

ARCO Pràctica 1 - Informe

Ixent Cornella

Exercici 1 - **Escribe aquí la rutina compara(A,B) en ensamblador:**

```
.text
    .align 2
    .globl main
main:
    addi $s0, $zero, 10    # movem els valors
    addi $s1, $zero, 6
    add  $s2, $s0, $zero
loop:
    move $a0, $s2
    move $a1, $s1
    addi $sp, $sp, -4
    sw   $ra, 0($sp)
    jal  compara
    lw   $ra, 0($sp)
    addi $sp, $sp, 4
    move $s2, $v0
    bgt  $s1, $s2, exit
    j    loop
exit:
    jr   $ra
    .end main

compara:
    move $t0, $zero
    ble $a0, $a1, elif
    sub  $t1, $a0, $a1
    move $t0, $t1
    j    exit1
elif: bge $a0, $a1, exit1
    move $t0, $a0
exit1:    move $v0, $t0
    jr   $ra
```

Prueba con diferentes valores de A y B: A=10, B=6, A=10, B=2, A=10, B=3 y A=2 y B=5:

R16 [s0] = a	R16 [s0] = a	R16 [s0] = a	R16 [s0] = 2
R17 [s1] = 6	R17 [s1] = 2	R17 [s1] = 3	R17 [s1] = 5
R18 [s2] = 4	R18 [s2] = 0	R18 [s2] = 1	R18 [s2] = 2

Aquí podem veure el resultat de “mòdul” al final de cada execució (s2). El resultat és correcte.

Exercici 2 - implementa el main y la subrutina en el simulador:

```
main:
    addi $a0, $zero, 1
    addi $sp, $sp, -4
    sw $ra, 0($sp)
    jal leaf_function
    lw $ra, 0($sp)
    addi $sp, $sp, 4
    jr $ra
.end main
leaf_function:
    addi $t0, $a0, 1
    addi $t1, $zero, 5
    slt $t2, $t1, $a0
    beq $t2, $zero, exit
    move $v0, $t0
    jr $ra
exit: move $a0, $t0
    addi $sp, $sp, -4
    sw $ra, 0($sp)
    jal leaf_function
    lw $ra, 0($sp)
    addi $sp, $sp, 4
    jr $ra
.data
.align 0
```

2.1 - Si el programa main llama a la subrutina con un valor inicial de 1, como está en el enunciado en C, describe la ejecución dinámica del programa:

```

main:
    addi $a0, $zero, 1
    addi $sp, $sp, -4
    sw $ra, 0($sp)
    jal leaf_function
    # saltamos a leaf_function
leaf_function:
    addi $t0, $a0, 1
    addi $t1, $zero, 5
    slt $t2, $t1, $a0
    # t2 = 0 ja que t1 = 5 t0 = 2
    beq $t2, $zero, exit
    # saltamos a exit
exit: move $a0, $t0
    addi $sp, $sp, -4
    sw $ra, 0($sp)
    jal leaf_function
    # saltamos a leaf_function otras 4 veces
    lw $ra, 0($sp)
    addi $sp, $sp, 4
    jr $ra
    # hacemos el jump 4 veces otra vez...
    # terminamos con el último jump, ahora en el main.
    lw $ra, 0($sp)
    addi $sp, $sp, 4
    jr $ra
.end main

```