   otherfiles  simple.sh
cfiles         foo.java   hello.c    javafiles      practice    test
checkError.sh  foo.sh     hello.sh   Lab3           public      txtfiles
csc3320        h1.awk     homeworks  Lab4           Result
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

a. A copy of each file in that folder must be made. Append the string "_copy" to the name of the file
ls -l | grep -v '^d' | awk 'orig=$NF {gsub(/\./," .",$NF); print orig " " $NF}' | awk '{print "cp -r " $1 " " $2 "_COPY" $3 }' | sh

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ ls -l | grep -v '^d' | awk 'orig=$NF {gsu
b(/\./," .",$NF); print orig " " $NF}' | awk '{print "cp -r " $1 " " $2 "_COPY"
$3 }' | sh
cp: cannot stat ‘132’: No such file or directory
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ ls
a_COPY.out           foo_COPY.sh    hello.sh           practice
a.out                foo.java       homeworks          public
awkfiles             foo.sh         hw2_rough_COPY.txt Result
cfiles               h1.awk         hw2Rough_COPY.txt  Result_COPY
checkError_COPY.sh   h1_COPY.awk    hw2_rough.txt      shfiles
checkError.sh        h2.awk         hw2Rough.txt       simple_COPY.sh
csc3320              h2_COPY.awk    javafiles          simple.sh
float                hello          Lab3               test
float_COPY           hello.c        Lab4               txtfiles
foo.class            hello_COPY     myName.c
foo_COPY.class       hello_COPY.c   myName_COPY.c
foo_COPY.java        hello_COPY.sh  otherfiles
[aiftikhar2@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).
ls -l | grep _COPY | awk 'orig=$NF{gsub(/\./," ",$NF); print orig " " $NF}' | awk '{print "mkdir " $3 "files; mv " $1 " ./" $3 "files"}' | sh

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ ls
a.out          files      h2.awk         hw2Rough.txt  outfiles   test
awkfiles       float      hello          javafiles     practice   txtfiles
cfiles         foo.class  hello.c        Lab3          public
checkError.sh  foo.java   hello.sh       Lab4          Result
classfiles     foo.sh     homeworks      myName.c      shfiles
csc3320        h1.awk     hw2_rough.txt  otherfiles    simple.sh
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

In this step, all directories of respective names are made simultaneously, and copy files are moved to them at the same time.

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

ls -l | grep files | awk '{print "sort -M " $NF "/* " $NF "/*"}' | sh

```
[aiftikhar2@gsuad.gsu.edu@snowball shfiles]$ ls
checkError_COPY.sh  foo_COPY.sh  hello_COPY.sh  simple_COPY.sh
[aiftikhar2@gsuad.gsu.edu@snowball shfiles]$
```

This is just an example of one directory, above command is working on all directories simultaneously.

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

ls -l | grep files | awk '{print "tar -czvf " $NF ".tar " $NF}' | sh

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ ls -l | grep files | awk '{print "tar -cz
vf " $NF ".tar " $NF}' | sh
awkfiles/
awkfiles/h1_COPY.awk
awkfiles/h2_COPY.awk
cfiles/
cfiles/hello_COPY.c
cfiles/myName_COPY.c
classfiles/
classfiles/foo_COPY.class
files/
files/hello_COPY
files/Result_COPY
files/float_COPY
javafiles/
javafiles/foo_COPY.java
otherfiles/
outfiles/
outfiles/a_COPY.out
shfiles/
shfiles/simple_COPY.sh
shfiles/hello_COPY.sh
shfiles/checkError_COPY.sh
shfiles/foo_COPY.sh
txtfiles/
txtfiles/hw2Rough_COPY.txt
txtfiles/hw2_rough_COPY.txt
```

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ ls
a.out           files      hello.c        myName.c        shfiles.tar
awkfiles        files.tar  hello.sh       otherfiles      simple.sh
awkfiles.tar    float      homeworks      otherfiles.tar  test
cfiles          foo.class  hw2_rough.txt  outfiles        txtfiles
cfiles.tar      foo.java   hw2Rough.txt   outfiles.tar    txtfiles.tar
checkError.sh   foo.sh     javafiles      practice
classfiles      h1.awk     javafiles.tar  public
classfiles.tar  h2.awk     Lab3           Result
csc3320         hello      Lab4           shfiles
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

e. An archive file of all the .tar archive files must be made and be available in your home directory.
tar -czvf ~/final_tar.tar *.tar

```
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ tar -czvf ~/final_tar.tar *.tar
awkfiles.tar
cfiles.tar
classfiles.tar
files.tar
javafiles.tar
otherfiles.tar
outfiles.tar
shfiles.tar
txtfiles.tar
[aiftikhar2@gsuad.gsu.edu@snowball ~]$ ls
a.out           files          hello          Lab4            shfiles
awkfiles        files.tar      hello.c        myName.c        shfiles.tar
awkfiles.tar    final_tar.tar  hello.sh       otherfiles      simple.sh
cfiles          float          homeworks      otherfiles.tar  test
cfiles.tar      foo.class      hw2_rough.txt  outfiles        txtfiles
checkError.sh   foo.java       hw2Rough.txt   outfiles.tar    txtfiles.tar
classfiles      foo.sh         javafiles      practice
classfiles.tar  h1.awk         javafiles.tar  public
csc3320         h2.awk         Lab3           Result
[aiftikhar2@gsuad.gsu.edu@snowball ~]$
```

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