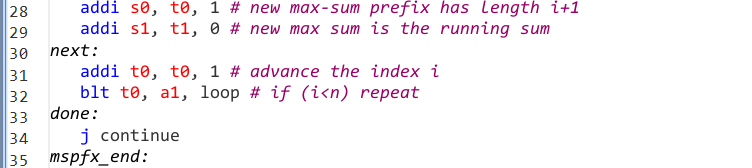
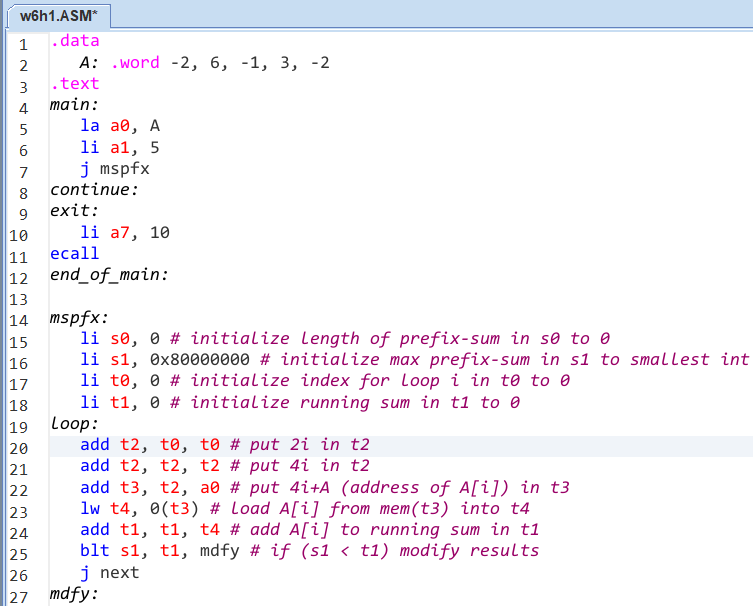
BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 6

Họ và tên: Nguyễn Quý Đức MSSV: 20235682

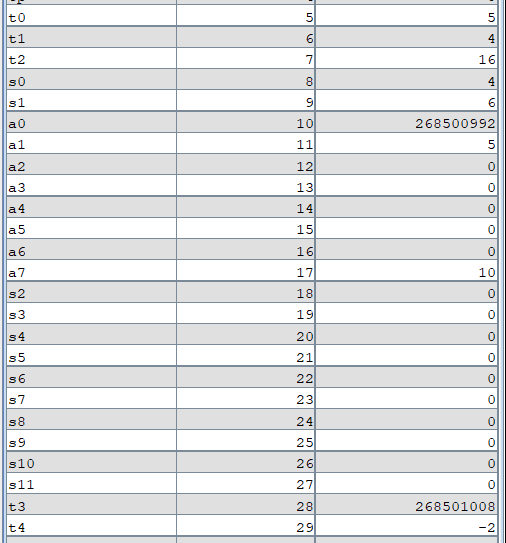
Mã HP: IT3280 Mã lớp: 156788

1. Assignment 1

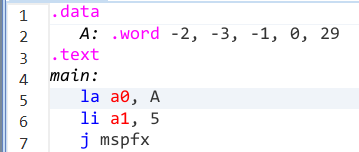


Thanh ghi a0 = 0x10010000, chứa địa chỉ của array A; a1 = 5 là số phần tử của A; t2 = 4i là số byte off-set so với a0 để tính địa chỉ A[i]

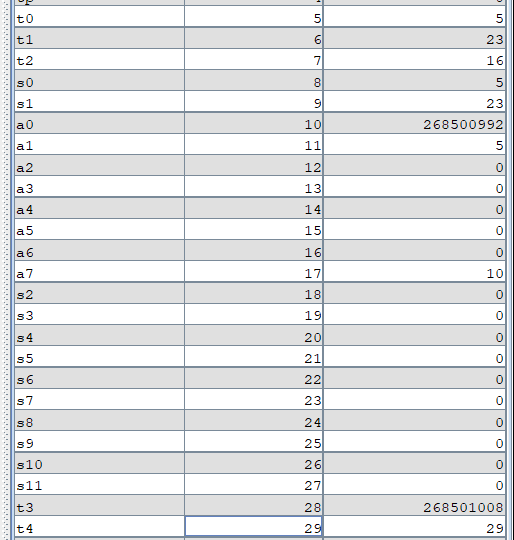
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| t0 (i) | t3 (A[i] addr) | t4 (A[i]) | t1 (sum) | s1 (max sum) | s0 (length) |
| 0 | 0x10010000 | -2 | -2 | -2 | 1 |
| 1 | 0x10010004 | 6 | 4 | 4 | 2 |
| 2 | 0x10010008 | -1 | 3 | 4 | 2 |
| 3 | 0x1001000c | 3 | 6 | 6 | 4 |
| 4 | 0x10010010 | -2 | 4 | 6 | 4 |



Với bộ số với A = [-2, -3, -1, 0, 29]



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| t0 (i) | t3 (A[i] addr) | t4 (A[i]) | t1 (sum) | s1 (max sum) | s0 (length) |
| 0 | 0x10010000 | -2 | -2 | -2 | 1 |
| 1 | 0x10010004 | -3 | -5 | -2 | 1 |
| 2 | 0x10010008 | -1 | -6 | -2 | 1 |
| 3 | 0x1001000c | 0 | -6 | -2 | 1 |
| 4 | 0x10010010 | 29 | 23 | 23 | 5 |



1. Assignment 2

.data

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

Aend: .word

# Aend duoc cap phat dia chi ngay sau A

nl: .asciz "\n"

space: .asciz " "

msg: .asciz "th loop: "

.text

main:

la a2, A

la a1, Aend

addi a1, a1, -4

li a5, 1 #loop counter

j sort

after\_sort:

li a7, 10

ecall

end\_main:

sort:

beq a2, a1, done

j max

after\_max:

lw t0, 0(a1)

sw t0, 0(s0)

sw s1, 0(a1)

addi a1, a1, -4

j print\_array

done:

j after\_sort

max:

addi s0, a2, 0

lw s1, 0(s0)

addi t0, a2, 0

loop:

beq t0, a1, ret

addi t0, t0, 4

lw t1, 0(t0)

blt t1, s1, loop

addi s0, t0, 0

addi s1, t1, 0

j loop

ret:

j after\_max

print\_array:

mv a0, a5

li a7, 1

ecall

la a0, msg # "nth loop: "

li a7, 4

ecall

la a3, A

la a4, Aend

addi a4, a4, -4

print\_loop:

lw a0, 0(a3)

li a7, 1

ecall

la a0, space

li a7, 4

ecall

addi a3, a3, 4

ble a3, a4, print\_loop

print\_last:

la a0, nl

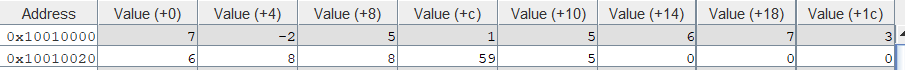
li a7, 4

ecall

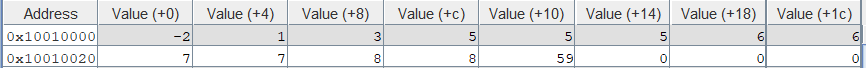
addi a5, a5, 1 # counter++

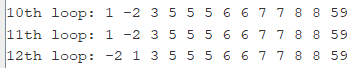
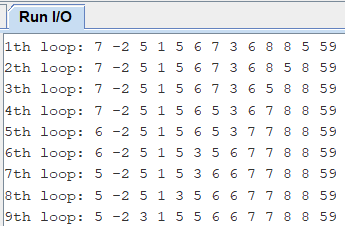
j sort

A trước khi được sort:



A sau khi được sort:





1. Assignment 3

.data

A: .word 7, -2, 5, 1, 129, 192, 12378, 5, 6, 7, 3, 6, 8, 8, 59, 5, 17

Aend: .word

nl: .asciz "\n"

space: .asciz " "

msg: .asciz "th loop: "

.text

main:

la a1, A

la a2, Aend

addi a2, a2, -4

j bubble\_sort

after\_sort:

li a7, 10

ecall

bubble\_sort:

add a3, a1, zero # a3 = &(A[0])

outer:

addi a3, a3, 4 # a3 = &(A[i+1])

add a4, a2, zero # a4 = &(A[n-1])

li t3, 0 # t3 = swapped = false

beq a3, a2, after\_sort # a3 == Aend-4?

inner\_loop:

lw t0, 0(a4) # t0 = A[j]

lw t1, -4(a4) # t1 = A[j-1]

blt t0, t1, swap # A[j] < A[j-1] then swap

no\_swap:

addi a4, a4, -4 # j--

bge a4, a3, inner\_loop # j >= i+1 then continue innerloop

beqz t3, after\_sort # swapped = false

j print\_array

swap:

xor t0, t1, t0

xor t1, t1, t0

xor t0, t1, t0

sw t0, 0(a4)

sw t1, -4(a4)

li t3, 1 # swapped = true

j no\_swap

print\_array:

la s0, A

la s1, Aend

addi s1, s1, -4

mv a0, a5

li a7, 1

ecall

addi a5, a5, 1

la a0, msg # "nth loop: "

li a7, 4

ecall

print\_loop:

lw a0, 0(s0)

li a7, 1

ecall

la a0, space

li a7, 4

ecall

addi s0, s0, 4

ble s0, s1, print\_loop

print\_last:

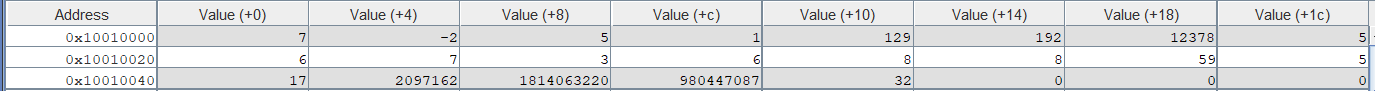
la a0, nl

li a7, 4

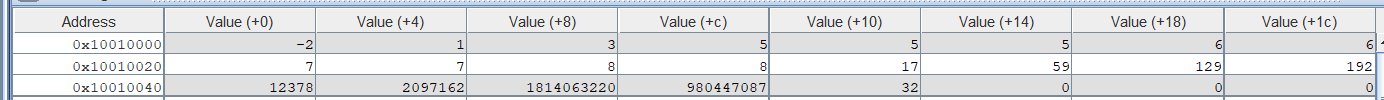
ecall

j outer # continue sorting

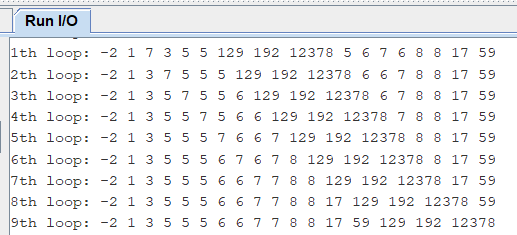
Mảng ban đầu: (từ 0x10010000 đến 0x10010040)



Mảng sau khi sort:



Kết quả in ra:



* Khởi tạo:
  + a1 trỏ vào đầu mảng A.
  + a2 trỏ vào cuối mảng A.
* Vòng lặp ngoài (outer):
  + Duyệt từ A[0] đến A[n-1].
  + Đặt swapped = false (t3 = 0).
  + Vòng lặp trong (inner\_loop):
    - Duyệt từ cuối mảng về đầu, so sánh từng cặp phần tử A[j] và A[j-1].
    - Nếu A[j] < A[j-1], thực hiện hoán đổi bằng toán tử XOR. Và đánh dấu swapped = true.
  + Nếu không có hoán đổi (swapped = false), thoát vòng lặp (after\_sort).
  + In mảng ra sau mỗi lần loop.
  + Lặp lại cho đến khi mảng được sắp xếp.

1. Assignment 4

.data

A: .word 7, -2, 5, 1, 129, 192, 12378, 5, 6, 7, 3, 6, 8, 8, 59, 5, 17

Aend: .word

nl: .asciz "\n"

space: .asciz " "

msg: .asciz "th loop: "

.text

main:

la a1, A

la a2, Aend

addi a2, a2, -4

addi a5, a5, 1 #loop count

j insertion\_sort

after\_sort:

li a7, 10

ecall

insertion\_sort:

add a3, a1, zero # &(A[0])

addi a3, a3, 4 # &(A[1])

outer:

bgt a3, a2, after\_sort # reached Aend

lw t0, 0(a3) # t0 = A[i]

addi a4, a3, -4 # a4 = &(A[i-1]) = &(A[j])

inner\_loop:

blt a4, a1, insert\_key # checked A[i-1] through A[0]

lw t1, 0(a4)

ble t1, t0, insert\_key

sw t1, 4(a4)

addi a4, a4, -4 # j--

j inner\_loop

insert\_key:

sw t0, 4(a4)

j print\_array

print\_array:

la s0, A

la s1, Aend

addi s1, s1, -4

addi a0, a5, 0

li a7, 1

ecall

addi a5, a5, 1

la a0, msg # "nth loop: "

li a7, 4

ecall

print\_loop:

lw a0, 0(s0)

li a7, 1

ecall

la a0, space

li a7, 4

ecall

addi s0, s0, 4

ble s0, s1, print\_loop

print\_last:

la a0, nl

li a7, 4

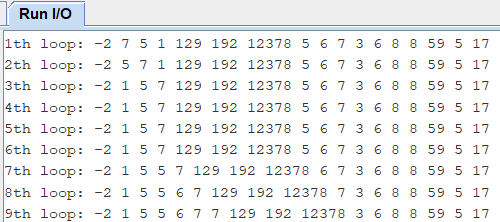
ecall

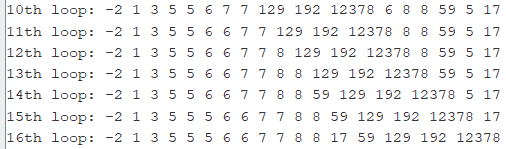
addi a3, a3, 4 # i++

j outer

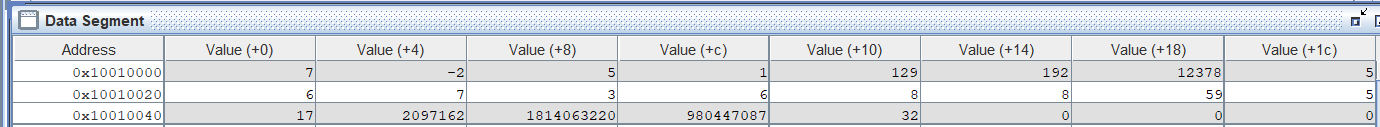
* Khởi tạo:
  + a1 trỏ vào đầu mảng A.
  + a2 trỏ vào cuối mảng A.
  + a5 đếm số vòng lặp.
* Vòng lặp ngoài (outer):
  + Duyệt từ A[1] đến A[n-1].
  + Vòng lặp trong (inner\_loop):
    - Dịch các phần tử lớn hơn A[i] sang phải.
    - Nếu tìm được vị trí thích hợp, chèn A[i] vào (insert\_key).
  + In mảng ra sau mỗi lần loop.
  + Lặp lại cho đến khi mảng được sắp xếp.

Kết quả in ra:





Chuỗi trước khi sort: (từ 0x10010000 đến 0x10010040)



Chuỗi sau khi sort:

