Laboratory Exercise 6 – Report:

Array and Pointer

Lê Văn Duẩn - 20194508

1. Assignment 1

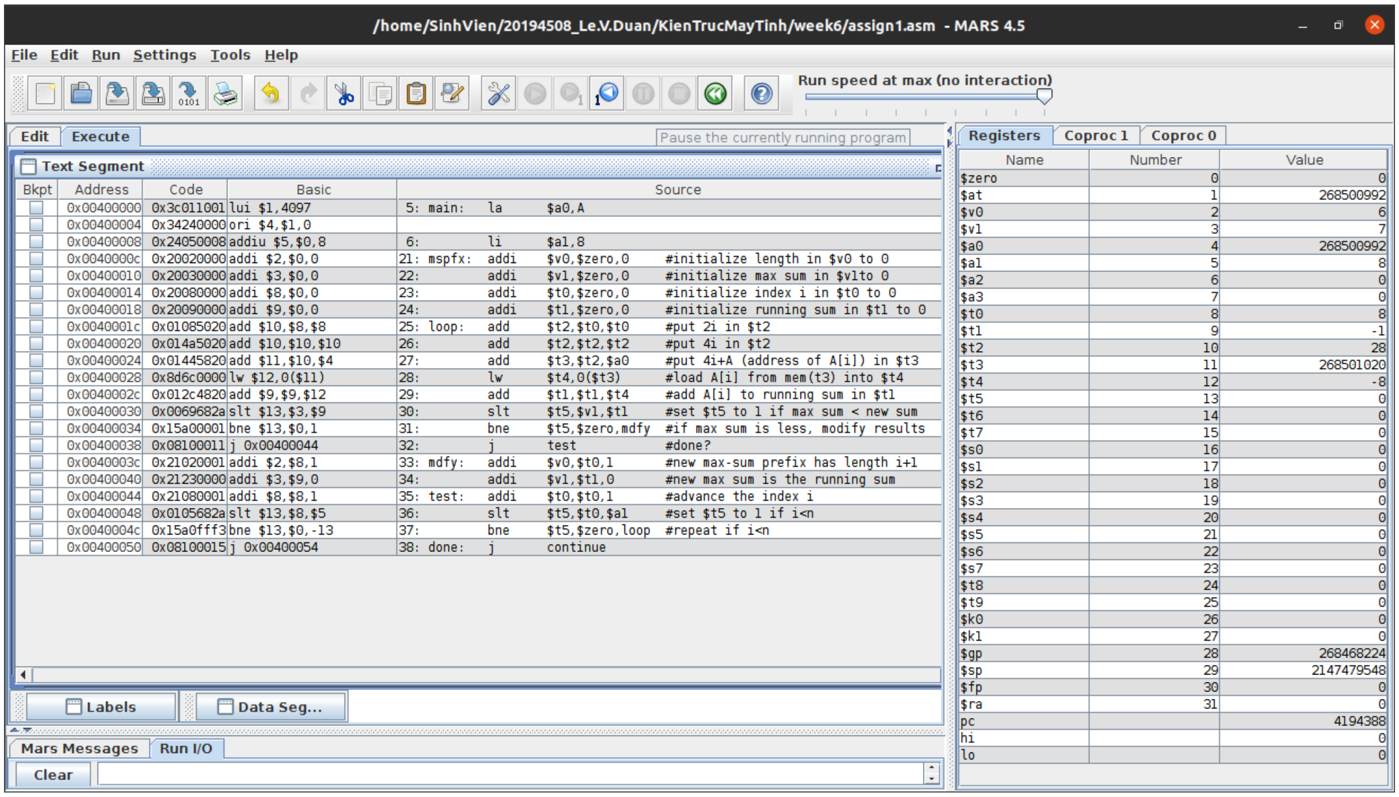
- Mã nguồn:

```
#Laboratory Exercise 6, Home Assignment 1
.data
A:      .word -2, 0, -1, 9, -4, 5, 0, -8      #MSSV 20194508
.text
main: la        $a0,A
li        $a1,8

#-------------------------------------------------------------
#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:

continue:
```

end_of_main:

- Kết quả chạy mô phỏng:


/home/SinhVien/20194508_Le.V.Duan/KienTrucMayTinh/week6/assign1.asm - MARS 4.5

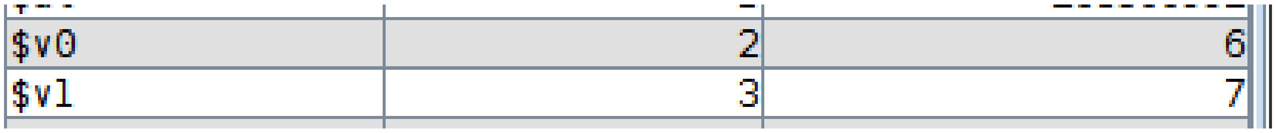
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Pause the currently running program
Text Segment
Labels
Data Seg...
Mars Messages
Run I/O
Clear
Registers Coproc 1 Coproc 0

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui $1,4097	5: main: la $a0,A
	0x00400004	0x34240000	ori $4,$1,0	
	0x00400008	0x24050008	addiu $5,$0,8	6: li $a1,8
	0x0040000c	0x20020000	addi $2,$0,0	21: mspfx: addi $v0,$zero,0 #initialize length in $v0 to 0
	0x00400010	0x20030000	addi $3,$0,0	22: addi $v1,$zero,0 #initialize max sum in $v1to 0
	0x00400014	0x20080000	addi $8,$0,0	23: addi $t0,$zero,0 #initialize index i in $t0 to 0
	0x00400018	0x20090000	addi $9,$0,0	24: addi $t1,$zero,0 #initialize running sum in $t1 to 0
	0x0040001c	0x01085020	add $10,$8,$8	25: loop: add $t2,$t0,$t0 #put 2i in $t2
	0x00400020	0x014a5020	add $10,$10,$10	26: add $t2,$t2,$t2 #put 4i in $t2
	0x00400024	0x01445820	add $11,$10,$4	27: add $t3,$t2,$a0 #put 4i+A (address of A[i]) in $t3
	0x00400028	0x8d6c0000	lw $12,0($11)	28: lw $t4,0($t3) #load A[i] from mem(t3) into $t4
	0x0040002c	0x012c4820	add $9,$9,$12	29: add $t1,$t1,$t4 #add A[i] to running sum in $t1
	0x00400030	0x0069682a	slt $13,$3,$9	30: slt $t5,$v1,$t1 #set $t5 to 1 if max sum < new sum
	0x00400034	0x15a00001	bne $13,$0,1	31: bne $t5,$zero,mdfy #if max sum is less, modify results
	0x00400038	0x08100011	j 0x00400044	32: j test #done?
	0x0040003c	0x21020001	addi $2,$8,1	33: mdfy: addi $v0,$t0,1 #new max-sum prefix has length i+1
	0x00400040	0x21230000	addi $3,$9,0	34: addi $v1,$t1,0 #new max sum is the running sum
	0x00400044	0x21080001	addi $8,$8,1	35: test: addi $t0,$t0,1 #advance the index i
	0x00400048	0x0105682a	slt $13,$8,$5	36: slt $t5,$t0,$a1 #set $t5 to 1 if i<n
	0x0040004c	0x15a0fff3	bne $13,$0,-13	37: bne $t5,$zero,loop #repeat if i<n
	0x00400050	0x08100015	j 0x00400054	38: done: j continue

Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	6
$v1	3	7
$a0	4	268500992
$a1	5	8
$a2	6	0
$a3	7	0
$t0	8	8
$t1	9	-1
$t2	10	28
$t3	11	268501020
$t4	12	-8
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194388
hi		0
lo		0

- Giải thích:

+ Kết quả chạy lưu trong thanh ghi $v0 và $v1 với $v0 là chiều dài và $v1 là tổng lớn nhất tìm được



$v0	2	6
$v1	3	7

+ Như mảng ví dụ: -2, 0, -1, 9, -4, 5, 0, -8 thì kết quả đúng như chạy mô phỏng:

-2 + 0 + -1 + 9 + -4 + 5 = 7 với 6 phần tử.

2. Assignment 2

- Mã nguồn:

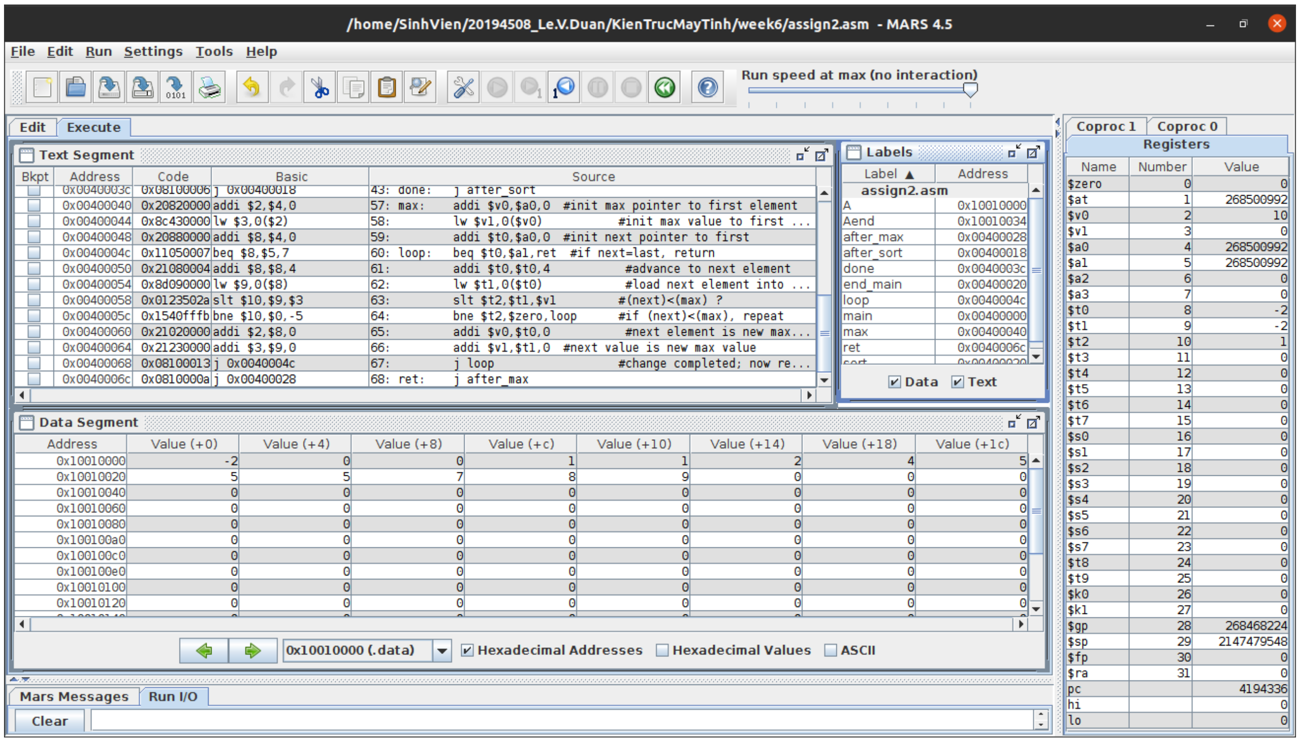
#Laboratory Exercise 6, Home Assignment 2

.data

```
A:      .word 7, -2, 5, 1, 5, 2, 0, 1, 9, 4, 5, 0, 8 #MSSV 20194508
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 chạy:


/home/SinhVien/20194508_Le.V.Duan/KienTrucMayTinh/week6/assign2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Mars Messages Run I/O
Clear

Text Segment

Bkpt	Address	Code	Basic	Source
	0x0040003c	0x08100006	j 0x00400018	43: done: j after_sort
	0x00400040	0x20820000	addi $2,$4,0	57: max: addi $v0,$a0,0 #init max pointer to first element
	0x00400044	0x8c430000	lw $3,0($2)	58: lw $v1,0($v0) #init max value to first ...
	0x00400048	0x20880000	addi $8,$4,0	59: addi $t0,$a0,0 #init next pointer to first
	0x0040004c	0x11050007	beq $8,$5,7	60: loop: beq $t0,$a1,ret #if next=last, return
	0x00400050	0x21080004	addi $8,$8,4	61: addi $t0,$t0,4 #advance to next element
	0x00400054	0x8d090000	lw $9,0($8)	62: lw $t1,0($t0) #load next element into ...
	0x00400058	0x0123502a	slt $10,$9,$3	63: slt $t2,$t1,$v1 #(next)<(max) ?
	0x0040005c	0x1540fffb	bne $10,$0,-5	64: bne $t2,$zero,loop #if (next)<(max), repeat
	0x00400060	0x21020000	addi $2,$8,0	65: addi $v0,$t0,0 #next element is new max...
	0x00400064	0x21230000	addi $3,$9,0	66: addi $v1,$t1,0 #next value is new max value
	0x00400068	0x08100013	j 0x0040004c	67: j loop #change completed; now re...
	0x0040006c	0x0810000a	j 0x00400028	68: ret: j after_max

Labels

Label ▲	Address
assign2.asm	
A	0x10010000
Aend	0x10010034
after_max	0x00400028
after_sort	0x00400018
done	0x0040003c
end_main	0x00400020
loop	0x0040004c
main	0x00400000
max	0x00400040
ret	0x0040006c

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	-2	0	0	1	1	2	4	5
0x10010020	5	5	7	8	9	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0

0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

Coproc 1 Coproc 0 Registers

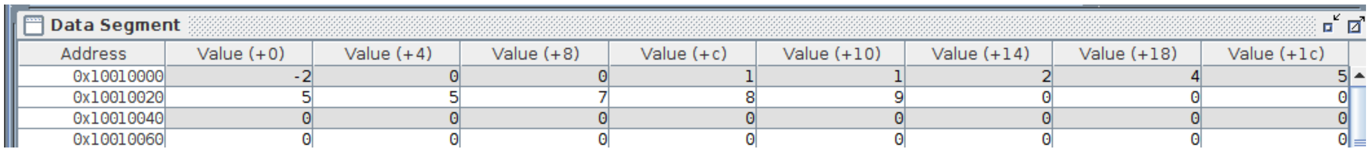
Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	10
$v1	3	0
$a0	4	268500992
$a1	5	268500992
$a2	6	0
$a3	7	0
$t0	8	-2
$t1	9	-2
$t2	10	1
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194336
hi		0
lo		0

- Giải thích:

Chương chình sắp xếp mảng đầu vào và kết quả được hiển thị ở data segment

Với mảng đầu vào: 7, -2, 5, 1, 5, 2, 0, 1, 9, 4, 5, 0, 8

kết quả chạy đúng như mong đợi: -2, 0, 0, 1, 1, 2, 4, 5, 5, 5, 7, 8, 9



Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	-2	0	0	1	1	2	4	5
0x10010020	5	5	7	8	9	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0

3. Assignment 3

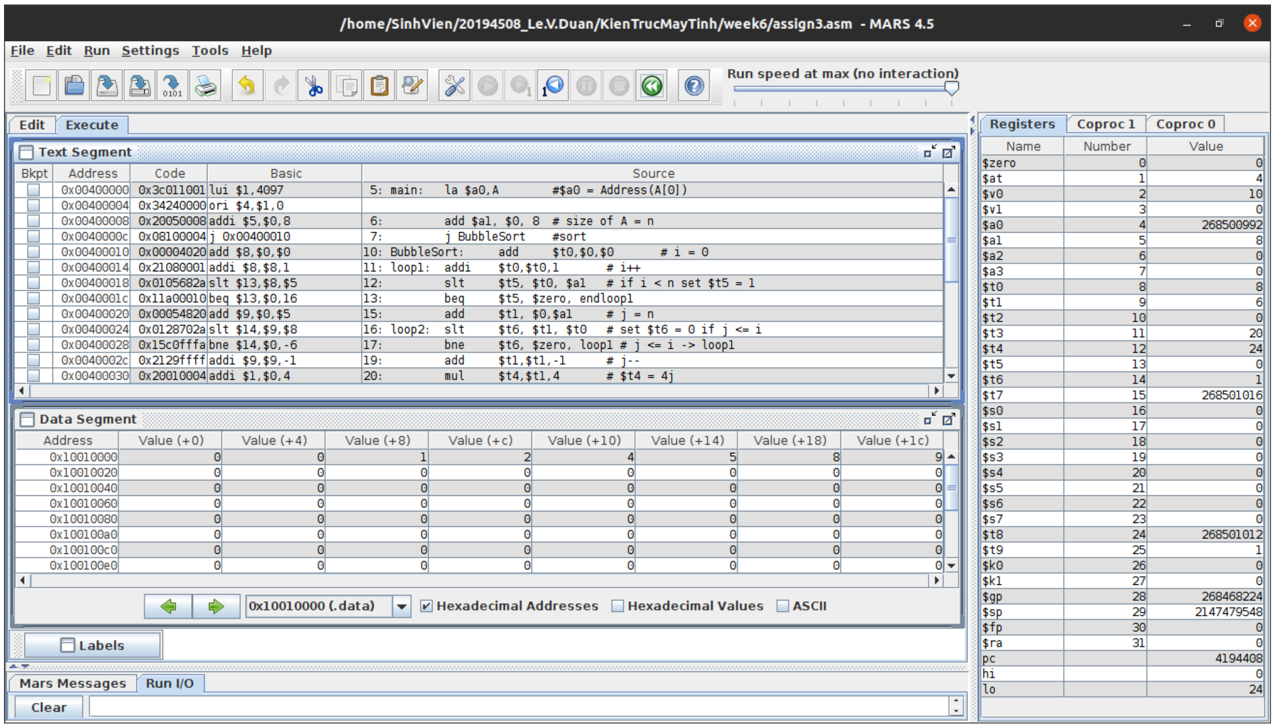
-Mã nguồn:

```
#Laboratory Exercise 6, Home Assignment 3
.data
A:      .word 2, 0 , 1, 9, 4, 5, 0, 8
.text
main: la $a0,A       #$a0 = Address(A[0])
```

```
add $a1, $0, 8          # size of A = n
j BubbleSort            #sort
end_main:

BubbleSort:  add    $t0,$0,$0      # i = 0
loop1:       addi   $t0,$t0,1      # i++
slt     $t5, $t0, $a1# if i < n set $t5 = 1
beq     $t5, $zero, endloop1

add     $t1, $0,$a1   # j = n
loop2:       slt    $t6, $t1, $t0 # set $t6 = 0 if j <= i
bne     $t6, $zero, loop1 # j <= i -> loop1

add     $t1,$t1,-1    # j--
mul     $t4,$t1,4     # $t4 = 4j
addi    $t3,$t4,-4    # $t3 = 4j - 4
add     $t7,$t4,$a0   # $t7 = 4j + $a0 = address A[j]
add     $t8,$t3,$a0   # $t8 = 4j - 4 + $a0 = address A[j-1]
lw      $t5,0($t7)    # load word A[j] to $t5
lw      $t6,0($t8)    # load word A[j-1] to $t6

slt     $t9,$t6,$t5   # if A[j-1] < A[j] set $t9 = 1
bne     $t9,$zero,loop2 # if A[j-1] < A[j] -> countinue loop
```

```
# $t9 = 0 or if A[j-1] > A[j]
# swap A[j-1] and A[j]
sw      $t5,0($t8)
sw      $t6,0($t7)
j       loop2
endloop1:    li $v0, 10 #exit
syscall
```

- Kết quả chạy mô phỏng:


/home/SinhVien/20194508_Le.V.Duan/KienTrucMayTinh/week6/assign3.asm - MARS 4.5

Run speed at max (no interaction)

Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x3c011001	lui $1,4097	5: main: la $a0,A #$a0 = Address(A[0])
☐	0x00400004	0x34240000	ori $4,$1,0	
☐	0x00400008	0x20050008	addi $5,$0,8	6: add $a1, $0, 8 # size of A = n
☐	0x0040000c	0x08100004	j 0x00400010	7: j BubbleSort #sort
☐	0x00400010	0x00004020	add $8,$0,$0	10: BubbleSort: add $t0,$0,$0 # i = 0
☐	0x00400014	0x21080001	addi $8,$8,1	11: loop1: addi $t0,$t0,1 # i++
☐	0x00400018	0x0105682a	slt $13,$8,$5	12: slt $t5, $t0, $a1 # if i < n set $t5 = 1
☐	0x0040001c	0x11a00010	beq $13,$0,16	13: beq $t5, $zero, endloop1
☐	0x00400020	0x00054820	add $9,$0,$5	15: add $t1, $0,$a1 # j = n
☐	0x00400024	0x0128702a	slt $14,$9,$8	16: loop2: slt $t6, $t1, $t0 # set $t6 = 0 if j <= i
☐	0x00400028	0x15c0fffa	bne $14,$0,-6	17: bne $t6, $zero, loop1 # j <= i -> loop1
☐	0x0040002c	0x2129ffff	addi $9,$9,-1	19: add $t1,$t1,-1 # j--
☐	0x00400030	0x20010004	addi $1,$0,4	20: mul $t4,$t1,4 # $t4 = 4j

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0	0	1	2	4	5	8	9
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) [✓ Hexadecimal Addresses] ☐ Hexadecimal Values ☐ ASCII

Registers

Name	Number	Value
$zero	0	0
$at	1	4
$v0	2	10
$v1	3	0
$a0	4	268500992
$a1	5	8
$a2	6	0
$a3	7	0
$t0	8	8
$t1	9	6
$t2	10	0
$t3	11	20
$t4	12	24
$t5	13	0
$t6	14	1
$t7	15	268501016
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	268501012
$t9	25	1
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194408
hi		0
lo		24

- Giải thích:

Sử dụng thuật toán bubble sort với 2 vòng lặp loop1 và loop2 với 2 biến chạy từ 2 đầu của dãy

Swap 2 giá trị bằng cách sw chéo 2 địa chỉ của 2 phần tử cần swap

Với dãy đầu vào: 2, 0, 1, 9, 4, 5, 0, 8

kết quả chạy đúng như mong đợi ở data segment: 0, 0, 1, 2, 4, 5, 8, 9

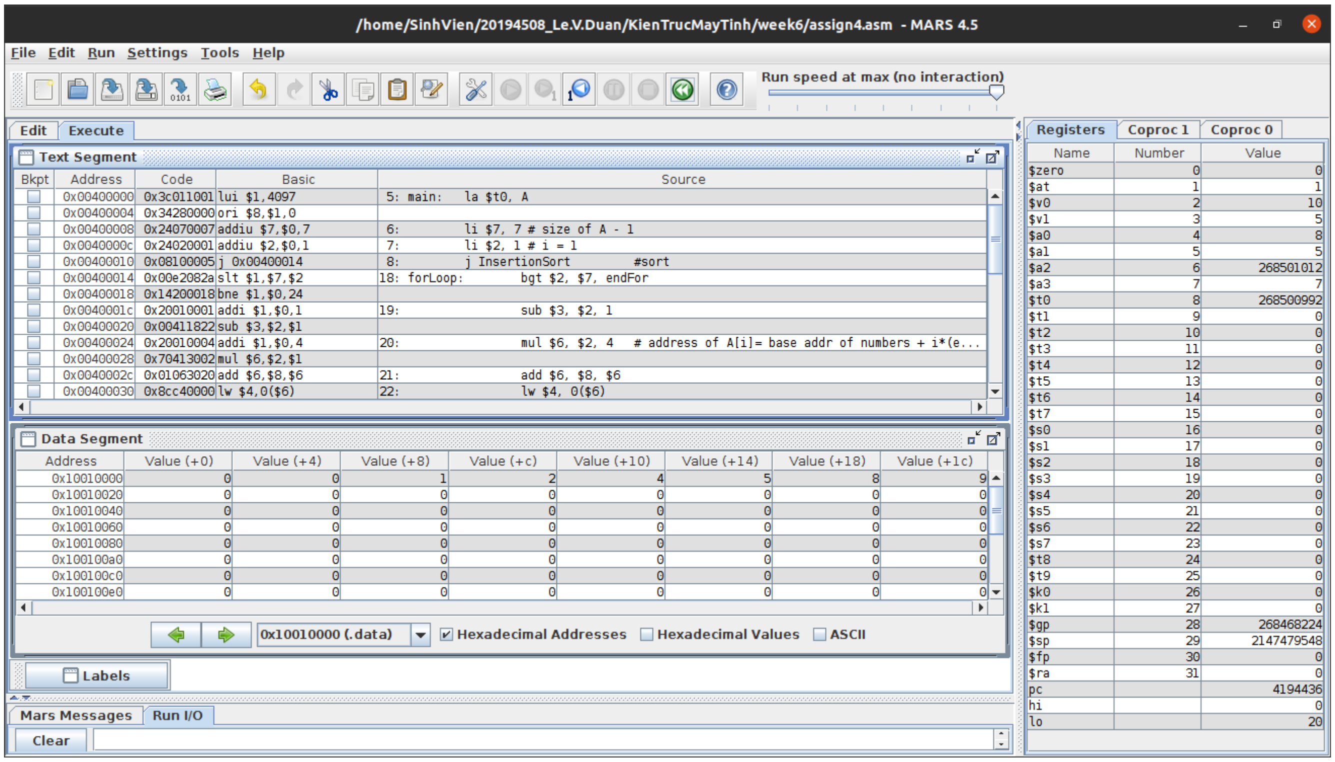
4. Assignment 4

- Mã nguồn:

```
#Laboratory Exercise 6, Home Assignment 4
.data
A:      .word           2, 0, 1, 9, 4, 5, 0, 8
.text
main: la $t0, A
li $7, 7 # size of A - 1
li $2, 1 # i = 1
j InsertionSort        #sort
end_main:
# Use $2 to hold firstUnsortedIndex
# Use $3 to hold testIndex
# Use $4 to hold elementToInsert
# Use $5 to hold value of numbers[ .. ]
# Use $6 to calculate the address of numbers[ ... ] in
# Use $7 to hold the value of (length-1)
# Use $8 to hold the base/starting address of the numbers array
InsertionSort:
forLoop:      bgt $2, $7, endFor
sub $3, $2, 1
mul $6, $2, 4          # address of A[i]= base addr of numbers + i*(element size)
add $6, $8, $6
lw $4, 0($6)
```

```
while:          blt $3, 0, endWhile
mul $6, $3, 4 # address of numbers[i]= base addr of numbers + i*(element size)
add $6, $8, $6
lw $5, 0($6)
ble $5, $4, endWhile
sw $5, 4($6)
sub $3, $3, 1
j while
endWhile:    mul $6, $3, 4 # address of numbers[i]= base addr of numbers + i*(element size)
add $6, $8, $6
sw $4, 4($6)
addi $2, $2, 1
j forLoop
endFor:         li $v0, 10 # system call to exit
syscall
```

- Kết quả chạy:


/home/SinhVien/20194508_Le.V.Duan/KienTrucMayTinh/week6/assign4.asm - MARS 4.5

Run speed at max (no interaction)

Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x3c011001	lui $1,4097	5: main: la $t0, A
☐	0x00400004	0x34280000	ori $8,$1,0	
☐	0x00400008	0x24070007	addiu $7,$0,7	6: li $7, 7 # size of A - 1
☐	0x0040000c	0x24020001	addiu $2,$0,1	7: li $2, 1 # i = 1
☐	0x00400010	0x08100005	j 0x00400014	8: j InsertionSort #sort
☐	0x00400014	0x00e2082a	slt $1,$7,$2	18: forLoop: bgt $2, $7, endFor
☐	0x00400018	0x14200018	bne $1,$0,24	
☐	0x0040001c	0x20010001	addi $1,$0,1	19: sub $3, $2, 1
☐	0x00400020	0x00411822	sub $3,$2,$1	
☐	0x00400024	0x20010004	addi $1,$0,4	20: mul $6, $2, 4 # address of A[i]= base addr of numbers + i*(e...
☐	0x00400028	0x70413002	mul $6,$2,$1	
☐	0x0040002c	0x01063020	add $6,$8,$6	21: add $6, $8, $6
☐	0x00400030	0x8cc40000	lw $4,0($6)	22: lw $4, 0($6)

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0	0	1	2	4	5	8	9
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) [✓ Hexadecimal Addresses] ☐ Hexadecimal Values ☐ ASCII

Name	Number	Value
$zero	0	0
$at	1	1
$v0	2	10
$v1	3	5
$a0	4	8
$a1	5	5
$a2	6	268501012
$a3	7	7
$t0	8	268500992
$t1	9	0
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194436
hi		0
lo		20

- Giải thích:

Sử dụng thuật toán insertion sort để sắp xếp mảng cho sẵn

Sử dụng 1 vòng lặp forLoop và 1 vòng lặp while

Với dãy đầu vào: 2, 0, 1, 9, 4, 5, 0, 8

kết quả chạy đúng như mong đợi ở data segment: 0, 0, 1, 2, 4, 5, 8, 9