

Práctica 7: Convenciones de llamada a subrutina

Organización y Arquitectura de Computadoras

Hernández Ferreiro Enrique Ehécatl
López Soto Ramses Antonio

14 de abril de 2019

1 Desarrollo

El ejercicio 1 se muestra a continuación:

```
.data

a:      .word    5
b:      .word    4

.text

main:   #Preambulo
        #Invocaci n
        #Retorno de mist_1
        #Conclusi n main
        #mist_1 recibe como argumentos $a0 y $a1
        lw      $a0 a
        lw      $a1 b
        jal     mist_1
        j       end_1
mist_1: subi    $sp $s0 24
        sw      $ra 20($sp)
        sw      $s0 16($sp)
        #Preambulo mist_1
        move    $s0 , $a0
        move    $t0 , $a1
        li      $t1 , 1
loop_1: beqz    $s0 , end_1
        #invocamos de mist_0
```

```

        move    $a0 , $t0          # Se pasa el argumento $a0
        move    $a1 , $t1          # Se pasa el argumento $a1
        jal     mist_0
        # Retorno de mist_0
        move    $t1 , $v0
        subi    $s0 , $s0 , 1
        j       loop_1
end_1:   # Conclusi n mist_1
        move    $v0 , $t1
        lw      $s0 , 16($sp)
        lw      $ra , 20($sp)
        addi    $sp , $sp , 24

        mult    $a0 , $a1
        # Conclusi n mist_0
        mflo    $v0                # Se retorna el resultado en $v0

```

El ejercicio 2 se muestra a continuación:

```

.data
n:      .asciiz "Introduzca n: "
k:      .asciiz "Introduzca k: "
sl:     .asciiz "\n"
result: .asciiz "El resultado es "

.text
main:   addi $sp , $sp , -4
        sw $ra , 0($sp)
        jal combinaciones

entero:  li $v0 , 4
        syscall
        li $v0 , 5
        syscall
        jr $ra

combinaciones:  addi $sp , $sp , -4
               sw $ra , 0($sp)
               la $a0 , n
               jal entero
               move $s0 , $v0
               la $a0 , k

```

```

jal entero
move $a1, $v0
move $a0, $s0
jal coeficiente
move $t0, $v0
la $a0, result
li $v0, 4
syscall
move $a0, $t0
li $v0, 1
syscall
la $a0, s1
li $v0, 4
syscall
lw $ra, 0($sp)
addi $sp, $sp, 4
jr $ra

salir:
lw $ra, 0($sp)
addi $sp, $sp, 4
jr $ra

coeficiente:
addi $sp, $sp, -16
sw $s2, 12($sp)
sw $s1, 8($sp)
sw $s0, 4($sp)
sw $ra, 0($sp)
beq $a0, $a1, Cbase1
beq $a1, 0, Cbase0
move $s0, $a0
move $s1, $a1

subi $a0, $s0, 1           #n - 1 k-1
subi $a1, $s1, 1
jal coeficiente
move $s2, $v0

subi $a0, $s0, 1           #n - 1 k
move $a1, $s1
jal coeficiente

add $v0, $v0, $s2         #sumar ambas partes
j fin

```

```

Cbase1: li $v0, 1                                #Caso base 1
j fin
Cbase0: li $v0, 1                                #Caso base 0

fin: lw $ra, 0($sp)
lw $s0, 4($sp)
lw $s1, 8($sp)
lw $s2, 12($sp)
addi $sp, $sp, 16
jr $ra

```

2 Conclusión

2.1 Preguntas

1. ¿Qué utilidad tiene el registro \$fp? ¿Se puede prescindir de él?
R. Sirve para preservar datos que no serán modificados por una llamada del sistema. Y sí, se puede prescindir de él, pues puede ser decrementado una vez que el procedimiento en el que se utilizó se acabe.
2. Definimos como **subrutina nodo** a una subrutina que realiza una o más invocaciones a otras subrutinas y como **subrutina hoja** a una subrutina que no realiza llamadas a otras subrutinas.
 - a) ¿Cuál es el tamaño mínimo que puede tener un marco para una subrutina nodo?
¿Bajo qué condiciones ocurre?
¿Cuál es el tamaño mínimo que puede tener un marco para una subrutina hoja?
¿Bajo qué condiciones ocurre?