Lab for ITSC 2181, Module 06 - Unit 1

Converting C programs to RISC-V assembly and simulating their execution using RARS simulator

We will use the RISC-V Assembler and Runtime Simulator (RARS) for this lab, which is available from https://github.com/TheThirdOne/rars.

A video introducing how to use RARS simulator is available from https://passlab.github.io/ITSC3181/resources/UsingRARS_ITSC3181.mp4.

1. Download the latest RARS jar file from https://github.com/TheThirdOne/rars/releases/download/v1.6/rars1_6.jar on your computer. You should then be able to launch the jar file by either double clicking it or from launcher such as using the following command in the Mac OS X terminal:

   ```
   java -jar rars1_6.jar
   ```

2. Create and execute the hello world program following instructions from https://github.com/TheThirdOne/rars/wiki/Creating-Hello-World.
   a. Spend some time reading and understand each line of that page.
   b. In RARS, create the Hello World program, and then assemble and run the program by clicking the sub-menu items of the Run menu.
   c. Play with the example, menu items and the RARS interface to get familiar with the RARS tool.
   d. Check the address, binary code, instructions and source of the assembled code, and also check the register values and memory values (data segment part) of the program execution.
   e. After you run the program multiple times, you should run it step-by-step, i.e., instruction by instruction, and observe the change of values in registers and other locations.

3. Read the document Fundamentals-of-RISC-V-Assembly.

4. Make sure you understand the code structure of an assembly program:
   a. There are two sections, the `.text` section and the `.data` section, in the program.
   b. The `.text` section is for the code and the `.data` section is for the values used in the program, e.g., string constants.

c. There is a "main" label in the `.text` section and an "str" label in the `.data` section. The two labels are symbols representing the addresses of the memory locations where corresponding code or data are stored in memory.

Submission Instructions:

1. **Submit a single PDF file that shows the execution screenshot of the *Hello World* program in RARS.**

2. **Submit the Assembly source (`.asm` file) of the *Hello World* program.**