

# Using SPIM

Tom Kelliher, CS 220

Sept. 30, 2009

## Starting SPIM; Running a MIPS Program

### 1. Starting SPIM:

- (a) Logon to phoenix and open a shell window.
- (b) Run SPIM:

```
% spim
```

### 2. Running a MIPS program from SPIM:

- (a) Load a MIPS program in SPIM memory:

```
(spim) load "addn.spim"
```

Note that the quotes are required.

- (b) Run the program in memory:

```
(spim) run
```

- (c) Reboot the simulator:

```
(spim) reinitialize
```

- (d) Exit the simulator:

```
(spim) quit
```

- (e) The `help` command gives a list of the available commands.

## SPIM Debugging

- 1. Remember, you can only set breakpoints at instructions which are labeled with a global (`.globl`) label.

- 2. Set a breakpoint:

```
(spim) br while
```

- 3. Continue normal execution from a breakpoint:

```
(spim) co
```

- 4. Single step from a breakpoint:

```
(spim) st
```

Step through next `n` instructions:

```
(spim) st n
```

5. Print the value of register `n`:

```
(spim) print $n
```

This also works for the symbolic register names.

6. Print the values of all registers:

```
(spim) print_all_regs
```

7. Print the value of memory word `n`:

```
(spim) print n
```

Exercise:

1. Modify `addn.spim` so that you can break on the unconditional branch which terminates the loop.
2. Run SPIM, load `addn.spim`, and set a breakpoint at the end of the loop.
3. Run the program and examine the values of the sum and loop count at the end of each iteration of the loop. Do these values make sense?

## Assembly Language Programming

The easiest way to write an assembly language program is to first write a HLL program and then compile it manually. Rewrite the following C program in MIPS assembly and run it in SPIM.

```
/*  
 * fibonacci.c --- Compute and display the Fibonacci sequence.  
 *  
 * Tom Kelliher  
 *  
 * Input: Length of sequence to generate.  
 *  
 * Output: Fibonacci sequence of given length.  
 *  
 * Notes: The boundary conditions of lengths 0 and 1 aren't handled  
 *        properly. Correcting this is left as an exercise to the  
 *        student.  
 */
```

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int len;          /* Use $s0. */
    int i;           /* Use $s1. */
    int current;     /* Etc.    */
    int old;
    int older;

    printf("Enter sequence length: ");
    scanf("%d", &len);
    len = len - 2;

    old = 1;
    older = 1;

    printf("1\n1\n");

    for (i = 0; i < len; ++i)
    {
        current = old + older;
        printf("%d\n", current);
        older = old;
        old = current;
    }

    exit(0);
}

```