

Laboratory Exercise 6

Array and Pointer

Đỗ Hải Dương – 20194528

Assignment 1

```
1  .data
2  A: .word -2, 6, -1, 3, 20194528, -20194534
3  .text
4  main:  la $a0,A
5         li $a1,6
6         j  mspfx
7         nop
8
9  continue:
10 lock:  j  lock
11        nop
12 end_of_main:
13
14 mspfx:  addi $v0,$zero,0 #initialize length in $v0 to 0
15        addi $v1,$zero,0 #initialize max sum in $v1 to 0
16        addi $t0,$zero,0 #initialize index i in $t0 to 0
17        addi $t1,$zero,0 #initialize running sum in $t1 to 0
18
19 loop:  add $t2,$t0,$t0 #put 2i in $t2
20        add $t2,$t2,$t2 #put 4i in $t2
21        add $t3,$t2,$a0 #put 4i+A (address of A[i]) in $t3
22        lw $t4,0($t3) #load A[i] from mem(t3) into $t4
23        add $t1,$t1,$t4 #add A[i] to running sum in $t1
24        slt $t5,$v1,$t1 #set $t5 to 1 if max sum < new sum
25        bne $t5,$zero,mdfy #if max sum is less, modify results
26        j test #done?
27
28 mdfy:  addi $v0,$t0,1 #new max-sum prefix has length i+1
29        addi $v1,$t1,0 #new max sum is the running sum
30
31 test:  addi $t0,$t0,1 #advance the index i
32        slt $t5,$t0,$a1 #set $t5 to 1 if i<n
33        bne $t5,$zero,loop #repeat if i<n
34
35 done:  j continue
36 mspfx_end:
```

- Kết quả chạy (max prefix sum lưu ở v1).

Max prefix sum = -2 + 6 -1 + 3 + 20194528 = 20194534

Registers		
Coproc 1		Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	5
\$v1	3	20194534
\$a0	4	268500992
\$a1	5	6
\$a2	6	0
\$a3	7	0
\$t0	8	6
\$t1	9	0
\$t2	10	20
\$t3	11	268501012
\$t4	12	-20194534
\$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		4194324
hi		0
lo		0

Giải thích:

- Khai báo các giá trị của mảng
- Gán địa chỉ của A vào thanh ghi a0
- Gán $a1 = 6$ (vì mảng có 6 phần tử)
- Khởi tạo các giá trị $v0 = 0$, max prefix sum $v1 = 0$, $t0 = i = 0$, tổng chạy $t1 = 0$
- Vòng lặp (nhãn loop) gán $t2 = 2i$
- Tiếp theo gán $t2 = 4i$
- Gán địa chỉ $A[i]$ vào thanh ghi t3 → load giá trị của $A[i]$ vào t4 → thêm $A[i]$ vào tổng chạy t1
- So sánh giá trị v1 (tổng cần tìm) với t1 (tổng cần chạy)
 - +) Nếu $v1 < t1$ thì $t5 = 1$
 - +) Nếu $t5 \neq 0$ thì nhảy đến nhãn mdify
- Trong nhãn mdify: dãy max prefix sum mới có độ dài $i + 1$
- Gán $v1 = t1$
- So sánh: Nếu $t0$ (biến chạy) $< a1$ (số phần tử của mảng) thì $t5 = 1$
- Nếu $t5 \neq 0$ (tức $i < n$) thì quay lại vòng lặp loop

Assignment 2

```
1 .data
2 A: .word 7, -2, 28, 1, 20194528, 5
3 Aend: .word
4
5 .text
6 main:      la $a0,A          #$a0 = Address (A[0])
7            la $a1,Aend
8            addi $a1,$a1,-4    #$a1 = Address (A[n-1])
9            j sort            #sort
10
11 after_sort: li $v0, 10       #exit
12            syscall
13
14 end_main:
15
16 sort:      beq $a0,$a1,done   #single element list is sorted
17            j max              #call the max procedure
18 after_max: lw $t0,0($a1)      #load last element into $t0
19            sw $t0,0($v0)      #copy last element to max location
20            sw $v1,0($a1)      #copy max value to last element
21
22            addi $a1,$a1,-4     #decrement pointer to last element
23            j sort            #repeat sort for smaller list
24 done:      j after_sort
25
26 max:
27            addi $v0,$a0,0      #init max pointer to first element
28            lw $v1,0($v0)       #init max value to first value
29            addi $t0,$a0,0      #init next pointer to first
30 loop:
31            beq $t0,$a1,ret     #if next=last, return
32            addi $t0,$t0,4      #advance to next element
33            lw $t1,0($t0)       #load next element into $t1
34            slt $t2,$t1,$v1     #(next)<(max) ?
35            bne $t2,$zero,loop  #if (next)<(max), repeat
36            addi $v0,$t0,0      #next element is new max element
37            addi $v1,$t1,0      #next value is new max value
38            j loop              #change completed; now repeat
39 ret:
40            j after_max
```

Kết quả:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)
0x10010000	-2	1	5	7	28	20194528

Giải thích:

- Khai báo các giá trị của mảng
- Khai báo Aend
- Gán địa chỉ của A vào thanh ghi a0, gán địa chỉ của Aend vào thanh ghi a1
- Giảm a1 đi 4 → địa chỉ của A[n-1]
- Nhãn after_sort: gán v0 = 10 (10 là exit)
- Nhãn sort: so sánh nếu a0 = a1 thì nhảy đến nhãn done
- Nhãn after_max: lưu giá trị của phần tử cuối vào t0

- Copy giá trị phần tử cuối cùng vào vùng lớn nhất → copy giá trị lớn nhất vào phần tử cuối
- Giảm địa chỉ đi 4 → nhảy quay lại sort cho mảng bé hơn (trừ phần tử đã đưa về cuối)
- Nhãn done: nhảy đến after_sort để thoát
- Gán địa chỉ v0 = a0 (v0 là con trỏ lớn nhất) → load địa chỉ của v0 vào v1 → gán t0 về địa chỉ đầu tiên của mảng
- So sánh nếu phần tử tiếp theo là cuối cùng thì nhảy đến ret
- Tăng t0 lên 4 → địa chỉ của t0 (phần tử tiếp theo) vào t1
- So sánh nếu $t1 < v1$ thì $t2 = 1$. Nếu $t2 \neq 0$ thì quay lại vòng lặp loop
- Gán phần tử tiếp theo thành phần tử lớn nhất mới → gán giá trị t1 vào v1 (giá trị phần tử kế tiếp là giá trị max)

Assignment 3

```
1  .data
2  arr: .word 4528, 2019, 0, -1, -2, 100, -28, -9, 10
3  space: .asciiz " "
4  .text
5      la      $s0, arr
6      li      $t0, 0          # i = 0
7      li      $s1, 8          # n = 8
8      li      $s2, 9
9      add     $t2, $zero, $s0
10
11  outer_loop:
12      li      $t1, 0          # j = 0
13      addi    $s2, $s2, -1
14      add     $t3, $zero, $s0
15
16  inner_loop:
17      lw      $s3, 0($t3)
18      addi    $t3, $t3, 4
19      lw      $s4, 0($t3)
20      addi    $t1, $t1, 1
21      slt     $t4, $s3, $s4
22      bne     $t4, $zero, cond
23
24  swap:      sw      $s3, 0($t3)
25             sw      $s4, -4($t3)
26             lw      $s4, 0($t3)
27
28  cond:      bne     $t1, $s2, inner_loop    # j != n - 1
29             addi    $t0, $t0, 1            # i++
30             bne     $t0, $s1, outer_loop    # i != n
31             li      $t0, 0
32
33
34  print:     li      $v0, 1
35             lw      $a0, 0($t2)
36             syscall
37             li      $v0, 4
38             la      $a0, space
39             syscall
40             addi    $t2, $t2, 4
41             addi    $t0, $t0, 1
42             bne     $t0, $s1, print         # i != n
43
```

Kết quả:

```
-28 -9 -2 -1 0 10 100 2019 4528
-- program is finished running (dropped off bottom) --
```

Giải thích:

- Khai báo mảng, khai báo chuỗi space(dấu cách để in mảng)
- Gán địa chỉ của mảng arr vào thanh ghi s0, gán biến chạy i = 0, gán s1 = 8 (phần tử mảng từ 0 đến 8), s2 = 0 (số phần tử mảng) để giảm dần, t2 = địa chỉ của mảng
- Nhãn outer_loop: gán t1 = 0 (biến j), giảm s2 đi 1, gán t3 = địa chỉ của arr[0]
- Nhãn inner_loop: load giá trị của arr[j] vào s3, tăng t3 lên 4, load giá trị arr[j+1] vào s4, tăng t1 lên 1(biến j)
- So sánh nếu s3 < s4 thì t4 = 1
Nếu t4 != 0 thì nhảy đến cond
- Nhãn swap: ghi địa chỉ của s3 vào t3, ghi địa chỉ của t3 - 4 vào s4 → ghi giá trị của t3 vào s4
- Nhãn cond: so sánh nếu t1 != s2 thì nhảy đến inner_loop → tăng t0(biến i) lên 1
- So sánh nếu t0 != s1 thì nhảy đến outer_loop
- Nhãn print: in mảng
- Nếu t0 != s1 (chưa đến cuối mảng) thì nhảy lại đến print để in phần tử còn lại.