

# Laboratory Exercise 6

## Assignment 1

.data

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

.text

main:

la \$a0,A

li \$a1,5

j mspfx

nop

continue:

lock:

li \$v0, 10#exit

syscall

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 reaches.

# @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

mshpf:

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

mshpf\_end:

**Kết quả:**

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	6
\$a0	4	268500992
\$a1	5	5
\$a2	6	0
\$a3	7	0
\$t0	8	5
\$t1	9	4
\$t2	10	16
\$t3	11	268501008
\$t4	12	-2
\$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		4194332
hi		0
lo		0

## Assignment 2

### Sắp xếp theo 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: find the value and address of max element in the list

#$a0 pointer to first element
#$a1 pointer to last element
#-----

max:

```

Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	
0x10010000	-2	1	3	5	5	5	5	5	▲
0x10010020	7	7	8	8	59	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	▼
0x10010140	0	0	0	0	0	0	0	0	▲
0x10010160	0	0	0	0	0	0	0	0	▼
0x10010180	0	0	0	0	0	0	0	0	▲
0x100101a0	0	0	0	0	0	0	0	0	▼

◀
▶
0x10010000 (.data)
☒ Hexadecimal Addresses
 ☐ Hexadecimal Values
 ☐ ASCII

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	1
\$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

## Sắp xếp theo 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: find 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





Registers	Coproc 1	Coproc 0	
Name	Number	Value	
\$zero	0	0	
\$at	1	268500992	
\$v0	2	10	
\$v1	3	8	
\$a0	4	268500992	
\$a1	5	268500992	
\$a2	6	0	
\$a3	7	0	
\$t0	8	8	
\$t1	9	8	
\$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		4194336	
hi		0	
lo		0	

## Assignment 3

### 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

Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	
0x10010000	-2	1	3	5	5	5	6		
0x10010020	7	7	8	8	59	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	
0x10010140	0	0	0	0	0	0	0	0	
0x10010160	0	0	0	0	0	0	0	0	
0x10010180	0	0	0	0	0	0	0	0	
0x100101a0	0	0	0	0	0	0	0	0	

◀
▶
0x10010000 (data)
Hexadecimal Addresses
Hexadecimal Values
ASCII

Registers	Coproc 1	Coproc 0
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	268500996
\$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		4194332
hi		0
lo		0

## 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

```

## Kết quả:

Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	
0x10010000	59	8	8	7	7	6	6	5	▲
0x10010020	5	5	3	1	-2	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	
0x10010140	0	0	0	0	0	0	0	0	
0x10010160	0	0	0	0	0	0	0	0	
0x10010180	0	0	0	0	0	0	0	0	
0x100101a0	0	0	0	0	0	0	0	0	

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

Registers	Coproc 1	Coproc 0
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	268500996
\$t1	9	59
\$t2	10	8
\$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		4194332
hi		0
lo		0