CS270 Recitation 11
“LC-3 Lite Simulator Project Session”

Goal: Help you make progress on Programming Assignment 6.

The Assignment

This recitation should help you with PA6. If you have already done some of the assignment, proceed to the first step that you haven’t completed. You may not finish all the steps during the recitation, but you can use this document to help you when you’re working on your own.

Step One

If you have not started on PA6, go to the assignment and read it. Download the package PA6.PROVIDED.tar.gz using this link. Make a directory called PA6 in your home directory and move the code there.

Step Two

This project consists of a complex system (the lc3-lite simulator) for which its simplest components (the individual instructions BR, STR, NOT, etc.) have been removed. You will need to re-implement these components to make the simulator work again.

Can you fix the lc3-lite simulator in one try (i.e., implement all of the instructions, throw them all together, and expect the lc3-lite simulator to work)? Probably not. More likely, the successful approach will be to implement and test each LC-3 instruction individually to ensure that each one works, and then integrate them to test the entire lc3-lite simulator.

For step two, consider using a test-driven development approach—that is, first write test cases for a particular LC-3 instruction of code before implementing that instruction (such that it passes the your test cases).

Pick any two of the LC-3 instructions, and create test cases for them (considering their equivalence classes). Don't forget boundary cases. Show these test cases to the TA for recitation credit.

Step Three (optional)

Testing the individual LC-3 instructions in this assignment may be tricky. One solution is to write your own main function which is capable of initializing and running each instruction individually. We have provided a template test file which does just that (some assembly required).

Download the skeleton test file test.c and develop a test for the ADD instruction. This will require you to first initialize the machine state (set IR to an ADD instruction, and set SR1 and SR2 to the desired values). This will also require you to print out the results (i.e., the DR).

The test driver can be compiled by copying it into your PA6.PROVIDED directory and executing:

```
linux prompt> gcc -std=c99 test.c logic.c lc3sim.c lc3memlib.a infilesparser.c -o test
```