

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: Faisal Musa

Campus ID: 002-39-6860

Panther #: 002-39-6860

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 open-source meaning its source code is available to the public, Unix is a closed source software

• Unix is a clone of unix

• Unix default shell is Bash, however Unix shells are: Bourne shell

• Unix has a fast kernel architecture, while Unix has user level file system, or user level dependency library pack

• OS that belong to Unix is: HP-UX, AIX, BSD, OS X (Belongs to Unix, just not Linux)

• OS that belong to Linux is: Redhat, Ubuntu, Solaris, OpenSuse (Belongs to Linux, just not Unix)

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

• Pipe mechanism is redirection (where the transfer of a standard output to another command) with help in Unix/Linux which sends the output of a command or program to another command or program. Also Linux/Unix systems allow for standard input or a command to help get connected to an stdin of another command or process and Linux/Unix pipes are bidirectional

Example syntax:

command-1 | command-2 | command-3 | command-n

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: Super binaries are stored here. Here are only work of root. /etc: Home for "root" and for "others" for some files stored out. /etc - Directory, maintained from the University and standard for "UNIX System Resources" and "Interface between hosts", which can be used to multiple hosts safely.

• /var: Stores for variable where system data that varies can be stored. /var: temporary files are mounted here

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

• Multitask OS enables 2 or more users to run programs simultaneously, while multi-tasking OS supports 2 or more processor running programs 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?

Position (s-t) r-x = permission for group to read and execute | Position 2-4 (rwx) = permission for user to read, write, execute

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 = Read permission on a directory gives you the ability to list its contents

• Write = Gives you authority to add, remove and rename files stored in the directory

• Execute = In Linux/Unix, you can't run a program unless execute permission is set

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' • Generates a string containing the symbol 'a' at start and one or more [] in UNX allows any listed characters w/ * making the 0 or more occurrences of preceding character
- abaaa , - aaaba , - aaabaa , - aaaa
- 8) 'a(bc)?' • ? is part of input string and is preceded by \ character, starts w/ a followed by (bc)?
- a(bc)?
- 9) '.[ind]*' • Starts w/ single character since a wild card is specified. The prefix character is replaced by any single character from input string, followed by zero or multiple occurrences of character at string
- .ine , - .ee
- 10) '[a-z]+[a-z]' • If '+' is not preceded by \ character, so all string characters w/ single characters will start from 'a-z' followed by a '+' sign and then ending w/ 'a-z'
- xxa
- 11) '[a-z] (\+ [a-z])+' • Extended regular expression, the '+' in parentheses is part of string input and preceded by \ character. All input strings w/ single character from 'a-z' and then followed by a + symbol and a single character from 'a-z' once or multiple times
- aabc
- xra
- 12) 'a.[bc]+' • Input starts w/ 'a' character followed by one (unique) single character, then followed by 'bc' string characters once/multiple times. String ends w/ 'b' or 'c'
- 0abc
- 0abc
- 0cc
- 13) 'a.[0-9]' • Input character starts w/ 'a' followed by (unique) any single character, then by 0 to 9 and string ends w/ a digit
- 001
- 14) '[a-z]+[\.\?!\]' • Starts w/ 'a-z' characters at least once or multiple times and ends w/ !, ?, or !) symbol. Since preceded by a '\' character and its considered part of
- good , - hor?
- 15) '[a-z]+[\.\?!]\s*[A-Z]' • Starts w/ 'a-z' at least once or multiple times, followed by symbols, then followed by single/multiple occurrences of white space characters and w/ uppercase character. String starts w/ lowercase letter, then ends in uppercase letter.
- book< - Z
- 16) '(very)+(cool)?(good|bad) weather' • Starts w/ one or multiple occurrences of string 'very' followed by one or zero occurrences
- very good weather
- very cool bad weather
- very cool
- very good
- very bad
- 17) '-?[0-9]+' • String starts w/ zero or one occurrence of '-' symbol, then followed by multiple occurrences of digits from 0 to 9
- (-223)
- 3312
- 0.5
- -3312
- 18) '-?[0-9]*\.[0-9]?' • String starts w/ zero or one occurrence of '-' symbol, then multiple occurrences of digits from 0 to 9
- 3312
- 0.5
- (-223)

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://]+)[\w\.-]+([\w\.-]+)+([\w\.-]+)*+(\.edu){2}

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-zA-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]{10}

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

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

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.

