ASSIGNMENT WEEK 5

-Họ tên: Hồ Tuấn Huy

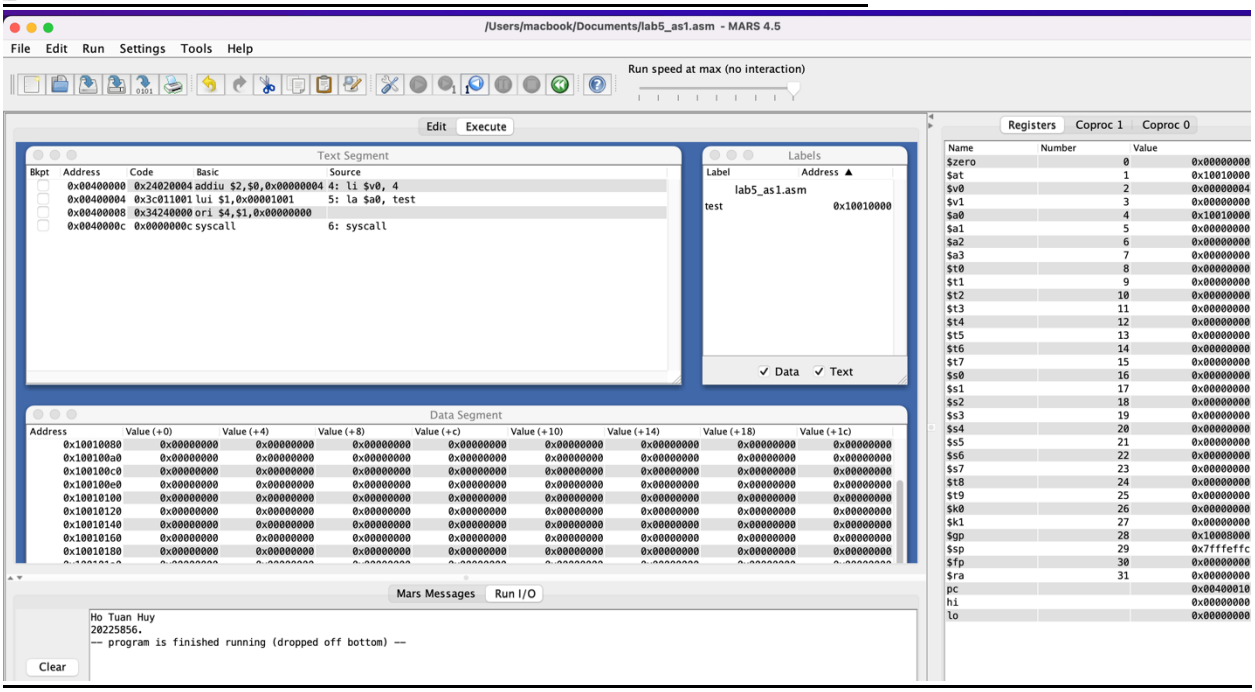
-MSSV: 20225856

Assignment 1:

Assignment 1

Create a new project to implement the program in Home Assignment 1. Compile and upload to simulator. Run and observe the result. Go to data memory section, check how test string are stored and packed in memory.

```
1  .data
2  test: .asciiz "Ho Tuan Huy\n20225856."
3  .text
4  li $v0, 4
5  la $a0, test
6  syscall
```



- Lệnh li $v0, 4 => load giá trị 4 vào thanh ghi $v0, giá trị 4 trong $v0 tương ứng với syscall để hiển thị chuỗi ra màn hình.
- Lệnh la $a0, test => load địa chỉ của chuỗi test (địa chỉ của ký tự đầu của chuỗi) vào thanh ghi $a0. Đây là địa chỉ của chuỗi cần hiển thị ra màn hình.
- Lệnh syscall gọi hệ thống để thực hiện chức năng được chỉ định trong thanh ghi $v0, trong trường hợp này là hiển thị chuỗi có địa chỉ bắt đầu từ thanh ghi $a0.

Assignment 2:

Assignment 2

Create a new project to print the sum of two register $s0 and $s1 according to this format:

"The sum of (s0) and (s1) is (result)"

```
1   .data
2   text1: .asciiz "The sum of "
3   text2: .asciiz " and "
4   text3: .asciiz " is "
5   .text
6   li $s0, 3
7   li $s1, 9
8   add $s2, $s0, $s1
9
10  li $v0, 4
11  la $a0, text1
12  syscall
13
14  li $v0, 1
15  add $a0, $zero, $s0
16  syscall
17
18  li $v0, 4
19  la $a0, text2
20  syscall
21
22  li $v0, 1
23  add $a0, $zero, $s1
24  syscall
25
26  li $v0, 4
27  la $a0, text3
28  syscall
```

```
27
28  li $v0, 1
29  add $a0, $zero, $s1
30  syscall
31
32  li $v0, 4
33  la $a0, text3
    syscall
```

- Trước hết ta gán giá trị $s0=3, $s1=9, $s2=$s0+$s1 (=3+9=12).

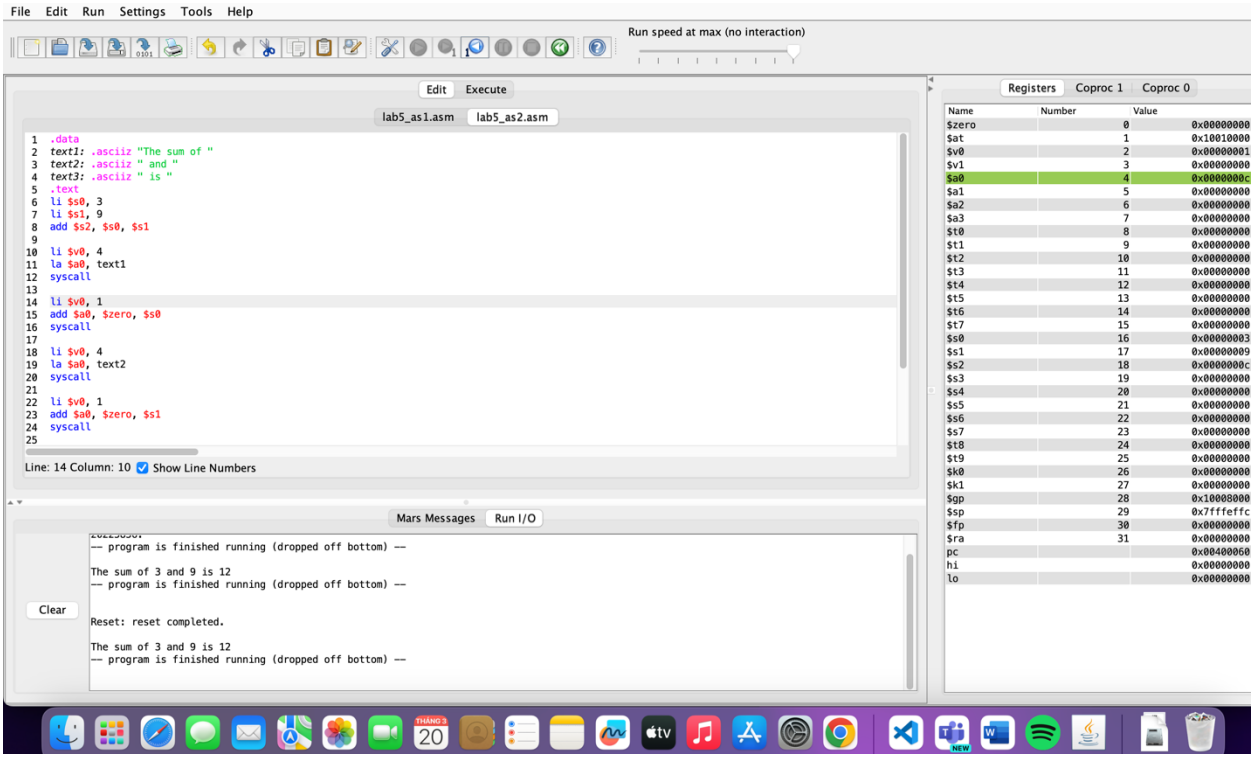
- Lệnh li $v0, 4 => load giá trị 4 vào thanh ghi $v0, giá trị 4 trong $v0 tương ứng với syscall để hiển thị chuỗi ra màn hình.
- Lệnh la $a0, text1 => load địa chỉ của chuỗi text1 (địa chỉ của ký tự đầu của chuỗi) vào thanh ghi $a0. Đây là địa chỉ của chuỗi cần hiển thị ra màn hình.
- Lệnh syscall gọi hệ thống để thực hiện chức năng được chỉ định trong thanh ghi $v0, trong trường hợp này là hiển thị chuỗi có địa chỉ bắt đầu từ thanh ghi $a0. Và kết quả được in ra đầu tiên là “The sum of”:

```
The sum of
```

- Lệnh li $v0, 1 => load giá trị 1 vào thanh ghi $v0, giá trị 1 trong $v0 tương ứng với syscall để in ra màn hình một số nguyên.
- Add $a0, $zero, $s0 => $a0=$zero+$s0=0+3=3.
- Syscall để gọi hệ thống in ra số nguyên 3:

```
The sum of 3
```

- Các dòng lệnh tiếp theo có cấu trúc và chức năng tương tự. Cuối cùng ta thu được kết quả đúng với lý thuyết là:



Assignment 3:

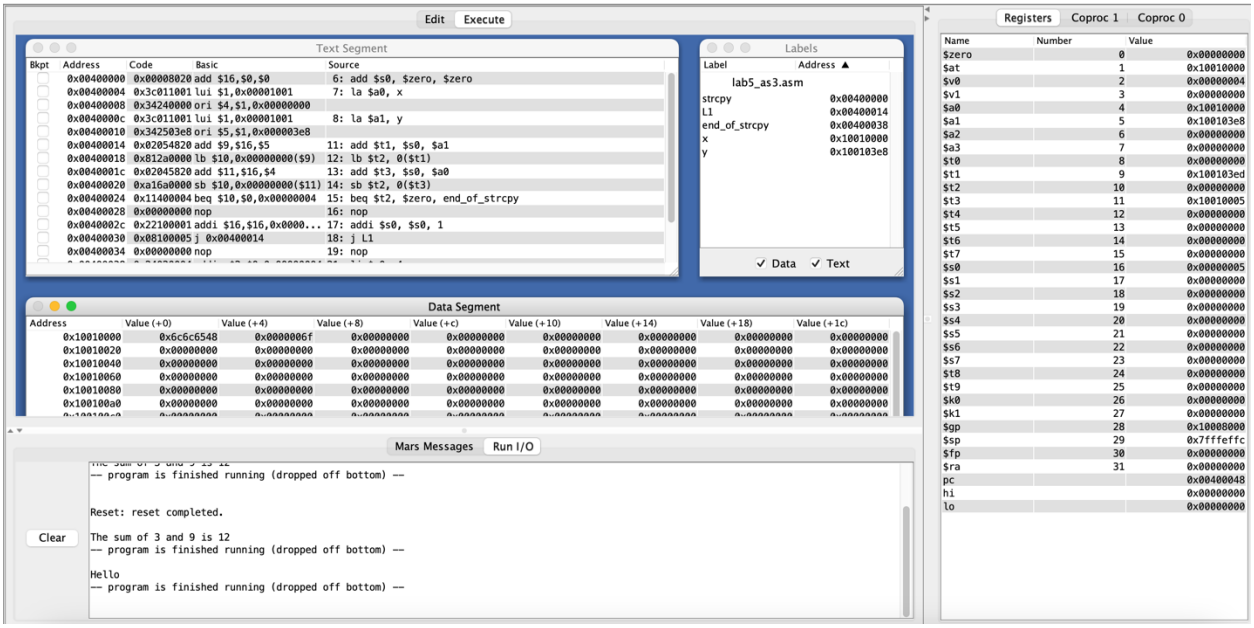
Assignment 3

Create a new project to implement the program in Home Assignment 2. Add more instructions to assign a test string for y variable, and implement *strcpy* function. Compile and upload to simulator. Run and observe the result.

```
1  .data
2  x: .space 1000
3  y: .asciiz "Hello"
4  .text
5  strcpy:
6  add $s0, $zero, $zero
7  la $a0, x
8  la $a1, y
9
10 L1:
11 add $t1, $s0, $a1
12 lb $t2, 0($t1)
13 add $t3, $s0, $a0
14 sb $t2, 0($t3)
15 beq $t2, $zero, end_of_strcpy
16 nop
17 addi $s0, $s0, 1
18 j L1
19 nop
20 end_of_strcpy:
21 li $v0, 4
22 la $a0, x
23 syscall
```

- Nhãn strcpy đánh dấu điểm bắt đầu của hàm sao chép chuỗi.
- Khởi tạo thanh ghi $s0 làm chỉ số bắt đầu của chuỗi.
- Load địa chỉ của vùng nhớ x vào thanh ghi $a0, địa chỉ của chuỗi y vào $a1.
- Nhãn L1 đánh dấu điểm bắt đầu của vòng lặp sao chép chuỗi.
- Gán giá trị tại địa chỉ đầu tiên của string y vào địa chỉ đầu tiên của string x bằng lệnh:
  +lb=>lấy giá trị tại địa chỉ đó.
  +sb=>gán giá trị vào địa chỉ đó.
  ⇨ Tạo 1 vòng lặp để lấy lần lượt từng địa chỉ (ký tự) tiếp theo và thực hiện gán các ký tự tại địa chỉ đó của string y vào địa chỉ tương ứng của string x.

- Dùng lệnh beq để kiểm tra ký tự kết thúc. Nếu chưa gặp NULL thì tiếp tục lặp, nếu gặp thì dừng.
- Sau đó lặp lại tương tự.
- Lệnh li, la và syscall tương tự như assignment 1 và 2=>Kết quả:



Assignment 4:

Assignment 4

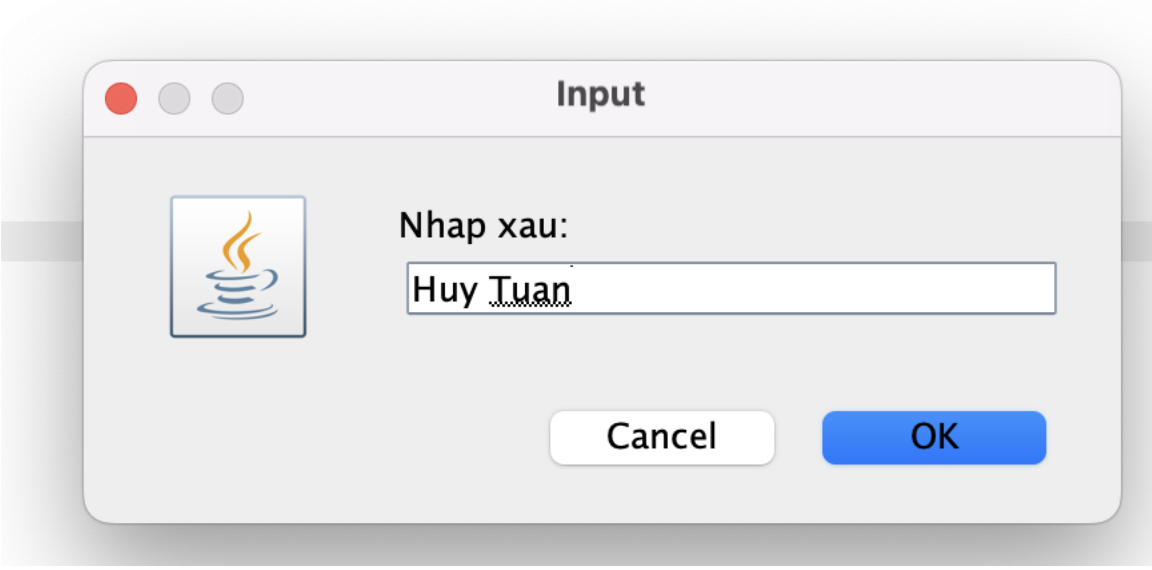
Accomplish the Home Assignment 3 with syscall function to get a string from dialog, and show the length to message dialog.

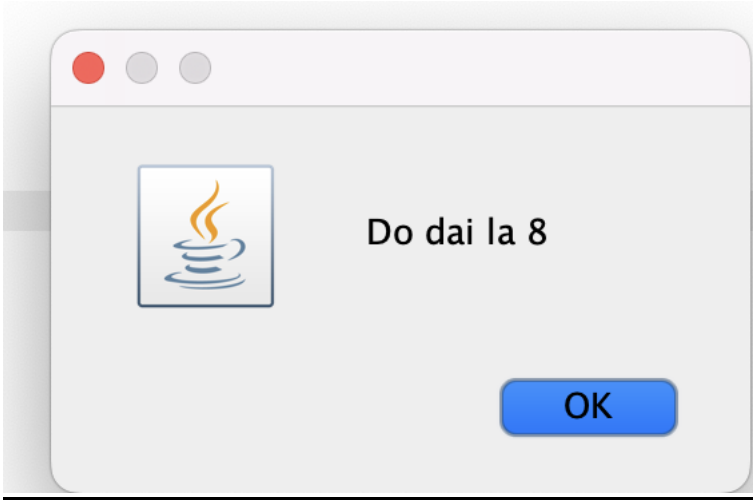
```
.data
string: .space 50
Message1: .asciiz "Nhap xau:"
Message2: .asciiz "Do dai la "
.text
main:
get_string:
li $v0, 54
la $a0, Message1
la $a1, string
la $a2, 50
syscall
get_length:
la $a0, string # a0 = Address(string[0])
xor $s0, $zero, $zero # s0 = length = 0
xor $t0, $zero, $zero # t0 = i = 0
check_char: add $t1, $a0, $t0 # t1 = a0 + t0
#= Address(string[0]+i)
lb $t2, 0($t1) # t2 = string[i]
beq $t2,$zero,end_of_str # Is null char?
addi $s0, $s0, 1 # v0=v0+1->length=length+1
addi $t0, $t0, 1 # t0=t0+1->i = i + 1
j check_char
end_of_str:
end_of_get_length:
```

```
23    j check_char
24  end_of_str:
25  end_of_get_length:
26  print_length:
27  li $v0, 56
28  la $a0, Message2
29  addi $a1, $s0, -1
30  syscall
```

- Khai báo biến string rỗng để lưu chuỗi nhập từ bàn phím.
- Tạo hàm get_string=>tạo ra dialog nhập chuỗi từ bàn phím rồi lưu chuỗi đó vào string.
- Tạo hàm get_length=>lưu địa chỉ của string vào $a0.
- Tạo vòng lặp check_char để kiểm tra lần lượt từng ký tự của chuỗi.
  +Khác NULL=>tăng độ dài thêm 1 và tăng i thêm 1 (chuyển đến ký tự tiếp theo).
  +Bằng NULL=>Kết thúc vòng lặp.
- Hàm print_length=>tạo ra dialog in độ dài của chuỗi.

Kết quả:





Assignment 5:

Assignment 5

Write a program that let user input a string. Input process will be terminated when user press Enter or then length of the string exceed 20 characters. Print the reverse string.

```
1   .data
2   a: .asciiz "nhap chuoi:"
3   b: .asciiz "chuoi dao:"
4   c: .space 100
5   d: .space 100
6   .text
7   li $s4, 19
8   li $s0, 0
9
10  la $a1, c
11  la $a2, d
12
13  input_loop:
14  li $v0, 12
15  syscall
16  beq $v0, 10, end_input
17  add $t1, $s0, $a1
18  sb $v0, 0($t1)
19  addi $s0, $s0, 1
20  bgt $s0, $s4, end_input
21  j input_loop
22  end_input:
23  strcpy:
24  addi $s0, $s0, -1
25  add $s1, $zero, $zero
```

```
24  addi $s0, $s0, -1
25  add $s1, $zero, $zero
26  addi $s4, $zero, -1
27  loop1:
28  beq $s0, $s4, end1
29  add $t1, $s0, $a1
30  lb $t2, 0($t1)
31  add $t3, $s1, $a2
32  sb $t2, 0($t3)
33  addi $s0, $s0, -1
34  addi $s1, $s1, 1
35  j loop1
36  end1:
37  li $v0, 59
38  la $a0, b
39  la $a1, d
40  syscall
```

- $s4 khởi tạo là 19=>giới hạn độ dài tối đa của chuỗi nhập (là 20 bao gồm cả ký tự kết thúc chuỗi NULL).
-$s0 khởi tạo là 0 là chỉ số vị trí ký tự trong chuỗi nhập.
- Khi nhập chuỗi:
+Sử dụng syscall 12 để đọc 1 ký tự từ người dùng.
+Nếu ký tự là ký tự xuống dòng thì quá trình nhập sẽ kết thúc.
+Lưu trữ các ký tự nhập vào mảng c.
- Xử lý chuỗi đảo:
+Sử dụng vòng lặp để duyệt qua chuỗi nhập từ cuối về đầu và lưu trữ vào mảng d để tạo chuỗi đảo.
+Sử dụng các thanh ghi $s0, $s1, $s4=>duyệt qua và lưu trữ chuỗi đảo.
- In chuỗi đảo:
+Sử dụng syscall 59 để in ra chuỗi thông báo “chuoi dao:”.
+Truyền địa chỉ của chuỗi đảo (d) để in ra chuỗi đảo.

=>Kết quả: