CSCI 3308 Software Development Methods and Tools [Spring 2018]
Instructor: Alan Paradise
TAs: Chelsea Chandler, Ajay Kedia, Rohit Mehra, Michael Schneider, Nikhil Sulegaon

Regex

Objectives
- Use Regex with common UNIX/Linux commands
- Practice using useful UNIX/Linux commands (remember man can help you understand how each command operates!)
  o diff

  o grep

  o cut

  o uniq

  o sed

  o awk
- Practice creating and running bash shell scripts
- Practice using pipes

Lab 2 Exercise
For each step, record the commands (and options) that you used to complete the task in a file called Lab2_Solutions.txt. At the end you will receive credit for the lab by showing your TA these commands.

Download Practice Files From Moodle
For Today's lab we will be using the following data files:
- scene1_v1.txt
- scene2_v2.txt
- password_demo.txt
- grades.txt
- cryptic.txt
- regex_practice_data.txt

Download Lab2 from Moodle and copy the provided zip file to your home directory and decompress it using the unzip command:
    unzip Lab2.zip -d Lab2
    cd Lab2
Check to make sure each of the above files was correctly unzipped into the Lab2 directory.

Part 1: Use Unix Commands to compare Monty Python Scripts!

<u>Step 1</u>: - **Use the diff command**
    A. Use diff to display all the lines that have changed from scene1_v1.txt to get to scene2_v2.txt.

    B. For the original diff output, what do the '>' or '<' character mean at the beginning of each line?

    C. Try using the –c option, what does that do?

For the Steps 2 - 4, we will have 2 problems to solve. The first will demonstrate a standard use case for a specific unix command (grep, cut, uniq). The second problem utilizes piping to build a complex unix statement that will eventually use all of the unix commands at once!

<u>Step 2:</u> - **Use the grep command**
    A. Use grep to display each line that contains the word "pigeon", as well as its line number, in scene1_v1.txt

    B. Use grep to display the lines that were modified in scene1_v1.txt?
    (Hint: Pipe the output from your first diff command into the grep command)

<u>Step 3</u>: - **Use the cut command**
    A. Using the delimiter ':', display the name of the characters who are speaking in scene1_v1.txt (make sure to ignore any lines that do not include the delimiter).

    B. Now use cut to only display the name of the characters that have had their lines altered from scene1_v1.txt to scene1_v2.txt.

<u>Step 4:</u> - **Use the uniq command**

    A. Use the uniq command to list only the duplicate lines in scene1_v1.txt.

    B. Use uniq to show how many times each character has had their lines altered from scene1_v1.txt to scene1_v2.txt.
        As a note on uniq, only compares adjacent lines. To find all repeated lines, the text must be sorted

Part 2: Working with Regular Expressions & AWK

Step 5: - **Use the sed command**

    A. Using sed and regular expressions try playing around with cryptic.txt file.
        1) Remove all the letters
        2) Replace all numbers with an '_'

    B. Using pipes, create a script that pipes together multiple sed commands to replace each number with its matching character. How can this be done without piping?
       LEET Alphabet Used:

       a – 4   e – 3   i – 1   o – 0 (oh – zero)
       s – 5   t – 7

Step 6: - **More practice with regular expressions**

For the following problems use grep or egrep with the regex_practice_data.txt file.

    A. How many phone numbers are in the dataset?

    B. How many city of Boulder phone numbers (e.g. starting with 303-441-...)?

Step 7: - **Use the awk command**

    pizzaOrders.txt Column Descriptions:
        ID - Order IDentification Number
        TP - Total Number of Pizzas Ordered
        NP - Number of Pepperoni Pizzas Ordered ($5.50)
        NS - Number of Sausage Pizzas Ordered   ($5.75)
        NC - Number of Cheese Pizzas Ordered   ($5.00)
        TC - Total Cost

    A. Using pizzaOrders.txt, print out the average cost per pizza for each order.

    B. Using pizzaOrders.txt, calculate and print the percent of all pizzas sold that were cheese.

Credit: To get credit for this lab exercise, get your Lab2_Solutions.txt file checked with your TA and submit the file on Moodle. All partners should submit copies of the same file.