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: Dileep Reddy Pemmana

Campus ID: dreddy2

Panther #: 002531328

PART 1 (2.5 points each): 10pts

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

   Ans: ***grep*** is a command to match data given against basic regular expressions if no options are specified. ***egrep*** is a command to match data given against extended regular expressions (grep -E has the same behavior). ***fgrep*** is a command to match data given against fixed strings, used for files which use a lot of regex metacharcters (grep -F has the same behavior).

   For a file `file.txt`:

   ```
   This is a cold, cold, cold day
   (c\w*d, ){2}
   ```

   `grep 'c\w*d' file.txt` matches the three 'cold's and does not match the 'c1w*d' in the file, `egrep '(c\w*d, ){2}' file.txt` matches the two 'cold's with a comma while `fgrep '(c\w*d, ){2}' file.txt` matches the '(c\w*d,){2}' in the file

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.

   Ans: The `tar` utility can be used to compress and decompress files. Multiple files can be compressed into a single file by specifying each file as input for the tar utility,
   for e.g `tar -cvf files.tar file1 file2 file3`
   (or by including those files in a folder which tar then compresses.)

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.

   Ans: The `awk` and `sort` utilities can break a line into multiple fields by defining a separator. The default separator is the space character ‘ ’.

   Examples for defining separator for each utility

   ```
   awk -F, ‘{print $1,$3}’ table.csv
   Sort -t, +2 -3 table.csv
   ```

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

   Ans: The `sort` command separates data in each line of the text of a file or input based on a defined separator into fields and sorts in ascending or descending order based on one or more fields based on their ASCII values

   sort -tc -r +POS1 -POS2 -bfMn fileName
   where c is the separator, -r sorts the file in reverse order, +col1 -col2 sorts based on the data in the columns between columns col1 and col2 -b ignore leading blanks, -f ignores the case, -M sorts by month and -n sorts by number

For a file greek_alpha.txt:
Alpha 1 a
Beta 2 b
Iota 3 i
Gamma 4 g
Delta 5 d

`sort +2 -3 greek_alpha.txt (sort -k3 greek_alpha.txt` in newer versions of `sort`) outputs:
Alpha 1 a
Beta 2 b
Delta 5 d
Gamma 4 g
Iota 3 i

Part IIa (5 points each): 25pts

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

   Ans: `Hello World!!!`

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

   -- 1 <= NF { print $5 }
   Ans: If the number of fields in the line is at least one, prints the 5th field of that line
   -- NR >= 1 && NR >= 5 { print $1 }
   Ans: If the current line is greater than 5, prints the 1st field of that line
   -- 1,5 { print $0 }
   Ans: Prints the entire line for each line
   -- {print $1 }
   Ans: Prints the first field of each line

7. What is the output of the following command line:
   echo good | sed '/Good/d'
   Ans: good because the regex scans for capital-G 'Good' to delete while the input only has small-g 'good'

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

   Ans: `/\+$/{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?

   Ans: The command to delete only the first 5 lines of a file foo is
   `sed '1,5d' foo > foo`
   The command to delete only the last 5 lines of a file foo is
   `tail -r foo | sed '1,5d' | tail -r |cat > 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

Ans: `cat float` and `cat h1.awk` displayed the text in those files as shown. `awk '/.*ing/{print NR ":" $1}' float` selected each line that has the pattern of a set of characters ending with 'ing' and prints the line number, a colon and the first word of that line
Output:

```
1:Wish
3:When
4:Now
```

10. As the next command following question 9,

$ awk -f h1.awk float

Ans: The output is an error message because the awk script does not have a closing curly brace

Output:

```
awk: syntax error at source line 0 source file h1.awk
 context is
          NR>2 && NR<4{print  NR  ":" $0 >>>
 <<<
awk: illegal statement at source line 0 source file h1.awk
          missing }
```

11.

$ **cat h2.awk**

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

$ awk -f h2.awk float

Ans: This script first prints 'Start to scan file' before printing the first word of each line and then number of words in each line, and at the end prints 'END- float'

Output:

```
Start to scan file
Wish,13
And,6
When,5
Now,9
END- float
```

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

Ans: This regex replaces every space character in the poem with a tab.

Output:

```
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.

$ 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)*

Ans: This set of piped commands scans for all the .awk files in a directory and then takes it as an input for a text that prints out a grep command that displays each line with the word ‘BEGIN’ in that line, with the world highlighted in color. This is fed as a text file to sh which executes the grep command

Sample output:

```
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

Ans: The first two commands create a directory ‘test’ with subdirectories ‘test1’ and ‘test2’ and a text file ‘testt.txt’ with some text. The third command switches directories to test. The fourth command is a set of piped commands. `ls -l .` lists the files in the directory with the permissions listed first. This list is fed to `grep` which scans for the items that begin with the letter ‘d’, i.e. directories. This is fed to an `awk` script that scans for the last field of each line (ie the name of that directory) creates text of a `cp` command each line that copies that directory and appends ‘.bak’ to that copy. `sh` then runs the generated command

Output (Using 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:

a. A copy of each file in that folder must be made. Append the string "_copy" to the name of the file
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).
c. The files in each directory must be sorted in chronological order of months.
d. An archive file (.tar) of each directory must be made. The .tar files must be sorted by name in ascending order.
e. 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.

ANSWER:

For requirement b, creating the folders

```
ls -l . | grep '^-' | awk -F. '{print "mkdir "$NF"files"}'|sh
```

```
(base) dileepreddy@Dileeps-MacBook-Air done % ls -l . | grep '^-' | awk -F. '{print "mkdir "$NF"files"}'|sh
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pdffiles: File exists
mkdir: pdffiles: File exists
mkdir: pdffiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
mkdir: pngfiles: File exists
```

Satisfying requirements a and b

```
ls -l | grep '^-' | awk '{print $NF}'|awk -F. '{print "cp "$0 " "$2"files/" $1"_copy\."$2}'|sh
```

```
(base) dileepreddy@Dileeps-MacBook-Air done % ls -l | grep '^-' | awk '{print $NF}'|awk -F. '{print "cp "$0 " "$2"files/" $1"_copy\."$2}'|sh
cp: PM.png: No such file or directory
cp: PM.png: No such file or directory
cp: PM.png: No such file or directory
(base) dileepreddy@Dileeps-MacBook-Air done % ls -R
1.png                                       L1_P1_DileepReddy.png
10.png                                      L1_P2_DileepReddy.png
11.png                                      Lab2-InLab-Dileep.pdf
12.png                                      Lab2_2_DileepReddy.txt
13.png                                      Lab2_DileepReddy.pdf
14.png                                      Lab3_DileepReddy.pdf
15.png                                      Lab4-Out-of-Lab.pdf
2.png                                       Lab4-Out-of-Lab.tm
3.png                                       Screen Shot 2021-09-02 at 12.28.30 PM.png
4A.png                                      Screen Shot 2021-09-02 at 12.30.47 PM.png
4B.png                                      Screen Shot 2021-09-09 at 12.36.07 PM.png
5.png                                       new
6.png                                       pdffiles
7.png                                       pngfiles
8.png                                       tmfiles
9.png                                       txtfiles

./new:

./pdffiles:
Lab2-InLab-Dileep_copy.pdf      Lab3_DileepReddy_copy.pdf
Lab2_DileepReddy_copy.pdf       Lab4-Out-of-Lab_copy.pdf

./pngfiles:
10_copy.png                     1_copy.png                      6_copy.png
11_copy.png                     2_copy.png                      7_copy.png
12_copy.png                     3_copy.png                      8_copy.png
13_copy.png                     4A_copy.png                     9_copy.png
14_copy.png                     4B_copy.png                     L1_P1_DileepReddy_copy.png
15_copy.png                     5_copy.png                      L1_P2_DileepReddy_copy.png

./tmfiles:
Lab4-Out-of-Lab_copy.tm

./txtfiles:
Lab2_2_DileepReddy_copy.txt
```

Satisfying requirement c
(for current directory)

```
ls -l | grep '^-' | sort -k6 -M
```

```
(base) dileepreddy@Dileeps-MacBook-Air done % ls -l | grep '^-' | sort -k6 -M
-rw-r--r--@  1 dileepreddy  staff    13151 Aug 26 12:12 L1_P1_DileepReddy.png
-rw-r--r--@  1 dileepreddy  staff  4769280 Aug 26 14:59 L1_P2_DileepReddy.png
-rw-r--r--@  1 dileepreddy  staff     1367 Sep  8 21:42 Lab2_2_DileepReddy.txt
-rw-r--r--@  1 dileepreddy  staff     3571 Sep 18 00:37 Lab4-Out-of-Lab.tm
-rw-r--r--@  1 dileepreddy  staff    22878 Sep  8 21:42 Lab2_DileepReddy.pdf
-rw-r--r--@  1 dileepreddy  staff    38934 Sep 16 22:58 3.png
-rw-r--r--@  1 dileepreddy  staff    40104 Sep 16 22:58 2.png
-rw-r--r--@  1 dileepreddy  staff    83738 Sep 16 23:01 4B.png
-rw-r--r--@  1 dileepreddy  staff   104801 Sep  2 12:30 Screen Shot 2021-09-02 at 12.30.47 PM.png
-rw-r--r--@  1 dileepreddy  staff   123112 Sep 17 15:18 7.png
-rw-r--r--@  1 dileepreddy  staff   203733 Sep 17 15:22 8.png
-rw-r--r--@  1 dileepreddy  staff   207848 Sep 16 22:54 1.png
-rw-r--r--@  1 dileepreddy  staff   212174 Sep 17 15:58 9.png
-rw-r--r--@  1 dileepreddy  staff   212305 Sep 17 16:24 10.png
-rw-r--r--@  1 dileepreddy  staff   222878 Sep 17 17:59 15.png
-rw-r--r--@  1 dileepreddy  staff   298860 Sep 17 16:50 12.png
-rw-r--r--@  1 dileepreddy  staff   299805 Sep 17 00:03 5.png
-rw-r--r--@  1 dileepreddy  staff   299952 Sep 17 16:53 13.png
-rw-r--r--@  1 dileepreddy  staff   307766 Sep 17 16:54 14.png
-rw-r--r--@  1 dileepreddy  staff   324228 Sep 17 15:14 6.png
-rw-r--r--@  1 dileepreddy  staff   336836 Sep 17 16:45 11.png
-rw-r--r--@  1 dileepreddy  staff   339487 Sep 16 23:01 4A.png
-rw-r--r--@  1 dileepreddy  staff   350428 Sep 10 17:58 Lab3_DileepReddy.pdf
-rw-r--r--@  1 dileepreddy  staff   484301 Sep  2 12:36 Lab2-InLab-Dileep.pdf
-rw-r--r--@  1 dileepreddy  staff   504276 Sep  9 12:36 Screen Shot 2021-09-09 at 12.36.07 PM.png
-rw-r--r--@  1 dileepreddy  staff   509897 Sep  2 12:28 Screen Shot 2021-09-02 at 12.28.30 PM.png
-rw-r--r--@  1 dileepreddy  staff  1961024 Sep 18 00:38 Lab4-Out-of-Lab.pdf
```

(for folders inside current directory)

```
ls -l | grep '^d'| awk '{print "cd "$NF ";ls -l|sort -k6 -M; cd ..";}'| sh
```

```
(base) dileepreddy@Dileeps-MacBook-Air done % ls -l | grep '^d'| awk '{print "cd "$NF ";ls -l|sort -k6 -M; cd ..";}'|
total 5520
-rw-r--r--@ 1 dileepreddy  staff    22878 Sep 22 22:50 Lab2_DileepReddy_copy.pdf
-rw-r--r--@ 1 dileepreddy  staff   350428 Sep 22 22:50 Lab3_DileepReddy_copy.pdf
-rw-r--r--@ 1 dileepreddy  staff   484301 Sep 22 22:50 Lab2-InLab-Dileep_copy.pdf
-rw-r--r--@ 1 dileepreddy  staff  1961024 Sep 22 22:50 Lab4-Out-of-Lab_copy.pdf
total 16352
-rw-r--r--@ 1 dileepreddy  staff    13151 Sep 22 22:50 L1_P1_DileepReddy_copy.png
-rw-r--r--@ 1 dileepreddy  staff    38934 Sep 22 22:50 3_copy.png
-rw-r--r--@ 1 dileepreddy  staff    40104 Sep 22 22:50 2_copy.png
-rw-r--r--@ 1 dileepreddy  staff    83738 Sep 22 22:50 4B_copy.png
-rw-r--r--@ 1 dileepreddy  staff   123112 Sep 22 22:50 7_copy.png
-rw-r--r--@ 1 dileepreddy  staff   203733 Sep 22 22:50 8_copy.png
-rw-r--r--@ 1 dileepreddy  staff   207848 Sep 22 22:50 1_copy.png
-rw-r--r--@ 1 dileepreddy  staff   212174 Sep 22 22:50 9_copy.png
-rw-r--r--@ 1 dileepreddy  staff   212305 Sep 22 22:50 10_copy.png
-rw-r--r--@ 1 dileepreddy  staff   222878 Sep 22 22:50 15_copy.png
-rw-r--r--@ 1 dileepreddy  staff   298860 Sep 22 22:50 12_copy.png
-rw-r--r--@ 1 dileepreddy  staff   299805 Sep 22 22:50 5_copy.png
-rw-r--r--@ 1 dileepreddy  staff   299952 Sep 22 22:50 13_copy.png
-rw-r--r--@ 1 dileepreddy  staff   307766 Sep 22 22:50 14_copy.png
-rw-r--r--@ 1 dileepreddy  staff   324228 Sep 22 22:50 6_copy.png
-rw-r--r--@ 1 dileepreddy  staff   336836 Sep 22 22:50 11_copy.png
-rw-r--r--@ 1 dileepreddy  staff   339487 Sep 22 22:50 4A_copy.png
-rw-r--r--@ 1 dileepreddy  staff  4769280 Sep 22 22:50 L1_P2_DileepReddy_copy.png
total 8
-rw-r--r--@ 1 dileepreddy  staff  3571 Sep 22 22:50 Lab4-Out-of-Lab_copy.tm
total 8
-rw-r--r--@ 1 dileepreddy  staff  1367 Sep 22 22:50 Lab2_2_DileepReddy_copy.txt
```

For Requirement d;

Making the tar archives:

```
ls -l | grep '^d'| awk '{print "tar -cvf" $NF ".tar " $NF}'| sh
```

Sorting the .tar files:

```
ls | grep '\.tar' | sort
```

```
(base) dileepreddy@Dileeps-MacBook-Air done % ls -l | grep '^d'| awk '{print "tar -cvf" $NF ".tar " $NF}'| sh
a new
a pdffiles
a pdffiles/Lab2-InLab-Dileep_copy.pdf
a pdffiles/Lab3_DileepReddy_copy.pdf
a pdffiles/Lab2_DileepReddy_copy.pdf
a pdffiles/Lab4-Out-of-Lab_copy.pdf
a pngfiles
a pngfiles/5_copy.png
a pngfiles/L1_P2_DileepReddy_copy.png
a pngfiles/4A_copy.png
a pngfiles/10_copy.png
a pngfiles/11_copy.png
a pngfiles/8_copy.png
a pngfiles/9_copy.png
a pngfiles/2_copy.png
a pngfiles/3_copy.png
a pngfiles/6_copy.png
a pngfiles/7_copy.png
a pngfiles/14_copy.png
a pngfiles/15_copy.png
a pngfiles/1_copy.png
a pngfiles/13_copy.png
a pngfiles/12_copy.png
a pngfiles/L1_P1_DileepReddy_copy.png
a pngfiles/4B_copy.png
a tmfiles
a tmfiles/Lab4-Out-of-Lab_copy.tm
a txtfiles
a txtfiles/Lab2_2_DileepReddy_copy.txt
(base) dileepreddy@Dileeps-MacBook-Air done % ls | grep '\.tar' | sort
new.tar
pdffiles.tar
pngfiles.tar
tmfiles.tar
txtfiles.tar
```

For Requirement e:

```
ls | grep '\.tar' | awk 'BEGIN{printf "tar -cvf archive.tar "}{printf "%s ", $0}'|sh
```

```
(base) dileepreddy@Dileeps-MacBook-Air done % ls | grep '\.tar' | awk 'BEGIN{printf "tar -cvf archive.tar "}{printf "%s ", $0}'|sh
a new.tar
a pdffiles.tar
a pngfiles.tar
a tmfiles.tar
a txtfiles.tar
(base) dileepreddy@Dileeps-MacBook-Air done % ls -l  archive.tar
-rw-r--r--  1 dileepreddy  staff  11245056 Sep 22 23:15 archive.tar
```