,

Computer Architecture (Lab). Week 6

Vladislav, Artem, Hamza, Manuel

Innopolis University

vl.ostankovich@innopolis.ru
a.burmyakov@innopolis.ru
m.rodriguez.osuna@innopolis.university
h.salem@innopolis.university

October 8, 2020

Topic of the lab

- MIPS Arithmetic Operations + MARS

MARS

- MARS is MIPS Assembler and Runtime Simulator.
- MARS is a lightweight Interactive Development Environment (IDE) for programming in MIPS assembly language
- It is intended for educational-level use

MIPS Register Names and Conventions

Number	Name	Usage
$0	$zero	constant 0x00000000
$1	$at	assembler temporary
$2–$3	$v0–$v1	function return values
$4–$7	$a0–$a3	function arguments
$8–$15	$t0–$t7	temporaries
$16–$23	$s0–$s7	saved temporaries
$24–$25	$t8–$t9	more temporaries
$26–$27	$k0–$k1	reserved for OS kernel
$28	$gp	global pointer
$29	$sp	stack pointer
$30	$fp	frame pointer
$31	$ra	return address

Syscall Services

Service	$v0	Arguments / Result
Print Integer	1	$a0 = integer value to print
Print Float	2	$f12 = float value to print
Print Double	3	$f12 = double value to print
Print String	4	$a0 = address of null terminated string
Read Integer	5	$v0 = integer read
Read Float	6	$f0 = float read
Read Double	7	$f0 = double read
Read String	8	$a0 = address of input buffer, $a1 = maximum number of characters to read
Exit Program	10	
Print Char	11	$a0 = character to print
Read Char	12	$v0 = character read

Example 1: Hello World!

In MARS create a file, copy and paste next code, assemble it, then run

```
# Author: Your name
# Date: Today Date
# Description: A simple hello world program!

.data                   # add this stuff to the data segment
                        # load the hello string into data memory

    hello: .asciiz "Hello, world!"

.text                   # now we are in the text segment

    li $v0, 4               # set up print string syscall
    la $a0, hello           # argument to print string
    syscall                 # tell the OS to do the syscall
    li $v0, 10              # set up exit syscall
    syscall                 # tell the OS to do the syscall
```

Example 2: Read and Write an Integer

In MARS create a file, copy and paste next code, assemble it, then run. Instead of $zero you can put another register

```
# Description: Addition of two numbers!
    #------------ Code segment ---------------
    .text
    .globl main
     main:                              # main program entry
            li       $v0, 5             # Read integer
            syscall                     # $v0 = value read

            add      $a0, $v0, $zero    # $a0 = value to print
            li       $v0, 1             # Print integer
            syscall

            li       $v0, 10            # Exit program
            syscall
```

Example 3: Addition of Integers (1)

In MARS create a file, copy and paste next code, assemble it, then run.

```
# Description: Addition of two numbers!
        .data
        msg1: .asciiz "Enter the first number: "
        msg2: .asciiz "\nEnter the second number: "
        result: .asciiz "\nThe result of addition is: "

        .text
        li $v0,4
        la $a0,msg1
        syscall

        li $v0,5
        syscall
        move $t1,$v0

        li $v0,4
        la $a0,msg2
        syscall
```

Example 3: Addition of Integers (2)

```
li $v0,5
syscall
move $t2,$v0

add $t3,$t1,$t2

li $v0,4
la $a0,result
syscall

li $v0,1
move $a0,$t3
syscall

li $v0,10
syscall
```

Exercise 1

- Write a program that will give result for $x = (y * z^2)/f - q$
- All numbers should be integer.

Exercise 2

- Write a program that prints 10 Fibonacci numbers

Useful links

- http://www.mrc.uidaho.edu/mrc/people/jff/digital/MIPSir.html
- http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html
- http://progopedia.ru/implementation/assembler-mips/
- http://logos.cs.uic.edu/366/notes/mips%20quick%20tutorial.htm

Acknowledgements

- This lab was created and maintained by Vitaly Romanov, Aidar Gabdullin, Munir Makhmutov, Ruzilya Mirgalimova, Muhammad Fahim, Vladislav Ostankovich, Alena Yuryeva and Artem Burmyakov