An Assembly Language I.D.E. To Engage Students Of All Levels
** A Tutorial **
2007 CCSC: Central Plains Conference

Pete Sanderson, Otterbein College, PSanderson@otterbein.edu
Ken Vollmar, Missouri State University, KenVollmar@missouristate.edu
PART 1 OF TUTORIAL: UPDATED JANUARY 2015 BY THEODORE MANIKAS, SOUTHERN METHODIST UNIVERSITY, FOR MARS VERSION 4.5 (AUG. 2014)

MARS is a software simulator for the MIPS assembly language intended for educational use. We will explore the capabilities of MARS in this tutorial.

Part 1 : Basic MARS Use

The example program is **`Fibonacci.asm`** to compute everyone's favorite number sequence.

1. Start MARS from the Start menu or desktop icon.

2. Use the menubar File…Open or the Open icon to open Fibonacci.asm in the default folder. *(All icons have menubar equivalents; the remainder of these steps will use the icon whenever possible.)*

3. The provided assembly program is complete. Assemble the program using the icon

4. Identify the location and values of the program's initialized data. Use the checkbox to toggle the display format between decimal and hexadecimal □ Hexadecimal Values.
   - The twelve-element array **`fibs`** is initialized to zero, at addresses 0x10010000 … 0x1001002c.
   - The data location **`size`** has value 12_{10} (C_{16}) at 0x10010030.

Use the checkbox to toggle the display format between decimal and hexadecimal, □ Hexadecimal Values.

5. Locate the Registers display, which shows the 32 common MIPS registers. Other tabs in the Registers display show the floating-point registers (Coproc 1) and status codes (Coproc 0).

6. Use the slider bar to change the run speed to about 10 instructions per second. This allows us to "watch the action" instead of the assembly program finishing directly.

7. Choose how you will execute the program:

   - The ▶ icon runs the program to completion. Using this icon, you should observe the yellow highlight showing the program's progress and the values of the Fibonacci sequence appearing in the Data Segment display.
   - The ⏪ icon resets the program and simulator to initial values. Memory contents are those specified within the program, and register contents are generally zero.
   - The ▶1 icon is "single-step." Its complement is 1◀, "single-step backwards" (undoes each operation).

8. Observe the output of the program in the Run I/O display window:

```
The Fibonacci numbers are:
1 1 2 3 5 8 13 21 34 55 89 144
-- program is finished running --
```

9. Modify the contents of memory. (Modifying a register value is exactly the same.)
   - Set a breakpoint at the first instruction of the subroutine which prints results. Use the checkbox at the left of the instruction whose address is 0x0040005c.
   - Reset ⏪ and re-run ▶ the program, which stops at the breakpoint.
   - Double-click in one of the memory locations containing the computed Fibonacci numbers. The cell will be highlighted and will accept keyboard entry, similar to a spreadsheet. Enter some noticeably different value, and use the Enter key or click outside the cell to indicate that the change is complete. *Example: Memory address 0x10010020 = 268501024 $_{ten}$ presently contains data 0x00000022 = 34 $_{ten}$.*
   - Click ▶ to continue from the breakpoint. The program output includes your entered value instead of the computed Fibonacci number.

10. Open the Help ? for information on MIPS instructions, pseudoinstructions, directives, and syscalls.