

Báo cáo thực hành Lab5

Môn: IT012

GVHD: Phan Đình Duy

Họ và tên: Đỗ Phương Duy

MSSV: 23520362

1. Thao tác mảng:

In ra cửa sổ I/O của MARS tất cả các phần tử của mảng array1 và array2

The screenshot shows the MARS 4.5 assembly debugger interface. The assembly code window displays the following code:

```
5 size2: .word 16
6 array3: .space 8
7 size3: .word 8
8 endline: .asciiz "\n"
9 space: .asciiz " "
10 .text
11
12 #mang1
13 la $s0, array1
14 lw $t0, size1
15 Lloop1: bne $t1, $t0, end1
16 add $t2, $t1, 2
17 add $t3, $t2, $t2
18 lw $t1, ($t3)
19 li $v0, 1
20 add $s0, $t0, $s1
21 syscall
22 addi $t1, $t1, 1
23 add $v0, $s0, 4
24 la $s0, space
25 syscall
26 j Loop1
27 end1:
28 la $s0, endline
29 syscall
```

The Registers window shows the following register values:

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7ffffefc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400004
hi		0x00000000
lo		0x00000000

The Mars Messages window shows the output: "5 6 7 8 1 2 3 9 10 4" and "— program is finished running (dropped off bottom) —".

The screenshot shows the MARS 4.5 assembly editor interface. The assembly code in the main window is:

```
26 j Loop1
27 end1:
28 la $a0, endline
29 syscall
30 #mang2
31 la $s0, array2
32 lw $t0 size2
33 addi $t1, $0, 0
34 Loop2: beq $t1, $t0, end2
35 add $t3, $s0, $t1
36 lb $s1, ($t3)
37 li $v0, 1
38 add $a0, $0, $s1
39 syscall
40 addi $t1, $t1, 1
41 addi $v0, $0, 4
42 la $a0, space
43 syscall
44 j Loop2
45 end2:
46 la $a0, endline
47 syscall
48
49
```

The status bar at the bottom left indicates "Line: 37 Column: 15" and has a checked "Show Line Numbers" option.

In the "Mars Messages" window, the output is:

```
5 6 7 8 1 2 3 9 10 4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
-- program is finished running (dropped off bottom) --
```

Gán các giá trị cho mảng array3 sao cho $\text{array3}[i] = \text{array2}[i] + \text{array2}[\text{size2} - 1 - i]$

```

.text
lw $t0 size3
lw $s0 size2
la $s1 array2
la $s2 array3
Loop:
beq $t1, $t0, end
add $t2, $t1, $s1
lb $t3 ($t2)

sub $t2, $s0, $t1
subi $t2, $t2, 1
add $t2, $t2, $s1
lb $t4 ($t2)

add $t5, $t3, $t4
add $t6, $t1, $s2
sb $t5, ($t6)
addi $t1, $t1, 1
j Loop
end:

```

Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	
0x10010000	0x00000005	0x00000006	0x00000007	0x00000008	0x00000001	0x00000002	0x00000003	0x00000009	
0x10010020	0x0000000a	0x00000004	0x0000000a	0x04030201	0x08070605	0x0c0b0a09	0x100f0e0d	0x00000010	
0x10010040	0x11111111	0x11111111	0x00000008	0x0000000a	0x00000000	0x00000000	0x00000000	0x00000000	
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	

← → 0x10010000 (.data) ↻ Hexadecimal Addresses Hexadecimal Values ASCII

Người sử dụng nhập vào mảng thứ máy và chỉ số phần tử cần lấy trong mảng đó, chương trình xuất ra phần tử tương ứng.

```

1 .data
2 array1: .word 5, 6, 7, 8, 1, 2, 3, 9, 10, 4
3 size1: .word 10
4 array2: .byte 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
5 size2: .word 16
6 array3: .space 8
7 size3: .word 8
8 endlne: .asciiz "\n"
9 space: .asciiz "
10
11 .text
12 li $v0, 5
13 syscall
14 move $t0, $v0
15 li $v0, 5
16 syscall
17 move $t1, $v0
18
19 bne $t0, 1, secondarray
20 la $s0, array1
21 sll $t1, $t1, 2
22 add $t2, $s0, $t1
23 lw $s0, ($t2)
24 li $v0, 1
25 syscall
26 j end
27
28 secondarray:
29 bne $t0, 2, thirdarray
30 la $s0, array2
31 sll $t1, $t1, 2
32 add $t2, $s0, $t1
33 lw $s0, ($t2)
34 li $v0, 1
35 syscall
36
37 thirdarray:
38 la $s0, array3
39 add $t2, $s0, $t1
40 lw $s0, ($t2)
41 li $v0, 1
42 syscall
43 end:

```

Line: 36 Column: 10 Show Line Numbers

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Name	Number	Value
\$8 (vaddr)	8	0x00000000
\$12 (status)	12	0x0000f11
\$13 (cause)	13	0x00000000
\$14 (epc)	14	0x00000000

Mars Messages Run I/O

```

1
5
2
-- program is finished running (dropped off bottom) --

```

2. Thao tác con trỏ:

Con trỏ là một biến lưu địa chỉ của một biến khác. Thao tác với con trỏ trong MIPS là thao tác trực tiếp với địa chỉ bộ nhớ.

Thực hiện lại các yêu cầu của nội dung 1 với con trỏ.

In ra cửa sổ I/O của MARS tất cả các phần tử của mảng array1 và array2

ASM Editor (Top Window):

```

5 size2: .word 16
6 array3: .space 8
7 size3: .word 8
8 endlne: .asciiz "\n"
9 space: .asciiz " "
10 .text
11
12 #mang1
13 la $s0 array1
14 lw $t0 size1
15 Loop1: beq $t1, $t0, endl1
16 lw $s1, ($s0)
17 li $v0, 1
18 add $s0, $s0, $s1
19 syscall
20 addi $s0, $s0, 4
21 addi $t1, $t1, 1
22 addi $v0, $s0, 4
23 la $a0, space
24 syscall
25 j Loop1
26 endl1:
27 la $a0, endlne
28 syscall
29
30 #mang2

```

Line: 39 Column: 17 ✓ Show Line Numbers

MARS Simulator (Bottom Window):

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000004
\$v1	3	0x00000000
\$a0	4	0x1001004c
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000010
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x1001003c
\$s1	17	0x00000010
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0xffffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x0000009c
hi		0x00000000
lo		0x00000000

Mars Messages Run I/O

5 6 7 8 1 2 3 9 10 4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Clear -- program is finished running (dropped off bottom) --

Gán các giá trị cho mảng array3 sao cho array3[i] = array2[i] + array2[size2 - 1 - i]

Edit Execute

Bai1.asm Bai1b.asm Bailc.asm bai2a.asm bai2b

```

7 size3: .word 8
8 endline: .asciiz "\n"
9 space: .asciiz "
10
11 .text
12 lw $t0 size3
13 lw $s0 size2
14 la $s1 array2
15 la $s2 array3
16 Loop:
17 beq $t1, $t0, end
18 lb $t3, ($s1)
19 addi $s1, $s1, 1
20
21 sub $t2, $s0, $t1
22 subi $t2, $t2, 1
23 add $t2, $t2, $s1
24 lb $t4, ($t2)
25
26 add $t5, $t3, $t4
27 sb $t5, ($s2)
28 addi $s2, $s2, 1
29 addi $t1, $t1, 1
30 j Loop
31 end:

```

Line: 27 Column: 15 Show Line Numbers

Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	
0x10010000	0x00000005	0x00000006	0x00000007	0x00000008	0x00000001	0x00000002	0x00000003	0x00000009	
0x10010020	0x0000000a	0x00000004	0x0000000a	0x04030201	0x08070605	0x0c0b0a09	0x100f0e0d	0x00000010	
0x10010040	0x11111111	0x11111111	0x00000008	0x0000000a	0x00000000	0x00000000	0x00000000	0x00000000	
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	

← → 0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Người sử dụng nhập vào mảng thứ mây và chỉ số phần tử cần lấy trong mảng đó, chương trình xuất ra phần tử tương ứng.

```

1 .data
2 array1: .word 5, 6, 7, 8, 1, 2, 3, 9, 10, 4
3 size1: .word 10
4 array2: .byte 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
5 size2: .word 16
6 array3: .space 8
7 size3: .word 8
8 endlne: .asciiz "\n"
9 space: .asciiz " "
10
11 .text
12 li $v0, 5
13 syscall
14 move $t0, $v0
15 li $v0, 5
16 syscall
17 move $t1, $v0
18
19 bne $t0, 1, secondarray
20 la $s0, array1
21 sll $t1, $t1, 2
22 add $t2, $s0, $t1
23 lw $a0, ($t2)
24 li $v0, 1
25 syscall
26 j end
27
28 secondarray:
29 beq $t0, 2, thirddarray
30 la $s0, array2
31 add $t2, $s0, $t1
32 lb $a0, ($t2)
33 li $v0, 1
34 syscall
35
36 thirddarray:
37 la $s0, array3
38 add $t2, $s0, $t1
39 lb $a0, ($t2)
40 li $v0, 1
41 syscall
42
43 end:

```

Line: 16 Column: 9 Show Line Numbers

3. Bài tập: (chỉ sử dụng con trỏ):

a. Nhập một mảng các số nguyên n phần tử (nhập vào số phần tử và giá trị của từng phần tử), xuất ra cửa sổ I/O của MARS theo từng yêu cầu sau:

✓ Xuất ra giá trị lớn nhất và nhỏ nhất của mảng

✓ Tổng tất cả các phần tử của mảng:

```

1 .data
2 phantu: .asciiz "Nhập số phần tử: "
3 tong: .asciiz "\nTổng: "
4 arr: .word 0:10
5 space: .asciiz " "
6 endlne: .asciiz "\n"
7 .text
8 li $v0, 4
9 la $a0, phantu
10 syscall
11 li $v0, 5
12 syscall
13 move $s0, $v0
14 li $t0, 0
15 la $a1, arr
16 loop1:
17 beq $t0, $s0, end_loop1
18 li $v0, 5
19 syscall
20 sw $v0, ($a1)
21 addi $t0, $t0, 1
22 addi $a1, $a1, 4
23 j loop1
24

```

Line: 14 Column: 10 Show Line Numbers


```

19 syscall
20 sw $v0, ($a1)
21 addi $t0, $t0, 1
22 addi $a1, $a1, 4
23 j loop1
24
25 end_loop1:
26 li $t0, 0
27 la $a1, arr
28 li $t3, 0
29 loop2:
30 beq $t0, $s0, end_loop2
31 lw $t2, ($a1)
32 add $t3, $t3, $t2
33 addi $t0, $t0, 1
34 addi $a1, $a1, 4
35 j loop2
36 end_loop2:
37 li $v0, 4
38 la $a0, tong
39 syscall
40 move $a0, $t3
41 li $v0, 1
42 syscall

```

Line: 14 Column: 10 Show Line Numbers

Mars Messages	Run I/O
Nhập số phần tử: 5	
1	
2	
3	
4	
5	
Tổng: 15	

✓ Người sử dụng nhập vào chỉ số của một phần tử nào đó và giá trị của phần tử đó được in ra cửa sổ.

b. Nhập một mảng các số nguyên n phần tử (nhập vào số phần tử và giá trị của từng phần tử). Mảng này gọi là A.

Chuyển dòng lệnh C dưới đây sang mã assembly của MIPS. Với các biến nguyên i, j được gán lần lượt vào thanh ghi \$s0, \$s1; và địa chỉ nền của mảng số nguyên A được lưu trong thanh ghi \$s3

```
if (i<j) A[i]= i;  
else A[i] = j;  
.text  
  
    slt $t0, $s0, $s1  
    beq $t0, 0, else  
  
        sll $t1, $s0, 2|  
        add $s3, $s3, $t1  
        sw $s0, ($s3)  
else:  
    sll $t1, $s0, 2  
    add $s3, $s3, $t1  
    sw $s1, ($s3)
```