Thực hành Kiến trúc máy tính

Week 6

Vũ Ngọc Đức – 20225816

Assignment 1

.data

A: .word -2, 6, -1, 3, -2

.text

main:

la $a0,A #load address a0 = A

li $a1,5 #load int a1 = 5

j mspfx #jump to mspfx

nop

continue:

addi $t6, $v0, 0 #save length of max-sum to $t6

addi $t7, $v1, 0 #save value of max-sum to $t7

li $v0, 10 #exit

syscall

nop

end\_of\_main:

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:

Cho mảng A : -2, 6, -1, 3, -2

=> Kết quả = 6 (Đúng với lí thuyết)

Assignment 2

.data

A: .space 100 # khai báo mảng A

Aend: .word

Message1: .asciiz "Do dai cua mang la: "

Message2: .asciiz "Nhap phan tu mang: "

Message3: .asciiz "\n "

ms: .asciiz " "

.text

main:

la $a3, A # gán $a3 là địa chỉ phần tử đầu tiên của mảng

j insert

after\_insert:

la $a0,A #$a0 = Địa chỉ (A[0])

la $a1,Aend

la $t8, ($t0)

mul $t7, $t0, 4

add $a1, $a0, $t7

add $a1, $a1, -4

j sort # Sắp xếp

after\_sort:

li $v0, 10 # exit

syscall

end\_main:

print:

beq $t9, $t8, after\_print

la $a0, A

mul $t6, $t9, 4

add $t7, $a0, $t6

lw $a0, ($t7)

li $v0, 1

syscall

li $v0, 4

la $a0, ms

syscall

addi $t9, $t9, 1

j print

insert:

li $v0, 4 # Syscall in ra chuỗi

la $a0, Message1

syscall

li $v0, 5

syscall

la $t0, ($v0) # Lưu tạm thời độ dài mảng vào $t0

li $t1, 0

loop\_insert:

beq $t1, $t0, after\_insert # quay trở lại main

li $v0, 4 # Syscall in ra chuỗi

la $a0, Message2

syscall

li $v0, 5

syscall

sw $v0, 0($a3)

addi $t1, $t1, 1

add $a3, $a3, 4

j loop\_insert

sort:

beq $a0,$a1,done # Danh sách chỉ có một phần tử đã được sắp xếp

j max # Gọi quy trình tìm phần tử lớn nhất

after\_max:

lw $t0,0($a1) # Tải phần tử cuối cùng vào $t0

sw $t0,0($v0) # Sao chép phần tử cuối cùng vào vị trí lớn nhất

sw $v1,0($a1) # Sao chép giá trị lớn nhất vào phần tử cuối cùng

addi $a1,$a1,-4 # Giảm con trỏ xuống phần tử cuối cùng

sort: # Lặp lại quá trình sắp xếp cho danh sách nhỏ hơn

li $v0, 4 # Syscall in ra chuỗi

la $a0, Message3

syscall

li $t9, 0

j print

after\_print:

j sort

done:

j after\_sort

max:

la $a0, A

addi $v0,$a0,0 # Khởi tạo con trỏ lớn nhất là phần tử đầu tiên

lw $v1,0($v0) # Khởi tạo giá trị lớn nhất là giá trị đầu tiên

addi $t0,$a0,0 # Khởi tạo con trỏ tiếp theo là phần tử đầu tiên

loop:

beq $t0,$a1,ret # Nếu tiếp theo = cuối cùng, trả về

addi $t0,$t0,4 # Tiến tới phần tử tiếp theo

lw $t1,0($t0) # Tải phần tử tiếp theo vào $t1

slt $t2,$t1,$v1 # (Tiếp theo) < (Lớn nhất) ?

bne $t2,$zero,loop # Nếu (tiếp theo) < (lớn nhất), lặp lại

addi $v0,$t0,0 # Phần tử tiếp theo là phần tử lớn nhất mới

addi $v1,$t1,0 # Giá trị tiếp theo là giá trị lớn nhất mới

j loop # Hoàn thành thay đổi; lặp lại

ret:

j after\_max

Kết quả:

Đầu vào : 4 2 3 1 5

=> 1 2 3 4 5 (Đúng với lý thuyết)

Assignment 3

.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

la $a2,Aend #$a2 = Address(A[n])

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

addi $t0,$a0,0 #$t0 = Address(a[0])

j bubble\_sort #sort

after\_sort:

j print\_array

after\_print:

li $v0, 10 #exit

syscall

end\_main:

bubble\_sort:

loop1:

slt $t6, $t0, $a1 #i<n

beq $t6, $zero, return #if i>=n return

addi $t1, $a0, 0 # $t1 = address of a[0]

subi $a2, $a2, 4 # $a2 = address of a[n-1]

loop2:

slt $t6,$t1,$a2# j < n-i

beq $t6, $zero, exit\_loop2# if j>=n-i

addi $t2, $t1, 4 # $t2 = address of a[j+1]

lw $t3, 0($t1) #$t3 = value of a[j]

lw $t4, 0($t2) #$t4 = value of a[j+1]

slt $t5, $t3, $t4

beq $t5, $zero, swap

addi $t1, $t1, 4# $t1 = address of next elemets

j loop2

swap:

sw $t3, 0($t2)

sw $t4, 0($t1)

addi $t1, $t1, 4# $t1 = address of next elemets

j loop2

exit\_loop2:

addi $t0, $t0, 4# $t0 = address of next elemets

j loop1

return:

j after\_sort

print\_array:

addi $t0, $a0, 0

addi $a1, $a1, 4

loop3:

slt $t6, $t0, $a1

beq $t6, $zero, after\_print

li $v0, 1

lw $a0, 0($t0)

syscall

li $v0, 11

li $a0, 32

syscall

addi $t0, $t0, 4

j loop3

Assignment 4

.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])

addi $a2,$a0,0 #$a2 = Address(A[0])

j insertion\_sort #sort

after\_sort:

j print\_array

after\_print:

li $v0, 10 #exit

syscall

end\_main:

insertion\_sort:

loop1:

addi $a2,$a2,4

slt $t2,$a1,$a2

bne $t2,$zero,return

addi $t0,$a2,0

loop2:

sgt $t2,$t0,$a0

beq $t2,$zero,exit\_loop2

addi $t1, $t0, -4

lw $t3,0($t0)#$t3 = A[j]

lw $t4,0($t1)#$t4 = A[j-1]

slt $t2,$t3,$t4#A[j-1]>A[j]

bne $t2,$zero,swap

j exit\_loop2

swap:

sw $t3,0($t1)

sw $t4,0($t0)

addi $t0,$t0,-4

j loop2

exit\_loop2:

j loop1

return:

j after\_sort

print\_array:

addi $t0, $a0, 0

addi $a1, $a1, 4

loop3:

slt $t6, $t0, $a1

beq $t6, $zero, after\_print

li $v0, 1

lw $a0, 0($t0)

syscall

li $v0, 11

li $a0, 32

syscall

addi $t0, $t0, 4

j loop3