

UNIVERSIDAD CARLOS III DE MADRID
DEPARTAMENTO DE INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA. ESTRUCTURA DE COMPUTADORES

Para la realización del presente examen se dispondrá de **1:30 horas**.
NO se podrán utilizar libros, apuntes **ni** calculadoras de ningún tipo.

Ejercicio 1. Considere el siguiente fragmento en ensamblador:

```
.data
    A1: .word 5, 8, 7, 9, 2, 4, 5, 9
    A2: .word 1, 4, 3, -8, 5, 6, 5, 9
    .align 2
    A3: .space 32

.text
```

Se pide:

- ¿Qué representa A1? ¿Cuántos bytes ocupa la estructura de datos A1 (justifique su respuesta)?
- Se desea implementar una función cuyo prototipo en un lenguaje de alto nivel es la siguiente:

```
void Mezclar(int a[], int b[], int c[], int N)
```

Esta función recibe 4 parámetros, los tres primeros son vectores de números enteros y el cuarto indica el número de componentes de cada uno de estos vectores. La función se encarga de almacenar en cada componente i de c , el siguiente valor: $c[i] = \max(a[i], b[i])$.

Escriba, utilizando el ensamblador del MIPS32, el código correspondiente a esta función. Utilice para ello la convención de paso de parámetros que se ha descrito a lo largo del curso.

- Escriba el código necesario para llamar a la función desarrollada en el apartado anterior, para los vectores A1, A2 y A3 definidos en la sección de datos anterior. Asuma que A3 es el vector donde se deben dejar los elementos máximos.

Solución del ejercicio 1

- A1 representa un vector de ocho números enteros
Si son 8 enteros, y siendo una máquina de 32 bits cada entero se representa 4 bytes, supone un total de $8 \cdot 4 = 32$ bytes.
- Y c)

```
.data

v1: .word 1, 2, 3, 4, 5
v2: .word 5, 4, 3, 2, 1
v3: .word 0, 0, 0, 0, 0
```

```

.text
.globl main

maximo:
    move $t0 $a0
    li $t4 0

    move $t1 $a1
    move $t2 $a2
    move $t3 $a3

bucle1: bge $t4 $t0 fin1
    lw $t5 ($t1)
    lw $t6 ($t2)

    bgt $t5 $t6 es5
    sw $t6 ($t3)
    b next1
es5: sw $t5 ($t3)

next1: add $t1 $t1 4
    add $t2 $t2 4
    add $t3 $t3 4
    add $t4 $t4 1

    b bucle1

fin1: jr $ra

main:
    li $a0 5

    la $a1 v1
    la $a2 v2
    la $a3 v3

    jal maximo

    li $t0 0
bucle2: bge $t0 5 fin2

    lw $a0 ($a3)
    li $v0 1
    syscall

    add $a3 $a3 4
    add $t0 $t0 1
    b bucle2

fin2: li $v0 10
    syscall

```

Ejercicio 2. ¿Cuál es el número positivo normalizado más pequeño que se puede representar utilizando el estándar de simple precisión IEEE 754. Justifique su respuesta. Indique también el número positivo no normalizado más pequeño que se puede representar. Justifique de igual forma su respuesta.

Solución del ejercicio 2

a) El número positivo normalizado más pequeño representable en el estándar IEEE 754 (32 bits) es:

0 0000001 000000000000000000000000
positivo normalizado más pequeño

Cuyo valor es $1.0 * 2^{-127} = 2^{-126}$

b) El número positivo no normalizado más pequeño representable en el estándar es:

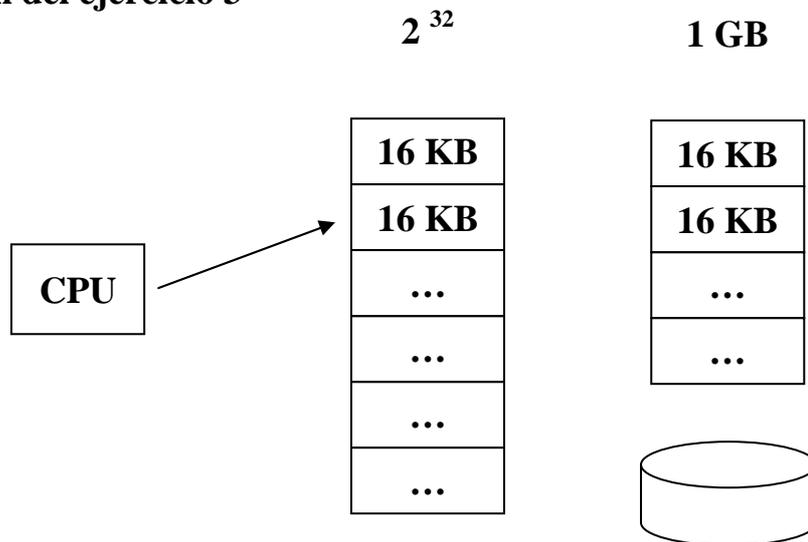
0 0000000 000000000000000000000001
positivo no normalizado más pequeño

Cuyo valor es $2^{-23} * 2^{-126} = 2^{-149}$

Ejercicio 3. Considere un computador de 32 bits que dispone de un sistema de memoria virtual que emplea páginas de 16 KB y tiene instalada una memoria principal de 1 GB. Indique de forma razonada:

- a) El formato de la dirección virtual.
- b) El número máximo de páginas en este computador.
- c) El número de marcos de página de este computador.
- d) El tamaño del bloque que se transfiere entre disco y memoria principal cuando ocurre un fallo de página
- e) El elemento del computador que genera el fallo de página y quién lo trata.

Solución del ejercicio 3



a)

Id. página	Desplazamiento
$32 - 14$	$2^{14} = 16 \text{ KB}$
18 bits	14 bits

- b) $2^{32} / 2^{14} = 2^{32-14} = 2^{18}$ **marcos**
- c) $2^{30} / 2^{14} = 2^{30-14} = 2^{16}$ **páginas**
- d) El tamaño de una página es **16 KB**
- e) La **MMU genera la excepción** y **la rutina de fallo de página** del sistema operativo **trata la excepción**

Ejercicio 4. ¿Qué es una interrupción y que sucede cuando se genera?

Solución del ejercicio 4

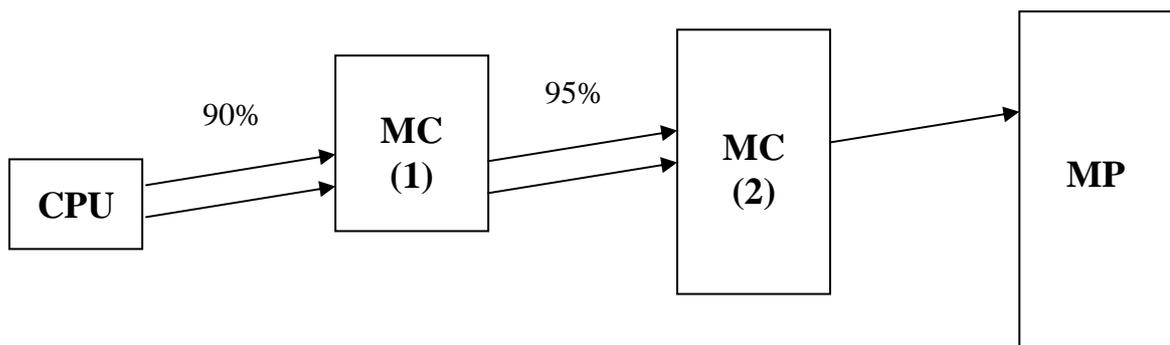
Una interrupción es una excepción asíncrona.

Se guarda el estado y se ejecuta la rutina de interrupción asociada.

Ejercicio 5. Se dispone de un sistema con una memoria caché de 2 niveles. En la ejecución de una determinada aplicación, la tasa de aciertos de la caché de nivel 1 es del 90% y la tasa de aciertos de la caché de nivel 2 es del 95%. La aplicación genera durante su ejecución un millón de accesos a memoria. Indique de forma razonada:

- a) El número de accesos que se genera a la caché de nivel 1.
- b) El número de accesos que se genera a la caché de nivel 2.
- c) El número de accesos que se genera a memoria principal.

Solución del ejercicio 5



- a) El número de accesos con acierto en MC(1) es: 10^6
- b) El número de accesos con acierto en MC(2) es: $10^6 * 0,1$
- c) El número de accesos con acceso a MP por fallos de MC1 y MC2: $10^6 * 0,1 * 0,05$

