

ATIVIDADE AVALIATIVA

Instruções - 1

Estudante: Wagner Clemente Coelho Batalha 20/0044486

Estudante: Eder De Amaral Amorim 17/0140636

Obs.: a) Cole no texto de solução de cada exercício *print* de tela do simulador MARS (ou DrMIPS) com o código em assembly, tabela de registradores (abra Registers) e tabela de memória RAM (Data Segment). Use-o para buscar erros de sintaxe.

b) Introduza valores nos registradores e memória (manualmente, diretamente nas tabelas) para verificar o funcionamento do código assembly. Por padrão, há somente zero em todas as posições no início da execução.

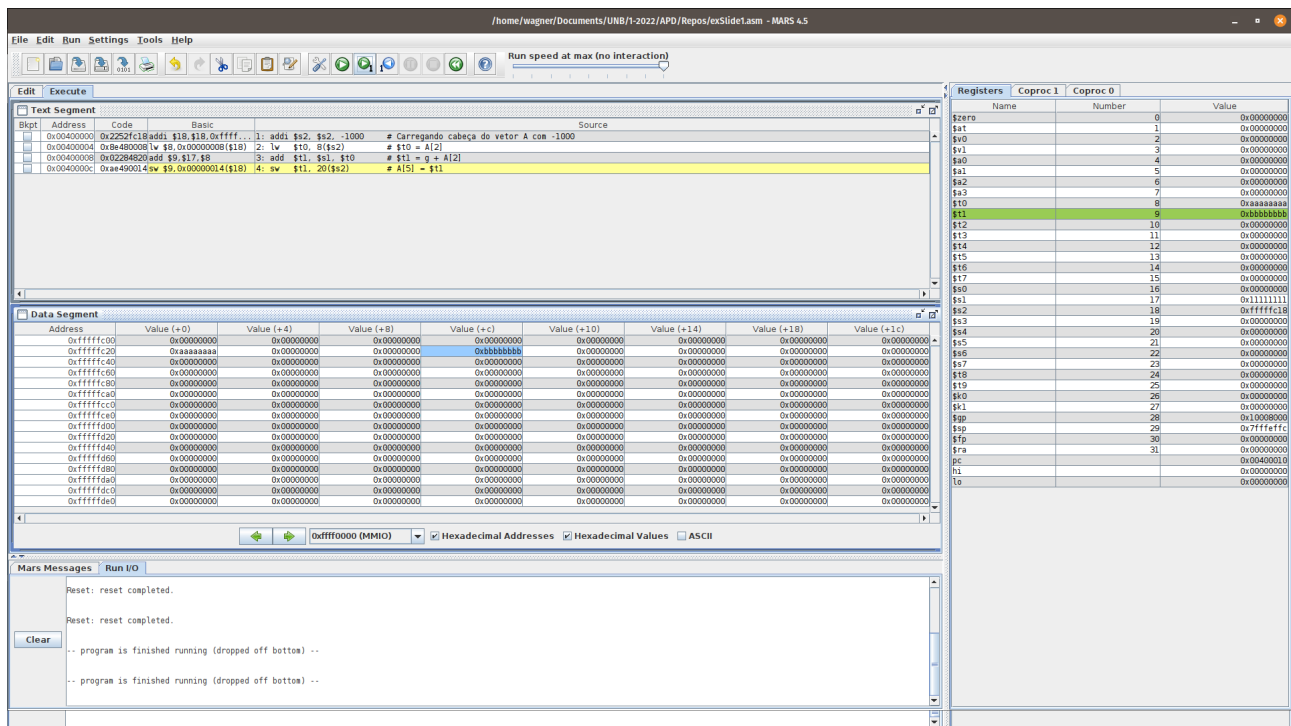
c) Para carregar (“atribuir”) valor a um registrador via código assembly, é possível usar a instrução *addi*. Ex.: *addi \$t0, \$zero, 2* (o efeito é $\$t0=2$). Isso ajuda a verificar o funcionamento do código.

d) Caso não for possível visualizar o valor na RAM (Data Segment), altere o *offset* (*load*, *store*) para um número negativo (-10, -100, -1000, -10000 etc) até ele constar na tela. Escreva um comentário (#) no código assembly para avisar que foi necessário ajustar o *offset*. Essa problemática de *range* de endereços de memória depende da versão do MARS; basta avisar esse problema em comentário no código assembly que o/a estudante não perderá pontos na correção.

1) Traduza a seguinte expressão em C para mnemônicos do *assembly* do MIPS (qual o código?):

$$A[5] = g + A[2]$$

Para isso, suponha que o endereço base do vetor **A** é dado em $\$s2$ e que a variável **g** é dada em $\$s1$.



$A[2] = 0xAAAAAAAA$
 $g = 0x11111111$
 $A[5] = 0xBBBBBBBB$

2) Traduza a seguinte expressão em C para mnemônicos do *assembly* do MIPS (qual o código?):

$$A[i+g] = g + A[i] - A[0]$$

Para isso, suponha que o endereço base do vetor **A** é dado em **\$s2** e que as variáveis **i** e **g** são dadas em **\$s0** e **\$s1**, respectivamente.

/home/wagner/Documents/UNB/1-2022/APD/Repos/ex2 - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x3c01000	lui	\$1, 0x00000001	4: addi \$s2, \$s2, 0x10010000 # Carregando cabeça do vetor A com -1000
0x00400004	0x3421000	ori	\$1, \$1, 0x00000000	
0x00400008	0x0241000	addi	\$s8, \$s1, 1	
0x0040000c	0x2210000	addi	\$s0, \$s0, 2	# i = 2
0x00400010	0x2230000	addi	\$s1, \$s1, 5	# g = 5
0x00400014	0x0210000	add	\$t0, \$s0, \$s0	# t0 = i + i
0x00400018	0x0184000	add	\$t0, \$t0, \$t0	# t0 = t0 + t0 = 2i + 2i = 4i
0x0040001c	0x0112400	add	\$s, \$s, \$s	# s = s + s
0x00400020	0x0000000	lw	\$t5, 0(\$t0)	# \$t5 = A[i]
0x00400024	0x0e40000	lw	\$t6, 0(\$s2)	# \$t6 = A[0]
0x00400028	0x01a7000	sub	\$t6, \$t5, \$t6	# \$t6 = \$t5 - \$t6 = A[i] - A[0]
0x0040002c	0x01d1700	add	\$t6, \$t6, \$s1	# \$t6 = \$t6 + g
0x00400030	0x0238700	add	\$t7, \$s1, \$s1	# \$t7 = 4 * g
0x00400034	0x01e7800	add	\$t0, \$t0, \$t7	# \$t0 = 4A[i] + 4 * g
0x00400038	0x0100400	add	\$s, \$s, \$s	

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000007	0x00000000	0x00000014	0x00000000	0x00000000	0x00000000	0x00000000	0x00000012
0x10010004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010008	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001000c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010010	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010014	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010018	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001001c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010024	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010028	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001002c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010030	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010034	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010038	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001003c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run I/O

-- program is finished running (dropped off bottom) --

-- program is finished running (dropped off bottom) --

-- program is finished running (dropped off bottom) --

-- program is finished running (dropped off bottom) --

$A[0] = 7$
 $A[2] = 14$
 $i = 2$
 $g = 5$

/home/wagner/Documents/UNB/1-2022/APD/Repos/ex2 - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

hello.asm exSlide1.asm ex2

```

1 # A = $s2
2 # i = $s0
3 # g = $s1
4 addi $s2, $s2, 0x10010000 # Carregando cabeça do vetor A com -1000
5 addi $s0, $s0, 2 # i = 2
6 addi $s1, $s1, 5 # g = 5
7
8 add $t0, $s0, $s0 # t0 = i + i
9 add $t0, $t0, $t0 # t0 = t0 + t0 = 2i + 2i = 4i
10 add $t0, $t0, $s2 # t0 = t0 + 6A
11 lw $t5, 0($t0) # $t5 = A[i]
12 lw $t6, 0($s2) # $t6 = A[0]
13 sub $t6, $t5, $t6 # $t6 = $t5 - $t6 = A[i] - A[0]
14 add $t6, $t6, $s1 # $t6 = $t6 + g
15 add $t7, $s1, $s1
16 add $t7, $t7, $t7 # $t7 = 4 * g
17 add $t0, $t0, $t7 # $t0 = 6A[i] + 4 * g
18 sw $t6, 0($t0)

```

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x1001001c
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000014
\$t6	14	0x00000012
\$t7	15	0x00000000
\$s0	16	0x00000002
\$s1	17	0x00000005
\$s2	18	0x10010000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$fp	28	0x00000000
\$sp	29	0x7ffffeffc
\$tp	30	0x00000000
\$ra	31	0x00000000
\$pc		0x00400040
\$h		0x00000000
\$lo		0x00000000

Mars Messages Run I/O

-- program is finished running (dropped off bottom) --

-- program is finished running (dropped off bottom) --

-- program is finished running (dropped off bottom) --

-- program is finished running (dropped off bottom) --