

CSc 3320: Systems Programming

Spring 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: Venkata Mani Mohana Rishitha Srikakulapu

Campus ID: vsrikakulapu1

Panther #: 002523638

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.

Unix is the complete pack of Operating system whereas Linux is just the kernel. Unix is developed by the AT & T Bell labs and so is not a free source but Linux is developed by the community developers of Linux, and so is an open source to use. Unix is used widely in server systems, workstations and PCs while Linux is used in desktop servers, smartphones, and mainframes. The default shell for Unix is Bourne shell whereas bash (Bourne Again Shell) is the default shell for Linux. Unix used to be a command based OS in the beginning and now uses Gnome as GUI. However, Linux supports GUIs like KDE, LXDE, Unity and Mate.

The operating systems that belong only to Unix are :

AIX, HP/UX, SCO UNIX, Solaris, SunOS, ULTRIX.

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

Pipe mechanism in Unix helps in providing the output of one command as input to another command. It makes the task easier by fragmenting the work and then combining it using pipe mechanism.

For example: `cat hw.txt | wc -l`

In the above example, the cat command used to view the content of hw.txt file, and gives the hw.txt as input to word count command. This word count with -l field, lists the number of newlines in the file hw.txt.

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.

The meanings of the directories are :

/bin contains binary executable files.

/dev stores special device files for all devices.

/boot keeps all the boot related information files needed for booting the operating system.

/usr holds all the header files, user binaries, related documentation, libraries and also their supporting libraries.

/etc stores all the static configuration files of the system which helps in the operation of programs.

/mnt directory provides temporary mounting options for storage devices like floppy disks, USB drives and CDRoms.

/sbin holds executable programs that are administrative i.e. those are only available to the root user and also stores all the other directories and the files

as well.

/var handles variable data files including administrative and logging data, spool files and directories, and temporary files.

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

Multitask means running various programs at the same time i.e. different jobs are being done at a single instance of time.

Multi-user means many number of users can work at the same time through different computing systems with shared processing 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?

In terms of file permissions, 'r' stands for read, 'w' for write and 'x' for execute. The first part 'rwx' states that the **user** has all the 3 permissions i.e. read, write and execute the file while the second part is 'r-x' which means the **group** has read and execute permissions but not the write permission. The third part gives access for **others** to only read and execute the file but not the write permission.

The unix command (with the octal representation) for changing the permissions to this setting is :

```
$ chmod 755 <file-spec>
```

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.

The permissions of directory can be obtained by the command

```
$ ls -ld <dir>
```

Here -d option will help us check the detailed information for the directory instead of its contents..

In a directory level,

The read permission for a directory allows to list all its contents which include files and directories present in that directory.

The write permission allows to create new files and directories and also to remove files or directories present in that directory.

The execute permission of the directory allows to run a search on that directory by the user.

In the file permissions setting “ -rwxr--r-- , ” the beginning ‘-’ stands for a regular file whereas for a directory, that specific permission setting changes as :

```
drwxr--r--
```

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'**

aaa , aba, aa, ababa;

Pattern: The matched string should begin and end with an 'a' and 'a' , 'b' can occur zero or more times between the leading and ending 'a'.

8) **'a(bc)?'**

a, abc;

Pattern: The matched string begins with 'a' and the group 'bc' together can occur at most once after the leading 'a'.

9) **'[ind]*'**

0i, 9n, ddi977, 879ind;

Pattern: The matched string begins with any character and contains zero or more occurrences of 'i', 'n' and 'd'. Basically, this pattern matches every string.

10) **'[a-z]+[a-z]'**

aa, ab, az, bbnh;

Pattern: The matched string contains at least one lowercase character from

the range 'a' to 'z' . Basically, this pattern matches every lower case alphabet in any string.

11) '[a-z] (\+[a-z])+'

a+a, a+c, c+h+h+h+h+h, f+y+s;

Pattern: The matched string begins with a lowercase character between the range 'a' to 'z' followed by at least one group containing a '+' character and a lowercase character from the range 'a' to 'z' .

12) 'a.[bc]+'

a*b, a/c, ayb;

Pattern: The matched string should begin with 'a' and then followed by any single character and then followed by at least one occurrence of any character out of 'b' and 'c'.

13) 'a.[0-9]'

a56, ab0, a01, a&2;

Pattern: The matched string begins with 'a' and ends with a digit from the range '0' to '9' and any one single character occurs in between the leading 'a' and the digit.

14) '[a-z]+[\.\\?!]'

a!, a?, deujbc., vbfh?;

Pattern: The matched string begins with at least one lowercase character from the range 'a' to 'z' and ends with any one literal out

of '.' or '?' or '!'.

15) '[a-z]+[\\.\\?!]\\s*[A-Z]'

a.A, a! G, egydhc! J, dc bd. K;

Pattern: The matched string begins with at least one lowercase character from the range 'a' to 'z' followed by a literal which can either be a '.' or a '?' or a '!' and then followed by zero or more white spaces and then followed by any one uppercase character from the range 'A' to 'Z'.

16) '(very)+(cool)?(good|bad) weather'

very cool good weather, very bad weather, very very good weather;

Pattern: The matched string begins with at least one 'very' followed by at most one occurrence of 'cool' and then only one out of 'good' or 'bad' and then ends with 'weather'.

17) '-?[0-9]+'

-535, -64, 654, 886;

Pattern: The matched string contains the character '-' at most once, followed by at least one digit from the range '0' to '9'.

18) '-?[0-9]*\\.?[0-9]*'

-32.67, ----789, 867, 987.56, 56.45-6.7;

Pattern: The matched string contains the character '-' at most once then followed by zero or more digits from the range '0' to '9' and then contains the character '.' at most once and then followed by zero or more digits out of the range '0' to '9'.

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]+\.`**edu**

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

`(\+|)?[0-9]+`

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

`'/[a-zA-Z0-9_\.]+'`

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-zA-Z_][a-zA-Z0-9_]{0,9}'`

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)

`'(\([0-9]{3}\))[-][0-9]{3}[-][0-9]{4}|[0-9]{10}|[0-9]{3}[-][0-9]{3}[-][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.

Step 1: Logged into snowball terminal using campusID

```
$ ssh campusid@snowball.cs.gsu.edu
```

Step 2: Created a directory named *homeworks* in the home directory.

```
$ mkdir homeworks
```

Step 3: Checked whether the directory is created or not

```
$ ls
```

Step 4 : Created a new file named `homework_instructions.txt` in the home directory.

```
$ vi homework_instructions.txt
```

Step 5: Started with an insert key (pressing `i`) and then typed all the submission instructions into the file, then pressed Esc key, then saved the file using `:w`

Step 6 : Copied the `homework_instructions.txt` file into the *homeworks* directory.

```
$ cp homework_instructions.txt homeworks
```

Step 7: Changed directory to *homeworks*

\$ cd homeworks

Step 8 : Set the file permissions to -rw-r--r-- by using octal number.

\$ chmod 644 homework_instructions.txt

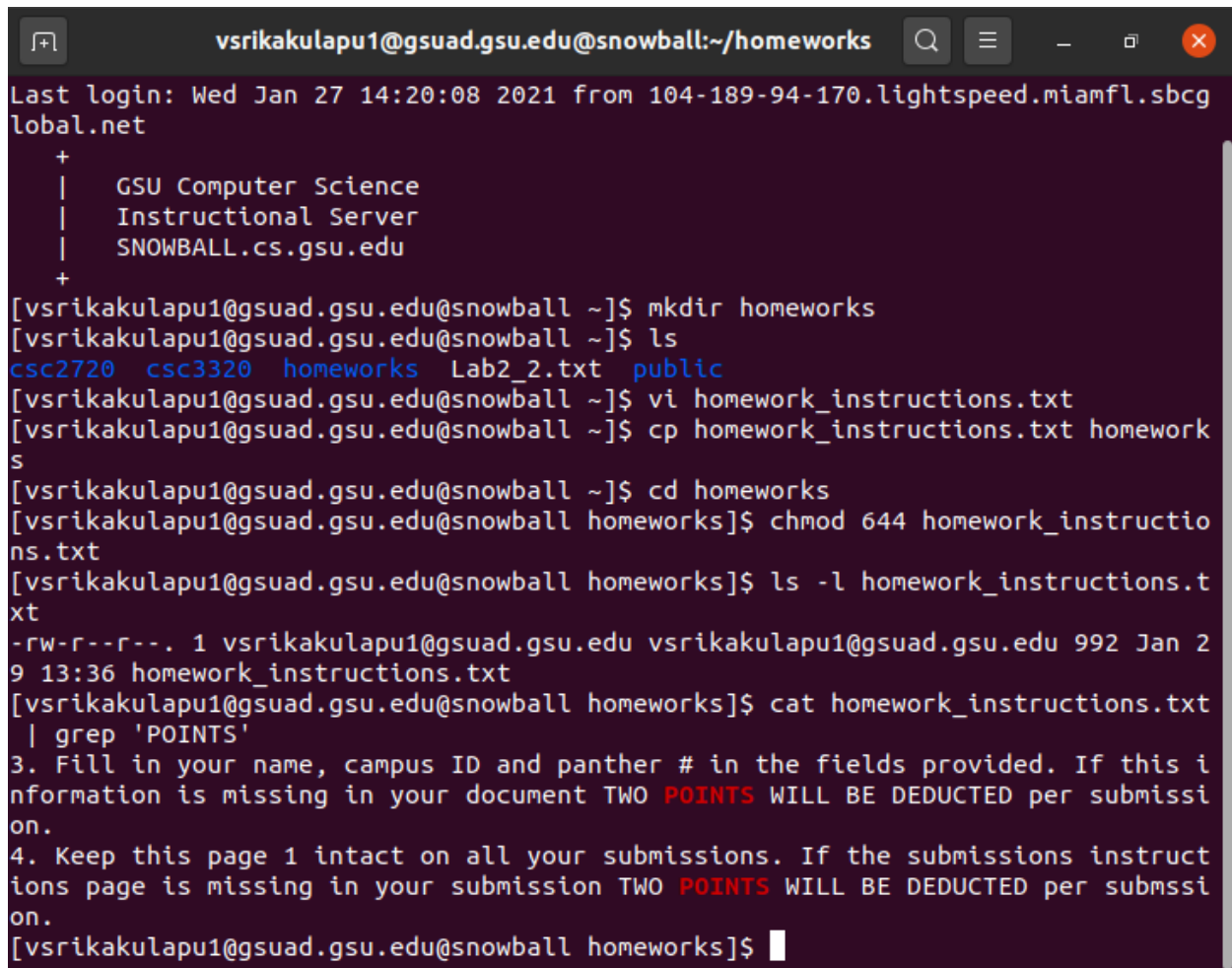
Step 9 : Cross - checked whether the permissions are changed or not.

\$ ls -l homework_instructions.txt

Step 10 : Listed all the lines containing the word 'POINTS' using grep command

\$ cat homework_instructions.txt | grep 'POINTS'

Step 11 : Took a screenshot of the displayed lines in the output.

A terminal window titled 'vsrikakulapu1@gsuad.gsu.edu@snowball:~/homeworks' with standard window controls. The terminal shows a series of commands and their outputs. It starts with a login banner for 'vsrikakulapu1@gsuad.gsu.edu@snowball' and a directory listing. Then, the user runs 'mkdir homeworks', 'ls', 'vi homework_instructions.txt', and 'cp homework_instructions.txt homeworks'. Next, they run 'cd homeworks', 'chmod 644 homework_instructions.txt', and 'ls -l homework_instructions.txt', which shows the file permissions as '-rw-r--r--'. Finally, they run 'cat homework_instructions.txt | grep 'POINTS'', which displays two lines of text: '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.' and '4. Keep this page 1 intact on all your submissions. If the submissions instructions page is missing in your submission TWO POINTS WILL BE DEDUCTED per submission.' The terminal ends with a prompt for the user to enter more commands.

```
vsrikakulapu1@gsuad.gsu.edu@snowball:~/homeworks
Last login: Wed Jan 27 14:20:08 2021 from 104-189-94-170.lightspeed.miamfl.sbcg
lobal.net
+
|   GSU Computer Science
|   Instructional Server
|   SNOWBALL.cs.gsu.edu
+
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ mkdir homeworks
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ls
csc2720 csc3320 homeworks Lab2_2.txt public
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi homework_instructions.txt
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ cp homework_instructions.txt homeworks
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ cd homeworks
[vsrikakulapu1@gsuad.gsu.edu@snowball homeworks]$ chmod 644 homework_instructions.txt
[vsrikakulapu1@gsuad.gsu.edu@snowball homeworks]$ ls -l homework_instructions.txt
-rw-r--r--. 1 vsrikakulapu1@gsuad.gsu.edu vsrikakulapu1@gsuad.gsu.edu 992 Jan 29 13:36 homework_instructions.txt
[vsrikakulapu1@gsuad.gsu.edu@snowball homeworks]$ cat homework_instructions.txt | grep 'POINTS'
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 the submissions instructions page is missing in your submission TWO POINTS WILL BE DEDUCTED per submission.
[vsrikakulapu1@gsuad.gsu.edu@snowball homeworks]$
```