[C208] Atividade - Revisão

**Questão 1 -** Analise as afirmativas a seguir e as classifique com verdadeiras ou falsas. Além disto, explique o porquê daquelas que classificadas como falsas. Não é necessário explicar as verdadeiras.

(F ) É possível afirmar que os compiladores são usados para gerar o programa executável a partir do código objeto.

Correção: É GERADO O CODIGO OBJETO A PARTIR DO CODIGO FONTE EM ALTO NIVEL.

Os compiladores são utilizados para transformar o código-fonte de um programa em código objeto

(F ) A compilação cruzada é realizada somente quando se tem sistemas operacionais iguais mas em versões diferentes.

Correção: É indepejndente do SO

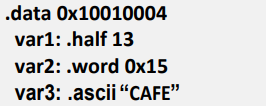
( F) O código objeto se difere do código executável apenas pela etapa de montagem.

Correção: Se difere pelo Linker

(F ) Os programas executáveis gerados em dois computadores idênticos com sistemas operacionais diferentes, a partir do mesmo código fonte em baixo nível, serão sempre iguais.

Correção: Serão diferentes devido ao SO.

**Questão 2 -** Considere a seguinte parte de programa em linguagem Assembly do MIPS:

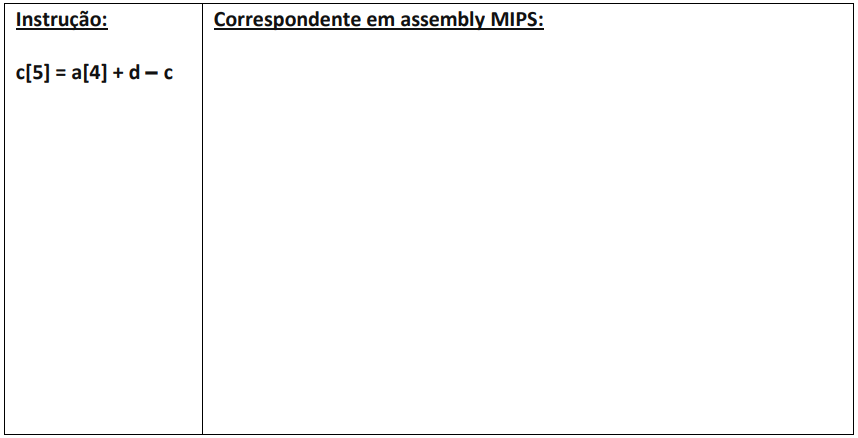


Complete o quadro abaixo considerando as variáveis declaradas no código acima. (Não há necessidade de se utilizar hexadecimal para representar a palavra “CAFE”). Ométodode armazenamento é Big Endian.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MEMÓRIA DE DADOS** | | | | |
| **Endereço** | **Dado** |  | **Endereço** | **Dado** |
| 0x10010000 |  | 0x10010008 | 0x00 |
| 0x10010001 |  | 0x10010009 | 0x00 |
| 0x10010002 |  | 0x1001000A | 0x00 |
| 0x10010003 |  | 0x1001000B | 0x15 |
| 0x10010004 | 0x00 | 0x1001000C | C |
| 0x10010005 | 0x0D | 0x1001000D | A |
| 0x10010006 | 0x00 | 0x1001000E | F |
| 0x10010007 | 0x00 | 0x1001000F | E |

**Questão 3:** Converta a seguinte instrução em assembly MIPS. Considere a = $s0, b = $s1, c =

$s2, d = $s4. Tente reutilizar os registradores temporários.



.text

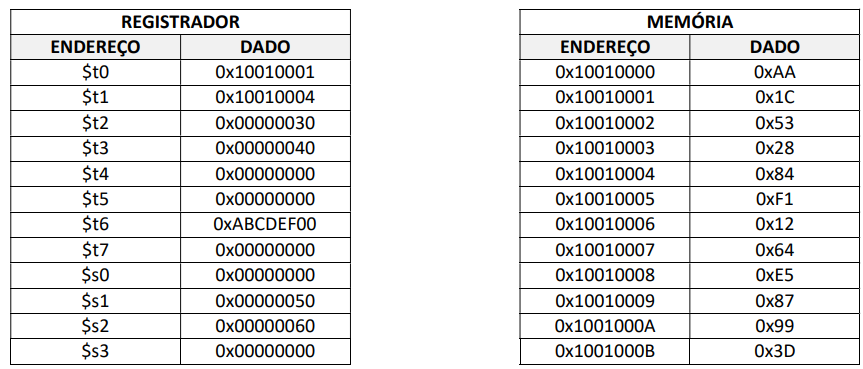
Lw $t1, 16($s0) #$t1 <- a[4] (4x4)

Add $t1, $t1, $s4 #$t1 <- $t1+$s4

Sub $t1, $t1 - $s2 #$t1 <- $t1-$s2

Sw $t1, 20($s2) #$t1 <- c[5] (5x4)

**Questão 4 :** Dado o estado atual dos registradores e memória de dados:

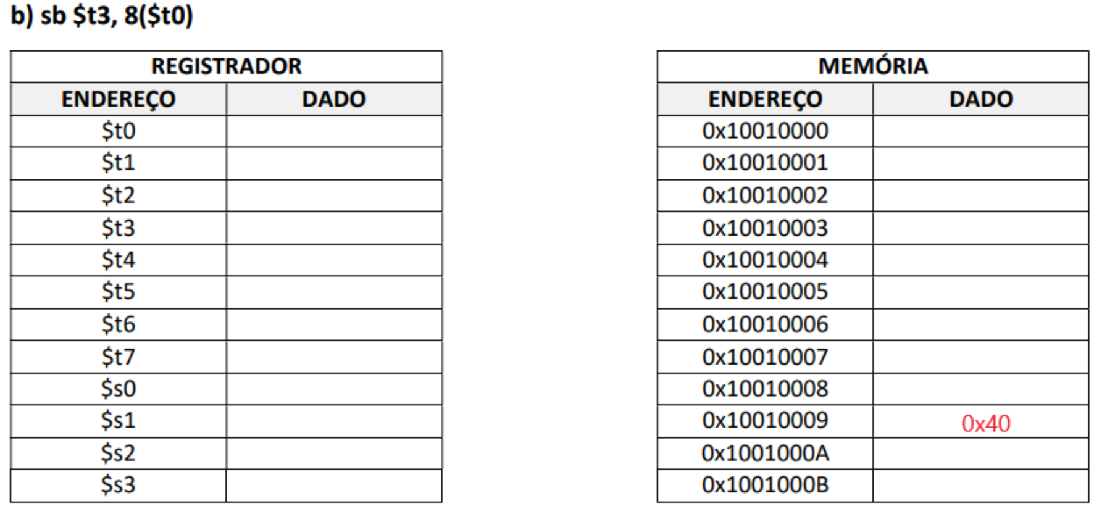


Mostre os efeitos na memória e nos registradores das seguintes instruções, considerando que cada uma é executada independentemente das outras.

# a) lh $t2, 4($t1)

# 

**b) sb $t3, 8($t0)**



PEGA O VALOR DE $T3, Vai no endereco de $t0 e desloca em 8

**Questão 5**: Escreva um programa em Assembly MIPS que faça a leitura de três valores numéricos inteiros fornecidos pelo usuário e exiba uma mensagem informando se a soma destes é maior, menor ou igual a 100.

.data

A: .byte 0x00

B: .byte 0x00

C: .byte 0x00

F1: .asciiz “Entre com o primeiro numero”

F2: .asciiz “Entre com o segundo numero”

F3: .asciiz “Entre com o terceiro numero”

F4: .asciiz “\n Maior do que 100”

F5: .asciiz “\n igual a 100”

F6: .asciiz “\n Menor do que 100”

.text

#entre com o primeiro valor

La $a0, F1

Li $v0, 4

Syscall

Li $v0, 5

Syscall

Add $t1, $v0, 0

#entre com o segundo valor

La $a0, F2

Li $v0, 4

Syscall

Li $v0, 5

Syscall

Add $t2, $v0, 0

#entre com o terceiro valor

La $a0, F3

Li $v0, 4

Syscall

Li $v0, 5

Syscall

Add $t3, $v0, 0

#ETAPA DE PROCESSAMENTO

Add $t1, $t1, $t2

Add $t1, $t1, $t3

Blt $t1, 100, menor

Bgt $t1, 100, maior

La $a0, F5

Li $v0, 4

Syscall

J fim

Menor:

La $a0, F6

Li $v0, 4

Syscall

J fim

Menor:

La $a0, F3

Li $v0, 4

Syscall

Fim: li $v0, 10

syscall

|  |  |  |  |
| --- | --- | --- | --- |
| **Senice** | Code |  | Result |
| print integer | I | $a0 = integer to print |  |
| print flat | 2 | gft2 = flat to print |  |
| print double | 3 | gft2 = double to print |  |
|  | 4 | $a0 = address ofnull-teuninated |  |
| ixdmeg‹x |  |  | $v0 contains integer read |
| eadQoW | 6 |  | $f0 contains Ooat read |
| ixddouble | 7 |  | $f0 contains double read |
|  | 8 | $a0 = address of input buffer  $a1 = maximum number of characters to read | M role h‹z/ou- /ofJe |
| * \*' h p   « | 9 | - number of bytes to allocate | $v0 contains address of allocated memory |
| exit  **execubon)** | 10 |  |  |