CS 21 Laboratory 2: Register-to-register operations and Syscalls

University of the Philippines - Diliman
College of Engineering
Department of Computer Science

1 Register-to-register operations

Create an assembly language program. Do not forget the required comment header.

Using the Load Immediate command, load 3 into register t0 and 5 into register t1. Then add the command in Code Block 1.

Code Block 1 The add instruction

```
add $t2, $t0, $t1 # add instruction.
```

Run the program. What happens to the value of register t2?

2 Syscalls

We could get user input via the console using Syscall code 5. Its usage is demonstrated in Code Block 2. Upon encountering the syscall, a console must pop-up(see Figure 1),

Code Block 2 Syscall code 5

```
li $v0, 5 # load syscall read_int into $v0.
syscall # make the syscall.
```

where inputs could be made. *Note that SPIM could not move on the next instruction until it receives an input for the syscall.*

The user input is stored in register v0, the very same register where we store the desired syscall number immediately before calling the syscall. Since v0 is frequently used by the system, we could not let the input value stay there for long. Thus, after getting console input, it is good practice to transfer the value in v0 to another register. How would you do this? What instruction would you use? Consult the SPIM/MIPS manual.

3 To Try

- get two integers via the console and store their sum in a register of your choosing
- get two integers via the console and store their difference in a register of your choosing

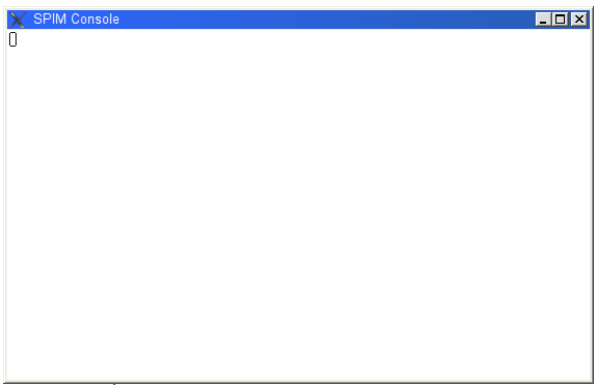

Figure 1: The console.

- get two integers via the console and store their product in a register of your choosing

4 Machine Exercise

Create an assembly language program that takes in **5** integers via the console, and then prints out two numbers at the console (refer to your MIPS Instruction Set handouts for syscall on printing integers): first, the *sum* of the 5 numbers; second, the *product* of the 5 numbers.

Brute force is allowed. (we've not yet discussed loops anyway)

It's okay if outputs are not separated.

Save the program under the filename **cs21me2.asm**. Again, do not forget the required header comments.

Call my attention upon being able to complete the ME for checking.

5 Homework

Register-to-register operation instructions usually have an *immediate version* such as *addi*, *subi*, etc. Figure out how those instructions are used.

6 Learning Checklist

In this session you should have learned...

- how to use syscalls
- how to transfer values between registers
- how to use register-to-register operation instructions