

Relatório 09 - Laboratório de Arquitetura de Computadores

Luiz Junio Veloso Dos Santos - Matricula: 624037

1 de agosto de 2019

1. Digite, compile e observe o programa a seguir:

```
1 # x mapeado em $s1
2 .text
3 .globl teste
4 teste:
5
6     addi $s1,$zero,1 # x = 1
7     lw $s1, 0($t0)
8
9 .data
10 x1: .word 15
11 x2: .word 25
12 x3: .word 13
13 x4: .word 17
```

programa9.asm

- (a) Quando o programa é carregado, em quais posições de memória os dados foram armazenados?

R: 0x10010000 0x10010004 0x10010008 0x1001000c

- (b) Complete o programa anterior de maneira a ler os valores armazenados em x1, x2, x3 e x4 em registradores (\$s1, \$s2, \$s3, \$s4)

R:

```
1 # x mapeado em $s1
2 .text
3 .globl teste
4 teste:
5     addi $s1,$zero,1 # x = 1
6
7     addi $t0,$zero,0x1001
8     sll $t0,$t0,16 # t0 = primeiro endereco de dados
9
10    lw $s1,0($t0) # s1 = mem[0]
11    lw $s2,4($t0) # s2 = mem[1]
12    lw $s3,8($t0) # s3 = mem[2]
13    lw $s4,12($t0) # s4 = mem[3]
14
15 .data
16 x1: .word 15
17 x2: .word 25
18 x3: .word 13
19 x4: .word 17
```

programa9b.asm

- (c) Crie mais uma variável denominada soma e atribua um valor inicial de -1. A variável soma deverá estar na posição seguinte a x4. Escrever um programa que leia todos os números, calcule e substitua o valor da variável soma por este valor.

R:

```
1 # x - $s1
2 # y - $s2
3 # z - $s3
4 # w - $s4
5
6 .text
7 .globl teste
8 teste:
9     addi $t0,$zero,0x1001
10    sll $t0,$t0,16 # t0 = primeiro endereço de dados
11
12    lw $s1,0($t0) # x = mem[0]
13    lw $s2,4($t0) # y = mem[1]
14    lw $s3,8($t0) # z = mem[2]
15    lw $s4,12($t0) # w = mem[3]
16
17    add $t1,$s1,$s2 # t1 = x + y
18    add $t1,$t1,$s3 # t1 = t1 + z
19    add $t1,$t1,$s4 # t1 = t1 + w
20
21    sw $t1,16($t0) # soma = t1
22
23 .data
24 x1: .word 15
25 x2: .word 25
26 x3: .word 13
27 x4: .word 17
28 soma: .word -1
```

programa9c.asm

2. Considere o seguinte programa: $y = 127x - 65z + 1$

Faça um programa que calcule o valor de y conhecendo os valores de x e z . Os valores de x e z estão armazenados na memória e, na posição imediatamente a seguir, o valor de y deverá ser escrito.

R:

```
1 # x - $s1
2 # z - $s2
3
4 .text
5 .globl teste
6 teste:
7     addi $t0,$zero,0x1001
8     sll  $t0,$t0,16 # t0 = primeiro endereço de dados
9
10    lw  $s1,0($t0) # x = mem[0]
11    lw  $s2,4($t0) # z = mem[1]
12
13    sub $t1,$s1,$s2 # t1 = x - y
14    addi $t2,$zero,100000
15    addi $t1,$t1,1 # t3 = t3 + 1
16
17    sw  $t3,8($t0) # y = t1
18
19 .data
20 x: .word 100000
21 z: .word 200000
22 y: .word 0
```

programa10.asm

3. Considere o seguinte programa: $y = x - z + 300000$

Faça um programa que calcule o valor de y conhecendo os valores de x e z. Os valores de x e z estão armazenados na memória e, na posição imediatamente a seguir, o valor de y deverá ser escrito.

R:

```
1 # x - $s1
2 # z - $s2
3
4 .text
5 .globl teste
6 teste:
7     addi $t0,$zero,0x1001
8     sll  $t0,$t0,16 # t0 = primeiro endereço de dados
9
10    lw  $s1,0($t0) # x = mem[0]
11    lw  $s2,4($t0) # z = mem[1]
12
13    sub $t1,$s1,$s2 # t1 = x - y
14
15    addi $t2,$zero,18750 # t2 = 18750
16    sll  $t2,$t2,4 # t2 = 18750 * 2^4 = 300000
17
18    add $t1,$t1,$t2 # t1 = t1 + t2
19
20    sw  $t1,8($t0) # y = t1
21
22 .data
23 x: .word 100000
24 z: .word 200000
25 y: .word 0
```

programa11.asm

4. Considere a seguinte situação:

int ***x;

onde x contem um ponteiro para um ponteiro para um ponteiro para um inteiro.

Nessa situação, considere que a posição inicial de memória contenha o inteiro em questão.

Coloque todos os outros valores em registradores, use os endereços de memória que quiser dentro do espaço de endereçamento do Mips.

Resumo do problema:

$k = \text{MEM} [\text{MEM} [\text{MEM} [x]]]$.

Crie um programa que crie a estrutura de dados acima, leia o valor de K, dobre e o reescreva conhecendo-se apenas o valor de x.

R:

```
1 .text
2 .globl teste
3 teste:
4     addi $t0,$zero,0x1001
5     sll  $t0,$t0,16 # t0 = primeiro endereco de dados
6
7     lw  $t1,12($t0) # t1 = mem[3]
8     lw  $t2,0($t1)  # t2 = mem(t1)
9     lw  $t3,0($t2)  # t3 = mem(t2)
10    lw  $t4,0($t3)  # t4 = 42 = mem(t3)
11
12    sll  $t4,$t4,1 # t4 = t4 * 2
13    sw  $t4, 0($t3) # mem[t3] = t4
14
15 .data
16 k: .word 42
17 x: .word k
18 y: .word x
19 z: .word y
```

programa12.asm

Figura 1: Programa 9

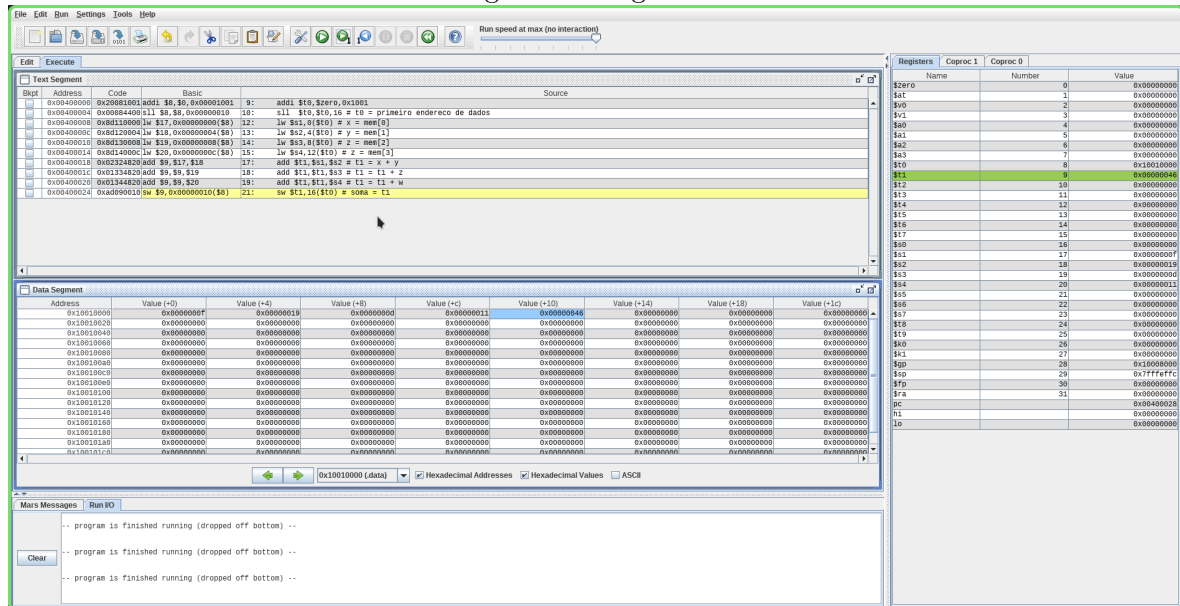


Figura 2: Programa 10

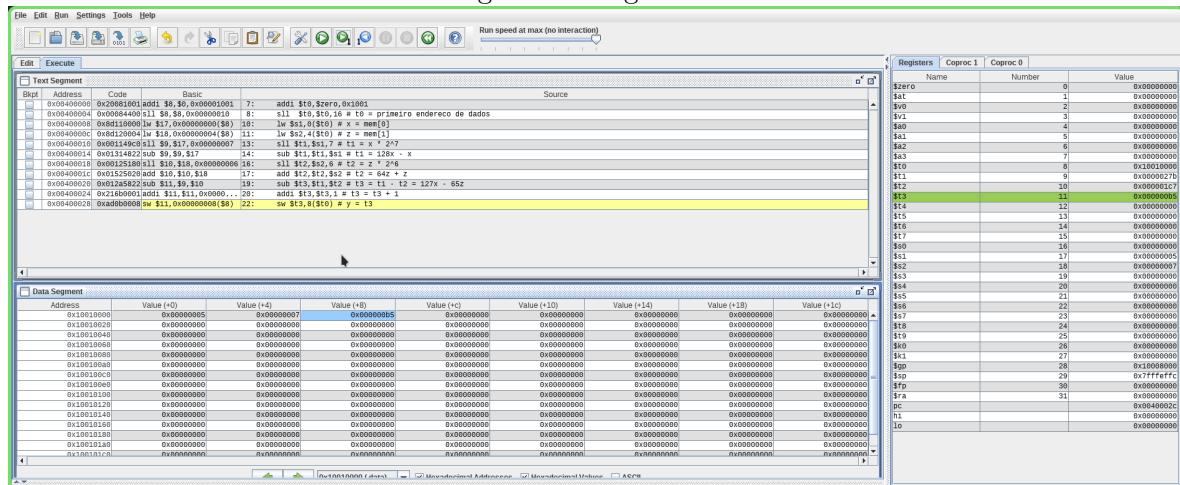


Figura 3: Programa 11

The screenshot shows a debugger interface for Program 11. The main window displays assembly code with addresses, hex codes, and comments. The right panel shows the state of registers, with \$t0, \$t1, and \$t2 highlighted. The bottom panel shows memory data segments.

Registers	Coproc 1	Coproc 0	Name	Number	Value
\$Zero				0	0
\$at				1	0
\$v0				2	0
\$v1				3	0
\$a0				4	0
\$a1				5	0
\$a2				6	0
\$a3				7	0
\$t0				8	268509997
\$t1				9	260000
\$t2				10	300000
\$t3				11	0
\$t4				12	0
\$t5				13	0
\$t6				14	0
\$t7				15	0
\$t8				16	0
\$t9				17	160000
\$s0				18	200000
\$s1				19	0
\$s2				20	0
\$s3				21	0
\$s4				22	0
\$s5				23	0
\$s6				24	0
\$s7				25	0
\$s8				26	0
\$s9				27	0
\$t0				28	25848224
\$t1				29	214711944
\$t2				30	0
\$t3				31	0
\$t4				32	4184346
\$t5				33	0
\$t6				34	0
\$t7				35	0
\$t8				36	0
\$t9				37	0
\$s0				38	0
\$s1				39	0
\$s2				40	0
\$s3				41	0
\$s4				42	0
\$s5				43	0
\$s6				44	0
\$s7				45	0
\$s8				46	0
\$s9				47	0
\$t0				48	0
\$t1				49	0
\$t2				50	0
\$t3				51	0
\$t4				52	0
\$t5				53	0
\$t6				54	0
\$t7				55	0
\$t8				56	0
\$t9				57	0
\$s0				58	0
\$s1				59	0
\$s2				60	0
\$s3				61	0
\$s4				62	0
\$s5				63	0
\$s6				64	0
\$s7				65	0
\$s8				66	0
\$s9				67	0
\$t0				68	0
\$t1				69	0
\$t2				70	0
\$t3				71	0
\$t4				72	0
\$t5				73	0
\$t6				74	0
\$t7				75	0
\$t8				76	0
\$t9				77	0
\$s0				78	0
\$s1				79	0
\$s2				80	0
\$s3				81	0
\$s4				82	0
\$s5				83	0
\$s6				84	0
\$s7				85	0
\$s8				86	0
\$s9				87	0
\$t0				88	0
\$t1				89	0
\$t2				90	0
\$t3				91	0
\$t4				92	0
\$t5				93	0
\$t6				94	0
\$t7				95	0
\$t8				96	0
\$t9				97	0
\$s0				98	0
\$s1				99	0
\$s2				100	0
\$s3				101	0
\$s4				102	0
\$s5				103	0
\$s6				104	0
\$s7				105	0
\$s8				106	0
\$s9				107	0
\$t0				108	0
\$t1				109	0
\$t2				110	0
\$t3				111	0
\$t4				112	0
\$t5				113	0
\$t6				114	0
\$t7				115	0
\$t8				116	0
\$t9				117	0
\$s0				118	0
\$s1				119	0
\$s2				120	0
\$s3				121	0
\$s4				122	0
\$s5				123	0
\$s6				124	0
\$s7				125	0
\$s8				126	0
\$s9				127	0
\$t0				128	0
\$t1				129	0
\$t2				130	0
\$t3				131	0
\$t4				132	0
\$t5				133	0
\$t6				134	0
\$t7				135	0
\$t8				136	0
\$t9				137	0
\$s0				138	0
\$s1				139	0
\$s2				140	0
\$s3				141	0
\$s4				142	0
\$s5				143	0
\$s6				144	0
\$s7				145	0
\$s8				146	0
\$s9				147	0
\$t0				148	0
\$t1				149	0
\$t2				150	0
\$t3				151	0
\$t4				152	0
\$t5				153	0
\$t6				154	0
\$t7				155	0
\$t8				156	0
\$t9				157	0
\$s0				158	0
\$s1				159	0
\$s2				160	0
\$s3				161	0
\$s4				162	0
\$s5				163	0
\$s6				164	0
\$s7				165	0
\$s8				166	0
\$s9				167	0
\$t0				168	0
\$t1				169	0
\$t2				170	0
\$t3				171	0
\$t4				172	0
\$t5				173	0
\$t6				174	0
\$t7				175	0
\$t8				176	0
\$t9				177	0
\$s0				178	0
\$s1				179	0
\$s2				180	0
\$s3				181	0
\$s4				182	0
\$s5				183	0
\$s6				184	0
\$s7				185	0
\$s8				186	0
\$s9				187	0
\$t0				188	0
\$t1				189	0
\$t2				190	0
\$t3				191	0
\$t4				192	0
\$t5				193	0
\$t6				194	0
\$t7				195	0
\$t8				196	0
\$t9				197	0
\$s0				198	0
\$s1				199	0
\$s2				200	0
\$s3				201	0
\$s4				202	0
\$s5				203	0
\$s6				204	0
\$s7				205	0
\$s8				206	0
\$s9				207	0
\$t0				208	0
\$t1				209	0
\$t2				210	0
\$t3				211	0
\$t4				212	0
\$t5				213	0
\$t6				214	0
\$t7				215	0
\$t8				216	0
\$t9				217	0
\$s0				218	0
\$s1				219	0
\$s2				220	0
\$s3				221	0
\$s4				222	0
\$s5				223	0
\$s6				224	0
\$s7				225	0
\$s8				226	0
\$s9				227	0
\$t0				228	0
\$t1				229	0
\$t2				230	0
\$t3				231	0
\$t4				232	0
\$t5				233	0
\$t6				234	0
\$t7				235	0
\$t8				236	0
\$t9				237	0
\$s0				238	0
\$s1				239	0
\$s2				240	0
\$s3				241	0
\$s4				242	0
\$s5				243	0
\$s6				244	0
\$s7				245	0
\$s8				246	0
\$s9				247	0
\$t0				248	0
\$t1				249	0
\$t2				250	0
\$t3				251	0
\$t4				252	0
\$t5				253	0
\$t6				254	0
\$t7				255	0
\$t8				256	0
\$t9				257	0
\$s0				258	0
\$s1				259	0
\$s2				260	0
\$s3				261	0
\$s4				262	0
\$s5				263	0
\$s6				264	0
\$s7				265	0
\$s8				266	0
\$s9				267	0
\$t0				268	0
\$t1				269	0
\$t2				270	0
\$t3				271	0
\$t4				272	0
\$t5				273	0
\$t6				274	0
\$t7				275	0
\$t8				276	0
\$t9				277	0
\$s0				278	0
\$s1				279	0
\$s2				280	0
\$s3				281	0
\$s4				282	0
\$s5				283	0
\$s6				284	0
\$s7				285	0
\$s8				286	0
\$s9				287	0
\$t0				288	0
\$t1				289	0
\$t2				290	0
\$t3				291	0
\$t4				292	0
\$t5				293	0
\$t6				294	0
\$t7				295	0
\$t8				296	0
\$t9				297	0
\$s0				298	0
\$s1				299	0
\$s2				300	0
\$s3				301	0
\$s4				302	0
\$s5				303	0
\$s6				304	0
\$s7				305	0
\$s8				306	0
\$s9				307	0
\$t0				308	0
\$t1				309	0
\$t2				310	0
\$t3				311	0
\$t4				312	0
\$t5				313	0
\$t6				314	0
\$t7				315	0
\$t8				316	0
\$t9				317	0
\$s0				318	0
\$s1				319	0
\$s2		</			