

Programação em Assembly MIPS I

1. Qual o código C que corresponde ao seguinte código assembly do MIPS? Considere que as variáveis f, g, h, i, e j e que o endereço base dos vetores A[] e B[] estão atribuídos aos registos \$s0, \$s1, \$s2, \$s3, \$s4, \$s6 e \$s7, respetivamente. Considere ainda que os elementos dos vetores A[] e B[] ocupam 1 word (4 bytes).

- a. `add $s0, $s1, $s2`
`add $s0, $s3, $s0`
- b. `sll $t0, $s0, 2`
`add $t0, $s6, $t0`
`lw $t1, 0($t0)`
`lw $t2, 4($t0)`
`add $t0, $t1, $t2`
`sll $t1, $s1, 2`
`add $t1, $s7, $t1`
`sw $t0, 0($t1)`
- c. `li $t2, 100`
`move $t1, $s6`
`li $s3, 0`
`loop: lw $t0, 0($t1)`
`add $s0, $s0, $t0`
`addi $t1, $t1, 4`
`addi $s3, $s3, 1`
`blt $s3, $t2, loop`

2. Qual o código assembly do MIPS que corresponde ao seguinte código C? Considere que as variáveis f, g, h, i, e j e que o endereço base dos vetores A[] e B[] estão atribuídos aos registos \$s0, \$s1, \$s2, \$s3, \$s4, \$s6 e \$s7, respetivamente. Considere ainda que os elementos dos vetores A[] e B[] ocupam 1 word (4 bytes).

- a. `h = A[0] << 4;`
- b. `B[8] = A[i - j];`
- c. `B[8] = A[i] + A[j];`
- d. `f = A[B[g] + 1];`

3. O que calcula o seguinte programa?

```
.data
_msg1: .asciiz "The result is "
_msg2: .asciiz "\n"

.text
_main:
    li    $s0, 0
    li    $s1, 100
    li    $s2, 0
_loop:
    bge    $s0, $s1, _end
    mul    $t0, $s0, $s0
    add    $s2, $s2, $t0
    addiu  $s0, $s0, 1
    j      _loop
_end:
    li    $v0, 4
    la    $a0, _msg1
    syscall
    li    $v0, 1
    move  $a0, $s2
    syscall
    li    $v0, 4
    la    $a0, _msg2
    syscall
```

```
li    $v0, 10
syscall
```

4. O que calcula o seguinte programa?

```
.data
_str: .asciiz "abracadabra"
_msg1: .asciiz "The result is "
_msg2: .asciiz "\n"

.text
_main:
    la    $a0, _str
    jal   _proc
    move  $s0, $v0
    li    $v0, 4
    la    $a0, _msg1
    syscall
    li    $v0, 1
    move  $a0, $s0
    syscall
    li    $v0, 4
    la    $a0, _msg2
    syscall
    li    $v0, 10
    syscall
_proc:
    li    $v0, 0
_loop:
    lb    $t0, 0($a0)
    beqz  $t0, _end
    addiu $v0, $v0, 1
    addiu $a0, $a0, 1
    j     _loop
_end:
    jr    $ra
```

5. Qual o valor do registo \$t2 no final das seguintes sequências de instruções assembly do MIPS?

- a.

```
li    $t0, 0xAAAAAAAA
li    $t1, 0x12345678
sll   $t0, $t0, 4
or    $t2, $t0, $t1
```
- b.

```
li    $t0, 0xAAAAAAAA
sll   $t1, $t0, 4
andi  $t2, $t1, -1
```
- c.

```
li    $t0, 0xAAAAAAAA
srl   $t1, $t0, 3
andi  $t2, $t1, 0xFFEF
```
- d.

```
li    $t0, 0x10010000
li    $t1, 0x11223344
sw    $t1, 0($t0)
lbu   $t2, 0($t0)
```

6. Indique um conjunto mínimo de instruções que podem ser utilizadas para implementar as seguintes pseudo-instruções do assembly do MIPS.

- a.

```
not $t1, $t2    # bitwise inversion
```
- b.

```
bge $t1, $t2, label    # branch if greater or equal
```