

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: Zoe Boyle

Campus ID: zboyle1

Panther #: 900903403

PART 1

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

Grep uses BREs to search for a string. Egrep, extended grep, can use EREs as well as BREs. Fgrep, fixed grep, fixed grep, treats all metacharacters and regex characters as literals.

Example:

grep 'a|b' file > finds a|b

egrep 'a|b' file > finds a or b

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.

tar or tape archive can store files and can extract files from its archive

Compress files: create a tar file and add the files to be archived Example: tar -cvf tarFile fileList

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 and sort can break lines into fields. The default separator is a space or tab. The -t command followed by another character can manually define a separator.

Example: -t, to make a comma the separator

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

The sort command sorts one or more fields alphabetically in ascending order by default. The fields are numbered by their column position with the first column starting at 0.

Example: sort +0 -3 file

PART 2 a

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

Hello World!!!

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

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

Number of fields greater than or equal to 1, print field 5

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

If the line number is greater than or equal to one, and greater than or equal to 5, print the first field

-- 1,5 { print \$0 }

Prints lines 1-5

-- {print \$1 }

Prints the first field 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?

awk '/+\$/ {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 lines: sed '5, d' foo

Last lines: sed '5, \$d' foo

PART 2 b

10. **\$ 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

Function: for lines containing a string matched by the regex `/.*/ing/`, print the line number followed by a colon and the first word of each sentence

Output:

1:Wish

3:When

4:Now

11. As the next command following question 9,

\$ awk -f h1.awk float

Function: For the lines greater than two and less than 4, print the line number a colon and then the whole line

Output:

3. When everything seemed so clear.

12.

\$ cat h2.awk

BEGIN { print "Start to scan file" }

{print \$1 "," \$NF}

END {print "END-" , FILENAME }

awk -f h2.awk float

Function: before the first line executes, print "Start to scan file", then print the first field of each line followed by the last field with a comma in between. After the last line print END- and the filename

Output:

Start to scan file

Wish, strong,

And,days

When,clear.

Now,all...

13. sed 's/s/t/g' float

Function: Replaces every space with a tab

Output:

```
[zboyle1@gsuad.gsu.edu@snowball ~]$ sed 's/s/t/g' float
Wish I was floating in blue across the sky, my imagination is stron
g,
And I often visit the days
When everything seemed so clear.
Now I wonder what I'm doing here at all...[zboyle1@gsuad.gsu.edu@snowball ~]$
```

14. \$ ls *.awk | awk '{print "grep --color 'BEGIN' " \$1 }' | sh

Function: Lists all .awk files, pipes the files found to the awk command. The awk command prints a grep command to be piped to the sh command. This prints a line with 'BEGIN', making 'BEGIN' print in red

Output:

```
[zboyle1@gsuad.gsu.edu@snowball ~]$ ls *.awk | awk '{print "grep --color 'BEGIN' " $1 }' | sh
BEGIN {print "Start to scan file"}
13.
```

15. \$ mkdir 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

Function: lists the contents and properties of the contents in the test directory. This output is piped into grep which finds files without 'd', the output of the grep match is piped into awk. The awk command prints the cp command using -r and the last field from the grep output to copy a new file with .bak as the extension

Output:

```
[zboyle1@gsuad.gsu.edu@snowball test]$ ls
test1 test1.bak test2 test2.bak testt.txt
[zboyle1@gsuad.gsu.edu@snowball test]$
```

PART 3

16. 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

```
[zboyle1@gsuad.gsu.edu@snowball ~]$ ls
checkError.sh float hello.sh Lab3 public simple.sh test3.txt
csc3320 h1.awk homeworks Lab4 Result test2.txt test.txt

[zboyle1@gsuad.gsu.edu@snowball ~]$ ls
checkError_copy.sh float h1_copy.awk homeworks public simple_copy.sh test2.txt test_copy.txt
checkError.sh float_copy hello_copy.sh Lab3 Result simple.sh test3_copy.txt test.txt
csc3320 h1.awk hello.sh Lab4 Result_copy test2_copy.txt test3.txt
```

- 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).

```
[zboyle1@gsuad.gsu.edu@snowball ~]$ ls
awkfiles csc3320 h1.awk homeworks Lab4 public shfiles test2.txt test.txt
checkError.sh float hello.sh Lab3 otherfiles Result simple.sh test3.txt txtfiles
```

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

```
[zboyle1@gsuad.gsu.edu@snowball ~]$ ls -lt awkfiles
total 0
-rw-rw-r--. 1 zboyle1@gsuad.gsu.edu zboyle1@gsuad.gsu.edu 0 Oct 3 12:50 h1_copy.awk
[zboyle1@gsuad.gsu.edu@snowball ~]$ ls -lt shfiles
total 12
-rwxrwxr-x. 1 zboyle1@gsuad.gsu.edu zboyle1@gsuad.gsu.edu 126 Oct 3 12:49 simple_copy.sh
-rwxrwxr-x. 1 zboyle1@gsuad.gsu.edu zboyle1@gsuad.gsu.edu 393 Oct 3 12:49 hello_copy.sh
-rwxrwxr-x. 1 zboyle1@gsuad.gsu.edu zboyle1@gsuad.gsu.edu 621 Oct 3 12:48 checkError_copy.sh
[zboyle1@gsuad.gsu.edu@snowball ~]$ ls -lt txtfiles
total 8
-rw-rw-r--. 1 zboyle1@gsuad.gsu.edu zboyle1@gsuad.gsu.edu 45 Oct 3 12:51 test_copy.txt
-rw-rw-r--. 1 zboyle1@gsuad.gsu.edu zboyle1@gsuad.gsu.edu 18 Oct 3 12:51 test2_copy.txt
-rw-rw-r--. 1 zboyle1@gsuad.gsu.edu zboyle1@gsuad.gsu.edu 0 Oct 3 12:50 test3_copy.txt
[zboyle1@gsuad.gsu.edu@snowball ~]$ ls -lt otherfiles
total 8
-rw-rw-r--. 1 zboyle1@gsuad.gsu.edu zboyle1@gsuad.gsu.edu 49 Oct 3 12:51 Result_copy
-rw-rw-r--. 1 zboyle1@gsuad.gsu.edu zboyle1@gsuad.gsu.edu 172 Oct 3 12:48 float_copy
[zboyle1@gsuad.gsu.edu@snowball ~]$
```

- 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.

```
[zboyle1@gsuad.gsu.edu@snowball archive]$ ls|sort
awkfiles.tar
otherfiles.tar
shfiles.tar
txtfiles.tar
[zboyle1@gsuad.gsu.edu@snowball archive]$ ls
awkfiles.tar  otherfiles.tar  shfiles.tar  txtfiles.tar
[zboyle1@gsuad.gsu.edu@snowball archive]$ cd ~
[zboyle1@gsuad.gsu.edu@snowball ~]$ tar -cvf archive.tar archive
archive/
archive/otherfiles.tar
archive/shfiles.tar
archive/awkfiles.tar
archive/txtfiles.tar
[zboyle1@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.