

UNIVERSIDADE FEDERAL DE SANTA CATARINA CAMPUS FLORIANÓPOLIS INE-DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA INE5411 - ORGANIZAÇÃO DE COMPUTADORES I

PEDRO TAGLIALENHA (22203674) VITOR PRAXEDES CALEGARI (22200379)

RELATÓRIO LABORATÓRIO 5

FLORIANÓPOLIS 2023

1 INTRODUÇÃO	3
2 CÓDIGOS DESENVOLVIDOS	4
2.1 ANÁLISE DO CÓDIGO 1	6
2.2 ANÁLISE DO CÓDIGO 2	8
3 SAÍDAS DOS CÓDIGOS NO MARS	9
4 CONCLUSÃO	10

1 INTRODUÇÃO

Este relatório tem como objetivo apresentar a análise e implementação de dois programas desenvolvidos como parte do laboratório 5 da disciplina de Organização de Computadores. Os programas em questão têm relevância na avaliação do desempenho em memória cache e consistem em tarefas que percorrem uma matriz de 16 por 16 elementos, atribuindo valores de 0 a 255 de maneiras distintas.

O primeiro programa aborda a tarefa de preencher a matriz linha a linha, enquanto o segundo programa foca na atribuição dos valores coluna a coluna. Ambos os programas foram desenvolvidos no ambiente MARS.

O relatório apresentará os códigos-fonte dos programas, suas análises e conclusões.

2 CÓDIGOS DESENVOLVIDOS

Figura 1 - Código da questão 1

```
row: .word 0 col: .word 0
    $s0, row
     $s1, col
   $s2, mat
lw $s3, 0($s0)
    $s4, 0($s1)
addi $s2, $s2, 20
la $s6, value
lw $s7, 0($s6)
    $s3, 16, exit
add $s4, $zero, $zero
sw $s4, 0($s1)
      coluna
addi $s3, $s3, 1
   $s3, 0($s0)
     linha
    $v0, 10
     $s4, 16, continua
     $ra
mul $t0, $s3, 64
mul $t1, $s4, 4
add $t0, $t0, $t1
add $t0, $t0, $s2
     $s7, 0($s6)
     $s7, 0($t0)
addi $s7, $s7, 1
addi $s4, $s4, 1
sw $s4, 0($s1)
     coluna
```

Fonte: Elaborado por autores (2023)

Figura 2 - Código da questão 2

```
mat: .word
   $s0, row
la
    $s1, col
la
   $s2, mat
lw $s3, 0($s0)
lw $s4, 0($s1)
addi $s2, $s2, 20
la $s6, value
    $s7, 0($s6)
beq $s3, 16, exit
add $s4, $zero, $zero
sw $s4, 0($s1)
jal
      coluna
addi $s3, $s3, 1
sw $s3, 0($s0)
     linha
li $v0, 10
syscall
bne
    $s4, 16, continua
    $t0, $s3, 4
mul $t1, $s4, 64
add $t0, $t0, $t1
add $t0, $t0, $s2
    $s7, 0($s6)
     $s7, 0($t0)
addi $s7, $s7, 1
addi $s4, $s4, 1
     $s4, 0($s1)
     coluna
```

Fonte: Elaborado por autores (2023)

2.1 ANÁLISE DO CÓDIGO 1

O código desenvolvido pode ser quebrado nos seguintes principais pontos:

1. Alternância entre Linha e Coluna:

 O programa possui dois loops principais, "linha" e "coluna," que alternam entre o preenchimento linha a linha e coluna a coluna da matriz.

2. Multiplicações para Cálculo de Deslocamento:

 Dentro das funções "coluna" e "continua," são realizadas operações de multiplicação para calcular o deslocamento na matriz. Isso é necessário porque os elementos da matriz não estão armazenados em uma estrutura bidimensional, mas sim em uma área de memória contígua.

3. Função "linha":

 A função "linha" é responsável por preencher os elementos da matriz em uma linha. Ela começa com a coordenada da coluna (\$s4) definida como zero e, em seguida, chama a função "coluna" para preencher a linha atual. Após isso, incrementa a coordenada da linha e repete o processo até preencher todas as linhas.

4. Função "coluna":

A função "coluna" preenche os elementos da matriz em uma coluna. Ela começa com a coordenada da coluna (\$s4) definida como zero e, em seguida, entra em um loop. Dentro desse loop, o código calcula o deslocamento no endereço da matriz com base nas coordenadas da linha e coluna, realiza a atribuição do valor (0 a 255) e, em seguida, incrementa a coordenada da coluna e o valor a ser atribuído (\$s7). O loop continua até que a coordenada da coluna seja igual a 16.

5. Função "continua":

 A função "continua" é chamada a partir de "coluna" e lida com a tarefa de calcular o deslocamento no endereço da matriz. Isso envolve multiplicar as coordenadas da linha e coluna pelos valores apropriados, adicionar o deslocamento ao endereço base da matriz e realizar as operações de armazenamento. Esses loops aninhados garantem que a matriz seja preenchida linha por linha, pois o loop externo controla a iteração pelas linhas, enquanto o loop interno preenche os elementos da coluna dentro de cada linha. No final do loop externo, todas as linhas da matriz terão sido preenchidas com os valores de 0 a 255.

2.2 ANÁLISE DO CÓDIGO 2

O Código 2 é semelhante ao Código 1, pois ambos visam preencher uma matriz 16x16 com valores de 0 a 255 alternando entre preenchimento por linha e coluna. No entanto, existe algumas diferenças:

1. Função "coluna" e "continua":

- No Código 2, as funções "coluna" e "continua" foram modificadas em relação ao Código 1.
- A função "continua" calcula o deslocamento de maneira diferente: multiplica \$s3 (linha) por 4 e \$s4 (coluna) por 64 para obter o deslocamento apropriado no endereço da matriz.
- A ordem das multiplicações e a estratégia de cálculo são invertidas em relação ao Código 1.

2. Diferenças na Ordem de Preenchimento:

Enquanto o Código 1 preenche a matriz linha a linha, o Código 2 preenche coluna a coluna. Isso significa que, no Código 2, a coordenada da coluna é reiniciada a cada iteração do loop "linha," enquanto a coordenada da linha é reiniciada no Código 1 para cada iteração do loop "coluna."

3. Multiplicação para Cálculo do Deslocamento:

 O uso da multiplicação no Código 2 para calcular o deslocamento é diferente da estratégia do Código 1. O Código 2 multiplica a coordenada da linha por 4 e a coordenada da coluna por 64 para calcular o deslocamento, enquanto o Código 1 utiliza uma abordagem inversa.

O Código 2 é semelhante ao Código 1 em termos de objetivo, mas há diferenças nas estratégias de cálculo de deslocamento e na ordem de preenchimento da matriz.

3 SAÍDAS DOS CÓDIGOS NO MARS

Figura 3 - Saída do código 1

Text Segment										
Bkpt Address	Code	Basic						Source		
0x00400000	0x3c011001 lu	i \$1,4097	9:	la	\$s0, row					
0x00400004	0x34300000 or	i \$16,\$1,0								
0x00400008	0x3c011001 lu	i \$1,4097	10:	la	\$sl, col					
0x0040000c	0x34310004 or	i \$17,\$1,4								
0x00400010	0x3c011001 lu	i \$1,4097	11:	la	\$s2, mat					
0x00400014	0x3432000c or	1 \$18,\$1,12								
	0x8e130000 lw		13:	lw	\$83, 0(\$80)					
	0x8e340000 lw		14:	lw	\$84, 0(\$81)					
	0x22520014 ad		16:	addi	\$82, \$82, 20					
	0x3c011001 lu		18:	la	\$s6, value					
	0x34360008 or									
	0x8ed70000 lw		19:	lw	\$87, O(\$86)					
	0x20010010 ad		23:	beq	\$83, 16, exit					
	0x10330006 be									
	0x0000a020 ad		25:	add	\$s4, \$zero, \$zero					
	0xae340000 sw		26:	sw	\$s4, 0(\$s1)					
	0x0c100016 ja		28:	jal	coluna					
	0x22730001 ad		30:	addi	\$s3, \$s3, l					
	0xae130000 sw		31:	sw	\$s3, 0(\$s0)					
	0x0810000cj		33:	j	linha					
	0x2402000a ad		35:	li	\$v0, 10					
0X00400054	0x0000000c sy	SCAII	36:	sysca.						
Data Segment										
Addres		Value (+0)			Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)
	0x10010000		16		16	255		0	0	0
	0x10010020		0		1	2	3	4	5	6
	0x10010040		8		9	10		12	13	14
	0x10010060		16		17	18		20	21	22
	0x10010080		24		25	26		28	29	30
	0x100100a0 0x100100c0		32		33	34		36	37	38
			40		41	42		44	45	46
	0x100100e0		48		49	50	51	52	53	54
	0x100100e0 0x10010100		56		57	58	59	60	61	62
	0x100100e0 0x10010100 0x10010120		56 64		57 65	58 66	59 67	60 68	61 69	62 70
	0x100100e0 0x10010100 0x10010120 0x10010140		56 64 72		57 65 73	58 66 74	59 67 75	60 68 76	61 69 77	62 70 78
	0x100100e0 0x10010100 0x10010120 0x10010140 0x10010160		56 64 72 80		57 65 73 81	58 66 74 82	59 67 75 83	60 68 76 84	61 69 77 85	62 70 78 86
	0x100100e0 0x10010100 0x10010120 0x10010140 0x10010160 0x10010180		56 64 72 80 88		57 65 73 81	58 66 74 82 90	59 67 75 83 91	60 68 76 84 92	61 69 77 85 93	62 70 78 86 94
	0x100100e0 0x10010100 0x10010120 0x10010140 0x10010160 0x10010180 0x100101a0		56 64 72 80 88 96		57 65 73 81 89 97	58 66 74 82 90 98	59 67 75 83 91 99	60 68 76 84 92 100	61 69 77 85 93	62 70 78 96 94 102
	0x100100e0 0x10010100 0x10010120 0x10010140 0x10010160 0x10010180		56 64 72 80 88		57 65 73 81	58 66 74 82 90	59 67 75 83 91 99	60 68 76 84 92	61 69 77 85 93	62 70 78 86 94

Fonte: Elaborado por autores (2023)

Figura 4 - Saída do código 2

ct Segment										
Address	Code	Basic						Source		
	0x3c011001 lu		9:	la	\$s0, row			000100		
	0x34300000 or			24	7007 2011					
	0x3c011001 lu		10:	la	\$sl, col					
	0x34310004 or				,,					
	0x3c011001 lu		11:	la	\$s2, mat					
	0x3432000c or									
0x00400018	0x8e130000 lw	\$19,0(\$16)	13:	lw	\$s3, 0(\$s0)					
0x0040001c	0x8e340000 lw	\$20,0(\$17)	14:	1w	\$84, 0(\$81)					
		di \$18,\$18,20	16:	addi	\$82, \$82, 20					
	0x3c011001 lu		18:	la	\$s6, value					<u> </u>
	0x34360008 or									
	0x8ed70000 lw		19:	1w	\$87, D(\$86)					
	0x20010010 ad		23:	beq	\$s3, 16, exit					
	0x10330006 be									
	0x0000a020 ad		25:	add	\$s4, \$zero, \$zero					
	0xae340000 sw		26:	SW	\$s4, 0(\$s1)					
0x00400040II	0x0c100016 ja	1 0x00400058	28:	jal	coluna					
0x00400044	0x22730001 ad		30:	addi	\$83, \$83, 1					
0x00400044 0x00400048	0xae130000 sw	\$19,0(\$16)	31:	SW	\$s3, 0(\$s0)					
0x00400044 0x00400048 0x0040004c	0xae130000 sw 0x0810000c j	\$19,0(\$16) 0x00400030	31:	sw j	\$s3, 0(\$s0) linha					
0x00400044 0x00400048 0x0040004c 0x00400050	0xae130000 sw 0x0810000c j 0x2402000a ad	\$19,0(\$16) 0x00400030 diu \$2,\$0,10	31: 33: 35:	sw j li	\$s3, 0(\$s0) linha \$v0, 10					
0x00400044 0x00400048 0x0040004c 0x00400050	0xae130000 sw 0x0810000c j	\$19,0(\$16) 0x00400030 diu \$2,\$0,10	31:	sw j	\$s3, 0(\$s0) linha \$v0, 10					
0x00400048 0x0040004c 0x00400050 0x00400054	0xae130000 sw 0x0810000c j 0x2402000a ad	\$19,0(\$16) 0x00400030 diu \$2,\$0,10	31: 33: 35:	sw j li	\$s3, 0(\$s0) linha \$v0, 10					
0x00400048 0x0040004c 0x00400050 0x00400054	0xae130000 sw 0x0810000c j 0x2402000a ad 0x0000000c sy	\$19,0(\$16) 0x00400030 diu \$2,\$0,10	31: 33: 35: 36:	sw j li	\$83, 0(\$80) linha \$v0, 10 ll	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 Segment	0xae130000 sw 0x0810000c j 0x2402000a ad 0x0000000c sy	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36:	sw j li	\$3, 0(\$0) linha \$v0, 10 .1 Value(+4)	255	0	0	0	0
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 Segment	0xae130000 sw 0x0810000c j 0x2402000a ad 0x0000000c sy	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36:	sw j li	\$83, 0(\$80) linha \$90, 10 ll Value(+4) 16	255 32	0 48	0 64	0 80	0 96
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 Segment	0xae130000 sw 0x0810000cj 0x2402000a ad 0x0000000c sy	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 36:	sw j li	683, 0(680) linha 590, 10 1 Value (+4) 16 16 16 144	255 32 160	0 48 176	0 64 192	0 80 208	0 96 224
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 Segment	0xae130000 sw 0x0810000c j 0x2402000a ad 0x0000000c sy 8 0x10010000 0x10010020 0x10010040 0x10010060	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128	sw j li	\$30, 0(\$30) 11nha \$50, 10 11 Value (+4) 16 16 144 17	255 32 160 33	0 48 176 49	0 64 192 65	0 80 208 81	0 96 224 97
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 Segment	0xae130000 sw 0x0810000c; 0x2402000a ad 0x0000000c sy 0x10010000 0x10010000 0x10010000 0x10010000 0x10010000	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128 1	sw j li	### ##################################	255 32 160 33 161	0 48 176 49 177	0 64 192 65 193	0 80 208 81 209	0 96 224 97 225
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 Segment	0xae130000 sw 0x8810000c j 0x2402000a do 0x0000000c sy ss 0x10010000 0x10010020 0x10010040 0x10010060 0x10010080 0x10010080	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128 1 129 2	sw j li	value (+4) Value (+4) Value (+4) 16 144 17 145 18	255 32 160 33 161 34	0 48 176 49 177 50	0 64 192 65 193 66	0 80 208 81 209 82	0 96 224 97 225 98
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 a Segment	0xae130000 sw 0x8810000cj ox2402000a ad 0x00000000 sy 0x10010000 0x10010000 0x10010000 0x10010000 0x10010000 0x10010000	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128 1 129 2 130	sw j li	### ##################################	255 32 160 33 161 34 162	0 48 176 49 177 50	0 64 192 65 193 66 194	0 80 208 81 209 82 210	0 96 224 97 225 98
0x00400044 0x0040004c 0x0040004c 0x00400050 0x00400054 a Segment	0xae130000 sw 0x0801000c c c c c c c c c c c c c c c c c c	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128 1 129 2 130 3	sw j li	value (+4) Value (+4) Value (+4) 16 144 17 145 16 146	255 32 160 33 161 34 162 35	0 48 176 49 177 50 178	0 64 192 65 193 66 194 67	0 80 208 81 209 82 210	0 96 224 97 225 98 226
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 a Segment	0xxe130000 st 0x2402000a ad 0x2402000a ad 0x0000000c sy 0x10010000 0x10010100 0x1001000 0x10010000 0x1001000 0x1001000 0x1001000 0x1001000 0x1001000 0x10010000 0x10010000 0x10010000 0x10010000 0x10010000 0x10010000 0x1001000000 0x100100000 0x100100000 0x100000000	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128 1 129 2 130 3 131	sw j li	### ##################################	255 32 160 33 161 34 162 35 163	0 48 176 49 177 50 178 51	0 64 192 65 193 66 194 67	0 80 208 81 209 82 210 83 211	0 96 224 97 225 98 226 99
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 a Segment	0xae130000 sw 0x081000c2 0x2402000 ad 0x000000c sy 0x10010000 0x100101000 0x100101000 0x100101000 0x10010100 0x1001000 0x10010000 0x10010000 0x10010000 0x10010000 0x100100000 0x10000000 0x100100000000	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128 1 129 2 130 3 131 4	sw j li	value (+4) Value (+4) Value (+4) 16 144 17 145 18 146 19 147	255 32 160 33 161 34 162 35 163	0 48 176 49 177 50 178 51 179 52	0 64 192 65 193 66 194 67 195	0 80 208 81 209 82 210 83 211 84	0 96 224 97 225 98 226 59 227
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 a Segment	0xee130000 st 0x202000a ad 0x0010000 st 0x202000a ad 0x0000000c st 0x10010000 ox10010000 ox1001001000 ox100101000 ox10010000 ox100100000 ox100000000 ox10000000000	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128 1 129 2 130 3 131 4	sw j li	\$\(\frac{\pmatrix}{3}, 0(\pmatrix) \) \[\text{Value} (+4) \] \[\text{Value} (+4) \] \[\text{16} \] \[\text{144} \] \[\text{17} \] \[\text{145} \] \[\text{16} \] \[\text{146} \] \[\text{19} \] \[\text{147} \] \[\text{20} \] \[\text{141} \]	255 32 160 33 161 34 162 35 163 36 164	0 48 176 49 177 50 178 51 179 52 180	0 64 192 65 193 66 194 67 195 68	0 80 208 91 209 82 210 83 211 84 212	0 96 224 97 225 98 226 99 227 100 228
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054 a Segment	0xae130000 avae130000 cox4020000a ad 0x00000000 cy SS 0x10010000 cox10010000 cox10010100 cox1001000 cox10010000 cox1001000 cox1001000 cox1001000 cox10000 cox1001000 cox10010000 cox1001000 cox100000 cox100000 cox1000000 cox100000 cox1000000	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128 1 1 129 2 2 130 3 3 131 14 132 5	sw j li	value (+4) Value (+4) Value (+4) 16 144 17 145 19 147 20 148 21	255 32 160 33 161 34 162 25 15 163 163 36 164	0 48 176 49 177 50 178 51 179 52 180	0 64 192 65 193 66 194 67 195 68 196	0 80 208 81 209 82 210 83 211 84 212 85	0 96 224 97 225 98 226 99 227 100 228
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054	0Xxe130000 st 0Xx2602000a ad 0Xx00000000c sy 0Xx100000000 ct 0Xx10010000 ct 0Xx100101000 ct 0Xx10010100 ct 0Xx100101000 ct 0Xx10010100 ct 0Xx10010100 ct 0Xx10010100 ct 0Xx10010100 ct 0Xx10010100 ct 0Xx1001000 ct 0Xx1001000 ct 0Xx1001000 ct 0Xx10010100 ct 0Xx1001000 ct 0Xx10010000 ct 0Xx1001000 ct 0Xx10010000 ct 0Xx10010000 ct 0Xx100000 ct 0Xx1000000 ct 0Xx10000000 ct	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128 1 129 2 2 130 3 131 4 132 5	sw j li	\$\(\frac{\pmatrix}{3}, 0(\pmatrix) \) \[\text{Value} (+4) \] \[\text{Value} (+4) \] \[\text{16} \] \[\text{144} \] \[\text{17} \] \[\text{145} \] \[\text{10} \] \[\text{146} \] \[\text{19} \] \[\text{147} \] \[\text{20} \] \[\text{148} \] \[\text{21} \] \[\text{149} \]	255 222 160 33 161 34 162 35 163 36 164 37 165	0 48 176 49 177 50 178 51 179 52 180 53	0 64 192 65 193 66 194 67 195 68 196 69	0 80 208 81 209 82 210 83 211 84 212 85	0 96 224 97 225 99 226 99 227 100 228 101 229
0x00400044 0x00400048 0x0040004c 0x00400050 0x00400054	0xae130000 avae130000 cox4020000a ad 0x00000000 cy SS 0x10010000 cox10010000 cox10010100 cox1001000 cox10010000 cox1001000 cox1001000 cox1001000 cox10000 cox1001000 cox10010000 cox1001000 cox100000 cox100000 cox1000000 cox100000 cox1000000	\$19,0(\$16) 0x00400030 diu \$2,\$0,10 scall	31: 33: 35: 36: 16 0 128 1 1 129 2 2 130 3 3 131 14 132 5	sw j li	value (+4) Value (+4) Value (+4) 16 144 17 145 19 147 20 148 21	255 32 160 33 161 34 162 25 15 163 163 36 164	0 48 176 49 177 50 178 51 179 52 180	0 64 192 65 193 66 194 67 195 68 196	0 80 208 81 209 82 210 83 211 84 212 85	0 96 224 97 225 98 226 99 227 100 228

Fonte : Elaborado por autores (2023)

4 CONCLUSÃO

Em resumo, os códigos desenvolvidos para o laboratório 5 atingiram com êxito os objetivos estabelecidos. Ambos os programas foram implementados de acordo com as especificações das questões, preenchendo uma matriz 16x16 com valores de 0 a 255, alternando entre preenchimento por linha e por coluna.

Agora, estamos prontos para avançar para o laboratório 6, onde analisaremos o desempenho desses códigos em relação à memória cache, buscando compreender como o acesso sequencial à memória afeta o desempenho do sistema de computadores.