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: Ramey Serdah

Campus ID: rserdah1

Panther #: 002 48 9675

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 a variant of Unix and Unix is the original software. Linux is open source whereas Unix is not. System V, Solaris, and MacOS are operating systems that belong to Unix but not Linux.

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 allows the output of one command to be used as the input for another command. An example would be

```
awk 'BEGIN {FS=","} ; {print $3}' mountainList.txt | sort -bgr
```

This command would take the third group of the text file, pipe it to the sort command, and sort reverse numerically.

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 binary executables.
`/dev`	stands for device. It contains files used for devices like the CPU
`/boot`	contains files needed for startup/booting.
`/usr`	contains all the executables belonging to a user.
`/etc`	is a folder for files that don't have a dedicated folder.
`/mnt`	is a manual mount point for mounting devices and disks.
`/sbin`	is for binary executables that require superuser privileges.
`/var`	is for variable files that change in size and content frequently.

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

Multitasking is running multiple programs at once. Multi-user means that more than one user can use the system at once as well. Multi-user functionality is obtained by sharing processing among each user.

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?

This means that this is a file, not a directory; has read, write, and execute permissions for the owner/user; and has read and execute permissions for users in the same group and the public/others. The command to change permissions to this would be `chmod 755 <fileName>`

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.

For a directory, the read permission allows a user to list the contents of a directory; write to change contents of a directory (add, remove, rename files/directories); and execute to enter the directory (e.g. `cd <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' ***aba, abbbbba, aa, aaa*** Matches 'a', followed by either 'a' or 'b' zero or more times, and lastly 'a'

8) 'a(bc)?' ***a(bc)?*** Since '(', ')', and '?' are all extended meta characters and this is a basic regular expression, this matches exactly 'a(bc)?'.

9) '.[ind]*' ***z, zi, zn, zd, ziii, ai*** Matches one character, either 'i', 'n', or 'd' zero or more times.

10) '[a-z]+[a-z]' ***c+z, c+a, a+b*** Matches any lowercase letter followed by a '+', followed by any letter from 'a' to 'z'. The + meta character is not used in basic regular expressions.

11) '[a-z] (\+[a-z])+' ***f +a, a +j+j+j*** Matches any lowercase letter,

followed by a space, followed by the group of '+' or any lowercase

letter either occurring one or more times.

12) 'a.[bc]+' ***akbbb, alcccc*** Matches 'a' followed by any character,

followed by either 'b' or 'c' occurring one or more times.

13) 'a.[0-9]' ***ak0, al9, ah7*** Matches 'a' followed by any character,

followed by any digit from 0 to 9.

14) '[a-z]+[\.\?!]' ***sss., ff?, wwww!*** Matches any lowercase letter once or more, followed by either '.', '?', or '!'

15) '[a-z]+[\.\?!]\s*[A-Z]' ***jjj. A, aa?K*** Matches any lowercase letter once or more, followed by '.', '?', or '!', followed by whitespace zero or more times, followed by any capital letter.

16) '(very)+(cool)?(good|bad) weather' ***very cool good weather, very very bad weather*** Matches the group "very " once or more, followed by "cool " zero or one times, followed by either the group "good" or "bad" followed by " weather".

17) '-?[0-9]+' ***-44, 555, -7*** Matches '-' zero or one times, followed by any digit from 0 to 9 once or more.

18) '-?[0-9]*\.?[0-9]*' ***-888.555, 7, -, -5*** Matches '-' zero or one times, followed by any digit from 0 to 9, followed by any string, followed by a '.' zero or one times followed by any digit from 0 to 9 zero or more times.

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://).*(\.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/])*(.[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]{10}|[0-9]{3}-[0-9]{3}-[0-9]{4}|\([0-9]{3}\)-[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.

Step1: `mkdir homeworks`

Step2: `cd homeworks`

Step3: `vi homework_instructions.txt`

Step4: `:wq`

Step5: `chmod 644  homework_instructions.txt`

Step6: `vi homework_instructions.txt`

Step7: `:wq`

Step8: `grep -w 'POINTS' homework_instructions.txt`