

Báo cáo thực hành KTMT tuần 5

Họ và tên: Đỗ Gia Huy

MSSV: 20215060

Assignment 1

Code:

#Laboratory Exercise 5, Assignment 1

.data

test: .asciiz "Do Gia Huy la sinh vien lop KTMT Lab cua thay Le Ba Vui"

.text

li \$v0, 4 # \$v0 = 4

la \$a0, test # Dia chi cua test duoc ghi vao \$a0

syscall # Loi gọi dịch vụ hệ thống

Result:

The screenshot shows the MARS MIPS simulator interface. The Text Segment window displays the assembly code with addresses and comments. The Data Segment window shows memory addresses and values. The Registers window lists MIPS registers and their values. The MARS Messages window shows the output of the program.

Registers	Coproc 1	Coproc 0	Name	Number	Value
\$zero			0	0	0x00000000
\$at			1	1	0x10010000
\$v0			2	2	0x00000004
\$v1			3	3	0x00000000
\$a0			4	4	0x10010000
\$a1			5	5	0x00000000
\$a2			6	6	0x00000000
\$a3			7	7	0x00000000
\$t0			8	8	0x00000000
\$t1			9	9	0x00000000
\$t2			10	10	0x00000000
\$t3			11	11	0x00000000
\$t4			12	12	0x00000000
\$t5			13	13	0x00000000
\$t6			14	14	0x00000000
\$t7			15	15	0x00000000
\$s0			16	16	0x00000000
\$s1			17	17	0x00000000
\$s2			18	18	0x00000000
\$s3			19	19	0x00000000
\$s4			20	20	0x00000000
\$s5			21	21	0x00000000
\$s6			22	22	0x00000000
\$s7			23	23	0x00000000
\$s8			24	24	0x00000000
\$s9			25	25	0x00000000
\$k0			26	26	0x00000000
\$k1			27	27	0x00000000
\$gp			28	28	0x10000000
\$sp			29	29	0x7fffffc0
\$fp			30	30	0x00000000
\$ra			31	31	0x00000000
pc					0x00400010
hi					0x00000000
lo					0x00000000

⇒ Kết quả trên đúng với lý thuyết

Assignment 2

Code:

#Laboratory Exercise 5, Assignment 2

.data

str1: .asciiz "The sum of "

str2: .asciiz " and "

str3: .asciiz " is "

.text

li \$s0, 4 # number1 = 4

li \$s1, 8 # number2 = 8

add \$t0, \$s0, \$s1 # \$t0 = Sum of 4 and 8

Print string "str1"

li \$v0, 4

la \$a0, str1

syscall

Print \$s0

li \$v0, 1

move \$a0, \$s0

syscall

Print string "str2"

li \$v0, 4

la \$a0, str2

syscall

Print \$s1

li \$v0, 1

move \$a0, \$s1

syscall

```
# Print string "str3"
```

```
li    $v0, 4
```

```
la    $a0, str3
```

```
syscall
```

```
# Print $t0
```

```
li    $v0, 1
```

```
move $a0, $t0
```

```
syscall
```

```
Exit: li    $v0, 10
```

```
syscall
```

Result:

- Bắt đầu thực hiện chương trình

The screenshot displays the Mars MIPS simulator interface. The top panel shows the assembly code with the following instructions:

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24100004	addiu \$t6,\$0,0x0000...	7: li \$s0, 4 # number1 = 4
	0x00400004	0x24110008	addiu \$t7,\$0,0x0000...	8: li \$s1, 8 # number2 = 8
	0x00400008	0x02114020	add \$t8,\$t6,\$t7	9: add \$t0, \$s0, \$s1 # \$t0 = Sum...
	0x0040000c	0x24020004	addiu \$t2,\$0,0x00000004	12: li \$v0, 4
	0x00400010	0x3c011001	lui \$l,0x00001001	13: la \$a0, str1
	0x00400014	0x34240000	ori \$4,\$l,0x00000000	
	0x00400018	0x0000000c	syscall	14: syscall
	0x0040001c	0x24020001	addiu \$t2,\$0,0x00000001	17: li \$v0, 1
	0x00400020	0x00102021	addu \$4,\$0,\$t6	18: move \$a0, \$s0
	0x00400024	0x0000000c	syscall	19: syscall

The bottom panel shows the memory dump (Data Segment) with the following values:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x20e5e854	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

The bottom panel also shows the "Mars Messages" and "Run I/O" buttons, and a "Clear" button.

- In xâu "The sum of"

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24100004	addiu \$16,\$0,0x0000...	7: li \$s0, 4 # number1 = 4
	0x00400004	0x24110008	addiu \$17,\$0,0x0000...	8: li \$s1, 8 # number2 = 8
	0x00400008	0x02114020	add \$8,\$16,\$17	9: add \$t0, \$s0, \$s1 # \$t0 = Sum...
	0x0040000c	0x24020004	addiu \$2,\$0,0x00000004	12: li \$v0, 4
	0x00400010	0x3c011001	lui \$1,0x00001001	13: la \$a0, str1
	0x00400014	0x34240000	ori \$4,\$1,0x00000000	
	0x00400018	0x0000000c	syscall	14: syscall
	0x0040001c	0x24020001	addiu \$2,\$0,0x00000001	17: li \$v0, 1
	0x00400020	0x00102021	addu \$4,\$0,\$16	18: move \$a0, \$s0
	0x00400024	0x0000000c	syscall	19: syscall

Labels

Label	Address
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

☒ Hexadecimal Addresses ☒ Hexadecimal Values ☐ ASCII

Mars Messages **Run I/O**

The sum of

- In số 4

Text Segment

Bkpt	Address	Code	Basic	Source
	0x0040000c	0x24020004	addiu \$2,\$0,0x00000004	12: li \$v0, 4
	0x00400010	0x3c011001	lui \$1,0x00001001	13: la \$a0, str1
	0x00400014	0x34240000	ori \$4,\$1,0x00000000	
	0x00400018	0x0000000c	syscall	14: syscall
	0x0040001c	0x24020001	addiu \$2,\$0,0x00000001	17: li \$v0, 1
	0x00400020	0x00102021	addu \$4,\$0,\$16	18: move \$a0, \$s0
	0x00400024	0x0000000c	syscall	19: syscall
	0x00400028	0x24020004	addiu \$2,\$0,0x00000004	22: li \$v0, 4
	0x0040002c	0x3c011001	lui \$1,0x00001001	23: la \$a0, str2
	0x00400030	0x34240000	ori \$4,\$1,0x00000000	

Labels

Label	Address
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

☒ Hexadecimal Addresses ☒ Hexadecimal Values ☐ ASCII

Mars Messages **Run I/O**

The sum of 4

- In xâu "and"

Assembly code window showing instructions for W5A2.asm:

Bkpt	Address	Code	Basic	Source
	0x00400014	0x34240000	ori \$4,\$1,0x00000000	
	0x00400018	0x0000000c	syscall	14: syscall
	0x0040001c	0x24020001	addiu \$2,\$0,0x00000001	17: li \$v0, 1
	0x00400020	0x00102021	addu \$4,\$0,\$16	18: move \$a0, \$s0
	0x00400024	0x0000000c	syscall	19: syscall
	0x00400028	0x24020004	addiu \$2,\$0,0x00000004	22: li \$v0, 4
	0x0040002c	0x3c011001	lui \$1,0x00001001	23: la \$a0, str2
	0x00400030	0x3424000c	ori \$4,\$1,0x0000000c	
	0x00400034	0x0000000c	syscall	24: syscall
	0x00400038	0x24020001	addiu \$2,\$0,0x00000001	27: li \$v0, 1

Labels window (W5A2.asm):

Label	Address
Exit	0x00400060
str1	0x10010000
str2	0x1001000c
str3	0x10010012

Data Segment window:

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

Mars Messages window:

The sum of 4 and

Clear

- In số 8

Assembly code window showing instructions for W5A2.asm:

Bkpt	Address	Code	Basic	Source
	0x00400020	0x00102021	addu \$4,\$0,\$16	18: move \$a0, \$s0
	0x00400024	0x0000000c	syscall	19: syscall
	0x00400028	0x24020004	addiu \$2,\$0,0x00000004	22: li \$v0, 4
	0x0040002c	0x3c011001	lui \$1,0x00001001	23: la \$a0, str2
	0x00400030	0x3424000c	ori \$4,\$1,0x0000000c	
	0x00400034	0x0000000c	syscall	24: syscall
	0x00400038	0x24020001	addiu \$2,\$0,0x00000001	27: li \$v0, 1
	0x0040003c	0x00112021	addu \$4,\$0,\$17	28: move \$a0, \$s1
	0x00400040	0x0000000c	syscall	29: syscall
	0x00400044	0x24020004	addiu \$2,\$0,0x00000004	32: li \$v0, 4

Labels window (W5A2.asm):

Label	Address
Exit	0x00400060
str1	0x10010000
str2	0x1001000c
str3	0x10010012

Data Segment window:

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

Mars Messages window:

The sum of 4 and 8

Clear

- In xâu "is"

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400030	0x3424000c	ori \$4,\$1,0x0000000c	
	0x00400034	0x0000000c	syscall	24: syscall
	0x00400038	0x24020001	addiu \$2,\$0,0x00000001	27: li \$v0, 1
	0x0040003c	0x00112021	addu \$4,\$0,\$17	28: move \$a0, \$s1
	0x00400040	0x0000000c	syscall	29: syscall
	0x00400044	0x24020004	addiu \$2,\$0,0x00000004	32: li \$v0, 4
	0x00400048	0x3c011001	lui \$1,0x00001001	33: la \$a0, str3
	0x0040004c	0x34240012	ori \$4,\$1,0x00000012	
	0x00400050	0x0000000c	syscall	34: syscall
	0x00400054	0x24020001	addiu \$2,\$0,0x00000001	37: li \$v0, 1

Labels

Label	Address
W5A2.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

☒ Hexadecimal Addresses ☒ Hexadecimal Values ☐ ASCII

Mars Messages

The sum of 4 and 8 is

- In kết quả ra màn hình, cụ thể $4 + 8 = 12$

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400040	0x0000000c	syscall	29: syscall
	0x00400044	0x24020004	addiu \$2,\$0,0x00000004	32: li \$v0, 4
	0x00400048	0x3c011001	lui \$1,0x00001001	33: la \$a0, str3
	0x0040004c	0x34240012	ori \$4,\$1,0x00000012	
	0x00400050	0x0000000c	syscall	34: syscall
	0x00400054	0x24020001	addiu \$2,\$0,0x00000001	37: li \$v0, 1
	0x00400058	0x00082021	addu \$4,\$0,\$8	38: move \$a0, \$t0
	0x0040005c	0x0000000c	syscall	39: syscall
	0x00400060	0x2402000a	addiu \$2,\$0,0x0000000a	41: Exit: li \$v0, 10
	0x00400064	0x0000000c	syscall	42: syscall

Labels

Label	Address
W5A2.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

☒ Hexadecimal Addresses ☒ Hexadecimal Values ☐ ASCII

Mars Messages

The sum of 4 and 8 is 12

⇒ Kết quả trên đúng với lý thuyết

Assignment 3

Code:

#Laboratory Exercise 5, Assignment 3

.data

```

x: .space 32                                # destination string x, empty

y: .asciiz "Hello"                          # source string y

.text

strcpy:   add    $s0,$zero,$zero            # $s0 = i = 0

         la      $a1,y                      # Load address of y to $a1

         la      $a0,x                      # Load address of x to $a0

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:

```

Result:

The screenshot displays the Mars MIPS simulator interface. The **Text Segment** window shows the assembly code for the `strcpy` function, with addresses and comments. The **Data Segment** window shows the memory layout, including the source string `Hello` and the destination buffer `x`. The **Registers** window shows the values of registers `$s0`, `$a0`, `$a1`, `$t1`, `$t2`, and `$t3`. The **Messages** window shows the program's output, indicating that the program is finished running.

⇒ Kết quả trên đúng với lý thuyết

Assignment 4

Code:

#Laboratory Exercise 5, Assignment 4

.data

string: .space 50

Message1: .asciiz "Nhap xau: "

Message2: .asciiz "Do dai xau la: "

.text

main:

```
get_string: li    $v0, 54          # Get a string from dialog
            la    $a0, Message1   # Load address of the Message1 to $a0
            la    $a1, string     # Load address of input buffer "string"
to $a1
            la    $a2, 50         # Maximum number of characters to
read = 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:


```

print_length:    addi $t0, $t0, -1

                li    $v0, 56

                la    $a0, Message2

                move $a1, $t0

                syscall

```

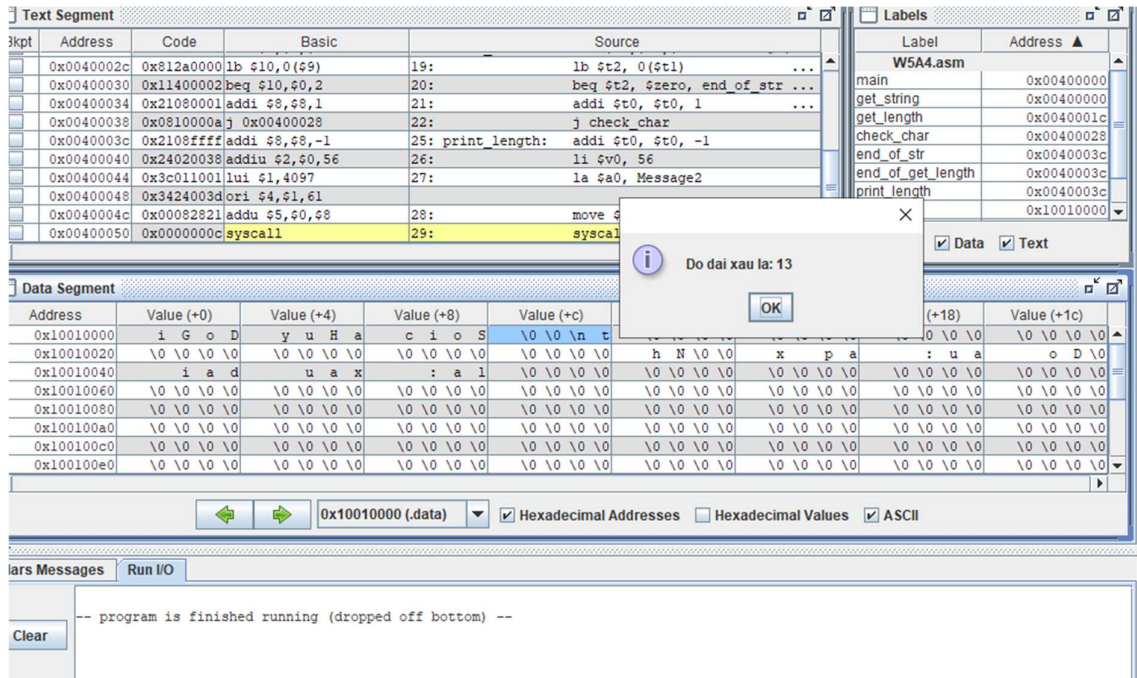
Result:

- Nhập xâu, giả sử nhập xâu “DoGiaHuySoict” :

The screenshot displays a MIPS simulator interface with the following components:

- Text Segment:** A table showing assembly instructions and their corresponding addresses. The instruction `syscall` at address `0x0040001c` is highlighted.
- Data Segment:** A table showing memory addresses and their values. The address `0x10010000` is selected, and the value is `\0 \0 \0 \0`.
- Input Dialog:** A small window titled "Nhập xâu:" (Enter string) with the text "DoGiaHuySoict" entered.
- Console Window:** A window at the bottom showing the message "-- program is finished running (dropped off bottom) --".

- Trả về độ dài ký tự xâu:



- ⇒ Nhận thấy xâu “DoGiaHuySoict” dài đúng 13 ký tự, ta có thể kết luận được rằng kết quả trên đúng với lý thuyết

Assignment 5

Code:

#Laboratory Exercise 5, Assignment 5

.data

get_char: .space 20

message1: .asciiz "Nhap ky tu thu "

message2: .asciiz ": "

message3: .asciiz "\n"

message4: .asciiz "Chuoi ky tu vua nhap (Bi dao nguc thu tu) la: "

.text

li \$s0, 20 # N = 20

li \$s1, 0 # i = 0

la \$s2, get_char # Load address of get_char[0]

```
li    $s3, 10                # Char \n in ASCII
```

```
read_char:
```

```
    beq  $s1, $s0, end_read_char # i = N branch to exit
```

```
    # Show message "Nhap ky tu thu i: "
```

```
    li   $v0, 4
```

```
    la   $a0, message1
```

```
    syscall
```

```
    addi $t1, $s1, 1
```

```
    li   $v0, 1
```

```
    move $a0, $t1
```

```
    syscall
```

```
    li   $v0, 4
```

```
    la   $a0, message2
```

```
    syscall
```

```
    li   $v0, 12    # Read character
```

```
    syscall
```

```
    move $t0, $v0
```

```
    beq  $t0, $s3, end_read_char # Press "Enter" branch to exit
```

```
    li   $v0, 4
```

```
    la   $a0, message3
```

```
    syscall
```

```

    add  $s5, $s2, $s1  # $s5 = Address of get_char[i] = get_char[0] + i
    sb   $t0, 0($s5)    # Store character to get_char[i]
    addi $s1, $s1, 1     # i++
    j     read_char

end_read_char:
    li    $v0, 4
    la    $a0, message4
    syscall

print_string:
    li    $v0, 11
    lb    $a0, 0($s5)
    syscall

    beq   $s5, $s2, exit
    addi  $s5, $s5, -1
    j     print_string

exit:
    li    $v0, 10
    syscall

```

Result:

- Trường hợp nhập đủ 20 ký tự. Giả sử xâu ta nhập là “abcdefghijklmnopqrst” :

The screenshot displays the Mars MIPS simulator interface. The top section shows the assembly code with columns for Bkpt, Address, Code, Basic, and Source. The right panel lists labels for the program 'W5A5.asm'. The middle section shows the data segment with memory addresses and their corresponding values in hexadecimal and ASCII. The bottom section shows the I/O messages, including the program's execution flow and user input.

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24100014	addiu \$16,\$0,20	10: li \$s0, 20
	0x00400004	0x24110000	addiu \$17,\$0,0	11: li \$s
	0x00400008	0x3c011001	lui \$1,4097	12: la \$s2, get_char
	0x0040000c	0x34320000	ori \$18,\$1,0	# L...
	0x00400010	0x2413000a	addiu \$19,\$0,10	13: li \$s3, 10
	0x00400014	0x12300018	beq \$17,\$16,24	16: beq \$s1, \$s0, end_read_char
	0x00400018	0x24020004	addiu \$2,\$0,4	19: li \$v0, 4
	0x0040001c	0x3c011001	lui \$1,4097	20: la \$a0, message1
	0x00400020	0x34240014	ori \$4,\$1,20	
	0x00400024	0x0000000c	syscall	21: syscall

Labels

Label	Address
W5A5.asm	
read_char	0x00400014
end_read_char	0x00400078
print_string	0x00400088
exit	0x004000a0
get_char	0x10010000
message1	0x10010014
message2	0x10010024
message3	0x10010027

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	d c b a	h g f e	l k j i	p o n m	t s r q	p a h n	y k	t u t
0x10010020	\0 \0 u h	\n \0 :	u h C \0	k i o	u t y	a u v	a h n	B (p
0x10010040	a d i	g n o	c o u	u h t) u t	: a l	\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
0x100100e0	\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

Run I/O

Clear

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

Nhap ky tu thu 1: a
Nhap ky tu thu 2: b

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

Nhap ky tu thu 1: a
Nhap ky tu thu 2: b
Nhap ky tu thu 3: c
Nhap ky tu thu 4: d
Nhap ky tu thu 5: e
Nhap ky tu thu 6: f
Nhap ky tu thu 7: g
Nhap ky tu thu 8: h
Nhap ky tu thu 9: i
Nhap ky tu thu 10: j
Nhap ky tu thu 11: k
Nhap ky tu thu 12: l
Nhap ky tu thu 13: m
Nhap ky tu thu 14: n
Nhap ky tu thu 15: o
Nhap ky tu thu 16: p
Nhap ky tu thu 17: q
Nhap ky tu thu 18: r
Nhap ky tu thu 19: s
Nhap ky tu thu 20: t
Chuoi ky tu vua nhap (Bi dao nguoc thu tu) la: tsrqponmlkjihgfedcba
-- program is finished running --
```

- Trường hợp không nhập đủ 20 ký tự, cụ thể là nhập “GiaHuySoict\n”:

The screenshot displays the Mars IDE interface with three main panels:

- Text Segment:** Shows assembly code for the program. The code reads 12 characters into a buffer and then prints them in reverse order. The final instruction is a system call to exit.
- Data Segment:** Shows the memory layout of the program. It includes a buffer for 12 characters, initialized with spaces, and a string for the prompt "Nhap ky tu thu 12:". The data is displayed in hexadecimal and ASCII formats.
- Mars Messages / Run I/O:** Shows the output of the program. It displays the prompt "Nhap ky tu thu 12:" and the user input "GiaHuySoict\n". The program then prints the input in reverse order: "tuoSuyHag".

The output shows that the program correctly handles the input "GiaHuySoict\n" and prints the reverse of the input, "tuoSuyHag", as expected.

⇒ Kết quả trên đúng với lý thuyết