CSc 3320: Systems Programming

Spring 2021

Homework

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: Dang Truong

Campus ID:

Panther #: 002420309

Part 1:

1. Linux refers to the kernel of the GNU/  
   Linux operation system, and Unix refers to the original operating system. Linux is more desktop OS friendly than Unix, because Unix lacks the user friendliness for general computing. Linux can go from mainframes down to low end computers, but for Unix can only go for mainframes high ends computers. FreeBSD, Mac OS X, and OpenBSD are belong to Unix and not Linux.
2. Pipe mechanism in Unix is providing one way traffic flow of data. It sends the output of one program to another program. $ ls -l | more, this means the more command takes the output of $ ls -l as its input and displayed the output of ls -l and gives it to more as input.
3. /bin- contains applications and executable commands and files.

/var- variable storage directory that contains data that can be altered during the system runtime. This stores files generated by services like mail or log files.

/boot- the kernel is residing in this directory and loads up during the boot-up.

/usr- all user related application and files are stored here.

/etc- contains system configuration files.

/mnt- we mount our filesystems and device files like CD-ROM under this directory.

/sbin- short for system binaries, this contains system administrative executable program files and commands that are only needed for booting, restoring, and repairing the system.

/dev- contails all the devices that can be accessed just like any other file in the linux file system.

1. Multitasking is operating system supports two or more processors running programs at the same time, while multiuser operating system enables two or more users to run programs simultaneously.
2. - rwx means the User(owner) permissions, owner can read, write or edit, and execute the given file.

* r-x means that group permission, usergroup can read and execute the given file but cannot write or edit the given file.
* r-x means other permission, others can also read and execute the given file but cannot write or edit the given file.

1. - Read permission means that the contents of a directory can be seen by the user.

* Write permission means that files can be created in the given directory by a user.
* Execute permission means that the user can access the directory that is present in the current directory.

Part2-a:

1. ‘a[ab]\*’a= aa, aaa, aba, aaba

* The matches string should begin and end with ‘a’ and in between there can be zero or more occurrences of any character from bracket.

1. ‘a(bc)?’= a, abc

* The matches string should begin with ‘a’ and can have ‘bc’ zero or one time after ‘a’.

1. ‘.[ind]\*’= mi, t, tn, td, ti, miidn

* The matched string should begin with any character except newline can have any character from bracket zero time or more times.

1. ‘[a-z]+[a-z]’= abcd, avc, ad

* The matched string should end with any alphabet ranging from a to z and also have an alphabet between a to z before the end character.

1. ‘[a-z] (\+[a-z])+’= a+b, a+b+b

* The matched string should start with any alphabet between range a to z and have at least one occurrence of + and any character between a to z.

1. ‘a.[bc]+’= 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 the bracket but more than one can also be there.

1. ‘a.[0-9]’= at0, ar9

* The matched string should start with a and end with a digit ranging from 0 to 9 and in middle there can be any character but only once.

1. ‘[a-z]+[\.\?!]’= a., abcd?, az!

* The matched string should start with any character between a to z and should end with any one character among ., ?, and !.

1. ‘[a-z]+[\.\?!]\s\*[A-Z]’= 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, after that there can be zero or more black spaces and the string should end with an upper case alphabet between a to z.

1. ‘(very )+(cool )?(good|bad) weather’= verycoolgood weather, veryverybad weather

* The matched string should contain one or more occurrences of very and after that should have zero or more occurrence of cool followed by good or bad word and in the end the string should have weather.

1. ‘-?[0-9]+’= 9, -2, -345, 454

* The matched string should start with zero or more occurences of – followed by one or more occurrences of any digit between the 0 to 9.

1. ‘-?[0-9]\*\.?[0-9]\*’= ‘ ‘, -0t9, -y, 9u9

* The matched string should contain zero or one occurrence of – followed by zero or more occurrences of any digit between 0 to 9 followed by zero or one occurrence, and in the end there should have zero or more occurrences of any digit between 0 to 9.

Part2-b:

1. /^(http):\/\/[\w\-\_]+(\.[\w\-\_]+)+([\w\-\.]\*+(.edu)?
2. (([1-9][0-9])\*|0)?
3. ([.\/]+[a-z]\*)\*
4. [\_a-Z]{10}
5. [0-9]{10}

[0-9]{3}-[0-9]{3}-[0-9]{4}

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

Part3:

1. mkdir homeworks
2. cd homeworks
3. vi homework\_instructions.txt
4. chmod go-w homework\_instructions.txt
5. :grep POINTS homework\_instructions.txt
6. :wq

