

# CSCI 3194 (Seminar (UNIX Power Tools)), Fall 2014

## Homework 3

**Credit:** 30 points.

### 1 Reading

Be sure you have read the assigned readings for classes through 10/29.

### 2 Programming Problems

Do the following “programming” problems. (They’re not exactly programming problems, but I want you to submit your answers by e-mail in case I need to enlist a computer’s help to decide whether they work.) Turn in your answers by sending mail to `bmassing@cs.trinity.edu`. You can send me one file per problem (e.g., make a miniature shell script for each problem), or put them all in a single file; either way, send the file or files as attachments. Please use a subject line that mentions the course number and the assignment (e.g., “csci 3194 homework 3”). If there’s any question of which file(s) correspond to which problems, explain in the body of the mail message. You can develop your “programs” on any system that provides the needed functionality, but I will test them on one of the department’s Linux machines, so you should probably make sure they work in that environment before turning them in.

#### 2.1 Regular expressions

1. (5 points) Write a `grep` command to search all the files in the current directory for lines containing either your Linux username (e.g., `bmassing`) or your last name. (You don’t have to also search subdirectories, just files in the current directory itself.)
2. (5 points) Write a `grep` command to search all files in the current directory for strings that could be Social Security numbers (strings of the form `nnn-nn-nnnn`, where each `n` is a decimal digit).
3. (5 points) Suppose you have a Java program in file `Foo.java` containing the not-very-well-named variable `Bar`, and you want to change this variable to `bar` everywhere it occurs, without changing other occurrences of the string `Bar` (such as `myBar` or `BarFoo`). Write a `sed` command that makes this change. (Note that essentially the same command would work in `vim`.) Don’t worry about saving the output back into the file; the point here is the command to make the change.
4. (15 points) Suppose you have a text file (call it `foo`) containing a list of names and e-mail addresses. Each line contains first an e-mail address (which for simplicity we’ll define as any combination of letters, digits, underscores, and periods, followed by a `@`, followed by any combination of letters, digits, underscores, and periods), then some delimiter (a space, a colon, a comma – anything that can’t be part of the address as we defined it), and then a name. Write a `sed` command to reverse the addresses and names, removing the delimiters and leaving at least one space between the name and the address. For example, “`me@mail.com: This Is Me!`” would be changed to “`This Is Me! me@mail.com`”. (*Hint:* Backreferences may be useful.)

Again, don't worry about saving the changes back to the file. Also don't worry about what happens to lines that don't have the form described (e.g., no e-mail address), or about pruning out extra spaces.