**­­­­Computer Architecture Lab Report Week 7**

**Full name: Nguyễn Đức Đại Dương**

**Student ID: 20225616**

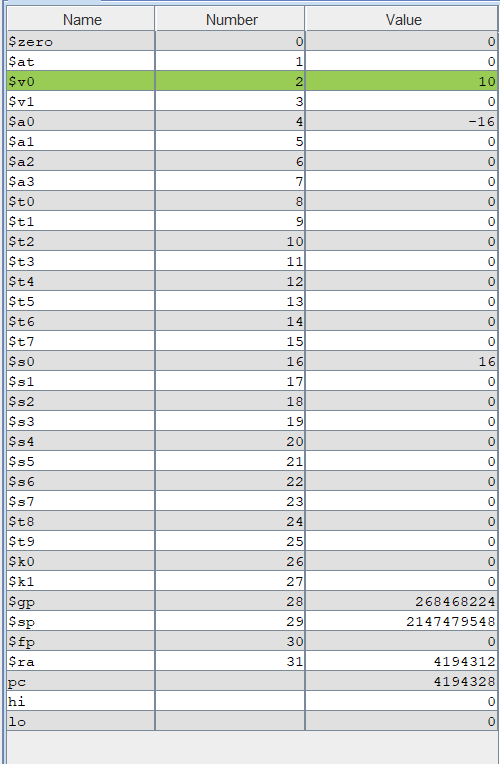
Assignment 1

Code:

A screenshot of a computer code

Description automatically generated

Kết quả:



Kết quả được lưu vào thanh ghi $s0 đúng với lí thuyết

Assignment 2

Code:

A screenshot of a computer

Description automatically generated

Kết quả

A table with numbers and letters

Description automatically generated

Xét 3 số 16, -10, 32 . Giá trị max được lưu vào thanh ghi $s1 là 32

=>Kết quả đúng với lí thuyết

Assignment 3

Code:

A screenshot of a computer

Description automatically generated

Xét trường hợp $s0 = 15, $s1 = -4

Kết quả:

A screenshot of a computer

Description automatically generated

Giá trị 2 thanh ghi $s0, $s1 đã bị hoán đổi

=>Kết quả đúng với lí thuyết

Assignment 4

Code:

.data

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

.text

main:

jal WARP

print:

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

li $v0, 56

la $a0, Message

syscall

quit:

li $v0, 10 #terminate

syscall

endmain:

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 stack’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:

Kết quả: tính 3!

A screenshot of a computer

Description automatically generated

=>Kết quả đúng với lí thuyết

Assignment 5

Code:

.data

largest: .asciiz "Largest: "

smallest: .asciiz "\nSmallest: "

comma: .asciiz ","

.text

main:

li $s0, 1

li $s1, 4

li $s2, -6

li $s3, 0

li $s4, 10

li $s5, 2

li $s6, -65

li $s7, 8

jal Duong

nop

print:

#print Largest

li $v0,4

la $a0, largest

syscall

#print max

add $a0, $t0, $zero

li $v0, 1

syscall

#print ,

li $v0, 4

la $a0, comma

syscall

#print the register of max

add $a0, $t5, $zero

li $v0,1

syscall

#print Smallest

li $v0,4

la $a0, smallest

syscall

#print min

add $a0, $t1, $zero

li $v0, 1

syscall

#print ,

li $v0, 4

la $a0, comma

syscall

#print the register of min

add $a0, $t6, $zero

li $v0,1

syscall

endmain:

li $v0, 10

syscall

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

# $t0 = max

#$t1 = min

#index of max = $t5

#index of min = $t6

#$v0 the largest value

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

swapMax:

add $t0,$t3,$zero

add $t5, $t2, $zero

jr $ra

swapMin:

add $t1,$t3,$zero

add $t6, $t2, $zero

jr $ra

Duong:

add $t9,$sp,$zero #save address of orgin $sp

addi $sp,$sp, -32

#sw $s0,0($sp)

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

add $t1,$s0,$zero

li $t5, 0

li $t6 ,0

li $t2,0

findmaxmin:

addi $sp,$sp,4

lw $t3,-4($sp)

sub $t4,$sp,$t9

beq $t4,$zero,done

nop

addi $t2,$t2,1

sub $t4,$t0,$t3

bltzal $t4,swapMax

nop

sub $t4,$t3,$t1

bltzal $t4,swapMin

nop

j findmaxmin

done:

lw $ra,-4($sp)

jr $ra

A screenshot of a table

Description automatically generated

Kết quả:

A white background with black text

Description automatically generated

=>Đúng với lí thuyết