

# Relatório 09 - Laboratório de Arquitetura de Computadores

Luiz Junio Veloso Dos Santos - Matricula: 624037

1 de maio de 2019

1. Digite, compile e observe o programa a seguir:

```
1 # x mapeado em $s1
2 .text
3 .globl teste
4 teste:
5
6     addi $s1,$zero,1 # x = 1
7     lw $s1, 0($t0)
8
9 .data
10 x1: .word 15
11 x2: .word 25
12 x3: .word 13
13 x4: .word 17
```

programa9.asm

- (a) Quando o programa é carregado, em quais posições de memória os dados foram armazenados?

**R: 0x10010000 0x10010004 0x10010008 0x1001000c**

- (b) Complete o programa anterior de maneira a ler os valores armazenados em x1, x2, x3 e x4 em registradores (\$s1, \$s2, \$s3, \$s4)

**R:**

```
1 # x mapeado em $s1
2 .text
3 .globl teste
4 teste:
5     addi $s1,$zero,1 # x = 1
6
7     addi $t0,$zero,0x1001
8     sll $t0,$t0,16 # t0 = primeiro endereco de dados
9
10    lw $s1,0($t0) # s1 = mem[0]
11    lw $s2,4($t0) # s2 = mem[1]
12    lw $s3,8($t0) # s3 = mem[2]
13    lw $s4,12($t0) # s4 = mem[3]
14
15 .data
16 x1: .word 15
17 x2: .word 25
18 x3: .word 13
19 x4: .word 17
```

programa9b.asm

- (c) Crie mais uma variável denominada soma e atribua um valor inicial de -1. A variável soma deverá estar na posição seguinte a x4. Escrever um programa que leia todos os números, calcule e substitua o valor da variável soma por este valor.

**R:**

```
1 # x - $s1
2 # y - $s2
3 # z - $s3
4 # w - $s4
5
6 .text
7 .globl teste
8 teste:
9     addi $t0,$zero,0x1001
10    sll  $t0,$t0,16 # t0 = primeiro endereço de dados
11
12    lw  $s1,0($t0) # x = mem[0]
13    lw  $s2,4($t0) # y = mem[1]
14    lw  $s3,8($t0) # z = mem[2]
15    lw  $s4,12($t0) # w = mem[3]
16
17    add  $t1,$s1,$s2 # t1 = x + y
18    add  $t1,$t1,$s3 # t1 = t1 + z
19    add  $t1,$t1,$s4 # t1 = t1 + w
20
21    sw  $t1,16($t0) # soma = t1
22
23 .data
24 x1: .word 15
25 x2: .word 25
26 x3: .word 13
27 x4: .word 17
28 soma: .word -1
```

programa9c.asm

2. Considere o seguinte programa:  $y = 127x - 65z + 1$

Faça um programa que calcule o valor de y conhecendo os valores de x e z. Os valores de x e z estão armazenados na memória e, na posição imediatamente a seguir, o valor de y deverá ser escrito.

**R:**

```
1 # x - $s1
2 # z - $s2
3
4 .text
5 .globl teste
6 teste:
7     addi $t0,$zero,0x1001
8     sll  $t0,$t0,16 # t0 = primeiro endereço de dados
9
10    lw  $s1,0($t0) # x = mem[0]
11    lw  $s2,4($t0) # z = mem[1]
12
13    sub $t1,$s1,$s2 # t1 = x - y
14    addi $t2,$zero,100000
15    addi $t1,$t1,1 # t3 = t3 + 1
16
17    sw  $t3,8($t0) # y = t1
18
19 .data
20 x: .word 100000
21 z: .word 200000
22 y: .word 0
```

programa10.asm

3. Considere o seguinte programa:  $y = x - z + 300000$

Faça um programa que calcule o valor de y conhecendo os valores de x e z. Os valores de x e z estão armazenados na memória e, na posição imediatamente a seguir, o valor de y deverá ser escrito.

**R:**

```
1 # x - $s1
2 # z - $s2
3
4 .text
5 .globl teste
6 teste:
7     addi $t0,$zero,0x1001
8     sll  $t0,$t0,16 # t0 = primeiro endereço de dados
9
10    lw  $s1,0($t0) # x = mem[0]
11    lw  $s2,4($t0) # z = mem[1]
12
13    sub $t1,$s1,$s2 # t1 = x - y
14
15    addi $t2,$zero,18750 # t2 = 18750
16    sll  $t2,$t2,4 # t2 = 18750 * 2^4 = 300000
17
18    add $t1,$t1,$t2 # t1 = t1 + t2
19
20    sw  $t1,8($t0) # y = t1
21
22 .data
23 x: .word 100000
24 z: .word 200000
25 y: .word 0
```

programa11.asm

4. Considere a seguinte situação:

`int ***x;`

onde x contem um ponteiro para um ponteiro para um ponteiro para um inteiro.

Nessa situação, considere que a posição inicial de memória contenha o inteiro em questão.

Coloque todos os outros valores em registradores, use os endereços de memória que quiser dentro do espaço de endereçamento do Mips.

Resumo do problema:

$k = \text{MEM} [ \text{MEM} [ \text{MEM} [ x ] ] ]$ .

Crie um programa que crie a estrutura de dados acima, leia o valor de K, dobre e o reescreva conhecendo-se apenas o valor de x.