CS240 Lab 2 More on MIPS

Lab Assignment 2 Problem 1

Assume that the starting address of the text segment is 0x400000, and that the starting address of the data segment is 0x10010000.

Show the contents of all 4 bytes for each register and memory location, using hexadecimal notation.

```
   .text
  .globl main

main:   li $v0,4                    // $v0 = 0x00000004

        la $a0,prompt               // $a0 = 0x10010000

        lb $t0,value                // $t0 = 0x00000004

        lh $t1, value               // $t1 = 0x00000304

        lw  $t2, value              // $t2 = 0x01020304


        .data
prompt:  .asciiz "Enter a value: "
value: .word 0x01020304
```

Lab Assignment 2 Problem 2

	$v0	$a0	$t0	x	y
	0x00000000	**0x00000000**	**0x00000000**	**0x00000003**	
.text					
.globl main					
main: li $v0,4	**0x00000004**				
la $a0,prompt		**0x100100000**			
syscall					
li $v0,5	**0x00000005**				
syscall	**0x00000007**	#assuming a ‘7’ was entered by the user			
lw $t0,x			**0x00000003**		
add $t0,$t0,$v0			**0x0000000A**		
sw $t0,y					**0x0000000A**

```
     .data
prompt:      .asciiz  “Enter a value”
x:           .word   3
y:           .word   0
```

Lab Assignment 2 Problem 3

	.data	Address	Label	Data(bytes)	Address	Label	Data(words)
v:	.asciiz “8”						
w:	.space 8	10	Z	08			
x:	.half 8	0F		00			
y:	.word 8	0E		00			
z:	.byte 8	0D		00			
		0C	Y	08			
		0B		00			
		0A	X	08			
		09		00			
		08		00			
		07		00			
		06		00			
		05		00			
		04		00	10	Z	00000008
		03		00	0C	Y	00000008
		02	W	00	08	X	00080000
		01		00	04		00000000
		0x10010000	V	0x38	0x10010000		0x00000038
							W **V**

Two’s Complement Arithmetic

- The most significant bit of an n-bit number is used to indicated sign (+ or -)
- To determine the two’s complement form of a negative number, take the positive binary version of the number, flip all the bits, and add 1
- A positive and negative number added together cannot produce an overflow
- Two numbers of the same sign added together produces an overflow if the result is the opposite sign of the two numbers
- An addition can produce a carry-out without it indicating an overflow (so, carry-out is not the same as overflow!)

Example:

Given a 5 bit number, using two’s complement, represent 5 and -5

00101 = +5 (flip bits and add 1)→ 11011 = -5

```
              00101
Cout         +11011
1             00000   no overflow!

            10001      -15 + -14 = -29, which will not fit in 5 bits
Cout       +10010
1           00011      overflow!
```

Lab Exercise 3

Write a MIPS program which does the same thing as the following Java statements.

```
//initialize only these two strings
String phrase = “Change: inevitable”;
String addon = “ except from vending machines”;

//should output ‘Change: inevitable’
System.out.println(phrase);

//should output ‘Change: inevitable except from vending machines’ with a single output call
phrase = phrase.concat(addon);
System.out.println(phrase);

//should output ‘Charge!’
phrase = phrase.replace(‘:’,’!’);
phrase = phrase.substring(0,6)
phrase = phrase.replace(‘n’,’r’);
System.out.println(phrase);
```

```
    .text
        li $v0,4        #output the first string
        la $a0,phrase
        syscall

        li $t0,’  ‘      #output both strings by replacing the null with a space
        sb $t0,18($a0)
        syscall


    .data
phrase:  .asciiz      “Change: inevitable”
addon:  .asciiz       “except from vending machines”
```