Lab 4

More Bash plus Regular Expressions

CSCI 344 Shell Programming

Due Friday February 17, 11:59pm on Canvas
30 points

OBJECTIVE

The objective of this lab is to continue writing scripts using bash plus starting with regular expressions.

1. File permissions using *chmod*
2. Reading in input
3. Command line argument passing
4. Regular expressions

Part 1. (*4 pts.*) Bash with read input

Write a bash script that updates the ***owner*** permissions of a file. The script should prompt the user with 4 questions to capture the name of the file, the *read*, the *write*, and the *execute* permissions. Then, update the file's ***owner*** settings with the appropriate rwx settings. As per the ***group*** and ***other*** rwx permissions, these should be restricted (i.e. set to ---). Execution display should be similar to the following:

$ ./myscript.sh
Enter the name of the file > **myfile**
Do you want to allow read permission? > **yes**
Do you want to allow write permission? > **no**
Do you want to allow execute permission? > **yes**

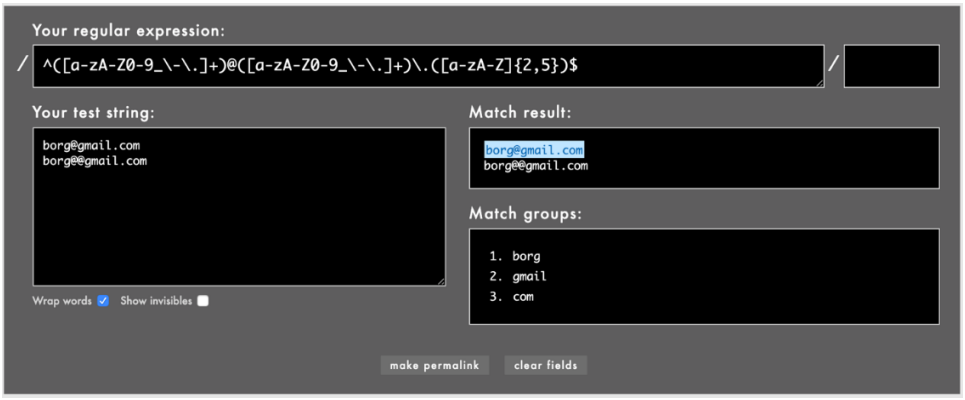
Part 2. (*4 pts.*) Bash with argument passing

Write a bash script that updates the ***owner*** permissions of a file similar to Part 1. The difference is that instead of prompting the user for the 4 values, the script's input values should come from the command line arguments. For example, with the first example below, *myscript.sh* sets the *read* and *execute* permissions to the ***owner*** of the file named *myfile*. And for the second example below, *myscript.sh* sets only the *write* permission to the owner of the file named *myfile*.

$ ./myscript.sh myfile read execute
$ ./myscript.sh myfile write

Part 3. (*22 pts.*) Regular Expressions

Create RegExps for the 17 problems listed below. Use http://rubular.com/ or http://regexpal.com/ to develop and test your RegExp. For each regular expression problem, test and provide screenshots. Answers may vary.
EXAMPLE SOLUTION: show the input data, your RegExp and the resulting match(es)



1. Email addresses.
borg@gmail.com
borg@@gmail.com

2. Telephone numbers in either of the following formats: (555)555-5555, 555-555-5555, or 555-5555.
(123)456-7890
123-456-7890
123-4567
1234-123

3. Zip codes in either of the following formats: 55555 or 55555-5555.
12345
12345-1234
123456-12345

4. Visa numbers start with 4 and are 13 or 16 digits.
4123456789012
041234567890123

5. MasterCard numbers are always 16 digits, the first is always 5, and the second digit is 1 through 5
5212345678901234
5012345678901234

6. American Express numbers are 15 digits and start with 34 or 37.
341234567890123
371234567890123
381234567890123

7 All repeating words which are space delimited. The words should be adjacent.
The house house is red
Greetings my my friend
Hello to you to

8. HTML text matching H1-H6 or h1-h6 tags. Matching tags should have the same case-sensitive H/h letter and same integer of values between 1 and 6. For example, the first 2 pairs below would match the RegExp but not the third pair listed.
<H6>Hello</H6>
<h3>Hello</h3>
<H5>Hello</H1>

9. URLs beginning with http:// and ending with .html suffix.
http://thisismy.site.html
http:thisismy.site.html

10. Files having the .sh extension.
file1.sh
file2.csh
file3.shsh

11. Patterns of digits having no leading zeros. For example, the following should match 4 patterns:
021 45 4 09 3 55 09

12. Patterns of digits representing even numbers. For example, the following should match 3 patterns:
2 22 29 45 34

13. Patterns of digits having occurrences of the digit 2 before occurrences of the digit 9. For example, the following should match 3 patterns:
22 29 92 45 4452893 23 22992 192

14. Provide 5 different examples of text which matches the following RegExp (note that the ending question mark ? is part of the RegExp):
(\+|-)?([0-9]+\.?[0-9]*|\.[0-9]+)([eE](\+|-)?[0-9]+)?

15. Provide the RegExp that matches words containing foo or bar, but not both. Make sure to show screenshots that your RegExp works for the following 3 test cases.
myfoo
youbarout
myfooxbarcow

SUBMITTING WORK

If you are only able to complete the lab partially, clearly state it. Explain where you time was spent and any hurdles that were or were not met. Given the steps mentioned above, your deliverables are the following:

1. Start by pasting as many screen shots that describe your work for each step into one continuous document.
2. Spend much time on adding your personal comments within your lab submission. Refer to syllabus and example as shown in class. Anyone should be able to read your lab write-up and understand what it is trying to present.
3. Turn in a copy of all your RegExp code, and include any screen shots of your output, all within your **one** PDF/Word formatted document. DO NOT SIMPLY TAKE SCREEN SHOTS. Include highlights and arrows to describe specific points.
4. Add a conclusion section to your write-up. Elaborate on at any aspect of your lab. Some examples are: what was challenging, how were you able to overcome any hurdle(s), and/or what did you learn.
5. *** Please refer to the required format specified in the syllabus (i.e. turn in one PDF/Word/OpenOffice document). No individual image files. And no zipped, no compressed and no rtf-formatted files.

GRADING

To achieve a maximum score, students will need to clearly prove that they completed the goal. A clear description of the steps taken, and screen shots are essential. Partial credit is given if students can clearly state what was done and what is not working. If the instructor is required to decipher incompleteness, much less partial credit will be given.

Points lost for incompleteness, sloppiness, lateness, or failure to follow instructions.

Late policy: refer to syllabus.