

# Tema-5-Unidad-de-Memoria.pdf



**CodeWolf**



**Arquitectura de Computadores**



**2º Grado en Ingeniería Informática**



**Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba**



[Accede al documento original](#)



Escuela de  
Organización  
Industrial

Contigo que evolucionas.  
Contigo que lideras. Contigo que transformas.

**Esto es EOI.  
Mismo propósito,  
nueva energía.**



Descubre más aquí



**EOI** Escuela de  
Organización  
Industrial

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato  
→ Planes pro: más coins

pierdo espacio



Necesito concentración

ali ali ooh  
esto con 1 coin me  
lo quito yo...

WUOLAH

## Tema 5 Unidad de Memoria

### ➤ Visión Global :

La estructura de Von Neumann está formada por :

→ Unidad de Cálculo

→ Unidad de Control

→ Unidad de Memoria

→ Unidad de E/S

➤ Unidad de Memoria

→ Características de la Unidad de Memoria:

#### ◆ Almacenamiento:

- Programas y Datos del Usuario
- Programas y Datos del Sistema.

→ Parámetros a tener en cuenta:

◆ Capacidad

◆ Velocidad de acceso

◆ Costo por byte.

#### ◆ Condiciones óptimas

- Gran capacidad (Muchos gigabytes)
- Muy rápidas (mínimo nanosegundos)
- Baratas

➤ Primera Clasificación

→ Memoria principal

◆ Acceso directo por CPU.

◆ Almacena los programas que se están ejecutando actualmente.

◆ De tipo semiconductora:

- De lectura/escritura (RAM) o sólo lectura (ROM)
- Estáticas (SRAM) o dinámicas (DRAM)

◆ Capacidad del orden de los GB.

◆ Tiempos de acceso del orden de las decenas de nanosegundos.

◆ Costo alto y proporcional a la velocidad de acceso.

→ Memoria Auxiliar o Secundaria.

◆ Acceso a través de Interfaces.

◆ Almacenamiento de programas y datos no ejecutándose.

◆ De tipo Magnético u Óptico.

◆ Capacidad del orden de los GB/TB.

◆ Tiempo de acceso entre milisegundos y segundos.

◆ Baratas y proporcional a la capacidad.

➤ Jerarquía de memoria

→ Memoria Asociativa

Es un tipo de memoria al cual accedes por contenido, en caso de tenerlo, proporciona una serie de datos relacionados al contenido buscado. Son muy lentas, debido a que va buscando y comparando por línea de memoria, se proporciona como solución una búsqueda en paralelo, comparando todos los contenidos al mismo tiempo, sin embargo, son muy costosas.

WUOLAH

→ **Búsquedas rápidas => Circuitería compleja**

- ◆ Incluir comparaciones
- ◆ Incluir estado de las posiciones, etc.

→ **Componentes de una Memoria Asociativa**

- ◆ Matriz de memoria que incluye:
  - Celdas de almacenamiento ( $m \times n$ )
  - Circuitería de comparación
  - Registro de Argumento (A) : Contiene el dato o instrucción que se busca ( $n$ )
  - Registro clave (K) : Para enmascarar parte del argumento de búsqueda ( $m$ )
  - Registro Equiparador (M): Indica las posiciones que tienen ese contenido ( $m$ )

→ **Lógica de lectura:**

Se busca la información cuya etiqueta coincida con la que se presenta.

- Testeo de bits registro  $M(m)$ :
- Si todos los bits a 0 el contenido no está en memoria asociativa.
- Si algún  $M_i = 1$  Ese contenido está en alguna celda de memoria.

→ **Lógica de escritura:**

Introducción de un dato en una posición que acelere la búsqueda.

Permite el direccionamiento secuencial.

→ **Memoria Caché**

Permite almacenar la información más utilizada (instrucciones y datos).

Es un tipo de memoria muy rápida (casi tanto como la CPU), de tamaño reducido y muy costosa. **Se accede directamente desde CPU y memoria Principal.**

**Su funcionamiento se basa en:**

- **CPU** solicita una dirección → Primero busca en la memoria caché.
- **Si la encuentra en caché** → Toma el contenido y continúa operando.
- **Si no está en caché** → Accede a la memoria principal y toma el dato, a su vez lleva de memoria principal a caché el bloque de la información accedida.

**Tasa de aciertos:** Mide la cantidad media de solicitudes de direcciones con acierto frente al número total, por tanto:

- Si la dirección a buscar está en memoria caché Acierto.
- Si la dirección a buscar NO está en memoria caché Fallo.
- ★ **Localidad de referencia:** Permite dada una búsqueda, si no encuentra el dato en caché, lo copiará desde memoria principal a memoria caché, por si en un futuro se vuelve a solicitar el dato requerido, acelerando en un futuro el procesamiento.

◆ **Existen 3 tipos de memoria caché:**

- **Asociativo:** Más rápida y más cara de implementar. Cada posición de la memoria almacena: Dirección y Datos.
- **Mapeo directo:** Memoria de acceso aleatorio pequeña que se comporta como la memoria principal, sin embargo, su tasa de aciertos es muy baja.

La dirección se divide en: Índice y Rótulo. Cada posición de la memoria caché almacena: Campo de rótulo y campo de datos.

→ **Mapeo Asociativo por vías:** Conjunto formado por asociativa y mapeo directo.

Agrupación de palabras con su rótulo asociados a un mismo bloque de memoria caché.

Para realizar la búsqueda, usa un campo de índice de la dirección física, determina la posición en caché, y mediante asociativa se determina si el rótulo está en el bloque.

- Si está, devuelve el dato
- Si no está, se lanza un algoritmo de reemplazo.

#### ❖ **Escritura en Memoria Caché**

Para escribir en caché, es necesario que sea copiado en memoria principal, para evitar una falta de coherencia entre la caché y memoria física. Para garantizar esta coherencia, se definen:

- **Escritura Directa:** Se escribe en paralelo en caché y en principal, sin embargo, la principal siempre tiene los datos actualizados, haciendo más lenta a la caché.
- **Escritura Diferida:** Se escribe solo en caché y se marca un bit de modificación, al intentar eliminar la palabra, se mira el bit:
  - ◆ Si está activo, se copiará esa palabra a Memoria Principal
  - ◆ Si no, eliminar el dato.

#### ❖ **Validación de datos en caché:**

Para saber si una posición de la memoria caché está vacía, incluimos un bit de validación que nos indicará si el dato de dicha posición que se encuentra en memoria es basura (0) o es un dato válido (1), permitiendo así eliminar grandes cantidades de posiciones de caché.

#### → **Memoria Virtual**

Permite la ejecución de programas de gran tamaño, almacenando dichos programas en memoria secundaria, y parte de las tareas en memoria física. La CPU solicita datos o instrucciones usando direcciones virtuales, para ello primero ha de convertirlas en direcciones físicas. Si el dato no está en MP:

- 1.Lo trae desde memoria Secundaria
- 2.Da fallo de acceso, y si el dato no cabe, habrá que eliminar registros.

#### **Definiciones:**

- **Espacio de Direcciones:** Conjunto de direcciones virtuales que genera el programa.
- **Espacio de Memoria:** Conjunto de direcciones físicas de memoria.

Para traducir las direcciones virtuales a físicas surge la tabla de **mapeo de memoria**, ésta se puede encontrar almacenada en: **memoria separada** (tiempo de acceso extra), en **memoria principal** (muy lenta), o en (uso de memoria rápida para la traducción).

Se realiza la división del espacio de direcciones y memoria en dos partes:

- **Página:** Para el espacio de direcciones.
- **Bloque:** Para el espacio de memoria.

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato  
→ Planes pro: más coins

perdo  
espacio



Necesito  
concentración

ali ali ooh  
esto con 1 coin me  
lo quito yo...

WUOLAH

❖ **Inconvenientes:**

- **Tamaño:** Se necesitan tantas posiciones como páginas.
- **Acceso:** Dará fallo si la página no está en MP, necesitamos acceder a una memoria secundaria, así como un algoritmo de reemplazo (saber que página quitar de memoria).

➤ **Administrador de memoria**

Es el encargado de:

- Decidir que página retirar cuando se produce un fallo de página.
- Decidir cuando se transfiere una página de Memoria secundaria a MP.
- En que bloque de MP colocar una página.

Un fallo de página puede darse cuando la CPU hace referencia a una página que se encuentra en memoria secundaria, en este caso las acciones a realizar pueden ser: suspender la ejecución del proceso.

**Transferencia de MS a MP:**

Si la memoria principal está llena, se ha de retirar una página aplicando un algoritmo de reemplazo, sin embargo, si la página ha cambiado ha de copiarse a memoria secundaria previamente y actualizar la tabla de página.

➤ **Algoritmos de Reemplazo**

- **FIFO:** Elimina la página que entró primero (Primero que entra, primero que sale). Es muy malo cuando tenemos procesos cortos que van tras un proceso muy largo.
- **LRU:** Elimina la página menos usada recientemente. Cada ciclo aumentaremos el contador individual de cada página, de modo que se resetea cada vez que se menciona a dicha página, eliminando la página cuyo contador es más alto (es la que menos se ha usado).
- **LFU:** Elimina la página menos usada frecuentemente. Se asocia un contador individual a cada página, de modo que lo incrementaremos cada vez que se haga referencia a dicha página, eliminando la página que tiene el contador más bajo.

El algoritmo LFU no es apto para aplicarlo como algoritmo principal, sin embargo si es muy buena opción añadirlo como algoritmo secundario usando de principal el LRU (PE) y en caso de duda entre dos procesos utilizar LFU.

➤ **Memoria Caché Vs Memoria Virtual**

❖ **Memoria Caché**

- Contiene las Instrucciones y Datos utilizados más frecuentemente
- Acceso directo por la CPU
- Tiempo de acceso de 5 a 10 veces más reducido
- Tamaño de bloque pequeño (entre 4 y 16 palabras)
- Administración Hardware

❖ **Memoria Virtual**

- Contiene (en Mem. Secundaria) las partes de código y de datos de un programa no usados en ese momento
- Acceso a través de un proceso de E/S
- Tiempo de acceso de 5 a 10 veces más grande
- Tamaño de bloque grande (entre 64 y 4K palabras)
- Administración es parte Hardware y parte Software

WUOLAH



### ➤ Segmentación (Pregunta examen, que es la segmentación)

Es una técnica de gestión de memoria, consistente en la división variable de un programa en N segmentos, asociando la longitud de cada segmento a un número variable de páginas que conformarán una dirección lógica, que puede ser mayor, igual o menor que la dirección física. Encontramos 3 campos: **Campo de Segmento, Campo de Página y Campo de Palabra**. Su funcionamiento de traducción se basa en:

Para realizar la traducción será necesario utilizar dos memorias adicionales, que en memoria aleatoria implica un tiempo de acceso muy lento, pero en asociativa determinaremos que par de (Segmento y Página) encontramos en la tabla, devolviendo el valor del Bloque (si está presente) y si no lanzará un algoritmo de reemplazo.

#### ❖ Translation Lookaside Buffer (TLB): (Pregunta Examen, que es la TLB)

Es una memoria caché administrada por la Unidad de gestión de memoria, que contiene partes de la tabla de paginación, es decir, relaciones entre direcciones lógicas y físicas.

Se utiliza para obtener la traducción rápida de direcciones, ya que la CPU solicitará una página, primeramente, se buscará en la TLB, al no encontrarla la solicitará a la tabla de paginación, que, al obtener la página y bloque, la devolverá a CPU (insertándose previamente en la TLB).

De esta forma cuando la CPU vuelva a solicitar dicha página, bastará con revisar la TLB. De esta forma se acelera en muchos ciclos la traducción de direcciones lógica a física.

### ➤ Protección de Memoria

Es necesario establecer una protección para evitar accesos no autorizados a ciertas zonas de memoria, para ello podemos establecerla por permisos (lectura, escritura, ejecución). Dicha protección puede ser establecida en:

→ Dirección Física.

→ Dirección Lógica (descriptor de segmentos).

★ **Descriptor de segmento:** Contienen la dirección base y el límite del segmento en el espacio lógico, su estructura es la siguiente:

→ Dirección: Dirección de la tabla de Páginas / Dirección del Bloque Base.

→ Longitud: Tamaño del segmento.

→ Protección: Permisos de acceso dados a dicho segmento.