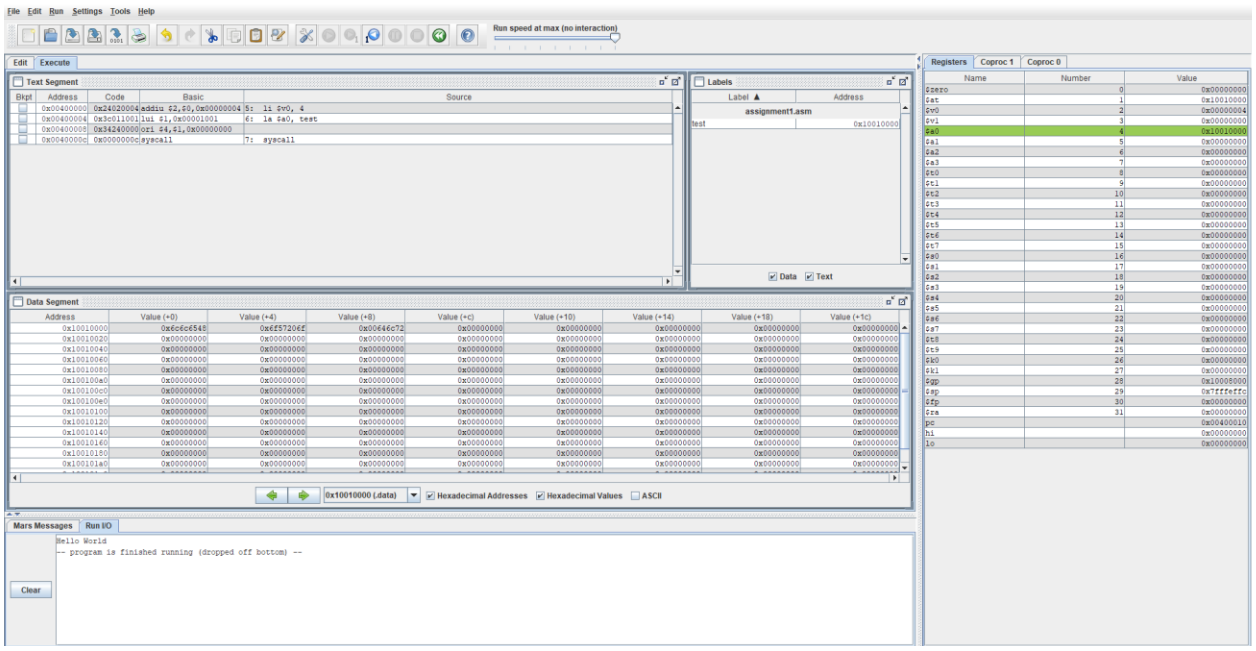
Computer Architecture Lab Report Week 5

Full name: On Quang Tung

Student ID: 20226096

Assignment 1

```
#Laboratory Exercise 5, Home Assignment 1
.data
test: .asciiz "Hello World"
.text
 li $v0, 4
 la $a0, test
 syscall
```



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x24020004	addiu $2,$0,0x00000004	5: li $v0, 4
☐	0x00400004	0x3c011001	lui $1,0x00001001	6: la $a0, test
☐	0x00400008	0x34240000	ori $4,$1,0x00000000	
☐	0x0040000c	0x0000000c	syscall	7: syscall

Labels

Label	Address
assignment1.asm	
test	0x10010000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x6c6c6548	0x6f57206f	0x00646c72	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

Run I/O

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

Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000004
$v1	3	0x00000000
$a0	4	0x10010000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400010
hi		0x00000000
lo		0x00000000

➔ Chương trình chạy đúng kết quả mong đợi.

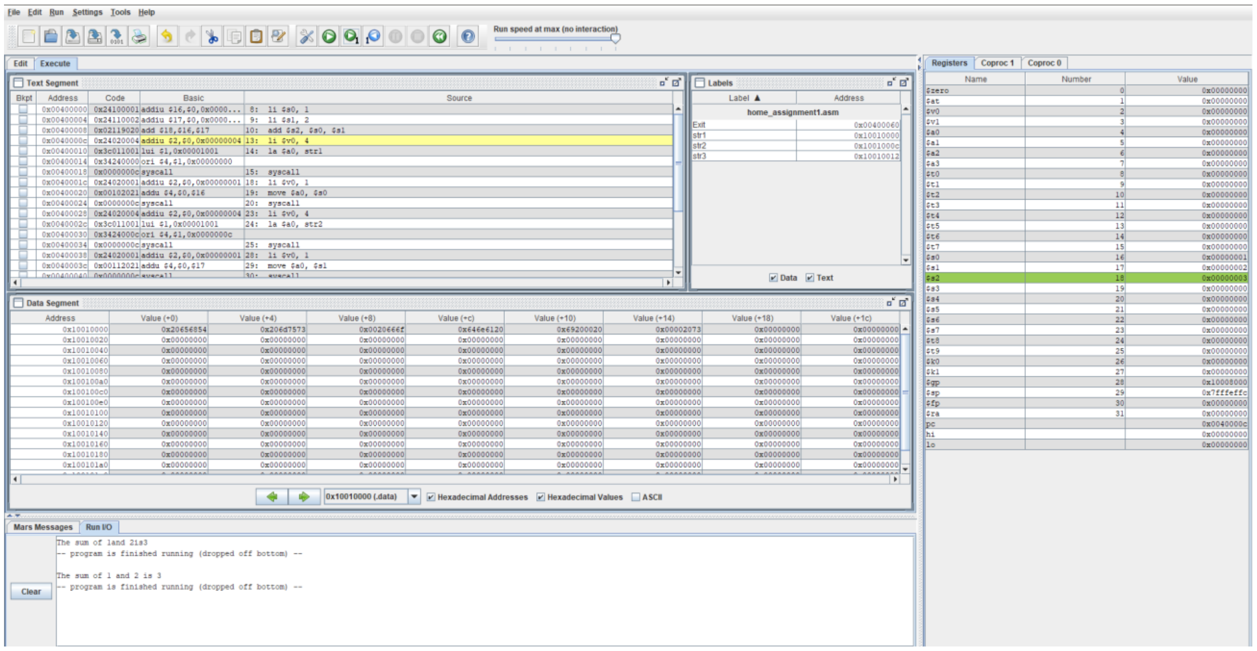
Assignment 2

```
.data
str1: .asciiz "The sum of "
str2: .asciiz " and "
str3: .asciiz " is "
```

```
.text
 li $s0, 1
 li $s1, 2
 add $s2, $s0, $s1

 #Print s1 = "The sum of "
 li $v0, 4
 la $a0, str1
 syscall

 #Print $s0
 li $v0, 1
 move $a0, $s0
 syscall

 #Print s2 = " and "
 li $v0, 4
 la $a0, str2
 syscall

 #Print $s1
 li $v0, 1
 move $a0, $s1
 syscall

 #Print s3 = " is "
 li $v0, 4
 la $a0, str3
 syscall

 #Print $s2
 li $v0, 1
 move $a0, $s2
 syscall

 Exit: li $v0, 10
 syscall
```

Kết quả chạy:

- Load các giá trị $s0, $s1, và $s2



File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Data Segment
Registers Coproc 1 Coproc 0
Mars Messages Run I/O
Clear

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24100001	addiu $16,$0,0x0000...	8: li $s0, 1
	0x00400004	0x24110002	addiu $17,$0,0x0000...	9: li $s1, 2
	0x00400008	0x02119020	add $18,$16,$17	10: add $s2, $s0, $s1
	0x0040000c	0x24020004	addiu $2,$0,0x00000004	13: li $v0, 4
	0x00400010	0x3c011001	lui $1,0x00001001	14: la $a0, str1
	0x00400014	0x34240000	ori $4,$1,0x00000000	
	0x00400018	0x0000000c	syscall	15: syscall
	0x0040001c	0x24020001	addiu $2,$0,0x00000001	18: li $v0, 1
	0x00400020	0x00102021	addu $4,$0,$16	19: move $a0, $s0
	0x00400024	0x0000000c	syscall	20: syscall
	0x00400028	0x24020004	addiu $2,$0,0x00000004	23: li $v0, 4
	0x0040002c	0x3c011001	lui $1,0x00001001	24: la $a0, str2
	0x00400030	0x3424000c	ori $4,$1,0x0000000c	
	0x00400034	0x0000000c	syscall	25: syscall
	0x00400038	0x24020001	addiu $2,$0,0x00000001	28: li $v0, 1
	0x0040003c	0x00112021	addu $4,$0,$17	29: move $a0, $s1
	0x00400040	0x0000000c	syscall	30: syscall

Labels

Label ▲	Address
home_assignment1.asm	
Exit	0x00400060
str1	0x10010000
str2	0x1001000c
str3	0x10010012

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x20656854	0x206d7573	0x0020666f	0x646e6120	0x69200020	0x00002073	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

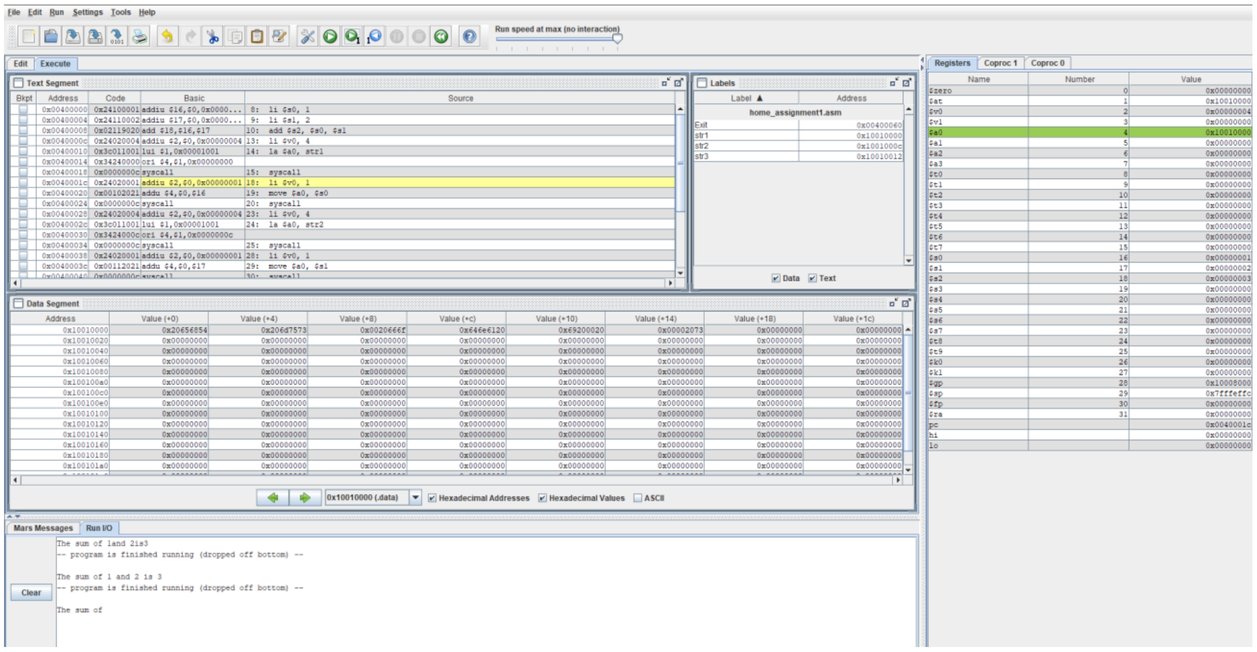
0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000001
$s1	17	0x00000002
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040000c
hi		0x00000000
lo		0x00000000

```
The sum of land 2is3
-- program is finished running (dropped off bottom) --

The sum of 1 and 2 is 3
-- program is finished running (dropped off bottom) --
```

- In ra "The sum of "



File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Data Segment
Registers Coproc 1 Coproc 0
Mars Messages Run I/O
Clear

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24100001	addiu $16,$0,0x0000...	8: li $s0, 1
	0x00400004	0x24110002	addiu $17,$0,0x0000...	9: li $s1, 2
	0x00400008	0x02119020	add $18,$16,$17	10: add $s2, $s0, $s1
	0x0040000c	0x24020004	addiu $2,$0,0x00000004	13: li $v0, 4
	0x00400010	0x3c011001	lui $1,0x00001001	14: la $a0, str1
	0x00400014	0x34240000	ori $4,$1,0x00000000	
	0x00400018	0x0000000c	syscall	15: syscall
	0x0040001c	0x24020001	addiu $2,$0,0x00000001	18: li $v0, 1
	0x00400020	0x00102021	addu $4,$0,$16	19: move $a0, $s0
	0x00400024	0x0000000c	syscall	20: syscall
	0x00400028	0x24020004	addiu $2,$0,0x00000004	23: li $v0, 4
	0x0040002c	0x3c011001	lui $1,0x00001001	24: la $a0, str2
	0x00400030	0x3424000c	ori $4,$1,0x0000000c	
	0x00400034	0x0000000c	syscall	25: syscall
	0x00400038	0x24020001	addiu $2,$0,0x00000001	28: li $v0, 1
	0x0040003c	0x00112021	addu $4,$0,$17	29: move $a0, $s1
	0x00400040	0x0000000c	syscall	30: syscall

Labels

Label ▲	Address
home_assignment1.asm	
Exit	0x00400060
str1	0x10010000
str2	0x1001000c
str3	0x10010012

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x20656854	0x206d7573	0x0020666f	0x646e6120	0x69200020	0x00002073	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

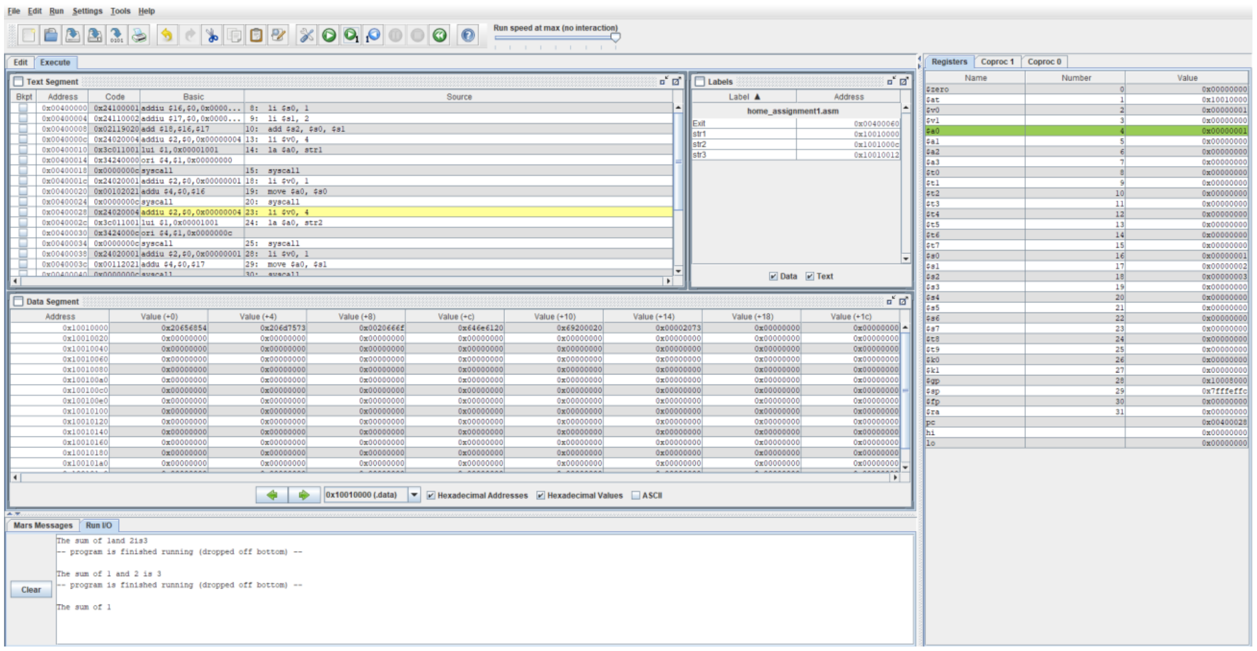
0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000004
$v1	3	0x00000000
$a0	4	0x10010000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000001
$s1	17	0x00000002
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040001c
hi		0x00000000
lo		0x00000000

```
The sum of land 2is3
-- program is finished running (dropped off bottom) --

The sum of 1 and 2 is 3
-- program is finished running (dropped off bottom) --

The sum of
```

- In ra $s0



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24100001	addiu $16,$0,0x0000...	8: li $s0, 1
	0x00400004	0x24110002	addiu $17,$0,0x0000...	9: li $s1, 2
	0x00400008	0x02119020	add $18,$16,$17	10: add $s2, $s0, $s1
	0x0040000c	0x24020004	addiu $2,$0,0x00000004	13: li $v0, 4
	0x00400010	0x3c011001	lui $1,0x00001001	14: la $a0, str1
	0x00400014	0x34240000	ori $4,$1,0x00000000	
	0x00400018	0x0000000c	syscall	15: syscall
	0x0040001c	0x24020001	addiu $2,$0,0x00000001	18: li $v0, 1
	0x00400020	0x00102021	addu $4,$0,$16	19: move $a0, $s0
	0x00400024	0x0000000c	syscall	20: syscall
	0x00400028	0x24020004	addiu $2,$0,0x00000004	23: li $v0, 4
	0x0040002c	0x3c011001	lui $1,0x00001001	24: la $a0, str2
	0x00400030	0x3424000c	ori $4,$1,0x0000000c	
	0x00400034	0x0000000c	syscall	25: syscall
	0x00400038	0x24020001	addiu $2,$0,0x00000001	28: li $v0, 1
	0x0040003c	0x00112021	addu $4,$0,$17	29: move $a0, $s1
	0x00400040	0x0000000c	syscall	30: syscall

Labels

Label	Address
home_assignment1.asm	
Exit	0x00400060
str1	0x10010000
str2	0x1001000c
str3	0x10010012

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x20656854	0x206d7573	0x0020666f	0x646e6120	0x69200020	0x00002073	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

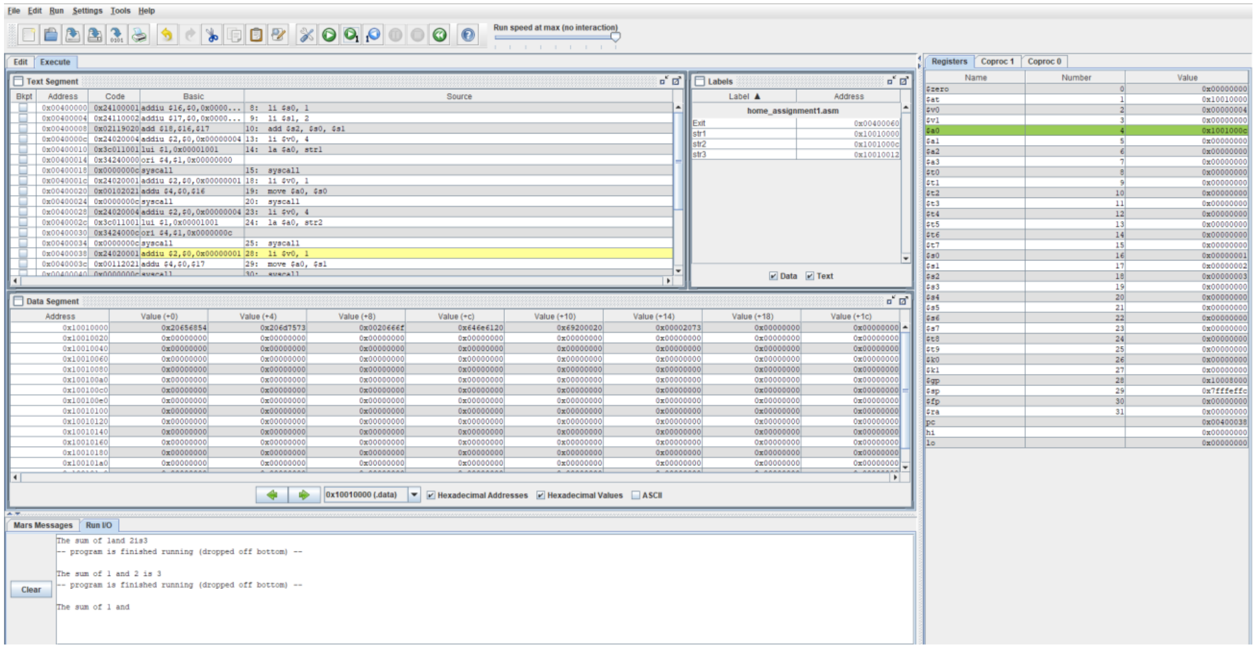
Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000001
$v1	3	0x00000000
$a0	4	0x00000001
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000001
$s1	17	0x00000002
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400028
hi		0x00000000
lo		0x00000000

Run I/O

```
The sum of land 2is3
-- program is finished running (dropped off bottom) --

The sum of 1 and 2 is 3
-- program is finished running (dropped off bottom) --

The sum of 1
```

- In ra " and "



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24100001	addiu $16,$0,0x0000...	8: li $s0, 1
	0x00400004	0x24110002	addiu $17,$0,0x0000...	9: li $s1, 2
	0x00400008	0x02119020	add $18,$16,$17	10: add $s2, $s0, $s1
	0x0040000c	0x24020004	addiu $2,$0,0x00000004	13: li $v0, 4
	0x00400010	0x3c011001	lui $1,0x00001001	14: la $a0, str1
	0x00400014	0x34240000	ori $4,$1,0x00000000	
	0x00400018	0x0000000c	syscall	15: syscall
	0x0040001c	0x24020001	addiu $2,$0,0x00000001	18: li $v0, 1
	0x00400020	0x00102021	addu $4,$0,$16	19: move $a0, $s0
	0x00400024	0x0000000c	syscall	20: syscall
	0x00400028	0x24020004	addiu $2,$0,0x00000004	23: li $v0, 4
	0x0040002c	0x3c011001	lui $1,0x00001001	24: la $a0, str2
	0x00400030	0x3424000c	ori $4,$1,0x0000000c	
	0x00400034	0x0000000c	syscall	25: syscall
	0x00400038	0x24020001	addiu $2,$0,0x00000001	28: li $v0, 1
	0x0040003c	0x00112021	addu $4,$0,$17	29: move $a0, $s1
	0x00400040	0x0000000c	syscall	30: syscall

Labels

Label	Address
home_assignment1.asm	
Exit	0x00400060
str1	0x10010000
str2	0x1001000c
str3	0x10010012

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x20656854	0x206d7573	0x0020666f	0x646e6120	0x69200020	0x00002073	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

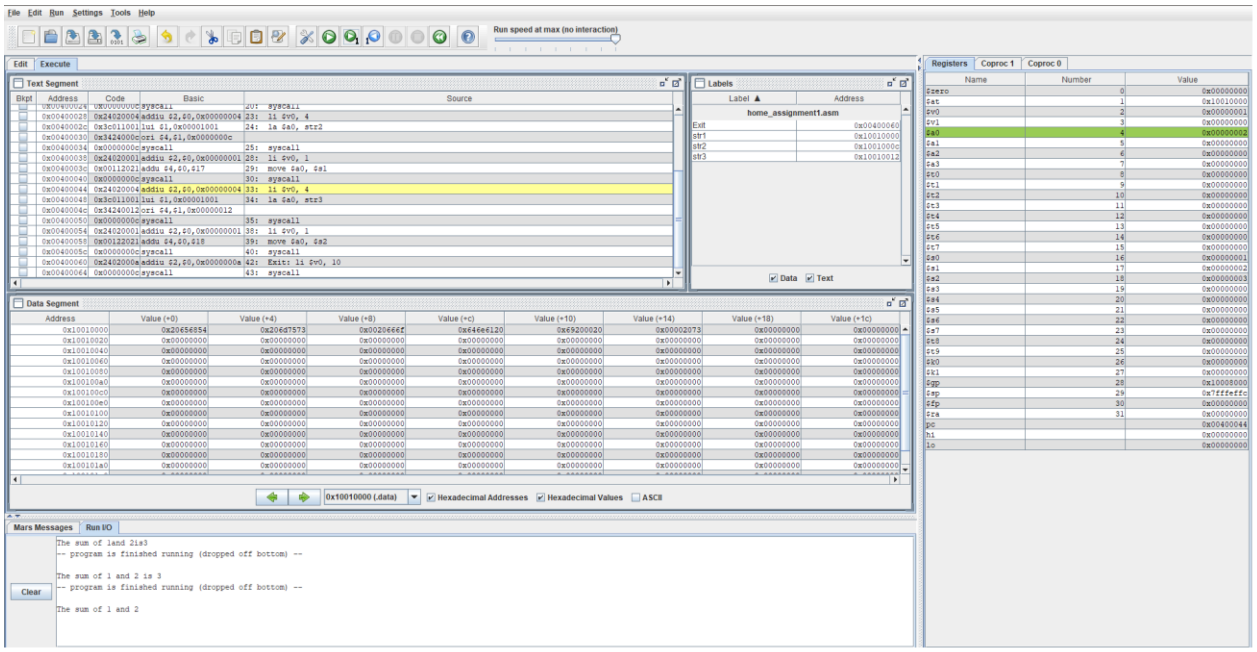
Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000004
$v1	3	0x00000000
$a0	4	0x1001000c
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000001
$s1	17	0x00000002
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400038
hi		0x00000000
lo		0x00000000

Run I/O

```
The sum of land 2is3
-- program is finished running (dropped off bottom) --

The sum of 1 and 2 is 3
-- program is finished running (dropped off bottom) --

The sum of 1 and
```

- In ra $s1



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400024	0x0000000c	syscall	20: syscall
	0x00400028	0x24020004	addiu $2,$0,0x00000004	23: li $v0, 4
	0x0040002c	0x3c011001	lui $1,0x00001001	24: la $a0, str2
	0x00400030	0x3424000c	ori $4,$1,0x0000000c	
	0x00400034	0x0000000c	syscall	25: syscall
	0x00400038	0x24020001	addiu $2,$0,0x00000001	28: li $v0, 1
	0x0040003c	0x00112021	addu $4,$0,$17	29: move $a0, $s1
	0x00400040	0x0000000c	syscall	30: syscall
	0x00400044	0x24020004	addiu $2,$0,0x00000004	33: li $v0, 4
	0x00400048	0x3c011001	lui $1,0x00001001	34: la $a0, str3
	0x0040004c	0x34240012	ori $4,$1,0x00000012	
	0x00400050	0x0000000c	syscall	35: syscall
	0x00400054	0x24020001	addiu $2,$0,0x00000001	38: li $v0, 1
	0x00400058	0x00122021	addu $4,$0,$18	39: move $a0, $s2
	0x0040005c	0x0000000c	syscall	40: syscall
	0x00400060	0x2402000a	addiu $2,$0,0x0000000a	42: Exit: li $v0, 10
	0x00400064	0x0000000c	syscall	43: syscall

Labels

Label	Address
home_assignment1.asm	
Exit	0x00400060
str1	0x10010000
str2	0x1001000c
str3	0x10010012

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x20656854	0x206d7573	0x0020666f	0x646e6120	0x69200020	0x00002073	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

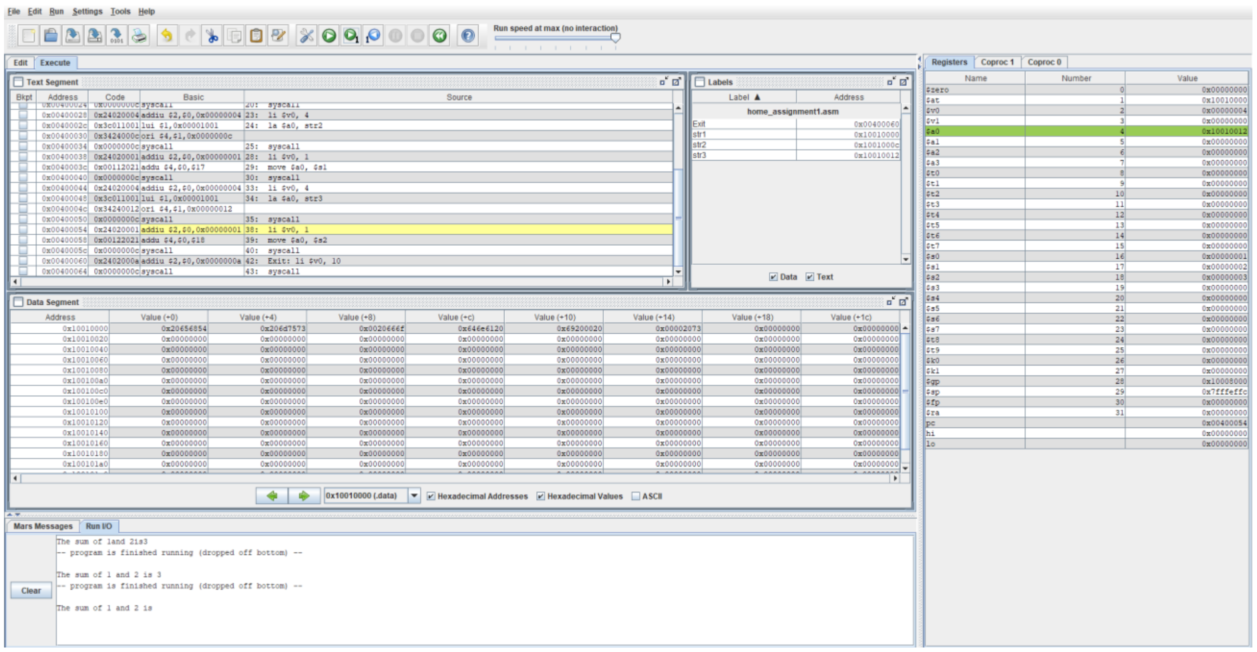
Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000001
$v1	3	0x00000000
$a0	4	0x00000002
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000001
$s1	17	0x00000002
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400044
hi		0x00000000
lo		0x00000000

Run I/O

```
The sum of land 2is3
-- program is finished running (dropped off bottom) --

The sum of 1 and 2 is 3
-- program is finished running (dropped off bottom) --

The sum of 1 and 2
```

- In ra " is "



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400024	0x0000000c	syscall	20: syscall
	0x00400028	0x24020004	addiu $2,$0,0x00000004	23: li $v0, 4
	0x0040002c	0x3c011001	lui $1,0x00001001	24: la $a0, str2
	0x00400030	0x3424000c	ori $4,$1,0x0000000c	
	0x00400034	0x0000000c	syscall	25: syscall
	0x00400038	0x24020001	addiu $2,$0,0x00000001	28: li $v0, 1
	0x0040003c	0x00112021	addu $4,$0,$17	29: move $a0, $s1
	0x00400040	0x0000000c	syscall	30: syscall
	0x00400044	0x24020004	addiu $2,$0,0x00000004	33: li $v0, 4
	0x00400048	0x3c011001	lui $1,0x00001001	34: la $a0, str3
	0x0040004c	0x34240012	ori $4,$1,0x00000012	
	0x00400050	0x0000000c	syscall	35: syscall
	0x00400054	0x24020001	addiu $2,$0,0x00000001	38: li $v0, 1
	0x00400058	0x00122021	addu $4,$0,$18	39: move $a0, $s2
	0x0040005c	0x0000000c	syscall	40: syscall
	0x00400060	0x2402000a	addiu $2,$0,0x0000000a	42: Exit: li $v0, 10
	0x00400064	0x0000000c	syscall	43: syscall

Labels

Label	Address
home_assignment1.asm	
Exit	0x00400060
str1	0x10010000
str2	0x1001000c
str3	0x10010012

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x20656854	0x206d7573	0x0020666f	0x646e6120	0x69200020	0x00002073	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

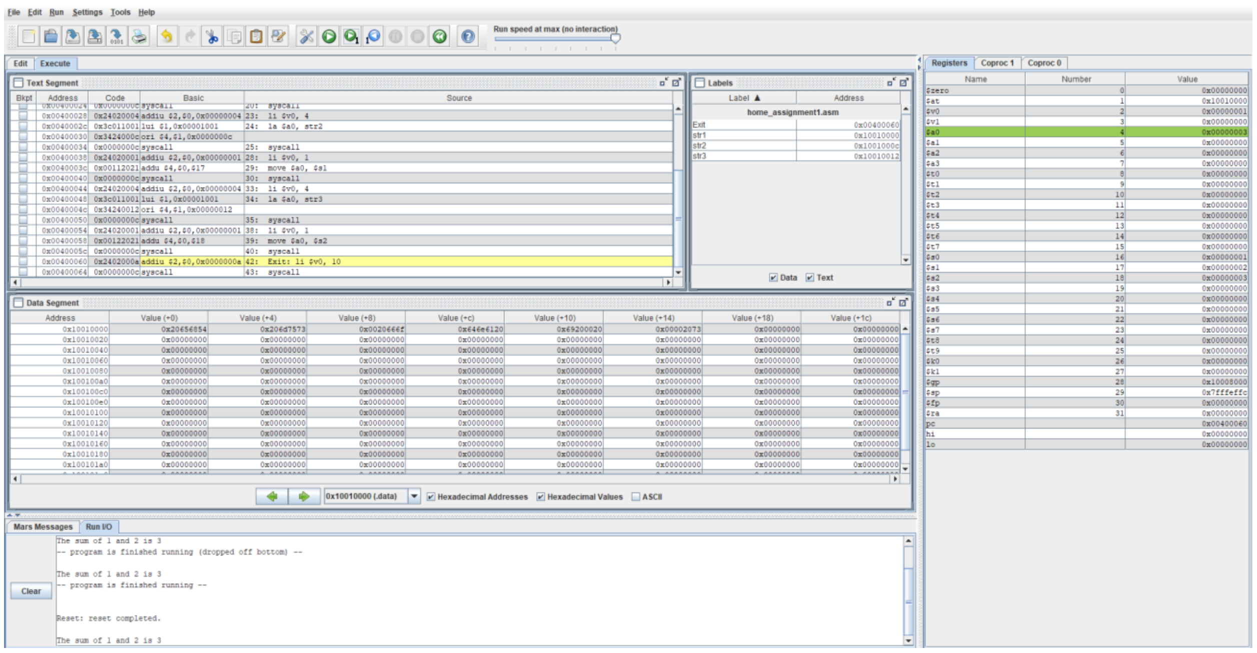
Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000004
$v1	3	0x00000000
$a0	4	0x10010012
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000001
$s1	17	0x00000002
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400054
hi		0x00000000
lo		0x00000000

Run I/O

```
The sum of land 2is3
-- program is finished running (dropped off bottom) --

The sum of 1 and 2 is 3
-- program is finished running (dropped off bottom) --

The sum of 1 and 2 is
```

- In ra $s2 (=3 do $s0 = 1 + $s1 = 2)



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400024	0x0000000c	syscall	20: syscall
	0x00400028	0x24020004	addiu $2,$0,0x00000004	23: li $v0, 4
	0x0040002c	0x3c011001	lui $1,0x00001001	24: la $a0, str2
	0x00400030	0x3424000c	ori $4,$1,0x0000000c	
	0x00400034	0x0000000c	syscall	25: syscall
	0x00400038	0x24020001	addiu $2,$0,0x00000001	28: li $v0, 1
	0x0040003c	0x00112021	addu $4,$0,$17	29: move $a0, $s1
	0x00400040	0x0000000c	syscall	30: syscall
	0x00400044	0x24020004	addiu $2,$0,0x00000004	33: li $v0, 4
	0x00400048	0x3c011001	lui $1,0x00001001	34: la $a0, str3
	0x0040004c	0x34240012	ori $4,$1,0x00000012	
	0x00400050	0x0000000c	syscall	35: syscall
	0x00400054	0x24020001	addiu $2,$0,0x00000001	38: li $v0, 1
	0x00400058	0x00122021	addu $4,$0,$18	39: move $a0, $s2
	0x0040005c	0x0000000c	syscall	40: syscall
	0x00400060	0x2402000a	addiu $2,$0,0x0000000a	42: Exit: li $v0, 10
	0x00400064	0x0000000c	syscall	43: syscall

Labels

Label	Address
home_assignment1.asm	
Exit	0x00400060
str1	0x10010000
str2	0x1001000c
str3	0x10010012

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x20656854	0x206d7573	0x0020666f	0x646e6120	0x69200020	0x00002073	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

Registers

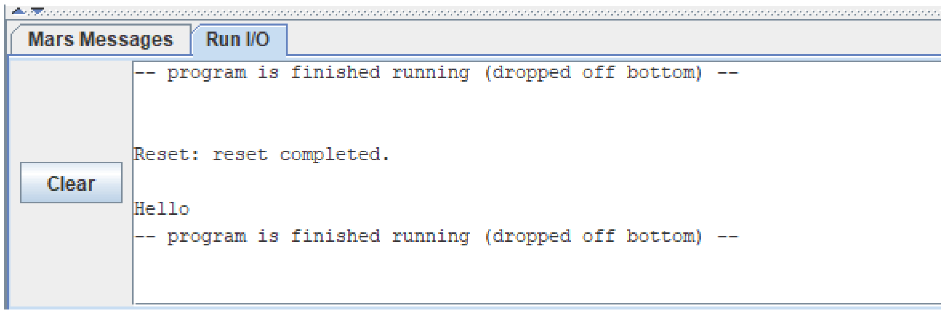
Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000001
$v1	3	0x00000000
$a0	4	0x00000003
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000001
$s1	17	0x00000002
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400060
hi		0x00000000
lo		0x00000000

Run I/O

```
The sum of 1 and 2 is 3
-- program is finished running (dropped off bottom) --

The sum of 1 and 2 is 3
-- program is finished running --


Reset: reset completed.

The sum of 1 and 2 is 3
```

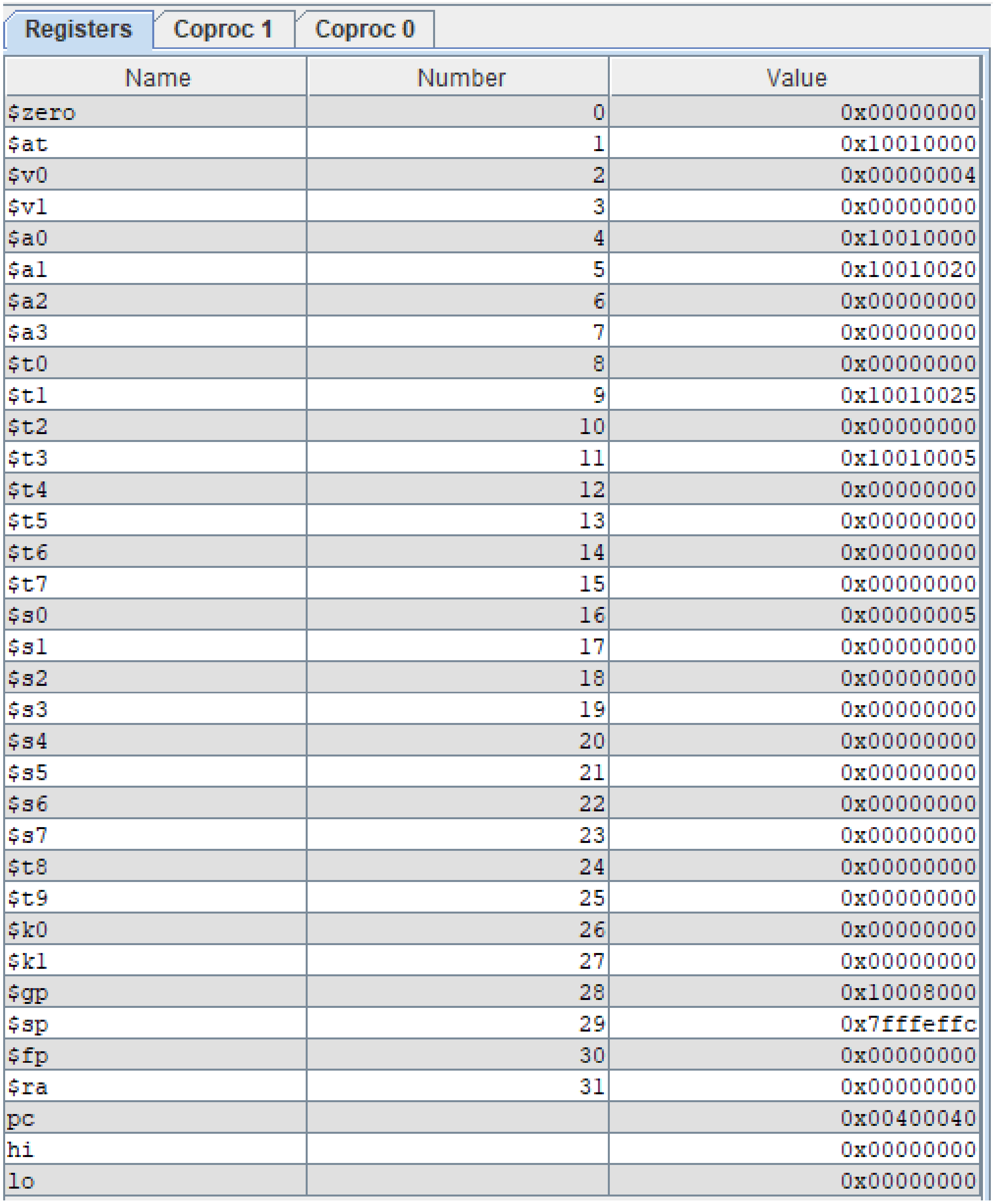
Assignment 3

```
#Laboratory Exercise 5, Home Assignment 2
.data
x: .space 32 # destination string x, empty
y: .asciiz "Hello" # source string y
.text
la $a0, x
la $a1, y
strcpy:
add $s0,$zero,$zero # $s0 = i = 0
L1:
add $t1,$s0,$a1 # $t1 = $s0 + $a1 = i + y[0]
 # = address of y[i]
lb $t2,0($t1) # $t2 = value at $t1 = y[i]
add $t3,$s0,$a0 # $t3 = $s0 + $a0 = i + x[0]
 # = address of x[i]
sb $t2,0($t3) # x[i]= $t2 = y[i]
beq $t2,$zero,end_of_strcpy # if y[i] == 0, exit
nop
addi $s0,$s0,1 # $s0 = $s0 + 1 <-> i = i + 1
j L1 # next character
nop
end_of_strcpy:

#print x <-> $a0 to check
li $v0, 4
#la $a0, x
syscall
```



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


Reset: reset completed.

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

➔ Chương trình chạy đúng mong đợi với kết quả in ra – x khớp với y



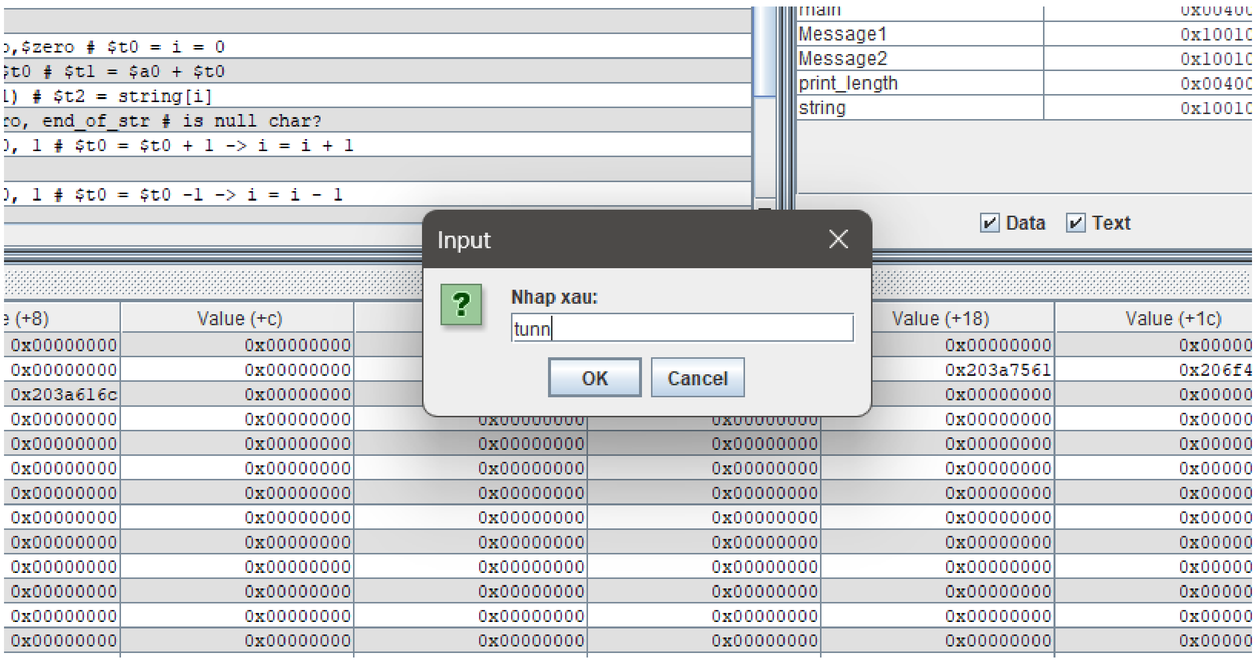
Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000004
$v1	3	0x00000000
$a0	4	0x10010000
$a1	5	0x10010020
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x10010025
$t2	10	0x00000000
$t3	11	0x10010005
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000005
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400040
hi		0x00000000
lo		0x00000000

Assignment 4

```
#Laboratory Exercise 5, Home Assignment 3
.data
string: .space 50
Message1: .asciiz "Nhap xau: "
Message2: .asciiz "Do dai xau la: "

.text
main:
get_string:

#Input string from dialog
        li $v0, 54
        la $a0, Message1
        la $a1, string
        la $a2, 50
        syscall
get_length:
        la $a0,string # $a0 = address(string[0])
        add $t0,$zero,$zero # $t0 = i = 0
check_char:
        add $t1,$a0,$t0 # $t1 = $a0 + $t0
                        # = address(string[i])
        lb $t2, 0($t1) # $t2 = string[i]
        beq $t2, $zero, end_of_str # is null char?
        addi $t0, $t0, 1 # $t0 = $t0 + 1 -> i = i + 1
        j check_char
end_of_str:
end_of_get_length:
        subi $t0, $t0, 1 # $t0 = $t0 -1 -> i = i - 1
print_length:
        li $v0, 56
        la $a0, Message2
        move $a1, $t0
        syscall
exit:
        li $v0, 10
```

- Giả sử nhập xâu "tunn" có độ dài là 4 chữ cái



```
),$zero # $t0 = i = 0
$t0 # $t1 = $a0 + $t0
1) # $t2 = string[i]
ro, end_of_str # is null char?
), 1 # $t0 = $t0 + 1 -> i = i + 1

), 1 # $t0 = $t0 -1 -> i = i - 1
```

main	0x0040
Message1	0x1001
Message2	0x1001
print_length	0x0040
string	0x1001

☑ Data ☑ Text

Input

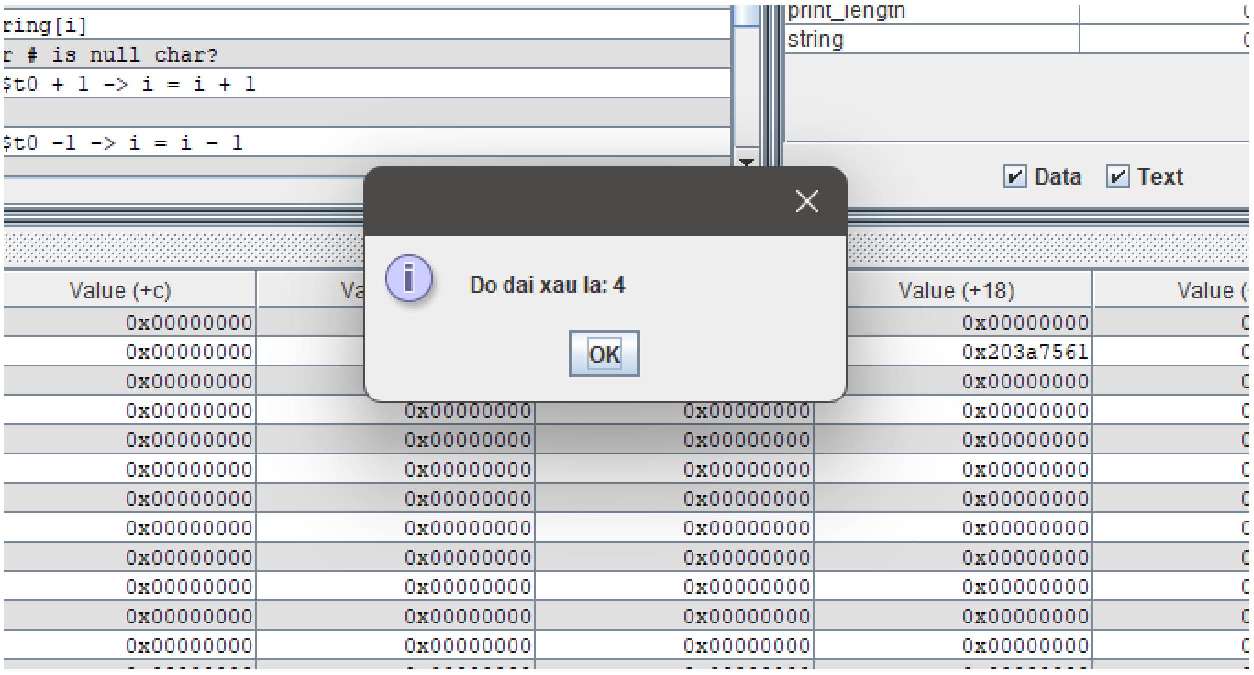
Nhap xau:

tunn

OK Cancel

(+8)	Value (+c)			Value (+18)	Value (+1c)
0x00000000	0x00000000			0x00000000	0x00000
0x00000000	0x00000000			0x203a7561	0x206f4
0x203a616c	0x00000000			0x00000000	0x00000
0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000
0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000
0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000
0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000
0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000
0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000
0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000
0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000
0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000
0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000

- Message Dialog trả về độ dài của xâu là 4



```
ring[i]
r # is null char?
$t0 + 1 -> i = i + 1

$t0 -1 -> i = i - 1
```

print_length	
string	

☑ Data ☑ Text

Do dai xau la: 4

OK

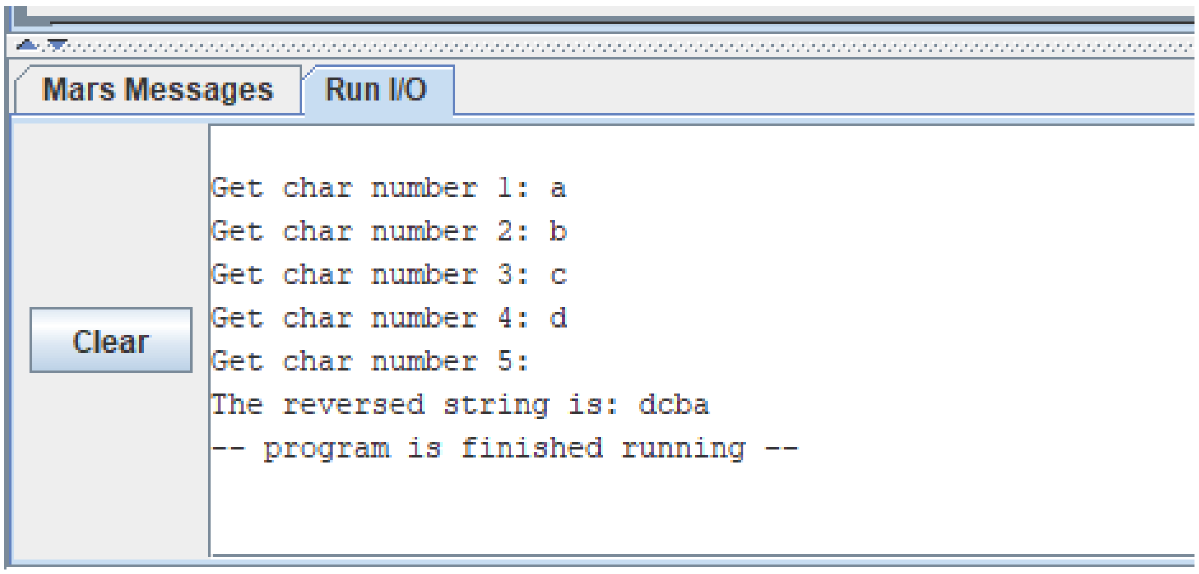
Value (+c)	Va		Value (+18)	Value (
0x00000000			0x00000000	
0x00000000			0x203a7561	
0x00000000			0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	

➔ Chương trình chạy đúng kết quả mong đợi

Assignment 5

```
.data
string: .space 20
mess1: .asciiz "Get char number "
mess2: .asciiz ": "
mess3: .asciiz "The reversed string is: "
endline: .asciiz "\n"

.text
li $s0, 0 # i = 0
li $s1, 20 # maximum chars
li $s2, 10 # char "\n"
la $s3, string # address of string[0]

read_char:
        beq $s0, $s1, end_read_char # if i = 20, exit
        # Print "Get char number "
        li $v0, 4
        la $a0, mess1
        syscall

        # Print i
        li $v0, 1
        addi $t1, $s0, 1
        move $a0, $t1
        syscall

        # Print ": "
        li $v0, 4
        la $a0, mess2
        syscall

        # Read char
        li $v0, 12 # $v0 is storing the input char
        syscall
        move $t1, $v0 # move to $t1
        beq $v0, $s2, end_read_char # if char = "\n", exit

        # Print "\n"
        li $v0, 4
        la $a0, endline
        syscall

        # Store char in string[i]
        add $t0, $s3, $s0 # $t0 = *string[i] = $s3 + $s0 = *string[0] +i
        sb $t1, 0($t0)# store the input char at *string[i]
        addi $s0, $s0, 1 # i = i + 1
        j read_char
end_read_char:

# Print "The reversed string is: "
li $v0, 4
la $a0, mess3
syscall

# Print string: for(n->0)
print_string:
        li $v0, 11
        lb $a0, 0($t0)
        syscall
        beq $t0, $s3, end_print_string
        subi $t0, $t0, 1
        j print_string
end_print_string:

# Exit
li $v0, 10
syscall
```

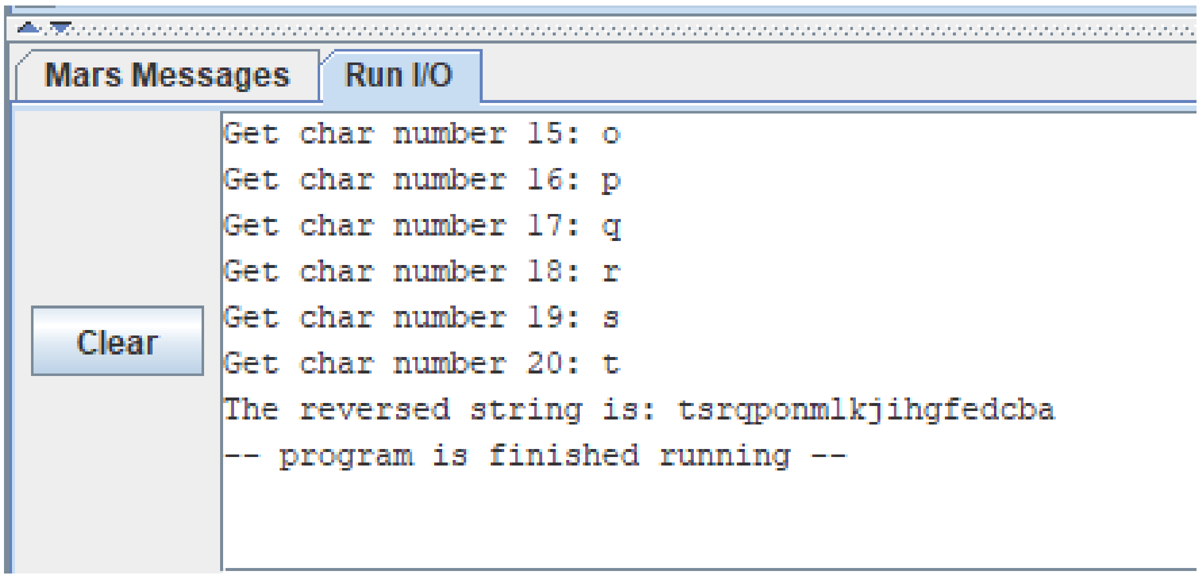
- Giả sử nhập xâu “abcd”, xâu có độ dài < 20, kết quả mong đợi sẽ là “dcba”



```
Get char number 1: a
Get char number 2: b
Get char number 3: c
Get char number 4: d
Get char number 5:
The reversed string is: dcba
-- program is finished running --
```

➔ Chương trình chạy đúng kết quả mong đợi

- Giả sử nhập xâu “abcdefghijklmnopqrst”, xâu có độ dài = 20, kết quả mong đợi sẽ là “tsrqponmlkjihgfedcba”



```
Get char number 15: o
Get char number 16: p
Get char number 17: q
Get char number 18: r
Get char number 19: s
Get char number 20: t
The reversed string is: tsrqponmlkjihgfedcba
-- program is finished running --
```

➔ Chương trình chạy đúng kết quả mong đợi