Lab 4: Shell Scripting

Nathan Jarus

February 10, 2016

Introduction

This lab will give you some experience writing shell scripts. You will need to sign in to https://git.mst.edu and `git clone` the repository for this lab. Your repository will be named something along the lines of `2016SP-A-lab04-nmjxv3`.

I strongly advise you to experiment and to test your code as you go along! Bash is a little weird, so checking to make sure it's doing what you think it is is important.

Problem 1: Menus and Files

Make a shell script that loops through all the files passed as arguments to it (or, if no files are given, everything in the current directory) and for each file, prints out a menu:

```
v) View file
e) Edit file
c) Compile file
x) Execute file
q) Quit
```

Then the script gets the user's choice and does it. Consult the following list to see what 'it' is:

- View file: Open the file with `less`
- Edit file: Open the file in an editor
- Compile file: Compile the file with `g++`
- Execute file: Run the file
- Quit: Bail out of the loop with `break` or `exit`
- Anything else: Print an error message and go to the next file

Hint: You can assign a wildcard to a variable, e.g. `textfiles=*.txt`.

Problem 2: Doing some math

CSV files are files that contain comma separated values. You can imagine them like a spread sheet where each row of the file is a row of the spreadsheet and commas indicate where the columns are. (In fact, you can import CSV files into various spreadsheet programs! They are nice for outputting tables from programs you write.) For an example, consult the `data.csv` file in your repository. (You can just open it in any text editor.)

Make a shell script that takes one CSV file as an argument. The script prints out the sum of the values in each row of the file and then prints the average of all row sums. For example, your output may look something like this:

```
Line 1: 15
Line 2: 195
Line 3: 2048
Average: 752
```

Hints:

- You can use `$IFS` to tell bash what character it should split a line into words on
- You can assign arithmetic results to variables like so: `(( a = 4 + 3 ))`.

Note: `(( ))` does integer division only, so your answer won't be exactly right, but it'll be close enough.

Epilogue

As with lab 3, your git repo on git.mst.edu is your submission. Don't forget to `git add` all the files you want to submit, `git commit` them, and `git push` your changes so the graders can download them!