

## [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.

**Justificativa:** O compilador traduz código fonte para código objeto, e em seguida o código objeto precisa ser ligado (Link-editor) para se tornar um código executável.

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

**Justificativa:** A compilação cruzada ocorre quando se compila código em um sistema que é diferente do sistema onde o programa será executado.

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

**Justificativa:** O código objeto se difere do código executável pela etapa de montagem e ligação (Link-editor).

(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.

**Justificativa:** Os programas executáveis gerados por dois computadores idênticos com sistemas operacionais diferentes não são iguais, apesar de partir de um mesmo código fonte. Isso ocorre devido a diferenças nos sistemas operacionais, versões de bibliotecas, configurações de sistema e outros fatores que podem afetar o resultado final do processo de compilação e ligação.

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

```
.data 0x10010004
var1: .half 13
var2: .word 0x15
var3: .ascii "CAFE"
```

Complete o quadro abaixo considerando as variáveis declaradas no código acima. (Não há necessidade de se utilizar hexadecimal para representar a palavra "CAFE"). O método de armazenamento é Big Endia

MEMÓRIA DE DADOS			
Endereço	Dado	Endereço	Dado
0x10010000		0x10010008	00
0x10010001		0x10010009	00
0x10010002		0x1001000A	00
0x10010003		0x1001000B	15
0x10010004	00	0x1001000C	C
0x10010005	0D	0x1001000D	A
0x10010006	00	0x1001000E	F
0x10010007	00	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.

Instrução:	Correspondente em assembly MIPS:
c[5] = a[4] + d – c	<pre>.data  lw \$t1, 16(\$s0) # load word (32 bits), \$t1 &lt;- a[4] 4*4 = 16(\$s0)  add \$t1, \$t1, \$s4 # soma \$t1 e \$s4  sub \$t1, \$t1, \$s2 # subtrai resultado da soma anterior com \$s2  sw \$t1, 20(\$s2) # store word (32 bits), \$t1 &lt;- c[5] 4*5 = 20(\$s2)</pre>

Load (carregar): memória -> registrador

Store(armazenar): registrador -> memória

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

REGISTRADOR	
ENDEREÇO	DADO
\$t0	0x10010001
\$t1	0x10010004
\$t2	0x00000030
\$t3	0x00000040
\$t4	0x00000000
\$t5	0x00000000
\$t6	0xABCDEF00
\$t7	0x00000000
\$s0	0x00000000
\$s1	0x00000050
\$s2	0x00000060
\$s3	0x00000000

MEMÓRIA	
ENDEREÇO	DADO
0x10010000	0xAA
0x10010001	0x1C
0x10010002	0x53
0x10010003	0x28
0x10010004	0x84
0x10010005	0xF1
0x10010006	0x12
0x10010007	0x64
0x10010008	0xE5
0x10010009	0x87
0x1001000A	0x99
0x1001000B	0x3D

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)

REGISTRADOR	
ENDEREÇO	DADO
\$t0	
\$t1	
\$t2	0x000000f1
\$t3	
\$t4	
\$t5	
\$t6	
\$t7	
\$s0	
\$s1	
\$s2	
\$s3	

MEMÓRIA	
ENDEREÇO	DADO
0x10010000	
0x10010001	
0x10010002	
0x10010003	
0x10010004	
0x10010005	
0x10010006	
0x10010007	
0x10010008	
0x10010009	
0x1001000A	
0x1001000B	

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

REGISTRADOR	
ENDEREÇO	DADO
\$t0	
\$t1	
\$t2	
\$t3	0xE5
\$t4	
\$t5	
\$t6	
\$t7	
\$s0	
\$s1	
\$s2	
\$s3	

MEMÓRIA	
ENDEREÇO	DADO
0x10010000	
0x10010001	
0x10010002	
0x10010003	
0x10010004	
0x10010005	
0x10010006	
0x10010007	
0x10010008	
0x10010009	
0x1001000A	
0x1001000B	

**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 10.

```
.data
f1: .asciiz "Entre com o primeiro numero: "
f2: .asciiz "Entre com o segundo numero: "
f3: .asciiz "Entre com o terceiro numero: "
maior: .asciiz "Valor maior que 10!"
menor: .asciiz "Valor menor que 10!"
igual: .asciiz "Valor igual a 10!"

.text
li $v0, 4
la $a0, f1 # imprime frase1
syscall

li $v0, 5 # leitura do primeiro numero
syscall

add $t1, $v0, $0 # armazenando valor lido em $t1

li $v0, 4
la $a0, f2 # imprime frase2
syscall

li $v0, 5 # leitura do segundo numero
syscall

add $t2, $v0, $0 # armazenando valor lido em $t2

li $v0, 4
la $a0, f3 # imprime frase3
syscall

li $v0, 5 # leitura do terceiro numero
syscall

add $t3, $v0, $0 # armazenando valor lido em $t3
```

```
add $t4, $t1, $t2 # somando os 3 numeros lidos
add $t4, $t4, $t3
```

```
bgt $t4,10,MAIOR
blt $t4,10,MENOR
beq $t4,10,IGUAL
```

```
MAIOR: li $v0, 4
    la $a0, maior # imprime frase do maior
    syscall
    j fim
```

```
MENOR: li $v0, 4
    la $a0, menor # imprime frase do menor
    syscall
    j fim
```

```
IGUAL: li $v0, 4
    la $a0, igual # imprime frase do igual
    syscall
```

```
fim:
```

Senice	Code in \$v0	Arguments	Result
print integer	1	\$a0 = integer to print	
print flat	2	gft2 = flat to print	
print double	3	gft2 = double to print	
print string	4	\$a0= address of null-terminated string to print	
ixdmeg< x	5		\$v0 contains integer read
eadQoW	6		\$f0 contains Ooat read
ixddouble	7		\$f0 contains double read
read string	8	\$a0 = address of input buffer \$a1 = maximum number of characters to read	M role hcz/ou- /ofJe
*%1 « P	9	- number of bytes to allocate	\$v0 contains address of allocated memory
exit execubon)	10		