CSc 3320: Systems Programming

Spring 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: Kimani Guchu

Campus ID: Kguchu

Panther #:002-39-3714

PART 1 (2.5 points each): 10pts

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

- Grep - basic regular functions

  Example: grep -w "new" test.txt - searches test file for lines containing the word "new"

- Egrep - Extended regular expression

  Example. egrep "new|fresh|happy" test.txt - searches test file for pattern help,fresh and regular.

- Fgrep - Deals with Fixed string

  Example. egrep "happy" test.txt - searches test file for the string happy.

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.

- The Tar utility allows us to compress and decompress files.
- To combine multiple tar files into a single file we can use "--concatenate"
- Example. Given we have a tar archive named new and old.

$Tar --concatenate --file=new.tar old.tar

After completion, the contents of our old.tar file will also be included in our current new.tar file.

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.

- We use the “cut” command to break lines into multiple fields
- The default separator used is a tab “ “ blank space
- To manually define a separator you will insert it after your cut command
- Example $cut -d “,” -f 1-4 Test.txt
- This command will use “,” as our separator and print fields 1-4 which are separated by a “,”

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

- The sort command sorts a file in ascending or descending order based on a imputed field
- Fields are based on the position of words in our lines of text for example “ Hello Im Kimani 1”, Hello will be field 0 and 1 will be field 3”
- If the previous text is placed in a text file with various names that follow the same format…..”$Sort +2 -3 Name.txt” will sort the resulting names in descending order based on field 2 which is the user's name in this case, thus outputting.
- “Hey im Alex 4”
- “ Hello Im Kimani 1”
- “Hola im Zeke 0”

Part IIa (5 points each): 25pts

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

Output: Hello World!!!

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

-- 1 <= NF { print $5 } = all contents of 5th column.
-- NR >= 1 && NR >= 5 { print $1 } = evaluates all row numbers, larger than 5 & prints 1st column of those rows.
-- 1,5 { print $0 } = prinits all content of our file.
-- {print $1 } = print all 1st column values.

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

Output: good

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

/\+$/{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?

- **First 5 lines -- sed -i 1,5d foo.txt**
- **Last 5 lines sed '$d' foo.txt | sed '$d' foo.txt | sed '$d' foo.txt| sed '$d' foo.txt| sed '$d' foo.txt - or we will need to use “head” command.**

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

- **$ cat float displays our float file**
-
- **$ cat h1.awk contains NR>2 && NR<4{print NR ":" $0**
-
- **$ awk '/.*ing/ {print NR ":" $1}' float - Will search our file for rows containing words ending in "ing" and will output only the 1st term of each thus,**
- **Output:**

  1.Wish

  2.When

  3.Now

10. As the next command following question 9,

$ awk -f h1.awk float

- **This command will use our code stored in h1.awk (NR>2 && NR<4{print NR ":" $0),**
- **This code Searches for the line larger than 2 but smaller than 4 thus it**

is searching for line 3 in our float text which will be,

- **output .**
  3.When everything seemed so clear.

11.

$ **cat h2.awk**

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

END {print "END-" , FILENAME }
$ awk -f h2.awk float

- **Will access float file and output will contain the first line “Start to scan file” and the last line will be “END” and given {print $1 "," $NF} well return the first column and last column or every row in our float file separated by a “,” as well as print the file name at the end of our output**
- Output:
- **Start to scan file**
- **Wish,imagination**
- **is, days**
- **When, clear**
- **Now,all...**
- **END -float**

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

- **Replaces all spaces” “ with tab spaces “ “**
- **Thus**
- **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 *)*

- This will search all files that end in .awk, and check if they contain the word "BEGIN" and output that line with "BEGIN" in a different color/
- Output:
- "Start to scan file" } BEGIN { print {print $1 "," $NF}

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

- $ mkdir test test/test1 test/test2 - creates our new test directory with subdirectories test 1, and test 2
- $cat>test/testt.txt
  This is a test file ^D - this will create test.txt file in test directory
- $ cd test -navigate to test directory
- $ ls -l **.** | grep '^d' | awk '{print "cp -r " $NF " " $NF ".bak"}' | sh
  - This will create .bak files for the test1 and test2 subdirectory

- **Output:**
- **Cp -r test1 test1.bak**
- **Cp -r test2 test2.bak**

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 o

```
[[kguchu1@gsuad.gsu.edu@snowball ~]$ ls
 copys     file1.txt  file3.txt  Lab3  PDF      simple.sh  test2.pdf
 csc3320   file2.txt  homeworks  Lab4  public   test1.pdf  TEXT
[[kguchu1@gsuad.gsu.edu@snowball ~]$ cp file1.txt TEXT/file1_copy.txt
[[kguchu1@gsuad.gsu.edu@snowball ~]$ cp file2.txt TEXT/file2_copy.txt
[[kguchu1@gsuad.gsu.edu@snowball ~]$ cp file3.txt TEXT/file3_copy.txt
[[kguchu1@gsuad.gsu.edu@snowball ~]$ cp test1.pdf PDF/test1_copy.txt
[[kguchu1@gsuad.gsu.edu@snowball ~]$ cp test2.pdf PDF/test2_copy.txt
 [kguchu1@gsuad.gsu.edu@snowball ~]$
```

```
[[kguchu1@gsuad.gsu.edu@snowball TEXT]$ sort -M file1_copy.txt
[[kguchu1@gsuad.gsu.edu@snowball TEXT]$ sort -M file2_copy.txt
[[kguchu1@gsuad.gsu.edu@snowball TEXT]$ sort -M file3_copy.txt
[[kguchu1@gsuad.gsu.edu@snowball TEXT]$ cd
[[kguchu1@gsuad.gsu.edu@snowball ~]$ tar -czvpf TEXT.tar.gz TEXT
TEXT/
TEXT/file3_copy.txt
TEXT/file2_copy.txt
TEXT/file1_copy.txt
[kguchu1@gsuad.gsu.edu@snowball ~]$
```

```
[[kguchu1@gsuad.gsu.edu@snowball PDF]$ cd
 [kguchu1@gsuad.gsu.edu@snowball ~]$ ls
[copys     file1.txt  file3.txt  Lab3  PDF      simple.sh  test2.pdf  TEXT.tar.gz
[csc3320   file2.txt  homeworks  Lab4  public   test1.pdf  TEXT
 [kguchu1@gsuad.gsu.edu@snowball ~]$
```