

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

Faculty of Computer Science & Engineering



Computer Architecture Week 3

Authors:

Lưu Nguyễn Hoàng Minh	1952845
Vũ Anh Nhi	1952380
Nguyễn Chính Khôi	1952793
Nguyễn Hoàng	1952255

Class:

CO2008_CC02

Lecturer:

Kiều Đỗ Nguyên Bình



Question 1:

Write a MIPS program with the following requirements:

- Declare an integer array with 10 synthetic data elements.
- Calculate the sum of all elements.
- Print the result to the terminal.

```
1
2  .data:
3  X:      .word 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
4  msg1:   .asciiz "The sum of all elements is "
5
6  # Start program code
7  .text:
8
9  .globl main
10 main:
11     la    $s7, X           # Address of X
12     add   $s0, $zero, $zero # Counter
13     add   $s1, $zero, $zero # Sum
14     addi  $t7, $zero, 40    # Limiter
15 Sum:
16     add   $t0, $s0, $s7     # Address of element
17     lw    $t1, 0($t0)
18     add   $s1, $s1, $t1     # Summer
19     addi  $s0, $s0, 4       # i = i + 1
20     beq   $s0, $t7, SExit
21     j     Sum
22
23 SExit:
24     # Print string msg1
25     li    $v0, 4
26     la    $a0, msg1
27     syscall
28
29     # Print sum
30     li    $v0, 1
31     move  $a0, $s1
32     syscall
33
```



```
34      # Exit
35      li      $v0, 10
36      syscall
```

Question 2:

Write a MIPS program with the following requirements:

- Declare an integer array with 10 synthetic data elements.
- Calculate the sum of all odd elements ($a[1], a[3], \dots$).
- Calculate the sum of all even elements ($a[0], a[2], \dots$).
- Print the results to the terminal.

```
1
2      .data:
3      X:      .word 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
4      msg1:    .asciiz "The sum of all elements of even indexes
5               is "
6      msg2:    .asciiz "The sum of all elements of odd indexes is
7               "
8      lf:      .asciiz "\n"
9
10     # Start program code
11     .text:
12
13     .globl main
14 main:
15     la      $s7, X          # Address of X
16     add     $s0, $zero, $zero # Counter
17     add     $s1, $zero, $zero # Sum of even
18     add     $s2, $zero, $zero # Sum of odd
19     addi    $t7, $zero, 40   # Limiter
20
21     Sum:
22     add     $t0, $s0, $s7    # Address of element
23     lw      $t1, 0($t0)
24     add     $s1, $s1, $t1    # Even summer
25     addi    $s0, $s0, 4      # i = i + 1
26     beq     $s0, $t7, SExit
```



```
25
26     add    $t0, $s0, $s7    # Address of element
27     lw     $t1, 0($t0)
28     add    $s2, $s2, $t1    # Odd summer
29     addi   $s0, $s0, 4      # i = i + 1
30     beq    $s0, $t7, SExit
31
32     j      Sum
33
34 SExit:
35     # Print string msg1
36     li     $v0, 4
37     la     $a0, msg1
38     syscall
39
40     # Print sum of odd
41     li     $v0, 1
42     move   $a0, $s1
43     syscall
44
45     # Print string new line
46     li     $v0, 4
47     la     $a0, lf
48     syscall
49
50     # Print string msg2
51     li     $v0, 4
52     la     $a0, msg2
53     syscall
54
55     # Print sum of even
56     li     $v0, 1
57     move   $a0, $s2
58     syscall
59
60     # Exit
61     li     $v0, 10
62     syscall
```



Question 3:

Write a MIPS program that receives 10 integer numbers from users through the terminal and store these numbers into an array. Print the sum of all array elements to the terminal.

```
1
2      .data:
3      X: .space 40 # Reserve 40 bytes
4      msg1: .asciiz "Enter an integer: "
5      msg2: .asciiz "The sum of all elements is "
6
7      # Start program code
8      .text:
9
10     .globl main
11     main:
12         la      $s7, X           # Address of X
13         add     $s0, $zero, $zero # Counter
14         add     $s1, $zero, $zero # Sum
15         addi    $t7, $zero, 40    # Limiter
16     Input:
17         add     $t0, $s0, $s7     # Address of element
18
19         # Print string msg1
20         li      $v0, 4
21         la      $a0, msg1
22         syscall
23
24         # Get input and save
25         li      $v0, 5
26         syscall
27         move    $t1, $v0
28
29         sw      $t1, 0($t0)
30         addi    $s0, $s0, 4        # i = i + 1
31         beq     $s0, $t7, IExit
32         j       Input
33
34     IExit:
35         add     $s0, $zero, $zero # Counter
36         add     $s1, $zero, $zero # Sum
```



```
37 Sum:
38     add    $t0, $s0, $s7    # Address of element
39     lw     $t1, 0($t0)
40     add    $s1, $s1, $t1    # Summer
41     addi   $s0, $s0, 4      # i = i + 1
42     beq    $s0, $t7, SExit
43     j      Sum
44
45 SExit:
46     # Print string msg2
47     li     $v0, 4
48     la     $a0, msg2
49     syscall
50
51     # Print sum
52     li     $v0, 1
53     move   $a0, $s1
54     syscall
55
56     # Exit
57     li     $v0, 10
58     syscall
```

Question 4:

Write a MIPS program with the following requirements:

- Declare an integer array with 10 synthetic data elements.
- Print a sentence to terminal to request an integer number that is greater than 0 and less than 10 (assume that user strictly follow this rule).
- Print value of the element at index collected from the previous step.

```
1
2     .data:
3     X: .word 0, 11, 22, 33, 44, 55, 66, 77, 88, 99
4     msg1: .asciiz "Enter an integer between 0 and 10: "
5     msg2: .asciiz "Data at index is "
6
7     # Start program code
```



```
8      .text:
9
10     .globl main
11     main:
12         la      $s7, X           # Address of X
13         add     $s0, $zero, $zero # Counter
14         addi    $t6, $zero, 4     # Sum limiter
15         addi    $t7, $zero, 40    # Limiter
16
17         # Print string msg1
18         li      $v0, 4
19         la      $a0, msg1
20         syscall
21
22         # Get input and save
23         li      $v0, 5
24         syscall
25         move    $t1, $v0
26         mul     $t6, $t1, $t6
27
28     Search:
29         add     $t0, $s0, $s7     # Address of element
30         lw      $t1, 0($t0)
31         beq     $s0, $t6, SExit
32         beq     $s0, $t7, SExit
33         addi    $s0, $s0, 4       # i = i + 1
34         j       Search
35
36     SExit:
37         # Print string msg2
38         li      $v0, 4
39         la      $a0, msg2
40         syscall
41
42         # Print sum
43         li      $v0, 1
44         move    $a0, $t1
45         syscall
46
47         # Exit
48         li      $v0, 10
49         syscall
```



Question 5:

Write a MIPS program that reverses an 10 elements integer array. For example, the array initially stores 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, the program will change the array to be 19, 17, 15, 13, 11, 9, 7, 5, 3, 1.

```
1
2  .data
3  arr: .word 34, 23, 15, 82, 38, 56, 48, 93, 49, 11
4  newarr: .word 0:10
5  space: .asciiz " "
6  endl: .asciiz "\n"
7
8  .text
9  .globl main
10
11 main:
12     la $s0, arr
13     add $s1, $zero, $zero
14
15 L1: #print the original array
16     lw $t1, 0($s0)
17     li $v0, 1
18     move $a0, $t1
19     syscall
20
21     addi $s0, $s0, 4
22     addi $s1, $s1, 1
23
24     li $v0, 4
25     la $a0, space
26     syscall
27
28     beq $s1, 10, r1
29     j L1
30
31 r1:
32     li $v0, 4
33     la $a0, endl
34     la $s3, newarr
35     syscall
36     addi $s0, $s0, -4
```




```
37
38 rev: # reverse the array
39     lw $t2, 0($s0)
40     sw $t2, 0($s3)
41     addi $s0, $s0, -4
42     addi $s3, $s3, 4
43     addi $s1, $s1, -1
44
45     beq $s1, 0, print
46     j rev
47
48 print:
49     la $s3, newarr
50     addi $s1, $zero, 0
51     j print2
52
53 print2: # print the reverse array
54     lw $t1, 0($s3)
55     li $v0, 1
56     move $a0, $t1
57     syscall
58
59     addi $s3, $s3, 4
60     addi $s1, $s1, 1
61
62     li $v0, 4
63     la $a0, space
64     syscall
65
66     beq $s1, 10, Exit
67     j print2
68
69 Exit:
70     li $v0, 10
71     syscall
```