

# Báo cáo thực hành KTMT tuần 5

Lê Quốc Đảng

20225801

## Assignment 1

| Data Segment |             |             |             |             |             |             |             |             |
|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Address      | Value (+0)  | Value (+4)  | Value (+8)  | Value (+c)  | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
| 0x10010000   | 1 l e H     | o w o       | \0 d 1 r    | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x10010020   | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x10010040   | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x10010060   | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x10010080   | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x100100a0   | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x100100c0   | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |

Xâu được chia thành các nhóm 4 byte, và thứ tự lưu lần lượt từ phải qua trái

- Chạy chương trình

-Thanh ghi \$v0: 0x00000000 -> 0x00000004

-Thanh ghi \$at: 0x00000000 -> 0x10010000

-Thanh ghi \$a0: 0x00000000 -> 0x10010000

-Giải thích: khi gọi lệnh syscall, HDH sẽ xem giá trị của \$v0. Vì giá trị thanh ghi \$v0 = 4 => thanh ghi \$a0 sẽ là 1 string và in ra màn hình.

-Sau đó in ra “Hello world”

The screenshot shows the MARS 4.5 assembly debugger interface. The assembly window displays the following code:

```
mips1.asm
test:    .text
        .bgez    $at, 0($zero)
        li      $v0, 4
        lui    $a0, 1($at)
        ori    $a0, $a0, 0x00000000
        syscall
```

The Registers window shows the following register values:

| 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     | 0x7fffffe0 |
| \$fp   | 30     | 0x00000000 |
| \$ra   | 31     | 0x00000000 |

The Data Segment window shows the memory dump starting at address 0x10010000. The first few bytes are 1 l e H, followed by several groups of four zeros. The Messages window shows the output "Hello World" and the message "program is finished running (dropped off bottom) --".

## Assignment 2

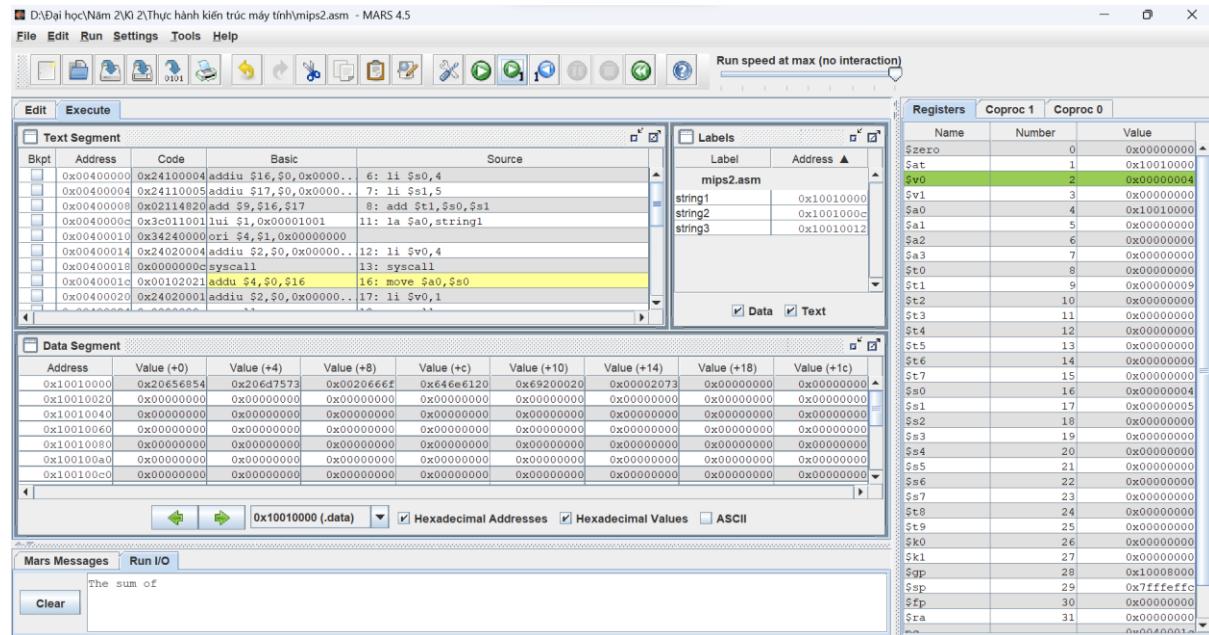
- Code:

```
.data
string1: .asciiz "The sum of "
string2: .asciiz " and "
string3: .asciiz " is "
.text
li $s0,4
li $s1,5
add $t1,$s0,$s1
#Print string1
la $a0,string1
li $v0,4
syscall
#Print $s0
move $a0,$s0
li $v0,1
syscall
#Print string2
la $a0,string2
li $v0,4
syscall
#Print $s1
move $a0,$s1
li $v0,1
syscall
#Print string3
la $a0,string3
li $v0,4
syscall
```

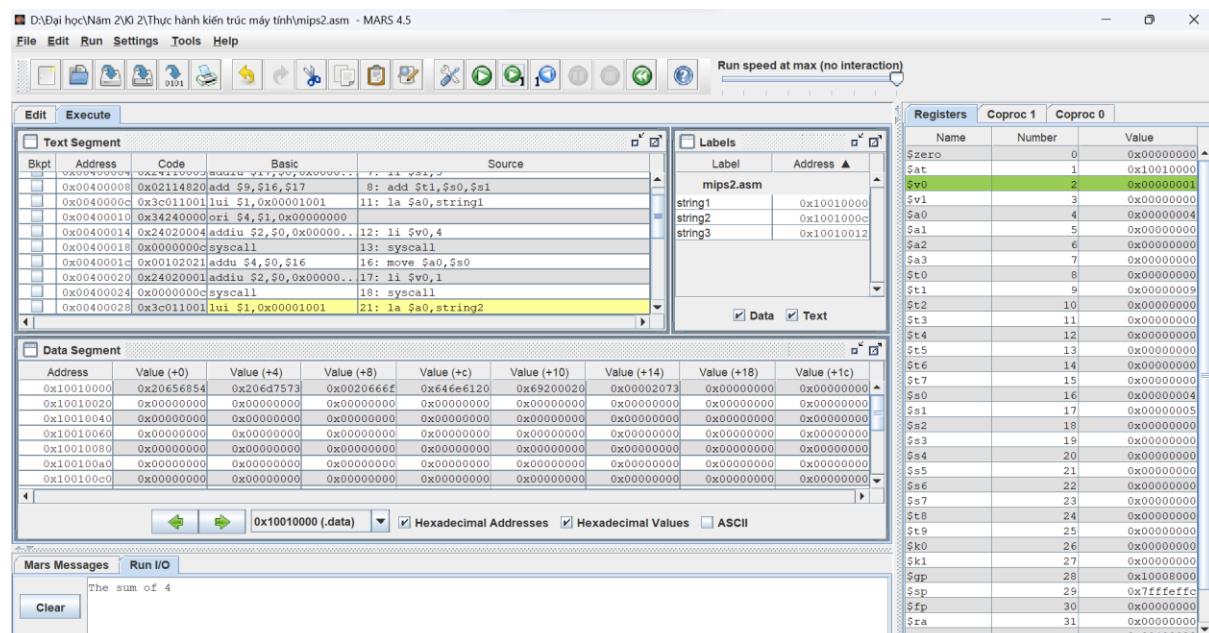
```
#Print $t1
move $a0,$t1
li $v0,1
syscall
```

- Thực thi chương trình

-In ra màn hình “The sum of”. \$v0 = 4 => thanh ghi \$a0 sẽ là 1 string và in ra màn hình



-In ra màn hình \$s0. \$v0 = 1 => thanh ghi \$a0 sẽ là 1 số và in ra màn hình.



-In ra màn hình “and”

$\$v0 = 4 \Rightarrow$  thanh ghi  $\$a0$  sẽ là 1 string và in ra màn hình.

D:\Đại học\Năm 2\Kì 2\Thực hành kiến trúc máy tính\mips2.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Registers Coproc 1 Coproc 0

| Name        | Number | Value             |
|-------------|--------|-------------------|
| \$zero      | 0      | 0x00000000        |
| \$at        | 1      | 0x10010000        |
| <b>\$v0</b> | 2      | <b>0x00000004</b> |
| \$v1        | 3      | 0x00000000        |
| \$a0        | 4      | 0x1001000c        |
| \$a1        | 5      | 0x00000000        |
| \$a2        | 6      | 0x00000000        |
| \$a3        | 7      | 0x00000000        |
| \$t0        | 8      | 0x00000000        |
| \$t1        | 9      | 0x00000009        |
| \$t2        | 10     | 0x00000000        |
| \$t3        | 11     | 0x00000000        |
| \$t4        | 12     | 0x00000000        |
| \$t5        | 13     | 0x00000000        |
| \$t6        | 14     | 0x00000000        |
| \$t7        | 15     | 0x00000000        |
| \$s0        | 16     | 0x00000004        |
| \$s1        | 17     | 0x00000005        |
| \$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     | 0x10000000        |
| \$sp        | 29     | 0x7ffffefffc      |
| \$fp        | 30     | 0x00000000        |
| \$ra        | 31     | 0x00000000        |

Text Segment

Labels

Data Segment

Mars Messages Run I/O

The sum of 4 and

-In ra màn hình  $\$s1$

$\$v0 = 1 \Rightarrow$  thanh ghi  $\$a0$  sẽ là 1 số và in ra màn hình.

D:\Đại học\Năm 2\Kì 2\Thực hành kiến trúc máy tính\mips2.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Registers Coproc 1 Coproc 0

| Name        | Number | Value             |
|-------------|--------|-------------------|
| \$zero      | 0      | 0x00000000        |
| \$at        | 1      | 0x10010000        |
| <b>\$v0</b> | 2      | <b>0x00000001</b> |
| \$v1        | 3      | 0x00000000        |
| \$a0        | 4      | 0x00000005        |
| \$a1        | 5      | 0x00000000        |
| \$a2        | 6      | 0x00000000        |
| \$a3        | 7      | 0x00000000        |
| \$t0        | 8      | 0x00000000        |
| \$t1        | 9      | 0x00000009        |
| \$t2        | 10     | 0x00000000        |
| \$t3        | 11     | 0x00000000        |
| \$t4        | 12     | 0x00000000        |
| \$t5        | 13     | 0x00000000        |
| \$t6        | 14     | 0x00000000        |
| \$t7        | 15     | 0x00000000        |
| \$s0        | 16     | 0x00000004        |
| \$s1        | 17     | 0x00000005        |
| \$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     | 0x10000000        |
| \$sp        | 29     | 0x7ffffefffc      |
| \$fp        | 30     | 0x00000000        |
| \$ra        | 31     | 0x00000000        |

Text Segment

Labels

Data Segment

Mars Messages Run I/O

The sum of 4 and 5

-In ra màn hình “is”

$\$v0 = 4 \Rightarrow$  thanh ghi \$a0 sẽ là 1 string và in ra màn hình.

| Name        | Number | Value             |
|-------------|--------|-------------------|
| \$zero      | 0      | 0x00000000        |
| \$at        | 1      | 0x10010000        |
| <b>\$v0</b> | 2      | <b>0x00000004</b> |
| \$v1        | 3      | 0x00000000        |
| \$a0        | 4      | 0x10010012        |
| \$a1        | 5      | 0x00000000        |
| \$a2        | 6      | 0x00000000        |
| \$a3        | 7      | 0x00000000        |
| \$t0        | 8      | 0x00000000        |
| \$t1        | 9      | 0x00000009        |
| \$t2        | 10     | 0x00000000        |
| \$t3        | 11     | 0x00000000        |
| \$t4        | 12     | 0x00000000        |
| \$t5        | 13     | 0x00000000        |
| \$t6        | 14     | 0x00000000        |
| \$t7        | 15     | 0x00000000        |
| \$s0        | 16     | 0x00000004        |
| \$s1        | 17     | 0x00000005        |
| \$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     | 0x10000000        |
| \$sp        | 29     | 0x7ffffeffc       |
| \$fp        | 30     | 0x00000000        |
| \$ra        | 31     | 0x00000000        |

-In ra màn hình kết quả của phép cộng  $\$s0 + \$s1 = 4 + 5 = 9$

$\$v0 = 1 \Rightarrow$  thanh ghi \$a0 sẽ là 1 số và in ra màn hình.

| Name        | Number | Value             |
|-------------|--------|-------------------|
| \$zero      | 0      | 0x00000000        |
| \$at        | 1      | 0x10010000        |
| <b>\$v0</b> | 2      | <b>0x00000001</b> |
| \$v1        | 3      | 0x00000000        |
| \$a0        | 4      | 0x00000009        |
| \$a1        | 5      | 0x00000000        |
| \$a2        | 6      | 0x00000000        |
| \$a3        | 7      | 0x00000000        |
| \$t0        | 8      | 0x00000000        |
| \$t1        | 9      | 0x00000009        |
| \$t2        | 10     | 0x00000000        |
| \$t3        | 11     | 0x00000000        |
| \$t4        | 12     | 0x00000000        |
| \$t5        | 13     | 0x00000000        |
| \$t6        | 14     | 0x00000000        |
| \$t7        | 15     | 0x00000000        |
| \$s0        | 16     | 0x00000004        |
| \$s1        | 17     | 0x00000005        |
| \$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     | 0x10000000        |
| \$sp        | 29     | 0x7ffffeffc       |
| \$fp        | 30     | 0x00000000        |
| \$ra        | 31     | 0x00000000        |

⇒ Kết quả đúng

## Assignment 3

The screenshot shows the MARS 4.5 assembly editor interface. The top menu bar includes File, Edit, Run, Settings, Tools, and Help. The main window displays the assembly code for `mips3.asm`. The code implements the `strcpy` function using a loop. It initializes pointers `$t1` and `$t3`, loads source bytes from `y[i]` into `$t1`, and stores them into `x[i]`. The loop continues until `y[i]` is null. The Registers pane on the right shows the state of all registers after execution.

```

2 .data
3 x: .space 1000 # destination string x, empty
4 y: .asciz "Hello Le Dang" # source string y
5 .text
6 la $a1,y
7 la $a0,x
8 stropsy:
9 add $s0,$zero,$zero #s0 = i=0
10 L1:
11 add $t1,$s0,$a1 #t1 = s0 + a1 = i + y[0]
12 # = address of y[i]
13 lb $t2,0($t1) #t2 = value at t1 = y[i]
14 add $t3,$s0,$a0 #t3 = s0 + a0 = i + x[0]
15 # = address of x[i]
16 sb $t2,0($t3) #x[i]= t2 = y[i]
17 beq $t2,$zero,end_of_stropsy #if y[i]==0, exit
18 nop
19 addi $s0,$s0,1 #s0=s0 + 1 <-> i=i+1
20 j L1 #next character
21 nop
22 end_of_stropsy:
23

```

Line: 22 Column: 15  Show Line Numbers

Mars Messages Run I/O

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

| 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     | 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     | 0x10000000 |
| \$sp   | 29     | 0xffffffff |
| \$fp   | 30     | 0x00000000 |
| \$ra   | 31     | 0x00000000 |

## Giải thích

- Nạp địa chỉ của X, Y
- Thanh ghi `$t1` trả vào `Y[i]`, `$t3` trả vào `X[i]`
- Vòng lặp thực hiện load từng byte của `Y[i]` và rồi store từng byte vào `X[i]`

The screenshot shows the MARS 4.5 assembly editor interface. The top menu bar includes File, Edit, Run, Settings, Tools, and Help. The main window displays the assembly code for `mips3.asm`. The Registers pane on the right shows the state of all registers after execution. The Registers pane highlights register `$t3` which contains the value `0x10010000`.

| Name        | Number    | Value             |
|-------------|-----------|-------------------|
| \$zero      | 0         | 0x00000000        |
| \$at        | 1         | 0x10010000        |
| \$v0        | 2         | 0x00000000        |
| \$v1        | 3         | 0x00000000        |
| \$a0        | 4         | 0x10010000        |
| \$a1        | 5         | 0x100103e8        |
| \$a2        | 6         | 0x00000000        |
| \$a3        | 7         | 0x00000000        |
| \$t0        | 8         | 0x00000000        |
| \$t1        | 9         | 0x100103e5        |
| \$t2        | 10        | 0x00000000        |
| <b>\$t3</b> | <b>11</b> | <b>0x10010000</b> |
| \$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        | 0x10000000        |
| \$sp        | 29        | 0xffffffff        |
| \$fp        | 30        | 0x00000000        |
| \$ra        | 31        | 0x00000000        |

⇒ Kết quả đúng

## Assignment 4

- Code

```
.data
    string: .space 50
    Mess1: .asciiz "Nhap xau: "
    Mess2: .asciiz "Do dai xau la: "

.text
main:
get_string:
    li $v0, 54
    la $a0, Mess1
    la $a1, string
    la $a2, 100
    syscall

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_length:
print_length_of_string:
    addi $t0, $t0, -1
    li $v0, 56
    la $a0, Mess2
```

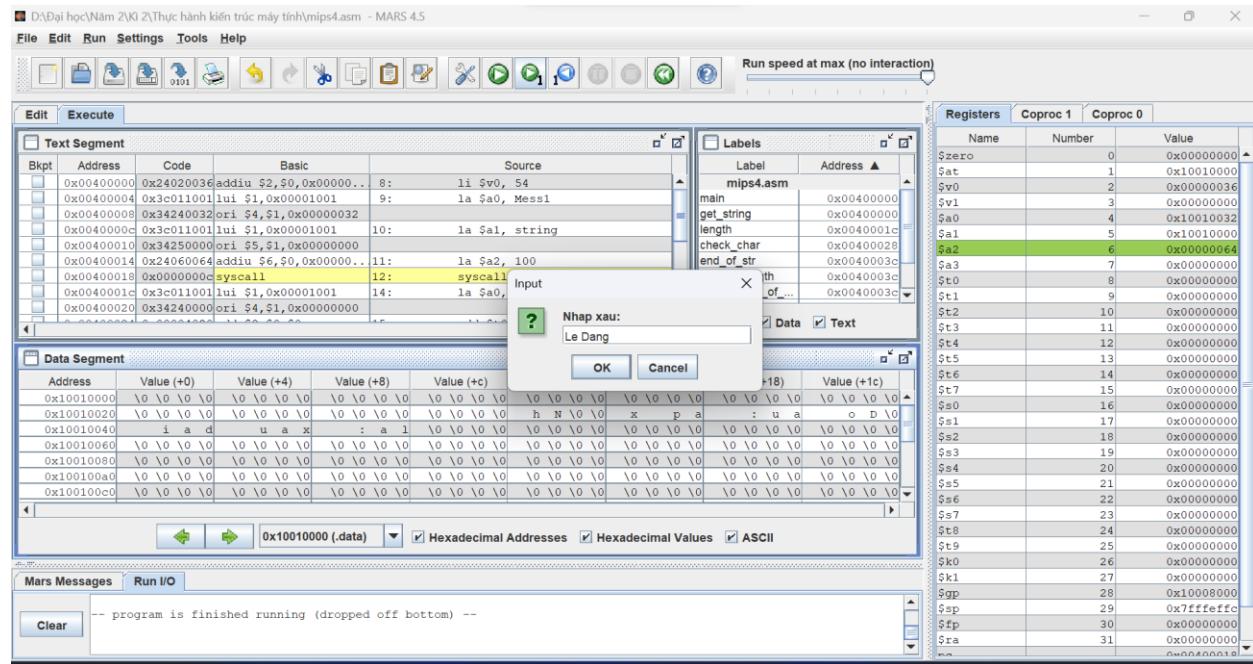
```

move $a1, $t0
syscall

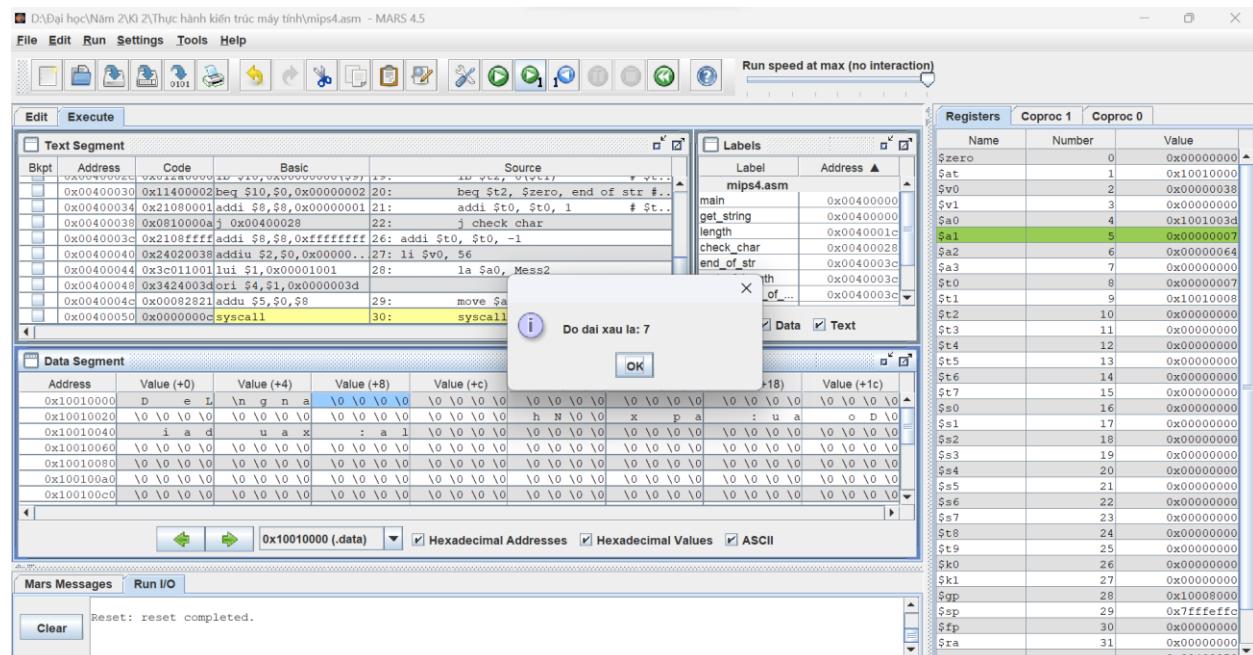
```

- Chạy chương trình

-Nhập xâu. \$v0 = 54 => nhập xâu vào hộp thoại và \$a1 là 1 string



-In ra độ dài xâu. \$v0 = 56 => hiển thị hộp thoại xâu



⇒ Kết quả đúng

## Assignment 5

- Code

```
# Dao nguoc str va luu vao reverse_str

.data

str: .space 50

reverse_str: .space 50

mess: .asciiz "Chuoi dao nguoc la: "

.text

    la    $a3, reverse_str #nap dia chi cua reverse_str

read_str:

    li    $v0,8

    la    $a0,str

    li    $a1,21 #20 ky tu

    syscall

reverse:

#Dem do dai xau str

    xor   $t3,$0,$0 #t3 = 0

    li    $t0,0 #t0 = i = 0

length:   add   $t1,$a0,$t0 #t1 = diachi str[i]

    lb    $t2,0($t1) #load byte t1 vao t2

    beq   $t2,$0,end_length #t2 = null -> ket thuc

    addi  $t0,$t0,1 #i++

    addi  $t3,$t3,1 #t3++

    j     length

end_length:

    subi $t3,$t3,1

    andi $t0,$t0,0 #reset lai bien i

change: add    $t1,$a0,$t3 #t1 = diachi str[i]

    lb    $t2,0($t1) #load byte t1 vao t2

    add   $t4,$a3,$t0 # t4 = dia chi reverse_str[i]
```

```

sb      $t2,0($t4) #luu t2 vao t4
beq    $t2,$0,end_change #t2=null -> end
addi   $t0,$t0,1 #i++
subi   $t3,$t3,1 #t3--
j     change

end_change:

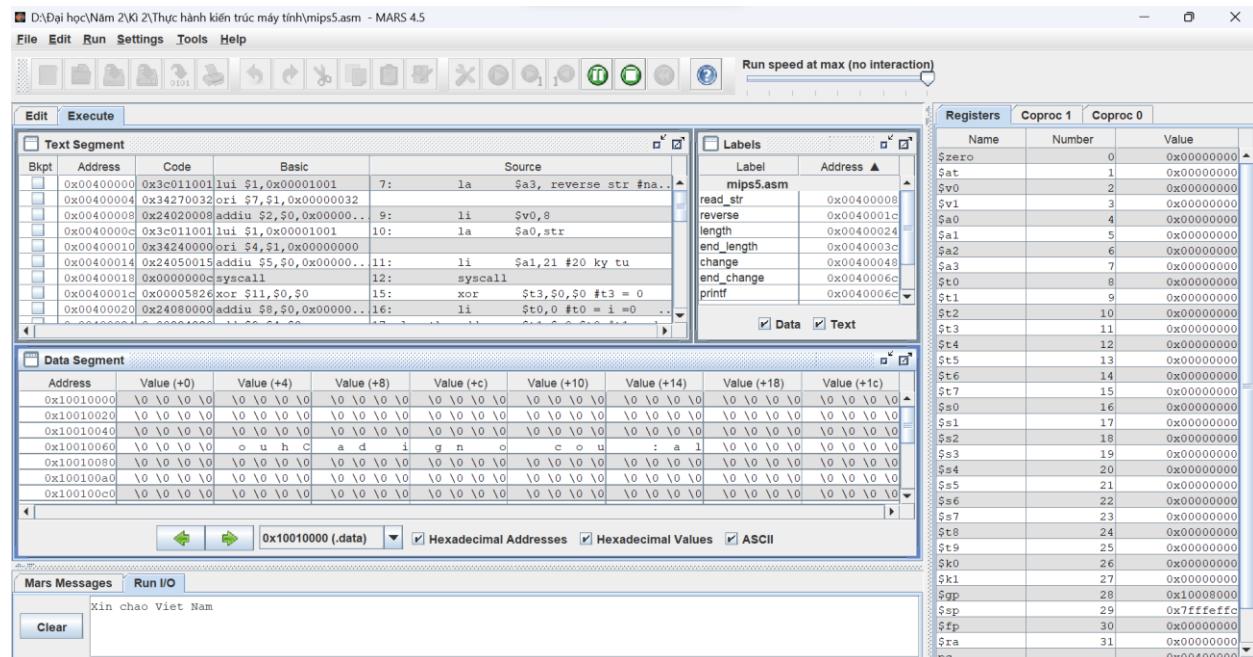
printf: # In xau reverse_str
li $v0,4
la $a0,mess
syscall

li $v0,4
la $a0, reverse_str
syscall

```

- Chạy chương trình

-Nhập string



-In ra string bị đảo ngược. \$v0 = 4 => in ra màn hình

D:\Đại học\Năm 2\K1 2\Thực hành kiến trúc máy tính\mips5.asm - MARS 4.5

**Registers**

| Name   | Number | Value      |
|--------|--------|------------|
| \$zero | 0      | 0x00000000 |
| \$at   | 1      | 0x10010000 |
| \$v0   | 2      | 0x00000004 |
| \$a0   | 4      | 0x10010032 |
| \$a1   | 5      | 0x00000015 |
| \$a2   | 6      | 0x00000000 |
| \$a3   | 7      | 0x10010032 |
| \$t0   | 8      | 0x00000012 |
| \$t1   | 9      | 0xffffffff |
| \$t2   | 10     | 0x00000000 |
| \$t3   | 11     | 0xffffffff |
| \$t4   | 12     | 0x10010044 |
| \$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     | 0x10000000 |
| \$sp   | 29     | 0xffffffff |
| \$fp   | 30     | 0x00000000 |
| \$ra   | 31     | 0x00000000 |

**Labels**

| Label      | Address    |
|------------|------------|
| mips5.asm  | 0x00400008 |
| read_str   | 0x00400008 |
| reverse    | 0x0040001c |
| length     | 0x00400024 |
| end_length | 0x0040003c |
| change     | 0x00400048 |
| end_change | 0x0040006c |
| printf     | 0x0040006c |

**Text Segment**

| Bkpt | Address    | Code       | Basic                    | Source                        |
|------|------------|------------|--------------------------|-------------------------------|
|      | 0x00400000 | 0x3c011001 | lui \$1,0x000001001      | 7: la \$a3, reverse str #na.. |
|      | 0x00400004 | 0x34270032 | ori \$7,\$1,0x00000032   |                               |
|      | 0x00400008 | 0x24020008 | addiu \$2,\$0,0x000000.. | 9: li \$v0,0                  |
|      | 0x0040000c | 0x3c011001 | lui \$1,0x000001001      | 10: la \$a0,str               |
|      | 0x00400010 | 0x34240000 | ori \$4,\$1,0x00000000   |                               |
|      | 0x00400014 | 0x24050015 | addiu \$5,\$0,0x00000..  | 11: li \$a1,21 #20 ky tu      |
|      | 0x00400018 | 0x0000000c | syscall                  | 12: syscall                   |
|      | 0x0040001c | 0x00000582 | xor \$11,\$0,\$0         | 15: xor \$t3,\$0,\$0 #t3 = 0  |
|      | 0x00400020 | 0x24080000 | addiu \$8,\$0,0x00000..  | 16: li \$t0,0 #t0 = i = 0     |

**Data Segment**

| Address    | Value (+0)  | Value (+4)  | Value (+8)  | Value (+c)  | Value (+10) | Value (+14) | Value (+18) | Value (+c)    |
|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|---------------|
| 0x10010000 | n i x       | o a h c     | e i v       | a N t       | \0 \0 \n m  | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0   |
| 0x10010020 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | t N a       | v i e c h a o |
| 0x10010040 | x i n       | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0   |
| 0x10010060 | \0 \0 \0 \0 | \0 \0 \0 \0 | o u h c     | a d i g n o | c o u :     | a l         | \0 \0 \0 \0 | \0 \0 \0 \0   |
| 0x10010080 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0   |
| 0x100100a0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0   |
| 0x100100c0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0   |

**Mars Messages**

```
Xin chao Viet Nam
Chuoi dao nguoc la:
maN teiV cahc niX
```

⇒ Kết quả đúng