Lab 4: Shell Scripting

Nathan Jarus

February 22, 2016

Introduction

In this lab, you'll get to use `grep` and `sed` to run some regular expressions you've written. Writing good regular expressions is like solving a puzzle!

You should make an answers text file to answer assignment questions.

Problem 1: That filter problem, AGAIN

So it turns out that all that C++ could be done with `grep` instead!

1. Write a command to filter out lines beginning with '#'. (You can test it with `story-plain.txt`.)
2. Modify the previous command to filter out lines beginning with whitespace and then '#' as well. (You can test it with `story-space.txt`.)

Problem 2: Lk m, n vwls!

Write a `sed` command to remove the vowels from a file.

Problem 3: Counting Lines

Your friend happens to run a service that generates random numbers. Each user request is recorded in a log file with the following format:

```
<YYY-MM-DD> <HH:MM:SS> <number>
```

Your friend wants to know how many requests in `numbers.log` were made in January. (Hint: `wc -l` will count the number of lines in files or STDIN.)

Problem 4: Phone numbers

`phonebook.txt` contains phone numbers (surprise, surprise). But, those numbers are not written in a consistent format; I just put them in however I felt like at the time. It'd be nice to have those formatted in a consistent manner.

1. Write a `grep` command to match all the phone numbers in the file.

2. Write a `sed` command to format all the numbers like so: `(ddd) ddd-dddd`.
   (Hint: backreferences are your friend!)
   (Hint 2: Sadly, `sed` doesn't have `\d`, so you'll have to use `[0-9]` instead.)