

## 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: Kevin Gallardo

Campus ID: kgallardowepster1

Panther #: 002-49-7849

## PART 1 (2.5 points each): 10pts

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

```
A test
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cat > test.txt
this is a tasty test
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cat test.txt
this is a tasty test
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ grep 't\(a|e\)st' test.txt
this is a tasty test
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ egrep 't(a|e)st' test.txt
this is a tasty test
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ fgrep test
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ fgrep test test.txt
this is a tasty test
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```

Grep supports basic regular expressions, so you must escape meta-characters.

Egrep provides more support for meta characters, so they are escaped by default.

Fgrep supports fix strings only.

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 zip utility is used to compress and decompress files.

zip [options] zipfile list\_of\_files

zip file.zip file1.txt file2.txt file3.txt

unzip file.zip

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.

The utility `cut` can break a line into multiple fields by defining a separator.

The default separator for the `cut` command is the space " ".

```
cut -d ":" -f file_name.txt
```

The utility `awk` can break a line into multiple fields by defining a separator.

The default separator for the `awk` command is the space " ".

```
Awk -F separator file_name.txt
```

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

The "sort" command sorts lines of text files.

- `sort -b`: Ignore blanks at the start of the line.
- `sort -r`: Reverse the sorting order.
- `sort -o`: Specify the output file.
- `sort -n`: Use the numerical value to sort.
- `sort -M`: Sort as per the calendar month specified.
- `sort -u`: Suppress lines that repeat an earlier key.
- `sort -k POS1, POS2`: Specify a key to do the sorting.
- `sort -t SEP`: Use the provided separator to identify the fields.

**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!!!

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ man sort
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ echo 'Hello World' | sed 's/$/!!!/g'
Hello World!!!
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```

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

-- 1 <= NF { print \$5 }

Prints the fifth item of each line in the file.

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

Prints the first item of each line in file where row number >=1 and >=5.

-- 1,5 { print \$0 }

Prints the current line.

-- {print \$1 }

Prints all the first fields of each line.

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

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?

**sed -i 1,5d foo**

**sed -i '\$(( \$(wc -l < foo) -5 +1 )), \$ 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**

It searched for the rows where there is a string with a word that contains "ing" in it.

From the data in the file, the rows that contains word with "ing" are 1, 3, 4

Then it prints the format of "row number" : "first word in that row".

output is:

1:Wish

3:When

4:Now

```
kgallardowepster1@gsuad.gsu.edu@snowball:~  
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ 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...  
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cat h1.awk  
NR>2 && NR<4{print NR ":" $0}  
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ awk '/.*ing/ {print NR ":" $1}' float  
1:Wish  
3:When  
4:Now  
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```

## 10. As the next command following question 9, **\$ awk -f h1.awk float**

It executes the code in the file h1.awk on file named float.

The command **NR>2 && NR<4{print NR":"\$0}** executed on the data in float.

It represents the row number 3.

It prints in the format "row number" : "the text in the row".

Output:

3:When everything seemed so clear,

```
float
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ awk -f h1.awk float
3:When everything seemed so clear.
[kgallardowepster1@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**

It executes the code in the file h2.awk on file named float.

Scans the file and prints the script "Start to scan file" first word of the line, adds a coma and prints the last word in the line. And in the end adds "END-" , FILENAME

The output will be:

Start to scan file

Wish,is

Strong,,days

When,clear.

Now,all...

End- float

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ awk -f h2.awk float
Start to scan file
Wish,is
strong,,days
When,clear.
Now,all...
END- float
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```

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

Command executes on file named float and it replaces the spaces between words to tab

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ sed 's/\s/\t/g' 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...
[kgallardowepster1@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 1<sup>st</sup> field )

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ls *.awk | awk '{print "grep --color 'BEGIN' " $1 }' | sh
BEGIN { print "Start to scan file" } {print $1 " ", $NF}
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```

## 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

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cd test
[kgallardowepster1@gsuad.gsu.edu@snowball test]$ ls -l . | grep '^d' | awk '{print "cp -r " $NF " " $NF ".bak"}' | sh
[kgallardowepster1@gsuad.gsu.edu@snowball test]$
```

**Part III Programming: 15pts**

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

- a. A copy of each file in that folder must be made. Append the string “\_copy” to the name of the file

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cp file.txt txtfiles/file.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cp fn.txt txtfiles/fn.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cp mandatabase.txt txtfiles/mandatabase.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cp test.txt txtfiles/test.txt_copy
[kgallardowepster1@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).

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ls
csc3320 file.txt fn.txt homeworks Lab3 Lab4 mandatabase.txt midterm public test test.txt txtfiles
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cd txtfiles/
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ ls
ad-bk.txt_copy book.txt_copy file.txt_copy fn.txt_copy mandatabase.txt_copy phone.txt_copy test.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ pwd
/home/kgallardowepster1/txtfiles
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$
```

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

```
ad-bk.txt_copy book.txt_copy file.txt_copy fn.txt_copy mandatabase.txt_copy phone.txt_copy test.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ sort -M ad-bk.txt_copy
Kevin Gallardo ; 4046106456 ; 7137 Silver Mine Xing Austell GA 30168
Kevin Wepster ; 4046106457 ; 1268 Wisteria Dr Sw Mableton Ga 30126
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ sort -M ad-bk.txt_copy > ad-bk.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ sort -M book.txt_copy > book.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ sort -M file.txt_copy > file.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ sort -M fn.txt_copy > fn.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ sort -M mandatabase.txt_copy > mandatabase.txt_copy
sort: cannot read: _M: No such file or directory
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ sort -M mandatabase.txt_copy > mandatabase.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ sort -M phone.txt_copy > phone.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ sort -M test.txt_copy
this is a tasty test
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$ sort -M test.txt_copy > test.txt_copy
[kgallardowepster1@gsuad.gsu.edu@snowball txtfiles]$
```

- 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.



```
txtfiles/txtfiles.tar.gz
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ls
csc3320  fn.txt      Lab3  mandatabase.txt  public  test.txt  txtfiles.tar.gz
file.txt homeworks  Lab4  midterm          test    txtfiles
[kgallardowepster1@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.