

Laboratory Exercise 5 – Report:

Character string with SYSCALL function, and sorting

Lê Văn Duẩn - 20194508

1. Assignment 1

- Mã nguồn:

```
#Laboratory Exercise 5, Home Assignment 1
```

```
.data
```

```
test: .asciiz "LeVDuan 20194508"
```

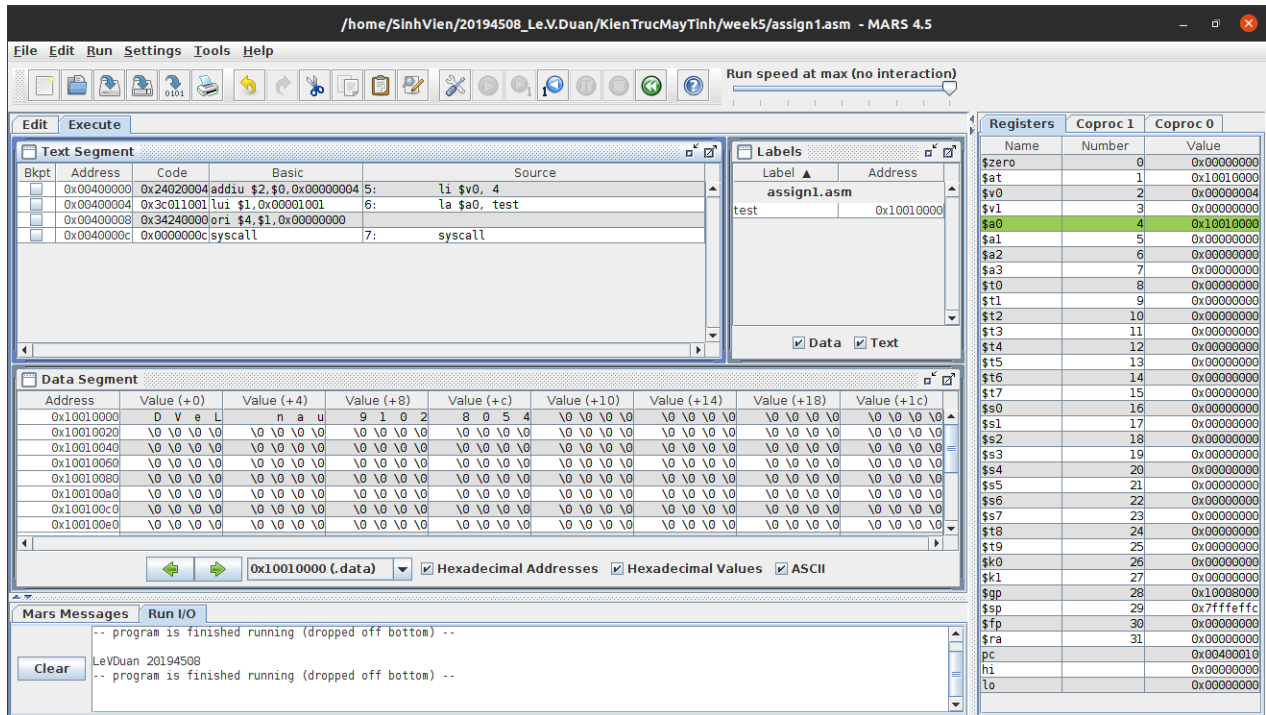
```
.text
```

```
li $v0, 4
```

```
la $a0, test
```

```
syscall
```

- Kết quả chạy mô phỏng:



- Giải thích:

\$v0 lưu trữ code là 4 -> thực hiện lệnh in ra màn hình console giá trị được lưu tại địa chỉ của thanh ghi \$a0

Lưu trữ chuỗi:

Tại địa chỉ của thanh ghi \$a0 chuỗi "LeVDuan 20194508" được lưu theo cụm ký tự tương ứng

2. Assignment 2

- Mã nguồn:

#Laboratory Exercise 5, Home Assignment 2

.data

Mess1: .asciiz "The sum of "

Mess2: .asciiz " and "

Mess3: .asciiz " is "

.text

li \$s1, 2019 # 4 so dau MSSV

```
li $s2, 4508 # 4 so sau MSSV
```

```
li $v0, 4
```

```
la $a0, Mess1
```

```
syscall
```

```
li $v0, 1
```

```
add $a0,$zero, $s1 # 2019
```

```
syscall
```

```
li $v0, 4
```

```
la $a0, Mess2
```

```
syscall
```

```
li $v0, 1
```

```
add $a0, $zero, $s2 # 4508
```

```
syscall
```

```
li $v0, 4
```

```
la $a0, Mess3
```

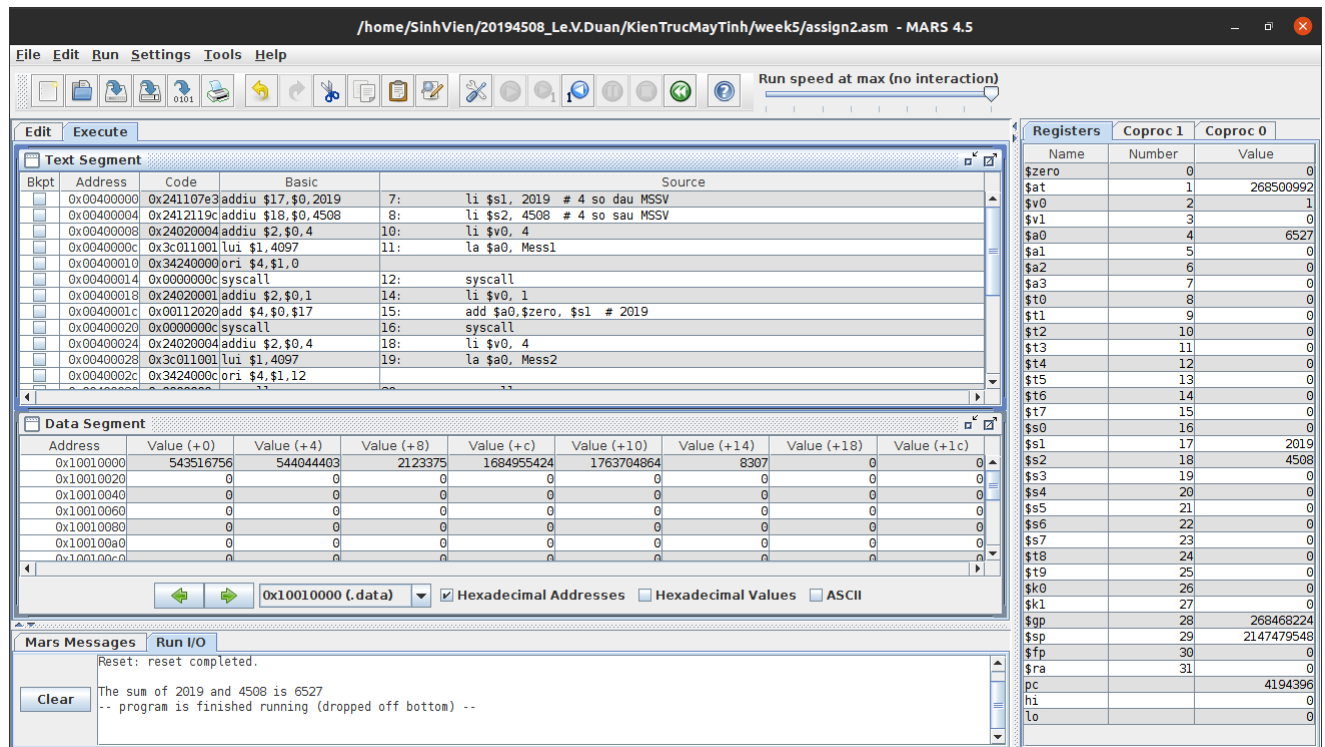
```
syscall
```

```
li $v0, 1
```

```
add $a0, $s1, $s2
```

syscall

- Kết quả chạy:



- Giải thích:

Lưu các giá trị 2019 và 4508 vào lần lượt các thanh ghi `$s0` và `$s1`

In ra từng đoạn của cụm câu "The sum of ", "and", "is " và xen kẽ là in các giá trị từ các thanh ghi `$s0`, `$s1` với code của `$v0` là 4 để in chuỗi và 1 để in số dạng interger. Tổng 2 số được gán trực tiếp vào `$a0` ở lệnh cuối để in tổng ra màn hình

3. Assignment 3

-Mã nguồn:

#Laboratory Exercise 5, Assignment 3

.data

x: .space 1000 # destination string x, empty

y: .asciiz "LeVDuan" # source string y

.text

la \$a0, x

la \$a1, y

strcpy: add \$s0, \$zero, \$zero # khai bao bien i = 0

L1:add \$t1, \$s0, \$a1 # \$t1 = \$s0 + \$a1 = i + y[0] = y[i]

lb \$t2, 0(\$t1) # \$t2 = value at \$t1 = y[i]

add \$t3, \$s0, \$a0 # \$t3 = \$s0 + \$a0 = i + x[0] = 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

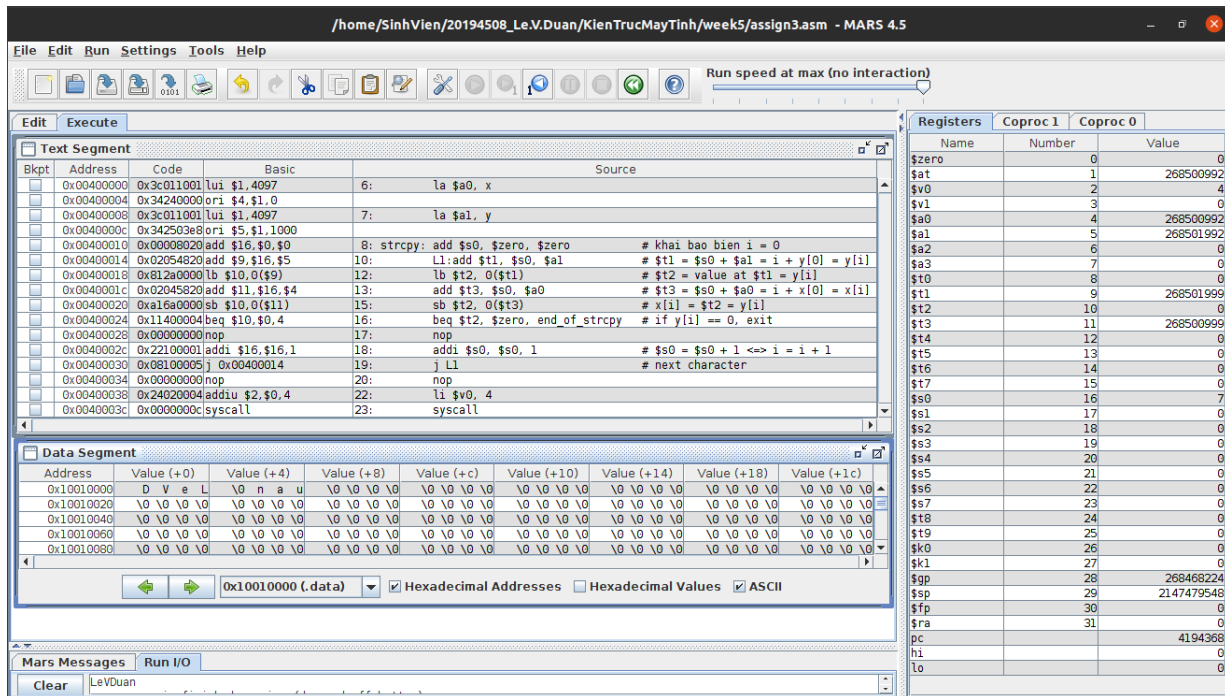
nop

end_of_strcpy:

li \$v0, 4

syscall

- Kết quả chạy mô phỏng:



- Giải thích:

Để copy chuỗi y sang chuỗi x:

triển khai vòng lặp từng vòng lặp từng kí tự của chuỗi y bằng cách truy cập đến từng địa chỉ sau đó lưu vào địa chỉ tương ứng của x với biến chạy i được thể hiện bằng giá trị của \$s0

kết thúc vòng lặp khi đọc đến kí tự '\0' của chuỗi y

lệnh cuối dùng syscall để in ra chuỗi x để kiểm tra -> "LeVDuan" -> đúng như mô tả.

4. Assignment 4

- Mã nguồn:

#Laboratory Exercise 5, Assignment 4

.data

string: .space 50

Message1: .asciiz "Nhap xau: "

Message2: .asciiz "Do dai xau la: "

```

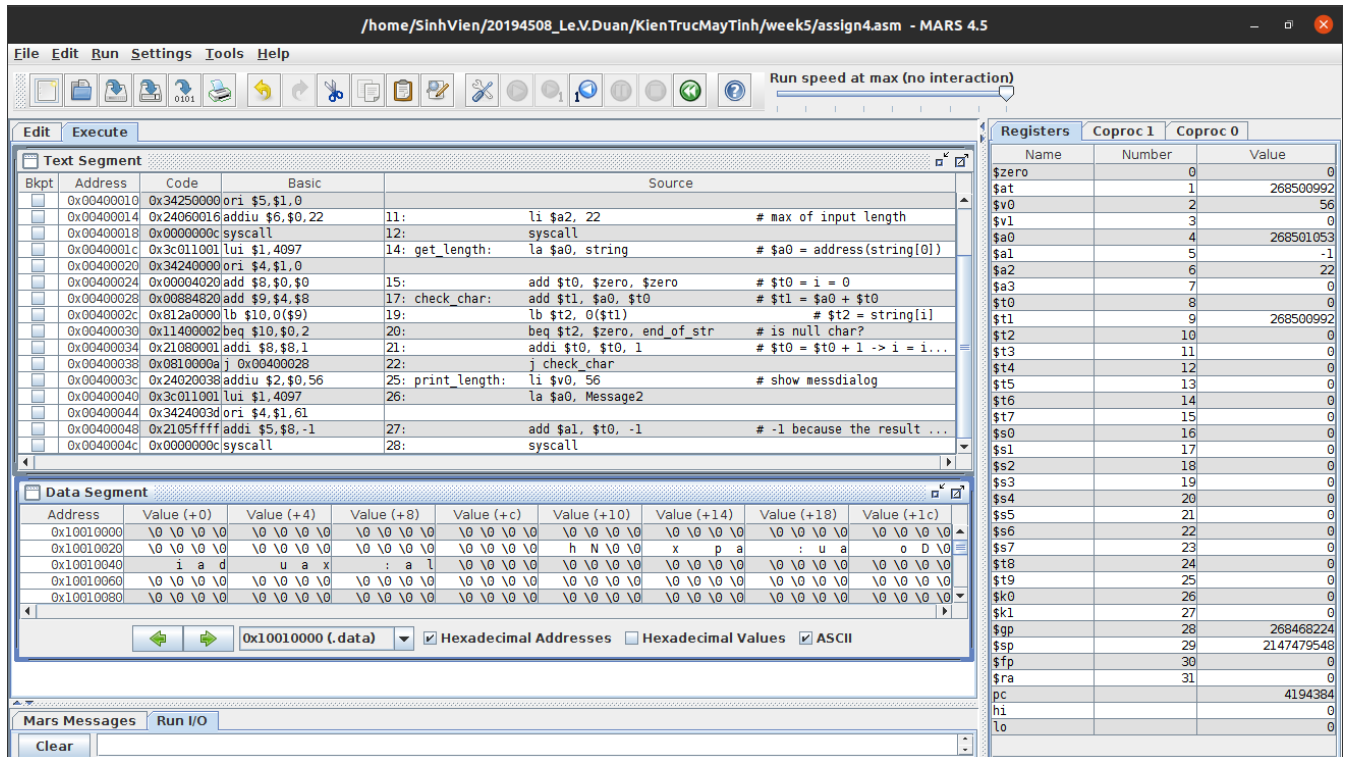
.text
main:
get_string: li $v0, 54          # 54 -> show input dialog
            la $a0, Message1
            la $a1, string
            li $a2, 22          # max of input length
            syscall

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:li $v0, 56          # show messdialog
            la $a0, Message2
            add $a1, $t0, -1        # -1 because the result includes last character '\0'
            syscall

```

- Kết quả chạy:



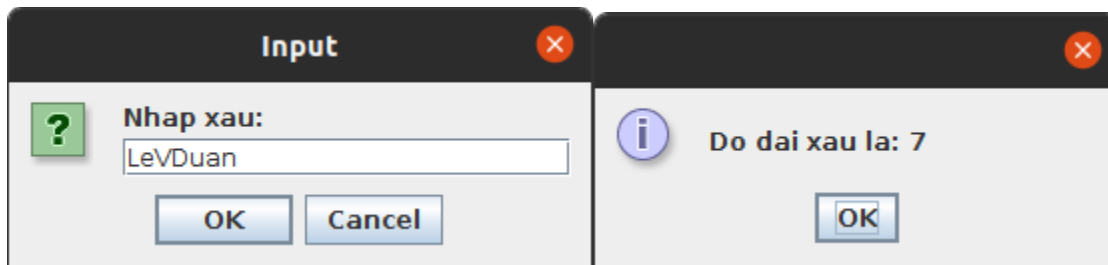
- Giải thích:

Sử dụng syscall số 54 để gọi InputDialogString và lưu nó vào biến string.

Sau đó duyệt từng kí tự của chuỗi và đếm chiều dài chuỗi.

kết quả phải -1 vì chuỗi luôn có kí tự kết thúc là '\0'

kết quả khi nhập 1 chuỗi ví dụ:



5. Assignment 5

- Mã nguồn:

#Laboratory Exercise 5, Assignment 5

.data

Message: .asciiz "Nhap tung ki tu"

string: .space 20

.text

init: add \$s0, \$zero, \$zero # i = 0

la \$s1, string # load address of string to \$s1

read_str:

read_char: li \$v0, 12 # 12 -> read character

syscall

nop

check_char: beq \$v0, 10, print_revert_str # gap ki tu enter LF = 10

-> ket thuc string -> print

add \$t1, \$s0, \$s1 # lay dia chi cua string[i]

sb \$v0, 0(\$t1) # luu \$v0 vao string[i]

addi \$s0, \$s0, 1 # i += 1

slti \$t0, \$s0, 20 # i < 20? 1 : 0

beq \$t0, \$zero, print_revert_str # neu khong thi ket thuc viec doc

j read_char

print_revert_str: slt \$t0, \$s0, \$zero # neu i < 0 ? 1 : 0

bne \$t0, \$zero, exit # if true ->string empty-> exit

add \$t1, \$s0, \$s1 # lay dia chi string[i]

lb \$t2, 0(\$t1) # load vao \$t2 string[i]

li \$v0, 11 # print character

add \$a0, \$zero, \$t2 # \$a0 = string[i]

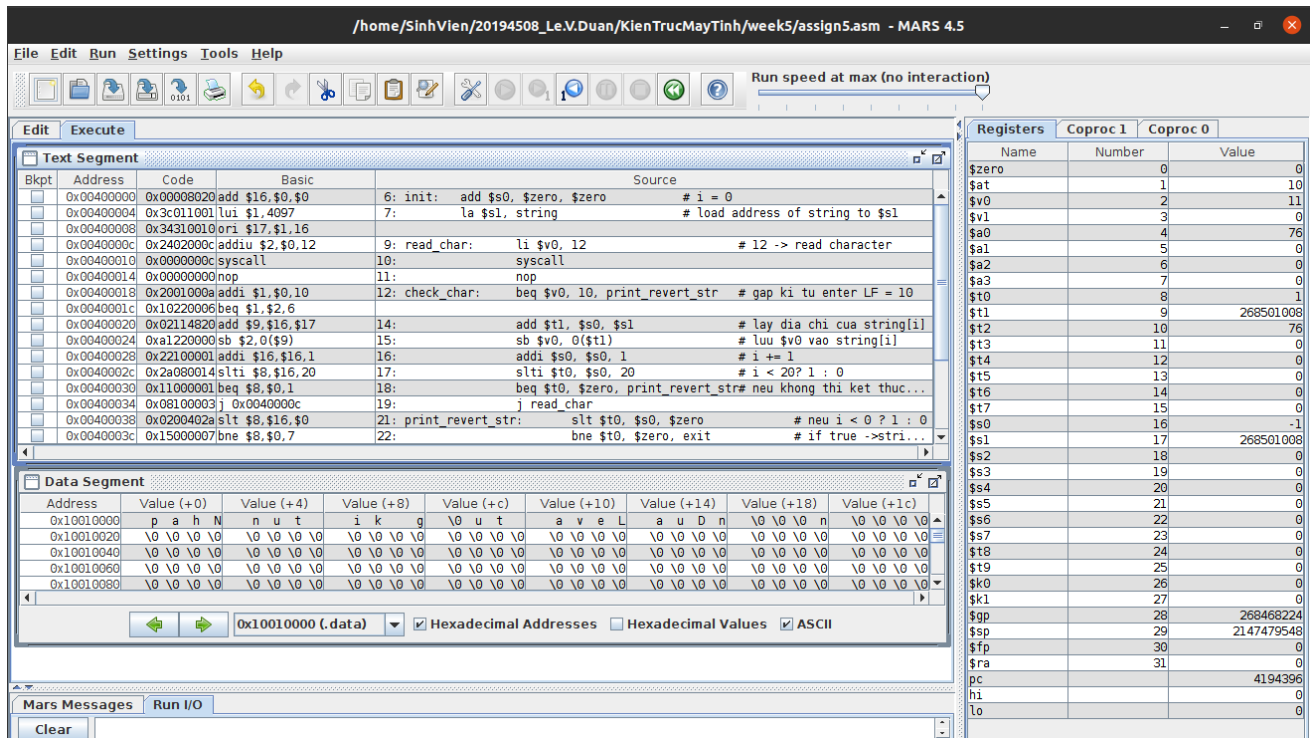
syscall

addi \$s0, \$s0, -1 # s0 -= 1

j print_revert_str

exit:

- Kết quả chạy:



- Giải thích:

Đọc input từng kí tự và dùng vòng lặp check xem đó có phải là enter hay ko để kết thúc việc đọc chuỗi và lưu vào string

In ngược chuỗi bằng cách in ngược từng kí tự với \$t0 lưu chỉ số của chuỗi và sau mỗi lần in kí tự thì giảm đi 1 và khi chỉ số này về 0 -> exit

- kết quả console:

