

CS270 Recitation 12

“Help with Programming Assignment 6”

Goal: Help you make progress on Programming Assignment 6.

The Assignment

This recitation should help you with PA6. 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

Go to class schedule page and read the PA6 assignment. Download [PA6.tar](#) in the Assignments/PA6 directory. Make a directory called PA6 in your home directory and move the code there.

Step Two (Questions)

- 1) Generate the .sym file from the given testST.asm file. What are the contents of the .sym file? Explain how symbol table is generated from symbol.c file. (describe in 2-3 sentences the methods: add_symbol(), find_symbol() and remove_symbol_at_addr()).
- 2) What do the enums field_t and format_t represent in file lc3sim.c? Explain their use in function print_operands().
- 3) Explain the following functions (in file lc3sim.c) with respect to the LC-3 architecture.
 - i. memory_enable()
 - ii. load_MAR()
 - iii. load_MDR()
 - iv. memory_read()
 - v. memory_write()
- 4) Explain the array ccodes

Submit the answers in file R12.txt for recitation credit.

Step Three (optional)

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

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

For this step, 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 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.

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