THỰC HÀNH KIẾN TRÚC MÁY TÍNH

IT-3280

Họ tên	Thân Cát Ngọc Lan
MSSV	20225646

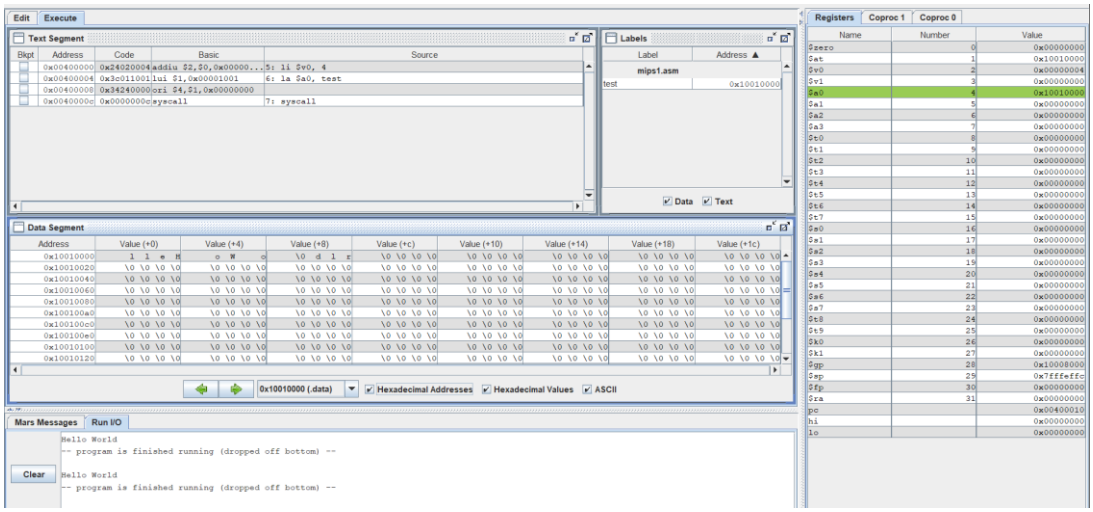
Assignment 1:

B1: Thanh ghi $v0 thay đổi giá trị từ 0x00000000 -> 0x00000004

B2: Thanh ghi $at thay đổi giá trị từ 0x00000000 -> 0x10010000

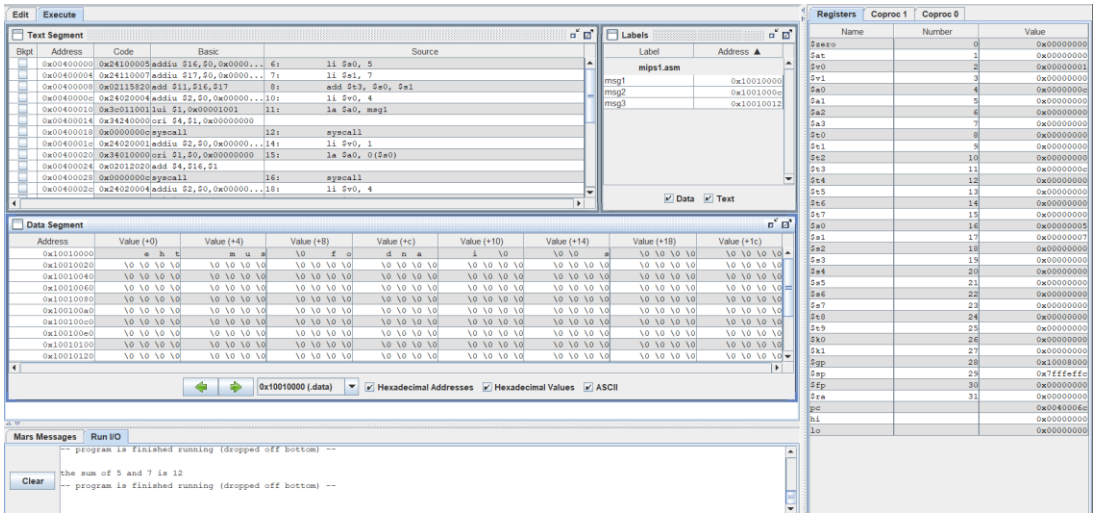
B3: Thanh ghi $a0 thay đổi giá trị từ 0x00000000 -> 0x10010000

- Giải thích: khi gọi lệnh syscall, HĐH sẽ xem giá trị của $v0
  Vì giá trị thanh ghi $v0 = 4 => thanh ghi $a0 sẽ là 1 string và in ra màn hình.



Assignment 2:

```
1   .data
2   msg1: .asciiz  "the sum of "
3   msg2: .asciiz  " and "
4   msg3: .asciiz " is "
5   .text
6           li $s0, 5
7           li $s1, 7
8           add $t3, $s0, $s1
9   #syscall nhap "the sum of"
10          li $v0, 4
11          la $a0, msg1
12          syscall
13  #syscall nhap $s0
14          li $v0, 1
15          la $a0, 0($s0)
16          syscall
17  #syscall nhap "and"
18          li $v0, 4
19          la $a0, msg2
20          syscall
21  #syscall nhap $s1
22          li $v0, 1
23          la $a0, 0($s1)
24          syscall
25  #sycall nhap "is"
26          li $v0, 4
27          la $a0, msg3
28          syscall
29  #syscall nhap tong cua (s0) vs (s1)
30          li $v0, 1
31          la $a0, 0($t3)
32          syscall
```



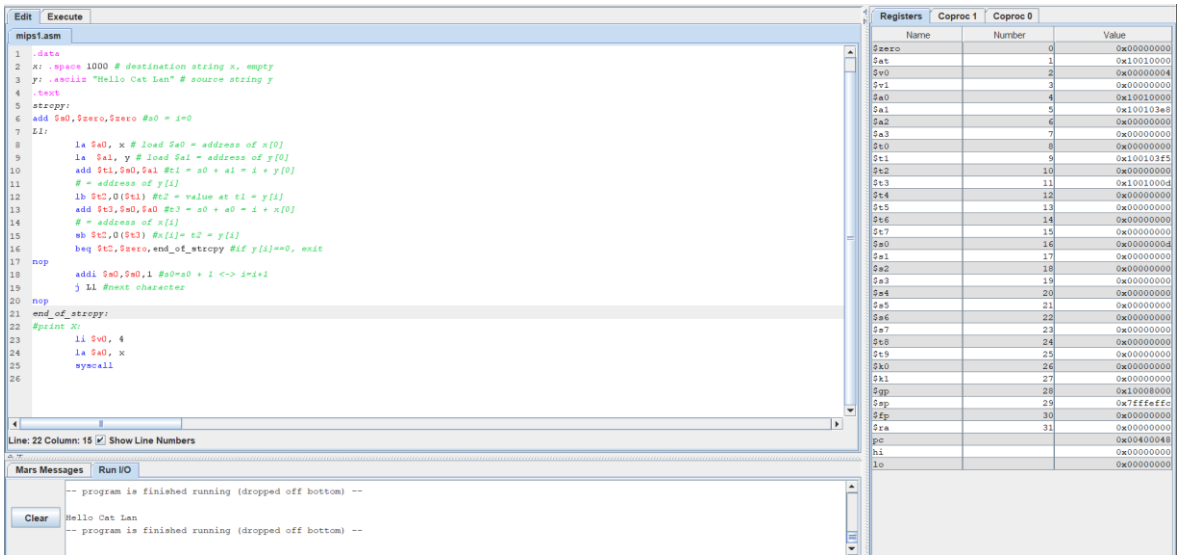
Giải thích: Mỗi khi lệnh syscall, HĐH sẽ xem giá trị của $v0.

- Khi $v0 = 4 => $a0 là 1 xâu
- Khi $v0 = 1 => $a0 là 1 số nguyên

Sau đó in ra màn hình.

⇨ Nhận xét: Kết quả chạy đúng và in ra đúng định dạng: **“the sum of … is …”**

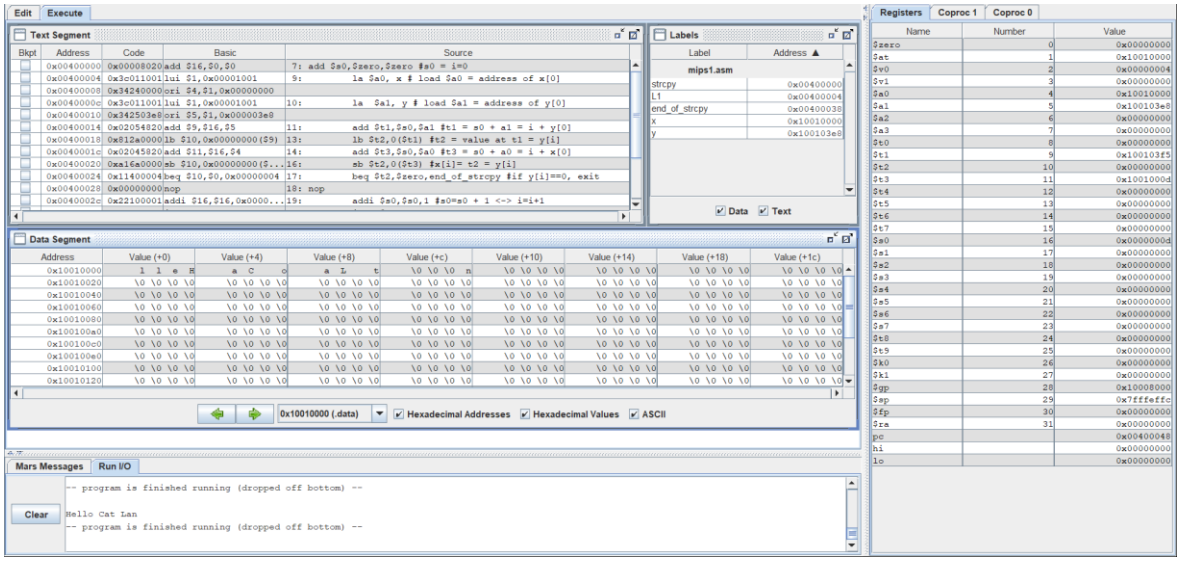
Assignment 3:



Giải thích:

- Lệnh la (dòng 8,9) -> nạp địa chỉ của xâu x,y
- Lệnh add $t1,$s0,$a1 -> trỏ vào y[i]
- Lệnh lb là lấy giá trị của y[i]
- Lệnh add $t3,$s0,$a0 -> trỏ vào x[i]
- Lệnh sb lấy giá trị đã lấy từ y[i] nạp vào x[i]

  ➔ Thực hiện vòng lặp để duyệt xâu



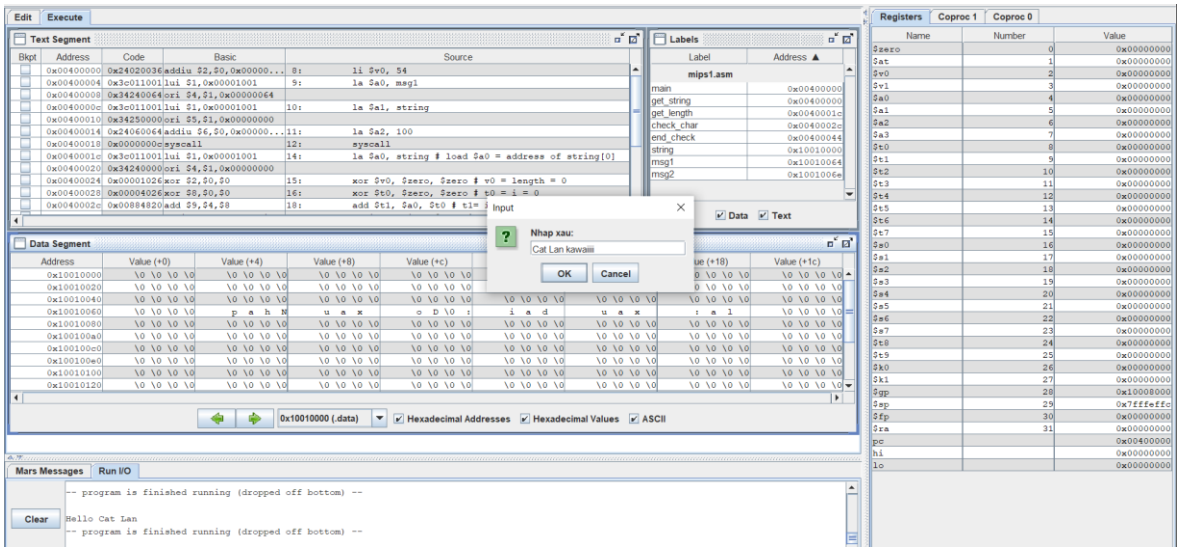
⇨ Kết quả chạy đúng với lí thuyết

Assignment 4:

mips1.asm

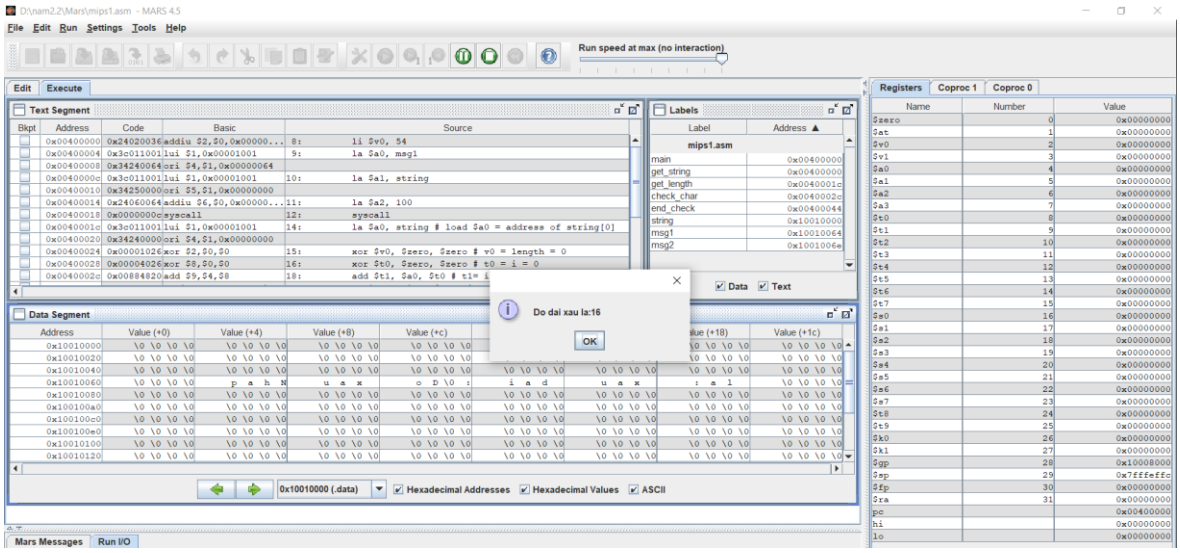
```
1  .data
2  string: .space 100
3  msg1: .asciiz "Nhap xau:"
4  msg2: .asciiz "Do dai xau la:"
5  .text
6  main:
7  get_string: #nhap xau vao string
8          li $v0, 54
9          la $a0, msg1
10         la $a1, string
11         la $a2, 100
12         syscall
13 get_length:
14         la $a0, string # load $a0 = address of string[0]
15         xor $v0, $zero, $zero # v0 = length = 0
16         xor $t0, $zero, $zero # t0 = i = 0
17 check_char:
18         add $t1, $a0, $t0 # t1= i +string[0]; t1  tro vaof dia chi cua string[i]
19         lb $t2, 0($t1) #nap gia tri string[i] vao t2 ; t2 = string[i]
20         beq $t2, $zero, end_check # if t2=null-> ket thuc
21         addi $v0, $v0, 1 # v0=v0+1
22         addi $t0, $t0, 1 #i=i+1
23         j check_char # vong lap moi
24 end_check:
25 #print length
26         addi $a1, $v0, -1 #do thua dau \n nen phai -1
27         li $v0, 56
28         la $a0, msg2
29         syscall
```

*Nhập xâu:



=> Hiển thị hộp thoại nhập xâu do $v0 = 54 và cho phép nhập 1 giá trị $a1 là 1 string.

*Hiển thị độ dài xâu trong hộp thoại:



- Do $v0 = 56, hiển thị tên hộp thoại và in ra màn hình kết quả độ dài xâu ($a1)

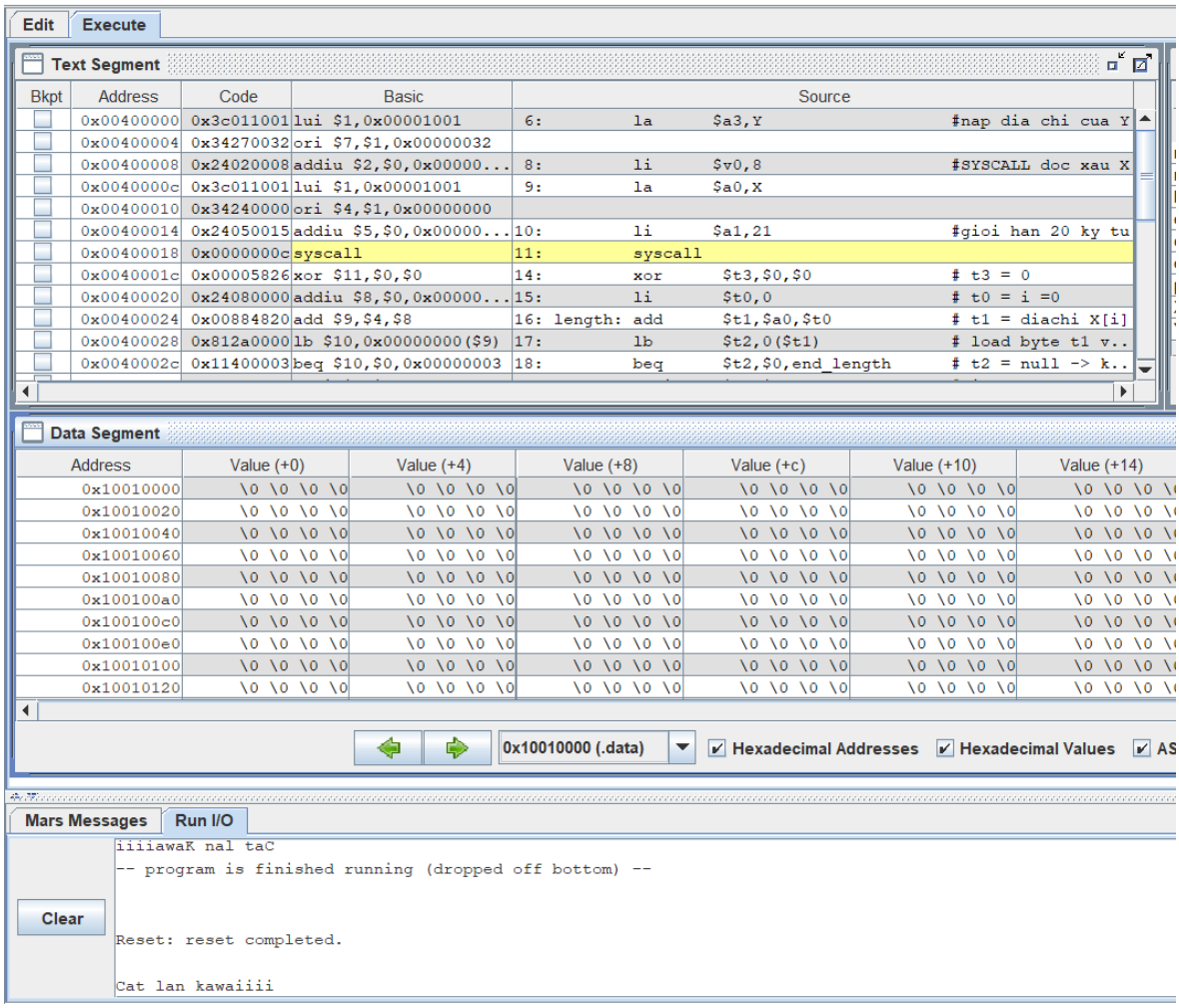
=> Kết quả chạy độ dài ra đúng

Assignment 5:

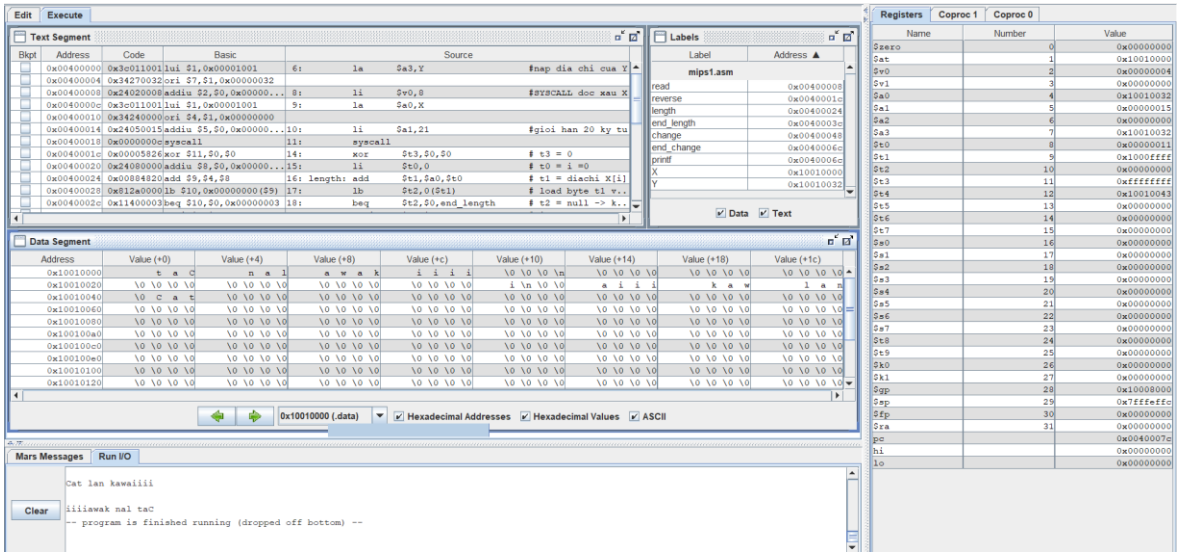
*Chương trình:

```
.data
X: .space 50
Y: .space 50
.text
        la      $a3,Y                   #nap dia chi cua Y
read:
        li      $v0,8                   #SYSCALL doc xau X
        la      $a0,X
        li      $a1,21                  #gioi han 20 ky tu
        syscall
reverse:
# dem do dai xau X
        xor     $t3,$0,$0               # t3 = 0
        li      $t0,0                   # t0 = i =0
length: add     $t1,$a0,$t0             # t1 = diachi X[i]
        lb      $t2,0($t1)              # load byte t1 vao t2
        beq     $t2,$0,end_length       # t2 = null -> ket thuc
        addi    $t0,$t0,1               # i++
        addi    $t3,$t3,1               # t3++
        j length
end_length:
        subi    $t3,$t3,1
        andi    $t0,$t0,0               # reset lai bien i
change: add     $t1,$a0,$t3             # t1 = diachi X[i]
        lb      $t2,0($t1)              # load byte t1 vao t2
        add     $t4,$a3,$t0             # t4 = dia chi Y[i]
        sb      $t2,0($t4)              # luu t2 vao t4
        beq     $t2,$0,end_change       # t2=null -> ket thuc
        addi    $t0,$t0,1               # i++
        subi    $t3,$t3,1               # t3-- (vi dang xet tu cuoi)
        j change
end_change:
printf: # In xau Y
        li $v0,4
        la $a0,Y
        syscall
```

*Nhập chuỗi



*Kết quả



⇨ In ra xâu đảo đúng