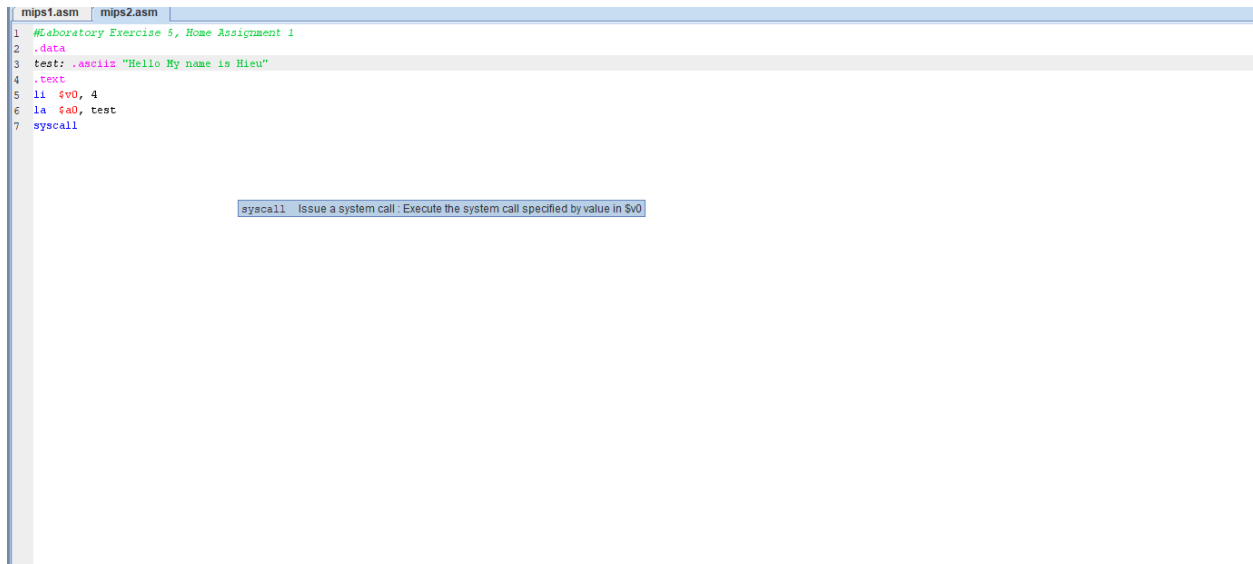


## Bài thực hành số 4: Chuỗi kí tự có hàm SYSCALL và sắp xếp

Họ tên: Nguyễn Văn Hiếu – 20225717

### Assignment 1:



```
mips1.asm | mips2.asm
1 #Laboratory Exercise 5, Home Assignment 1
2 .data
3 test: .ascii "Hello My name is Hieu"
4 .text
5 li $v0, 4
6 la $a0, test
7 syscall
```

syscall Issue a system call : Execute the system call specified by value in \$v0

Sau lần chạy đầu:

\$v0	2	0x00000004
------	---	------------

Sau lần chạy thứ hai:

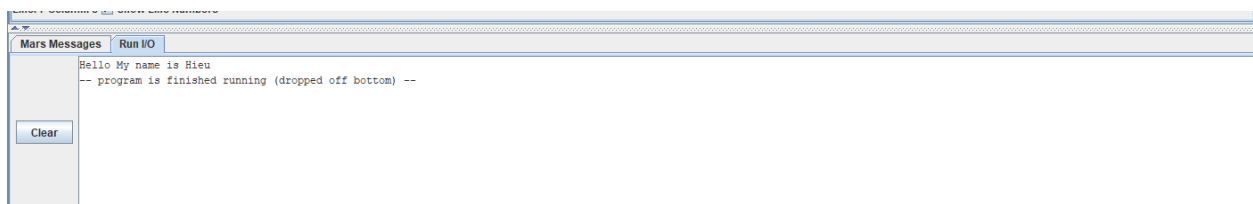
\$a0	1	0x10010000
------	---	------------

Lần chạy cuối cùng:

\$a0	4	0x10010000
------	---	------------

Chuỗi kí tự được lưu thành cụm 4 bytes, mỗi kí tự được dưới dạng 1 byte, kí tự đứng trước được đẩy vào trước.

Kết quả thu được:



```
Mars Messages | Run I/O
Hello My name is Hieu
-- program is finished running (dropped off bottom) --
Clear
```

⇒ Kết quả thu được đúng với lí thuyết.

## Assignment 2:

**.data**

**str1: .ascii "The sum of "**

**str2: .ascii " and "**

**str3: .ascii " is: "**

**.text**

**li \$s0,4**

**li \$s1,8**

**add \$s2,\$s0,\$s1      #sum=\$s0+\$s1**

**li \$v0,4      #printf str1**

**la \$a0,str1**

**syscall**

**li \$v0,1      #printf s0**

**move \$a0,\$s0**

**syscall**

**li \$v0,4      #printf str2**

**la \$a0,str2**

**syscall**

**li \$v0,1      #printf s1**

**move \$a0,\$s1**

**syscall**

```

li $v0,4          #printf str3
la $a0,str3
syscall

li $v0,1          #printf sum
move $a0,$s2
syscall

li $v0,10
syscall    #ket thuc chuong trinh

```

Ban đầu nhập 2 giá trị được lưu ở thanh ghi \$s0 và \$s1 lần lượt là 4 và 8. Tổng là 12.

Kết quả thu được:

```

Reset: reset completed.
The sum of 4 and 8 is: 12
-- program is finished running (dropped off bottom) --

```

⇒ Đúng với lí thuyết.

### Assignment 3:

#### #Laboratory Exercise 5, Home Assignment 2

**.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
la $a0,x

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:

```

Chương trình trên thực hiện copy chuỗi kí tự “Hello” từ source y sang des x bằng cách copy từng kí tự một. Vòng lặp dừng lại khi chuỗi kí tự cuối cùng được copy là NULL.

+Thanh ghi \$t1 lưu địa chỉ của của chuỗi cần copy.

+L1 thực hiện vòng lặp copy từng kí tự của chuỗi.

+Cập nhật giá trị kí tự y[i] vào thanh ghi t2

+Lệnh beq kiểm tra kí tự cuối cùng có bằng NULL không. Nếu bằng thì nhảy xuống dòng lệnh end\_of\_strcpy rồi dừng chương trình.

+Thanh ghi s0 dùng để tăng ghi trị  $i=i+1$  sau mỗi lần lặp.

Kết quả thu được:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1 1 e #	\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
0x10010020	\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
0x10010040	\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
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
0x10010100	\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
0x10010120	\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
0x10010140	\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
0x10010160	\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
0x10010180	\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
0x100101a0	\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

⇒ Đúng với lí thuyết.

#### Assignment 4:

#### #Laboratory Exercise 5, Home Assignment 3

**.data**

**string: .space 50**

**Message1: .asciiz "Nhap xau: "**

**Message2: .asciiz "Do dai xau la: "**

**.text**

**main:**

**get\_string: # TODO**

**li \$v0,54            #Nhap chuoi can tinh do dai xau**

**la \$a0, Message1**

**la \$a1, string**

**la \$a2, 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: # TODO

li $v0, 56

la $a0, Message2                #In ra dong lenh "do dai xau la"

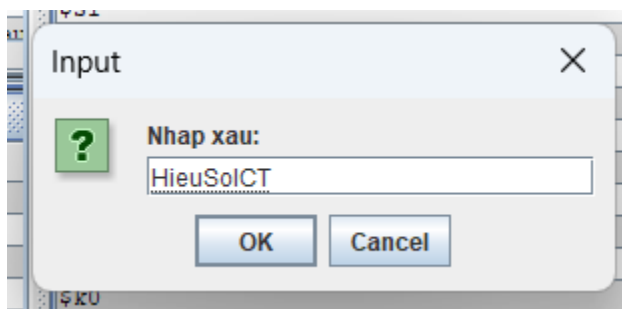
addi $t0,$t0,-1                 #Do dai xau dang in ca ki tu xuong dong nen phai -1

move $a1,$t0                     #In ra ket qua do dai xau

syscall

```

Đoạn chương trình trên thực hiện câu lệnh đếm số kí tự của chuỗi nhập vào xong in ra màn hình. Giả sử nhập vào chuỗi "HieuSolCT".



## Assignment 5:

## .data

```
str: .asciiz "\n"
```

**li \$s0, 20      # N = 20**

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

### read\_char:

```
li $v0, 12 #Doc vao tung ki tu
```

## syscall

```

    move $t0, $v0

    beq $t0, $s3, end_read_char #"Neu ki tu nhap vao la dau enter thi
dung"

    li $v0, 4

    la $a0, str

    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:

print:

li $v0, 11

lb $a0, 0($s5)

syscall

beq $s5, $s2, exit

addi $s5, $s5, -1    # in nguoc xau

j print

exit:

li $v0, 10 syscall

```

Khi nhập vào chuỗi kí tự “abcdefgh” thì kết quả thu được là:



Clear

a  
b  
c  
d  
e  
f  
g  
h

hgfedcba

⇒ Đúng với lí thuyết.