

# CSCI 3294 (Seminar: UNIX Power Tools), Fall 2016

## Homework 5

**Credit:** 30 points.

### 1 Reading

Be sure you have read, or at least skimmed, the assigned readings for 9/28.

### 2 Honor Code Statement

Please include with each part of the assignment the Honor Code pledge or just the word “pledged”, plus one or more of the following about collaboration and help (as many as apply).<sup>1</sup> Text *in italics* is explanatory or something for you to fill in. For written assignments, it should go right after your name and the assignment number; for programming assignments, it should go in comments at the start of your program.

- This assignment is entirely my own work.
- This assignment is entirely my own work, except for portions I got from the assignment itself (*some programming assignments include “starter code”*) or sample programs for the course (*from which you can borrow freely — that’s what they’re for*).
- I worked with *names of other students* on this assignment.
- I got help with this assignment from *source of help — ACM tutoring, another student in the course, the instructor, etc.*
- I got significant help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (“Significant” here means more than just a little assistance with tools — you don’t need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*
- I provided significant help to *names of students* on this assignment. (*“Significant” here means more than just a little assistance with tools — you don’t need to tell me about helping other students decipher compiler error messages, but beyond that, do tell me.*)

### 3 Programming Problems

(These aren’t exactly programming problems, but I want your answers by e-mail so I can enlist the computer’s help in checking them. So, follow the standard instructions below, except that’s okay to just send me all your answers in the body of the e-mail.)

Do the following programming problems. You will end up with at least one code file per problem. Submit your program source (and any other needed files) by sending mail to `bmassing@cs.trinity.edu` with each file as an attachment. Please use a subject line that mentions the course

---

<sup>1</sup>Credit where credit is due: I based the wording of this list on a posting to a SIGCSE mailing list. SIGCSE is the ACM’s Special Interest Group on CS Education.

and the assignment (e.g., “csci 3294 hw 5” or “UNIX hw 5”). 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.

### 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 Scala program in file `foo.scala` 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 just defined), 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.