CSCI 2321 (Computer Design), Spring 2021

Homework 3

Credit: 30 points.

1 Reading

Be sure you have read, or at least skimmed, the assigned readings from Chapter 2 up through 2.8, plus sections A.9 and A.10 (up to the list of instructions).

2 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 my TMail address with each file as an attachment. Please use a subject line that mentions the course and the assignment (e.g., "csci 2321 hw 3" or "computer design hw 3"). 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.

For each problem in this homework, your mission is to translate some C code into a complete program for SPIM, replacing the calls to `scanf()` and `printf()` with equivalent SPIM system calls. (Programs `echo.s` and `echoint.s` under "sample programs" on the course Web site show how to input and output text and integer values.)

1. (15 points) A problem with (nested) loops: C code in `p1-loops.c`.
2. (15 points) A problem with a recursive function. C code in `p2-recursion.c`.

I think both of the problems above are very doable if you take them step by step — e.g., start by writing the input/output code, get that working, and then consider how to write the rest. However, if you'd prefer to start with some much simpler programs, here are two simpler ones, which you can submit for partial credit:

- (Up to 2.5 points)

  One step beyond the "echo integer" example: C code in `p0a.c`.

- (Up to 5 points)

  A very simple loop. C code in `p0b.c`.

Tips:

- Note that one benefit of starting from a C program is that you can run the C program to check that your answers are correct.

- In written problems asking you to translate from C to MIPS I ask you to avoid pseudoinstructions. But for programming problems, I think it makes more sense to use pseudoinstructions when they make the job easier without too much performance cost (meaning that you should still use, e.g., `sll` rather than `mul` to multiply by a power of 2).
- It may be helpful to review the GCD example discussed in the recorded lecture for 3/15 and available on the course "sample programs" page.

3 Pledge

For programming assignments, this section should go in the body of the e-mail or in a plain-text file `pledge.txt` (no word-processor files please). For written assignments, please put it in the text or PDF file with your answers.

Include the Honor Code pledge or just the word "pledged", *plus* at least one of the following about collaboration and help (as many as apply). Text *in italics* is explanatory or something for you to fill in.

- I did not get outside help *aside from course materials, including starter code, readings, sample programs, the instructor.*
- 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, etc. (Here, "help" means significant help, beyond a little assistance with tools or compiler errors.)*
- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (Here too, you only need to mention significant help — 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 help to *names of students* on this assignment. *(And here too, you only need to tell me about significant help.)*

4 Essay

For programming assignments, this section should go in the body of the e-mail or in a plain-text file `pledge.txt` (no word-processor files please). For written assignments, please put it in the text or PDF file with your answers.

Include a brief essay (a sentence or two is fine, though you can write as much as you like) telling me what if anything you think you learned from the assignment, and what if anything you found interesting, difficult, or otherwise noteworthy.