**Thực hành kiến trúc máy tính tuần 30**

**Full name: Lê Quang Khải**

**Student ID: 20225638**

Assignment 1

.data

A: .word 1,2,3,-100,-200

messenger1: .asciiz "Max sum prefix is "

messenger2: .asciiz " with length: "

.text

main:   la $a0, A

        li $a1, 5

        j mspfx

        nop

continue: j mspfx\_end

end\_of\_main:

mspfx: addi $s0,$zero,0 #initialize length in $s0 to 0

       addi $v1,$zero,0 #initialize max sum in $v1 to 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 $s0,$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:

      li $v0, 4

      la $a0, messenger1

      syscall

      li $v0, 1

      addi $a0, $v1, 0

      syscall

      li $v0, 4

      la $a0, messenger2

      syscall

      li $v0, 1

      addi $a0, $s0, 0

      syscall

Ta được kết quả:

A screenshot of a computer

Description automatically generated

Assignment 2

.data

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

   Aend: .word

   comma: .asciiz ", "

   newline: .asciiz "\n"

.text

main:

   la $a0, A #$a0 = Address(A[0])

   la $a1, Aend

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

   addi $t3, $a1, 0 #$t3 = address(A[n-1])

   j sort #sort

after\_sort:

   li $v0, 10 #exitsyscall

   syscall

end\_main:

sort:

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

   j max #call the max procedure

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, after\_max #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

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

   li $v0, 4

   la $a0, newline

   syscall

   la $s0, A # $s0 = Address(A[0])

   add $s1, $zero, $t3 # $s1 = A[n-1]

print:

   li $v0, 1

   lw $a0, 0($s0) #load A[i] into $a0

   syscall

   addi $s0, $s0, 4 #advance to next element

   bgt $s0, $s1, endprint

   li $v0, 4

   la $a0, comma

   syscall

   j print

endprint:

   li $v0, 4

   la $a0, newline

   syscall

   la $a0, A #$a0 = Address(A[0])

   j sort #repeat sort for smaller list

done:

   j after\_sort

Kết quả thu được khi chạy:

A screenshot of a computer

Description automatically generated

Assignment 3

.data

   A: .word  5, 3, 1, 6, 7, 2, 4 ,8

   Aend: .word

   comma: .asciiz ", "

   newline: .asciiz "\n"

.text

main:

   la $a0, A

   la $a1, Aend

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

   addi $t3, $a1, 0 # $t3 = Address(A[n-1])

   j sort

after\_sort:

   li $v0, 10 #exitsyscall

   syscall

end\_main:

sort:

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

   j bubble\_sort

bubble\_sort:

   beq $a0, $a1, endbubble\_sort

   lw $s0, 0($a0) # s0 = A[i]

   lw $s1, 4($a0) # s1 = A[i+1]

   slt $t0, $s0, $s1 # A[i] < A[i+1] ?

   bne $t0, $0, skip\_swap

swap:

   sw $s0, 4($a0) # A[i+1] = $s0 = A[i]

   sw $s1, 0($a0) # A[i] = $s1 = A[i+1]

skip\_swap:

   addi $a0, $a0, 4 # trỏ tiếp lên A[i+1]

   j bubble\_sort

endbubble\_sort:

   la $s0, A

print:

   li $v0, 1

   lw $a0, 0($s0)

   syscall

   beq $s0, $t3, endprint # so sánh với A[i] chạy baoh tới A[n-1]

   li $v0, 4

   la $a0, comma

   syscall

   addi $s0, $s0, 4

   j print

endprint:

   li $v0, 4

   la $a0, newline

   syscall

   la $a0, A

   addi $a1, $a1, -4 # giảm con trỏ xuống A[n-2] rồi A[n-3] ....

   j sort

done: j after\_sort

Kết quả thu được

A screenshot of a computer

Description automatically generated

Assignment 4

.data

A: .word 9,8,7,6,5,4,3

Aend: .word

comma: .asciiz ", "

newline: .asciiz "\n"

.text

main:

   la $a0, A # address A into $a0

   move $a2, $a0 # address $a2 = A[0]

   la $a1, Aend

   addi $a1, $a1, -4 #address A[n-1] into $a1

   addi $t3, $a1, 0 #address A[n-1] into $t3

   j sort

after\_sort:

   li $v0, 10 #exitsyscall

   syscall

end\_main:

sort:

   addi $t0, $a0, 0 # address A[j] into $t0

insert\_sort:

   beq $t0, $a2, endinsert\_sort

   lw $s0, 0($t0) # $s0 = A[j]

   lw $s1, -4($t0) # $s1 = A[j-1]

   slt $t1, $s1, $s0 # so sánh A[j-1] < A[j]?

   bne $t1, $0, skip\_insert

insert:

   sw $s0, -4($t0) # A[j-1] = $s0 = A[j]

   sw $s1, 0($t0) # A[j] = $s1 = A[j-1]

skip\_insert:

   addi $t0, $t0, -4 # j--

   j insert\_sort

endinsert\_sort:

   addi $t3, $a0, 0 # $t3 = $a0

   la $s0, A

print:

   li $v0, 1

   lw $a0, 0($s0)

   syscall

   beq $s0, $a1, end\_print

   li $v0, 4

   la $a0, comma

   syscall

   addi $s0, $s0, 4

   j print

end\_print:

   addi $a0, $t3, 0

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

   li $v0, 4

   la $a0, newline

   syscall

   addi $a0, $t3, 0

   addi $a0, $a0, 4

   j sort

done: j after\_sort

Kết quả:

A screenshot of a computer program

Description automatically generated