

[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.

Compiladores são usados para gerar código objeto a partir do código fonte, e não o programa executável a partir do código objeto. A etapa de gerar o programa executável geralmente envolve um link-edição(ligador) que liga o código objeto com bibliotecas para criar o executável.

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

A compilação cruzada é realizada para gerar código executável para uma máquina alvo diferente.

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

A diferença envolve uma etapa de ligação (link-edição) entre o código fonte e o código objeto na qual utiliza bibliotecas para gerar o programa executavel.

(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.

Mesmo que os computadores sejam idênticos, se os sistemas operacionais são diferentes, as chamadas de sistema e as bibliotecas usadas podem ser diferentes. Isso resultará em programas executáveis diferentes.

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 Endian.

MEMÓRIA DE DADOS			
Endereço	Dado		Endereço Dado
0x10010000			0x10010008 0x00
0x10010001			0x10010009 0x00
0x10010002			0x1001000A 0x00
0x10010003			0x1001000B 0x15
0x10010004	0x00		0x1001000C 0xC
0x10010005	0x0D		0x1001000D 0xA
0x10010006	0x00		0x1001000E 0xF
0x10010007	0x00		0x1001000F 0xE

Questão 3: Converta a seguinte instrução em assembly MIPS. Considere a = \$s0, b = \$s1, c = \$s2, d = \$s4. Tente reutilizar os registradores temporários.

<u>Instrução:</u>	<u>Correspondente em assembly MIPS:</u>
$c[5] = a[4] + d - c$	lw \$t0, 16(\$s0) // a[4] add \$t1, \$t0, \$s4 sub \$t2, \$t0, \$s2 sw \$t2, 20(\$s2)

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	0x0000E587
\$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

MEMÓRIA

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

ENDEREÇO	DADO
0x10010000	
0x10010001	
0x10010002	
0x10010003	
0x10010004	
0x10010005	
0x10010006	
0x10010007	
0x10010008	
0x10010009	0x40
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 100.

.data

F1: .asciiz "Entre com o primeiro numero"

F2: .asciiz "Entre com o segundo numero"

F3: .asciiz "Entre com o terceiro numero"

F_maior: .asciiz "\nA soma é maior que 100"

F_menor: .asciiz "\nA soma é menor que 100" F_igual:

.asciiz "\nA soma é igual a 100"

.text

Imprimir a frase

la \$a0, F1 li \$v0,

4 syscall # Entrar

com o int li \$v0, 5

syscall

add \$t1, \$v0, \$0

Imprimir a frase

la \$a0, F2 li \$v0,

4 syscall # Entrar

com o int li \$v0, 5

syscall

add \$t2, \$v0, \$0

Imprimir a frase

la \$a0, F3 li \$v0,

4 syscall # Entrar

com o int li \$v0, 5

```
syscall add $t3,
$v0, $0
```

```
### PROCESSAMENTO ###
```

```
add $t1, $t1, $t2
add $t1, $t1, $t3
```

```
blt $t1, 100, menor
bgt $t1, 100, maior
# se for igual #
Imprimir a frase la
$a0, F_igual li
$v0, 4 syscall
```

```
menor: #
Imprimir a frase
la $a0, F_menor li
$v0, 4 syscall j
fim
```

```
maior: # Imprimir
a frase la $a0,
F_maior li $v0, 4
syscall j fim
```

```
fim: li $v0, 10
syscall
```

Senice	Code in \$v0	Arguments	Result
print integer	1	\$a0 = integer to print	
print flat	2	gf2 = flat to print	
print double	3	gf2 = double to print	
print string	4	\$a0 = address of null-terminated string to print	
ixdmegx	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
h * p «	9	- number of bytes to allocate	\$v0 contains address of allocated memory
exit execubon)	10		