Laboratory Exercise 5

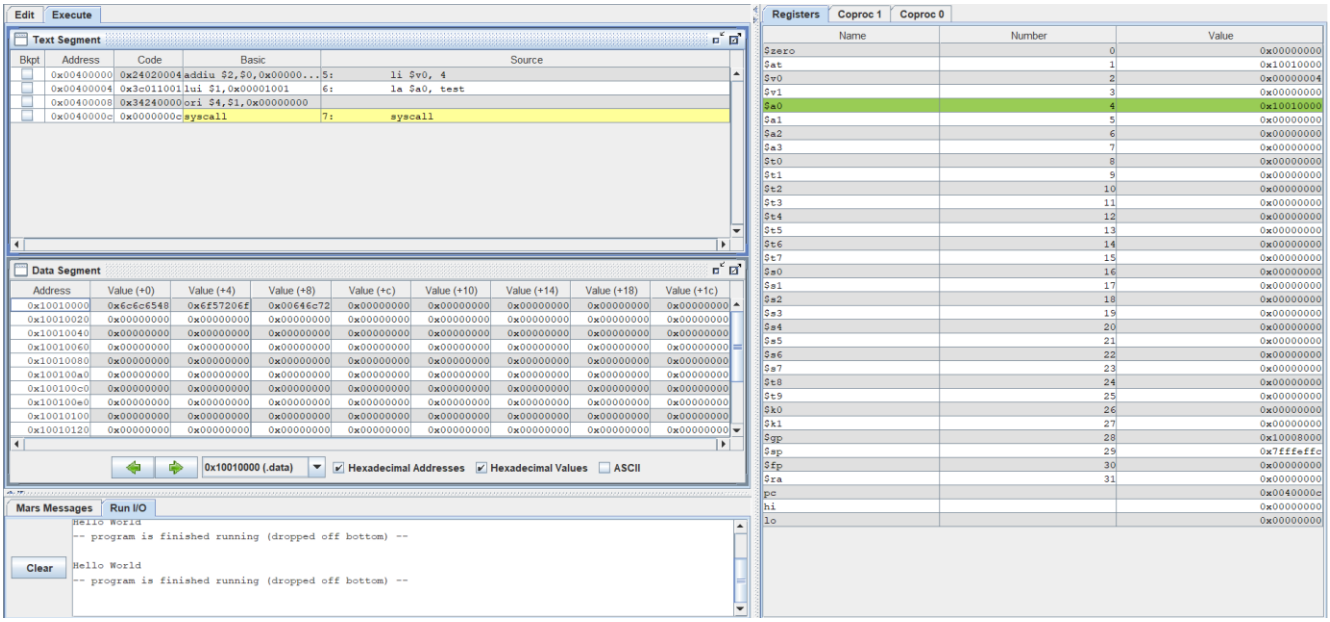
Character string with SYSCALL function, and sorting

Trần Khánh Quỳnh-20225762

1. Assignment 1:

```
1  #Lab 5, Assignment 1, Tran Khanh Quynh
2  .data
3  test: .asciiz "Hello World"
4  .text
5          li $v0, 4
6          la $a0, test
7          syscall
```

Kết quả chạy chương trình:



-Nhận xét:

+Thanh ghi $v0 được nạp vào giá trị 4, tiếp theo thanh ghi $at chuyển thành 0x10010000. Cuối cùng thanh ghi $a0 chuyển thành 0x10010000

+Thanh ghi $v0 nạp vào giá trị là 4, còn $a0 nạp địa chỉ của chuỗi kí tự (0x10010000 là địa chỉ của chuỗi kí tự biến test). Khi nạp 4 vào $v0 thì ghi gọi lệnh syscall → thực hiện in ra màn hình giá trị trong địa chỉ được lưu ở $a0

Lưu trữ trong địa chỉ 0x1001000:

Chuỗi “Hello World” được tách thành cụm 4 kí tự, rồi được lưu vào theo thứ tự đảo ngược lại với 4 byte địa chỉ

→Tách thành “lleH” “oW o” và “\0ldr”

“lleH”: chữ l có mã ascii là 108, khi chuyển sang hệ hexa là 0x6c

Chữ e có mã ascii là 101, chuyển sang hệ hexa là 0x65

Chữ H có mã ascii là 72, chuyển sang hệ hexa là 0x48

Lưu “lleH” sẽ là 0x6c6c6548

“oW o”: Chữ o có mã ascii là 111 → hexa là 0x6F

Chữ W có mã ascii là 87 → chuyển sang hexa là 0x57

Dấu cách → 0x20

Khi lưu “roW o” sẽ là 0x6F57206F

“\0dlr” : \0 khi chuyển sang hệ hexa là 0x00

Chữ l có mã ascii 108 → 0x6c

Chữ d có mã ascii 100 → 0x64

Chữ r có mã ascii 114 → 0x72

Khi lưu “\0 ldr” sẽ là 0x00646c72

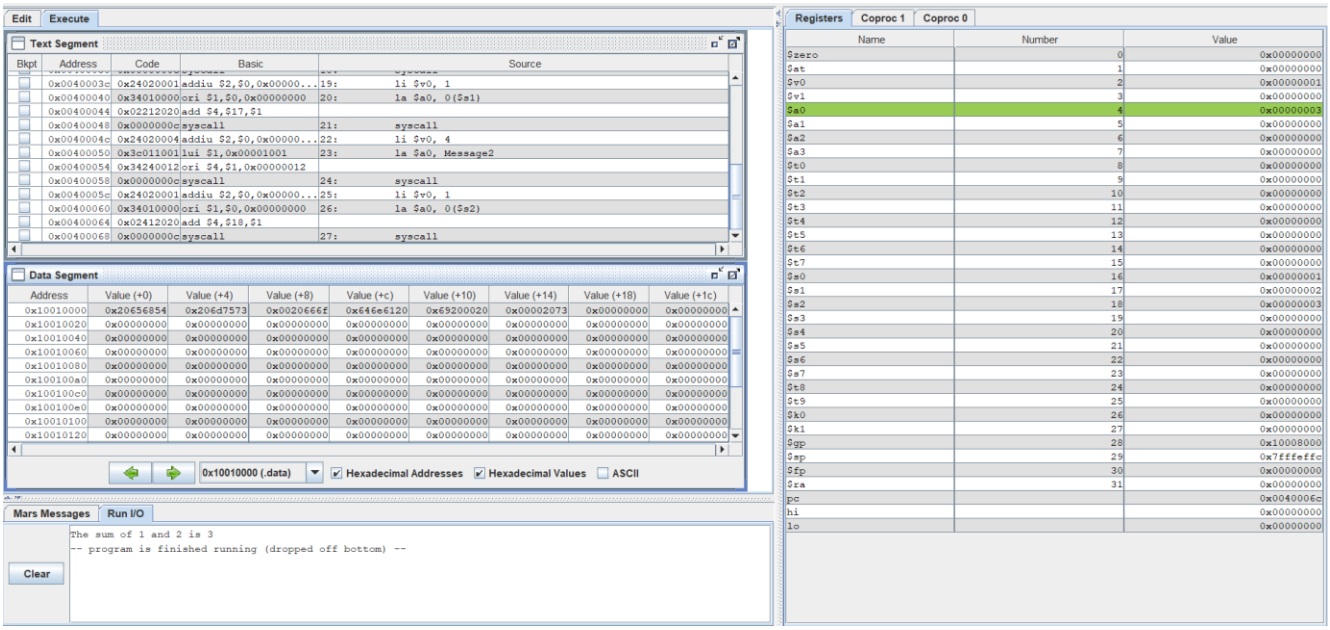
2. Assignment 2

-Mã nguồn

Lab5*

```
.data:
Message: .asciiz "The sum of "
Message1: .asciiz " and "
Message2: .asciiz " is "
.text
        li $s0, 1       #value of $s0
        li $s1, 2       #Value of $s1
        add $s2, $s0, $s1 #Sum

        li $v0, 4               #Print Message
        la $a0, Message
        syscall
        li $v0, 1               #Print value of $s0
        la $a0, 0($s0)
        syscall
        li $v0, 4               #Print Message1
        la $a0, Message1
        syscall
        li $v0, 1               #Print value of $s1
        la $a0, 0($s1)
        syscall
        li $v0, 4               #Print Message2
        la $a0, Message2
        syscall
        li $v0, 1               #Print sum
        la $a0, 0($s2)
        syscall
```

-Kết quả chạy chương trình



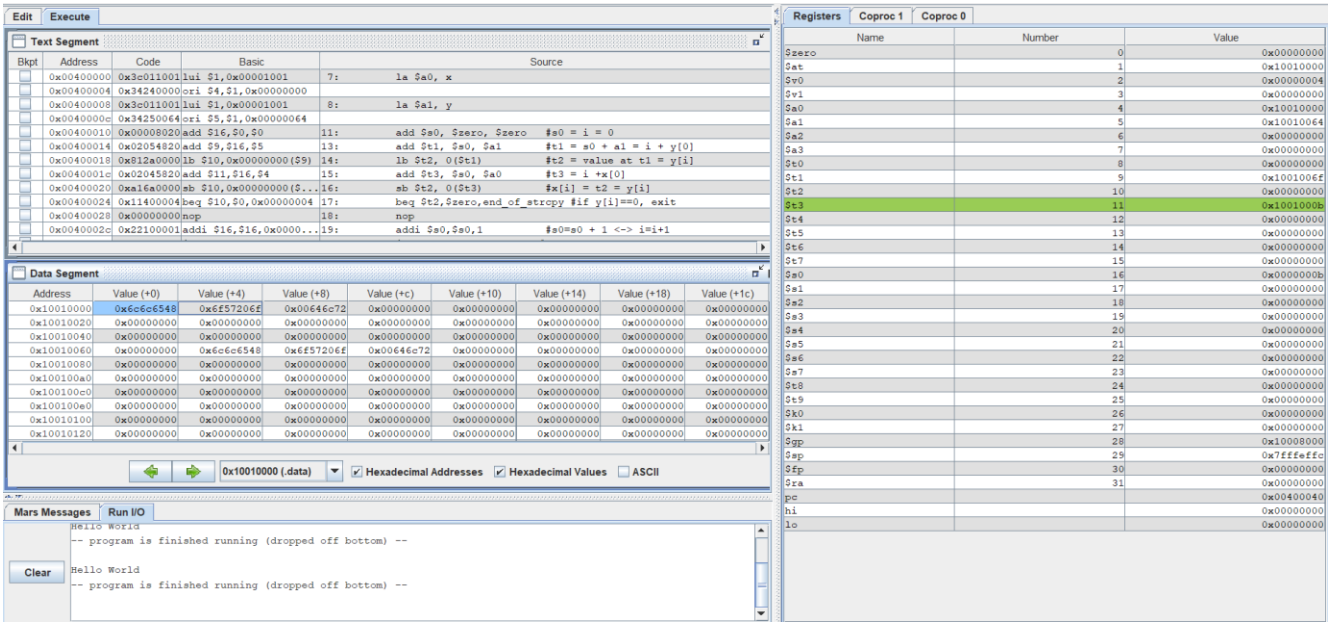
Lần lượt lưu giá trị 1 vào $s0, 2 vào $s1 và sau đó tính tổng rồi lưu vào $s2.

In lần lượt đầu tiên là đoạn “The sum of”, sau đó in ra giá trị của $s0; đoạn “and” với giá trị của $s1. Tiếp theo đó in “is” và giá trị của $s2 in cuối cùng.

3. Assignment 3

```
#Assignment 3, Lab 05, Tran Khanh Quynh
.data
x: .space 100
y: .asciiz "HelloWorld"

.text
        la $a0, x
        la $a1, y

strcpy:
        add $s0, $zero, $zero   #s0 = i = 0
L1:
        add $t1, $s0, $a1       #t1 = s0 + a1 = i + y[0]
        lb $t2, 0($t1)          #t2 = value at t1 = y[i]
        add $t3, $s0, $a0       #t3 = i +x[0]
        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



Giải thích:

+Đầu tiên sẽ lưu địa chỉ của biến x vào $a0; lưu địa chỉ của biến y vào $a1. Do ta quy định x có không gian là 100 (64 trong hệ hexa), vì vậy nên giá trị lưu trong $a0 (tức địa chỉ của x) các giá trị của $a1 (tức địa chỉ của y) 64 đơn vị(0x10010000 và 0x10010064)

+Thực hiện lệnh add để gán i = 0 vào thanh ghi $s0

+Đi vào lần đầu của vòng lặp L1:

Câu lệnh add để cộng i với địa chỉ của i, tức i sẽ đi lần lượt từng kí tự một trong chuỗi. Với lần đầu tiên thì i = 0

Thanh ghi $t1 sẽ lưu địa chỉ của y[i]. Thanh ghi $t1 lúc này sẽ lưu địa chỉ của y[0], nên $t1 có giá trị là 0x10010064

Câu lệnh lb dùng để lấy giá trị từ địa chỉ lưu trong thanh ghi $t1 lưu vào thanh $t2. Ta có y[0] = ‘H’. H có mã ascii là 72, khi chuyển sang hexa sẽ là 48. Vì vậy nên thanh $t2 sẽ chuyển thành giá trị là 0x00000048

Thanh ghi $t3 sẽ lưu địa chỉ của x[i]. Với i = 0 lúc này nên thanh ghi $t3 lưu địa chỉ của x[0], vì thế nên $t3 có giá trị là 0x10010000.

Lệnh sb $t2, 0($t3) lưu giá trị của $t2 vào địa chỉ mà $t3 trỏ đến. Tức trong thanh nhớ có địa chỉ 0x10010000 thì đã lưu lại kí tự ‘H’; hay đã lưu được kí tự đầu tiên của chuỗi y vào chuỗi x.

Câu lệnh beq để kiểm tra nếu $t2 có giá trị bằng 0, tức gặp kí tự ‘\0’ thì sẽ kết thúc chuỗi nên dừng copy.

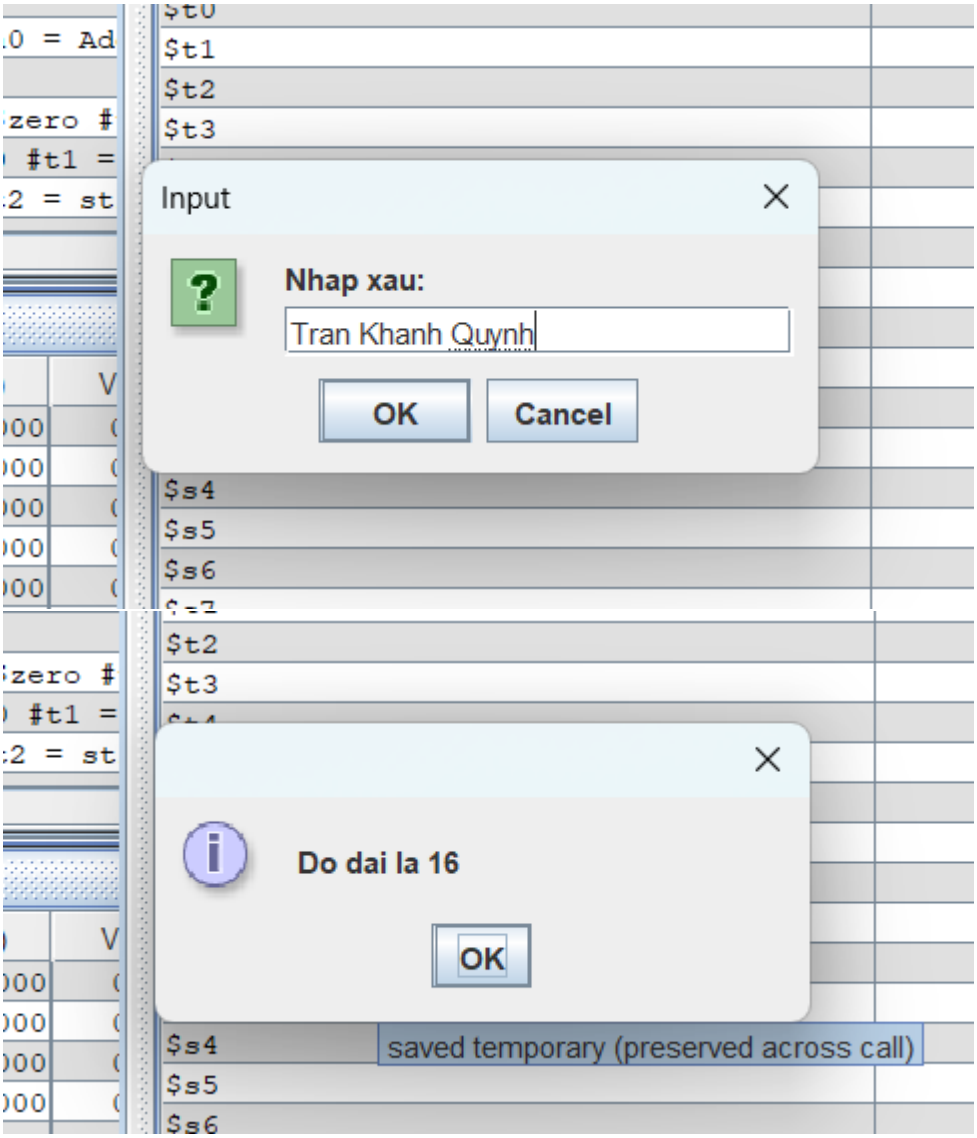
+Tiếp tục thực hiện vòng lặp cho đến khi gặp kí tự kết thúc, ta lần lượt lưu được nội dung của chuỗi y vào chuỗi x.

4. Assignment 4

-Mã nguồn:

```
#Assignment 4, Lab 05, Tran Khanh Quynh

.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 $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
        addi $t0, $t0, 1 #i = i+1
        j check_char
end_of_str:
end_of_get_length:
print_length:
        li $v0, 56
        la $a0, Message2
        sub $a1, $t0, 1
        syscall
```

-Kết quả chạy:



-Giải thích:

+get_string: Đặt $v0 bằng 54 để có thể InputDialogString. Sau đó đưa các tham số: $a0 lưu địa chỉ của chuỗi Message1, $a1 lưu địa chỉ của chuỗi nhập vào, $a2 lưu lại giới hạn số kí tự sẽ đọc

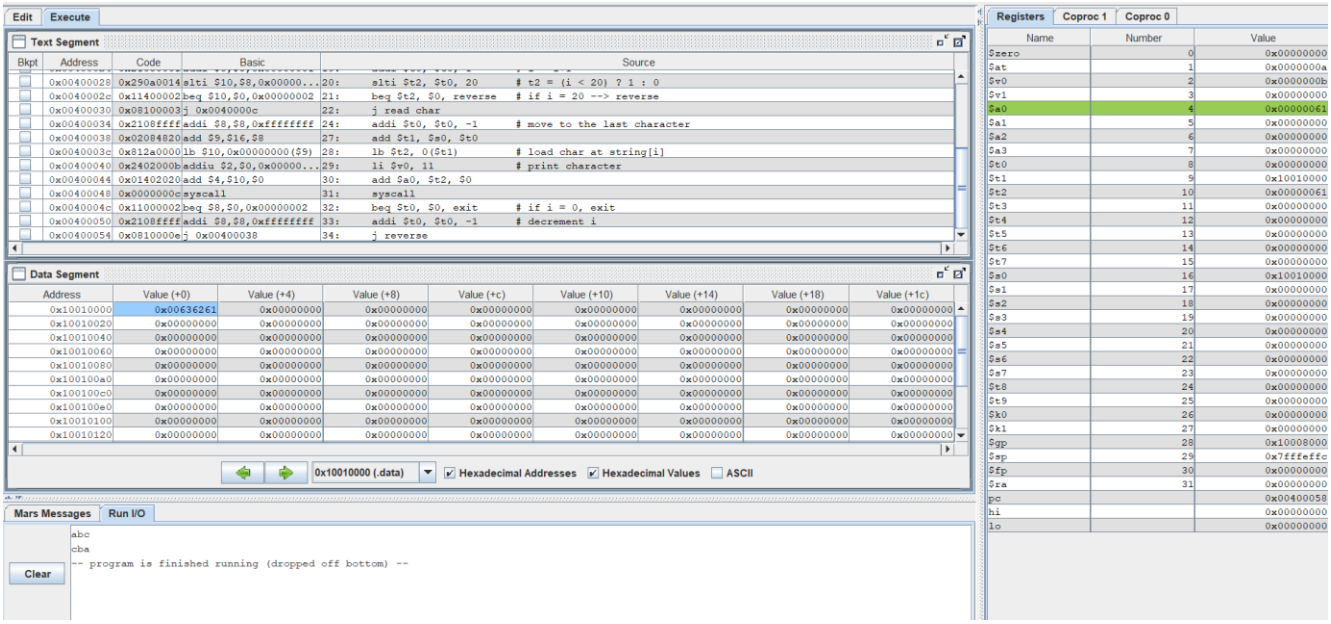
+print_length: Nạp 56 vào $v0 để gọi đến dịch vụ MessageDialogInt. Sau đó đưa các tham số: $a0 lưu địa chỉ chuỗi Message2, $a1 lưu giá trị phép tính trừ giá trị thanh $t0 với 1 (không tính kí tự kết thúc chuỗi '\0'; $t0 lưu độ dài của chuỗi)

5. Assignment 5

```
#Assignment 5, Lab 05, Tran Khanh Quynh
.data
string: .space 21
newline: .asciiz "\n"
.text
    li $t0, 0       # i = 0
    la $s0, string # load the address of string into $s0
read_char:
    li $v0, 12      # read character
    syscall

check_char:
    beq $v0, 10, reverse    # If the input char is enter then reverse
    add $t1, $s0, $t0       # t1 = address(string[i])
    sb $v0, 0($t1)          # store input char into string[i]
    addi $t0, $t0, 1        # i = i+1
    slti $t2, $t0, 20       # t2 = (i < 20) ? 1 : 0
    beq $t2, $0, reverse    # if i = 20 --> reverse
    j read_char

    addi $t0, $t0, -1       # move to the last character

    li $v0, 4
    la $a0, newline
    syscall
reverse:
    add $t1, $s0, $t0
    lb $t2, 0($t1)          # load char at string[i]
    li $v0, 11              # print character
    add $a0, $t2, $0
    syscall
    beq $t0, $0, exit       # if i = 0, exit
    addi $t0, $t0, -1       # decrement i
    j reverse
exit:
```

Kết quả chạy:

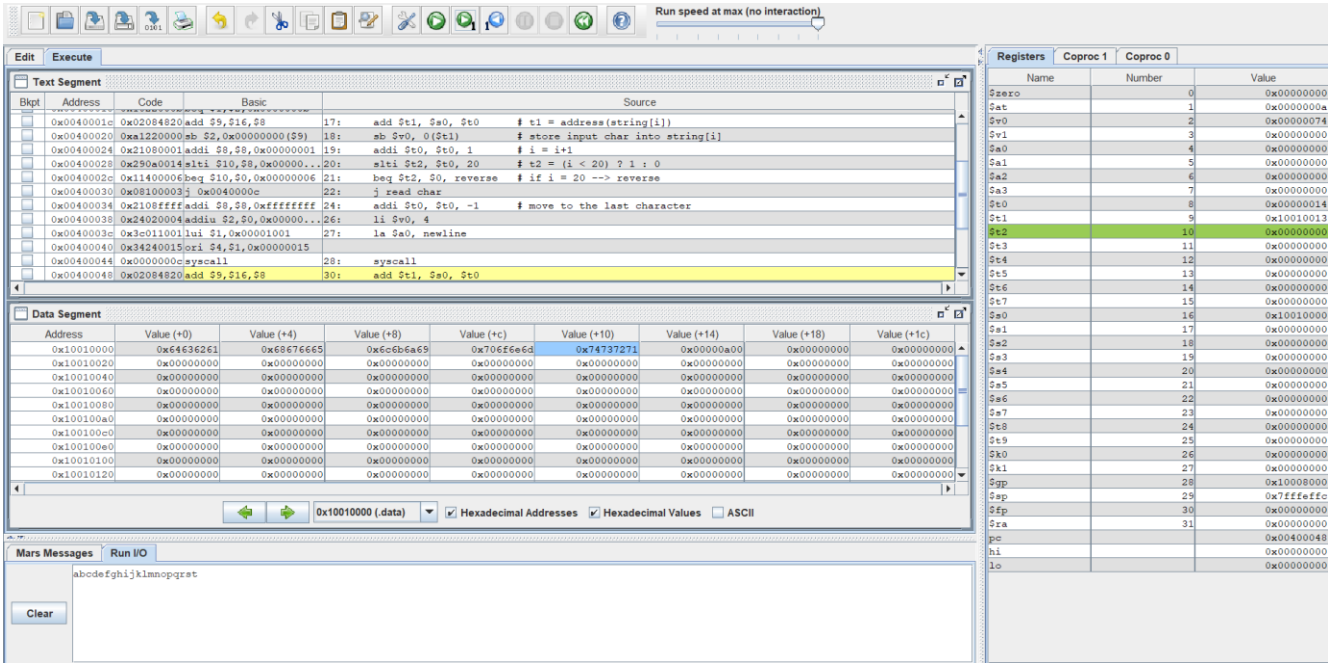
TH1: gặp enter



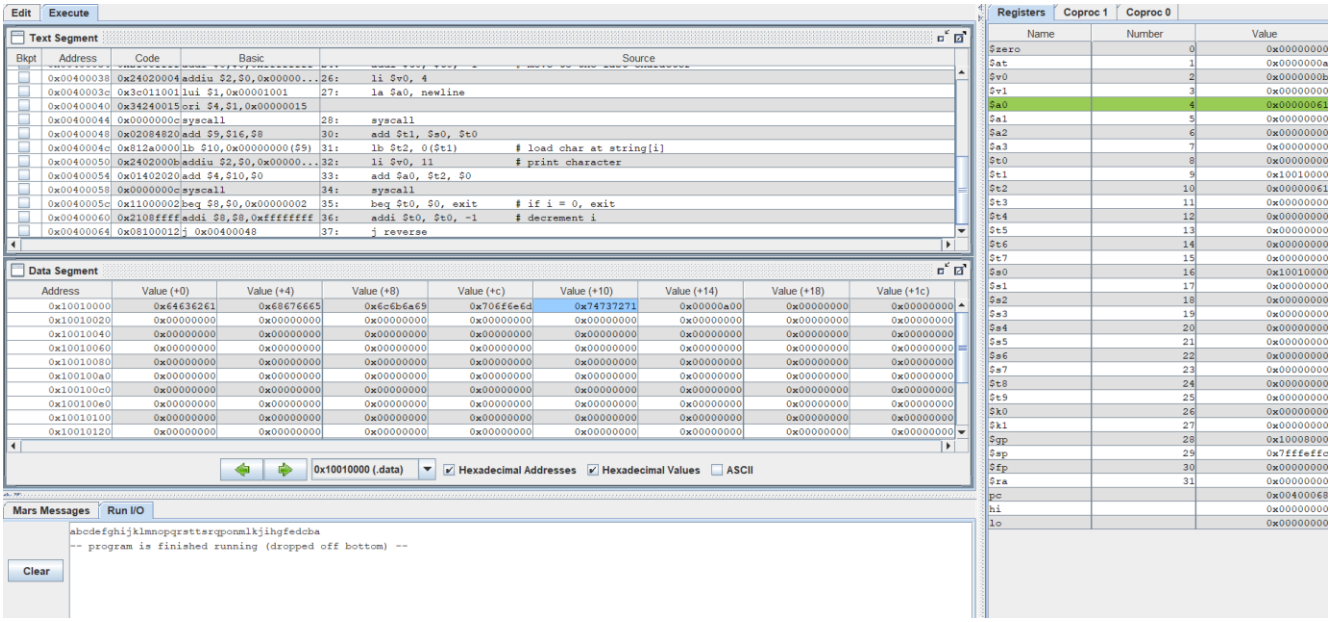
Lần lượt chạy chương trình, mỗi lần gọi syscall nhập vào các kí tự a, b, c và sau đó enter; chương trình sẽ in ra lần lượt c, b, a rồi kết thúc

TH2: 20 kí tự

Đầu vào:



Kết quả chạy:



Lần lượt chạy chương trình, mỗi lần gọi syscall nhập vào 20 kí tự; đến khi đủ 20 kí tự thì chương trình sẽ in ra lần lượt in ra các kí tự theo thứ tự ngược lại khi nhập vào.