

# BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH

## LAB 5

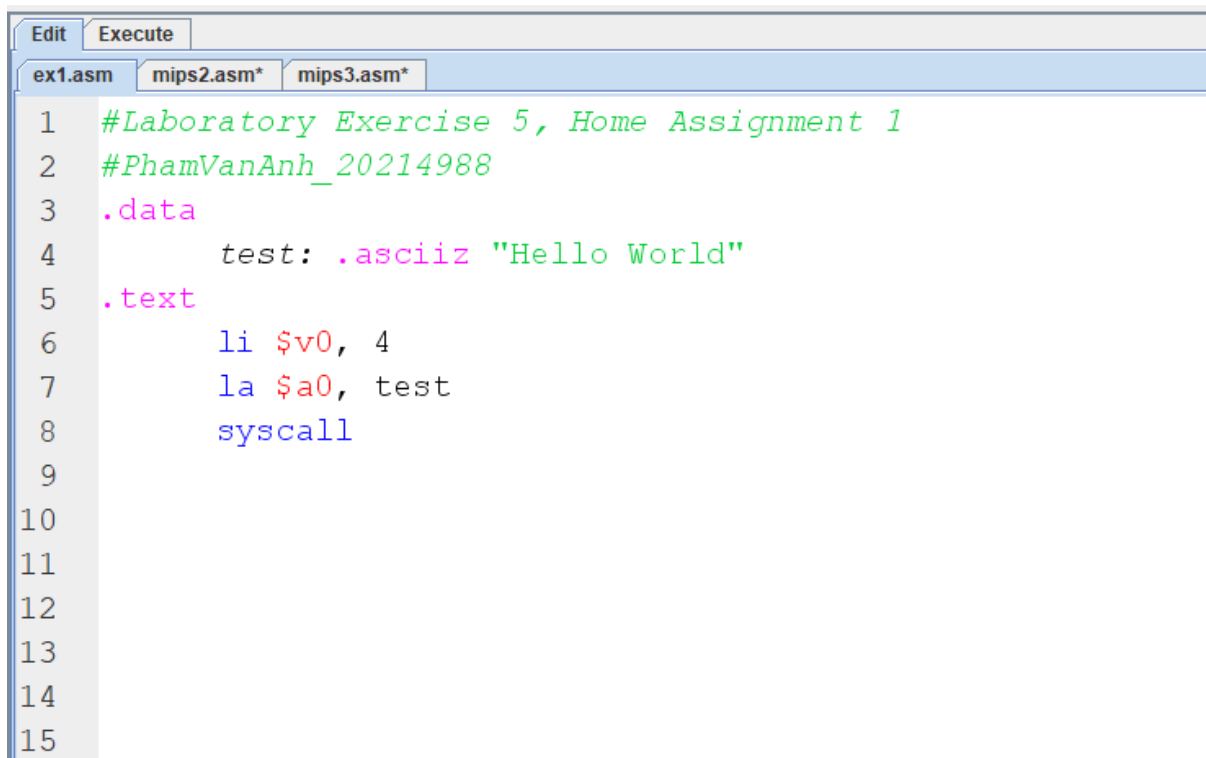
Họ và tên: Phạm Văn Anh

MSSV: 20214988

Mã lớp: 139365

### ASSIGNMENT 1:

#### 1. Code



The screenshot shows the MARS MIPS simulator with three tabs: 'ex1.asm', 'mips2.asm\*', and 'mips3.asm\*'. The 'ex1.asm' tab is active, displaying the following assembly code:

```
1  #Laboratory Exercise 5, Home Assignment 1
2  #PhamVanAnh_20214988
3  .data
4      test: .asciiz "Hello World"
5  .text
6      li $v0, 4
7      la $a0, test
8      syscall
9
10
11
12
13
14
15
```

Thực hiện gõ chương trình vào công cụ MARS

#### 2. Thực hiện dòng lệnh số 6:

- Gán giá trị  $\$v0 = 4 \rightarrow$  option của SYSCALL là PRINT STRING



The screenshot shows the MARS register window with a green background. The register  $\$v0$  is highlighted, and its value is 2. The register number 4 is also visible on the right side of the window.

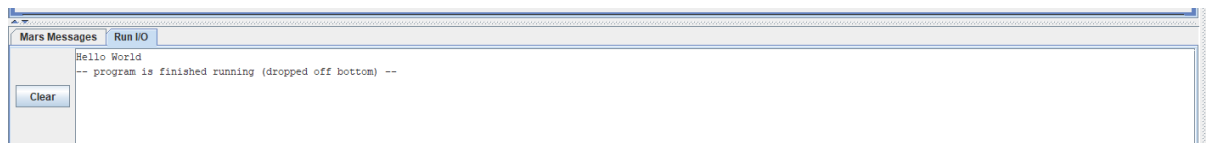
$\$v0$	2	4
--------	---	---

#### 3. Thực hiện dòng lệnh số 7:

- Gán \$a0 là giá trị địa chỉ của *test*:

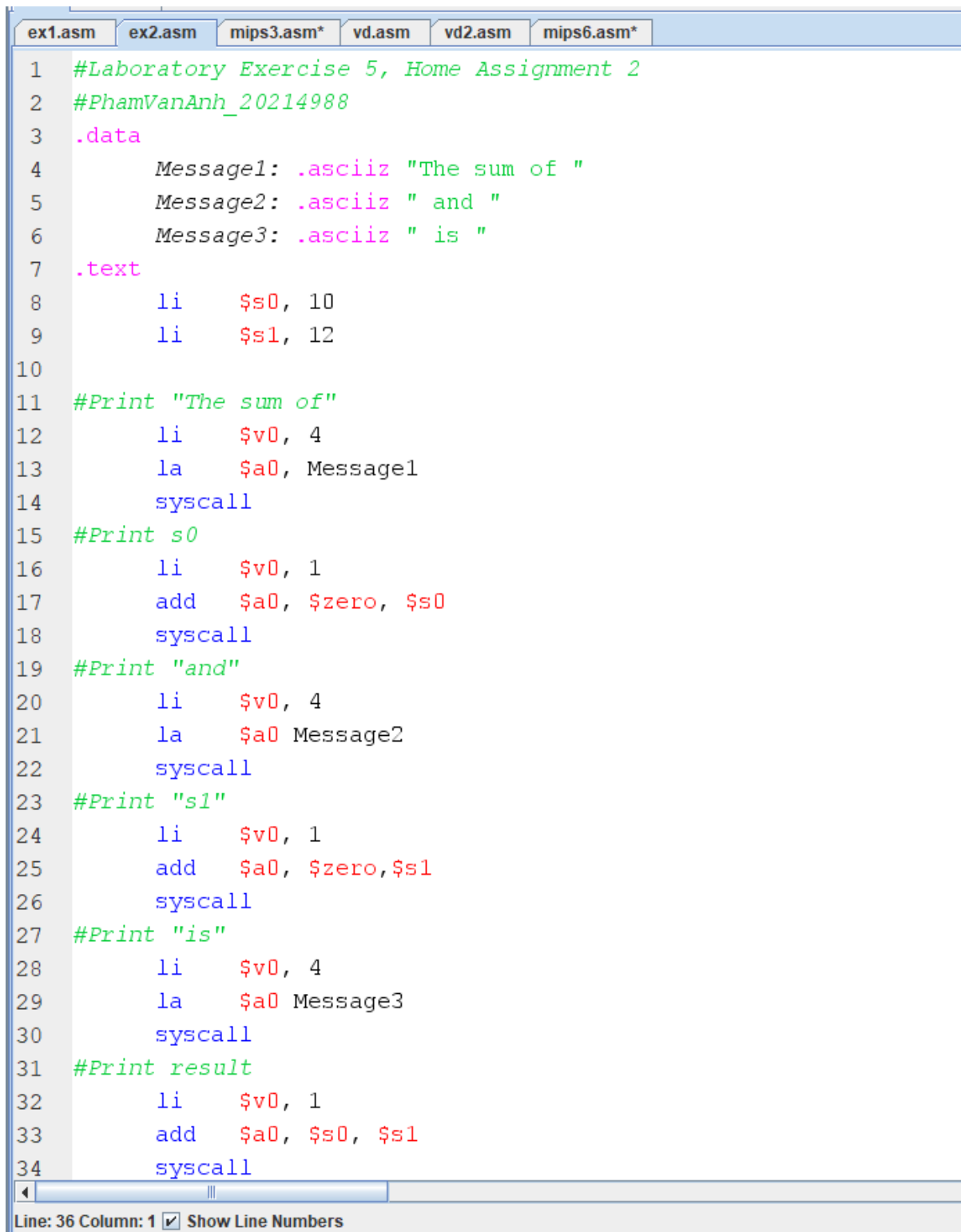
\$zero	0	0
\$at	1	268500992
\$v0	2	4
\$v1	3	0
\$a0	4	268500992

## 4. In chuỗi



## ASSIGNMENT 2:

### 1. Code



```
ex1.asm  ex2.asm  mips3.asm*  vd.asm  vd2.asm  mips6.asm*
1  #Laboratory Exercise 5, Home Assignment 2
2  #PhamVanAnh_20214988
3  .data
4      Message1: .asciiz "The sum of "
5      Message2: .asciiz " and "
6      Message3: .asciiz " is "
7  .text
8      li      $s0, 10
9      li      $s1, 12
10
11     #Print "The sum of"
12     li      $v0, 4
13     la      $a0, Message1
14     syscall
15     #Print s0
16     li      $v0, 1
17     add     $a0, $zero, $s0
18     syscall
19     #Print "and"
20     li      $v0, 4
21     la      $a0, Message2
22     syscall
23     #Print "s1"
24     li      $v0, 1
25     add     $a0, $zero, $s1
26     syscall
27     #Print "is"
28     li      $v0, 4
29     la      $a0, Message3
30     syscall
31     #Print result
32     li      $v0, 1
33     add     $a0, $s0, $s1
34     syscall
```

Line: 36 Column: 1 ☒ Show Line Numbers

Thực hiện gõ chương trình vào công cụ MARS

## 2. Thực hiện câu lệnh số 8 và 9

- Gán giá trị của \$s0 = 10, \$s1 = 12

\$s0	16	10
\$s1	17	12

## 3. Thực hiện câu lệnh số 12, 13, 14: In ra chuỗi “The sum of”

- Câu lệnh 12: Gán giá trị \$v0 = 4 → option của SYSCALL là *PRINT STRING*
- Gán địa chỉ của Message1 vào \$a0

\$a0	1	268500992
\$v0	2	4
\$v1	3	0
\$a0	4	268500992

## 4. Thực hiện câu lệnh số 16, 17, 18: In ra “s0”

- Câu lệnh 16: Gán giá trị của \$v0 = 1 → option của SYSCALL là *RINT DECIMAL INTEGER*
- Câu lệnh số 17:  $a0 = 0 + s0 = s0$

\$v0	2	1
\$v1	3	0
\$a0	4	10

## 5. Thực hiện câu lệnh số 20, 21, 22: In ra “and”

- Câu lệnh 20: Gán giá trị \$v0 = 4 → option của SYSCALL là *PRINT STRING*
- Gán địa chỉ của Message2 vào \$a0

\$v0	2	4
\$v1	3	0
\$a0	4	268501004

## 6. Thực hiện câu lệnh số 24, 25, 26: In ra “\$s1”

- Câu lệnh 24: Gán giá trị của \$v0 = 1 → option của SYSCALL là *RINT DECIMAL INTEGER*
- Câu lệnh số 25:  $a0 = 0 + s1 = s1$

\$v0	2	1
\$v1	3	0
\$a0	4	12

### 7. Thực hiện câu lệnh số 28, 29, 30: In ra “is”

- Câu lệnh 28: Gán giá trị \$v0 = 4 → option của SYSCALL là *PRINT STRING*
- Gán địa chỉ của Message3 vào \$a0

\$v0	2	4
\$v1	3	0
\$a0	4	268501010

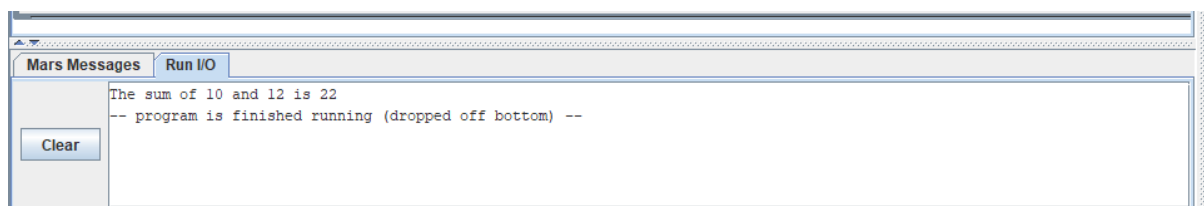
### 8. Thực hiện câu lệnh số 32, 33, 34: In ra “result”

- Câu lệnh 32: Gán giá trị của \$v0 = 1 → option của SYSCALL là *RINT DECIMAL INTEGER*
- Câu lệnh số 33:  $a0 = s0 + s1$

\$v0	2	1
\$v1	3	0
\$a0	4	22

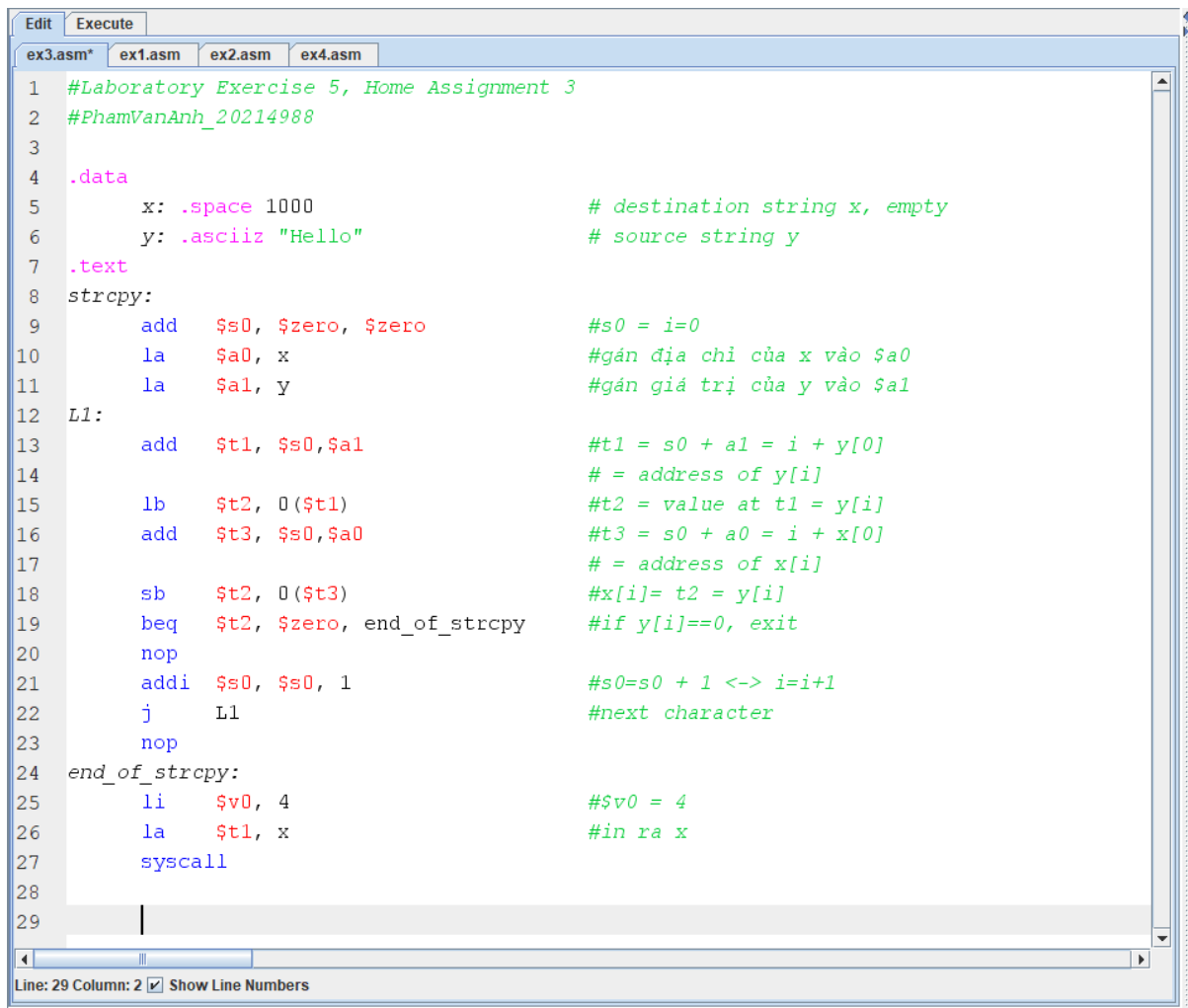
### 9. In ra kết quả

- $\$a0 = \$s0 + \$s1 = 10 + 12 = 22$



## ASSIGNMENT 3:

### 1. Code



```
1  #Laboratory Exercise 5, Home Assignment 3
2  #PhamVanAnh_20214988
3
4  .data
5      x: .space 1000          # destination string x, empty
6      y: .asciiz "Hello"     # source string y
7  .text
8  strcpy:
9      add    $s0, $zero, $zero    #s0 = i=0
10     la     $a0, x                #gán địa chỉ của x vào $a0
11     la     $a1, y                #gán giá trị của y vào $a1
12 L1:
13     add    $t1, $s0, $a1         #t1 = s0 + a1 = i + y[0]
14                                     # = address of y[i]
15     lb     $t2, 0($t1)           #t2 = value at t1 = y[i]
16     add    $t3, $s0, $a0         #t3 = s0 + a0 = i + x[0]
17                                     # = address of x[i]
18     sb     $t2, 0($t3)           #x[i]= t2 = y[i]
19     beq    $t2, $zero, end_of_strcpy #if y[i]==0, exit
20     nop
21     addi   $s0, $s0, 1           #s0=s0 + 1 <-> i=i+1
22     j      L1                   #next character
23     nop
24 end_of_strcpy:
25     li     $v0, 4                #$v0 = 4
26     la     $t1, x                #in ra x
27     syscall
28
29
```

Line: 29 Column: 2 ☒ Show Line Numbers

Thực hiện gõ chương trình vào công cụ MARS

## 2. Thực hiện câu lệnh số 9, 10, 11

- Câu lệnh số 9: Gán  $\$s0 = 0$

$\$s0$	16	0
$\$s1$	17	0

- Câu lệnh số 10: Gán địa chỉ của x vào  $\$a0$

$\$at$	1	268500992
$\$v0$	2	0
$\$v1$	3	0
$\$a0$	4	268500992

- Câu lệnh số 11: Gán địa chỉ của y vào  $\$a1$

$\$a0$	4	268500992
$\$a1$	5	268501992

## 3. Thực hiện câu lệnh số 13 → 23

- Câu lệnh số 13:  $\$t1 = \$s0 + \$a1 = i + y[0] \rightarrow \$t1$  mang địa chỉ của  $y[i]$

$\$t1$	9	268501992
--------	---	-----------

- Câu lệnh số 15:  $\$t2 = x[i] = y[i] \rightarrow$  lấy giá trị của  $\$t1$  lưu vào thanh ghi  $\$t2$

$\$t1$	9	268501992
$\$t2$	10	72

- Câu lệnh số 16:  $\$t3 = \$s0 + \$a0 = i + x[0] \rightarrow \$t3$  mang địa chỉ của  $x[i]$

$\$t3$	11	268500992
--------	----	-----------

- Câu lệnh số 18:  $\$t2 = x[i] = y[i] \rightarrow$  lấy giá trị của  $\$t2$  lưu vào thanh ghi  $\$t3$

$\$t3$	11	268500992
--------	----	-----------

- Câu lệnh số 19: so sánh  $\$t2$  và 0

- $\$t2 = 0 \rightarrow \text{end}$
- $\$t2 \neq 0 \rightarrow \text{chạy tiếp}$

- Câu lệnh số 21:  $\$s0 = \$s0 + 1$  ( $i=i+1$ ) (tăng giá trị của  $\$s0$ )

$\$s0$	16	1
--------	----	---

- Câu lệnh số 22: Vòng lặp L1 tới khi hết chuỗi

#### 4. Thực hiện câu lệnh số 25, 26, 27

- Câu lệnh 25: Gán giá trị \$v0 = 4 → option của SYSCALL là *PRINT STRING*

\$v0	2	4
------	---	---

- Câu lệnh 26: Gán địa chỉ của x vào \$t1

\$t1	9	268500992
------	---	-----------

- In ra:

```
Mars Messages Run I/O
Hello
-- program is finished running (dropped off bottom) --
```



## ASSIGNMENT 4:

### 1. Code

```

6  .text
7  main:
8  get_string:
9      li    $v0, 54          # 54 -> show input dialog
10     la    $a0, Message1
11     la    $a1, string
12     li    $a2, 100
13     syscall                # TODO
14  get_length:
15     la    $a0, string      # a0 = Address(string[0])
16     xor   $v0, $zero, $zero # v0 = length = 0
17     xor   $t0, $zero, $zero # t0 = i = 0
18  check_char:
19     add   $t1, $a0, $t0    # t1 = a0 + t0
20                     #= Address(string[0]+i)
21     lb    $t2, 0($t1)      # t2 = string[i]
22     beq   $t2, $zero, end_of_str # Is null char?
23     addi  $v0, $v0, 1      # v0=v0+1->length=length+1
24     addi  $t0, $t0, 1      # t0=t0+1->i = i + 1
25     j     check_char
26  end_of_str:
27  end_of_get_length:
28  print_length:              # TODO
29     li    $v0, 56          # show messdialog
30     la    $a0, Message2
31     add   $a1, $t0, -1      # -1 because the result includes last character '\0'
32     syscall
33
34
```

Thực hiện gõ chương trình vào công cụ MARS

## 2. Thực hiện câu lệnh số 10, 11, 12, 13, 14: Nhập xâu

- Câu lệnh số 10: Gán giá trị \$v0 = 54 → option của SYSCALL là *InputDialogString*

\$v0	2	54
------	---	----

- Câu lệnh số 11: Gán địa chỉ của Message1 vào \$a0

\$a0	4	268501042
------	---	-----------

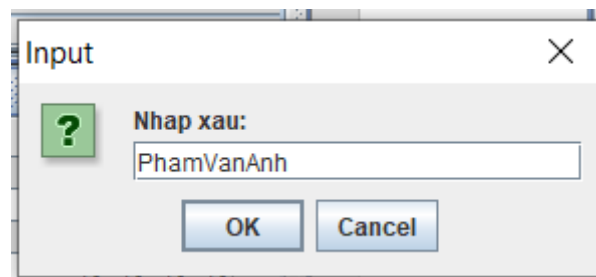
- Câu lệnh số 12: Gán địa chỉ của String vào \$a1

\$a1	5	268500992
------	---	-----------

- Câu lệnh số 13: \$a2 là max\_string → \$a2 = 100

\$a2	6	100
------	---	-----

- Kết quả:



Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)
0x10010000	m a h p	A n a v	\0 \n h n	\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

## 3. Thực hiện câu lệnh số 16, 17, 18

- Câu lệnh số 16: Gán địa chỉ của *string* vào \$a0

\$a0	4	268500992
------	---	-----------

- Câu lệnh số 17: Gán giá trị \$v0 = 0

\$v0	2	0
------	---	---

- Câu lệnh số 18: Gán giá trị của \$t0 = 0

\$t0	8	0
------	---	---

- Kết quả:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	m a h p	A n a v	\0 \n h n	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

#### 4. Thực hiện câu lệnh số 20 → 26: Check char

- Câu lệnh số 20:  $\$t1 = \$a0 + \$t0$  (Address(string[0]+i))

\$t1	9	268500992
------	---	-----------

- Câu lệnh số 22: Lấy giá trị từ địa chỉ \$t1 vào \$t2

\$t2	10	80
------	----	----

- Câu lệnh số 23: So sánh t2 và 0
  - Nếu  $t2 = 0 \rightarrow$  end
  - Nếu  $t2 \neq 0 \rightarrow$  chạy tiếp
- Câu lệnh số 24: Tăng giá trị của v0 lên  $v0 + 1$  (v0 là độ dài chuỗi)

\$v0	2	1
------	---	---

- Câu lệnh số 25: Tăng giá trị của t0 lên  $t0 + 1$  (t0: bước nhảy index)

\$t0	8	1
------	---	---

- Câu lệnh số 26: quay lại vòng lặp chevk\_char tới khi nào hết chuỗi

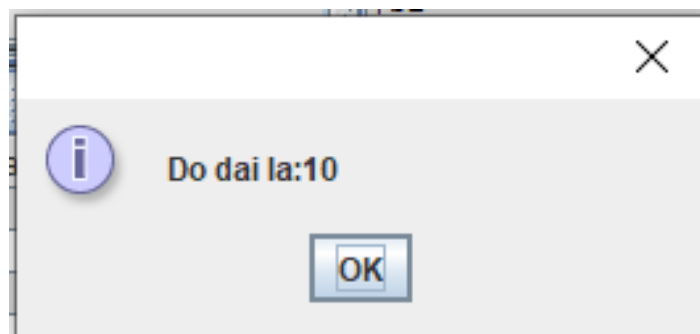
#### 5. Thực hiện câu lệnh số 27: end\_of\_str: dừng chuỗi

#### 6. Thực hiện câu lệnh số 28: end\_of\_get\_length: dừng lấy độ dài chuỗi

#### 7. Thực hiện câu lệnh số 29 → 33: In ra độ dài chuỗi

- Thực hiện câu lệnh số 30: Gán giá trị  $\$v0 = 6 \rightarrow$  option của SYSCALL là *MessageDialogInt*
- Thực hiện câu lệnh số 31: Gán địa chỉ của Message2 vào \$a0
- Thực hiện câu lệnh số 32: Phải -1 của \$a1 vì trừ đi kí tự cuối cùng là  $\backslash 0$

#### 8. Kết quả khi thử chuỗi “PhamVanAnh” → kết quả đúng



## ASSIGNMENT 5:

### 1. Code

```
1  #Laboratory Exercise 5, Home Assignment 5
2  #PhamVanAnh_20214988
3
4  .data
5      string:  .space  21
6      msg_rev:  .space  21
7      Message1: .asciiz "Input string: "
8      Message2: .asciiz "\nReverse string: "
9
10 .text
11 main:
12 get_string:
13     li $v0, 4
14     la $a0, Message1
15     syscall
16
17     li $v0, 8
18     la $a0, string
19     li $a1, 21
20     syscall
21
22 get_length:
23     la  $a0, string          # a0 = Address(string[0])
24     xor $v0, $zero, $zero    # v0 = length = 0
25     xor $t0, $zero, $zero    # t0 = i = 0
26 check_char:
27     add $t1, $a0, $t0        # t1 = a0 + t0
28                                     # = Address(string[0]+i)
29     lb  $t2, 0($t1)          # t2 = string[i]
30     beq $t2, $zero, end_of_str # Is null char?
31     addi $v0, $v0, 1          # v0=v0+1->length=length+1
32     addi $t0, $t0, 1          # t0=t0+1->i = i + 1
33     j   check_char
```

```

34 end_of_str:
35 end_of_get_length:
36
37 check_new_line:
38     addi $t4, $v0, -1
39     add $s0, $a0, $t4
40     lb $t5, 0($s0)
41     li $t8, '\n'
42     bne $t5, $t8, reverse_string
43     add $v0, $v0, -1
44
45 reverse_string:
46     la $a1, msg_rev
47     addi $v0, $v0, -1      # length of string
48     li $v1, 0             # v1 = i = -1
49 rev:
50     addi $t3, $v0, 1
51     beq $t3, $zero, end_of_rev
52     add $t1, $a0, $v0
53     lb $t2, 0($t1)
54     add $t3, $a1, $v1
55     sb $t2, 0($t3)
56     addi $v1, $v1, 1      # i=i+1
57     addi $v0, $v0, -1     # length = length - 1
58     j rev                 # next character
59 end_of_rev:
60
61 output:  li $v0, 4          # in ra "\nReverse string: "
62          la $a0, Message2
63          syscall
64
65          li $v0, 4          # in ra gia tri cua msg_rev (sau khi reverse string)
66          la $a0, msg_rev
67          syscall

```

## 2. Thực hiện câu lệnh 12 → 20: get\_string

10. Câu lệnh số 13: Gán giá trị \$v0 = 4 → option của SYSCALL là *PRINT STRING*

\$v0	2	4
------	---	---

11. Câu lệnh số 14: gán địa chỉ của Message1 vào a0

\$a0	1	268500992
\$v0	2	4
\$v1	3	0
\$a0	4	268501034

12. Câu lệnh số 17: gán v0 = 8 → option của SYSCALL là *READ STRING*

\$v0	2	8
------	---	---

13. Câu lệnh số 18: gán địa chỉ của string vào a0

\$zero	0	0
\$at	1	268500992
\$v0	2	8
\$v1	3	0
\$a0	4	268500992

### 1. Thực hiện câu lệnh 22 → 25: get\_length

14. Câu lệnh số 23: gán địa chỉ của string vào a0

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	8
\$v1	3	0
\$a0	4	268500992

15. Câu lệnh số 24: v0 = 0

\$v0	2	0
------	---	---

16. Câu lệnh số 25: t0 = 0

\$t0	8	0
------	---	---

### 1. Thực hiện câu lệnh 27 → 33: check\_char

17. Câu lệnh số 27: \$t1 = \$a0 + \$t0 (Address(string[0]+i))

\$t1	9	268500992
------	---	-----------

18. Câu lệnh số 29: Lấy giá trị từ địa chỉ \$t1 vào \$t2

\$t1	9	268500992
\$t2	10	86

- Câu lệnh số 30: So sánh t2 và 0
  - Nếu t2 = 0 → end
  - Nếu t2 != 0 → chạy tiếp
- Câu lệnh số 31: Tăng giá trị của v0 lên v0 + 1 (v0 là độ dài chuỗi)

\$v0	2	1
------	---	---

- Câu lệnh số 32: Tăng giá trị của t0 lên t0 + 1 (t0: bước nhảy index)

\$t0	8	1
------	---	---

- Câu lệnh số 33: quay lại vòng lặp check\_char tới khi nào hết chuỗi

### 1. Thực hiện câu lệnh 37 → 43

- Câu lệnh số 38: t4 = v0 - 1

\$t4	12	6
------	----	---

- Câu lệnh số 39:  $s0 = a0 + t4$

\$s0	16	268500998
------	----	-----------

- Câu lệnh số 40: Lấy giá trị từ thanh ghi s0 sang thanh ghi t5

\$t5	13	10
------	----	----

- Câu lệnh số 41:  $t8 = \backslash n$
- Câu lệnh số 42: so sánh t5 và t8
  - +  $t5 = t8 \rightarrow \text{add } v0 = v0 - 1$
  - +  $t5 \neq t8 \rightarrow \text{reverse\_string}$

## 2. Thực hiện câu lệnh 45 $\rightarrow$ 48: reverse\_string

- Câu lệnh số 46: gán địa chỉ của msg\_rev vào a1

\$at	1	268500992
\$v0	2	6
\$v1	3	0
\$a0	4	268500992
\$a1	5	268501013

- Câu lệnh số 47:  $v0 = v0 - 1$

\$at	1	268500992
\$v0	2	5

- Câu lệnh số 48:  $v1 = 0$

## 3. Thực hiện câu lệnh 49 $\rightarrow$ 58

- Câu lệnh số 50:  $t3 = v0 + 1$

\$t3	11	6
------	----	---

- Câu lệnh số 51: So sánh t3 và 0
  - +  $t3 = 0 \rightarrow \text{end}$
  - +  $t3 \neq 0$
- Câu lệnh số 52:  $t1 = a0 + v0$

\$t0	0	7
\$t1	9	268500997
\$t2	10	0

- Câu lệnh số 53: Lấy giá trị từ thanh ghi t1 sang thanh ghi t2

\$t2	10	104
------	----	-----

- Câu lệnh số 54:  $t3 = a1 + v1$

\$t2	10	104
\$t3	11	268501013

- Câu lệnh số 55: lấy giá trị của t3 lưu vào t2

\$t2	10	104
\$t3	11	268501013

- Câu lệnh số 56:  $v1 = v1 + 1$

\$v1	3	1
------	---	---

- Câu lệnh số 57:  $v0 = v0 - 1$

\$v0	2	4
------	---	---

- Câu lệnh số 58: vòng lặp *rev* tới hết chuỗi

## 2. Thực hiện câu lệnh 61 → 67: In chuỗi đảo

- Câu lệnh số 61:  $v0 = 4$

\$v0	2	4
------	---	---

- Câu lệnh số 62: Gán địa chỉ của Message2 vào a0

\$v1	3	6
\$a0	4	268501049

- Câu lệnh số 65:  $v0 = 4$

\$v0	2	4
------	---	---

- Câu lệnh số 66: Gán địa chỉ của reverse\_String vào \$a0

\$a0	4	268501013
------	---	-----------

## 3. Kết quả:

- Thử với chuỗi “PhamVanAnh”
- Kết quả:

