CSc 3320: Systems Programming

Fall 2021

Homework

# 1: 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 Wepster

Campus ID: kgallardowepster1

Panther #: 002-49-7849

PART 1

Answer the following questions briefly. Provide clear and succinct reasoning.

Points per question = 5

1. Tell the differences between Unix and Linux. Then please list some operating systems (at least three) which belong to Unix but not Linux.

Linux is open source while Unix is not. Meaning that you can view the source code for Linux but not Unix. Some operating systems that belong to Unix are HP-UX, AIS,and BSD.

2. What is the pipe mechanism in UNIX? And show one command using pipe and explain how the pipe works in it?

The pipe mechanism in Unix is used for connecting two processes together. Meaning that by using pipe the output of the first process will become the input for the second process and so on.

```
grep ‘Rabun County’ mountainList.txt | wc -l
```

In the previous command grep will look for the word : Rabun County and give as an output the lines that contain the word. Pipe will take that output and feed it to wc (word count) -l and wc will count the number of lines and give as output the number of lines that contained the word from the output of the grep search.

3. In a Linux system, you can issue the command **ls /** to check the sub directories under root. Please describe the meanings of directory /bin, /dev, /boot, /usr, /etc, /mnt, /sbin, /var separately. For example, you can say that /bin contains binary executable files.

/bin- contains the executable programs.

/var- contains variable data files.

/boot- contains files used in booting the operating system.

/usr- contains user binaries

/etc- contains system configuration files

/mnt- contains mount points for storage devices

/sbin- contains executable programs that should only be available to the root user.

/var- contains variable data files.

4. What is the meaning of Multitask and Multi-user in a Unix system?

Multitask means that more than one program is able to run at one time.

Multi-user means that more than one person can use the system at the same time.

5. What does -rwxr-xr-x mean in terms of permissions for a file? What is the exact unix command (with the octal representation) for changing the permissions to this setting?

The command permission -rwxr-xr-x means that it is a file and the user has read, write, and execute permissions. Groups will only have read and execute permissions. And others will have read and execute permission as well.

The command will be : chmod 755 nameOfFile

6. In class, you have learned the meaning of read, write and execute permission for regular files. However, these permissions are also applied to directories. So please describe the meaning of read, write, and execute permission for directory.

Read: contents can be seen by the user

Write: user can create files in the directory

Execute: user can access the directory

Part II-a

Regular Expression
Find outcomes for each given basic/extended regular expression (maybe multiple correct answers)

Points per question: 2.5

Example:
‘ab+a’（extended regex）

***Answer**: aba , abba ; Pattern : The matched string should begin and end with ‘a’ and ‘b’ occurs at least once between leading and ending ‘a’)*

Note: 7) to 10) are basic regexes; Note: 11) to 18) are extended regexes.

7) ‘a[ab]*a’

Answer: aa,aaba,aba; The matched string should begin and end with 'a' and in between there can be zero or more occurrences of any character from bracket [ab].

8) ‘a(bc)?’

Answer: a, abc; The matched string should begin with 'a' and can have 'bc' zero or one time after 'a'.

9) ‘.[ind]*’

Answer: mi, t, tn, td, ti, miidn; The matched string should begin with any character except newline and can have any character from bracket [ind] 0 time or more times.

10) ‘[a-z]+[a-z]’

Answer: abcd, avc, ad; The matched string should end with any alphabet ranging from 'a' to 'z' and also have an alphabet between range 'a' to 'z' before the end character, which means the matched string should start and end with a lowercase alphabet and can have any lowercase alphabet in the middle.

11) ‘[a-z] (\+[a-z])+’

Answer: a+b, a+b+b; The matched string should start with any

alphabet between the range 'a' to 'z' and have at least one occurrence

of + and any character between the range of 'a' to 'z'.

12) ‘a.[bc]+’

Answer: abb, atbc, aucb; The matched string should start with 'a' followed by any character except a newline character and after that character there should be at least one character from bracket [bc] but more than one can also be there, means the string should end with a 'b' or 'c'.

13) ‘a.[0-9]’

Answer: at0, ar9; The matched string should start with 'a' and end with a digit ranging from 0-9 and in middle there can be any character but only once.

14) ‘[a-z]+[\.\?!]’

Answer: a., abcd?, az! ; The matched string should start with any character between 'a' to 'z' and should end with any one character among '.', '?' and '!'.

15) ‘[a-z]+[\.\?!]\s*[A-Z]’

Answer: a. A, abc! Z, abcds?B ; The matched string should start with a lower case alphabet and can have any number of lowercase alphabets followed by a character '.' or '!' or '?', after that there can be zero or more white spaces and the string should end with an upper case alphabet between 'A' to 'Z'.

16) ‘(very)+(cool)?(good|bad) weather’

Answer: verycoolgood weather, veryverybad weather ; The matched string should contain one or more occurrences of 'very' and after that should have zero or one occurrence of 'cool' followed by 'good' or 'bad' word and in the end the string should have 'weather'.

17) ‘-?[0-9]+’

Answer: 9, -2, -345, 454 ; The matched string should start with zero or one occurrence of '-' followed by one or more occurrences of any digit between the range '0' to '9'.

18) ‘-?[0-9]*\.?[0-9]*’

Answer: 9, -2, -345, 454 ; The matched string should contain zero or one occurrence of '-' followed by zero or more occurrences of any digit between the range '0' to '9' followed by zero or once occurrence '.' and in the end there should have zero or more occurrences of any digit between the range '0' to '9'. This means the given regex can match with any string and returns true even with the empty string since each character can have zero occurrence.

Part II-b

Regular Expression
Write down the extended regular expression for following questions.
E.g. Social security number in the format of 999-99-9999. Answer:
[0-9]{3}-[0-9]{2}-[0-9]{4}

Points per question: 5

19) Valid URL beginning with “http://” and ending with ".edu" (e.g. http://cs.gsu.edu, http://gsu.edu)

(http:\/\/.)[-a-zA-Z0-9@:%._\+~#=]{2,256}\.(edu)

20) Non-negative integers. (e.g. 0, +1, 3320)

(([1-9][0-9])*|0)?

21) A valid absolute pathname in Unix (e.g. /home/ylong4, /test/try.c)

([.\/]+[a-z]*)*

22) Identifiers which can be between 1 and 10 characters long, must start with a letter or an underscore. The following characters can be letters or underscores or digits. (e.g. number, _name1, isOK).

[_a-Z]{10}

23) Phone number in any of the following format: 9999999999,999-999-9999, (999)-999-9999. (Note: all of these formats should be matched by a single regular expression)

[0-9]{10}
[0-9]{3}-[0-9]{3}-[0-9]{4}
([0-9]{3})-[0-9]{2}-[0-9]{4}

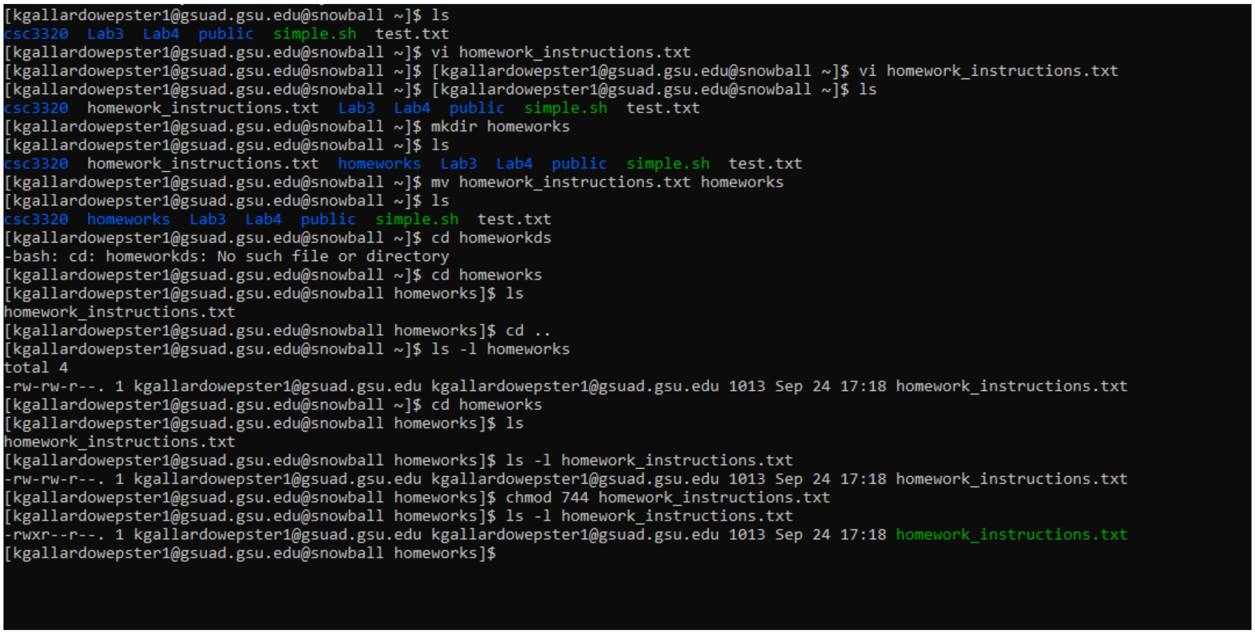
Part III

Programming

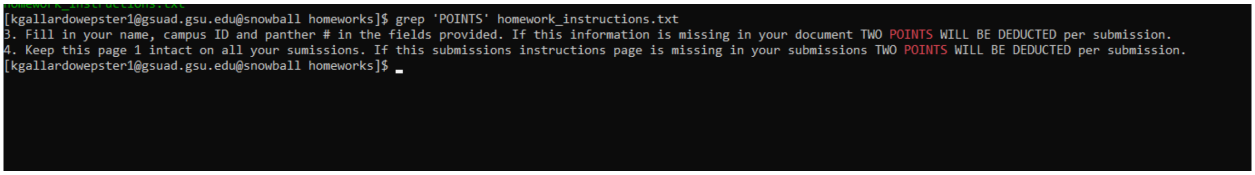
Points per question: 15

24. Create a file named homework_instructions.txt using VI editor and type in it all the submission instructions from page1 of this document. Save the file in a directory named *homeworks* that you would have created. Set the permissions for this file such that only you can edit the file while anybody can only read. Find and list (on the command prompt) all the statements that contain the word POINTS. Submit your answer as a description of what you did in a sequential manner (e.g. Step1 … Step 2… and so on..). Add a screenshot to your answer as a proof of evidence.1

Step 1 : vi homework_instructions.txt
Step 2: (Shift + i) to insert words into text file
Step 3: esc and then go to command mode and :wq to exit and save
Step 4: mkdir homeworks
Step 5: mv homeworks_instructions.txt homeworks
Step 6: cd homeworks
Step 7: chmod 744 homework_instructions.txt
Step 8: grep 'POINTS' homework_instructions



```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ls
csc3320  Lab3  Lab4  public  simple.sh  test.txt
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ vi homework_instructions.txt
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ [kgallardowepster1@gsuad.gsu.edu@snowball ~]$ vi homework_instructions.txt
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ [kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ls
csc3320  homework_instructions.txt  Lab3  Lab4  public  simple.sh  test.txt
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ mkdir homeworks
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ls
csc3320  homework_instructions.txt  homeworks  Lab3  Lab4  public  simple.sh  test.txt
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ mv homework_instructions.txt homeworks
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ls
csc3320  homeworks  Lab3  Lab4  public  simple.sh  test.txt
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cd homeworkds
-bash: cd: homeworkds: No such file or directory
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cd homeworks
[kgallardowepster1@gsuad.gsu.edu@snowball homeworks]$ ls
homework_instructions.txt
[kgallardowepster1@gsuad.gsu.edu@snowball homeworks]$ cd ..
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ls -l homeworks
total 4
-rw-rw-r--. 1 kgallardowepster1@gsuad.gsu.edu kgallardowepster1@gsuad.gsu.edu 1013 Sep 24 17:18 homework_instructions.txt
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cd homeworks
[kgallardowepster1@gsuad.gsu.edu@snowball homeworks]$ ls
homework_instructions.txt
[kgallardowepster1@gsuad.gsu.edu@snowball homeworks]$ ls -l homework_instructions.txt
-rw-rw-r--. 1 kgallardowepster1@gsuad.gsu.edu kgallardowepster1@gsuad.gsu.edu 1013 Sep 24 17:18 homework_instructions.txt
[kgallardowepster1@gsuad.gsu.edu@snowball homeworks]$ chmod 744 homework_instructions.txt
[kgallardowepster1@gsuad.gsu.edu@snowball homeworks]$ ls -l homework_instructions.txt
-rwxr--r--. 1 kgallardowepster1@gsuad.gsu.edu kgallardowepster1@gsuad.gsu.edu 1013 Sep 24 17:18 homework_instructions.txt
[kgallardowepster1@gsuad.gsu.edu@snowball homeworks]$
```



```
homework_instructions.txt
[kgallardowepster1@gsuad.gsu.edu@snowball homeworks]$ grep 'POINTS' homework_instructions.txt
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 sumissions. If this submissions instructions page is missing in your submissions TWO POINTS WILL BE DEDUCTED per submission.
[kgallardowepster1@gsuad.gsu.edu@snowball homeworks]$
```