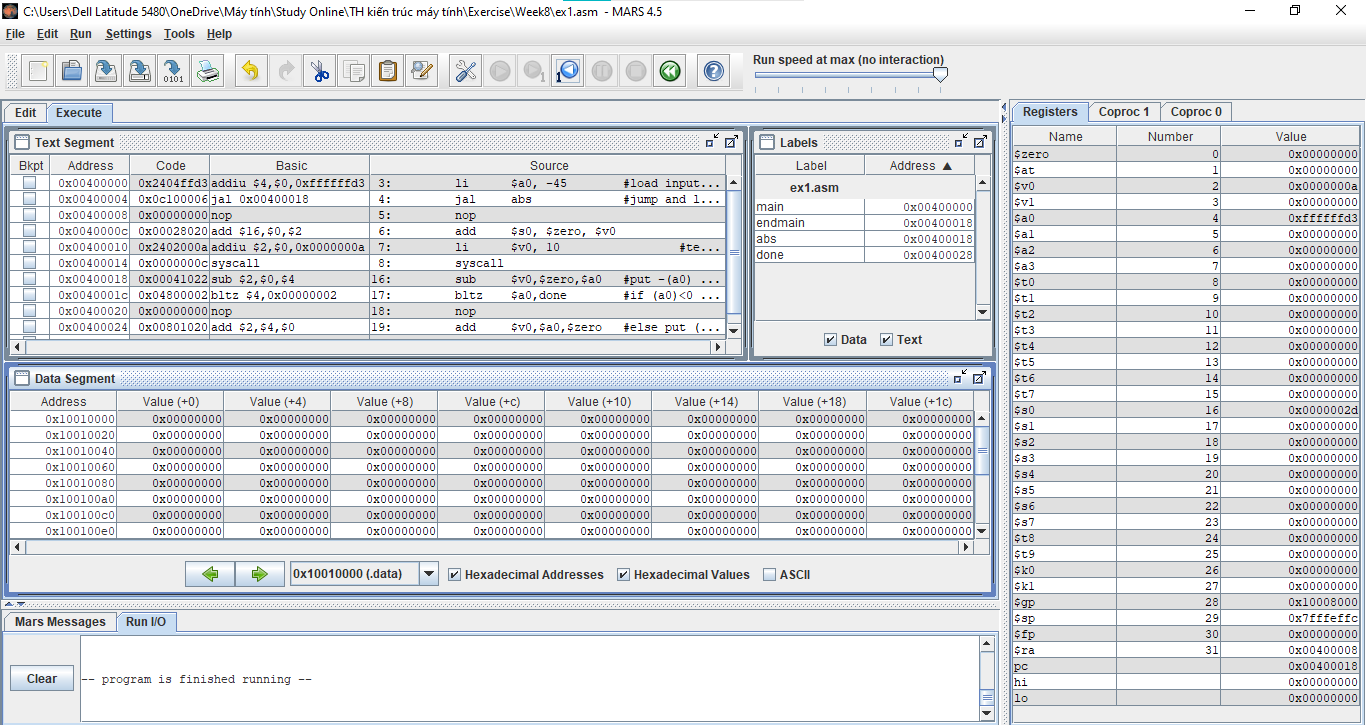
**BÁO CÁO MÔN THỰC HÀNH KIẾN TRÚC MÁY TÍNH – TUẦN 7**

Họ và Tên : Phí Quốc Hưng

MSSV : 20184270

**Assignment 1:**

* Sau khi chạy chương trình, ta thu được:



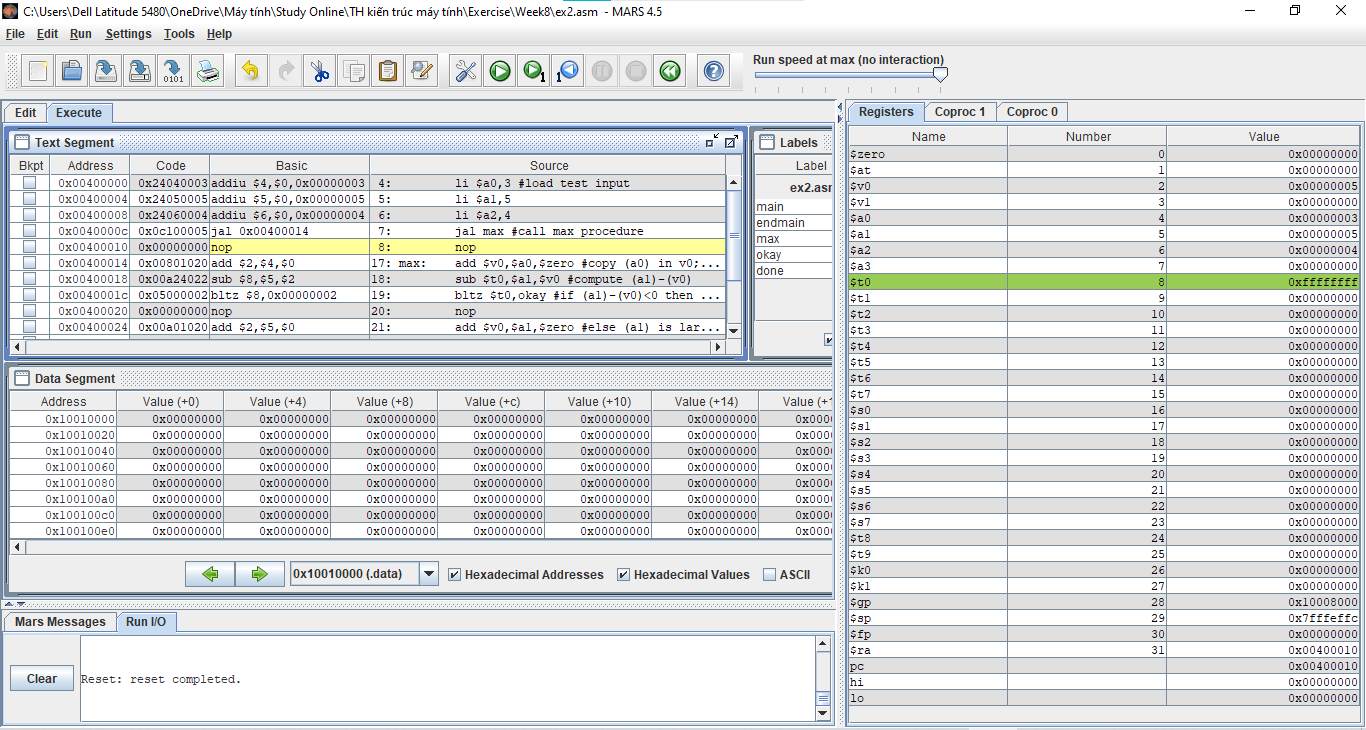
Nhận xét: - Khi chạy xong câu lệnh jal, thanh ghi $ra lưu giá trị là địa chỉ của lệnh tiếp theo (0x00400008), thanh ghi $pc chứa giá trị là địa chỉ của lệnh có nhãn là abs (0x0040018)

- Khi chạy xong câu lệnh jr, thanh ghi $pc chứa giá trị là giá trị của thanh ghi $ra

(0x00400008)

**Assignment 2:**

* Sau khi chạy chương trình, ta thu được:



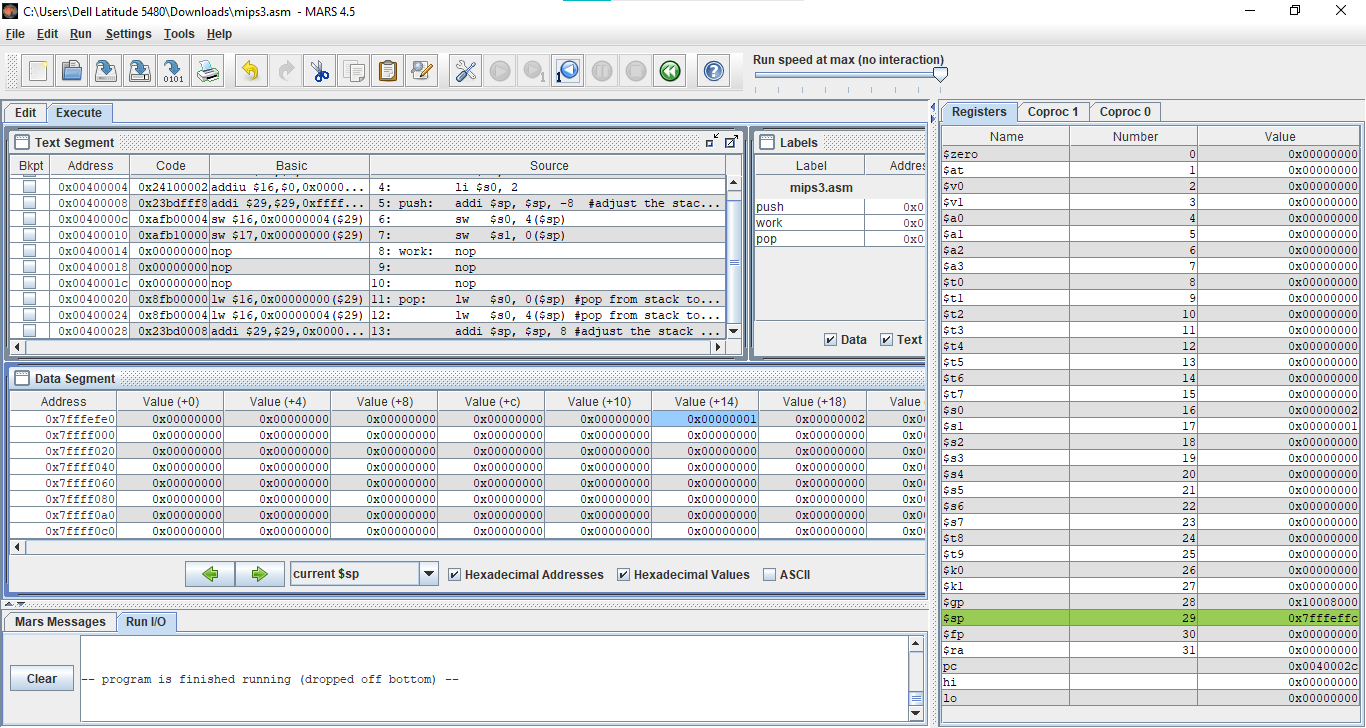
Nhận xét: - Khi chạy xong câu lệnh jal, thanh ghi $ra lưu giá trị là địa chỉ của lệnh tiếp theo (0x00400010), thanh ghi $pc chứa giá trị là địa chỉ của lệnh có nhãn là abs (0x0040018)

- Khi chạy xong câu lệnh jr, thanh ghi $pc chứa giá trị là giá trị của thanh ghi $ra

(0x00400018)

**Assignment 3:**

* Sau khi chạy chương trình, ta thu được:



Nhận xét: - Thanh ghi $sp (thay đổi 0x7fffeffc -> 0x7fffeff4 -> 0x7fffeffc)

- Địa chỉ 0x7fffeff8 lưu giá trị ban đầu $s0 = 2

-Địa chỉ 0x7fffeff4 lưu gía trị ban đầu $s1 = 1

**Assignment 4:**

* Sau khi chạy chương trình, ta thu được:

Ảnh có chứa văn bản, trong nhà, ảnh chụp màn hình

Mô tả được tạo tự động

Nhận xét: Nhận xét sự thay đổi của các thanh ghi:

- $pc: 0x00400000 -> 0x00400020 -> … -> 0x00400034 -> 0x0040004c -… -> 0x0040007c ->0x0040004c … ->

- $ra: 0x00000000 -> 0x00400004 -> 0x0040003 -> 0x00400080

- $sp: 0x7fffeffc -> 0x7fffeff4 -> 0x7fffefe8 -> 0x7fffefdc -> 0x7fffefd0 -> 0x7fffefc4 -> 0x7fffefb8 -> 0x7fffefac …

Tiếp theo thanh ghi $sp sẽ load dữ liệu trong thanh ghi $fp để pop dữ liệu

- $fp: 0x00000000 -> 0x7fffeffc -> 0x7fffeff4 -> 0x7fffefe8 -> 0x7fffefdc -> 0x7fffefd0 -> 0x7fffefc4 -> 0x7fffefb8 …

Thanh ghi $fp sẽ tăng dần trở lại để phục vụ việc pop dữ liệu của các stack cũ.

Ảnh có chứa bàn

Mô tả được tạo tự động

*Hình minh họa stack với n = 3*

**Assignment 5:**

* Chương trình:

#Laboratory Exercise 7, Assignment 5

.data

MaxMessage: .asciiz "The largest value is "

MinMessage: .asciiz "The smallest value is "

MaxIndex: .asciiz "The largest element is stored in $s"

MinIndex: .asciiz "The smallest element is stored in $s"

.text

main: li $s0, 5

li $s1, 3

li $s2, 2

li $s3, 1

li $s4, 8

li $s5, 10

li $s6, -4

li $s7, 7

jal init # call max procedure

nop

li $v0, 56

la $a0, MaxIndex # print index of max element

add $a1,$t8,$zero

syscall

la $a0, MaxMessage # print max value

add $a1,$t0,$zero

syscall

la $a0, MinIndex # print index of min value

add $a1,$t9,$zero

syscall

la $a0, MinMessage # print min value

add $a1,$t1,$zero

syscall

li $v0, 10 # exit program

syscall

endmain:

swapMax:add $t0,$t3,$zero # set Max = $t3

add $t8,$t2,$zero # set index of Max = $t2

jr $ra

swapMin:add $t1,$t3,$zero # set Min = $t3

add $t9,$t2,$zero # set index of Min = #$t2

jr $ra

init: add $fp,$sp,$zero # save address of origin sp

addi $sp,$sp, -32 # create space for stack

sw $s1, 0($sp)

sw $s2, 4($sp)

sw $s3, 8($sp)

sw $s4, 12($sp)

sw $s5, 16($sp)

sw $s6, 20($sp)

sw $s7, 24($sp)

sw $ra, 28($sp) # save $ra for main

add $t0,$s0,$zero # set Max = $s0

add $t1,$s0,$zero # set Min = $s0

li $t8, 0 # set index of Max to 0

li $t9, 0 # set index of Min to 0

li $t2, 0 # set current index to 0

max\_min:addi $sp,$sp,4

lw $t3,-4($sp)

sub $t4, $sp, $fp # check if meet $ra

beq $t4,$zero, done # if true, done

addi $t2,$t2,1 # increase index

sub $t4,$t0,$t3 # cal Max - $t3

bltzal $t4, swapMax # if Max < $t3, swap Max

sub $t4,$t3,$t1 # cal $t3 - Min

bltzal $t4, swapMin # if $t3 < Min, swap Min

j max\_min # repeat

done: lw $ra, -4($sp) # load #$ra

jr $ra # return to calling program