CS270 Recitation 14
"Help Session for LC-3 Assembler"

Goals

To help students with the LC-3 assembler assignment, or more specifically:

1. To explain the LC-3 assembler assignment in more detail and answer any questions you might have.
2. To provide some hints about various algorithmic and coding issues, without giving you the code.

Due to limitations on time and resource, this recitation is not intended as a help session to debug your C code!

Class Discussion

The teaching assistant will lead a discussion on what various functions should do, including:

asm_pass_one()

asm_pass_two()

Utility Functions

The teaching assistant will lead a discussion on how various utilities can help, including:

symbol.c/symbol.h // provided by student

tokens.c/tokens.h // linked in lc3as.a

util.c/util.h // linked in lc3as.a

Help Topics

1) The teaching assistant will discuss how to fill in the fields in a line_info structure, in order to pass the first round of the assignment. Friyz will also be supplying a working reference, to allow you to perform testing. Some details below:

- DR, SR, and CC are stored in the DR field, however there is an SR macro you can use.
- SR1 and BaseR and stored in the SR1 field.
- Immediate values and offsets are stored in the immediate field.
- Trap vectors, .FILL values, .BLKW counts, and .ORIG address are stored in the immediate fields.
- You should write a parseNumber that accepts #1234, xAABB, 1234 formats.
- The parseNumber should also take a maximum number of bits, and report an error if the number is too large.
- PC offsets are generated 'on the fly' during the second pass, and therefore do not have a field.

2) The teaching assistant will show how to run lc3convert on a .hex file to create an object file, and will show the format for the .hex, and .sym files.

3) The teaching assistant will discuss how the machine code (hex value) for an instruction can be generated, and will work through the example ADD R1, R2, #-5.

4) The teaching assistant will describe how to correctly evaluate an immediate value to make sure it fits into the number of bits allocated for it in the instruction, and how to store it in a 32-bit number, and combine it back into the immediate field.

5) The teaching assistant will describe how to find PC offsets from the delta between the address of the current instruction and the label it references, e.g. LD R0, MYDATA, taking into account the increment of the PC during decode.

6) The teaching assistant will describe how to handle the assembly opcodes .FILL and .BLKW in order to correctly generate machine code for these instructions.