

# PRÁCTICA 7: JERARQUÍA DE MEMORIA III

## ARQUITECTURA DE COMPUTADORES. 2º CURSO

### 1. OBJETIVOS

En esta práctica realizaremos un estudio de la influencia de los parámetros de una memoria caché sobre el rendimiento. En primer lugar, ubicaremos los datos en memoria y obtendremos la traza de acceso partiendo de un código en C o ensamblador, como ya se hizo en prácticas anteriores. Posteriormente trataremos de extraer el comportamiento de la caché en función de sus características y de la traza de accesos que realiza. Finalmente, pondremos de manifiesto cómo influyen los parámetros que definen la caché en la tasa de fallos.

### 2. PREPARACIÓN

Antes de comenzar con esta sesión de laboratorio el alumno debería:

- 1) Leer y asimilar los contenidos teóricos de la memoria caché (políticas de ubicación/localización).

### 3. DURANTE LA SESIÓN

#### 3.1. Cálculo a mano para accesos a memoria

En la sesión online anterior, se describieron dos métodos para obtener los accesos a memoria caché: Interpretando la dirección y calculándola de forma analítica.

En esta ocasión vamos a focalizar la atención en un caso particular del método basado en la interpretación de las direcciones, que podremos aplicar en la práctica en casos puntuales: *Interpretación rápida de direcciones en hexadecimal.*

##### 3.1.1. Caso particular del mecanismo 1: Interpretación rápida de la dirección en hexadecimal.

Este mecanismo simplifica notablemente la interpretación de la dirección, evitando tener que realizar la transformación a binario, pero sólo puede ser utilizado en casos muy concretos.

Puesto que un dígito hexadecimal requiere exactamente 4 bits para su codificación, podemos obtener rápidamente el campo desplazamiento e índice de la dirección de un acceso a memoria (expresado en hexadecimal) si cada uno de estos dos campos tiene un tamaño múltiplo de 4.

En caso de cumplirse este requisito, para obtener el valor del índice y desplazamiento, podremos evitar la transformación previa a binario y nos bastará con tomar tantos dígitos hexadecimales de la dirección como múltiplos de 4 contenga el campo. Por supuesto, habrá que empezar por el campo desplazamiento por encontrarse más a la derecha. En este caso, utilizaremos la base decimal para representar cualquier parámetro de la caché.

**Ejemplo:** Sea una caché de 4KB de mapeado directo, política CB-WA y con un tamaño de bloque de 16 bytes.

El tamaño del campo desplazamiento será de 4 bits ( $16\text{Bytes} = 2^4 \text{ Bytes} \rightarrow 4 \text{ bits de desplazamiento}$ ) y el tamaño del campo índice será de:  $N^\circ \text{Líneas} = T_{MC} / T_{LÍNEA} = 4\text{KB} / 16\text{B} = 256 = 2^8 \text{ líneas} \rightarrow 8 \text{ bits de índice}$ .

Si accediésemos a la dirección (de 32 bits) de memoria 0x12345678, se obtendría la siguiente descomposición y acceso a memoria caché:

Dir. Memoria	0x12345678		Memoria Caché		
Interpretación de MP	Dir./Num. Bloque (28)	Desplaz (4)	V	Etiqu.	Datos
	0x1234567	0x8	L 0x00	0	
				0	
Interpretación de MC	Etiqu.(20)	Índice (8)	Desplaz (4)	L 0x67	1
	0x12345	0x67	0x8	0	0x12345 Bloque 0x1234567
				L 0xFF	0