Bài thực hành 6

Assignment 1:

Mã chương trình :

.data

A: .word 4, 6, -3, 1, -3

.text

main: la $a0,A

li $a1,5

j mspfx

nop

continue:

lock: j lock

nop

end\_of\_main:

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

#Procedure mspfx

# @brief find the maximum-sum prefix in a list of integers

# @param[in] a0 the base address of this list(A) need to be processed

# @param[in] a1 the number of elements in list(A)

# @param[out] v0 the length of sub-array of A in which max sum reachs.

# @param[out] v1 the max sum of a certain sub-array

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

#Procedure mspfx

#function: find the maximum-sum prefix in a list of integers

#the base address of this list(A) in $a0 and the number of

#elements is stored in a1

mspfx: addi $v0,$zero,0 #initialize length in $v0 to 0

addi $v1,$zero,0 #initialize max sum in $v1to 0

addi $t0,$zero,0 #initialize index i in $t0 to 0

addi $t1,$zero,0 #initialize running sum in $t1 to 0

loop: add $t2,$t0,$t0 #put 2i in $t2

add $t2,$t2,$t2 #put 4i in $t2

add $t3,$t2,$a0 #put 4i+A (address of A[i]) in $t3

lw $t4,0($t3) #load A[i] from mem(t3) into $t4

add $t1,$t1,$t4 #add A[i] to running sum in $t1

slt $t5,$v1,$t1 #set $t5 to 1 if max sum < new sum

bne $t5,$zero,mdfy #if max sum is less, modify results

j test #done?

mdfy: addi $v0,$t0,1 #new max-sum prefix has length i+1

addi $v1,$t1,0 #new max sum is the running sum

test: addi $t0,$t0,1 #advance the index i

slt $t5,$t0,$a1 #set $t5 to 1 if i<n

bne $t5,$zero,loop #repeat if i<n

done: j continue

mspfx\_end:

Ban đầu ,khởi tạo một mảng gồm 5 phần tử A = {4,6,-3,1,-3}

Khởi tạo vị trí tiền tố max là 0(A[0]) và tổng tiền tố lớn nhất là max\_sum = 0

Ảnh có chứa bàn

Mô tả được tạo tự động

Tại vị trí A[1] thì tiền tố bằng 4

Ảnh có chứa bàn

Mô tả được tạo tự động

Tại vị trí A[2] thì tiền tố bằng 10

Ảnh có chứa bàn

Mô tả được tạo tự động

Xét mảng A = {4,6,-3,1,-3} có tiền tố lớn nhất bằng 10=>Đúng

**Assignment 2**

**TH1 :Tăng dần**

.data  
A: .word 7, -2, 5, 1, 5,6,7,3,6,8,8,59,5  
Aend: .word  
.text  
main:  
la $a0,A #$a0 = Address(A[0])  
la $a1,Aend  
addi $a1,$a1,-4 #$a1 = Address(A[n-1])  
j sort #sort  
after\_sort:  
li $v0, 10#exit  
syscall  
end\_main:  
#--------------------------------------------------------------  
#procedure sort (ascending selection sort using pointer)  
#register usage in sort program  
#$a0 pointer to the first element in unsorted part  
#$a1 pointer to the last element in unsorted part  
#$t0 temporary place for value of last element  
#$v0 pointer to max element in unsorted part  
#$v1 value of max element in unsorted part  
#--------------------------------------------------------------  
sort:  
beq $a0,$a1,done#single element list is sorted  
j max#call the max procedure  
after\_max:  
lw $t0,0($a1)#load last element into $t0  
sw $t0,0($v0)#copy last element to max location  
sw $v1,0($a1)#copy max value to last element  
addi $a1,$a1,-4 #decrement pointer to last element  
j sort#repeat sort for smaller list  
done:  
j after\_sort  
#------------------------------------------------------------------------  
#Procedure max#function: fax the value and address of max element in the list  
#$a0 pointer to first element  
#$a1 pointer to last element  
#------------------------------------------------------------------------  
max:  
addi $v0,$a0,0 #init max pointer to first element  
lw $v1,0($v0) #init max value to first value  
addi $t0,$a0,0 #init next pointer to first  
loop:  
beq $t0,$a1,ret #if next=last, return  
addi $t0,$t0,4 #advance to next element  
lw $t1,0($t0) #load next element into $t1  
slt $t2,$t1,$v1 #(next)<(max) ?  
bne $t2,$zero,loop#if (next)<(max), repeat  
addi $v0,$t0,0#next element is new max element  
addi $v1,$t1,0#next value is new max value  
j loop#change completed; now repeat  
ret:  
j after\_max

Kết quả :

Ảnh có chứa bàn

Mô tả được tạo tự động

**TH2: Giảm dần**

.data

A: .word 7, -2, 5, 1, 5,6,7,3,6,8,8,59,5

Aend: .word

.text

main:

la $a0,A #$a0 = Address(A[0])

la $a1,Aend

addi $a1,$a1,-4 #$a1 = Address(A[n-1])

j sort #sort

after\_sort:

li $v0, 10#exit

syscall

end\_main:

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

#procedure sort (ascending selection sort using pointer)

#register usage in sort program

#$a0 pointer to the first element in unsorted part

#$a1 pointer to the last element in unsorted part

#$t0 temporary place for value of last element

#$v0 pointer to max element in unsorted part

#$v1 value of max element in unsorted part

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

sort:

beq $a0,$a1,done#single element list is sorted

j min#call the min procedure

after\_min:

lw $t0,0($a1)#load last element into $t0

sw $t0,0($v0)#copy last element to min location

sw $v1,0($a1)#copy max value to last element

addi $a1,$a1,-4 #decrement pointer to last element

j sort#repeat sort for smaller list

done:j after\_sort

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

#Procedure max#function: fax the value and address of min element in the list

#$a0 pointer to first element

#$a1 pointer to last element

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

min:

addi $v0,$a0,0 #init min pointer to first element

lw $v1,0($v0) #init min value to first value

addi $t0,$a0,0 #init next pointer to first

loop:

beq $t0,$a1,ret #if next=last, return

addi $t0,$t0,4 #advance to next element

lw $t1,0($t0) #load next element into $t1

slt $t2,$v1,$t1 #(next)>(min) ?

bne $t2,$zero,loop#if (next)>(min), repeat

addi $v0,$t0,0#next element is new min element

addi $v1,$t1,0#next value is new min value

j loop#change completed; now repeat

ret:

j after\_min

Kết quả

Ảnh có chứa bàn

Mô tả được tạo tự động

**Assignment 3**

**TH1 : Tăng dần**

.data

A: .word 7, -2, 5, 1, 5,6,7,3,6,8,8,59,5

Aend: .word

.text

main:

la $a0,A #$a0 = Address(A[0])

la $a1,Aend

j sort #sort

after\_sort:

li $v0, 10#exit

syscall

end\_main:

sort:addi $a1,$a1,-4 #$a1 = Address(A[n-1])

beq $a0,$a1,after\_sort#single element list is sorted

addi $t0,$a0,0 #init next pointer to first

loop:

beq $t0,$a1,sort

lw $t1,0($t0) # $t1 = a(i)

lw $t2,4($t0) # $t2 = a(i+1)

slt $t3,$t2,$t1 #(i+1)<(i) ?

beq $t3,$zero,next#if (i)<=(i+1), repeat

sw $t1,4($t0) # a(i+1)= $t1

sw $t2,0($t0) # a(i)= $t2

j loop

next:

addi $t0,$t0,4 #advance to next element

j loop

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

Mô tả được tạo tự động

**TH2 :Giảm dần**

.data  
A: .word 7, -2, 5, 1, 5,6,7,3,6,8,8,59,5  
Aend: .word  
.text  
main:  
la $a0,A #$a0 = Address(A[0])  
la $a1,Aend  
j sort #sort  
after\_sort:  
li $v0, 10#exit  
syscall  
end\_main:  
sort:  
addi $a1,$a1,-4 #$a1 = Address(A[n-1])  
beq $a0,$a1,after\_sort#single element list is sorted  
addi $t0,$a0,0 #init next pointer to first  
loop:  
beq $t0,$a1,sort  
lw $t1,0($t0) #$t1 = a(i)  
lw $t2,4($t0) #$t2 = a(i+1)  
slt $t3,$t1,$t2 #(i)<(i+1) ?  
beq $t3,$zero,next#if (i+1)<=(i), repeat  
sw $t1,4($t0) # a(i+1)= $t1  
sw $t2,0($t0) # a(i) = $t2  
j loop  
next:  
addi $t0,$t0,4 #advance to next element  
j loop

Ảnh có chứa bàn

Mô tả được tạo tự động