

CPS 500 - Lab 2

William Katsak

Due: 02/01/2019 11:59pm ET

Overview

For this assignment, you will get some hands-on experience with MIPS assembly language using the MARS MIPS simulator.

Cheating

Please be sure that all work is your own. Looking up the solution on the Internet to any of these problems counts as cheating. If caught, you will receive a 0 for the assignment and will be reported to the academic integrity office.

Background and Setup

For this lab you need to install the MARS MIPS simulator. You can download MARS from <http://courses.missouristate.edu/KenVollmar/mars/index.htm>. You can either install the simulator in your VM, or on your main computer.

This simulator allows you to write and run MIPS assembly language programs. The simulator itself is written in Java so you will need a JRE (Java runtime environment) which most of you will likely already have. Installation instructions for running Java can be found here:

https://docs.oracle.com/javase/8/docs/technotes/guides/install/install_overview.html

You do not need any knowledge of Java to use this software.

What to turn in

Please submit your assembly language file from MARS.

Assignment (100 pts)

Description

Write a MIPS assembly snippet that manipulates an array of 9 elements as done in the following C snippet. Your assembly code can assume the number of elements in the list (as shown in the snippet), but it should not depend on their values.

```
int temp = array[0];
array[0] = array[8] + 1;
array[8] = temp - 1;

temp = array[1];
array[1] = array[7] + 3;
array[7] = temp - 3;

temp = array[2];
array[2] = array[6] + 5;
array[6] = temp - 5;

temp = array[3];
array[3] = array[5] * 8;
array[5] = temp / 8;
```

Restrictions

You may use only the following instructions:

- lw: Load word
- sw: Store word
- sll: Shift left logical
- srl: Shift right logical
- add: Add two registers
- addi: Add a constant to a register

Example Execution

```
// Given this input data:
[1, 3, 5, 7, 9, 11, 13, 15, 17]
// Your code should produce this output data:
[18, 18, 18, 88, 9, 0, 0, 0, 0]
```

Starting Point

Begin with the following assembly snippet:

```
.data                                # Defines variable section of an assembly routine.
array: .word 1,3,5,7,9,11,13,15,17  # Defines an array of integers (the input).
.text                                # Start the instructions of the program.
la $s0, array                        # Moves the address of array into register $s0.

# Your instructions go here
```

Comments

Remember that `temp` is a local variable, and thus should be stored in a `$sX` register. Additionally, your code does not need to produce any "output", as the array is being modified in-place. You can verify correct operation by using MARS to inspect the contents of memory.