

Lab 5

Character string with SYSCALL function, and sorting

Đinh Ngọc Khánh Huyền - 20225726

Assignment 1

Code

.data

str1: .asciiz "The sum of "

str2: .asciiz " and "

str3: .asciiz " is "

.text

li \$s0,7

li \$s1, 8

add \$t0,\$s0,\$s1

li \$v0,4

la \$a0,str1

syscall

li \$v0,1

move \$a0,\$s0

syscall

li \$v0,4

la \$a0,str2

syscall

```
li $v0,1  
move $a0,$s1  
syscall
```

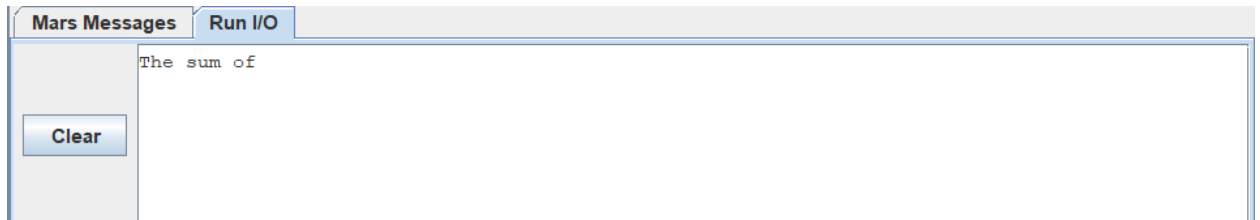
```
li $v0,4  
la $a0,str3  
syscall
```

```
li $v0,1  
move $a0,$t0  
syscall
```

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

Result

- In xâu “The sum of”



- In số 7

Mars Messages

Run I/O

Clear

The sum of 7

- In xâu “and”

Mars Messages

Run I/O

Clear

The sum of 7 and

- In số 8

Mars Messages

Run I/O

Clear

The sum of 7 and 8

- In xâu “is”

Mars Messages

Run I/O

Clear

The sum of 7 and 8 is

- In kết quả ra màn hình

EditExecute

Text Segment

| Blkpt | Address | Code | Basic | Source |
|-------|------------|------------|--------------------------|----------------|
| | 0x00400020 | 0x24020004 | addiu \$2,\$0,0x00000018 | li \$w0,4 |
| | 0x0040002c | 0x3c011001 | lui \$1,0x00001001 | la \$a0,str2 |
| | 0x00400030 | 0x3424000c | ori \$4,\$1,0x0000000c | |
| | 0x00400034 | 0x0000000c | syscall | syscall |
| | 0x00400038 | 0x24020001 | addiu \$2,\$0,0x00000022 | li \$w0,1 |
| | 0x0040003c | 0x00112021 | addiu \$4,\$0,217 | move \$a0,\$a1 |
| | 0x00400040 | 0x0000000c | syscall | syscall |
| | 0x00400044 | 0x24020004 | addiu \$2,\$0,0x00000026 | li \$w0,4 |
| | 0x00400048 | 0x3c011001 | lui \$1,0x00001001 | la \$a0,str3 |
| | 0x0040004c | 0x34240012 | ori \$4,\$1,0x00000012 | |
| | 0x00400050 | 0x0000000c | syscall | syscall |
| | 0x00400054 | 0x24020001 | addiu \$2,\$0,0x00000030 | li \$w0,1 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) |
|------------|------------|------------|------------|------------|-------------|
| 0x10010000 | 0x20656854 | 0x20647573 | 0x0020666f | 0xf46e6120 | 0x69200020 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

Registers

| Coproc 1 | Coproc 0 | Name | Number | Value |
|----------|----------|--------|--------|------------|
| | | \$zero | 0 | 0x00000000 |
| | | \$at | 1 | 0x10010000 |
| | | \$v0 | 2 | 0x0000000a |
| | | \$v1 | 3 | 0x00000000 |
| | | \$a0 | 4 | 0x0000000f |
| | | \$a1 | 5 | 0x00000000 |
| | | \$a2 | 6 | 0x00000000 |
| | | \$a3 | 7 | 0x00000000 |
| | | \$t0 | 8 | 0x0000000f |
| | | \$t1 | 9 | 0x00000000 |
| | | \$t2 | 10 | 0x00000000 |
| | | \$t3 | 11 | 0x00000000 |
| | | \$t4 | 12 | 0x00000000 |
| | | \$t5 | 13 | 0x00000000 |
| | | \$t6 | 14 | 0x00000000 |
| | | \$t7 | 15 | 0x00000000 |
| | | \$a0 | 16 | 0x00000007 |
| | | \$a1 | 17 | 0x00000000 |
| | | \$a2 | 18 | 0x00000000 |
| | | \$a3 | 19 | 0x00000000 |
| | | \$a4 | 20 | 0x00000000 |
| | | \$a5 | 21 | 0x00000000 |
| | | \$a6 | 22 | 0x00000000 |
| | | \$a7 | 23 | 0x00000000 |
| | | \$t8 | 24 | 0x00000000 |
| | | \$t9 | 25 | 0x00000000 |
| | | \$k0 | 26 | 0x00000000 |
| | | \$k1 | 27 | 0x00000000 |
| | | \$gp | 28 | 0x10000000 |
| | | \$fp | 29 | 0xffffffff |
| | | \$fp | 30 | 0x00000000 |
| | | \$ra | 31 | 0x00000000 |
| | | pc | | 0x0040006f |
| | | hi | | 0x00000000 |
| | | lo | | 0x00000000 |

Mars Messages

Run I/O

Clear

The sum of 7 and 8 is 15
-- program is finished running --

Explain

Với đoạn lệnh

```
li $v0,4  
la $a0,str1  
syscall
```

- Giá trị 4 vào thanh ghi \$v0 thực hiện chức năng in xâu
- Chương trình sẽ in ra chuỗi với địa chỉ byte đầu tiên của chuỗi được lưu ở thanh ghi \$a0 và chương trình sẽ đọc đến khi nào gặp giá trị null.
- Khi khai báo biến str1 chứa chuỗi “The sum of”, các kí tự của chuỗi sẽ được lưu vào từ ô nhớ có địa chỉ 0x10010000 theo từng byte từ MSB đến LSB. Đến khi gặp kí tự dừng lại “\0” là dấu hiệu kết thúc chuỗi.

Tương tự với đoạn lệnh in ra “is” và “and”

Với đoạn lệnh

```
li $v0,1  
move $a0,$s1  
syscall
```

- Giá trị 1 vào thanh ghi \$v0 thực hiện chức năng in số tự nhiên
- Chuyển giá trị của thanh ghi \$s1 vào thanh ghi \$a0 để in ra.

Tương tự với đoạn lệnh in ra “8” và “15”

Assignment 2

Code

.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

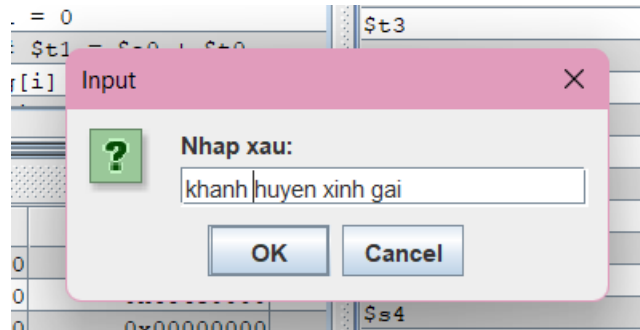
la \$a0, Message2

move \$a1, \$t0

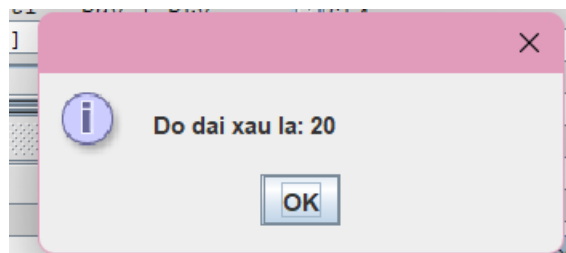
syscall

Result:

- Nhập xâu



- In ra độ dài xâu



⇒ **Kết quả đúng với lý thuyết**

Explain:

- Gọi hàm 'get_string' sử dụng syscall để nhập một xâu từ người dùng và lưu trữ vào vùng nhớ được chỉ định.
- Sau đó 'get_length' để bắt đầu vòng lặp đếm độ dài của xâu. Trong đó \$a0 lưu địa chỉ của string[0] và \$t0 được khởi tạo bằng 0 dùng để đếm số ký tự trong chuỗi. Vòng lặp check_char để kiểm tra xem ký tự hiện tại có phải ký tự kết thúc của xâu không (null character). Nếu không thì tăng t0 và tiếp tục lặp, ngược lại nhảy đến end_of_str.
- Đánh dấu kết thúc vòng lặp đếm độ dài xâu 'end_of_str'
- Sau đó in ra màn hình độ dài của xâu bằng cách sử dụng syscall như assignment 1 'print_length'.

Assignment 3

Code

.data

get_char: .space 15

message1: .asciiz "Input letter "

message2: .asciiz ": "

message3: .asciiz "\n"

message4: .asciiz "String input (Reverse): "

.text

li \$s0, 15 # N = 15

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 "Input letter i: "

li \$v0, 4

la \$a0, message1

syscall

addi \$t1, \$s1, 1

li \$v0, 1

move \$a0, \$t1

syscall

li \$v0, 4

```

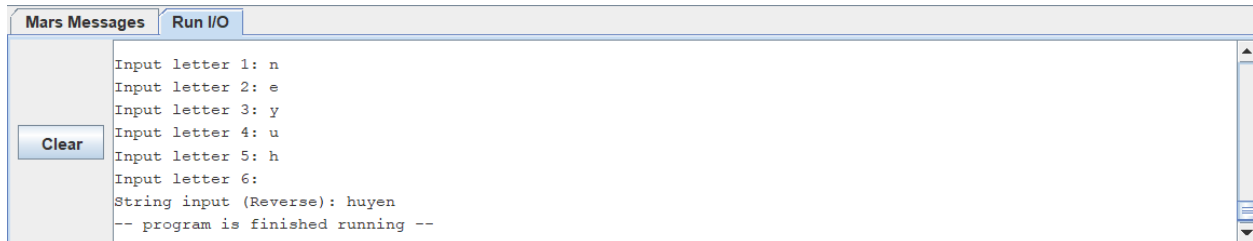
    la $a0, message2
    syscall
    li $v0, 12 # Read character
    syscall
    move $t0, $v0
    beq $t0, $s3, end_read_char # Press "Enter" branch to exit
    li $v0, 4
    la $a0, message3
    syscall
    add $s5, $s2, $s1 # $s5 = Address of get_char[i] = get_char[0] + i
    sb $t0, 0($s5) # Store character 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

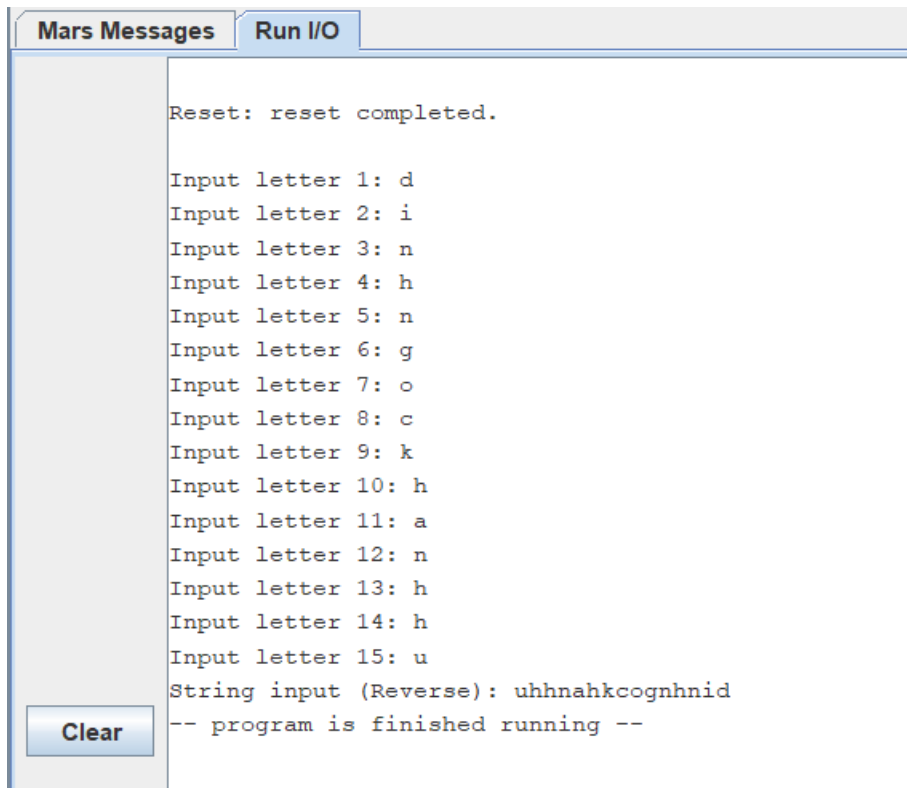
Result

Trường hợp ấn Enter



```
Mars Messages Run I/O
Input letter 1: n
Input letter 2: e
Input letter 3: y
Input letter 4: u
Input letter 5: h
Input letter 6:
String input (Reverse): huyen
-- program is finished running --
```

Trường hợp nhập đủ 15 kí tự



```
Mars Messages Run I/O
Reset: reset completed.

Input letter 1: d
Input letter 2: i
Input letter 3: n
Input letter 4: h
Input letter 5: n
Input letter 6: g
Input letter 7: o
Input letter 8: c
Input letter 9: k
Input letter 10: h
Input letter 11: a
Input letter 12: n
Input letter 13: h
Input letter 14: h
Input letter 15: u
String input (Reverse): uhhnahkcognhnid
-- program is finished running --
```

⇒ **Kết quả đúng với lý thuyết**

Explain:

- Gán giá trị 15 vào thanh ghi \$s0, đại diện cho độ dài tối đa của xâu cần nhập. Gán 0 vào thanh ghi \$s1, đại diện cho vị trí hiện tại trong xâu cần nhập.

- Load địa chỉ bắt đầu của vùng nhớ 'get_char' (nơi lưu trữ chuỗi nhập vào thanh ghi \$s2
- Gán giá trị 10 vào thanh ghi \$s3, đây là mã ASCII của ký tự xuống dòng '\n', được sử dụng để kiểm tra xem người dùng có nhập ký tự Enter không.
- 'beq \$s1, \$s0, end_read_char' Kiểm tra xem đã nhập đủ 15 ký tự chưa, nếu đủ rồi thì nhảy tới 'end_read_char' để kết thúc vòng lặp.
- Vòng lặp nhập ký tự vào. Vòng lặp ngược lại để in ký tự ra theo thứ tự ngược lại.