CSE 378: 09 Wi - Ceze

Homework 0: Getting Started with MIPS and SPIM

This assignment is intended to get you started with MIPS assembly language programming and familiarize you with the SPIM simulator and assembler. Taking the time to play with the simulator and verify your answers will pay off when you start on the labs and the next homework.

You will not be graded on this assignment. When you've answered the five questions at the end, check yourself using the solution. You are encouraged to work together and use whatever sources you find useful. There is a list of MIPS resources on the class homepage.

What you should do:

1. You should start by looking at the MIPS resources page. There you will find: 1) the link to the SPIM homepage, where you can find instructions on downloading and installing SPIM, 2) several short tutorials on SPIM, and 3) a template SPIM file to get you started with this assignment.

2. After installing SPIM, I would suggest downloading the sample assembly file (lecture02.s) posted with the slides from lecture 2. This will give you something to play around with and modify while you learn the ins and outs of SPIM. Don't worry about understanding everything in this program right now. Just try stepping through the code (F10) and setting a few breakpoints. Start paying attention to what happens to memory and registers when it starts executing the "random calculations" and the array example from class.

3. Now try to answer the five questions listed at the end of this assignment.

Don't forget to comment your code!

Go ahead and start getting in the practice of writing good comments as you write your code, rather than inserting them when you are done with the program. This will make it much easier for you to track down problems as you are writing, and will save you a lot of time when debugging. Leave the comments in the program when you turn it in as this will make it much easier for the TAs to figure out what you were trying to do with your code, and will be essential for getting partial credit if necessary.

Questions

The following problems are short exercises that will get you thinking about the details of assembly programming. Most of these problems can be completed successfully in 5 instructions or less. Note: Do No Harm! The problems describe the available registers. Your solutions may not use other registers, or destroy the input values. Also, *for this exercise*, avoid using pseudo-operations. It may be helpful to refer to the list of available MIPS instructions listed in Appendix A (starting on page A-51). When you are finished, a sample solution file is provided in a super secret location[1].

1. If a processor did not have a load immediate instruction (lui), how would you load constant values into registers? (Your answer should be in English, not MIPS)

2. Put 0x12348ABC in register $9. (Be Careful)

3. $8 and $9 contain 32-bit signed integers. Put the larger of the two in $10.

4. $8 contains 4 ASCII characters in the range [a-z]. Make the first and third characters upper-case. You may use $9 as a temporary register.

5. $8 holds the main memory address of an array of words; $9 is a signed integer index to the array, which we'll call N. Set the Nth entry in the array to 0x00000001. You may use $10 and $11 as temporaries.

[1] At the bottom of the MIPS Resources web page. Shhhhhhhh……..