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: Aparna Mandapaka

Campus ID: amandapaka2

Panther #: 002553239

PART 1 (2.5 points each): 10pts

1. What are the differences among ***grep***, ***egrep*** *and* ***fgrep***? Describe using an example.
    a. Before listing out the different, a similarity that **grep, egrep** and **fgrep** share is that they all come from the base **grep command.** Grep is considered as the combination of both the fgrep and egrep, the word grep means 'global regular expression print.' It supports the regular expression commands like the searching of words. An example of grep, a command to use grep is **grep [options] [pattern] [file]**, let's say for example we want to search a file in the directory, to do so we can use command grep. Like grep solution mathhomework , so basically the word **solution** is in the **mathhomework file**, so in order to find the word **solution** in the **mathhomework file**, we need to issue an command **grep solution mathhomework** .
        i. After executing this command, the grep will show all the lines that match with the word "**solution**" in the **mathhomework** file.

    b. Whereas the fgrep, means 'fixed string grep,' it doesn't understand regular or extended regular expression and also it doesn't support regex and uses direct string matching.
        i. An example of fgrep would be, if we want to show the count of number of matches, we can find the number of lines that would match the given string by **fgrep -c "usin.g" amandapaka** . The command outline that can be used to intiate the fgrep is **fgrep -inw 'pattern' {filename}***
    c. Finally the egrep, which means 'extended grep', it basically a command that allows the use of extended regular expression.
        i. An example of egrep would be searching for a specific string in a file and we can do this by **egrep because englishfile.txt** , so in the above command I typed in the word that I want to be searched for which is 'because' and also specified the file where the word should be looked for.

The command for the egrep file **is $egrep “search_string” filename**

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 utility that can be used to compress and decompress the files is the **tar** command. To compress the multiple files into a single file is **zip [options] zipfile files_list**
   An example would be **zip myfile.zip filename.txt filename1.txt filename2.txt** and we can use the command **unzip myfile.zip** to unzip all the files.

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 utilities can break a line into multiple fields by defining a separator are **awk** and **sed.** The default separator is **space for awk**. To define a separator manually in the command line with awk we can write **awk -F ‘{statement $linerule}** . An example for defining the separator for each utility would be **awk -F: ‘{print $1}’** , we want “:” to be our separator, then we can set “:” as a line separator for each word of the line like “;” .

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

   The **sort** command will sort a text file’s lines. The different possible fields such as **-n** for numeric, **-R** for random, **-r** for reverse sorting.
   An example of **sort file**

   ```
   sort file.txt
   e
   f
   g
   h
   sort -r file.txt
   ```

h

g

f

e

Part IIa (5 points each): 25pts

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

    a. **Output:** Hello World!!!

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

    -- 1 <= NF { print $5 }

    ⇒ **It prints the fifth item of each line in the file**

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

    ⇒ **After line 5, it prints the first word of each line**

    -- 1,5 { print $0 }

    ⇒ **Print $0 prints the current line instead of the first field of each line in the file**

    -- {print $1 }

    ⇒ **Print $1 prints all the first fields of each line in file instead of printing first field of first five lines**

7. What is the output of the 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} , the "/\" is used because it shouldn't be ignored.

9. What is the command to delete only the first 5 lines in a file "foo"? Which command deletes only the last 5 lines?

    command to delete only the first 5 lines in a file "foo"

    sed -l '1,5d' foo

The command to delete the only the last 5 lines
head -b -5 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

The above command is the structure of the file, which means that when we say NR>2 and NR<4, it will print only print line 3

$ awk '/.*ing/ {print NR ":" $1}' float

This command will print the first word of lines that having on the same line

10. As the next command following question 9,

$ awk -f h1.awk float

The awk file separator, gets data from both h1.awk and float and depending on the specifications of the h1.awk file. It will print the information between record 2 and 4 and not inclusive, which means 3

11.

$ cat h2.awk

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

END {print    "END-" , FILENAME }
```

$ awk -f h2.awk float

This whole command prints the "Start to scan file" and then print the endings and the starting words of the file float.

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

The s/ means substitute, \s means match whitespace, \t means insert a tab and \g means global (per line there's multiple ways that thing exists). The match regex whitespace with tabs gloablly.

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 *)*

The command above will find the word BEGIN and give it a color.

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
```

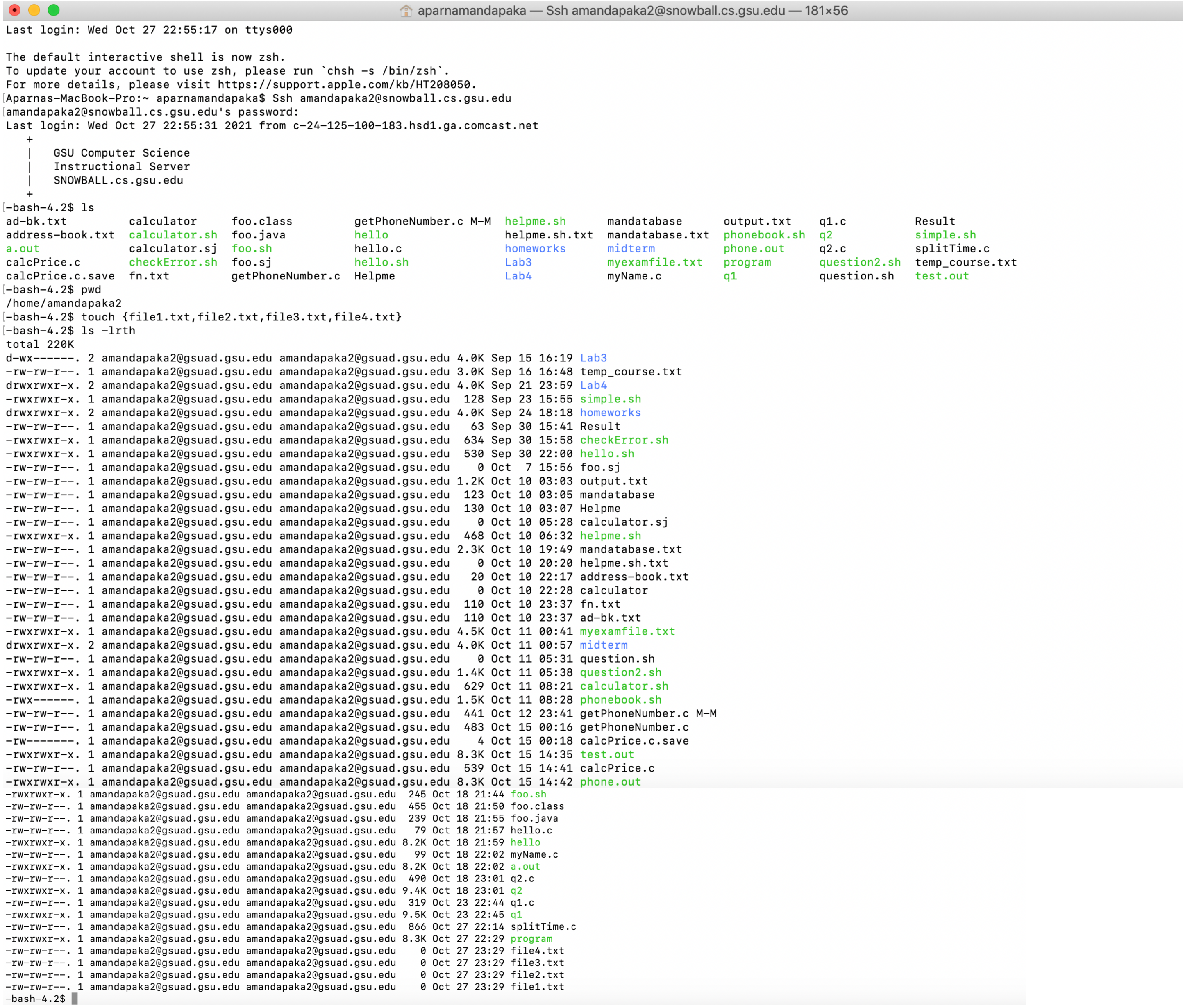
the command above will create something called backup directory, whoch will make test1 and test2 have the same names including the things inside of the directories, the names will be followed by test1.bak and test2.bak instead.

Part III Programming: 15pts

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

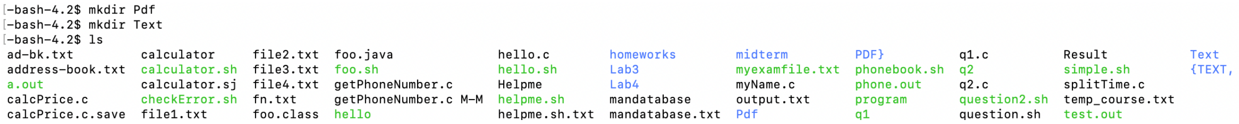
a. A copy of each file in that folder must be made. Append the string “_copy” to the name of the file
b. The duplicate (copied) files must be in separate directories with each directory specifying the type of the file (e.g. txt files in directory named txtfiles, pdf files in directory named pdf files 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.


aparnamandapaka — Ssh amandapaka2@snowball.cs.gsu.edu — 181×56

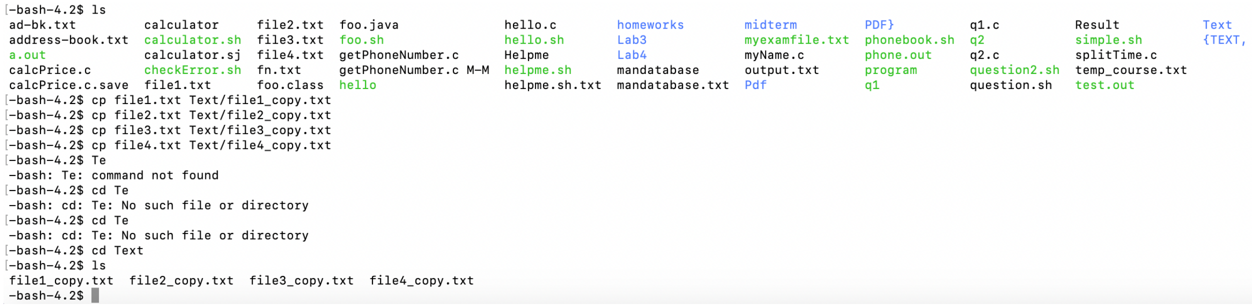
```
Last login: Wed Oct 27 22:55:17 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[Aparnas-MacBook-Pro:~ aparnamandapaka$ Ssh amandapaka2@snowball.cs.gsu.edu
[amandapaka2@snowball.cs.gsu.edu's password:
Last login: Wed Oct 27 22:55:31 2021 from c-24-125-100-183.hsd1.ga.comcast.net
    +
    |   GSU Computer Science
    |   Instructional Server
    |   SNOWBALL.cs.gsu.edu
    +
[-bash-4.2$ ls
ad-bk.txt         calculator     foo.class         getPhoneNumber.c M-M  helpme.sh      mandatabase      output.txt    q1.c          Result
address-book.txt  calculator.sh  foo.java          hello                 helpme.sh.txt  mandatabase.txt  phonebook.sh  q2            simple.sh
a.out             calculator.sj  foo.sh            hello.c               homeworks      midterm          phone.out     q2.c          splitTime.c
calcPrice.c       checkError.sh  foo.sj            hello.sh              Lab3           myexamfile.txt   program       question2.sh  temp_course.txt
calcPrice.c.save  fn.txt         getPhoneNumber.c  Helpme                Lab4           myName.c         q1            question.sh   test.out
[-bash-4.2$ pwd
/home/amandapaka2
[-bash-4.2$ touch {file1.txt,file2.txt,file3.txt,file4.txt}
[-bash-4.2$ ls -lrth
total 220K
d-wx------. 2 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 4.0K Sep 15 16:19 Lab3
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 3.0K Sep 16 16:48 temp_course.txt
drwxrwxr-x. 2 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 4.0K Sep 21 23:59 Lab4
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  128 Sep 23 15:55 simple.sh
drwxrwxr-x. 2 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 4.0K Sep 24 18:18 homeworks
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu   63 Sep 30 15:41 Result
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  634 Sep 30 15:58 checkError.sh
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  530 Sep 30 22:00 hello.sh
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu    0 Oct  7 15:56 foo.sj
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 1.2K Oct 10 03:03 output.txt
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  123 Oct 10 03:05 mandatabase
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  130 Oct 10 03:07 Helpme
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu    0 Oct 10 05:28 calculator.sj
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  468 Oct 10 06:32 helpme.sh
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 2.3K Oct 10 19:49 mandatabase.txt
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu    0 Oct 10 20:20 helpme.sh.txt
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu   20 Oct 10 22:17 address-book.txt
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu    0 Oct 10 22:28 calculator
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  110 Oct 10 23:37 fn.txt
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  110 Oct 10 23:37 ad-bk.txt
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 4.5K Oct 11 00:41 myexamfile.txt
drwxrwxr-x. 2 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 4.0K Oct 11 00:57 midterm
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu    0 Oct 11 05:31 question.sh
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 1.4K Oct 11 05:38 question2.sh
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  629 Oct 11 08:21 calculator.sh
-rwx------. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 1.5K Oct 11 08:28 phonebook.sh
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  441 Oct 12 23:41 getPhoneNumber.c M-M
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  483 Oct 15 00:16 getPhoneNumber.c
-rw-------. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu    4 Oct 15 00:18 calcPrice.c.save
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 8.3K Oct 15 14:35 test.out
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  539 Oct 15 14:41 calcPrice.c
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 8.3K Oct 15 14:42 phone.out
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  245 Oct 18 21:44 foo.sh
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  455 Oct 18 21:50 foo.class
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  239 Oct 18 21:55 foo.java
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu   79 Oct 18 21:57 hello.c
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 8.2K Oct 18 21:59 hello
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu   99 Oct 18 22:02 myName.c
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 8.2K Oct 18 22:02 a.out
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  490 Oct 18 23:01 q2.c
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 9.4K Oct 18 23:01 q2
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  319 Oct 23 22:44 q1.c
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 9.5K Oct 23 22:45 q1
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  866 Oct 27 22:14 splitTime.c
-rwxrwxr-x. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu 8.3K Oct 27 22:29 program
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu    0 Oct 27 23:29 file4.txt
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu    0 Oct 27 23:29 file3.txt
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu    0 Oct 27 23:29 file2.txt
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu    0 Oct 27 23:29 file1.txt
-bash-4.2$
```

a. Now lets create a directory for text files and pdf files



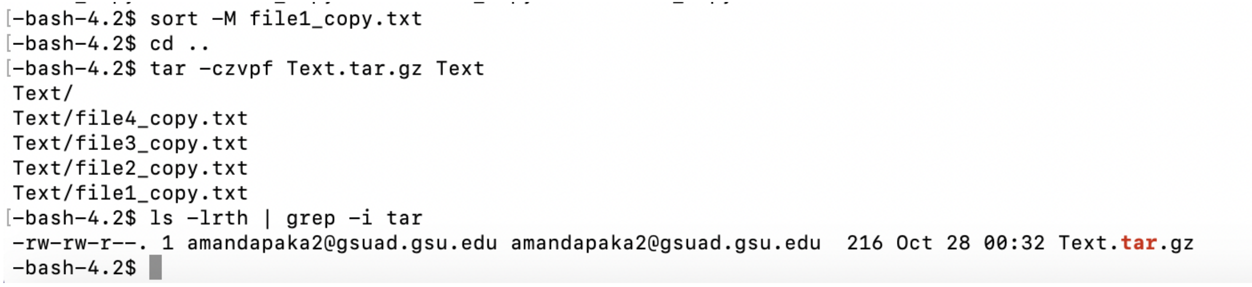
```
[-bash-4.2$ mkdir Pdf
[-bash-4.2$ mkdir Text
[-bash-4.2$ ls
ad-bk.txt         calculator     file2.txt  foo.java              hello.c        homeworks        midterm         PDF}          q1.c          Result           Text
address-book.txt  calculator.sh  file3.txt  foo.sh                hello.sh       Lab3             myexamfile.txt  phonebook.sh  q2            simple.sh        {TEXT,
a.out             calculator.sj  file4.txt  getPhoneNumber.c      Helpme         Lab4             myName.c        phone.out     q2.c          splitTime.c
calcPrice.c       checkError.sh  fn.txt     getPhoneNumber.c M-M  helpme.sh      mandatabase      output.txt      program       question2.sh  temp_course.txt
calcPrice.c.save  file1.txt      foo.class  hello                 helpme.sh.txt  mandatabase.txt  Pdf             q1            question.sh   test.out
```

b. Now we can put in the duplicate copy in separate directories



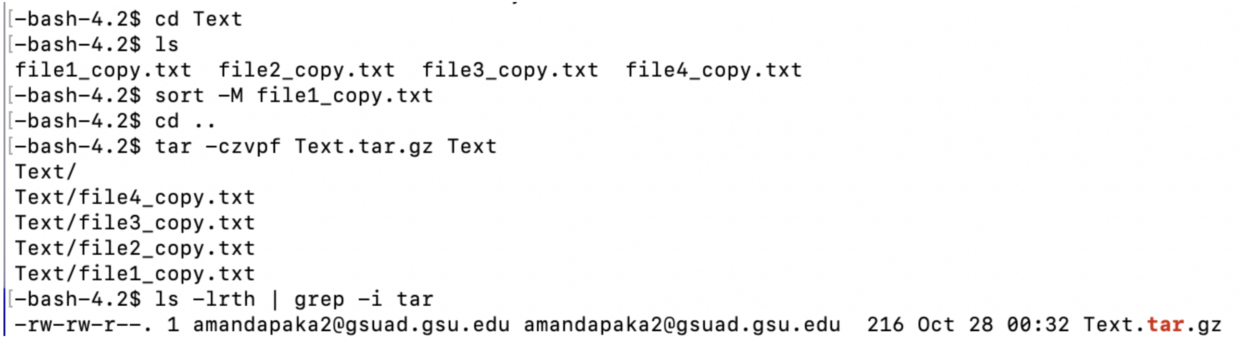
```
-bash-4.2$ ls
ad-bk.txt          calculator     file2.txt  foo.java             hello.c         homeworks         midterm          PDF}          q1.c          Result           Text
address-book.txt   calculator.sh  file3.txt  foo.sh               hello.sh        Lab3              myexamfile.txt   phonebook.sh  q2            simple.sh        {TEXT,
a.out              calculator.sj  file4.txt  getPhoneNumber.c     Helpme          Lab4              myName.c         phone.out     q2.c          splitTime.c
calcPrice.c        checkError.sh  fn.txt     getPhoneNumber.c M-M helpme.sh       mandatabase       output.txt       program       question2.sh  temp_course.txt
calcPrice.c.save   file1.txt      foo.class  hello                helpme.sh.txt   mandatabase.txt   Pdf              q1            question.sh   test.out
-bash-4.2$ cp file1.txt Text/file1_copy.txt
-bash-4.2$ cp file2.txt Text/file2_copy.txt
-bash-4.2$ cp file3.txt Text/file3_copy.txt
-bash-4.2$ cp file4.txt Text/file4_copy.txt
-bash-4.2$ Te
-bash: Te: command not found
-bash-4.2$ cd Te
-bash: cd: Te: No such file or directory
-bash-4.2$ cd Te
-bash: cd: Te: No such file or directory
-bash-4.2$ cd Text
-bash-4.2$ ls
file1_copy.txt  file2_copy.txt  file3_copy.txt  file4_copy.txt
-bash-4.2$
```

c. We are going to use the sort command to order the months in chronological order



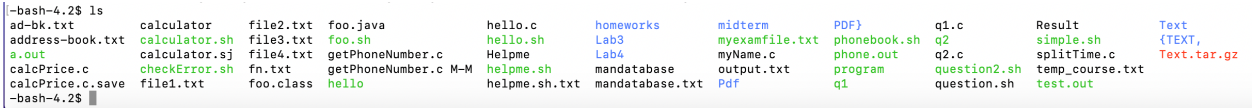
```
-bash-4.2$ sort -M file1_copy.txt
-bash-4.2$ cd ..
-bash-4.2$ tar -czvpf Text.tar.gz Text
Text/
Text/file4_copy.txt
Text/file3_copy.txt
Text/file2_copy.txt
Text/file1_copy.txt
-bash-4.2$ ls -lrth | grep -i tar
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  216 Oct 28 00:32 Text.tar.gz
-bash-4.2$
```

d. We can archive the files using the tar command



```
-bash-4.2$ cd Text
-bash-4.2$ ls
file1_copy.txt  file2_copy.txt  file3_copy.txt  file4_copy.txt
-bash-4.2$ sort -M file1_copy.txt
-bash-4.2$ cd ..
-bash-4.2$ tar -czvpf Text.tar.gz Text
Text/
Text/file4_copy.txt
Text/file3_copy.txt
Text/file2_copy.txt
Text/file1_copy.txt
-bash-4.2$ ls -lrth | grep -i tar
-rw-rw-r--. 1 amandapaka2@gsuad.gsu.edu amandapaka2@gsuad.gsu.edu  216 Oct 28 00:32 Text.tar.gz
```

e. Now there is a tar file in the home directory. To create the pdffiles archives you can use the command $ls -lrth | grep -I tar or you could use $tar -cvf thetarname.tar directoryname. All the tar files are in a tar archive Text.tar.gz



```
-bash-4.2$ ls
ad-bk.txt          calculator     file2.txt  foo.java             hello.c         homeworks         midterm          PDF}          q1.c          Result           Text
address-book.txt   calculator.sh  file3.txt  foo.sh               hello.sh        Lab3              myexamfile.txt   phonebook.sh  q2            simple.sh        {TEXT,
a.out              calculator.sj  file4.txt  getPhoneNumber.c     Helpme          Lab4              myName.c         phone.out     q2.c          splitTime.c      Text.tar.gz
calcPrice.c        checkError.sh  fn.txt     getPhoneNumber.c M-M helpme.sh       mandatabase       output.txt       program       question2.sh  temp_course.txt
calcPrice.c.save   file1.txt      foo.class  hello                helpme.sh.txt   mandatabase.txt   Pdf              q1            question.sh   test.out
-bash-4.2$
```