Homework# 2
CSC3320 System-Level Programming

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: Cassandra Lily Lundberg

Campus ID: clundberg3

Panther #: 002345582

Part 1:

1. Grep, egrep, and fgrep differ on the notion that each allows a different kind of text pattern to be matched. Grep allows for regular expression patterns to be searched for, egrep allows for extended regular expression patterns to be searched for, and fgrep allows for fixed string patterns only to be searched for. As egrep and fgrep are versions of grep, egrep is used when multiple patterns of regular expressions are needed. For example, the regular expression '^.ing' matches any pattern with the ending as "ing". In this case grep can be used, however if we were to search for a match of either string or wing, then the extended regular expression '(str|w)ing' would need to be used with egrep. Egrep takes the parenthesis and pipeline operation as metacharacters to mean search for either string or wing. Regular grep on the other hand would search for the entire expression '(str|w)ing' and would most likely not match. Fgrep, however, does not even recognize expressions and only performs fast searching for an entire fixed string. For example the command "fgrep 'hello'" would search for the fixed string 'hello' alone.

2. The tar and the gzip/gunzip utilities can be used to compress and decompress files. Although the gzip utility can be used to compress files and the gunzip utility can be used to decompress a file, the tar utility can compress and decompress large files. To compress multiple files into one archived file, first the tp archive needs to be created where the utility 'tar' can be used. For example, 'tar -cvf ArchivedFiles file1 file2 file3 file4'. This way all the files' storage are shortened, or "zipped up" to save room on a disk or for transportation. To decompress or extract the archived file or files, the "-x" option is used with tar.

3. The utility 'awk' breaks lines into multiple fields by defining a separator, usually by a tab or space. As the default field separator is a tab or space, there is a way to change the field separator. Using the option "-F'" followed by the preferred field separator in quotation marks, awk is able to specify between different fields. For example, "awk -F: fileName" will separate the file's field by colons instead of the defaulted spaces. Another utility that utilizes field separators is the "sort" utility. By default the sort utility also defines fields by spaces or tabs. The field separator preference can be changed by using the '-t' option followed by the new field separator. For example, "sort -t; filename" now sorts the file and divides the fields by each semicolon.

4. The "sort" utility or command sorts lines in one or more files by specific criteria such as, ascending/descending order, month order, and numerical order all on

the basis or field number. However the cases of characters and leading white spaces play a part in the specification. Different fields could contain uppercase and lowercase characters, numerical characters, specific words in which UNIX recognizes as months, or spaces or tabs. For example, if a line contained “Homework1: Question1:partA:Sectioniii:6”, the command ‘sort -t: filename’ would be needed to access each field. However, to sort based on, for instance, the 3rd field to the 5th field where case is ignored and numerical values are used, the command ‘sort -t: +2 -4 -b -n fileName’ would be needed.

Part 2a:

5) Hello World !!!

6) If the number of fields is greater than or equal to 1 then the first awk script command will output the fifth field of those specific lines.

If the current line number is greater than or equal to 1 and 5 then the second awk script command will output the first field.

One lines 1 and 5 print the entire line for the third awk script command.

The first field should be printed for the fourth awk script command.

7)The output is ‘good’ since the utility ‘sed’ is case sensitive; the ‘Good’ with a capital ‘G’ could not be deleted since there was no string matching it.

8) awk /+$/ {print $0}

9) sed ‘1,5d’ foo

sed ‘$d’ foo | sed ‘$d’ | sed ‘$d’ | sed ‘$d’ | sed ‘$d’

Part 2b:

9) The first function creates a file called float with text on 4 lines. The next function stores an awk function. That awk function prints the current line number then the character “:”, for every field if the line number is greater than 2 and less than 4. The awk script stating “awk ‘/.*ing/{print NR “:” $1}’ float”, takes the float text and matches a string with any character and any number of characters, ending with “ing”, then prints the line number, followed by a colon, followed by the first field of that line. Here the default remains where each field is separated by a space.

10) This awk function performed the awk function stored in the file “h1.awk” on the text file “float”. Since the awk function in “h1.awk” only prints the current line number then “:”, then the entire line for the 3rd line. This is because the third line is the only line greater than line number 2 and less than line number 4.

11) This group of code creates a text holding an awk function called h2.awk which first prints the string “Start to scan file” then prints the first field of each line, followed by a comma then the last field of each line. The awk function after this text is created, performs the awk function stored in h2.awk on the text file stored in float. The output portrays the first word of each line then “,” then the last word of each line.

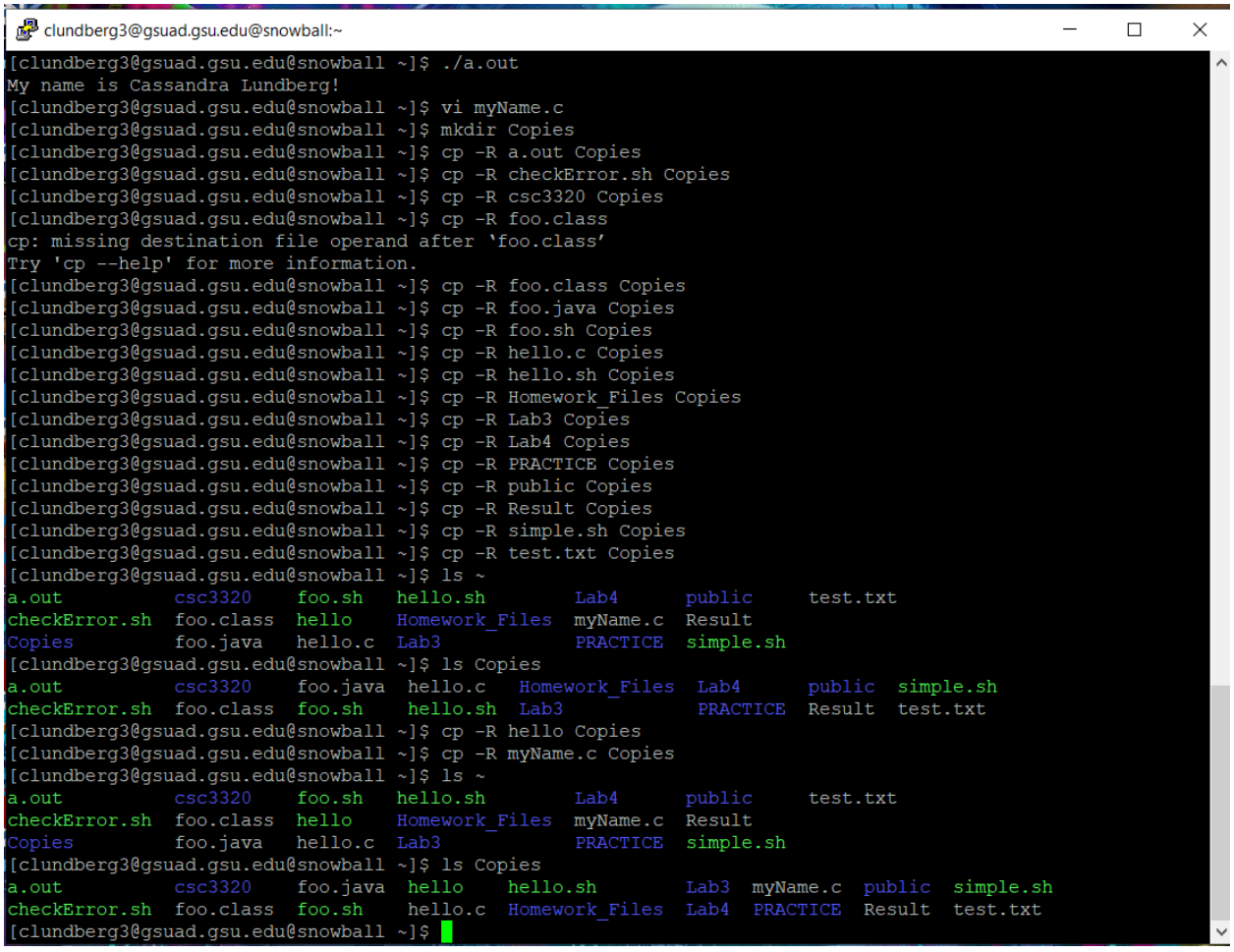
12) Since the character “\s” represents white space and the character “\t” represents tabs, this sed function replaces or substitutes all white spaces with tabs globally, in the text file float. So the output will be the same words contained in the text file float however each word will be separated by tabs instead of spaces.

13) In this command, the “ls” utility is listing every file with 0 or more characters ending with “.awk”. This part of the command is telling the system to search for every file ending in “.awk” and send that output to the input of the next pipe. The next command is an awk script that takes every file under the conditions of the previous command as input and prints the output ‘BEGIN’ special token in color using grep. Because the input of the second command are the two files h1.awk and h2.awk, and only the file h2.awk contains a BEGIN token, the action at that condition is “{print “Start to scan file”}”. Since the action at the condition of token ‘BEGIN’ is being piped to shell script, along with the colored string ‘BEGIN’, the action does not execute. Instead the colored string ‘BEGIN’ is output along with the action at the ‘BEGIN’ token of the file h2.awk.

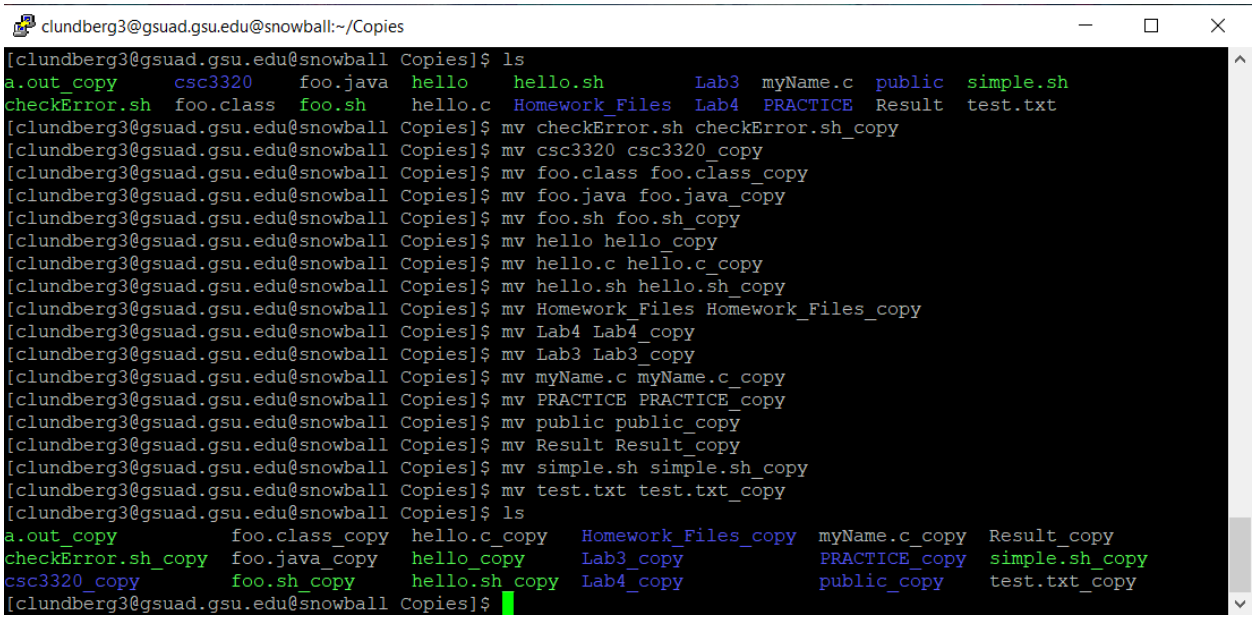
14) In these lines of commands, a directory named “test” is created along with subdirectories called “test1” and “test2”. Within this test directory, a file “testt.txt” is created with a string text stating: “This is a test file”. Then once in the test directory a command is typed to list all the files in the directory and take that output as a pipeline input to the next command. This command takes the list of files and directories, using grep, and filters out all the directories as output and sends the list of directories to the input of the next pipe. This pipe prints a recursive copy of the last field directory, otherwise known as the directory name then prints the string “.bak” at the end. Then this string “.bak” becomes the new last field.

Part 3:

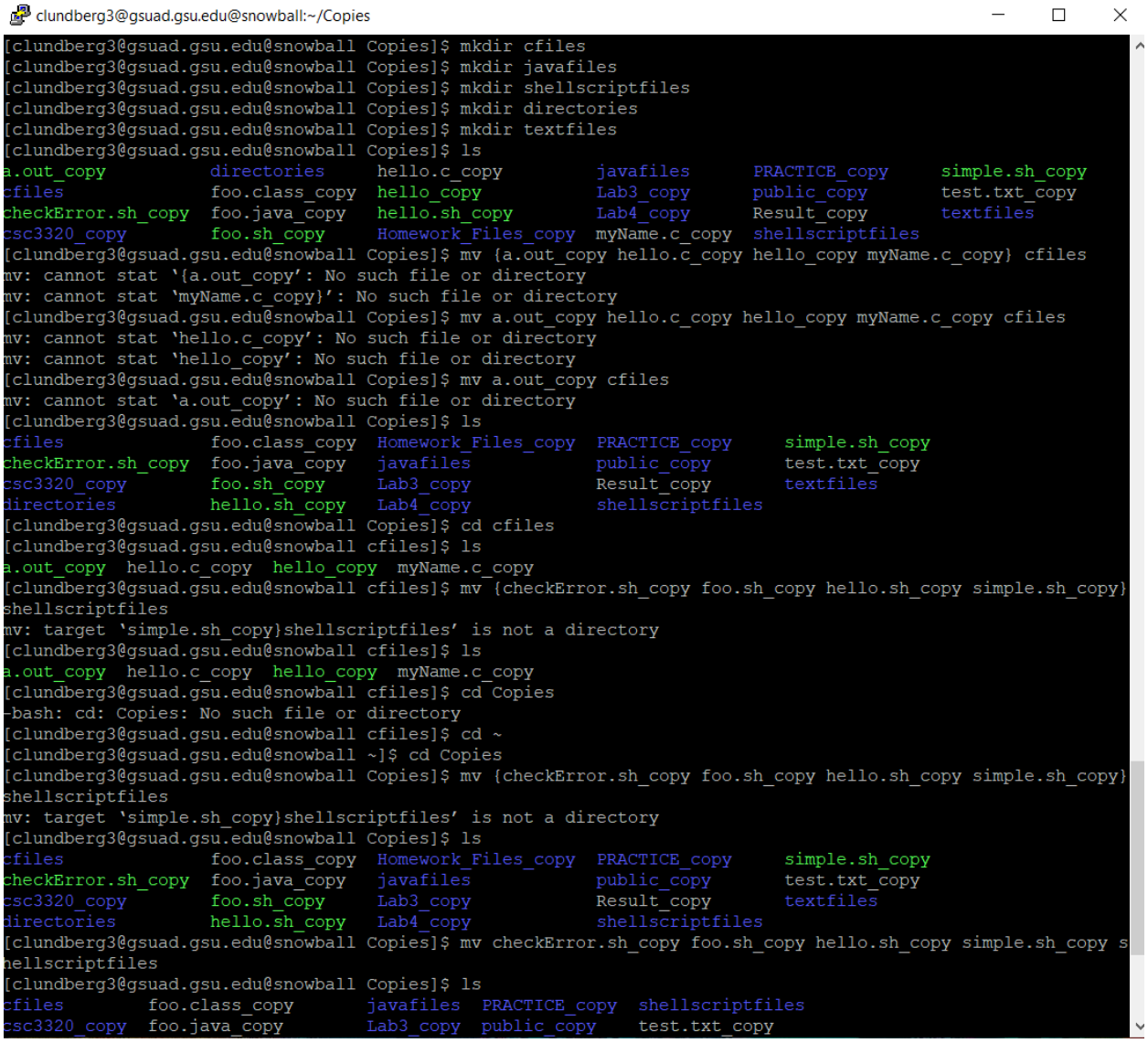
1) First I created a new directory to hold all the copies of every file in the home directory called “Copies” just for my sake to feel organized. Then I copied each file over to the new directory “Copies”. I presented a side by side of a list of all my files in the home directory and all the files existing in the new directory “Copies”.

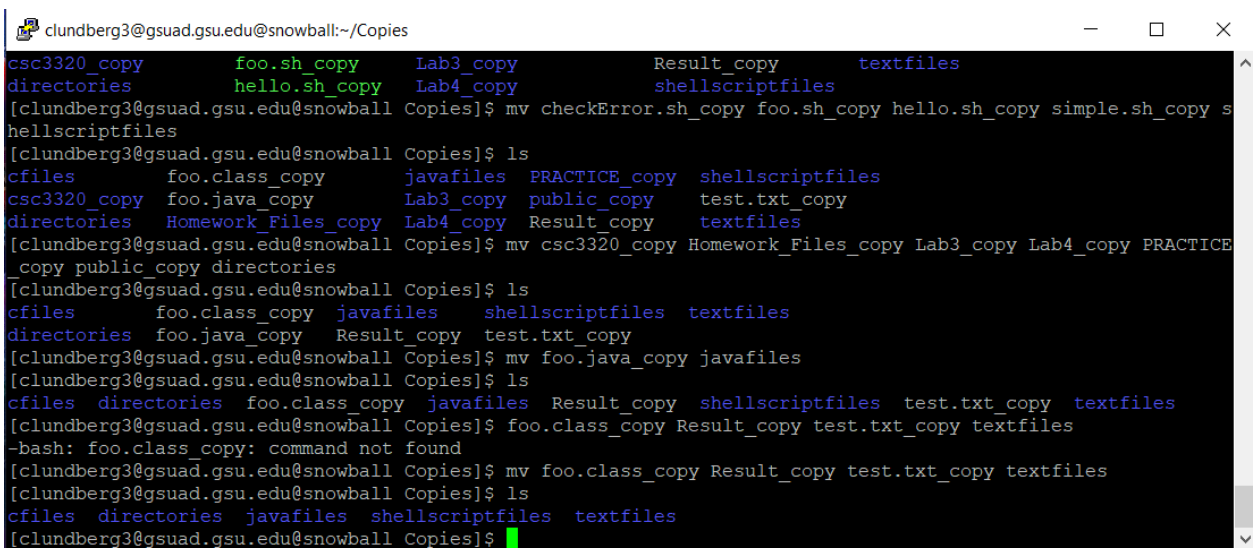


Next I took all the files in the directory “Copies and added the appending string “_copy to each of the names of the files.

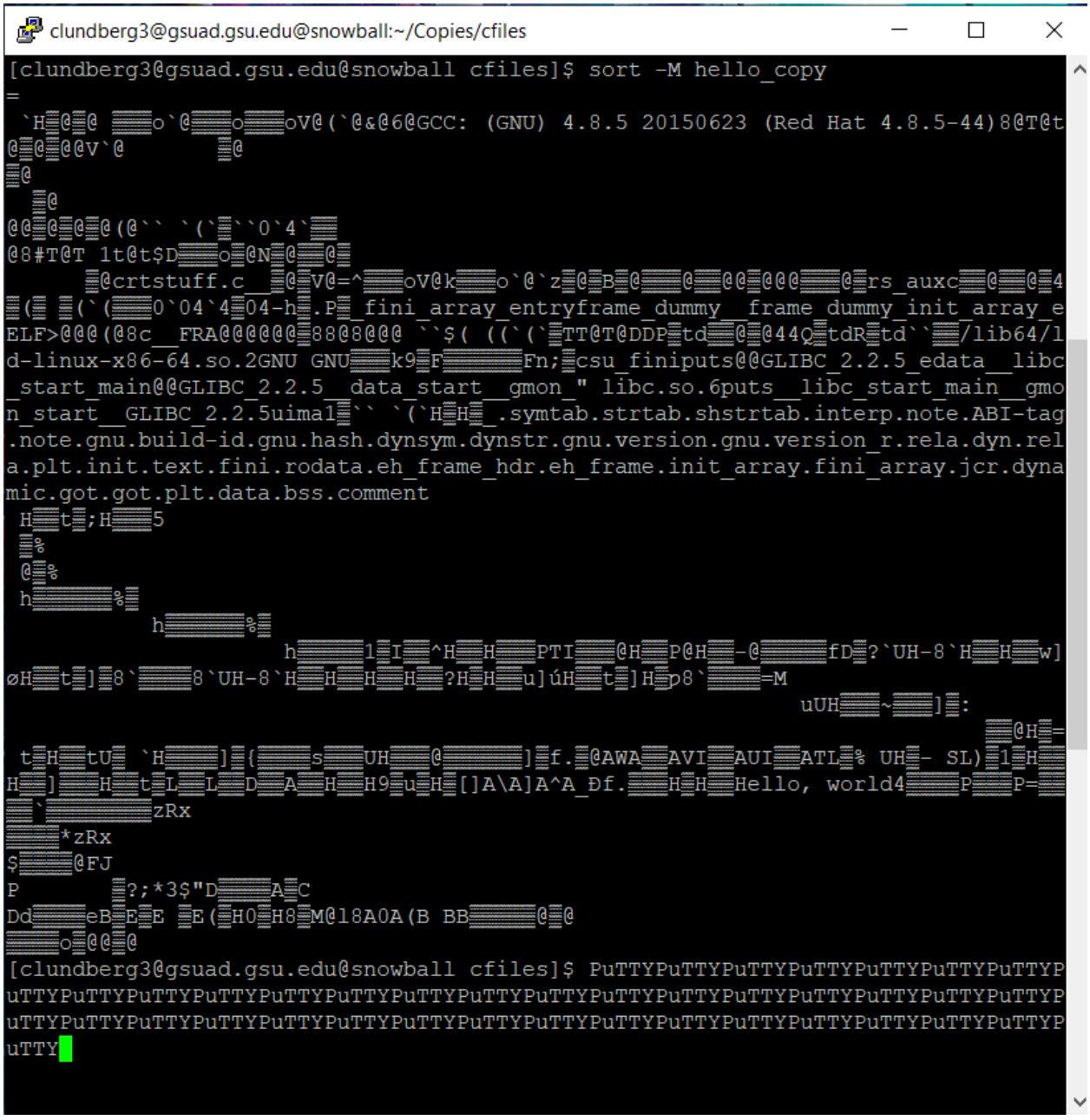


2) Once all the files are copied and appended with the string “_copy”, I create multiple directories inside the “Copy” directory holding all of the copied files. I created a directory for textfiles, shellscriptfiles, javafiles, cfiles, and directories. Then I used the “mv” utility in UNIX to move multiple files into the specified directories based on what type of file they were.

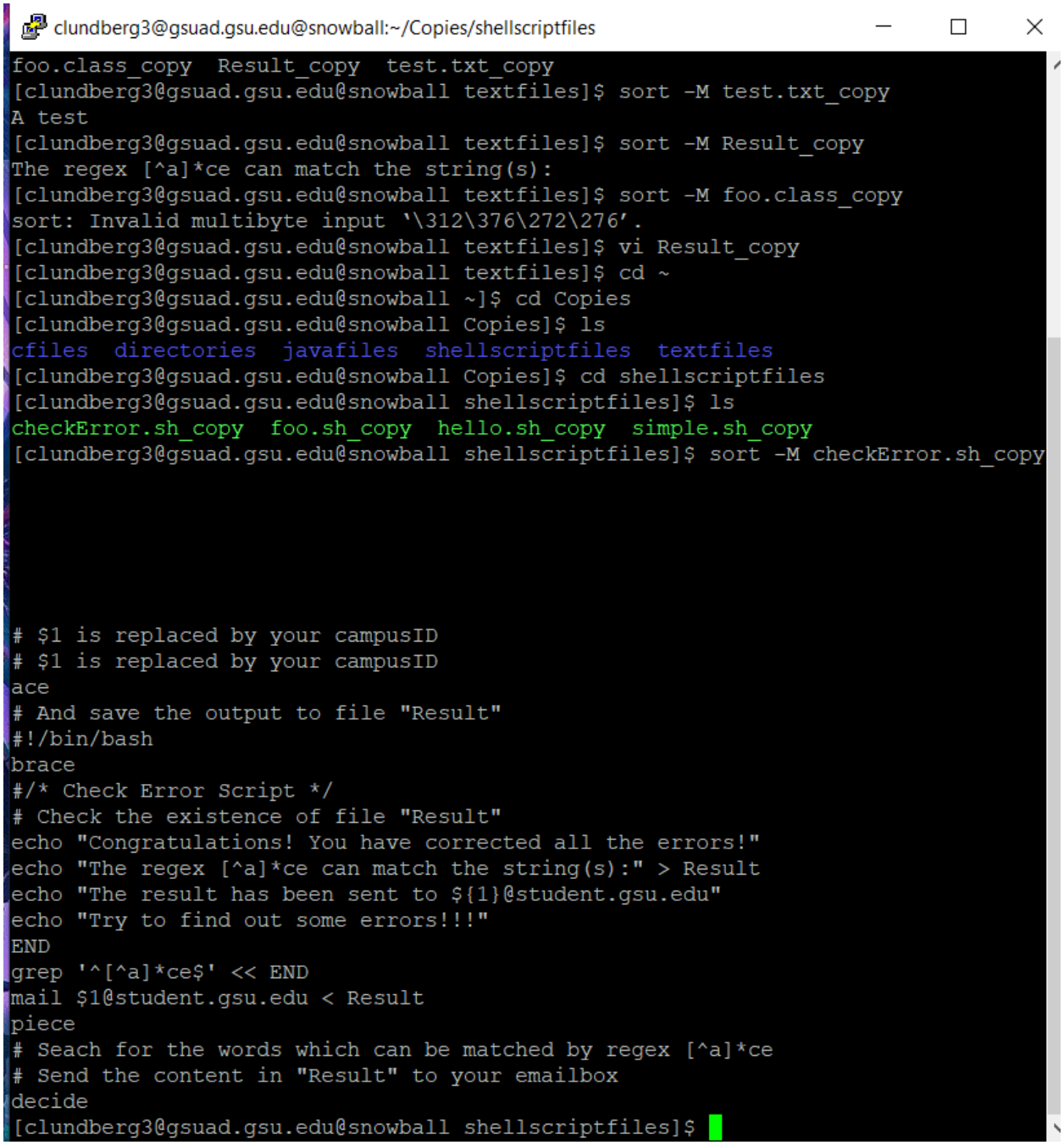

clundberg3@gsuad.gsu.edu@snowball:~/Copies
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mkdir cfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mkdir javafiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mkdir shellscriptfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mkdir directories
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mkdir textfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ ls
a.out_copy directories hello.c_copy javafiles PRACTICE_copy simple.sh_copy
cfiles foo.class_copy hello_copy Lab3_copy public_copy test.txt_copy
checkError.sh_copy foo.java_copy hello.sh_copy Lab4_copy Result_copy textfiles
csc3320_copy foo.sh_copy Homework_Files_copy myName.c_copy shellscriptfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mv {a.out_copy hello.c_copy hello_copy myName.c_copy} cfiles
mv: cannot stat '{a.out_copy': No such file or directory
mv: cannot stat 'myName.c_copy}': No such file or directory
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mv a.out_copy hello.c_copy hello_copy myName.c_copy cfiles
mv: cannot stat 'hello.c_copy': No such file or directory
mv: cannot stat 'hello_copy': No such file or directory
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mv a.out_copy cfiles
mv: cannot stat 'a.out_copy': No such file or directory
[clundberg3@gsuad.gsu.edu@snowball Copies]$ ls
cfiles foo.class_copy Homework_Files_copy PRACTICE_copy simple.sh_copy
checkError.sh_copy foo.java_copy javafiles public_copy test.txt_copy
csc3320_copy foo.sh_copy Lab3_copy Result_copy textfiles
directories hello.sh_copy Lab4_copy shellscriptfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ cd cfiles
[clundberg3@gsuad.gsu.edu@snowball cfiles]$ ls
a.out_copy hello.c_copy hello_copy myName.c_copy
[clundberg3@gsuad.gsu.edu@snowball cfiles]$ mv {checkError.sh_copy foo.sh_copy hello.sh_copy simple.sh_copy}
shellscriptfiles
mv: target 'simple.sh_copy}shellscriptfiles' is not a directory
[clundberg3@gsuad.gsu.edu@snowball cfiles]$ ls
a.out_copy hello.c_copy hello_copy myName.c_copy
[clundberg3@gsuad.gsu.edu@snowball cfiles]$ cd Copies
-bash: cd: Copies: No such file or directory
[clundberg3@gsuad.gsu.edu@snowball cfiles]$ cd ~
[clundberg3@gsuad.gsu.edu@snowball ~]$ cd Copies
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mv {checkError.sh_copy foo.sh_copy hello.sh_copy simple.sh_copy}
shellscriptfiles
mv: target 'simple.sh_copy}shellscriptfiles' is not a directory
[clundberg3@gsuad.gsu.edu@snowball Copies]$ ls
cfiles foo.class_copy Homework_Files_copy PRACTICE_copy simple.sh_copy
checkError.sh_copy foo.java_copy javafiles public_copy test.txt_copy
csc3320_copy foo.sh_copy Lab3_copy Result_copy textfiles
directories hello.sh_copy Lab4_copy shellscriptfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mv checkError.sh_copy foo.sh_copy hello.sh_copy simple.sh_copy s
hellscriptfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ ls
cfiles foo.class_copy javafiles PRACTICE_copy shellscriptfiles
csc3320_copy foo.java_copy Lab3_copy public_copy test.txt_copy


clundberg3@gsuad.gsu.edu@snowball:~/Copies
csc3320_copy foo.sh_copy Lab3_copy Result_copy textfiles
directories hello.sh_copy Lab4_copy shellscriptfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mv checkError.sh_copy foo.sh_copy hello.sh_copy simple.sh_copy s
hellscriptfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ ls
cfiles foo.class_copy javafiles PRACTICE_copy shellscriptfiles
csc3320_copy foo.java_copy Lab3_copy public_copy test.txt_copy
directories Homework_Files_copy Lab4_copy Result_copy textfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mv csc3320_copy Homework_Files_copy Lab3_copy Lab4_copy PRACTICE
_copy public_copy directories
[clundberg3@gsuad.gsu.edu@snowball Copies]$ ls
cfiles foo.class_copy javafiles shellscriptfiles textfiles
directories foo.java_copy Result_copy test.txt_copy
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mv foo.java_copy javafiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ ls
cfiles directories foo.class_copy javafiles Result_copy shellscriptfiles test.txt_copy textfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ foo.class_copy Result_copy test.txt_copy textfiles
-bash: foo.class_copy: command not found
[clundberg3@gsuad.gsu.edu@snowball Copies]$ mv foo.class_copy Result_copy test.txt_copy textfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$ ls
cfiles directories javafiles shellscriptfiles textfiles
[clundberg3@gsuad.gsu.edu@snowball Copies]$

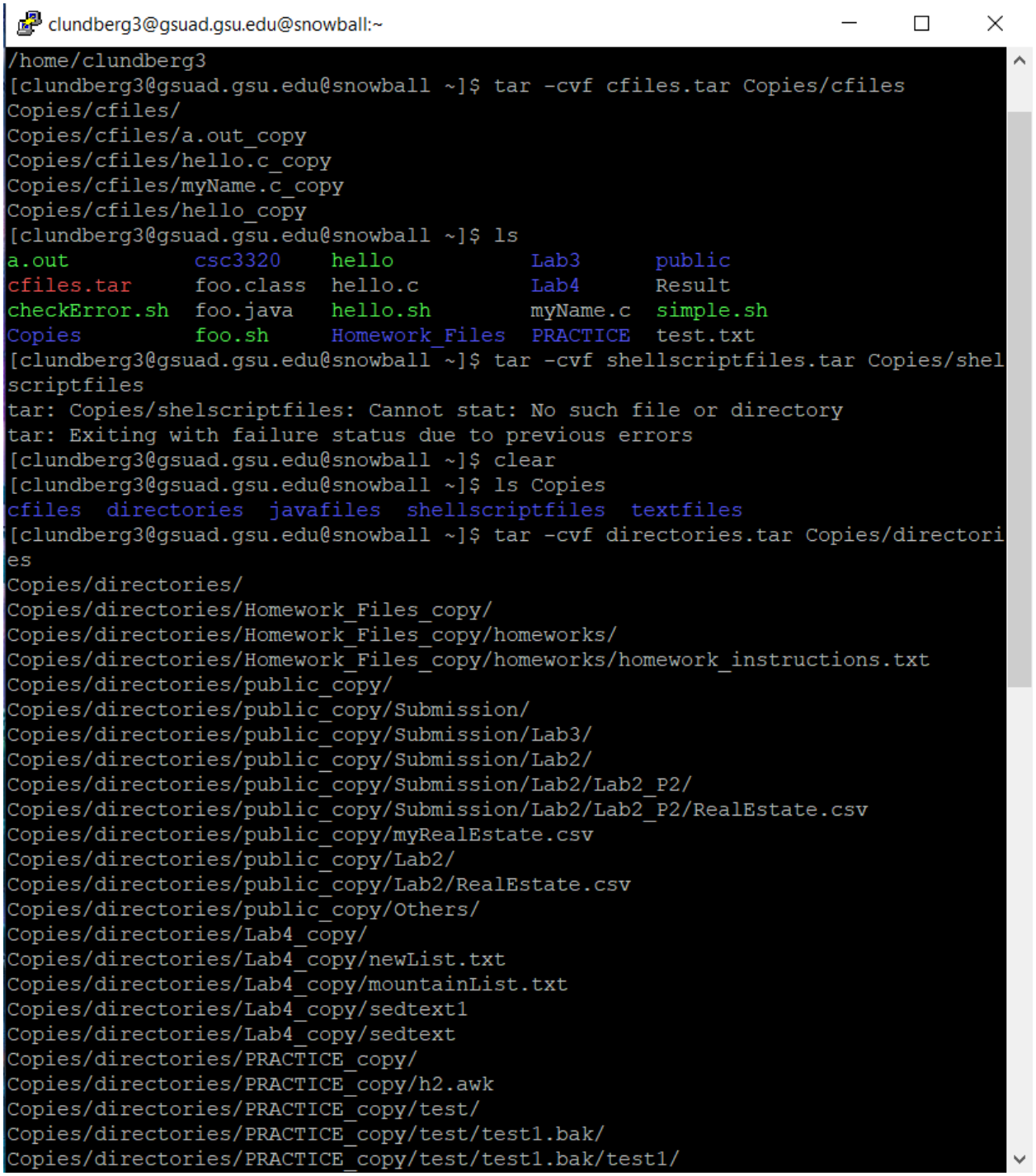
3) To sort any file in a chronological order of months the command sort should be used along with the option “-M” to specify the type of sortation. For example, to sort the text file called “test.txt_copy” the correct input command should look like: sort -M test.txt_copy. However this command does not work on the directory files and shows a jumbled mess with any c files and java files. With any shellscript files, the utility only prints the contents of the file. This utility only works on my copied text files.
This is the output of performing the “sort” utility on the c files:

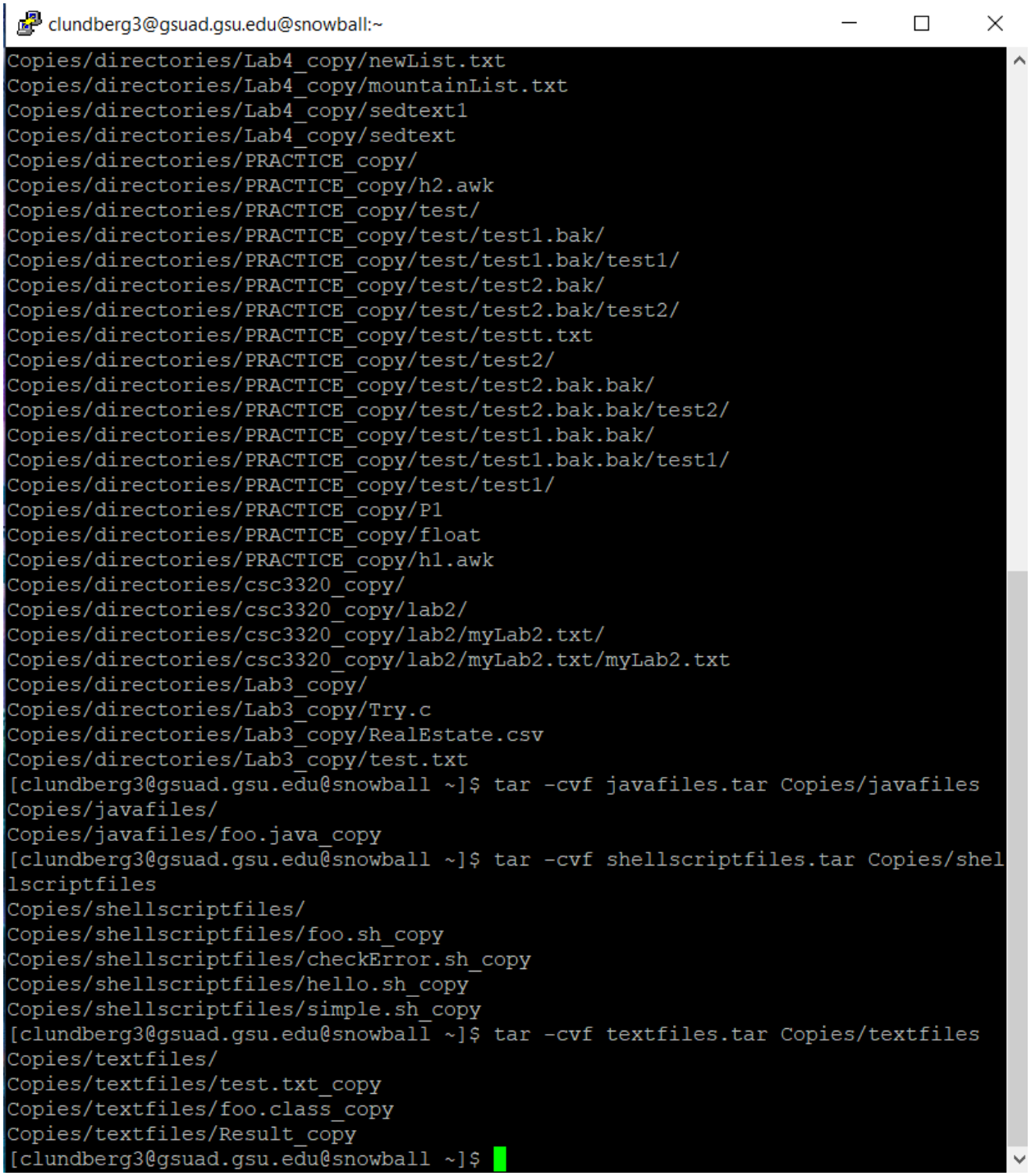


This is the output of performing the “sort” utility on the shellscript files and the text files:



4) Here I used the “tar” utility to create tar format archive files in the home directory.


clundberg3@gsuad.gsu.edu@snowball:~
/home/clundberg3
[clundberg3@gsuad.gsu.edu@snowball ~]$ tar -cvf cfiles.tar Copies/cfiles
Copies/cfiles/
Copies/cfiles/a.out_copy
Copies/cfiles/hello.c_copy
Copies/cfiles/myName.c_copy
Copies/cfiles/hello_copy
[clundberg3@gsuad.gsu.edu@snowball ~]$ ls
a.out csc3320 hello Lab3 public
cfiles.tar foo.class hello.c Lab4 Result
checkError.sh foo.java hello.sh myName.c simple.sh
Copies foo.sh Homework_Files PRACTICE test.txt
[clundberg3@gsuad.gsu.edu@snowball ~]$ tar -cvf shellscriptfiles.tar Copies/shel
scriptfiles
tar: Copies/shelscriptfiles: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
[clundberg3@gsuad.gsu.edu@snowball ~]$ clear
[clundberg3@gsuad.gsu.edu@snowball ~]$ ls Copies
cfiles directories javafiles shellscriptfiles textfiles
[clundberg3@gsuad.gsu.edu@snowball ~]$ tar -cvf directories.tar Copies/directori
es
Copies/directories/
Copies/directories/Homework_Files_copy/
Copies/directories/Homework_Files_copy/homeworks/
Copies/directories/Homework_Files_copy/homeworks/homework_instructions.txt
Copies/directories/public_copy/
Copies/directories/public_copy/Submission/
Copies/directories/public_copy/Submission/Lab3/
Copies/directories/public_copy/Submission/Lab2/
Copies/directories/public_copy/Submission/Lab2/Lab2_P2/
Copies/directories/public_copy/Submission/Lab2/Lab2_P2/RealEstate.csv
Copies/directories/public_copy/myRealEstate.csv
Copies/directories/public_copy/Lab2/
Copies/directories/public_copy/Lab2/RealEstate.csv
Copies/directories/public_copy/Others/
Copies/directories/Lab4_copy/
Copies/directories/Lab4_copy/newList.txt
Copies/directories/Lab4_copy/mountainList.txt
Copies/directories/Lab4_copy/sedtext1
Copies/directories/Lab4_copy/sedtext
Copies/directories/PRACTICE_copy/
Copies/directories/PRACTICE_copy/h2.awk
Copies/directories/PRACTICE_copy/test/
Copies/directories/PRACTICE_copy/test/test1.bak/
Copies/directories/PRACTICE_copy/test/test1.bak/test1/


clundberg3@gsuad.gsu.edu@snowball:~
Copies/directories/Lab4_copy/newList.txt
Copies/directories/Lab4_copy/mountainList.txt
Copies/directories/Lab4_copy/sedtext1
Copies/directories/Lab4_copy/sedtext
Copies/directories/PRACTICE_copy/
Copies/directories/PRACTICE_copy/h2.awk
Copies/directories/PRACTICE_copy/test/
Copies/directories/PRACTICE_copy/test/test1.bak/
Copies/directories/PRACTICE_copy/test/test1.bak/test1/
Copies/directories/PRACTICE_copy/test/test2.bak/
Copies/directories/PRACTICE_copy/test/test2.bak/test2/
Copies/directories/PRACTICE_copy/test/testt.txt
Copies/directories/PRACTICE_copy/test/test2/
Copies/directories/PRACTICE_copy/test/test2.bak.bak/
Copies/directories/PRACTICE_copy/test/test2.bak.bak/test2/
Copies/directories/PRACTICE_copy/test/test1.bak.bak/
Copies/directories/PRACTICE_copy/test/test1.bak.bak/test1/
Copies/directories/PRACTICE_copy/test/test1/
Copies/directories/PRACTICE_copy/P1
Copies/directories/PRACTICE_copy/float
Copies/directories/PRACTICE_copy/h1.awk
Copies/directories/csc3320_copy/
Copies/directories/csc3320_copy/lab2/
Copies/directories/csc3320_copy/lab2/myLab2.txt/
Copies/directories/csc3320_copy/lab2/myLab2.txt/myLab2.txt
Copies/directories/Lab3_copy/
Copies/directories/Lab3_copy/Try.c
Copies/directories/Lab3_copy/RealEstate.csv
Copies/directories/Lab3_copy/test.txt
[clundberg3@gsuad.gsu.edu@snowball ~]$ tar -cvf javafiles.tar Copies/javafiles
Copies/javafiles/
Copies/javafiles/foo.java_copy
[clundberg3@gsuad.gsu.edu@snowball ~]$ tar -cvf shellscriptfiles.tar Copies/shel
lscriptfiles
Copies/shellscriptfiles/
Copies/shellscriptfiles/foo.sh_copy
Copies/shellscriptfiles/checkError.sh_copy
Copies/shellscriptfiles/hello.sh_copy
Copies/shellscriptfiles/simple.sh_copy
[clundberg3@gsuad.gsu.edu@snowball ~]$ tar -cvf textfiles.tar Copies/textfiles
Copies/textfiles/
Copies/textfiles/test.txt_copy
Copies/textfiles/foo.class_copy
Copies/textfiles/Result_copy
[clundberg3@gsuad.gsu.edu@snowball ~]$

5) An archive file of all the directories is available in the home directory.