# CSCI 3294 (Seminar: UNIX Power Tools), Fall 2016
## Homework 6

**Credit:** 20 points.

## 1   Reading

Be sure you have read, or at least skimmed, the assigned readings for 10/05.

## 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).[1] 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

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 and the assignment (e.g., "csci 3294 hw 6" or "UNIX hw 6"). 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.

---

[1]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.

`make`

1. (20 points)   Do one of the following:

   - Suppose you have a C program consisting of the following files:
     - <u>defs.h</u>[2] containing macro and constant definitions.
     - <u>foo.h</u>[3] containing constants and prototype definitions for function `foo`.
     - <u>foo.c</u>[4] containing code for function `foo`.
     - <u>bar.h</u>[5] containing constants and prototype definitions for function `bar`.
     - <u>bar.c</u>[6] containing code for function `bar`.
     - <u>main.c</u>[7] containing code for the main program, which calls functions `foo` and `bar`.

     Write a makefile that:
     - When you type `make main`, creates executable `main` from the source files, compiling just the parts of this program that need to be recompiled (because either source code has changed or an `#include`'d file has changed). Compile using any C compiler (`cc` or `gcc`) but using the flags `-Wall`, `-pedantic`, and `-O`.
     - When you type `make clean`, deletes all the object files for the program.
     - When you type `make xclean`, deletes all the object files for the program and the executable.

     You can make use of the implicit rules defined by `make` if they will help. (They probably will.) Try to make good use of makefile variables to reduce duplication.

     Be advised that the command `touch` can be used to change a file's timestamp without opening it in an editor, etc. (This may be helpful in testing.)

   - If you have worked with `make` before and would have no trouble with the preceding problem, write your own problem for this assignment: Think of something interesting or useful you could do with `make` and write a makefile that will do it. (You should plan to come up with something you can finish in an hour or two.) Include comments at the top of the makefile describing what is meant to do.

---

[2]http://www.cs.trinity.edu/~bmassing/Classes/CS3294_2016fall/Homeworks/HW06/Problems/defs.h

[3]http://www.cs.trinity.edu/~bmassing/Classes/CS3294_2016fall/Homeworks/HW06/Problems/foo.h

[4]http://www.cs.trinity.edu/~bmassing/Classes/CS3294_2016fall/Homeworks/HW06/Problems/foo.c

[5]http://www.cs.trinity.edu/~bmassing/Classes/CS3294_2016fall/Homeworks/HW06/Problems/bar.h

[6]http://www.cs.trinity.edu/~bmassing/Classes/CS3294_2016fall/Homeworks/HW06/Problems/bar.c

[7]http://www.cs.trinity.edu/~bmassing/Classes/CS3294_2016fall/Homeworks/HW06/Problems/main.c