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.

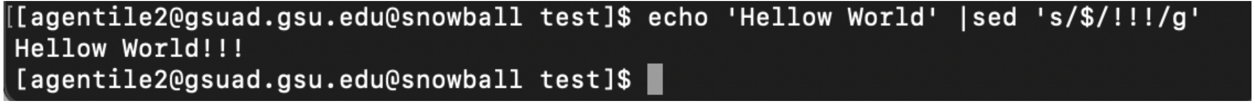
Full Name: Alexandra Gentile
Campus ID: agentile2
Panther #: 002413770

Part 1:

1. Grep is a basic regular expression that typically finds a particular word from the file and uses basic meta characters; an example would be finding the lines containing the pattern ‘sw.*ing’ using grep will return “swimming”, egrep is an extended regular expression that uses the same concept as grep but is able to use more and understand an extended version of regular expression patterns and meta character in the command; an example being finding the lines containing the pattern /s.+w/ using egrep will return “s will sw”, fgrep is a fixed string that doesn’t typically follow regular expression patterns or meta characters and typically searches for a direct string within the file.
2. The utility that can be used to compress and decompress files is compress/uncompress or gzip/gunzip. To zip multiple files into one file you could use command zip file1.zip folder1/folder2/folder3 that compresses all of the folders into the file “file1.zip”
3. The awk utility can break a line into multiple fields by defining a separator. The default separator for awk are tabs/spaces. To define a separate manually, the -F is used. For example: cat p1.awk { print $1 “ “ $6 “ “ $7 } awk -F: -f p1.awk will then execute the command
4. The sort command sorts a file in descending or ascending order on the initial bases of one or more sort fields. -r specifies a descending order, -f causes sort to ignore the case of the field, -M sorts the field in a month order, -n sorts the field in a numeric order, -b ignores leading spaces, -t is used to specify a different field separator. For example, $ cat sortfile will list the file to be sorted, $ sort sortfile will then sort the file, and $ sort -r sortfile will sort the file in reverse order.

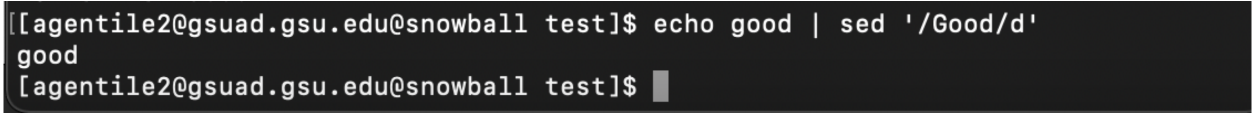
Part 2a:

1. The output is Hello World!!!



```
[[agentile2@gsuad.gsu.edu@snowball test]$ echo 'Hellow World' |sed 's/$/!!!/g'
Hellow World!!!
[agentile2@gsuad.gsu.edu@snowball test]$ 
```

2. – 1 <= NF { print $5 } prints the first 5 lines of the file
   - NR >= 1 && NR >= 5 { print $1 } prints the lines that are greater than 5
   - 1,5 { print $0 } prints out all of the file contents
   - { print $1 } prints out all of the first line
3. The output is “good”



```
[[agentile2@gsuad.gsu.edu@snowball test]$ echo good | sed '/Good/d'
good
[agentile2@gsuad.gsu.edu@snowball test]$ 
```

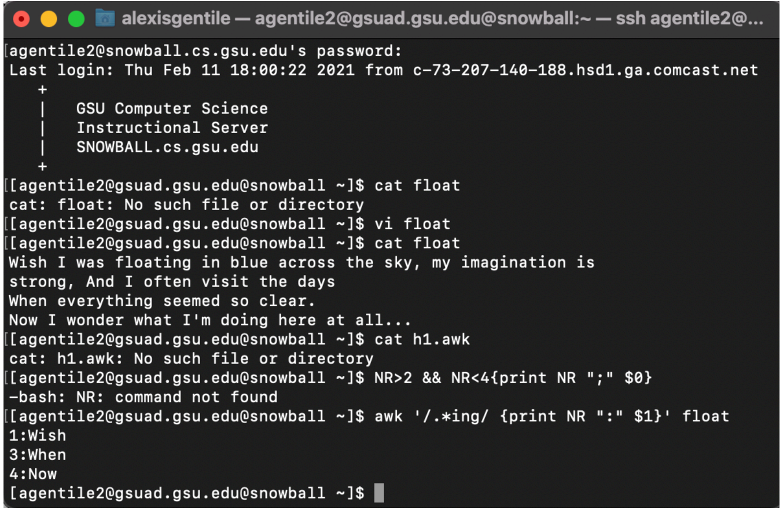
4. /\+$/{print $0}
5. Sed -i ‘1,5d’ foo only deletes the first five lines of the file, foo and sed -i ‘$5d’ foo will delete the last 5 lines

Part 2b.

1. The $ cat float command displays the file named “float” contents onto the screen. The command $ cat h1.awk NR>2 && NR<4{print NR “:” $0} this command does not output anything yet because the awk command wasn’t executed for it, but it searches for the lines between the row number greater than 2 and less and 4, printing the row number, ‘:’, and it’s contents.

   $ awk ‘/.*ing/ {print NR “:’ $1} float command searches through the file for the lines containing “ing” and prints out the row numbers AND the first word in that row therefore the output follows as 1: Wish 2: When 3: Now . this is the output because within these lines contains a word with “ing”.

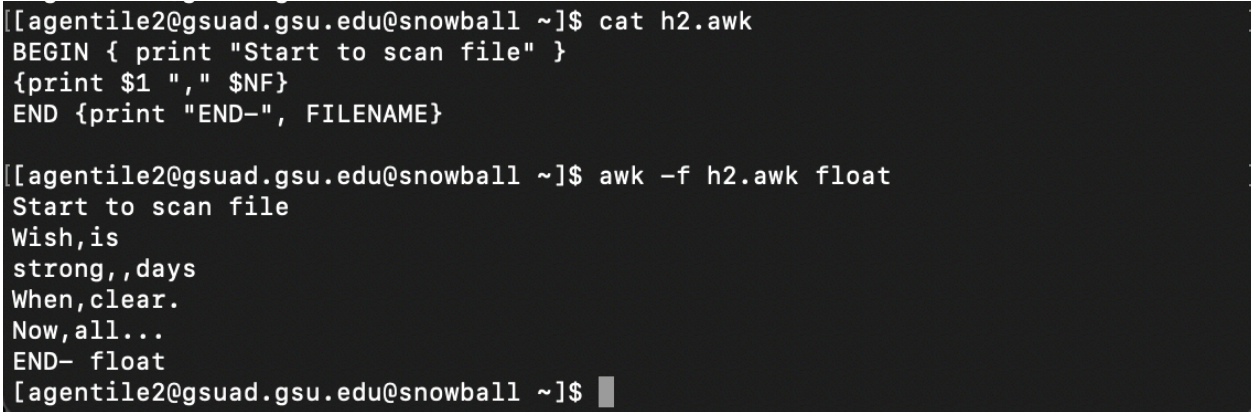
2. $ awk -f h1.awk float executes the command NR>2 && NR<4{print NR “:” $0} which searches and prints the row numbers between 2 and 4 and therefore prints 3:When everything seemed so clear,


alexisgentile — agentile2@gsuad.gsu.edu@snowball:~ — ssh agentile2@...

```
[agentile2@snowball.cs.gsu.edu's password:
 Last login: Thu Feb 11 18:00:22 2021 from c-73-207-140-188.hsd1.ga.comcast.net
    +
    |   GSU Computer Science
    |   Instructional Server
    |   SNOWBALL.cs.gsu.edu
    +
[[agentile2@gsuad.gsu.edu@snowball ~]$ cat float
 cat: float: No such file or directory
[[agentile2@gsuad.gsu.edu@snowball ~]$ vi float
[[agentile2@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...
[[agentile2@gsuad.gsu.edu@snowball ~]$ cat h1.awk
 cat: h1.awk: No such file or directory
[[agentile2@gsuad.gsu.edu@snowball ~]$ NR>2 && NR<4{print NR ";" $0}
 -bash: NR: command not found
[[agentile2@gsuad.gsu.edu@snowball ~]$ awk '/.*ing/ {print NR ":" $1}' float
 1:Wish
 3:When
 4:Now
 [agentile2@gsuad.gsu.edu@snowball ~]$ █
```

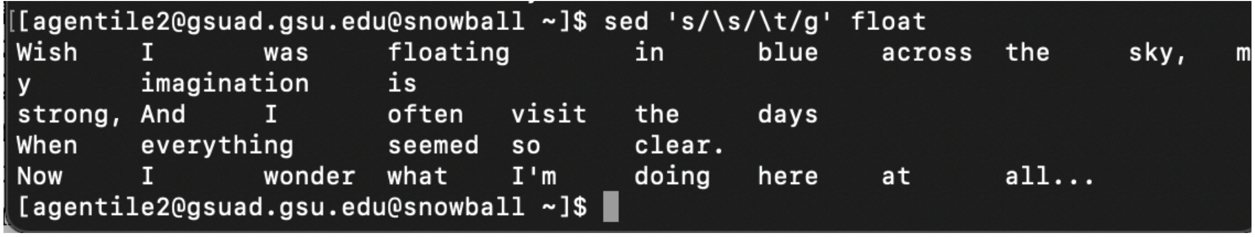
3. $ cat h2.awk and $ awk -f h2.awk float

The cat h2.awk command allows you to type in the commands BEGIN { print “Start to scan file” } {print $1 “,” $NF} End {print “End-“, FILENAME} which the BEGIN is a special token that tells the system to do a specific action. In this case, to print “Start to scan file”. {print $1 “,” $NF} then fetches every line of float and prints out the first word of every line, separated by commas, and the last word of every line. Then the last special token is implemented “END” which prints “END-“ with the filename(float). This command is then executed and printed to the screen through the command $ awk -f h2.awk float.



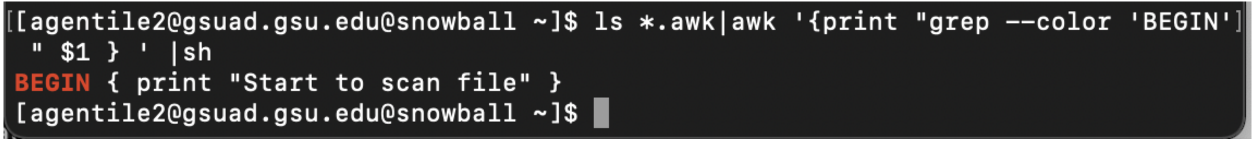
```
[[agentile2@gsuad.gsu.edu@snowball ~]$ cat h2.awk
BEGIN { print "Start to scan file" }
{print $1 "," $NF}
END {print "END-", FILENAME}

[[agentile2@gsuad.gsu.edu@snowball ~]$ awk -f h2.awk float
Start to scan file
Wish,is
strong,,days
When,clear.
Now,all...
END- float
[agentile2@gsuad.gsu.edu@snowball ~]$
```

4. Sed ‘s/\s/\t/g’ float tabs between every word in the file float.



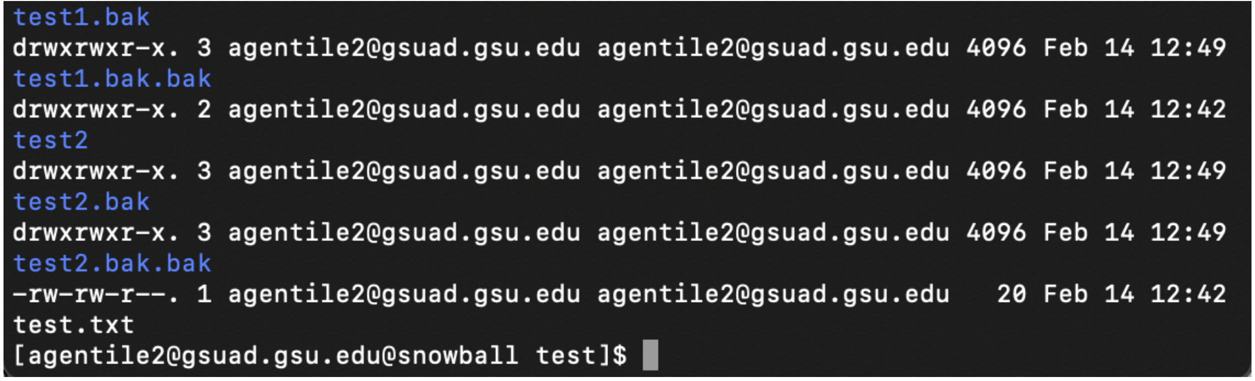
```
[[agentile2@gsuad.gsu.edu@snowball ~]$ sed 's/\s/\t/g' float
Wish    I       was     floating        in      blue    across  the     sky,    m
y       imagination     is
strong, And     I       often   visit   the     days
When    everything      seemed  so      clear.
Now     I       wonder  what    I'm     doing   here    at      all...
[agentile2@gsuad.gsu.edu@snowball ~]$
```

5. $ ls *.awkl awk ‘{print “grep –color ‘BEGIN’ “ $1 } lsh



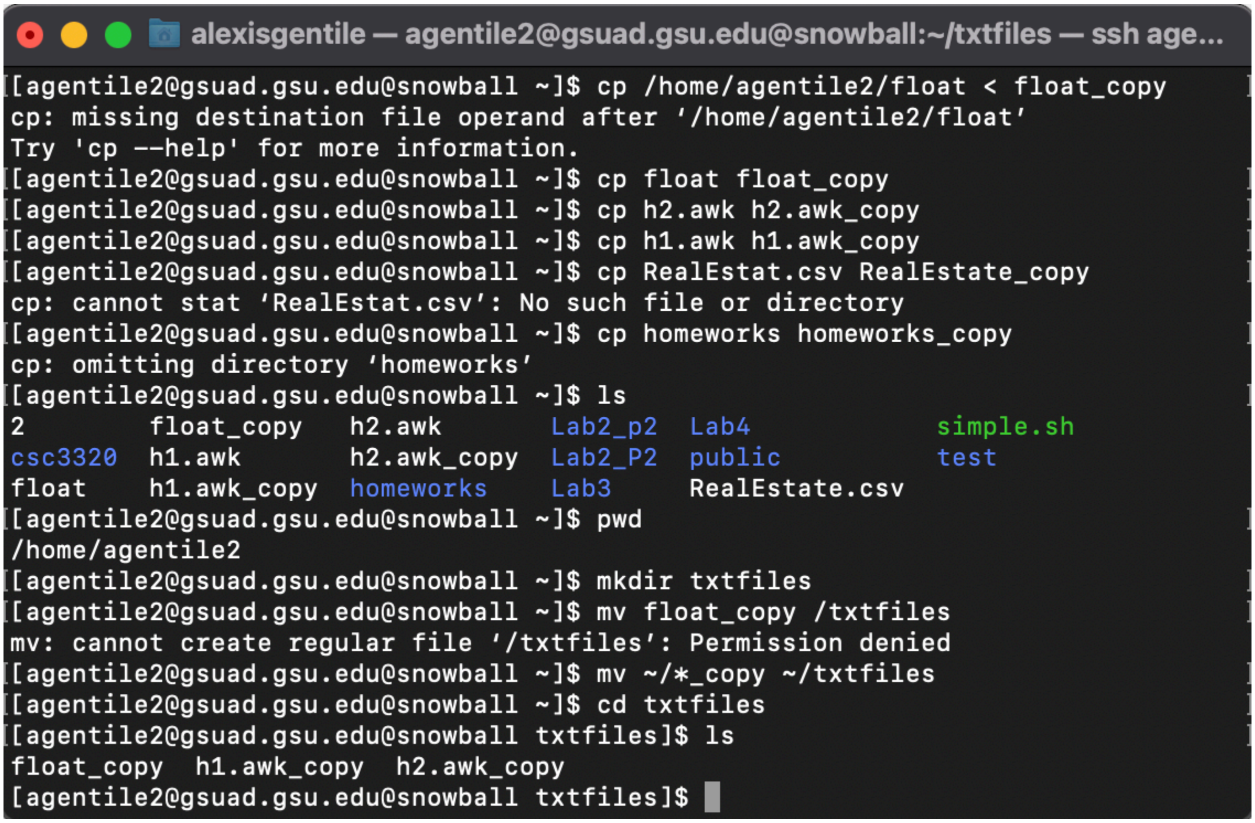
```
[[agentile2@gsuad.gsu.edu@snowball ~]$ ls *.awk|awk '{print "grep --color 'BEGIN']
 " $1 } ' |sh
BEGIN { print "Start to scan file" }
[agentile2@gsuad.gsu.edu@snowball ~]$
```

6. $ mkdir test test/test1 test/test2 $cat>test/test.txt This is a test file $ cd test $ ls -l . lgrep awk ‘{print “co -r “ $NF “ “ $NF “.bak”} l sh



```
test1.bak
drwxrwxr-x. 3 agentile2@gsuad.gsu.edu agentile2@gsuad.gsu.edu 4096 Feb 14 12:49
test1.bak.bak
drwxrwxr-x. 2 agentile2@gsuad.gsu.edu agentile2@gsuad.gsu.edu 4096 Feb 14 12:42
test2
drwxrwxr-x. 3 agentile2@gsuad.gsu.edu agentile2@gsuad.gsu.edu 4096 Feb 14 12:49
test2.bak
drwxrwxr-x. 3 agentile2@gsuad.gsu.edu agentile2@gsuad.gsu.edu 4096 Feb 14 12:49
test2.bak.bak
-rw-rw-r--. 1 agentile2@gsuad.gsu.edu agentile2@gsuad.gsu.edu   20 Feb 14 12:42
test.txt
[agentile2@gsuad.gsu.edu@snowball test]$
```

Part 3:


alexisgentile — agentile2@gsuad.gsu.edu@snowball:~/txtfiles — ssh age...

```
[agentile2@gsuad.gsu.edu@snowball ~]$ cp /home/agentile2/float < float_copy
cp: missing destination file operand after ‘/home/agentile2/float’
Try 'cp --help' for more information.
[agentile2@gsuad.gsu.edu@snowball ~]$ cp float float_copy
[agentile2@gsuad.gsu.edu@snowball ~]$ cp h2.awk h2.awk_copy
[agentile2@gsuad.gsu.edu@snowball ~]$ cp h1.awk h1.awk_copy
[agentile2@gsuad.gsu.edu@snowball ~]$ cp RealEstat.csv RealEstate_copy
cp: cannot stat ‘RealEstat.csv’: No such file or directory
[agentile2@gsuad.gsu.edu@snowball ~]$ cp homeworks homeworks_copy
cp: omitting directory ‘homeworks’
[agentile2@gsuad.gsu.edu@snowball ~]$ ls
2        float_copy   h2.awk       Lab2_p2  Lab4            simple.sh
csc3320  h1.awk       h2.awk_copy  Lab2_P2  public          test
float    h1.awk_copy  homeworks    Lab3     RealEstate.csv
[agentile2@gsuad.gsu.edu@snowball ~]$ pwd
/home/agentile2
[agentile2@gsuad.gsu.edu@snowball ~]$ mkdir txtfiles
[agentile2@gsuad.gsu.edu@snowball ~]$ mv float_copy /txtfiles
mv: cannot create regular file ‘/txtfiles’: Permission denied
[agentile2@gsuad.gsu.edu@snowball ~]$ mv ~/*_copy ~/txtfiles
[agentile2@gsuad.gsu.edu@snowball ~]$ cd txtfiles
[agentile2@gsuad.gsu.edu@snowball txtfiles]$ ls
float_copy  h1.awk_copy  h2.awk_copy
[agentile2@gsuad.gsu.edu@snowball txtfiles]$
```