

## PRINCIPIO DE COMPUTADORES. PRÁCTICA 2

### Descripción.

- 1) Crea un fichero de texto (usando un editor de texto plano como Notepad o cualquiera que tengas a tu disposición) y teclea el código que a continuación te proponemos. Guárdalo en disco, y cárgalo desde QtSPIM.

```
1  # PRACTICA 2 DE PRINCIPIOS DE COMPUTADORES. OPERACIONES ARITMETICAS Y ENTRADA SALIDA
2
3      .data
4      titulo: .asciiz    "Practica 2. Principios de Computadores\n"
5      petop1: .asciiz    "Introduzca primer operando: "
6      petop2: .asciiz    "Introduzca segundo operando: "
7      cadsum: .asciiz    "\nEl resultado de la suma es: "
8
9      .text
10
11  main:
12
13      # las siguientes lineas sirven para imprimir una cadena por la consola
14      li $v0,4 # el registro $v0 debe tener el valor 4 para imprimir cadenas
15      la $a0, titulo # el registro $a0 debe tener la direccion de la cadena a imprimir
16      syscall
17
18      # las siguientes lineas imprimen la cadena de peticion del operando 1 por consola
19      li $v0,4
20      la $a0, petop1
21      syscall
22      # las siguientes lineas leen un entero por consola y dejan el entero leído en $v0
23      li $v0,5 # $v0 debe tener un 5 para que se lea un entero
24      syscall
25      move $t0,$v0 # el entero leído lo muevo a $t0 para posteriormente operarlo
26
27      # repetimos lo mismo pero para el segundo operando
28      # las siguientes lineas imprimen la cadena de peticion del operando 1 por consola
29      li $v0,4
30      la $a0, petop2
31      syscall
32      # las siguientes lineas leen un entero por consola y dejan el entero leído en $v0
33      li $v0,5 # $v0 debe tener un 5 para que se lea un entero
34      syscall
35      move $t1,$v0 # el entero leído lo muevo a $t1 para posteriormente operarlo
36
37      # calculo la suma de ambos enteros y almaceno el resultado en $t2
38      add $t2,$t0,$t1 # $t2 = $t0 + $t1
39
40      # imprimo el resultado por la consola
41      # primero imprimo una cadena que sirva de titulo
42      li $v0,4
43      la $a0,cadsum
44      syscall
45      # las siguiente lineas
46      li $v0,1 # $v0 debe tener un 1 para que se escriba un entero
47      move $a0,$t2 # $a0 debe tener el entero que tiene que imprimir
48      syscall
49
50
51      # las siguiente lineas terminan el programa.
52      li $v0,10
53      syscall
```

Juega con el QtSPIM! Ejecuta paso a paso, examina como van variando los registros. Prueba a ver el resultado en diferentes formatos (hexadecimal, entero, etc).

2) Amplía tu programa para que calcule igualmente la resta, la multiplicación, la división y el resto de los dos mismos operandos. Debes tener en cuenta que la multiplicación y la división utilizan los registros especiales Lo y Hi. Utiliza tus apuntes de clase para poder resolver el problema. En el siguiente enlace <https://campusvirtual.ull.es/1718/mod/resource/view.php?id=152934> (Programación MIPS) tienes las transparencias sobre MIPS. En concreto en la transparencia 16 contiene cómo realizar operaciones aritméticas:

- El tamaño del operando obviamente es de 4 bytes.
- **Suma:**
  - `add $t0,$t1,$t2`      #  $\$t0 = \$t1 + \$t2$  suman enteros con signo.  
# Negativos representados en complemento a 2.
  - `addi $t0,$t3, 7`      #  $\$t0 = \$t3 + 7$  (o cualquier valor inmediato)
  - `addu $t1,$t6,$t7`      #  $\$t1 = \$t6 + \$t7$  suman como enteros sin signo
- **Resta:**
  - `sub $t2,$t3,$t4`      #  $\$t2 = \$t3 - \$t4$  resta de enteros
  - `subu $t1,$t6,$t7`      #  $\$t1 = \$t6 - \$t7$  resta como enteros sin signo
- **Multiplicación:**
  - `mult $t3,$t4`      # La multiplicación de dos registros de 32 bits da  
# un registro de 64 bits que será ((Hi,Lo))
- **División:**
  - `div $t5,$t6`      #  $Lo = \$t5 / \$t6$  (división entera)  
#  $Hi = \$t5 \bmod \$t6$  (Hi tendrá el resto)
- **Asignación de valores de resultado:**
  - `move $t2,$t3`      #  $\$t2 = \$t3$
  - `mfhi $t0`      #  $\$t0 = Hi$  para los resultados de mult y divis.
  - `mflo $t1`      #  $\$t1 = Lo$  para los resultados de mult y divis.

16

**NOTA:** Ten en cuenta que la multiplicación puede dar un número que no quepa en 32 bits. En este caso, el registro HI tendría la parte alta del resultado. Por el momento, para esta práctica ignora esta circunstancia y muestra únicamente los 32 bits menos significativos (que están en el registro LO).

- 3) Por último mejora tu programa, añadiendo estructuras de control para realizar la siguiente funcionalidad:
- a) Si el segundo operando leído por consola es un cero, no debe realizarse la operación de división, y en su lugar deberá escribirse un mensaje por consola anunciando que se trataría de una división por cero.
  - b) Al terminar de imprimir el resultado de las operaciones dale al usuario la opción de comenzar nuevamente. Para ellos debes escribir por la consola el siguiente mensaje:  
"Introduzca un 0 para salir, y cualquier otro número para volver a calcular: "  
El programa deberá terminar si se ha introducido un cero, y volver a comenzar en cualquier otro caso.