CSC 2510: DevOps

Lab 08 – Regular Expressions

# General Instructions

# Using your book and previous lecture material, fill out this assignment sheet. Use **red** text to signify your answers. Use information from your textbook and the provided lectures to answer the questions on this lab. You may utilize online resources to answer these questions if you cite them.

# Submission Instructions

To submit, attach your lab08.sh script to the iLearn dropbox.

# Lab Questions

For this lab, you will need to download phonebook.dat and place it into your home directory. Then you will write a single shell script that executes all of the required problems listed below on the given file*. If a file name is not given in the command, the script should execute on phonebook.dat by default.* Your shell script should **print the problem number and description in green**, and then execute the grep command that gives the answer. Note that it only has to execute the commands and output the results on the given file or default to the phonebook.dat, *there is no expectation of user input beyond the initial command*. Each command should only print out the lines with its corresponding description as listed below.

1. Starts or ends with Jose.
2. Contain at least 27 upper or lower-case alphabetic characters (a-z).
3. Consists of more than 18 characters. The characters can be anything, including alphabetic and numeric.
4. Contains exactly 10 characters. The characters can be anything, including alphabetic and numeric.
5. Contains a sequence between 6 and 8 upper- or lower-case alphabetic characters. The sequence must be separated from the rest of the line by a space or tab on each side.
6. Contains a local phone number. For example: 123-4567.
7. Contains a valid URL on a line by itself.

The rest of the document contains examples and help for each of these problems.

## Example Output

$ ./lab08.sh

1. starts or ends with Jose

Josephine

San Jose

San Jose

Jose

San Jose

San Jose

Joseph

2. start with at least 27 upper- or lower-case characters a-z

http://www.chinesetranslationresources.com

http://www.industrialpapershreddersinc.com

http://www.woodbridgefreepubliclibrary.com

http://www.marriotthotelsresortssuites.com

http://www.italianexpressfranchisecorp.com

http://www.kaminskikatherineandritsaki.com

http://www.baileytransportationprodinc.com

## Lab Prep: Using Grep

For this lab, your will be using a command called grep (GNU regular expressions). grep finds patterns of strings in files. By default, when grep finds a given string pattern in a file, it prints the line in which it found the pattern. Call up a man page on grep. The synopsis of grep is:

grep [options] [pattern] [file...]

The pattern is a regular expression. For our assignment, the most important option that we will use for grep is -E. This option turns on extended regular expressions.

So, what is a regular expression? It is a pattern that describes a set of possible strings. For example, the regular expression for any string with the word John in it is just 'John'. If you want to print the lines of a file called phonebook.dat that have the word John somewhere in the line, you would give the following command:

grep -E 'John' phonebook.dat

To help with the significant size of the file we can limit the number of matches to 10. This can be done with the -m flag as seen below. Note that there is *no space* between the m and the value. This flag should *only* be used while testing.

grep -E -m10 'John' phonebook.dat

Additionally, we can add color to our grep output to see what matched our expression by add a --color flag. Note there are two dashes in this flag.

grep -E -m10 --color=always 'John' phonebook.dat

## Viewing All of Your Output

To view all your output (with color) pipe your output to less with a -r flag as seen below.

./lab08.sh | less -r

## Printing in Color

So how do you get my problem to show up in green in the terminal? Use echo with the -e option and escape codes. For example, a command to print problem 1 looks like the following:

HI='\033[0;32m'

NORMAL='\033[0m'

echo -e "${HI}1. starts or ends with Jose$NORMAL"

Note that you only need to define the variables HI and NORMAL once at the top of you script file, and then use the variables in your echo commands. You use curly braces for HI inside of the echo string so that the dereference (the dollar sign) knows when the variable name begins and ends (e.g. using $HI1. starts instead of ${HI}1. starts would give an error).

## 1. Starts or Ends with Jose

Earlier in this lab, you saw how to find the lines with the string 'John' appearing anywhere in the line. However, for this problem, you only want to print lines that have 'Jose' at the very beginning or at the very end. You can specify that a pattern must be at the very beginning of the line using the ^ symbol. For example, the following will find lines with 'John' at the beginning.

grep -E --color=always '^John' phonebook.dat

Similarly, you can specify that a pattern must be at the very end of the line by following the pattern with a $ symbol, such as in the following example:

grep -E --color=always 'John$' phonebook.dat

To get lines where a pattern appears at the front or the back of the line, you use the | character. In the following example, grep prints the lines with either the string 'Mike' or 'John' in them.

grep -E --color=always 'Mike|John' phonebook.dat

Given the above description, you should now be able to answer problem 1 on your own.

## 2. Contain at least 27 upper or lower-case alphabetic characters

You can specify character classes with regular expressions using the [] symbols.  For example, if you wanted to find all lines with a digit in them, then you can use the following command:   
  
grep -E --color=always '[0-9]' phonebook.dat   
  
You can grep from a specific number of characters using the {} symbols.  For example, to find lines a sequence of three digits in them, you can use the following command:   
  
grep -E --color=always '[0-9]{3}' phonebook.dat

## 3. Consist of more than 18 characters

To represent any single character in a regular expression, you use the period.

## 4. Contains exactly 10 characters

To answer this problem, you only need to apply what you already know.  Finding a line that contains exactly 10 characters is that same as finding a line that begins and ends with exactly 10 characters.

## 5. Contains a sequence between 6 and 8 upper- or lower-case alphabetic characters

To match an expression if it occurs at least n times, but no more than m times, you can use {n,m}. For example, to match lines beginning with a sequence of 2 to 4 a, s, h, or i characters in any order, you can do the following:

grep -E --color=always '^[ashi]{2,3}' phonebook.dat

You can represent a space character in a regular expression with, you guessed it, a space. Tabs are represented with \t.

## 6. Contains a local phone number

This problem is a little tricky. To be a local phone number, the number has to be 3 digits followed by a dash and then 4 digits. However, the number cannot be preceded by a dash, because that signifies that the number has an area code in front of it. So, a local phone number is a number that has 3 digits, a dash, and then four digits and that is either at the beginning of a line, or somewhere else in the line but does not begin with a dash. So a typical phone number looks like 123-456-7890, but a local number would look like 123-4567.

You can specify that a pattern does not have a certain character by using the ^ symbol inside the [] symbols for character classes. For example, to find all lines that do not begin with a capital or lower-case a through w or a number, you can give the following command:

grep -E --color=always '^[^a-wA-W0-9]' phonebook.dat

## 7. Contains a valid URL on a line by itself

For the purposes of this lab, a valid URL starts with http:// or HTTP:// and is followed by any sequence of upper- and lower-case a-z characters, a dot, another sequence of upper- and lower-case characters, another dot, and then com or edu.

To make this regular expression work, your will need to use grouping with the () symbols. The URL can start with http or HTTP, so the beginning of your regular expression should look like so: '^(http|HTTP)://'

If you do not use the (), then the ^ symbol will be associated with the first http, but not the rest of the expression that occurs after the |. You will have use grouping at the end of the expression when handling the com or edu also.

How do you represent a sequence of upper- and lower-case a-z characters? You can represent one or more patterns using the + symbol. So [a-zA-Z]+ represents the sequence of at least one upper or lower case characters.

You need one more piece of information. You need to represent a dot in your regular expression. However, the '.' is a special symbol that represents any character. If you want the dot to represent a literal '.', then you have to escape it like so: \.