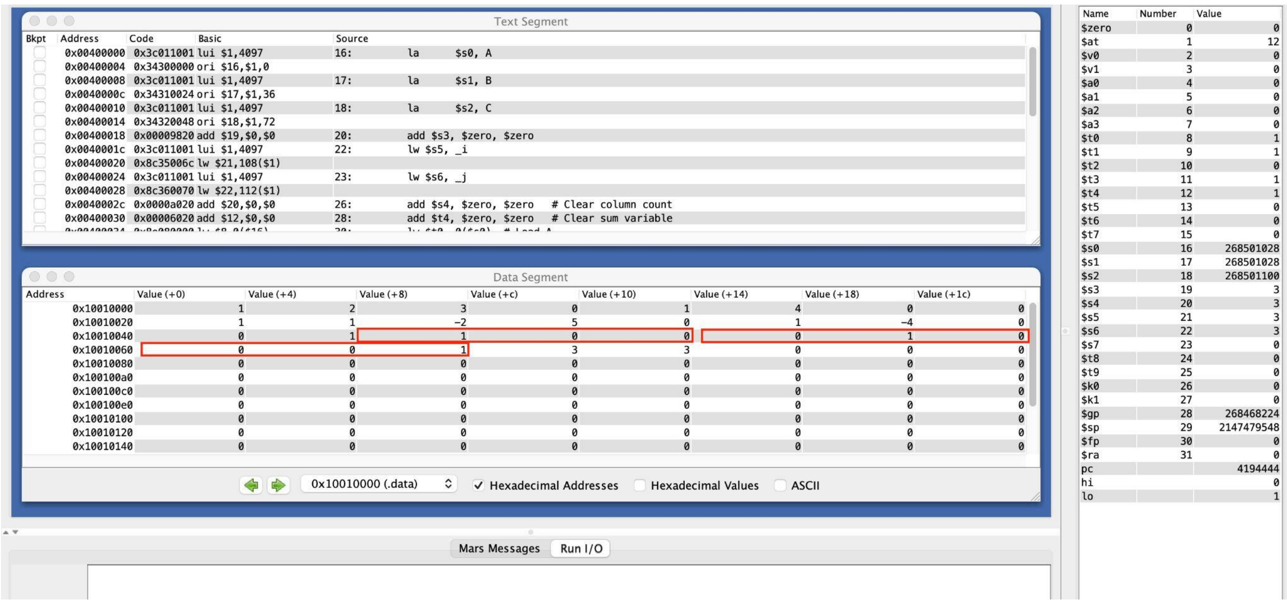
UNIVERSIDADE FEDERAL
DE SANTA CATARINA

Disciplina: INE5411 *Semestre: 2023.1*
Prof.: Marcelo Daniel Berejuck
Aluno: João Pedro Schmidt Cordeiro
Turma: 03208B

Laboratório 2

1. Exercício 1



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x3c011001	lui $1,4097	16: la $s0, A
□	0x00400004	0x34300000	ori $16,$1,0	
□	0x00400008	0x3c011001	lui $1,4097	17: la $s1, B
□	0x0040000c	0x34310024	ori $17,$1,36	
□	0x00400010	0x3c011001	lui $1,4097	18: la $s2, C
□	0x00400014	0x34320048	ori $18,$1,72	
□	0x00400018	0x00009820	add $19,$0,$0	20: add $s3, $zero, $zero
□	0x0040001c	0x3c011001	lui $1,4097	22: lw $s5, _i
□	0x00400020	0x8c35006c	lw $21,108($1)	
□	0x00400024	0x3c011001	lui $1,4097	23: lw $s6, _j
□	0x00400028	0x8c360070	lw $22,112($1)	
□	0x0040002c	0x0000a020	add $20,$0,$0	26: add $s4, $zero, $zero # Clear column count
□	0x00400030	0x00006020	add $12,$0,$0	28: add $t4, $zero, $zero # Clear sum variable

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1	2	3	0	1	4	0	0
0x10010020	1	1	-2	5	0	1	-4	0
0x10010040	0	1	1	0	0	0	1	0
0x10010060	0	0	1	3	3	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0	0	0

0x10010000 (.data) ☑ Hexadecimal Addresses □ Hexadecimal Values □ ASCII

Name	Number	Value
$zero	0	0
$at	1	12
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	1
$t1	9	1
$t2	10	0
$t3	11	1
$t4	12	1
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	268501028
$s1	17	268501028
$s2	18	268501100
$s3	19	3
$s4	20	3
$s5	21	3
$s6	22	3
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194444
hi		0
lo		1

Na imagem acima é possível constatar cada linha da matriz resultante sinalizada dentro dos três quadrados vermelhos. Desse modo, a matriz esperada foi gerada ao executar o programa.

Segue a matriz esperada abaixo:

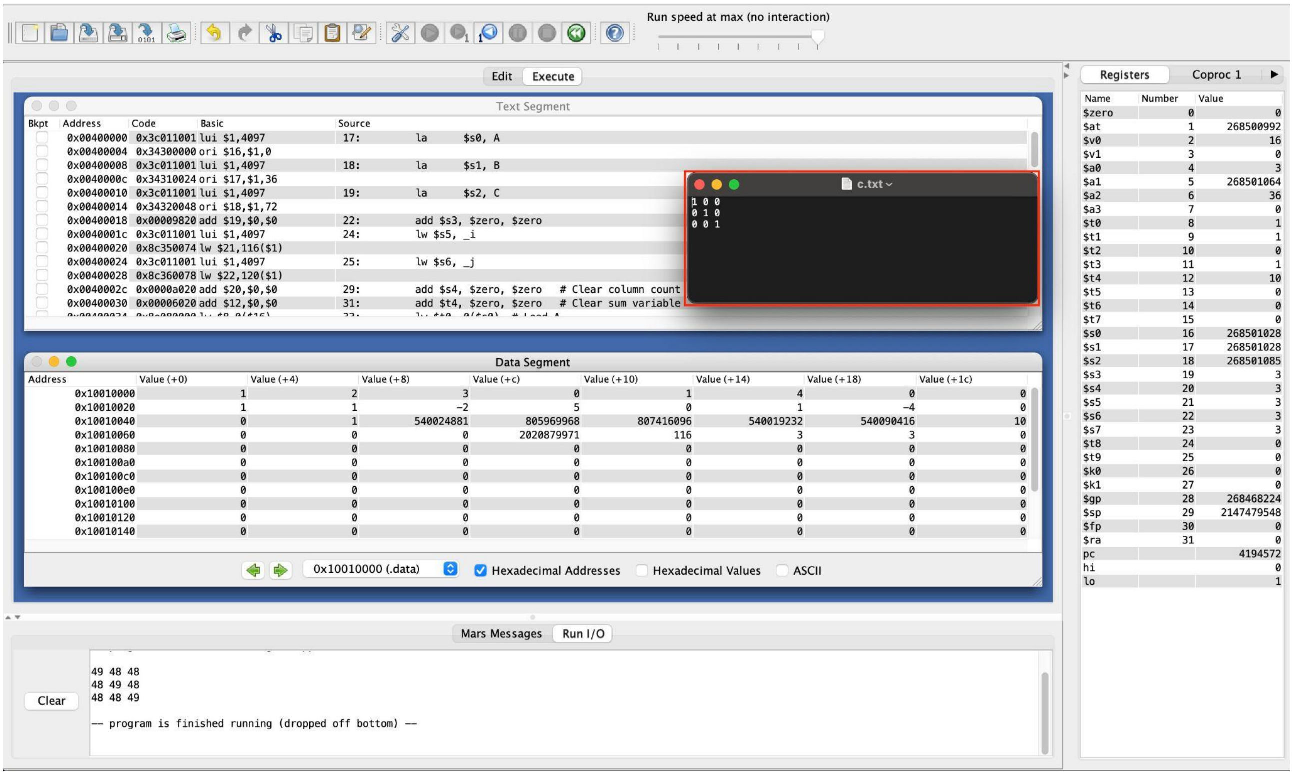
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
## Lab: 2
## Exercicio: 1
## Nome: Joao Pedro Schmidt Cordeiro
## Matricula: 22100628

.data
      A:    .word 1,2,3,0,1,4,0,0,1
      B:    .word 1,-2,5,0,1,-4,0,0,1
      C:    .space 36
      _i:   .word 3
```

```
	_j:	.word 3

.text
main:
	# Load all variable adresses
	la	$s0, A
	la	$s1, B
	la	$s2, C

	add $s3, $zero, $zero

	lw $s5, _i
	lw $s6, _j

Loop_i:
	add $s4, $zero, $zero # Clear column count
Loop_j:
	add $t4, $zero, $zero # Clear sum variable

	lw $t0, 0($s0)	# Load A
	lw $t1, 0($s1)	# Load B
	mul $t3, $t0, $t1
	add $t4, $t4, $t3

	lw $t0, 4($s0)	# Load A
	lw $t1, 12($s1)	# Load B
	mul $t3, $t0, $t1
	add $t4, $t4, $t3

	lw $t0, 8($s0)	# Load A
	lw $t1, 24($s1)	# Load B
	mul $t3, $t0, $t1
	add $t4, $t4, $t3

	sw $t4, 0($s2)		# Store C element
	addi $s2, $s2, 4# Change the C address for the next loop store in
the right position


	addi $s1, $s1, 4 # Go to next column in B

	addi $s4, $s4, 1
	bne $s6, $s4, Loop_j	# End of Loop_j
```

```
    subi $s1, $s1, 12     # Reset column address to first column
    addi $s0, $s0, 12     # Go to next line in A

    addi $s3, $s3, 1
    bne $s5, $s3, Loop_i  # End of Loop_i
```

2. Exercício 2



```
Bkpt Address    Code       Basic                Source
     0x00400000 0x3c011001 lui $1,4097          17:     la    $s0, A
     0x00400004 0x34300000 ori $16,$1,0
     0x00400008 0x3c011001 lui $1,4097          18:     la    $s1, B
     0x0040000c 0x34310024 ori $17,$1,36
     0x00400010 0x3c011001 lui $1,4097          19:     la    $s2, C
     0x00400014 0x34320048 ori $18,$1,72
     0x00400018 0x00009820 add $19,$0,$0        22:     add $s3, $zero, $zero
     0x0040001c 0x3c011001 lui $1,4097          24:     lw $s5, _i
     0x00400020 0x8c350074 lw $21,116($1)
     0x00400024 0x3c011001 lui $1,4097          25:     lw $s6, _j
     0x00400028 0x8c360078 lw $22,120($1)
     0x0040002c 0x0000a020 add $20,$0,$0        29:     add $s4, $zero, $zero   # Clear column count
     0x00400030 0x00006020 add $12,$0,$0        31:     add $t4, $zero, $zero   # Clear sum variable
```

```
1 0 0
0 1 0
0 0 1
```

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1	2	3	0	1	4	0	0
0x10010020	1	1	-2	5	0	1	-4	0
0x10010040	0	1	540024881	805969968	807416096	540019232	540090416	10
0x10010060	0	0	0	2020879971	116	3	3	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0	0	0

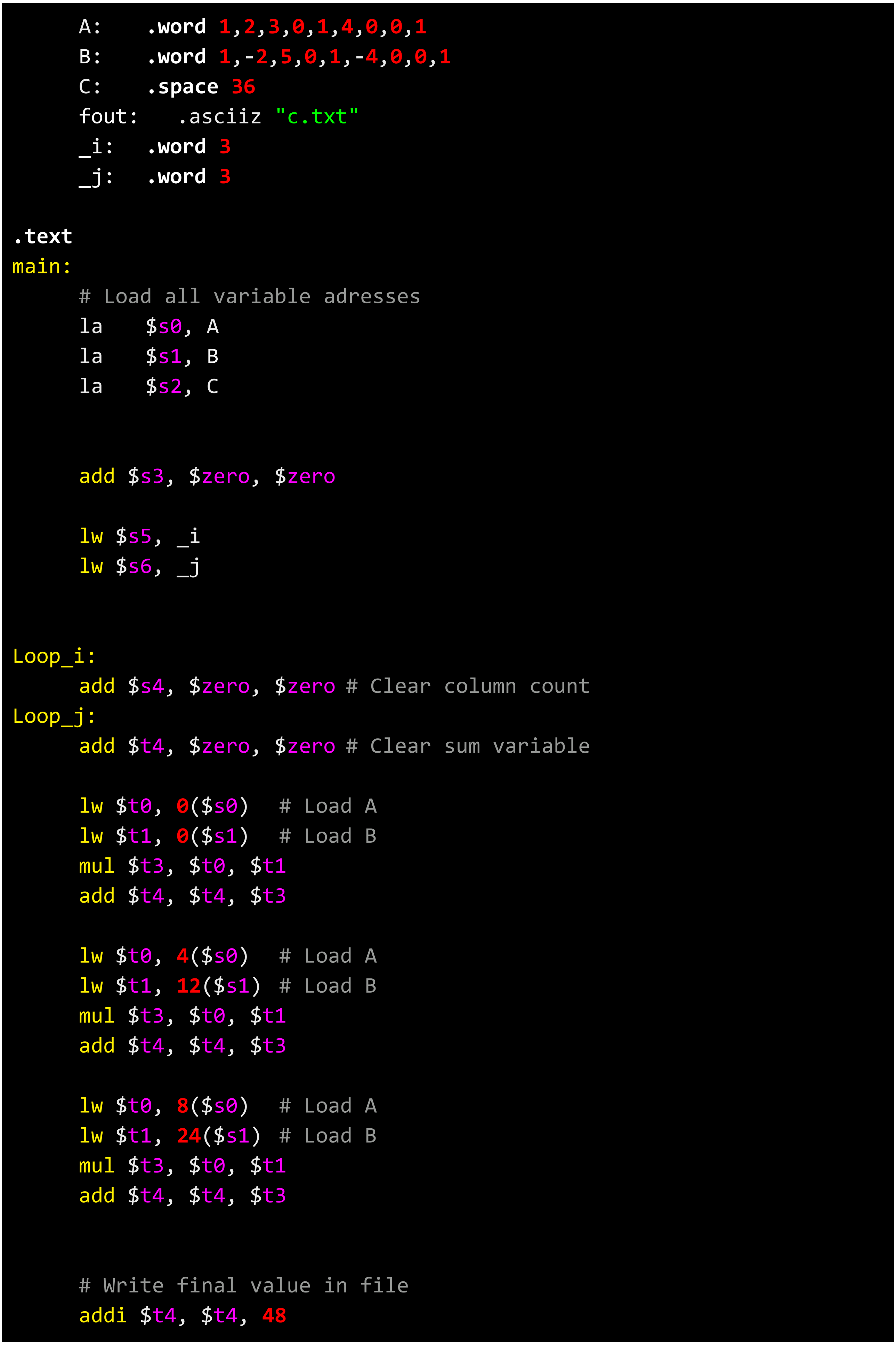
0x10010000 (.data) ☑ Hexadecimal Addresses □ Hexadecimal Values □ ASCII

Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	16
$v1	3	0
$a0	4	3
$a1	5	268501064
$a2	6	36
$a3	7	0
$t0	8	1
$t1	9	1
$t2	10	0
$t3	11	1
$t4	12	10
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	268501028
$s1	17	268501028
$s2	18	268501085
$s3	19	3
$s4	20	3
$s5	21	3
$s6	22	3
$s7	23	3
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194572
hi		0
lo		1

```
49 48 48
48 49 48
48 48 49

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

Ao executar o programa relacionado ao exercício 2 é possível observar que a matriz esperada é escrita no arquivo *c.txt* do modo esperado pelo exercício. Também é possível observar a matriz em ASCII impressa na caixa de texto *Run I/O* (parte inferior da imagem) que consta os valores 48 e 49, representando 0 e 1 respectivamente.

```
## Lab: 2
## Exercicio: 2
## Nome: Joao Pedro Schmidt Cordeiro
## Matricula: 22100628

.data
```



```
    A:    .word 1,2,3,0,1,4,0,0,1
    B:    .word 1,-2,5,0,1,-4,0,0,1
    C:    .space 36
    fout:   .asciiz "c.txt"
    _i:   .word 3
    _j:   .word 3

.text
main:
    # Load all variable adresses
    la    $s0, A
    la    $s1, B
    la    $s2, C


    add $s3, $zero, $zero

    lw $s5, _i
    lw $s6, _j


Loop_i:
    add $s4, $zero, $zero # Clear column count
Loop_j:
    add $t4, $zero, $zero # Clear sum variable

    lw $t0, 0($s0)   # Load A
    lw $t1, 0($s1)   # Load B
    mul $t3, $t0, $t1
    add $t4, $t4, $t3

    lw $t0, 4($s0)   # Load A
    lw $t1, 12($s1)  # Load B
    mul $t3, $t0, $t1
    add $t4, $t4, $t3

    lw $t0, 8($s0)   # Load A
    lw $t1, 24($s1)  # Load B
    mul $t3, $t0, $t1
    add $t4, $t4, $t3


    # Write final value in file
    addi $t4, $t4, 48
```

```
      sb $t4, 0($s2)
      addi $s2, $s2, 1
      add   $a0, $zero, $t4 # Valor a ser escrito
      li    $v0, 1     # Comando.
      syscall

      li $t4, ' '
      sb $t4, 0($s2)
      addi $s2, $s2, 1
      li    $a0, ' '   # Valor a ser escrito
      li    $v0, 11         # Comando
      syscall

      addi $s1, $s1, 4 # Go to next column in B

      addi $s4, $s4, 1
      bne $s6, $s4, Loop_j  # End of Loop_j

      # Line break
      li $t4, 10
      sb $t4, 0($s2)
      addi $s2, $s2, 1
      li    $a0, '\n'
      li    $v0, 11
      syscall

      subi $s1, $s1, 12     # Reset column address to first column
      addi $s0, $s0, 12     # Go to next line in A

      addi $s3, $s3, 1
      bne $s5, $s3, Loop_i  # End of Loop_i

      # Open (for writing) a file that does not exist
      li   $v0, 13          # system call for open file
      la   $a0, fout        # output file name
      li   $a1, 1           # Open for writing (flags are 0: read, 1:
write)
      li   $a2, 0           # mode is ignored
      syscall               # open a file (file descriptor returned in
$v0)
      move $s7, $v0         # save the file descriptor

      # Write to file just opened
      li   $v0, 15          # system call for write to file
```

```
    move $a0, $s7          # file descriptor
    la   $a1, C            # address of buffer from which to write
    li   $a2, 36           # hardcoded buffer length
    syscall                # write to file

    # Close the file
    li   $v0, 16           # system call for close file
    move $a0, $s7          # file descriptor to close
    syscall                # close file
```