

# Sesión 1: Lenguaje ensamblador MIPS y Simula3MS

PARTE I: El objetivo de esta parte es repasar las instrucciones en punto fijo. Carga y ejecuta (usando la configuración monociclo) el siguiente código:

```
.text
.globl main

main:
    la $t0, array1
    la $t2, array2
    la $t1, count
    lw $t1, 0($t1)

Loop:
    lw $t3, 0($t0)
    lw $t4, 0($t2)
    add $t3, $t3, $t4
    sw $t3, 0($t0)
    addi $t0, $t0, 4
    addi $t2, $t2, 4
    addi $t1, $t1, -1
    bne $t1, $0, Loop

fin:
    addi $v0,$0,10      # la llamada para salir del programa
    syscall

# segmento de datos
.data
array1: .word 3, 4, 9, 8, 5, 1
array2: .word 3, 4, 9, 8, 5, 1
count: .word 6

# fin
```

Analiza qué está haciendo el código y contesta a la primera batería de preguntas del cuestionario habilitado en la página de la asignatura en el Moodle de la UDC (<https://udconline.udc.gal>).

## PARTE II: Instrucciones de punto flotante en MIPS

MIPS soporta los formatos IEEE 754 de precisión simple y doble con las siguientes instrucciones:

- Suma en punto flotante, precisión simple (*add.s*) y doble (*add.d*)
- Resta en punto flotante, precisión simple (*sub.s*) y doble (*sub.d*)
- Multiplicación en punto flotante, precisión simple (*mul.s*) y doble (*mul.d*)
- División en punto flotante, precisión simple (*div.s*) y doble (*div.d*)
- Comparación en punto flotante, precisión simple (*c.x.s*) y doble (*c.x.d*), donde *x* puede ser igual (*eq*), diferente (*neq*), menor (*lt*), menor o igual (*le*), mayor (*gt*), mayor o igual (*ge*)
- Salto condicional en punto flotante, verdadero (*bc1t*) o falso (*bc1f*)

Los diseñadores de MIPS decidieron añadir 32 registros de 32 bits en punto flotante separados, llamados \$f0, \$f1, \$f2, ..., usados tanto para precisión simple como para precisión doble. Sólo 16 de los 32 registros en punto flotante de MIPS se pueden usar para operaciones en precisión doble: \$f0, \$f2, ... \$f30. La precisión doble se calcula usando pares de estos registros de precisión simple. Los registros impares se usan solo para la mitad derecha de los números en punto flotante de 64 bits. Es decir, un registro en precisión doble es en realidad una pareja de registros de precisión simple par-impar, que utiliza el registro par como nombre.

La comparación en punto flotante actualiza un bit a cierto o verdadero del registro *Status*, dependiendo de la condición de comparación, y el salto condicional en punto flotante decide si salta o no, dependiendo de la condición.

```
c.lt.s $f2, $f4    # if($f2 < $f4) cond = 1; else cond = 0;
bc1t Etiqueta     # salto relativo a Etiqueta si cond
```

Existen instrucciones de carga y almacenamiento separadas para los registros de punto flotante *lwc1* y *swc1*. Los registros base para las transferencias de datos en punto flotante siguen siendo los registros enteros. El código MIPS para cargar dos números en precisión simple desde memoria, y almacenar el resultado de la suma podría ser de la siguiente manera:

```
lwc1  $f2, 0($t0)   # carga en $f2 el número que está en la dirección 0+$t0
lwc1  $f4, 4($t1)   # carga en $f4 el número que está en la dirección 4+$t1
add.s $f6, $f2, $f4 # f6 = f2 + f4 en simple precisión
swc1  $f6, c($t2)   # Almacena f6 en la posición c+$t2
```

Para cargar o almacenar un número en doble precisión es necesario realizar dos operaciones de carga *lwc1* o almacenamiento *swc1*, cada una para uno de los registros par-impar que contiene el dato.

Carga y ejecuta (usando la configuración monociclo) el siguiente código:

```
.data

Num: .word 4
vector: .float 5, 4, 3, 7, 6

.text
.globl main

main:

    la $s4, Num
    la $s5, vector
    lw $s0, 0($s4)
    lwc1 $f12, 0($s5)
    addi $s1, $0, 1
    addi $s5, $s5, 4

Loop:
    slt $t1, $s1, $s0
    beq $t1, $0, fin
    lwc1 $f0, 0($s5)
    c.le.s $f0, $f12
    bclt mayor
    mov.s $f12, $f0

mayor:
    addi $s1, $s1, 1
    addi $s5, $s5, 4
    j Loop

fin:
    swc1 $f12, 0($s5)
    addi $v0, $0, 2
    syscall

    addi $v0, $0, 10
    syscall
```

Analiza qué está haciendo el código y contesta a la segunda batería de preguntas del cuestionario habilitado en la página de la asignatura en el Moodle de la UDC (<https://udconline.udc.gal>).