

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 *submission's 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: Vivian Do

Campus ID: vdo10

Panther #: 002486640

Due Sep 24, 2021

PART I

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 a powerful operating system used by large companies with legacy programs or by those willing to pay for more guaranteed uptime; it can run on multiple processors. Linux is a copy of Unix, but is licensed through the GNU license.
 - Unix is owned by OS; Linux is free to use
 - Unix's default shell is Bourne Shell; Linux's default shell is BASH (Bourne Again SHell)
 - Unix was a command-based OS, but most distributions have Gnome; Linux uses KDE and Gnome

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 is used to combine two or more commands; the output of one command acts as the input of another command, and this command's output may act as the input of the next command. This continues up until the last command's input/output.
- Example: `ls | wc -l`
 - `ls` is used to show directories, files, and links in the current directory
 - `wc` is used to count these file system objects

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` - binary files
 - Command holds the binary executable files. Commands, such as `ps`, `ls`, and `grep`, are listed under here.
- `/dev` - device files
 - Command holds the device files. Terminal devices like USB or any device connected to the system are potentially included in this command. Commands

such as /ttyl and /dev/usbmon0 are under here.

- /boot - boot loader files
 - Command has boot loader files that will start up a computer
- /usr - user programs
 - Command holds libraries, binaries, and source code for second-level programming. There are various commands under here.
 - /usr/bin - user program's binary files
 - /usr/sbin - system administrator's binary executable files
 - /usr/lib - libraries for /usr/bin and /usr/sbin
 - /usr/local - has programs installed from source
- /etc - configuration files
 - Command holds scripts for startup and shutdown of programs; these are important here!
- /mnt - mount directory
 - Command can be used for mount directory. This allows system admins to mount file systems here.
- /sbin - system binaries
 - Command holds the system administrator's binary executables; it's similar to the /bin command. Commands such as reboot and fdisk are under here.
- /var - variable files
 - Command holds files whose content can be grown or just variables. Command such as /var/mail (for emails) and /var/spool (for printing files) are under here.

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

- *multitask*: being able to do multiple jobs at once in Unix
 - Multitasking allows everything to look like it's all running at the same time by dividing the processor's runtime
- *multi-user*: allowing multiple users to access a singular system with only one operating system in Unix
 - multi-user is generally used on large mainframe computers

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?

- -rwxr-xr-x allows the user to read, write, and execute programs.; the group and others can only read and execute.
- command: \$chmod 755
- The directory has permission files of 755:
- read + write + execute (4 + 2 + 1), read + execute (4 + 1), read + execute (4 + 1)

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 permission: allows the user to see the contents of a directory
- write permission: allows the user to create files in the directory
- execute permission: allows the user to enter 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'

- This expression creates a string starting and ending with 'a' in the string. The brackets [] matches any one of the enclosed characters and the asterisk '*' matches 0 or more occurrences for the character before it in Unix.
- Answer: ababa, aabbbaa, aaba

8) 'a(bc)?'

- This expression creates a string starting with 'a' and ending with 0 or 1 occurrences of the substring 'bc'. The parenthesis () lets grouping several characters as one and the question mark ? matches the occurrences of 0 and 1 for the characters before it in Unix.
- Answer: abc, a

9) '[ind]*'

- This expression creates a string starting with any character and ending with 0 or more occurrences of any enclosed characters. The brackets [] matches any of the enclosed characters, the asterisk '*' matches the 0 or more occurrences for the character before it, and the period '.' matches any single character in Unix.
- Answer: find, and

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

- This expression creates a string starting with 1 or more occurrences of any of the enclosed characters and ending with any of the enclosed characters. The brackets [] matches any one of the enclosed characters and the plus sign '+' matches the 1 or more occurrences of the character before it in Unix.
- Answer: azaz

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

- This expression creates a string starting with any of the enclosed characters and ending with 1 or more occurrences of the sub pattern. The sub patterns start with '+' and end with any of the enclosed characters. The brackets [] matches any of the enclosed characters, the plus sign '+' matches the 1 or more occurrences of the character before it, and the backslash '\' turns off the special meaning of the character in Unix.
- Answer: a+b+c

12) 'a.[bc]+'

- This regular expression creates a string starting with any of the enclosed characters and ending with 1 or more occurrences of enclosed characters. The brackets [] matches any of the enclosed characters, the plus sign '+' matches the 1 or more occurrences of the characters before it, and the period '.' matches any single character in Unix.
- Answer: azbc, acdc

13) 'a.[0-9]'

- This regular expression creates a string starting with 'a', followed by any single character, and ending with any of the enclosed characters within the range. The brackets [] matches any of the enclosed characters, the period '.' matches any single characters, and the hyphen '-' is used to define the range in Unix.
- Answer: a02

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

- This regular expression creates a string starting with 1 or more occurrences of any enclosed characters within the range and ending with any of the enclosed characters. The brackets [] matches any of the enclosed characters, the period '.' matches any single character, and the backslash '\' turns off the special meaning of the character in Unix.
- Answer: yes!, no?

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

- This regular expression creates a string starting with 1 or more occurrences of any of the enclosed characters within the range, followed by any number of spaces, and ending with any of the enclosed characters within the range. The brackets [] matches any of the enclosed characters, the period '.' matches any single character, the backslash '\' turns off the special meaning of the next character, and the asterisk '*' matches the 0 or more occurrences of the character before it in Unix.
- Answer: freeze, K

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

- This regular expression creates a string starting with 'very', followed by 0 or 1 occurrences of 'cool', followed by 'good' or 'bad', and ending with 'weather'.
- Answer: very good weather, very cool bad weather

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

- This regular expression creates a string starting with 0 or more occurrences of the hyphen '-' and ending with 1 or more occurrences of the enclosed characters within the range.
- Answer: 123, -321

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

- This expression creates a string starting with 0 or 1 occurrences of the hyphen '-', followed by 0 or more occurrences of the enclosed characters within the range, followed by 0 or 1 occurrences of the period '.', and ending with 0 or more occurrences of the enclosed characters within the range.
- Answer: 1232, -321, 0.4

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-z]*\\.?[a-z]*(.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-z]+\\[a-z]+[0-9]*\\.?[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]*[a-z]*[0-9]*[A-Z]*){1,10}`

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]*{1,10}\\)([0-9]*{1,3}-[0-9]*{1,4})|(\\[0-9]*{1,3}\\)([0-9]*{1,3}-[0-9]*{1,4})`

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.

Whole Screenshot of Part 3:

```
vdo10@gsuad.gsu.edu@snowball:~/homeworks
login as: vdo10
vdo10@snowball.cs.gsu.edu's password:
Last login: Tue Sep 21 12:57:40 2021 from c-24-30-16-254.hsd1.ga.comcast.net
+
|   GSU Computer Science
|   Instructional Server
|   SNOWBALL.cs.gsu.edu
+
[vdo10@gsuad.gsu.edu@snowball ~]$ cd ~
[vdo10@gsuad.gsu.edu@snowball ~]$ ls
csc3320 Lab3 public
Step 1
[vdo10@gsuad.gsu.edu@snowball ~]$ mkdir homeworks
[vdo10@gsuad.gsu.edu@snowball ~]$ ls
csc3320 homeworks Lab3 public
Step 2
[vdo10@gsuad.gsu.edu@snowball ~]$ cd homeworks
[vdo10@gsuad.gsu.edu@snowball homeworks]$ vim homework_instructions.txt
Step 4
[vdo10@gsuad.gsu.edu@snowball homeworks]$ vim homework_instructions.txt
Step 5
[vdo10@gsuad.gsu.edu@snowball homeworks]$ ls
homework_instructions.txt
[vdo10@gsuad.gsu.edu@snowball homeworks]$ ls -l
total 4
-rw-rw-r--. 1 vdo10@gsuad.gsu.edu vdo10@gsuad.gsu.edu 1124 Sep 21 13:42 homework_instructions.txt
Step 6
[vdo10@gsuad.gsu.edu@snowball homeworks]$ chmod 644 homework_instructions.txt
[vdo10@gsuad.gsu.edu@snowball homeworks]$ ls -l
total 4
-rw-r--r--. 1 vdo10@gsuad.gsu.edu vdo10@gsuad.gsu.edu 1124 Sep 21 13:42 homework_instructions.txt
[vdo10@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 TW
0 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.
[vdo10@gsuad.gsu.edu@snowball homeworks]$ grep -i POINTS homework_instructions.txt
Homework # 1: Total points 100
3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing in your document TW
0 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.
[vdo10@gsuad.gsu.edu@snowball homeworks]$
```

Step 3 Screenshot:

```
vdo10@gsuad.gsu.edu@snowball:~/homeworks
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 test code or run specific commands or scripts 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:
Campus ID:
Panther #:

Step 3

"homework_instructions.txt" 22L, 1124C written 22,0-1 All
```

Step 1: I made a directory named *homeworks* (mkdir homeworks) and checked to see if it was actually there (ls).

```
[vdo10@gsuad.gsu.edu@snowball ~]$ mkdir homeworks
[vdo10@gsuad.gsu.edu@snowball ~]$ ls
csc3320  homeworks  Lab3  public
```

Step 2: I went to the directory (cd homeworks) and created *homework_instructions.txt* file using VI editor (vim homework_instructions.txt).

```
[vdo10@gsuad.gsu.edu@snowball ~]$ cd homeworks
[vdo10@gsuad.gsu.edu@snowball homeworks]$ vim homework_instructions.txt
```

Step 3: When going into the editor, I was in “command mode”. I pressed *i* and now I’m in “insert mode”. I looked back at the first page of the homework, copied the instructions and pasted it in the *homework_instructions.txt* file. After finishing copying each instruction line by line, I pressed Esc to go back to “command mode”. I then typed “:wq” which makes sure to write and quit (saving and exiting).

Step 4: I went back and checked to see if the .txt file was saved (vim homework_instructions.txt).

```
[vdo10@gsuad.gsu.edu@snowball homeworks]$ vim homework_instructions.txt
[vdo10@gsuad.gsu.edu@snowball homeworks]$ ls
homework_instructions.txt
```

Step 5: I wanted to see the current permission listings of the file before changing them (ls -l). I changed the permission so that only I can edit the file and everyone else can read the file (chmod 644 homework_instructions.txt). I checked again to see if my command went through (ls -l)

```
[vdo10@gsuad.gsu.edu@snowball homeworks]$ ls -l
total 4
-rw-rw-r--. 1 vdo10@gsuad.gsu.edu vdo10@gsuad.gsu.edu 1124 Sep 21 13:42 homework_instructions.txt
[vdo10@gsuad.gsu.edu@snowball homeworks]$ chmod 644 homework_instructions.txt
[vdo10@gsuad.gsu.edu@snowball homeworks]$ ls -l
total 4
-rw-r--r--. 1 vdo10@gsuad.gsu.edu vdo10@gsuad.gsu.edu 1124 Sep 21 13:42 homework_instructions.txt
```

Step 6: I went to find the statements with the word “POINTS” which was case sensitive (grep POINTS homework_instructions.txt). I wanted to see all of the statements affected by the case sensitivity, so I put “-i” to see the word “POINTS” in general (grep -i POINTS homework_instructions.txt).

```
[vdo10@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 TW
0 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.
[vdo10@gsuad.gsu.edu@snowball homeworks]$ grep -i POINTS homework_instructions.txt
Homework # 1: Total points 100
3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing in your document TW
0 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.
[vdo10@gsuad.gsu.edu@snowball homeworks]$
```