

Suponha que os valores hexadecimais 0x00000014, 0x0000000A e 0x0000000F correspondem a \$a0, \$a1 e \$a2, respectivamente. Qual é o valor decimal de \$s0 para o seguinte código assembly do MIPS?

```
addi $t0, $zero, 5
add $t0, $t0, $t0
sub $sp, $sp, 4
sw $t0, 0($sp)
jal Proc
lw $t0, 0($sp)
add $sp, $sp, 4
add $s0, $t0, $v0

Proc: add $t0, $a2, $a0
sub $v0, $t0, $a1
jr $ra
```

- Escolha uma opção:
- ☐ a. 20
 - ☒ b. 35
 - ☐ c. 25
 - ☐ d. 30
 - ☐ e. 15

main:
t0 = 0+5
t0 = 5+5 //10
sw: t0=10 -> sp

//empilhando t0 pra usar no procedimento
proc:
t0 = a2 + a0 = f+14 = 0x00000023
v0 = t0 - a1 = 0x00000019 (25 em decimal)
s0 = t0 + v0 = 0x0000003c

main:
lw: t0 <- sp=10
s0 = 10+25 = 35

Executando o código assembly do MIPS mostrado abaixo, qual é o valor binário final no registrador \$s1?

```
addi $t0, $zero, 2021
add $s1, $t0, $t0
lui $s1, 1234
```

- Escolha uma opção:
- ☒ a. 1111 1100 1010 0100 1101 0010
 - ☐ b. 0000 0100 1101 0010 0000 1111 1100 1010
 - ☐ c. 0000 1111 1100 1010 0000 0100 1101 0010
 - ☐ d. 0100 1101 0010 1111 1100 1010

t0 = 0+2021
s1 = 2021+2021
s1 = 4042
lui=1234

4042 = 1111 1100 1010
1234 = 0100 1101 0010

Qual é o tipo de endereçamento no MIPS de uma instrução "lui"?

- Escolha uma opção:
- ☒ a. endereçamento imediato e por registrador
 - ☐ b. endereçamento relativo ao PC e imediato
 - ☐ c. endereçamento pseudodireto e relativo ao PC
 - ☐ d. endereçamento por base ou deslocamento e por registrador
 - ☐ e. endereçamento por registrador e direto

Assinale a(s) alternativa(s) **incorreta(s)**.

- Escolha uma ou mais:
- ☐ a. A arquitetura MIPS é uma arquitetura RISC load-store e suas instruções contém no máximo 3 operandos.
 - ☐ b. Na arquitetura MIPS, a memória é organizada em bits e os endereços de palavras são múltiplos de 4, pois existem 4 bits em uma palavra.
 - ☐ c. Na arquitetura MIPS, a instrução "jump" é a instrução de desvio incondicional para chamada de procedimento e a instrução "jump register" é a instrução para retorno de chamada de procedimento.
 - ☐ d. Apesar de ter 3 formatos de instruções, todas as instruções são do mesmo tamanho na arquitetura MIPS.
 - ☒ e. Na linguagem assembly do MIPS, os dados sempre são guardados na memória.

Qual é o tipo de instrução das instruções beq (branch on equal), slt (set on less than) e jr (jump register), respectivamente?

- Escolha uma opção:
- ☐ a. tipo-I, tipo-R e tipo-J
 - ☐ b. tipo-R, tipo-J e tipo-R
 - ☐ c. tipo-I, tipo-I e tipo-J
 - ☒ d. tipo-I, tipo-R e tipo-R
 - ☐ e. tipo-R, tipo-J e tipo-J

Qual é a instrução em linguagem assembly descrita pelos seguintes campos MIPS?

op = 0, rs = 6, rt = 7, rd = 13, shamt = 0, funct = 34

- Escolha uma opção:
- ☐ a. sub \$a3, \$a0, \$t3
 - ☐ b. sub \$a2, \$a3, \$t5
 - ☒ c. sub \$t5, \$a2, \$a3
 - ☐ d. sub \$t3, \$a3, \$a0

op=0 funct=34 -> sub
Sub = op rs rt rd shamt funct
Rd = 13 = \$t5

Suponha que \$t0 contenha o valor hexadecimal 0x00000003. Qual é o valor de \$t2 após as instruções em assembly MIPS a seguir?

```
slt $t1, $zero, $t0
addi $t2, $zero, 1
beq $t1, $t2, L1
add $t2, $t1, $t2
j END
L1: addi $t1, $t1, 1
add $t2, $t2, $t1
END:
```

if(0<t0) t1=1;
t2=0+1;
if(t1 == t2) goto L1;

L1:
t1 = t1+1; # (t1==2)
t2 = t2+t1; # t2 == 3

- Escolha uma opção:
- ☒ a. \$t2 = 3
 - ☐ b. \$t2 = 4
 - ☐ c. \$t2 = 2
 - ☐ d. \$t2 = 7
 - ☐ e. \$t2 = 6

Qual é a representação binária da instrução lw \$t4, 20(\$s6)?

- Escolha uma opção:
- ☐ a. 1000 1101 1101 0101 0001 1000
 - ☐ b. 1000 1110 1100 1100 0001 0100
 - ☒ c. 1000 1110 1100 1100 0000 0001 0100
 - ☐ d. 1000 1101 1101 0101 0000 0000 1000

35 11 22 20
100011 1011 1011 10110 10100
0/

Qual é a instrução assembly do MIPS correspondente à seguinte representação binária?

1010 1101 0100 0011 0000 0000 0010 0100

- Escolha uma opção:
- ☒ a. sw \$t2, 36(\$v1)
 - ☐ b. sw \$s2, 44(\$v0)
 - ☐ c. sw \$v1, 36(\$t2)
 - ☐ d. sw \$v0, 44(\$s2)

35 10 3 36
sw v1 t2 36

Quanto valores podem ser retornados de uma chamada de procedimento no assembly MIPS?

- Escolha uma opção:
- ☐ a. 0
 - ☐ b. 1
 - ☒ c. 2
 - ☐ d. 3
 - ☐ e. 4

10 11