Họ tên: Kiều Đăng Nam

MSSV: 20176830

**Báo cáo thực hành tuần 7**

**Asignment 1**

1. Source code:

#Laboratory Exercise 7 Assignment 1

.text

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 value

# 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

1. Kết quả chạy chương trình

* Sự thay đổi của thanh ghi $pc và $ra:

Khi $pc chỉ đến lệnh jal -> $ra = 0x00400008



Sau khi kết thúc hàm abs, giá trị $pc = $ra (theo lệnh jr $ra)



* Kết quả:

Giá trị tuyệt đối của – 45 = $s0 = 0x0000002d -> Đúng



* Thay đổi đầu vào = 8

Kết quả: $s0 = 0x00000008 -> Đúng



**Asignment 2:**

1. Source code

#Laboratory Exercise 7, Assignment 2

.text

main:

li $a0, 1 #load test input

li $a1, 3

li $a2, 5

jal max #call max procedure

nop

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:

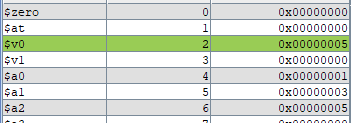
jr $ra #return to calling program

1. Kết quả chạy:

Trong trường hợp này khi chạy đến câu lệnh bltz trong chương trình max, do $a1 = 3 > $v0 = 1 nên thực hiện tiếp lệnh gán lại giá trị cho $v0 (max), sau đó mới chuyển xuống chương trình con okay để thực hiện tiếp so sánh $v0 với $a2

Sau đó thực hiện lệnh jal để nhảy về lệnh nop trong main

Kết quả: max = 5



**Assignment 3**

1. Source code

#Laboratory Exercise 7, Assignment 3

.text

init: li $s0,3

li $s1,1

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

1. Kết quả:

* Giá trị ban đầu: $s0 = 3, $s1 = 1



* Sau khi chạy chương trình: $s0 = 1, $s1 = 3



* Giải thích:

+ Khi chạy hàm push, 2 giá trị của $s0, $s1 được push vào trong stack

+ Khi chạy hàm pop, 2 giá trị của $s0, $s1 được pop ra ngược với thứ tự push

=> Giá trị của 2 thanh ghi được hoán đổi

**Asignment 4**

1. Source Code

#Laboratory Exercise 7 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 $a0,6 #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 stackâ€™s top

addi $sp,$sp,-12 #allocate space for $fp,$ra,$a0 in stack

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) >=2)

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

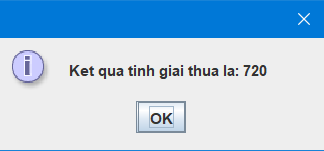
lw $fp,-4($sp) #restore frame pointer

jr $ra #jump to calling

fact\_end:

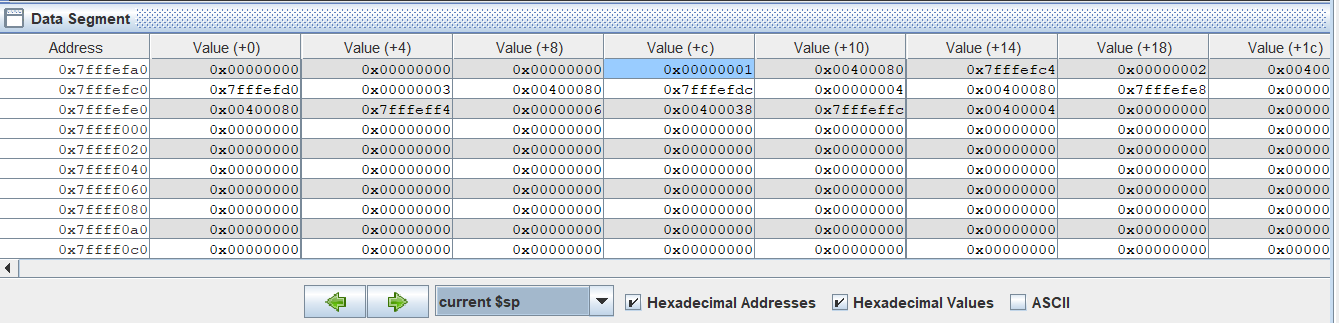
1. Kết quả:

* Chạy thử với input n = 6: - > kết quả = 720 -> Chương trình chạy đúng



* Sự thay đổi các thanh ghi:

Thanh ghi $sp:



* Khi n = 3

Bảng thể hiện giá trị ngăn xếp:

|  |
| --- |
| $fp = 0x7fffefcd |
| $a0 = 0x0000001 |
| $ra = 0x0040080 |
| $fp = 0x7fffefe8 |
| $a0 = 0x0000002 |
| $ra = 0x0040080 |
| $fp = 0x7fffeff4 |
| $a0 = 0x0000003 |
| $ra = 0x0040038 |
| $fp = 0x7fffeffc |
| $ra = 0x0040004 |
|  |

**Assignment 5**

Lấy ví dụ với array như sau: [-3, 12, -5, 48, 4, 3, -6, 7 ]

Giá trị lớn nhất: 48, giá trị nhỏ nhất -6

1. Hàm main

#Laboratory Exercise 7, Assignment 5

.data

MesOfMax: .asciiz "The largest value is "

MesOfMin: .asciiz "The smallest value is "

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

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

.text

main: li $s0, -3

li $s1, 12

li $s2, -5

li $s3, 48

li $s4, 4

li $s5, 3

li $s6, -6

li $s7, 7

jal init #call max procedure

nop

li $v0, 56

la $a0, MesMaxIndex #Print index of max element

add $a1,$t8,$zero

syscall

la $a0, MesOfMax #Print max value

add $a1,$t0,$zero

syscall

la $a0, MesMinIndex # Print index of min value

add $a1,$t9,$zero

syscall

la $a0, MesOfMin # Print min value

add $a1,$t1,$zero

syscall

li $v0, 10 # exit program

syscall

endmain:

1. Hàm swapMax: gán giá trị max là giá trị đang xét hiện tại, lưu lại index của nó

swapMax:

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

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

jr $ra

1. Hàm swapMin: gán giá trị min là giá trị đang xét hiện tại, lưu lại index của nó.

swapMin:

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

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

jr $ra

1. Hàm init: hàm khởi tạo giá trị cho array cần xử lí, lưu chúng vào stack, chứa hàm con là max\_min (gọi hàm tìm giá trị max, min)

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

1. Hàm max\_min: Tìm giá trị max, min trong stack

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

Idea bài toán:

Gọi hàm main:

Gọi hàm init:

Push các phần tử của array vào stack

Gọi hàm max\_min (tìm max, min)

Nếu tìm thấy max lớn hơn max hiện tại -> swapMax

Nếu tìm thấy min nhỏ hơn min hiện tại -> swapMin

Hiển thị giá trị max, maxIdx, min, minIdx ra màn hình.

Kết quả hiển thị:

