CSE31: Lab #5 – Intro to MIPS

Overview

In this lab, we will get ourselves familiar with the programming environment we will use for MIPS programming.

Getting started

Before we begin any activities, create a directory (Lab_5) inside the CSE31 directory you created during Lab #1. You will save all your works from this lab here.

(Exercise) What is MARS?

Since we cannot manipulate the CPU inside our computers directly (none of us uses MIPS CPU, and for security reasons), we need a software to simulate a MIPS CPU for us. The simulator we are using is called ***MARS***.

TPS (Think-Pair-Share) activity 1 Paired with the classmate sitting next to you and do the following tasks (25 minutes):

1. Before we use a new tool, we will need to find out how to use it. MARS **DOES NOT** mean that MIPS is an alien language (well, sort of). Work with your partner and find out what MARS stands for.
2. Since you have found out what MARS stands for, you probably have found out the webpage of MARS as well. Visit the download page and download MARS in your computer. To run MARS, just double click the downloaded jar file. You will need Java to run it. (MARS is pre-installed in the lab computers, so no need to download it if you are using a lab computer.)
3. From the Tutorial materials page (you can find the link to it from the home page), save both tutorial materials (***MARS feature map*** and ***MARS tutorial***) as well as ***Fibonacci.asm*** in your Lab_5 folder.
4. Follow ***Part 1 (Basic MARS Use)*** of the tutorial using ***Fibonacci.asm*** and discuss the following questions:
    a. How do you load an assembly file?

b. How do you assemble (compile) the program?

c. How do you run the assembled program?

d. Can you run a program before assembling it?

e. If you want to run the assembled program line by line, how to do it?

f. How do you run the program again after it has finished running?

(Exercise) A simpler Fibonacci

The sample program from the tutorial looks too complicated. Let's use a simpler version of it. Load ***fib.s*** (yes, you can save your assembly programs as .s files instead) into MARS and assemble the code. Note that Fibonacci number calculation is as follows:

```
fib[0] = 0;

fib[1] = 1;

fib[n] = fib[n-1] + fib[n-2];
```

TPS (Think-Pair-Share) activity 2 Paired with the same classmate and answer the following questions (25 minutes):

1. What do the .data, .word, .text directives mean (i.e., what do you put in each section)?
2. What does line 10 do?
3. What does line 15 do?
4. How do you set a breakpoint in MARS? Set breakpoint on line 15 and list the steps of achieving this.
5. After your program stops because of a breakpoint, how do you continue to execute your code? How do you step through your code?
6. How can you find out the content of a register? How do you modify the value of a register manually while running the program?
7. At what address is ***n*** stored in memory? Calculate the 13th fib number by modifying this memory location.
8. Line 19 and 21 use the ***syscall*** instruction. What is it and how do you use it?

(Assignment 1, individual) Create myFirstMIPS.s

Write a new piece of MIPS code that, given a value in $s0, put into the $tX registers the following:

$t0 = $s0

$t1 = $t0 + 1

$t2 = $t1 + 2

$t3 = $t2 + 3

...

$t7 = $t6 + 7

In other words, for each register from $t1 to $t7, it stores the sum of the previous $tX register value and an incremental constant. The $s0 register contains the initial value. Don't set the value of $s0 in your code. Instead, learn how to set it manually with MARS (Hint: question 6 in TPS 2). Save your code as ***myFirstMIPS.s***.

What to submit

When you are done with this lab assignments, you are ready to submit your work. Make sure you have included the following ***before*** you press Submit:

- Your ***myFirstMIPS.s***, answers to the TPS activities in a text file, and a list of Collaborators.
- Your assignment is closed **7 days after this lab is posted** (at 11:59pm).
- You must demo your submission to your TA **within 14 days** (preferably during next lab so you will have a chance to make correction and re-submit/re-demo.)