CS 203 Lab #4 - Learning Assembler

C W Liew

September 18, 2018

Goals

The purpose of this assignment is to start learning about assembly language programming for the ARM processor and examine the code transform that gcc performs to generate the assembly code from C code.

- learn ARM assembler (the GNU version)
- learn how to use the ARM simulator

Errata

For this course, we will be using the GNU compiler for the arm process (**arm-linux-gnueabi-gcc**). You can compile the example files from Lab 1 (sum.c, add.c) on the remote machine using:

```
arm-linux-gnueabi-gcc -S sum.c
```

Assignment

There are two parts of the assignment:

- The first part of the assignment is to recognize the different code and code sequences that are generated from C code. Given the assembly code in the file *exp.s* generated on the remote machine (with the -S option), what is the matching order of C math expressions?

  1. $y = -3 * x$
  2. $y = 8 * x - 3$
  3. $y = (x * 4) + (x - 8)$
  4. $y = -1$
  5. $y = (4 * x - 1)/8$
  6. $y = (8 * x) + (x - 3)$
  7. $y = 4 * x - 8$
  8. $y = (x/8) + (4 * x)$
  9. $y = x - 12$

- arm assembler simulator: The simulator we are using this semester can be found at: `ttps://salmanarif.bitbucket.io/visual/index.`tml. There are differences between what the GNU compiler (arm-linux-gnueabi-gcc) generates and what the simulator supports. The assignment is to translate the generated assembler into a version that will run on the simulator.