

MIPS – Part 1

Assembly language holds an important place in the hierarchy of programming languages. It is much closer to the machine language that is understood by the computer than are high-level programming languages. While it gives you greater control over the behavior of the CPU, a program written in it requires many more instructions to accomplish a task than if it had been written in a higher-level language. In addition, each architecture has its own dialect to support the instruction set of a given processor. This lab will introduce the basics of assembly language with the MIPS architecture.

Software

For this lab, you will be working with the assembly language that was designed to run on the MIPS architecture.

Point a browser to goo.gl/v7Jcc3 and open the jar file that is downloaded to run MARS, the MIPS Assembler and Runtime Simulator.

First Program

To start, you will create a simple MIPS program to learn how to operate the simulator. Make a new file and enter the following code:

```
.text
addi $t1, $zero, 10
addi $t1, $t1, 3
addi $t1, $t1, 2
addi $t1, $t1, 1
```

To assemble your program, select **Assemble** from the **Run** menu. You must do this each time you want to run your program. Select **Go** from the **Run** menu to execute your program. To the right of the **Execute** tab, you should see all the **Registers** and their values. In the middle of the screen is the **Data** section, which shows the memory used by the CPU on the MIPS processor that you are simulating. In the register `t1`, you should see the value `0x00000010`. In this case, `0x` indicates that the value is represented in hexadecimal.

Answer the following questions:

1. What is the value stored in \$t1 in decimal? _____
2. This time, run the program one command at a time. **Assemble** the program and use the **Step** command to execute the program in steps.
 - a. What are the values of \$t1 and pc before the program has run?

 - b. What are the values of \$t1 and pc after the first addi has executed?

 - c. What are the values of \$t1 and pc after the second addi has executed?

 - d. What are the values of \$t1 and pc after the third addi has executed?

 - e. What are the values of \$t1 and pc after the fourth addi has executed?

3. What does pc stand for? What does it do?
4. What does this assembly program do (in 1 English sentence)?
5. What would the equivalent be in idiomatic Java?