**Bài thực hành số 5**

Lớp: 147799 – Học phần: Thực Hành Kiến Trúc Máy Tính

Họ và tên: Lương Văn Khanh - MSSV: 20225728

**Asg1:**

* Code :

#Laboratory Exercise 7 Home Assignmen

start:

main: li $a0, -45 #load input parameter

jal abs #jump and link to abs procedure

nop

add $s0, $zero, $v0

li $v0, 10 #terminate

syscall

endmain:

#--------------------------------------------------------------------

# function abs

# param[in] $a0 the interger need to be gained the absolute

# return $v0 absolute value

#--------------------------------------------------------------------

abs:

sub $v0,$zero,$a0 #put -(a0) in v0; in case (a0)<0

bltz $a0,done #if (a0)<0 then done

nop

add $v0,$a0,$zero #else put (a0) in v0

done:

jr $ra

A screenshot of a computer

Description automatically generated

* Sự thay đổi của các thanh ghi:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trạng thái chạy |  |  |  |  |  |
| $a0 | $v0 | $s0 | $ra | $pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00400000 |
| Chạy lệnh li $a0 | 0xffffffd3 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00400004 |
| Chạy lệnh jal | 0xffffffd3 | 0x0000002d | 0x00000000 | 0x00400008 | 0x00400018 |
| Chạy lệnh sub | 0xffffffd3 | 0x0000002d | 0x00000000 | 0x00400008 | 0x0040001c |
| Chạy lệnh bltz | 0xffffffd3 | 0x0000002d | 0x00000000 | 0x00400008 | 0x00400028 |
| Chạy lệnh jr $ra | 0xffffffd3 | 0x0000002d | 0x00000000 | 0x00400008 | 0x00400008 |
| Chạy lệnh add $s0 | 0xffffffd3 | 0x0000002d | 0x0000002d | 0x00400008 | 0x0040000c |
| Chạy lệnh syscall | 0xffffffd3 | 0x0000000a | 0x0000002d | 0x00400008 | 0x00400018 |

* Giải thích code : Đề bài yêu cầu tính giá trị tuyệt đối của 1 số nguyên.Giá trị đầu vào được ghi ra thanh ghi $a0, còn kết quả in ra thanh ghi $s0.Dòng số 4 jump and link đến chương trình con abs để tính giá trị tuyệt đối. Sau khi thực hiện chương trình con xong, con trỏ trỏ đến $ra (địa chỉ trở về) để trở về sau khi thực hiện xong chương trình main.

**Asg2:**

* Code:

#Laboratory Exercise 7, Home Assignment 2

.text

main: li $a0,1 #load test input

li $a1,7

li $a2,4

jal max #call max procedure

nop

li $v0, 10 #terminate

syscall

endmain:

#---------------------------------------------------------------------

#Procedure max: find the largest of three integers

#param[in] $a0 integers

#param[in] $a1 integers

#param[in] $a2 integers

#return $v0 the largest value

#---------------------------------------------------------------------

max: add $v0,$a0,$zero #copy (a0) in v0; largest so far

sub $t0,$a1,$v0 #compute (a1)-(v0)

bltz $t0,okay #if (a1)-(v0)<0 then no change

nop

add $v0,$a1,$zero #else (a1) is largest thus far

okay: sub $t0,$a2,$v0 #compute (a2)-(v0)

bltz $t0,done #if (a2)-(v0)<0 then no change

nop

add $v0,$a2,$zero #else (a2) is largest overall

done: jr $ra #return to calling program

Giải thích code : Chương trình so sánh giá trị trong 3 thanh ghi $a0,$a1,$a2. $v0 là thanh ghi lưu giá trị lớn nhất

Đầu tiên lấy $v0=$a0

Lấy a1-v0 nếu bé hơn 0 thì gán v0=a1

Tiếp tục a2-v0 nếu bé hơn 0 thì gán v0=a2

* Kết quả : v0=7

A screenshot of a computer

Description automatically generated

**Asg3:**

* Code:

#Laboratory Exercise 7, Home Assignment 3

.text

li $s0,4

li $s1,5

push: addi $sp,$sp,-8 #adjust the stack pointer

sw $s0,4($sp) #push $s0 to stack

sw $s1,0($sp) #push $s1 to stack

work: nop

nop

nop

pop: lw $s0,0($sp) #pop from stack to $s0

lw $s1,4($sp) #pop from stack to $s1

addi $sp,$sp,8 #adjust the stack pointer

Giải thích code: Đảo vị trí nội dung thanh ghi $s0 và $s1. Sử dụng nguyên tắc LIFO (Last In First Out) trong Stack. Ta sẽ sử dụng con trỏ sp (stack pointer) để push lần lượt nội dung 2 thanh ghi $s0 và $s1 vào trong stack. Sau đó cũng dung sp để pop lần lượt nội dung ra thanh ghi $s1 và $s0.Khởi tạo s0=4;s1=5.

* Kết quả chạy được:

A screenshot of a table

Description automatically generated

**Asg4:**

* Code:

#Laboratory Exercise 7, Home Assignment 4

.data

Message: .asciiz "Ket qua tinh giai thua la: "

.text

main: jal WARP

print: add $a1, $v0, $zero # $a0 = result from N!

li $v0, 56

la $a0, Message

syscall

quit: li $v0, 10 #terminate

syscall

endmain:

#---------------------------------------------------------------------

#Procedure WARP: assign value and call FACT

#---------------------------------------------------------------------

WARP: sw $fp,-4($sp) #save frame pointer (1)

addi $fp,$sp,0 #new frame pointer point to the top (2)

addi $sp,$sp,-8 #adjust stack pointer (3)

sw $ra,0($sp) #save return address (4)

li $a0,3 #load test input N

jal FACT #call fact procedure

nop

lw $ra,0($sp) #restore return address (5)

addi $sp,$fp,0 #return stack pointer (6)

lw $fp,-4($sp) #return frame pointer (7)

jr $ra

wrap\_end:

#-----------------------------------------------------------

#Procedure FACT: compute N!

#param[in] $a0 integer N

#return $v0 the largest value

#-----------------------------------------------------------

FACT: sw $fp,-4($sp) #save frame pointer

addi $fp,$sp,0 #new frame pointer point to top

addi $sp,$sp,-12 #allocate space for $fp,$ra

sw $ra,4($sp) #save return address

sw $a0,0($sp) #save $a0 register

slti $t0,$a0,2 #if input argument N < 2

beq $t0,$zero,recursive#if it is false ((a0 = N) >

nop

li $v0,1 #return the result N!=1

j done

nop

recursive:

addi $a0,$a0,-1 #adjust input argument

jal FACT #recursive call

nop

lw $v1,0($sp) #load a0

mult $v1,$v0 #compute the result

mflo $v0

done: lw $ra,4($sp) #restore return address

lw $a0,0($sp) #restore a0

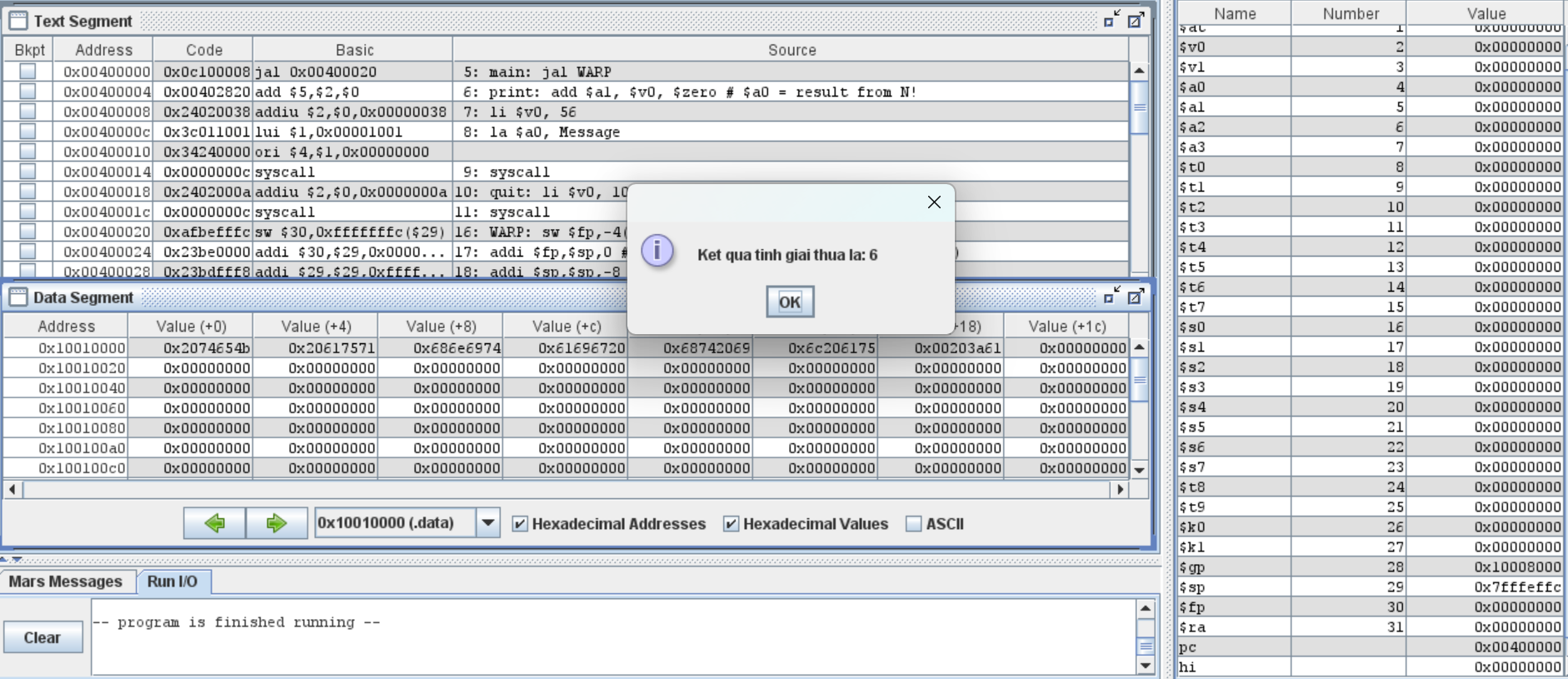
addi $sp,$fp,0 #restore stack pointer

lw $fp,-4($sp) #restore frame pointer

jr $ra #jump to calling

fact\_end:

* Kết quả chạy :



* Hình vẽ minh họa Stack:

Trước khi push:

A diagram of a number

Description automatically generated with medium confidence

Sau khi pop:

A screenshot of a graph

Description automatically generated

**Asg5:**

* Code:

.data

Message1: .asciiz "The largest element is stored in $s"

Message2: .asciiz ", largest value is "

Message3: .asciiz "\nThe smallest element is stored in $s"

Message4: .asciiz ", smallest value is "

.text

li $s0, 9

li $s1, 12

li $s2, 2

li $s3, 0

li $s4, -4

li $s5, 17

li $s6, -1

li $s7, -17

main:

sw $s0, -4($sp) #dat vao ngan xep cac gia tri tuong ung

sw $s1, -12($sp)

sw $s2, -20($sp)

sw $s3, -28($sp)

sw $s4, -36($sp)

sw $s5, -44($sp)

sw $s6, -52($sp)

sw $s7, -60($sp)

add $t0, $zero, $zero # khoi tao bien i = 0

jal loop

print\_max:

li $v0, 4

la $a0, Message1

syscall

li $v0, 1

add $a0, $t2,$zero

syscall

li $v0, 4

la $a0, Message2

syscall

li $v0, 1

add $a0, $s0, $zero

syscall

print\_min:

li $v0, 4

la $a0, Message3

syscall

li $v0, 1

add $a0, $t3, $zero

syscall

li $v0, 4

la $a0, Message4

syscall

li $v0, 1

add $a0, $s1, $zero

syscall

exit:

li $v0, 10

syscall

loop:

sw $t0, 0($sp)

addi $t0, $t0, 1 # tang bien don vi i = i + 1

slti $t1, $t0, 8

bne $t1, $zero, move\_to\_next\_number

lw $s0, -4($sp)

lw $t2, 0($sp) #vi tri

find\_max:

lw $t3, 4($sp) #phan tu ke tiep sau max

lw $t4, 8($sp) #vi tri cua phan tu do

slt $t1, $t3, $s0 #so sanh the tim max

bne $t1, $zero, adjust\_max

add $s0, $t3, $zero

add $t2, $t4, $zero

adjust\_max:

addi $sp,$sp,8 # di chuyen con tro stack

bgtz $t4, find\_max # Neu i > 0 thi loop

lw $s1,-4($sp) # khoi tao max ban dau

lw $t3,0($sp)

find\_min:

lw $t5, -8($sp)

lw $t4,-12($sp)

slt $t1, $s1, $t4

bne $t1, $zero, adjust\_min

add $s1, $t4, $zero

add $t3, $t5, $zero

adjust\_min:

addi $sp, $sp, -8

slti $t1, $t5, 7

bne $t1, $zero, find\_min

jr $ra

move\_to\_next\_number:

addi $sp, $sp, -8

j loop

* Kết quả chạy :

A screenshot of a computer

Description automatically generated