UNIVERSIDADE FEDERAL DE SANTA CATARINA

CAMPUS TRINDADE

INE-DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

INE5411 - ORGANIZAÇÃO DE COMPUTADORES I

Alunos: João Victor Cabral Machado e Pedro Alfeu Wolff Lemos

Relatório do laboratório 1

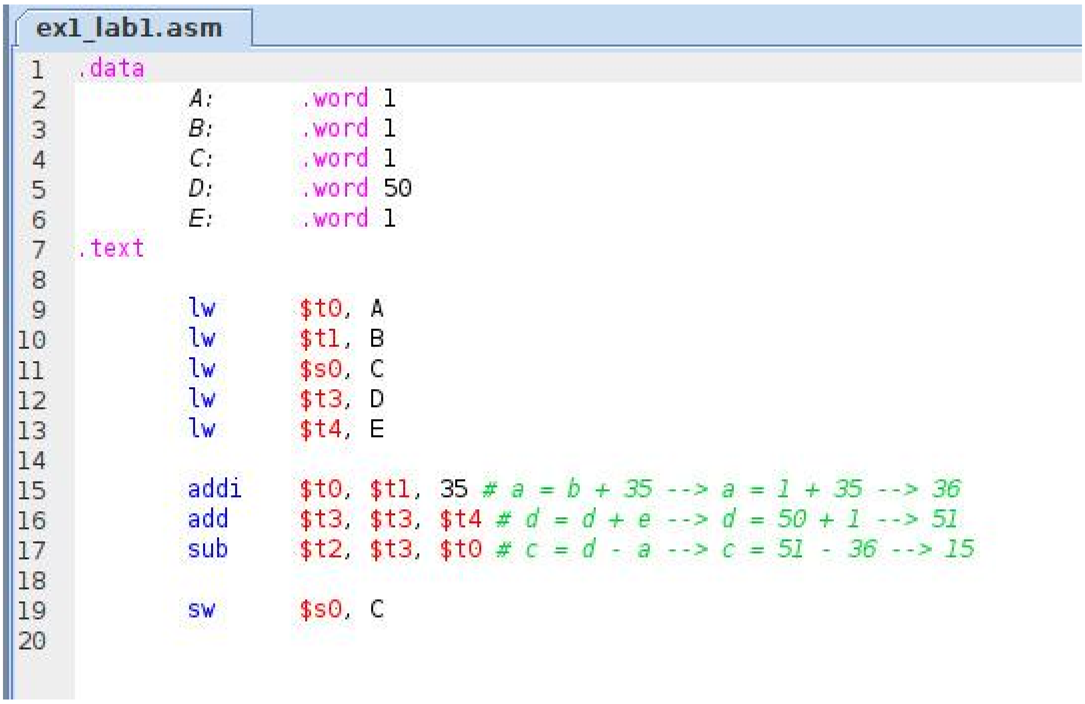
Florianópolis

2023

- **Instruções usadas:**
  lw - load word: Carrega palavra de 32 bits para um registrador
  addi - add immediate: Adiciona um valor imediato no registrador
  add: Adiciona o valor de dois registradores e armazena em um terceiro
  sub - subtract: Subtrai o valor de dois registradores e armazena em um terceiro
  sw - store word: Armazena a palavra de um registrador na memória
  syscall: Usada para chamar o sistema operacional da máquina

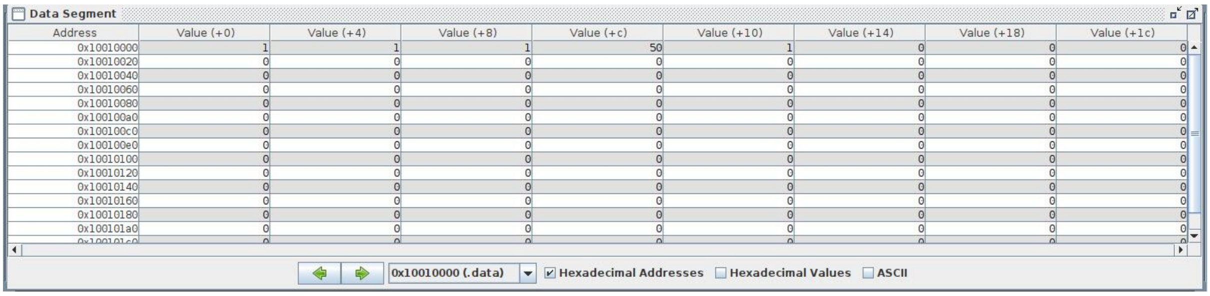
Exercício 1:

Código em assembly



```
.data
        A:      .word 1
        B:      .word 1
        C:      .word 1
        D:      .word 50
        E:      .word 1
.text

        lw      $t0, A
        lw      $t1, B
        lw      $s0, C
        lw      $t3, D
        lw      $t4, E

        addi    $t0, $t1, 35 # a = b + 35 --> a = 1 + 35 --> 36
        add     $t3, $t3, $t4 # d = d + e --> d = 50 + 1 --> 51
        sub     $t2, $t3, $t0 # c = d - a --> c = 51 - 36 --> 15

        sw      $s0, C

```

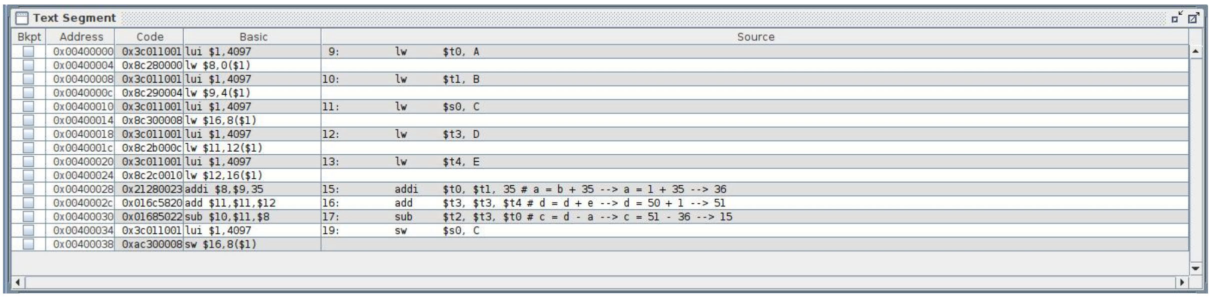
Data Segment



Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1	1	1	50	1	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	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
0x10010160	0	0	0	0	0	0	0	0
0x10010180	0	0	0	0	0	0	0	0
0x100101a0	0	0	0	0	0	0	0	0

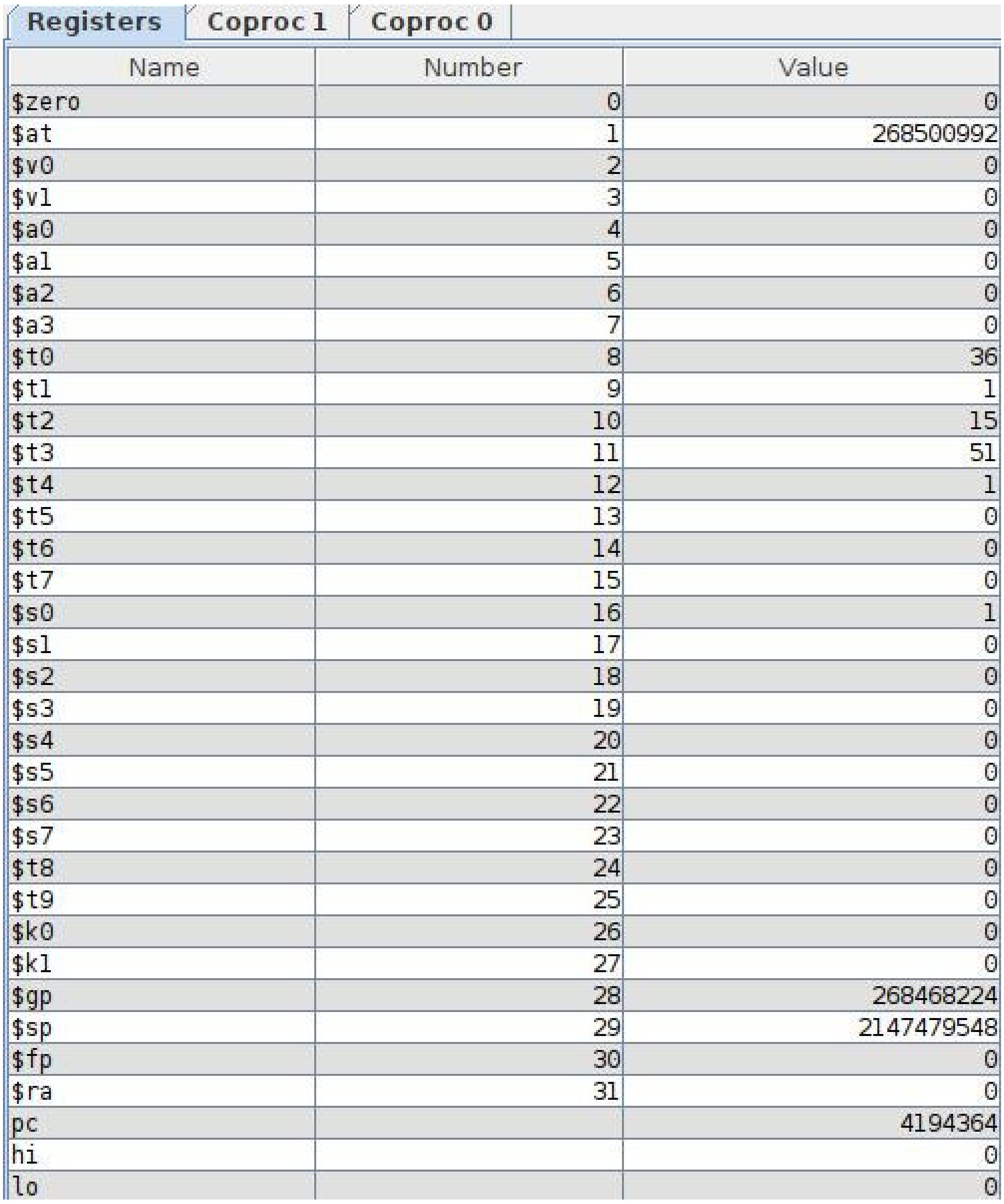
0x10010000 (.data) ☑ Hexadecimal Addresses □ Hexadecimal Values □ ASCII

Text segment



Bkpt	Address	Code	Basic	Source
□	0x00400000	0x3c011001	lui $1,4097	9: lw $t0, A
□	0x00400004	0x8c280000	lw $8,0($1)	
□	0x00400008	0x3c011001	lui $1,4097	10: lw $t1, B
□	0x0040000c	0x8c290004	lw $9,4($1)	
□	0x00400010	0x3c011001	lui $1,4097	11: lw $s0, C
□	0x00400014	0x8c300008	lw $16,8($1)	
□	0x00400018	0x3c011001	lui $1,4097	12: lw $t3, D
□	0x0040001c	0x8c2b000c	lw $11,12($1)	
□	0x00400020	0x3c011001	lui $1,4097	13: lw $t4, E
□	0x00400024	0x8c2c0010	lw $12,16($1)	
□	0x00400028	0x21280023	addi $8,$9,35	15: addi $t0, $t1, 35 # a = b + 35 --> a = 1 + 35 --> 36
□	0x0040002c	0x016c5820	add $11,$11,$12	16: add $t3, $t3, $t4 # d = d + e --> d = 50 + 1 --> 51
□	0x00400030	0x01685022	sub $10,$11,$8	17: sub $t2, $t3, $t0 # c = d - a --> c = 51 - 36 --> 15
□	0x00400034	0x3c011001	lui $1,4097	19: sw $s0, C
□	0x00400038	0xac300008	sw $16,8($1)	

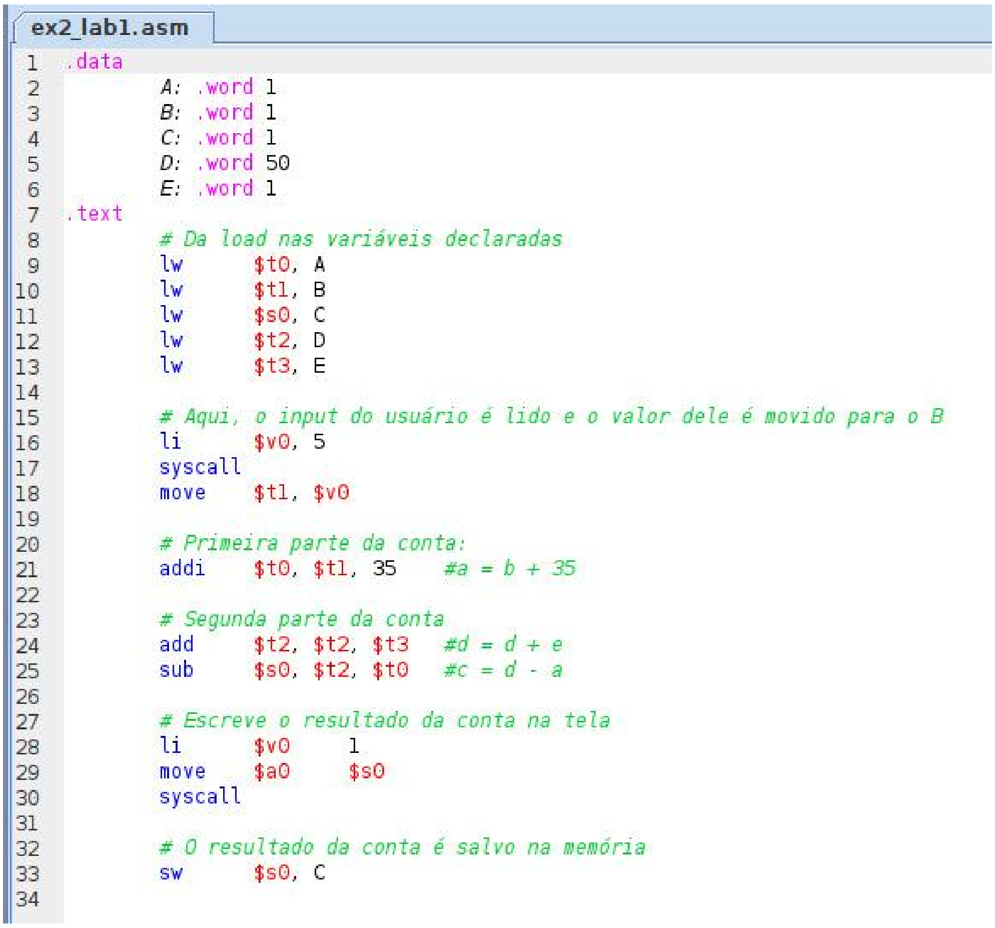
Registradores



Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	36
$t1	9	1
$t2	10	15
$t3	11	51
$t4	12	1
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	1
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$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		4194364
hi		0
lo		0

Exercício 2:

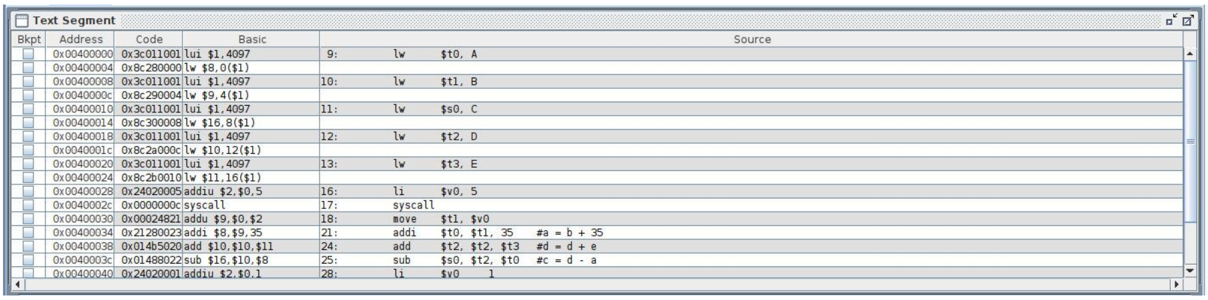
Código em Assembly



ex2_lab1.asm

```
.data
        A: .word 1
        B: .word 1
        C: .word 1
        D: .word 50
        E: .word 1
.text
        # Da load nas variáveis declaradas
        lw      $t0, A
        lw      $t1, B
        lw      $s0, C
        lw      $t2, D
        lw      $t3, E

        # Aqui, o input do usuário é lido e o valor dele é movido para o B
        li      $v0, 5
        syscall
        move    $t1, $v0

        # Primeira parte da conta:
        addi    $t0, $t1, 35    #a = b + 35

        # Segunda parte da conta
        add     $t2, $t2, $t3   #d = d + e
        sub     $s0, $t2, $t0   #c = d - a

        # Escreve o resultado da conta na tela
        li      $v0     1
        move    $a0     $s0
        syscall

        # O resultado da conta é salvo na memória
        sw      $s0, C

```

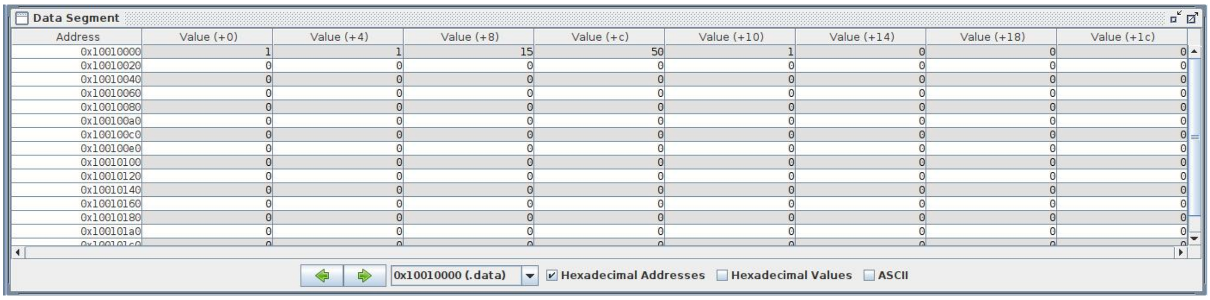
Text Segment



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x3c011001	lui $1,4097	9: lw $t0, A
□	0x00400004	0x8c280000	lw $8,0($1)	
□	0x00400008	0x3c011001	lui $1,4097	10: lw $t1, B
□	0x0040000c	0x8c290004	lw $9,4($1)	
□	0x00400010	0x3c011001	lui $1,4097	11: lw $s0, C
□	0x00400014	0x8c300008	lw $16,8($1)	
□	0x00400018	0x3c011001	lui $1,4097	12: lw $t2, D
□	0x0040001c	0x8c2a000c	lw $10,12($1)	
□	0x00400020	0x3c011001	lui $1,4097	13: lw $t3, E
□	0x00400024	0x8c2b0010	lw $11,16($1)	
□	0x00400028	0x24020005	addiu $2,$0,5	16: li $v0, 5
□	0x0040002c	0x0000000c	syscall	17: syscall
□	0x00400030	0x00024821	addu $9,$0,$2	18: move $t1, $v0
□	0x00400034	0x21280023	addi $8,$9,35	21: addi $t0, $t1, 35 #a = b + 35
□	0x00400038	0x014b5020	add $10,$10,$11	24: add $t2, $t2, $t3 #d = d + e
□	0x0040003c	0x01488022	sub $16,$10,$8	25: sub $s0, $t2, $t0 #c = d - a
□	0x00400040	0x24020001	addiu $2,$0,1	28: li $v0 1

Data segment

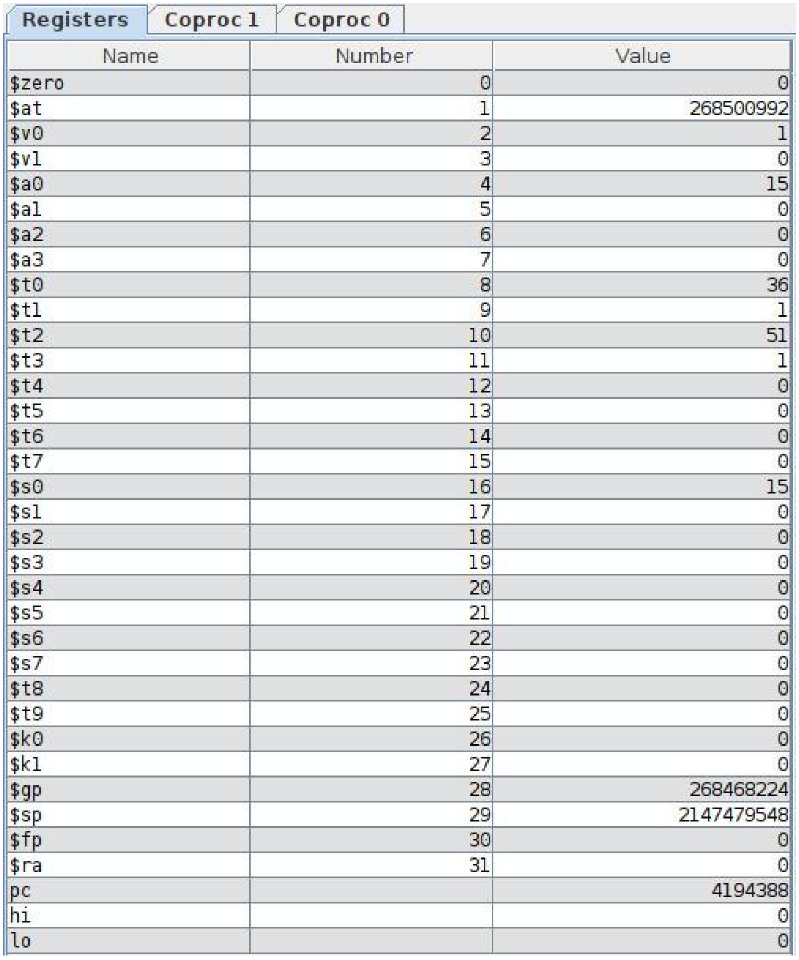


Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1	1	15	50	1	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	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
0x10010160	0	0	0	0	0	0	0	0
0x10010180	0	0	0	0	0	0	0	0
0x100101a0	0	0	0	0	0	0	0	0

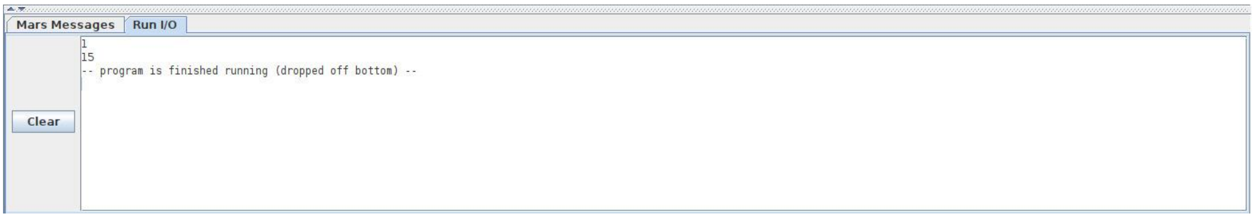
0x10010000 (.data) ☑ Hexadecimal Addresses □ Hexadecimal Values □ ASCII

Registradores



Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	1
$v1	3	0
$a0	4	15
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	36
$t1	9	1
$t2	10	51
$t3	11	1
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	15
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$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		4194388
hi		0
lo		0

Terminal



```
1
15
-- program is finished running (dropped off bottom) --
```