

**P1 - Informe Final:**

**Subrutinas**

Arquitectura de Computadors

**Daniel Aagaard Pérez i Pau Martín Nadal**

**Eva Marin Tordera**

**I4511**

**1. Traduce al ensamblador la rutina potencia(x,y) e impleméntala en el simulador junto con el programa main de la pregunta 2 del informe previo también en el ensamblador. Simúlalo paso a paso. NOTA: Aunque no sea una rutina que llame a otra ni tampoco recursiva seguiremos la metodología de al entrar en una subrutina guardar en la pila todos los registros $s0-s7 que vayamos a utilizar. Además si fuese una subrutina que llama a otra o a sí misma ha de guardar también en la pila la @ de retorno, $ra; así como los argumentos $a0 y $a1 si es necesario, por ejemplo en una recursiva.**

**1.1. Escribe aquí la rutina potencia(x, y) en ensamblador:**

.data

y: .word 0

.align 2

.text

.globl main

main:

addi $sp, $sp, -4

sw $ra, 0($sp)

addi $a0, $zero, 3 # a0 = x = 3

addi $a1, $zero, 2 # a1 = 2, per fer la potencia

jal potencia # cridem a la funció amb $a0 i $a1 i retorna a $v0

addi $t0, $zero, 10 # t0 = A = 10

addi $t1, $zero, 6 # t1 = B = 6

addi $t2, $zero, -1 # t2 = C = -1

mul $t3, $t1, $a0 # t3 = B\*x

mul $t0, $t0, $v0 # t0 = A\*potencia(x,2)

add $t0, $t0, $t3 # A\*potencia(x,2) + B\*x

add $t0, $t0, $t2 # A\*potencia(x,2) + B\*x + C

sw $t0, y # Guardem el resultat en la y.

lw $ra, 0($sp)

addi $sp, $sp, 4

jr $ra

.end main

potencia:

addi $sp, $sp, -4

sw $ra, 0($sp)

addi $v0, $zero, 1

beq $a1, $zero, fi

add $t1, $zero, $zero

for: slt $t0, $t1, $a1

beq $t0, $zero, fi

mul $v0, $v0, $a0

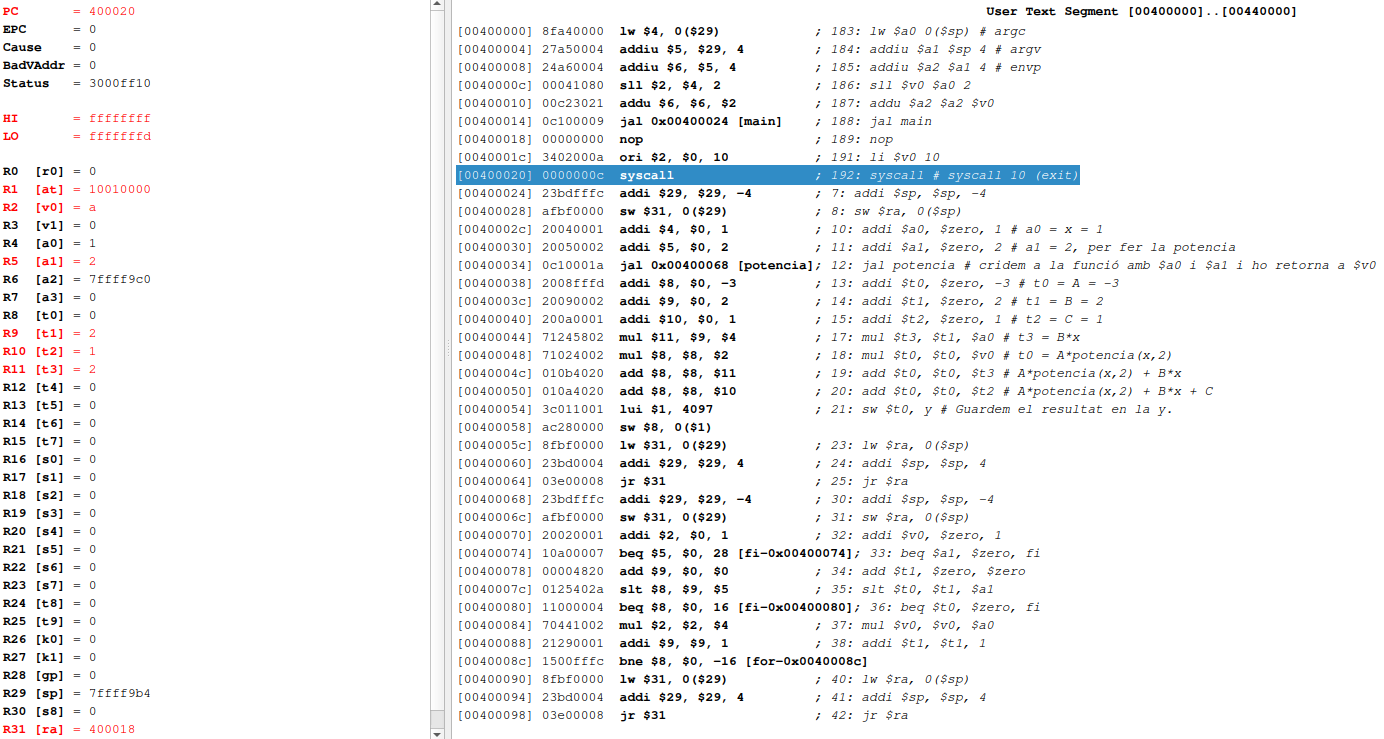
addi $t1, $t1, 1

bne $t0, $zero, for

fi: lw $ra, 0($sp)

addi $sp, $sp, 4

jr $ra

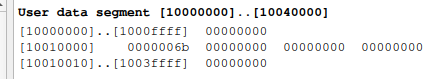


**1.2. Prueba con diferentes valores de A,B,C y X: A=10, B=6, C=-1, X=3,**

**A=-3, B=2, C=1, X=1, A=10, B=-3, C=2 y X=0; cuando funcione avisa al**

**profesor.**

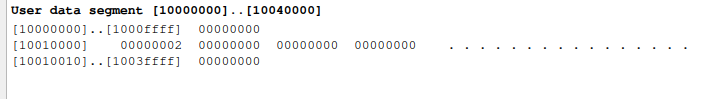
**A=10, B=6, C=-1, X=3 –> 107 –> 0x6B**



**A=-3, B=2, C=1, X=1 –> 0 –> 0x00**

****

**A=10, B=-3, C=2, X=0 –> 2 –> 0x02**

****

**2. A partir del ejercicio 3 del trabajo previo, implementa el main y la subrutina en el simulador.**

**a) En el archivo ejercicio3.s debes completar la traducción. Simúlalo**

**paso a paso.**

.data

result: .word 0

.text

.globl main

main:

addi $sp, $sp, -4

sw $ra, 0($sp)

addi $a0, $zero, 2 # x = 2

addi $a1, $zero, 5 # y = 5

jal potencia\_recursiva

sw $v0, result # guardem a memoria

lw $ra, 0($sp)

addi $sp, $sp, 4

jr $ra

.end main

potencia\_recursiva:

addi $sp, $sp, -12 # fem espai dos arguments i l’adreça de retorn

sw $ra, 8($sp)

sw $a0, 4($sp)

sw $a1, 0($sp)

bne $a1, $zero, cond2 # if(y != 0) cond 2

addi $v0, $zero, 1 # v0 = 1

addi $sp, $sp, 12 # pop 3 items de la pila

jr $ra

cond2: addi $t0, $zero, 1

bne $t0, $a1, else

add $v0, $zero, $a0

addi $sp, $sp, 12 # pop 3 items de la pila

jr $ra

else:

srl $a1, $a1, 1 # y = y >> 1 // y = y/2

jal potencia\_recursiva

lw $t0, 0($sp) # t0 = y

addi $sp, $sp, -4 # push $v0

sw $v0, 0($sp) # guardem $v0 a la pila

srl $a1, $t0, 1 # a1 = y/2

sub $a1, $t0, $a1 # a1 = y-(y/2)

jal potencia\_recursiva

lw $t1, 0($sp) # t1 = potencia\_recursiva(x,y/2)

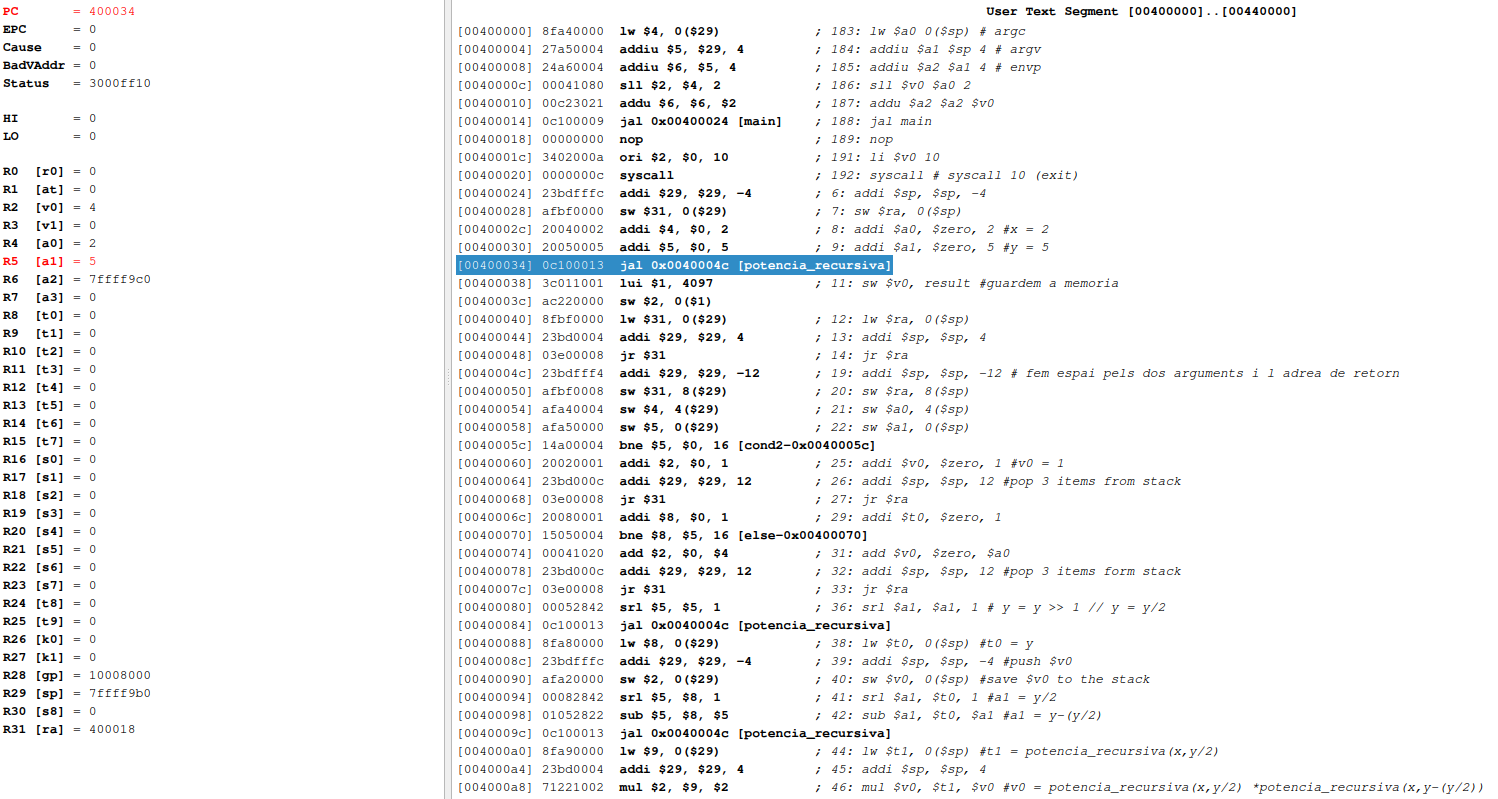
addi $sp, $sp, 4

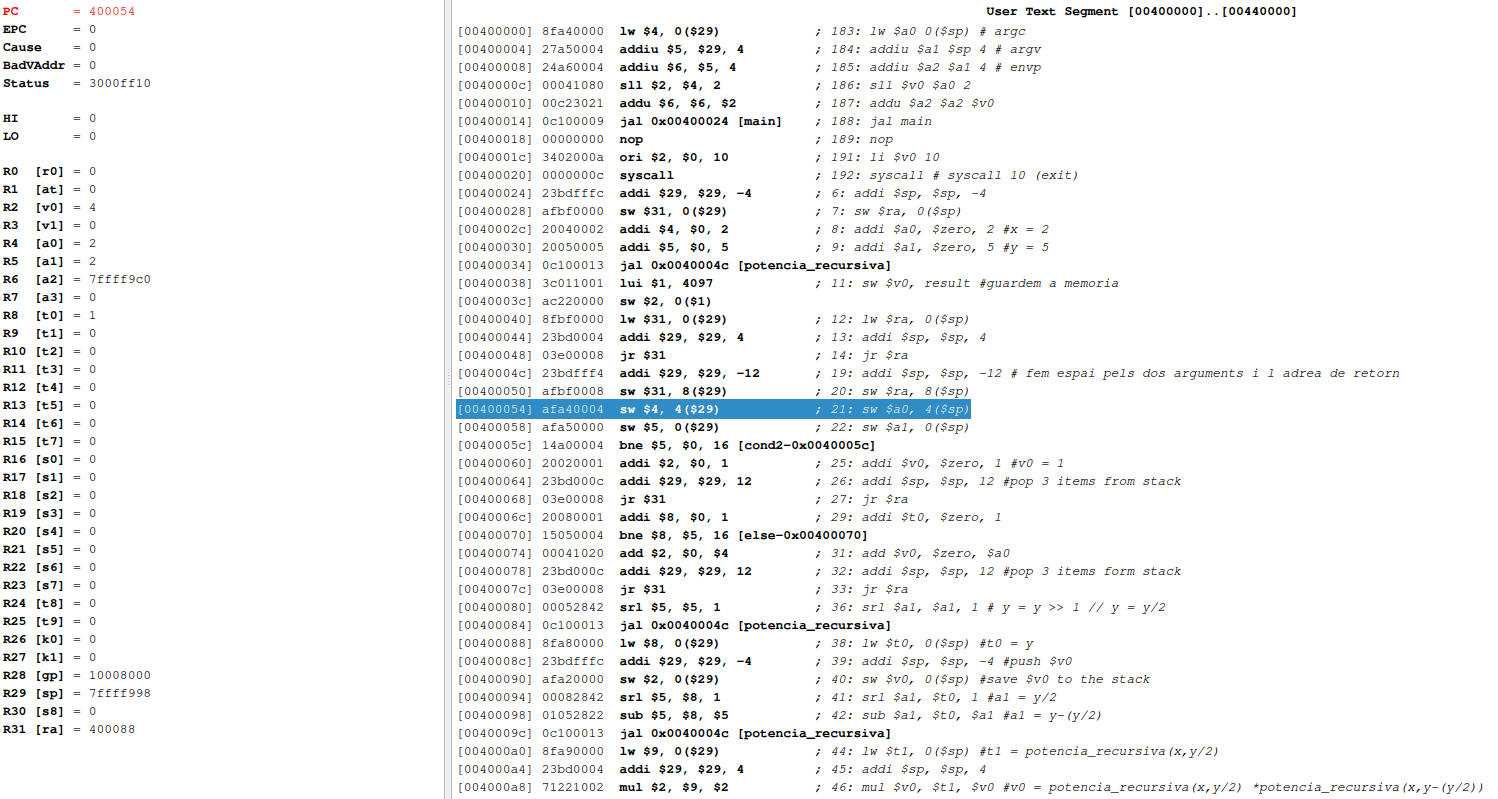
mul $v0, $t1, $v0 #v0=potencia\_recursiva(x,y/2)\*potencia\_recursiva(x,y-(y/2))

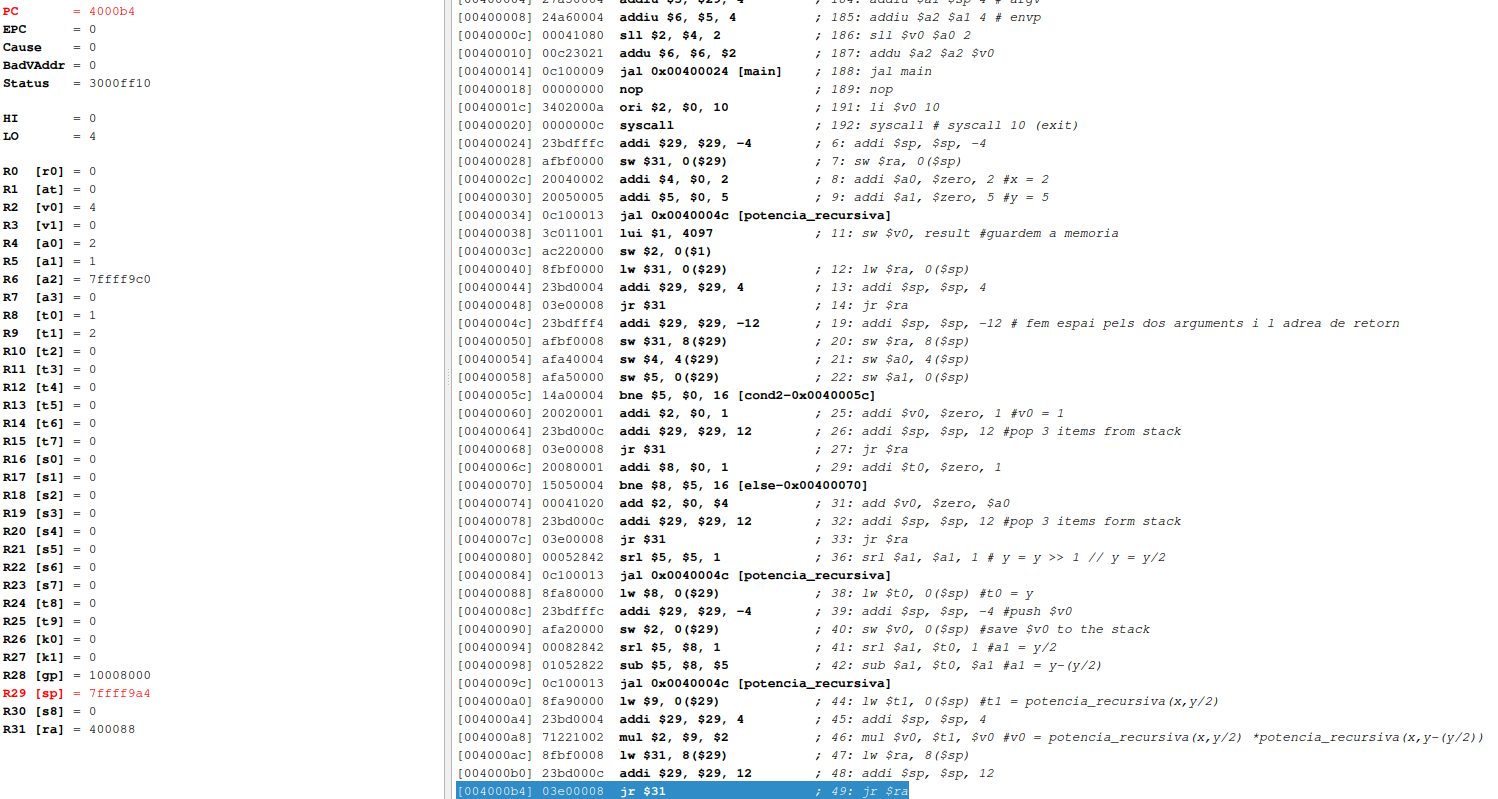
lw $ra, 8($sp)

addi $sp, $sp, 12

jr $ra







**b) Si el programa main llama a la subrutina potencia\_recursiva con unos**

**valores iniciales de x=2 e y=5, como está en el enunciado en C, describe**

**la ejecución dinámica del programa, ayúdate simulando paso a paso en**

**el simulador. Avisa al profesor cuando funcione.**

Primer ens guardem la direcció de retorn del main, perquè la necesitarem després de cridar a la funció de potencia\_recursiva. Després fiquem els valors de x e y a $a0 i $a1 respectivament perquè els necessitem per a la funció potencia\_recursiva.

Un cop dins de potencia\_recursiva, primer fem espai per guardar-nos els dos arguments i la direcció de retorn de la funció, perquè els necessitarem i en ser una funció recursiva en tornar-la a cridar podem perdre aquests valors.

En aquest cas com la y = 5, llavors saltem directament al else que divideix la y entre 2 i tornem a cridar a la funció. Això ho fa dos cops i llavors accedeix quan la y == 1. Sumem el valor de la x a $v0 que serà on retornarem el resultat final de la funció.

Llavors tornem a agafar de la pila el valor inicial de y i afegim el resultat de la primera crida recursiva a la pila (potencia\_recursiva(x,y/2)), perquè ho necessitarem després per calcular el resultat final.

Més tard, preparem les noves variables per a la crida de la funció, com que la y == 1 fem la suma del valor de la x a $v0 que serà on retornarem el resultat final de la funció.

Finalment, recuperem el resultat final de la primera crida recursiva de la pila i el multipliquem pel resultat aconseguit en la segona crida recursiva. Després d'això es faran totes les crides recursives que falten de fer-se.

Al final tornem a agafar la direcció de retorn que es troba a la pila i tornem al main. Guardem el resultat a la variable global resultat i tornem a agafar l'adreça de memòria del main de la pila.

