

# Bài thực hành số 7

Lớp : 139365 – Học phần: Thực hành Kiến trúc máy tính

Họ và tên : Nguyễn Thị Thùy Dung

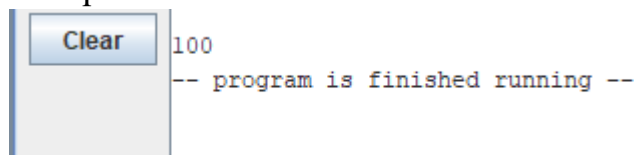
MSSV : 20215009

## Bài 1:

```
1  #Laboratory Exercise 7 Home Assignment 1
2  .text
3  main:
4      li $a1,-100
5      jal abs
6      nop
7      add $s0, $zero, $v0
8      li $v0,1
9      move $a0,$s0
10     syscall
11     li $v0,10
12     syscall
13 endmain:
14 abs:
15     sub $v0,$zero,$a1
16     bltz $a1,done
17     nop
18     add $v0,$a1,$zero
19 done:
20     jr $ra
```

Thực hiện gõ chương trình với công cụ MARS

- Kết quả:



- Giải thích :

- + Khởi tạo \$a1 bằng -100, sau đó nhảy đến chương trình con abs
- + Ở chương trình con (dòng 14 - 20), thực hiện tính  $\$v0 = 0 - \$a1$ . Sau đó so sánh \$a1 với 0, nếu  $\$a1 < 0$  thì \$v0 chính là giá trị tuyệt đối của

\$a1, trở về chương trình chính bằng jr \$ra, nếu không thì \$v0 = 0 + \$a1 và quay trở về chương trình chính

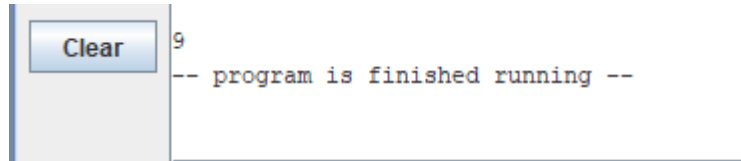
+ Khi quay trở lại chương trình chính, sẽ gọi syscall ( dòng 7-12) để in giá trị tuyệt đối ra màn hình và thoát chương trình. Kết quả in ra là 100

## Bài 2:

```
1  .text
2  main:
3      li $a0,2
4      li $a1,6
5      li $a2,9
6      jal max
7      nop
8      move $s0,$v0
9      li $v0,1
10     move $a0,$s0
11     syscall
12     li $v0,10
13     syscall
14 endmain:
15 max:
16     add $v0,$a0,$zero
17     sub $t0,$a1,$v0
18     bltz $t0,okay
19     nop
20     add $v0,$a1,$zero
21 okay:
22     sub $t0,$a2,$v0
23     bltz $t0,done
24     nop
25     add $v0,$a2,$zero
26 done:
27     jr $ra
28
```

Thực hiện gõ chương trình với công cụ MARS

- Kết quả :



- Giải thích:

+ Khai báo 3 số :  $\$a0 = 2$ ,  $\$a1 = 6$ ,  $\$a2 = 9$ , sau đó nhảy đến chương trình con max

+ Ở chương trình con (dòng 15 - 27), tính  $\$v0 = \$a0 + 0$ , sau đó tính  $\$t0 = \$a1 - \$v0$ , so sánh xem  $\$t0$  có nhỏ hơn 0 hay không. Nếu  $\$t0$  nhỏ hơn 0 tức là  $\$a1$  nhỏ hơn  $\$v0$ , tiếp tục kiểm tra xem  $\$a2$  có nhỏ hơn  $\$v0$ . Tính  $\$t0 = \$a2 - \$v0$ , so sánh  $\$t0$  với không, nếu  $\$t0$  nhỏ hơn 0 thì  $\$a2$  nhỏ hơn  $\$v0$  ( $\$a0$ ) thì  $\$v0(\$a0)$  chính là giá trị lớn nhất và quay trở về chương trình chính bằng jr \$ra. Nếu  $\$t0$  lớn hơn 0 thì  $\$a2$  là lớn nhất và thực hiện tính  $\$v0 = \$a2 + 0$ . Nếu  $\$v0$  ( $\$a0$ )  $<$   $\$a1$  thì cho  $\$v0 = \$a1 + 0$  và so sánh tiếp với  $\$a2$ . Như vậy giá trị  $\$v0$  chính là giá trị lớn nhất trong 3 số

+ Quay trả về chương trình chính, thực hiện syscall (dòng 9 – 13) in ra giá trị lớn nhất ra màn hình và thoát chương trình. Kết quả trả về là 9

### Bài 3:

```
1
2 .text
3         li $s0,4
4         li $s1,5
5 push:
6         addi $sp,$sp,-8
7         sw $s0,4($sp)
8         sw $s1,0($sp)
9 pop:
10        lw $s0,0($sp)
11        lw $s1,4($sp)
12        addi $sp,$sp,8
13
14
15
16
```

Thực hiện gõ chương trình với công cụ MARS

- Kết quả :

+ Sau khi push :

\$s0	16	4
\$s1	17	5
\$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	2147479540
\$fp	30	0
\$ra	31	0
pc		4194324
hi		0
lo		0

+ Sau khi pop :

\$s0	16	5
\$s1	17	4
\$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 :

+ Khai báo \$s0 = 4, \$s1 = 5

+ Ở push, tạo ra 2 ô nhớ bằng cách  $\$sp = \$sp - 8$  (bởi vì đỉnh \$sp ở địa chỉ giảm), sau đó lưu \$s0 vào  $\$sp + 4$ , lưu \$s1 vào \$sp. Như vậy thực hiện đẩy 2 số vào đỉnh stack

+ Ở pop, thực hiện ngược lại, lưu giá trị \$sp vào \$s0, lưu \$sp +4 vào \$s1, như vậy thực hiện lấy ra 2 số của stack. Tính  $\$sp = \$sp + 8$  để giảm đỉnh của stack đi 2 ô nhớ. Kết quả được như hình vẽ

#### Bài 4:

```
1  .data
2  Message: .asciiz "Ket qua tinh giai thua la: "
3  .text
4  main:
5      jal WARP
6
7  print:
8      add $a1, $v0, $zero    # $a0 = result from N!
9      li $v0, 56
10     la $a0, Message
11     syscall
12
13 quit:
14     li $v0, 10              #terminate
15     syscall
16
17 endmain:
18
19 WARP:
20     sw $fp, -4($sp)         #save frame pointer (1)
21     addi $fp, $sp, 0        #new frame pointer point to the top (2)
22     addi $sp, $sp, -8       #adjust stack pointer (3)
23     sw $ra, 0($sp)         #save return address (4)
24     li $a0, 3               #load test input N
25     jal FACT               #call fact procedure
26     nop
```

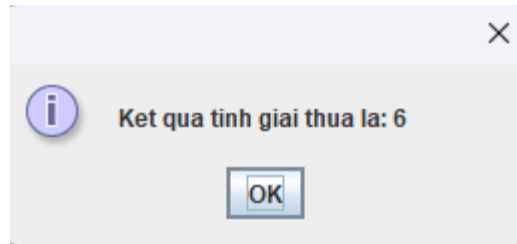
```

25
26     lw $ra,0($sp)           #restore return address (5)
27     addi $sp,$fp,0          #return stack pointer (6)
28     lw $fp,-4($sp)          #return frame pointer (7)
29     jr $ra
30 wrap_end:
31 FACT:
32     sw $fp,-4($sp)           #save frame pointer
33     addi $fp,$sp,0           #new frame pointer point to stack's top
34     addi $sp,$sp,-12         #allocate space for $fp,$ra,$a0 in stack
35     sw $ra,4($sp)            #save return address
36     sw $a0,0($sp)            #save $a0 register
37
38     slti $t0,$a0,2           #if input argument N < 2
39     beq $t0,$zero,recursive  #if it is false ((a0 = N) >=2)
40     nop
41     li $v0,1                 #return the result N!=1
42     j done
43     nop
44 recursive:
45     addi $a0,$a0,-1          #adjust input argument
46     jal FACT                  #recursive call
47     nop
48     lw $v1,0($sp)            #load a0
49     mult $v1,$v0              #compute the result
50
51 done:
52     lw $ra,4($sp)            #restore return address
53     lw $a0,0($sp)            #restore a0
54     addi $sp,$fp,0           #restore stack pointer
55     lw $fp,-4($sp)           #restore frame pointer
56     jr $ra                   #jump to calling
57 fact_end:
58

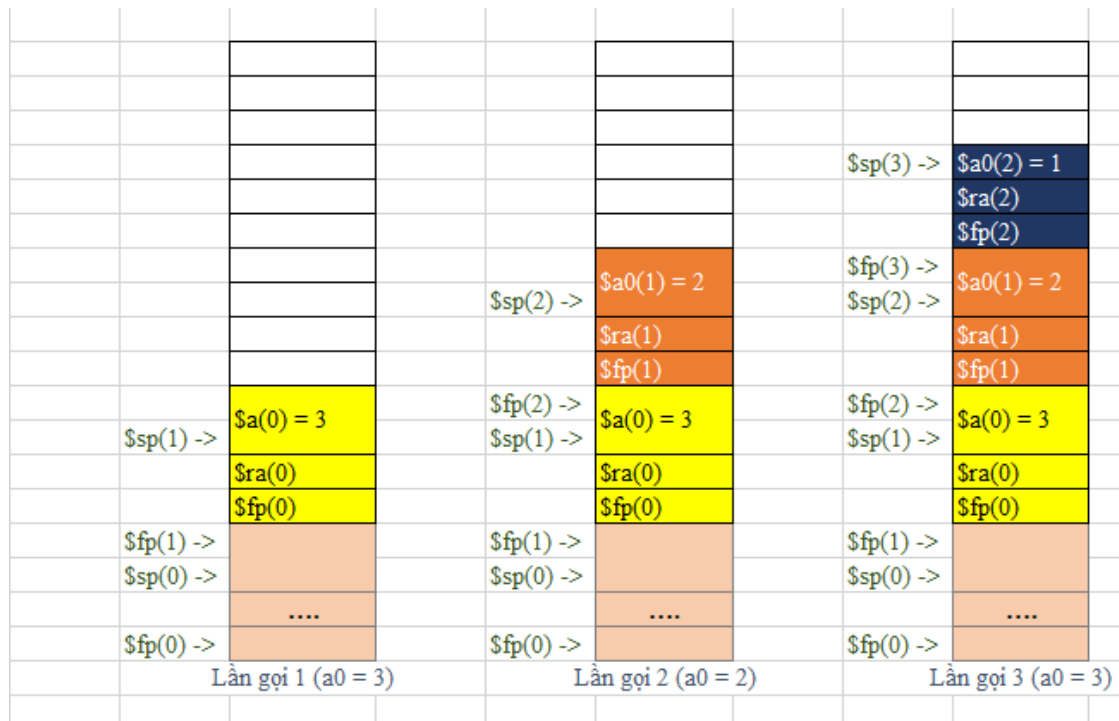
```

Thực hành chương trình bằng công cụ MARS

- Kết quả :



- Vẽ stack :



- Giải thích:

- + Khai báo input \$a0 = 3, Message : “Ket qua tinh giai thua la :”
- + Nhảy đến chương trình con WARP . Đầu tiên lưu \$fp vào \$sp – 4 ( lưu frame point cũ ), tính \$fp = \$sp + 0 ( tạo ra frame point mới từ đỉnh của cái cũ ), lưu \$ra vào \$sp ( lưu giá trị trả về vào đỉnh stack )
- + Nhảy đến chương trình con FACT. Thực hiện việc tạo ra stack mới để lưu trữ các giá trị \$ra, \$fp, \$a0. So sánh nếu \$a0 < 2 thì trả về \$v0 = 1 (n!=1). Nếu không thì thực hiện gọi lại hàm với n-1 ( \$a0 = \$a0 -1). Cứ thực hiện như vậy cho đến khi n < 2. Thực hiện lưu \$sp vào \$v1 (Lấy \$a0), nhân \$v1 \* \$ v0 . Lấy giá trị đó lưu vào \$v0 ( 3 \* 2 \* 1) Sau đó thực hiện lấy lại các giá trị \$ra, \$a0, \$fp từ \$sp và thực hiện quay lại chương trình chính bằng lệnh jr \$ra.
- + Thực hiện syscall để in giai thừa ra màn hình. Kết quả in ra 6 ra dialog

## Bài 5:

```
1  #Laboratory Exercise 7 Assignment 5
2  .data
3  message: .asciiz "Largest : "
4  message1: .asciiz "\n\nSmallest : "
5  message2: .asciiz " , "
6  .text
7      li $s0, 3
8      li $s1, 3
9      li $s2, 6
10     li $s3, 9
11     li $s4, 98
12     li $s5, 8
13     li $s6, 2
14     li $s7, 0
15     move $t0, $s0          # max = s0
16     move $t1, $s0          # min = s0
17     li $t3, 0              # i = 0
18     li $t4, 8              # n = 8
19     li $a2, 0              # index của max = 0
20     li $a3, 0              # index của min = 0
21 push:
22     addi $sp, $sp, -32
23     sw $s0, 0($sp)
24     sw $s1, 4($sp)
25     sw $s2, 8($sp)
26     sw $s3, 12($sp)
27     sw $s4, 16($sp)
28     sw $s5, 20($sp)
29     sw $s6, 24($sp)
30     sw $s7, 28($sp)
31 main:
32     jal work
33     j print
34 work:
35     slt $t5, $t3, $t4      # t3 < t4 ?
36     beq $t5, $zero, end_work # nha hon ket thuc
37     add $t6, $t3, $t3      # t6 = t3 + t3
38     add $t6, $t6, $t6      # t6 = t6 + t6 = 4*t3
39     add $t7, $t6, $sp      # t7 = t6 + sp
40     lw $t8, 0($t7)         # lay gia tri o dia chi t7 luu vao t8
41     slt $t5, $t8, $t0      #kiem tra t8 < t0 ( max > stack)
42     beq $t5, $zero, wrap
43 continue1: slt $t5, $t8, $t1 # t8 < t1 ( stack < min )
44     bne $t5, $zero, wrap1
45 continue: addi $t3, $t3, 1  # i = i + 1
46     j work
47
48 wrap:
```



```

48 wrap:
49     add $t0, $t8, $zero      # max = t8
50     move $a2, $t3           # a2 = i
51     j continue1
52 wrap1:
53     add $t1, $t8, $zero      # min = t8
54     move $a3, $t3           # a3 = i
55     j continue
56 end_work:
57     jr $ra
58 print:    li $v0, 4
59           la $a0, message
60           syscall
61           li $v0, 1
62           move $a0, $t0
63           syscall
64           li $v0, 4
65           la $a0, message2
66           syscall
67           li $v0, 1
68           move $a0, $a2
69           syscall

```

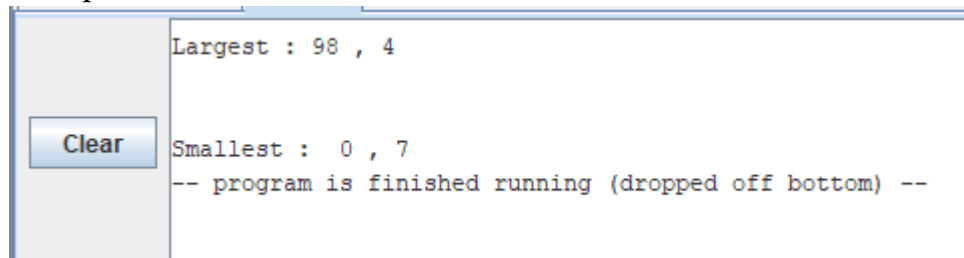
```

69           syscall
70           li $v0, 4
71           la $a0, message1
72           syscall
73           li $v0, 1
74           move $a0, $t1
75           syscall
76           li $v0, 4
77           la $a0, message2
78           syscall
79           li $v0, 1
80           move $a0, $a3
81           syscall
82

```

Thực hành chương trình bằng công cụ MARS

- Kết quả :



```
Largest : 98 , 4
Smallest : 0 , 7
-- program is finished running (dropped off bottom) --
```

Clear

- Giải thích :

- + Khởi tạo các giá trị từ \$s0 đến \$s7 là 8 giá trị của mảng,  $i = 0$ ,  $n = 8$ ,  $\text{max} = \$s0$ ,  $\text{min} = \$s0$ ,  $\$a2 = 0$ ,  $\$a3 = 0$
- + Thực hành push các giá trị của mảng vào stack bằng \$sp.  $\$sp = \$sp - 32$  để tạo ra 8 ô nhớ cho 8 phần tử của mảng. Lần lượt push bằng sw (lưu giá trị \$s0 vào \$sp, \$s1 vào \$sp + 4,.....)
- + Thực hiện hàm work , so sánh  $\$t3 < \$t4$  ( $i < n$  ). Nếu  $i > n$  thì thoát vòng lặp còn ngược lại tiếp tục thực hiện vòng lặp duyệt stack và tăng  $i = i + 1$  ( $\$t3 = \$t3 + 1$ ). Ở mỗi vòng lặp thực hiện  $\$t6 = \$t6 + \$t6 = 4 * \$t3$ ,  $\$t7 = \$sp + \$t6$ , lw để lấy giá trị của stack. Với mỗi giá trị đó so sánh \$t8 với max (\$t0) và min (\$t1). Nếu \$t8 lớn hơn max (\$t0) thì cập nhập lại max  $\$t0 = \$t8 + 0$  và cập nhập chỉ số lớn nhất bằng i (\$t3) . Nếu \$t8 nhỏ hơn min (\$t1) thì thực hiện cập nhập lại min  $\$t1 = \$t8 + 0$  và chỉ số nhỏ nhất  $\$a3 = \$t3 = i$  .
- + Thoát chương trình con và thực hiện chương trình chính bằng lệnh jr \$ra, thực hiện in max min ra màn hình bằng syscall (dòng 58 - 81)

