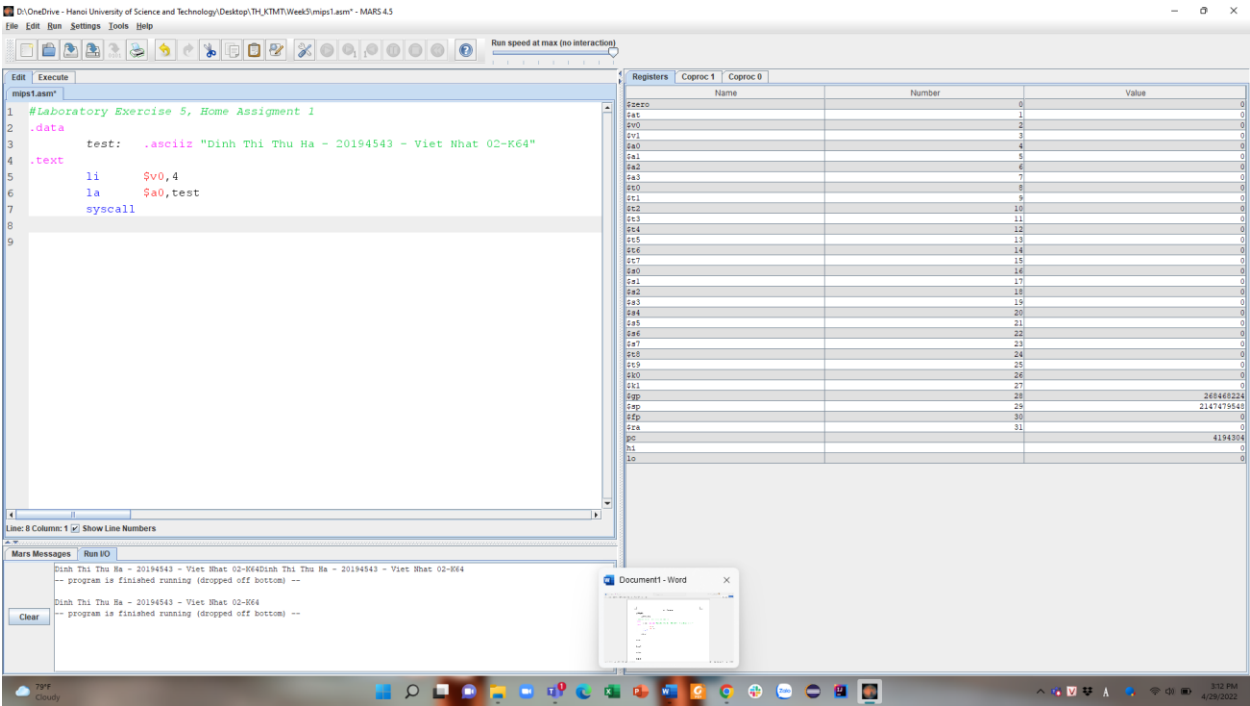


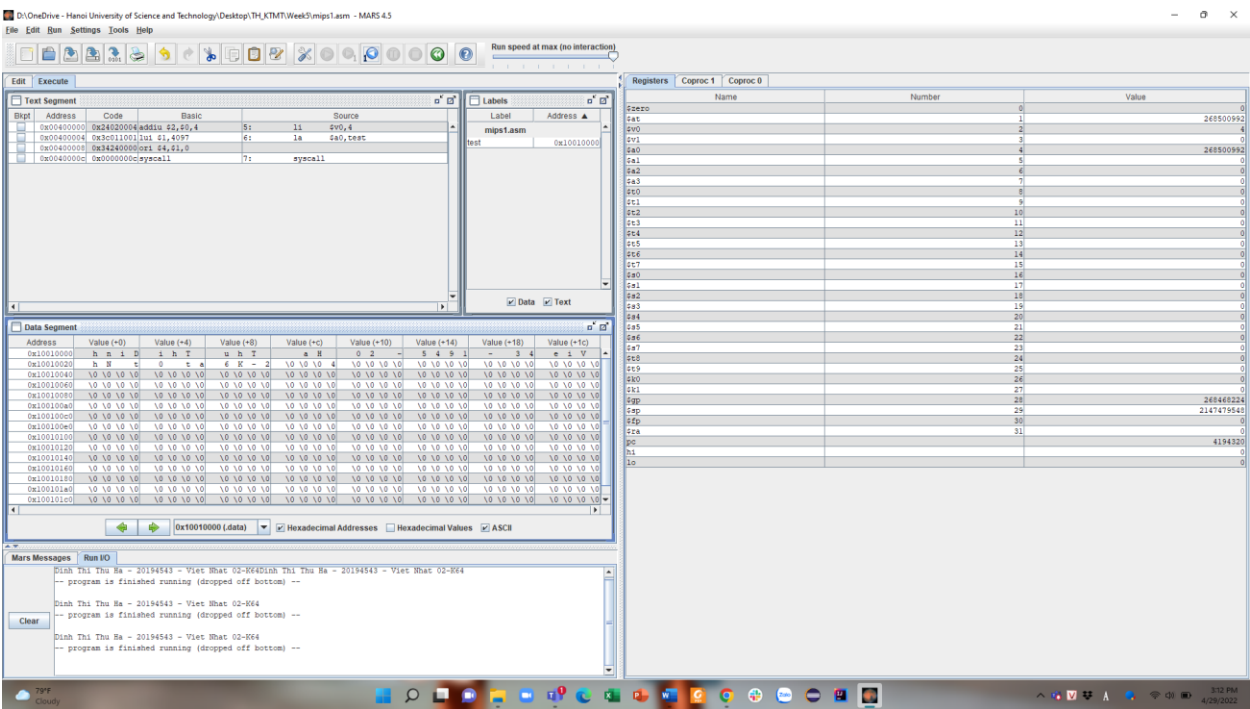
# BÁO CÁO TUẦN 5

## EXERCISE1:

- Chương trình:



- Kết quả:



## Bước 1:

D:\OneDrive - Hanoi University of Science and Technology\Desktop\TH\_KTMT\Week9\mpis1.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

**Edit Execute**

**Text Segment**

Offset	Address	Code	Basic	Source
0x00400000	0x24020004	addiu \$2,\$0,4	5:	li \$v0,4
0x00400004	0x3c011001	jal \$1,4097	6:	ja \$a0,\$test
0x00400008	0x34240000	ori \$4,\$1,0		
0x0040000c	0x0000000c	syscall	7:	syscall

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+C)	Value (+10)	Value (+14)	Value (+18)	Value (+1C)
0x10010000	h	a	i	d	s	h	t	
0x10010004	0	0	0	0	0	0	0	0
0x10010008	0	0	0	0	0	0	0	0
0x1001000c	0	0	0	0	0	0	0	0
0x10010010	0	0	0	0	0	0	0	0
0x10010014	0	0	0	0	0	0	0	0
0x10010018	0	0	0	0	0	0	0	0
0x1001001c	0	0	0	0	0	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010024	0	0	0	0	0	0	0	0
0x10010028	0	0	0	0	0	0	0	0
0x1001002c	0	0	0	0	0	0	0	0
0x10010030	0	0	0	0	0	0	0	0
0x10010034	0	0	0	0	0	0	0	0
0x10010038	0	0	0	0	0	0	0	0
0x1001003c	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010044	0	0	0	0	0	0	0	0
0x10010048	0	0	0	0	0	0	0	0
0x1001004c	0	0	0	0	0	0	0	0
0x10010050	0	0	0	0	0	0	0	0
0x10010054	0	0	0	0	0	0	0	0
0x10010058	0	0	0	0	0	0	0	0
0x1001005c	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010064	0	0	0	0	0	0	0	0
0x10010068	0	0	0	0	0	0	0	0
0x1001006c	0	0	0	0	0	0	0	0
0x10010070	0	0	0	0	0	0	0	0
0x10010074	0	0	0	0	0	0	0	0
0x10010078	0	0	0	0	0	0	0	0
0x1001007c	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x10010084	0	0	0	0	0	0	0	0
0x10010088	0	0	0	0	0	0	0	0
0x1001008c	0	0	0	0	0	0	0	0
0x10010090	0	0	0	0	0	0	0	0
0x10010094	0	0	0	0	0	0	0	0
0x10010098	0	0	0	0	0	0	0	0
0x1001009c	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100a4	0	0	0	0	0	0	0	0
0x100100a8	0	0	0	0	0	0	0	0
0x100100ac	0	0	0	0	0	0	0	0
0x100100b0	0	0	0	0	0	0	0	0
0x100100b4	0	0	0	0	0	0	0	0
0x100100b8	0	0	0	0	0	0	0	0
0x100100bc	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

**Registers** Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	4
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$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
\$d0	26	0
\$d1	27	0
\$fp	28	268469224
\$sp	29	2147479544
\$gp	30	0
\$ra	31	0
\$pc		4194384
\$lo		0

**Mars Messages** Run IO

Disasm This Ra - 20194543 - Viet What 02-04050b This Ra - 20194543 - Viet What 02-044  
-- program is finished running (dropped off bottom) --

Disasm This Ra - 20194543 - Viet What 02-044  
-- program is finished running (dropped off bottom) --

Disasm This Ra - 20194543 - Viet What 02-044  
-- program is finished running (dropped off bottom) --

Disasm This Ra - 20194543 - Viet What 02-044  
-- program is finished running (dropped off bottom) --

79°F Cloudy

3:13 PM 4/28/2022

## Bước 2:

D:\OneDrive - Hanoi University of Science and Technology\Desktop\TH\_KTMT\Week9\mpis1.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

**Edit Execute**

**Text Segment**

Offset	Address	Code	Basic	Source
0x00400000	0x24020004	addiu \$2,\$0,4	5:	li \$v0,4
0x00400004	0x3c011001	jal \$1,4097	6:	ja \$a0,\$test
0x00400008	0x34240000	ori \$4,\$1,0		
0x0040000c	0x0000000c	syscall	7:	syscall

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+C)	Value (+10)	Value (+14)	Value (+18)	Value (+1C)
0x10010000	h	a	i	d	s	h	t	
0x10010004	0	0	0	0	0	0	0	0
0x10010008	0	0	0	0	0	0	0	0
0x1001000c	0	0	0	0	0	0	0	0
0x10010010	0	0	0	0	0	0	0	0
0x10010014	0	0	0	0	0	0	0	0
0x10010018	0	0	0	0	0	0	0	0
0x1001001c	0	0	0	0	0	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010024	0	0	0	0	0	0	0	0
0x10010028	0	0	0	0	0	0	0	0
0x1001002c	0	0	0	0	0	0	0	0
0x10010030	0	0	0	0	0	0	0	0
0x10010034	0	0	0	0	0	0	0	0
0x10010038	0	0	0	0	0	0	0	0
0x1001003c	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010044	0	0	0	0	0	0	0	0
0x10010048	0	0	0	0	0	0	0	0
0x1001004c	0	0	0	0	0	0	0	0
0x10010050	0	0	0	0	0	0	0	0
0x10010054	0	0	0	0	0	0	0	0
0x10010058	0	0	0	0	0	0	0	0
0x1001005c	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010064	0	0	0	0	0	0	0	0
0x10010068	0	0	0	0	0	0	0	0
0x1001006c	0	0	0	0	0	0	0	0
0x10010070	0	0	0	0	0	0	0	0
0x10010074	0	0	0	0	0	0	0	0
0x10010078	0	0	0	0	0	0	0	0
0x1001007c	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x10010084	0	0	0	0	0	0	0	0
0x10010088	0	0	0	0	0	0	0	0
0x1001008c	0	0	0	0	0	0	0	0
0x10010090	0	0	0	0	0	0	0	0
0x10010094	0	0	0	0	0	0	0	0
0x10010098	0	0	0	0	0	0	0	0
0x1001009c	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100a4	0	0	0	0	0	0	0	0
0x100100a8	0	0	0	0	0	0	0	0
0x100100ac	0	0	0	0	0	0	0	0
0x100100b0	0	0	0	0	0	0	0	0
0x100100b4	0	0	0	0	0	0	0	0
0x100100b8	0	0	0	0	0	0	0	0
0x100100bc	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

**Registers** Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	4
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$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
\$d0	26	0
\$d1	27	0
\$fp	28	268469224
\$sp	29	2147479544
\$gp	30	0
\$ra	31	0
\$pc		4194384
\$lo		0

**Mars Messages** Run IO

Disasm This Ra - 20194543 - Viet What 02-04050b This Ra - 20194543 - Viet What 02-044  
-- program is finished running (dropped off bottom) --

Disasm This Ra - 20194543 - Viet What 02-044  
-- program is finished running (dropped off bottom) --

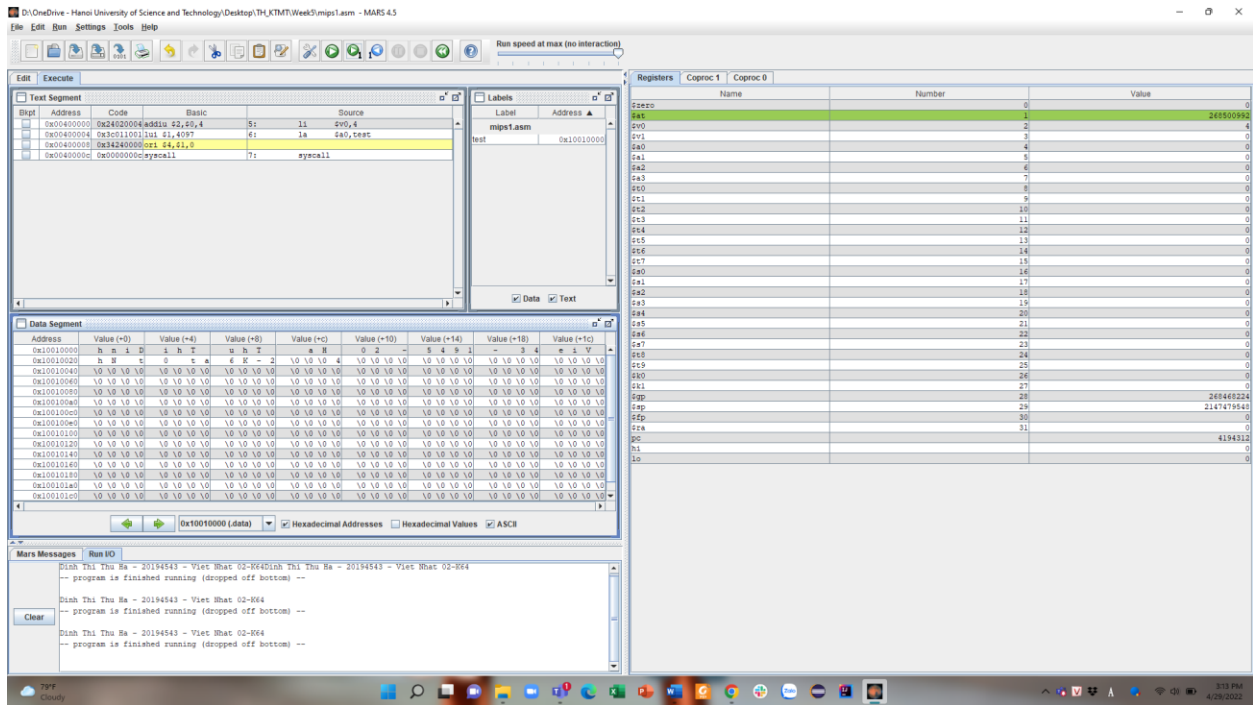
Disasm This Ra - 20194543 - Viet What 02-044  
-- program is finished running (dropped off bottom) --

Disasm This Ra - 20194543 - Viet What 02-044  
-- program is finished running (dropped off bottom) --

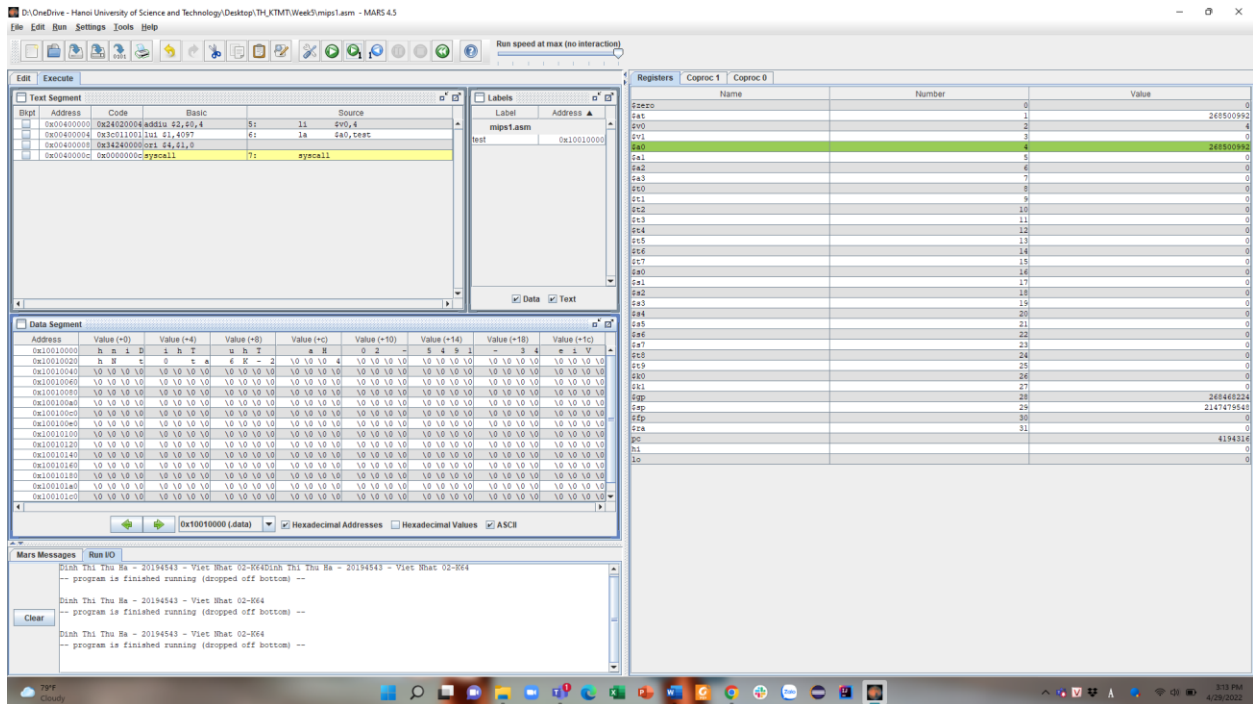
79°F Cloudy

3:13 PM 4/28/2022

## Bước 3:



## Bước 4:



- Giải thích:
  - + Dòng 3: Khai báo chuỗi kí tự "Dinh Thi Thu Ha - 20194543 - Viet Nhat 02-K64" với nhãn test
  - + Dòng 5: Gán giá trị thanh ghi \$v0=4
  - + Dòng 6: Load địa chỉ của biến test vào thanh ghi \$a0
  - + Dòng 7: Syscall gọi hàm v0 sau đó in ra chuỗi kí tự (do thanh ghi v0 có giá trị bằng 4).

## Exercise 2:

### - Chương trình:

```
#Laboratory Exercise 5, same Home Assignment 1
.data
sum: .asciiz "The sum of "
andW: .asciiz " and "
is: .asciiz " is "
.text
addi $s0, $0, 1101
addi $s1, $0, 1509
add $t0, $s0, $s1

li $v0, 4          # print string
la $a0, sum        # print "The sum of "
syscall

li $v0, 1          # print number
la $a0, 0($s0)     # print s0
syscall

li $v0, 4          # print string
la $a0, andW       # print " and "
syscall

li $v0, 1          # print number
la $a0, 0($s1)     # print s1
syscall

li $v0, 4          # print string
la $a0, is         # print " is "
syscall

li $v0, 1          # print number
la $a0, 0($t0)     # print t0 = s0 + s1
syscall
```

### - Kết quả:

The screenshot shows the MARS MIPS simulator interface. The 'Registers' window on the right displays the values of the registers: \$s0 is 1101, \$s1 is 1509, and \$t0 is 2610. The 'Data Segment' window on the left shows the memory layout, with the sum of 1101 and 1509 stored at address 0x10010000. The 'Mars Messages' window at the bottom shows the output of the program: 'The sum of 1101 and 1509 is 2610'.

### - Giải thích:

+ Dòng 3,4,5: Khai báo các chuỗi kí tự cùng nhãn tương ứng

- + Dòng 7,8: Gán giá trị tương ứng vào từng thanh ghi \$s0, \$s1
- + Dòng 9: Tính tổng giá trị của 2 thanh ghi \$s0, \$s1 sau đó lưu vào thanh ghi \$t0
- + Dòng 11: Gán thanh ghi v0 = 4( 4 là lệnh print string)
- + Dòng 12: Load địa chỉ của sum vào thanh ghi \$a0
- + Dòng 13: Syscall gọi hàm v0 in ra “The sum of ”
- + Dòng 15: Gán thanh ghi v0 = 1(1 là lệnh print integer number)
- + Dòng 16: Load địa chỉ của s0 vào thanh ghi a0
- + Dòng 17: Syscall gọi hàm v0 in ra ”1101”
- + Dòng 19->25: Tương tự dòng 9->17(thay sum bằng andW, \$s0 bằng \$s1) in ra “ and 1509”
- + Dòng 27->33: Tương tự dòng 9->17( thay sum bằng is, \$s0 bằng \$t0) in ra “ is 2610”

### Exercise 3:

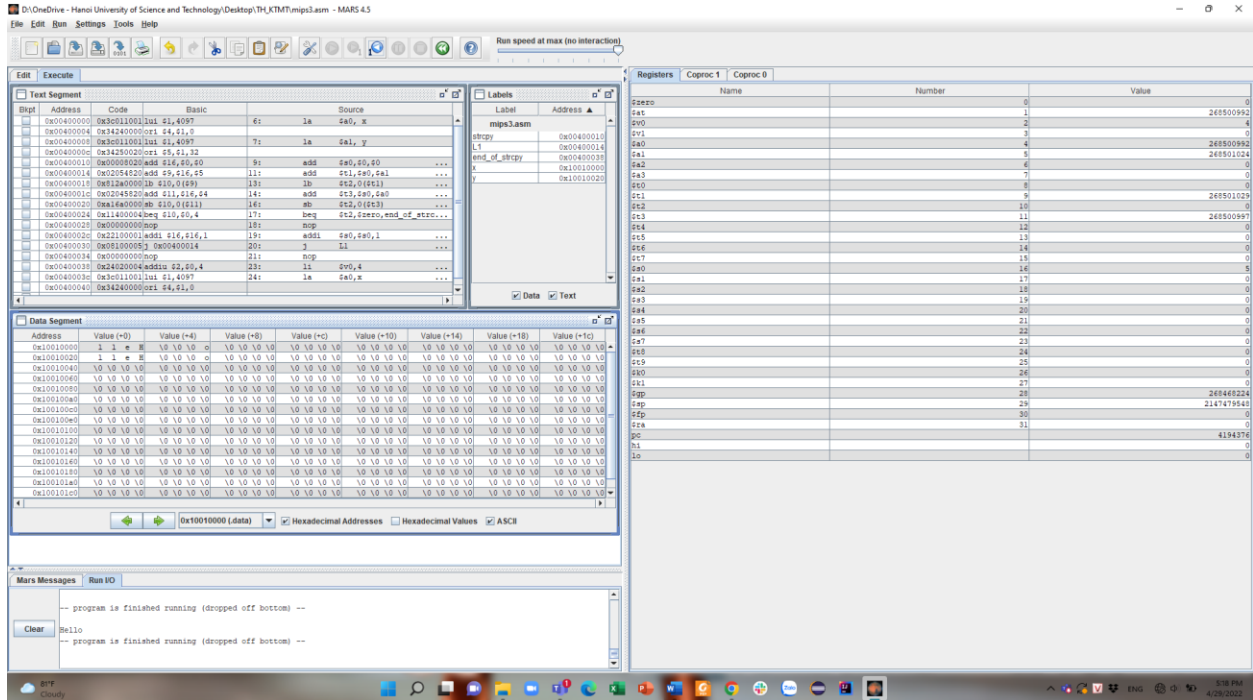
- Chương trình:

```

1 #Laboratory Exercise 5, Home Assignment 2
2 .data
3     x: .space 32          # destination string x, empty
4     y: .asciiz "Hello"    # source string y
5 .text
6     la $a0, x
7     la $a1, y
8 strcpy:
9     add $s0, $zero, $zero    # $s0=i=0
10
11 L1:
12     add $t1, $s0, $a1        # $t1=$s0+$a1=i+y[0]
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_strepy    #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_strepy:
23     li $v0, 4               #print string
24     la $a0, x               #print the sum of
25

```

- Kết quả:



- Giải thích:
  - + Dòng 3: Khai báo chuỗi x rỗng
  - + Dòng 4: Khai báo chuỗi ký tự "Hello" với nhãn y
  - + Dòng 6: Load địa chỉ biến x vào thanh ghi \$a0
  - + Dòng 7: Load địa chỉ biến y vào thanh ghi \$a1
  - + Dòng 9: Khai báo biến chạy \$s0 = i (= 0)
  - + Dòng 11: \$t1 = \$s0 + \$a1 = địa chỉ của y[i]
  - + Dòng 13: Load t1 vào t2 (địa chỉ của y[i])
  - + Dòng 14: Gán biến chạy t0 bằng địa chỉ của x[i]
  - + Dòng 16: Lưu địa chỉ của t2 vào t3
  - + Dòng 17: Nếu t2 = 0 => đã đọc hết chuỗi ký tự y => đến nhãn endOfStrcpy (bằng beq)
  - + Dòng 19: s0 = s0 + 1 tăng biến chạy i lên 1
  - + Dòng 20: Nhảy lại L1
  - + Dòng 22: Nhãn endOfStrcpy
  - + Dòng 23: Gán thanh ghi \$v0 = 4 (4 là lệnh print string)
  - + Dòng 24: Load địa chỉ biến x vào thanh ghi a0
  - + Dòng 25: Syscall gọi hàm v0 in "Hello" ra màn hình

## Exercise 4:

- Chương trình:

```
#Laboratory Exercise 5, Home Assignment 3
.data
    string: .space 111           # destination string x, empty
    MessageDia1: .asciiz "Input String: "
    MessageDia2: .asciiz "String length: "
.text
main:
    get_string:
        li $v0, 54
        la $a0, MessageDia1
        la $a1, string
        la $a2, 128
        syscall

    get_length:
        la $a0, string           #$a0 = Address(string[0])
        add $v1, $0, $0          #$v1 = length = 0
        add $t0, $0, $0          #$t0 = 1

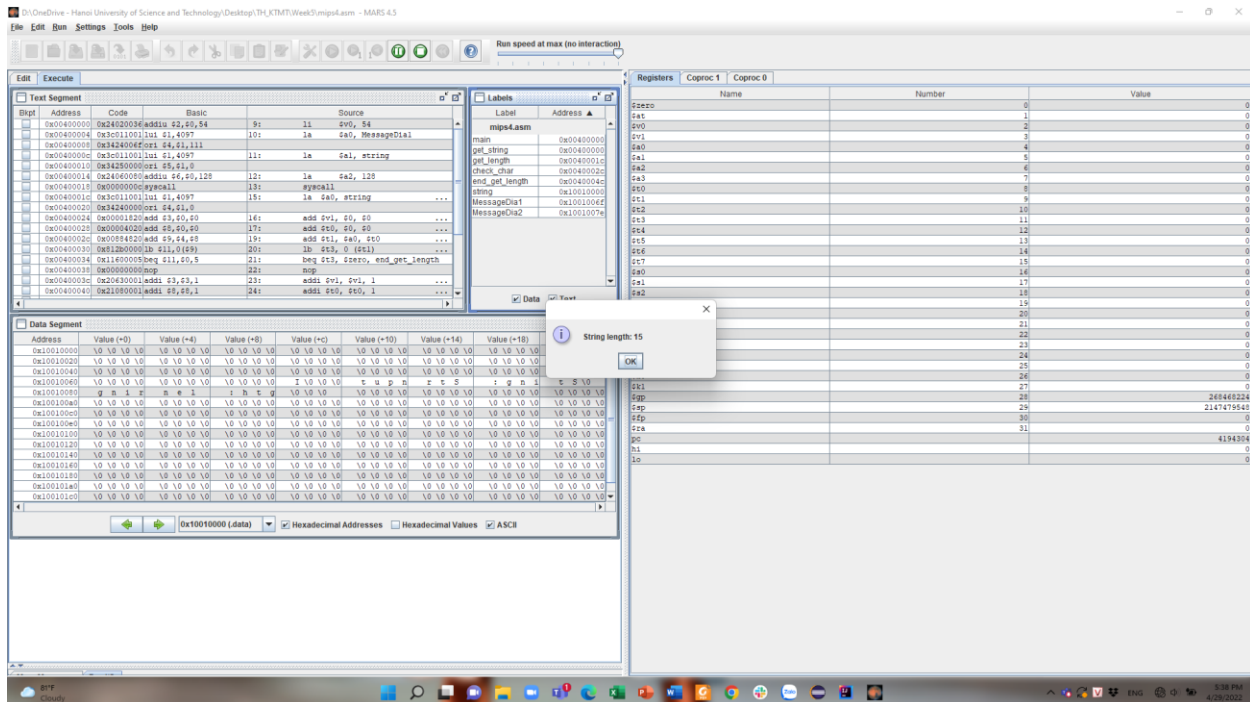
    check_char:
        add $t1, $a0, $t0        #$t1 = string[i]'s address
        lb $t3, 0($t1)           #$t3 = string[i]
        beq $t3, $zero, end_get_length
        nop
        addi $v1, $v1, 1          #$v1 = length ++
        addi $t0, $t0, 1          #$t0 = i ++
        j check_char
    end_get_length:
        li $v0, 56
        la $a0, MessageDia2      #print the string
        addi $v1, $v1, -1
        la $a1, 0($v1)           #print the length of string
        syscall
```

- Input string: "Dinh Thi Thu Ha"

Input String: Dinh Thi Thu Ha



## - Kết quả: String length: 15



⇒ Đúng độ dài xâu là 15.

- Giải thích:

- + Dòng 3: Khai báo chuỗi kí tự rỗng với nhãn string có độ dài 111
- + Dòng 4: Khai báo chuỗi ký tự "Input string: " với nhãn MessageDia1
- + Dòng 5: Khai báo chuỗi ký tự "String length: " với nhãn MessageDia2
- + Dòng 9: Gán thanh ghi v0 = 54 (vì 54 là hiện hộp thoại nhập chuỗi)
- + Dòng 10,11: Load địa chỉ biến Message1, string vào thanh ghi a0, a1
- + Dòng 12: Khai báo a2 có thể đọc tối đa 128 ký tự
- + Dòng 13: Syscall gọi hàm v0
- + Dòng 15: Load địa chỉ biến string vào thanh ghi \$a0
- + Dòng 16: Khởi tạo \$v1 = 0 lưu độ dài chuỗi ký tự
- + Dòng 17: Khởi tạo biến chạy (i) là \$t0 = 0
- + Dòng 19: Gán \$t1 bằng địa chỉ của string[i]
- + Dòng 20: \$t3 = \$t1 (= string[i])
- + Dòng 22: Check if \$t3 = 0 (đọc hết string) nhảy đến nhãn end\_get\_length
- + Dòng 24: v1=v1+1 tăng độ dài xâu lên 1
- + Dòng 25: t0=t0+1 tăng biến chạy i lên 1
- + Dòng 26: Jump đến nhãn check\_char
- + Dòng 30: Gán thanh ghi v0 = 56 (56 là lệnh hiển thị hộp thoại cho số nguyên)
- + Dòng 31: Load địa chỉ biến MessageDia2 vào thanh ghi \$a0
- + Dòng 32: \$v1 = \$v1 - 1
- + Dòng 33: Load giá trị của \$v1 gán vào thanh ghi \$a1
- + Dòng 34: Syscall gọi hàm v0 in "String length: " và số nguyên ra màn hình.



## Exercise 5:

- Chương trình:

```

1 .data
2     string: .space 50
3     message: .asciiz "\nChuoai dao nguoc:\n"
4 .text
5     addi $t0, $0, 21
6     la $s0, string
7 read_char:
8     li $v0, 12
9     syscall
10
11     addi $t0, $t0, -1
12     beq $v0, 10, end_read_char
13     beqz $t0, end_read_char
14
15     add $t1, $t0, $s0
16     sb $v0, 0($t1)
17
18     j read_char
19 end_read_char:
20     li $v0, 4
21     la $a0, message
22     syscall
23     li $v0, 4
24     la $a0, 0($t1)
25     syscall
26
27
28
29

```

Registers window (Coproc 1, Coproc 0):

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	4
\$v1	3	0
\$a0	4	269501005
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$a4	8	12
\$a5	9	269501005
\$a6	10	0
\$a7	11	0
\$t0	12	0
\$t1	13	0
\$t2	14	0
\$t3	15	0
\$t4	16	269500992
\$t5	17	0
\$t6	18	0
\$t7	19	0
\$t8	20	0
\$t9	21	0
\$s0	22	0
\$s1	23	0
\$s2	24	0
\$s3	25	0
\$s4	26	0
\$s5	27	0
\$s6	28	269469224
\$s7	29	2147479541
\$s8	30	0
\$s9	31	0
\$lo		4194384
\$hi		0

- Kết quả:  
+ Khi chuỗi kí tự <= 20

Text Segment:

Inst	Address	Code	Basic	Source
0x00000000	0x00000000	addi \$t0, \$0, 21	51	addi \$t0, \$0, 21
0x00000001	0x00000001	la \$s0, string	61	la \$s0, string
0x00000002	0x00000002	li \$v0, 12	81	li \$v0, 12
0x00000003	0x00000003	syscall	91	syscall
0x00000004	0x00000004	addi \$t0, \$t0, -1	111	addi \$t0, \$t0, -1
0x00000005	0x00000005	beq \$v0, 10, end_read_char	121	beq \$v0, 10, end_read_char
0x00000006	0x00000006	beqz \$t0, end_read_char	131	beqz \$t0, end_read_char
0x00000007	0x00000007	add \$t1, \$t0, \$s0	151	add \$t1, \$t0, \$s0
0x00000008	0x00000008	sb \$v0, 0(\$t1)	161	sb \$v0, 0(\$t1)
0x00000009	0x00000009	j read_char	181	j read_char
0x0000000A	0x0000000A	li \$v0, 4	201	li \$v0, 4
0x0000000B	0x0000000B	la \$a0, message	211	la \$a0, message
0x0000000C	0x0000000C	syscall	221	syscall

Data Segment:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+C)	Value (+10)	Value (+14)	Value (+18)	Value (+1C)
0x10010000	0	1791197456	1953002776	1952999712	1168843396	1800	0	0
0x10010004	0	0	0	0	1124720832	1769122332	1868634000	1969712672
0x10010008	171598703	0	0	0	0	0	0	0
0x1001000C	0	0	0	0	0	0	0	0
0x10010010	0	0	0	0	0	0	0	0
0x10010014	0	0	0	0	0	0	0	0
0x10010018	0	0	0	0	0	0	0	0
0x1001001C	0	0	0	0	0	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010024	0	0	0	0	0	0	0	0
0x10010028	0	0	0	0	0	0	0	0
0x1001002C	0	0	0	0	0	0	0	0
0x10010030	0	0	0	0	0	0	0	0
0x10010034	0	0	0	0	0	0	0	0
0x10010038	0	0	0	0	0	0	0	0
0x1001003C	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010044	0	0	0	0	0	0	0	0
0x10010048	0	0	0	0	0	0	0	0
0x1001004C	0	0	0	0	0	0	0	0
0x10010050	0	0	0	0	0	0	0	0

Registers window (Coproc 1, Coproc 0):

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	4
\$v1	3	0
\$a0	4	269500999
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$a4	8	5
\$a5	9	269500992
\$a6	10	0
\$a7	11	0
\$t0	12	0
\$t1	13	0
\$t2	14	0
\$t3	15	0
\$t4	16	0
\$t5	17	0
\$t6	18	0
\$t7	19	0
\$t8	20	0
\$t9	21	0
\$s0	22	0
\$s1	23	0
\$s2	24	0
\$s3	25	0
\$s4	26	0
\$s5	27	0
\$s6	28	269469224
\$s7	29	2147479541
\$s8	30	0
\$s9	31	0
\$lo		4194384
\$hi		0

+ Khi chuỗi ký tự >20 => Tự động ngưng

```
Chuoi dao nguoc:
ah
-- program is finished running (dropped off bottom) --

dinh thi thu ha xinh
Chuoi dao nguoc:
hnix ah uht iht hnid
-- program is finished running (dropped off bottom) --
```

- Giải thích:

- + Dòng 3: Khai báo chuỗi ký tự có độ dài 50
- + Dòng 4: Khai báo chuỗi ký tự "Chuoi dao nguoc:" với nhãn message
- + Dòng 6: khai báo \$t0 là chiều dài tối đa của string (ko vượt quá 20 ký tự)
- + Dòng 7: Load địa chỉ biến string vào thanh ghi \$s0
- + Dòng 9: Gán thanh ghi \$v0 = 12 (12 tức là lệnh đọc ký tự)
- + Dòng 10: Thực hiện hàm \$v0
- + Dòng 12: \$t0= \$t0-1
- + Dòng 13: If \$v0 == 10 => end\_read\_char
- + Dòng 14: If \$t0 == 0 (tức quá 20 ký tự) =>end\_read\_char
- + Dòng 16: Lsave địa chỉ ô ghi tiếp theo vào \$t1
- + Dòng 17: \$v0 = \$t1
- + Dòng 19: Jump đến read\_char
- + Dòng 21: Gán thanh ghi \$v0 = 4 (4 là lệnh in chuỗi)
- + Dòng 22: Load địa chỉ biến message vào thanh ghi a0
- + Dòng 23: Syscall gọi hàm \$v0 in chuỗi "Chuoi dao nguoc:" ra màn hình
- + Dòng 24: Gán thanh ghi \$v0 = 4 (4 là lệnh in chuỗi)
- + Dòng 25: Load địa chỉ thanh ghi \$t1 vào thanh ghi \$a0
- + Dòng 26: Syscall gọi hàm \$v0 in chuỗi đảo ngược ra màn hình.