

Laboratory Exercise 7

Procedure calls, stacks and parameters

Đỗ Hải Dương – 20194528

Assignment 1

```
1  #Laboratory Exercise 7 Home Assignment 1
2  .text
3  main:  li      $a0, -20194528  #load input parameter
4         jal     abs           #jump and link to abs procedure
5         nop
6         add     $s0, $zero, $v0
7         li      $v0, 10        #terminate
8         syscall
9  endmain:
10 #-----
11 # function abs
12 # param[in] $a1 the interger need to be gained the absolute
13 value:
14 # return $v0 absolute value
15 #-----
16 abs:
17         sub     $v0, $zero, $a0  #put -(a0) in v0; in case (a0)<0
18         bltz    $a0, done        #if (a0)<0 then done
19         nop
20         add     $v0, $a0, $zero  #else put (a0) in v0
21 done:
22         jr      $ra
```

Kết quả:

Registers	Coproc 1	Coproc 0	
Name	Number		Value
\$zero	0		0
\$at	1		-20250624
\$v0	2		10
\$v1	3		0
\$a0	4		-20194528
\$a1	5		0
\$a2	6		0
\$a3	7		0
\$t0	8		0
\$t1	9		0
\$t2	10		0
\$t3	11		0
\$t4	12		0
\$t5	13		0
\$t6	14		0
\$t7	15		0
\$s0	16		20194528
\$s1	17		0
\$s2	18		0
\$s3	19		0

Giải thích :

- Gán giá trị cho thanh ghi a0
- Lệnh jal đưa chương trình đến trường trình con abs . +
- Gán v0 = 0 - a0 . Nếu a0 < 0 thì nhảy đến nhãn done , giá trị tuyệt đối là v0
- Nếu a0 > 0 thì gán v0 = 0 + a0 . đến nhãn done và về chương trình chính
- Gán s0 = v0 và đưa ra màn hình giá trị tuyệt đối

Assignment 2

```
1  #Laboratory Exercise 7, Home Assignment 2
2  .text
3  main:  li      $a0,4    #load test input
4         li      $a1,9
5         li      $a2,3
6         jal     max      #call max procedure
7         add     $t1, $zero, $v0
8         nop
9         li      $v0, 10
10        syscall
11  endmain:
12  #-----
13  #Procedure max: find the largest of three integers
14  #param[in] $a0 integers
15  #param[in] $a1 integers
16  #param[in] $a2 integers
17  #return $v0 the largest value
18  #-----
19  max:   add     $v0,$a0,$zero  #copy (a0) in v0; largest so far
20        sub     $t0,$a1,$v0    #compute (a1)-(v0)
21        bltz    $t0,okay       #if (a1)-(v0)<0 then no change
22        nop
23        add     $v0,$a1,$zero  #else (a1) is largest thus far
24  okay:  sub     $t0,$a2,$v0    #compute (a2)-(v0)
25        bltz    $t0,done       #if (a2)-(v0)<0 then no change
26        nop
27        add     $v0,$a2,$zero  #else (a2) is largest overall
28  done:  jr      $ra           #return to calling program
```

Kết quả :

\$a3	7	0
\$t0	8	-6
\$t1	9	9
\$t2	10	0

Giải thích:

- Gán 3 số cho trước cho thanh ghi a0,a1,a2
- Gán v0 = a0 (đặt a0 là max)
- So sánh v0 – a1 < 0 (nếu đúng thì gán max = a1)
- So sánh v0 với a2(nếu v0 <a2) thì gán v0 a2 (max =a2)

Assignment 3

```
1  #Laboratory Exercise 7, Home Assignment 3
2  .text
3      li      $s0, 2019
4      li      $s1, 4528
5
6  push:      addi    $sp,$sp,-8      #adjust the stack pointer
7              sw     $s0,4($sp)     #push $s0 to stack
8              sw     $s1,0($sp)     #push $s1 to stack
9  work:      nop
10             nop
11             nop
12  pop:       lw     $s0,0($sp)      #pop from stack to $s0
13             lw     $s1,4($sp)     #pop from stack to $s1
14             addi    $sp,$sp,8      #adjust the stack pointer
```

Kết quả :

\$s0	16	4528
\$s1	17	2019

- Khai báo 1 stack chứa được 2 số nguyên (giảm \$sp đi 8)
- Lưu lần lượt \$s0, \$s1 vào \$sp[1], \$sp[0]
- Cho s0 ,sp1 vào stack .
- Rồi lấy ra bằng cách lưu s0 = \$sp[1],s1= \$sp[0]

Assignment 4

```
1  #Laboratory Exercise 7, Home Assignment 4
2  .data
3  Message: .asciiz "Ket qua tinh giai thua la: "
4  .text
5  main:    jal      WARP
6
7  print:   add      $a1, $v0, $zero # $a0 = result from N!
8          li       $v0, 56
9          la       $a0, Message
10         syscall
11  quit:    li       $v0, 10         #terminate
12         syscall
13  endmain:
14  #-----
15  #Procedure WARP: assign value and call FACT
16  #-----
17  WARP:    sw       $fp, -4($sp)     #save frame pointer (1)
18          addi     $fp, $sp, 0      #new frame pointer point to the top (2)
19          addi     $sp, $sp, -8     #adjust stack pointer (3)
20          sw       $ra, 0($sp)      #save return address (4)
21          li       $a0, 3           #load test input N
22          jal      FACT             #call fact procedure
23          nop
24
25          lw       $ra, 0($sp)      #restore return address (5)
26          addi     $sp, $fp, 0      #return stack pointer (6)
27          lw       $fp, -4($sp)     #return frame pointer (7)
28          jr       $ra
29  wrap_end:
30  #-----
31  #Procedure FACT: compute N!
32  #param[in] $a0 integer N
33  #return $v0 the largest value
34  #-----
35  FACT:    sw       $fp, -4($sp)     #save frame pointer
36          addi     $fp, $sp, 0      #new frame pointer point to stack's
37          addi     $sp, $sp, 12
38          sw       $ra, 4($sp)
39          sw       $a0, 0($sp)
40
41          slti     $t0, $a0, 2      # if input argument N < 2
42          beq      $t0, $zero, recursive # false ((a0 = N) >= 2)
43          nop
44          li       $v0, 1           # return the result N != 1
45          j        done
46          nop
47  recursive:
48          addi     $a0, $a0, -1      #adjust input argument
49          jal      FACT             #recursive call
50          nop
51          lw       $v1, 0($sp)      #load a0
52          mult     $v1, $v0         #compute the result
53          mflo     $v0
54  done:    lw       $ra, 4($sp)      #restore return address
55          lw       $a0, 0($sp)      #restore a0
56          addi     $sp, $fp, 0      #restore stack pointer
57          lw       $fp, -4($sp)     #restore frame pointer
58          jr       $ra             #jump to calling
59  fact_end:
```

Kết quả:



Ý nghĩa lệnh:

- Thủ tục warp:
 - Lưu con trỏ khung cũ
 - Tạo con trỏ khung mới $\$fp = \sp
 - Khai báo con trỏ stack mới
 - Lưu địa chỉ trả về $\$ra$ vào $\$sp[0] + 0$
 - Lưu giá trị input $\$a0 = 8$
 - Gọi đến thủ tục FACT
- Thủ tục FACT:
 - Lặp lại các thao tác:
 - Lưu con trỏ khung cũ
 - Tạo con trỏ khung mới $\$fp = \sp
 - Khai báo stack mới chứa $\$a0$, $\$ra$, $\$fp$
 - Khi $\$a0 \geq 2$, thực hiện recursive : $\$a0 = \$a0 - 1$
 - Khi $\$a0 = 1$, $\$v0 = 1$ và nhảy đến nhãn done
- Thủ tục done: lặp lại việc lấy các giá trị $\$ra$, $\$a0$, $\$sp$, $\$fp$ đã lưu, thực hiện tính $n! = 1.2 \dots n$