NP1 antiga – C208

Questão 1 (20 pontos): Analise as afirmativas a seguir e as classifique como 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

Os compiladores fazem a tradução de um programa em linguagem de alto nível para um código de baixo nível.

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

A compilação cruzada também pode ser usada para sistemas operacionais diferentes.

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

Eles se diferem pela etapa do ligador.

(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.

Cada sistema operacional possui dependências específicas para que o executável funcione. O ligador é a etapa onde essas dependências são adicionadas ao programa.

Questão 2 (20 pontos): 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	00
0x10010001		0x10010009	00
0x10010002		0x1001000A	C
0x10010003		0x1001000B	A
0x10010004	13	0x1001000C	F
0x10010005	00	0x1001000D	E
0x10010006	15	0x1001000E	
0x10010007	00	0x1001000F	

Questão 3 (20 pontos): 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 1 2	add $t0, 16($s0), $s4 sub 20($s2), $t0, $s2

Questão 4 (20 pontos): 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	0x00E5
$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	
$t4	
$t5	
$t6	
$t7	
$s0	
$s1	
$s2	
$s3	

Memória

Endereço	Dado
0x10010000	
0x10010001	
0x10010002	
0x10010003	
0x10010004	
0x10010005	
0x10010006	
0x10010007	
0x10010008	0x40
0x10010009	
0x1001000A	
0x1001000B	

Questão 5 (20 pontos): 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
msg1: .asciiz "Valor 1:"
msg2: .asciiz "Valor 2:"
msg3: .asciiz "Valor 3:"
msg4: .asciiz "Soma maior que 100"
msg5: .asciiz "Soma menor que 100"
msg6: .asciiz "Soma igual a 100"
.text
la $a0, msg1
li $v0, 4
syscall
li $v0, 5
add $t0, $v0, $0
la $a0, msg2
li $v0, 4
syscall
li $v0, 5
add $t1, $v0, $0
la $a0, msg3
li $v0, 4
syscall
li $v0, 5
add $t2, $v0, $0
add $t3, $t2, $t1
add $t3, $t3, $t0
blt $t3, 100, menor
bgt $t3, 100, maior
la $a0, msg6
li $v0, 4
syscall
j fim
menor:
la $a0, msg5
li $v0, 4
syscall
j fim
maior:
la $a0, 5
li $v0, 4
syscall
j fim
fim:
```