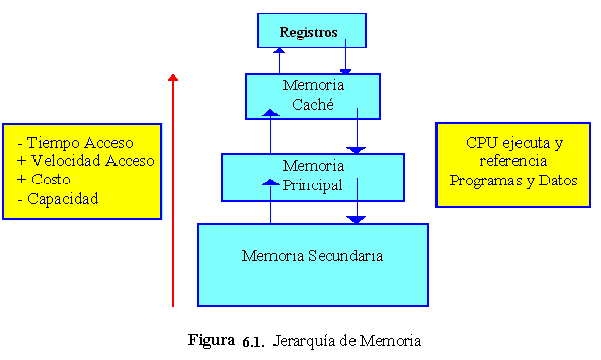
RESUMEN TEMA 5 – JERARQUÍA DE MEMORIA



**Principio de localidad**: Los programas acceden a una porción relativamente pequeña del espacio de direcciones en cualquier instante de tiempo

* **Localidad temporal**: si se referencia un elemento, tenderá a ser referenciado pronto
* **Localidad espacial**: Los elementos cercanos al elemento referenciado tenderán a ser referenciados pronto



La información se ubica en un nivel dependiendo de su probabilidad de uso, debido a que la información varía durante la ejecución de un programa, se produce un **tráfico continuo** de información entre los niveles.

**2 Niveles:**

* **Nivel superior:**  Más cercano a la CPU, memoria rápida, pequeña y cara
* **Nivel Inferior:** Más alejado a la CPU, memoria lenta, grande y barata

**Bloque**: mínima unidad de información que se puede referenciar en la jerarquía de dos niveles. Compuesto por palabras de memoria.

**Acierto (hit)**: acceso con éxito a la memoria del nivel superior, caso contrario **Fallo (miss)**

**Frecuencia o tasa de aciertos (hit ratio):** porcentaje de aciertos en accesos a memoria en el nivel superior.

**Tasa de fallos (miss ratio):** 1 – tasa de aciertos

**Tiempo de acierto (TA):** tiempo necesario para acceder al nivel superior cuando se produce un acierto. 🡺 t. en determinar si la info está en ese nivel + t. en acceder a la info

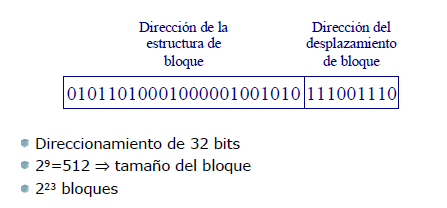
**Penalización de fallo (PF):** tiempo consumido en obtener la información cuando se produce un fallo. 🡺 t. en sustituir un bloque del nivel superior por el adecuado del inferior + t. en proporcionar el bloque al dispositivo que lo ha pedido

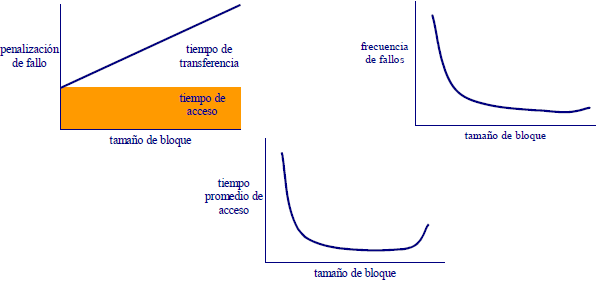
**Tiempo de acceso:** tiempo que se tarda en acceder a la primera palabra del bloque

**Tiempo de transferencia:** tiempo para transferir el resto del bloque. Depende del ancho de banda.

**DIRECCIÓN DE MEMORIA**

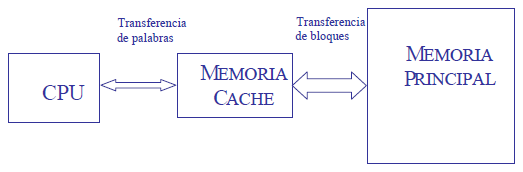
* **Dirección de la estructura del bloque**: identifica al bloque dentro de ese nivel
* **Dirección del desplazamiento del bloque**: identifica el elemento dentro de un bloque.



**Tiempo medio de acceso a memoria (TMA):** tiempo de acierto + tasa de fallos \* penalización de fallo.

**MEMORÍA CACHÉ**

Mucho más rápida que la memoria principal, es de alto coste. Se coloca una memora caché entre la CPU y la memoria principal. La CPU accede más veces a memoria caché que a memoria principal. Esta memoria contiene una copia de aquellas posiciones de memoria principal que están siendo utilizadas por la CPU.



**¿Dónde se ubica un bloque en la caché?**

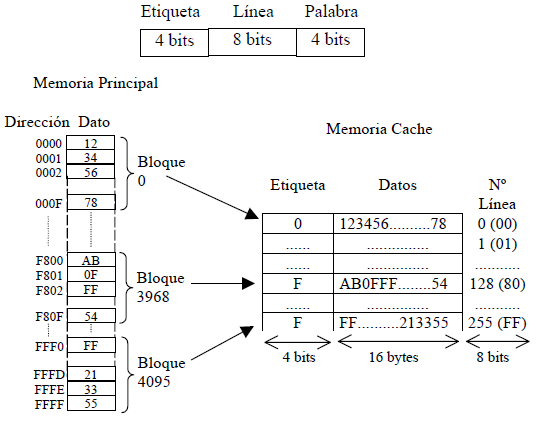
Esto depende del método de ubicación de bloques. Para la ubicación, es necesaria una función de correspondencia que haga corresponder bloques de memoria principal con líneas de memoria caché.

* **Correspondencia directa:** cada bloque sólo puede aparecer en un lugar en la caché.
  + Línea= dir bloque MOD num de línea
* **Correspondencia asociativa por conjuntos**: Un bloque se puede colocar en un conjunto restringido de lugares en la caché.
  + Línea= dir bloque MOD num de conjuntos
* **Correspondencia completamente asociativa**: Un bloque se puede colocar en cualquier parte de la caché

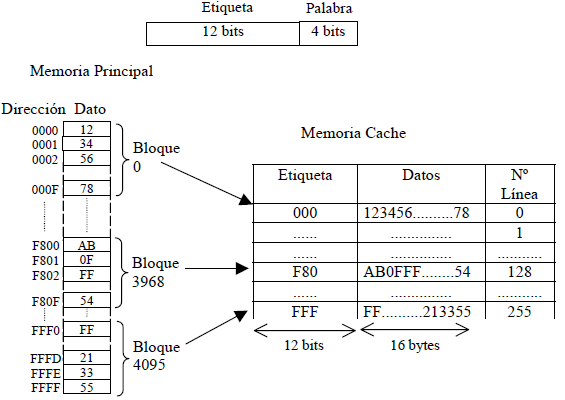
**¿Cómo se encuentra un bloque en la caché?**

Las cachés incluyen una etiqueta de dirección en cada línea que identifica la dirección de la estructura del bloque. Esta etiqueta posee un bit de validez que indica si la línea contiene información válida.

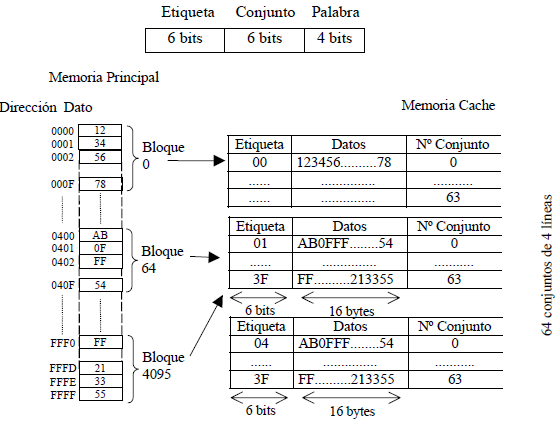
**CORRESPONDENCIA DIRECTA**



**CORRESPONDENCIA ASOCIATIVA**



**CORRESPONDENCIA ASOCIATIVA POR CONJUNTOS**



**¿Qué bloque debe reemplazarse en caso de fallo?**

Cuando se introduce un nuevo bloque en la caché, debe sustituirse uno de los bloques existentes.

En correspondencia directa solo hay una posible línea para cada bloque, en el caso de las técnicas asociativas se requieren **algoritmos de sustitución.**

**Método aleatorio**

Se escoge aleatoriamente uno de los bloques candidatos

**Método LRU (Least recently used)**

Se sustituye el bloque que hace más tiempo que no fue referenciado

**Método FIFO (First In First Out)**

Se reemplaza el bloque que hace más tiempo que está en caché

**Método LFU**

Se elimina el bloque que ha sido menos referenciado

**¿Qué ocurre en una escritura?**

**ESCRITURA DIRECTA**

La información se escribe en la caché y en la MP en paralelo`.

**POSTESCRITURA**

La información se escribe sólo en el bloque de caché. Este bloque sólo se escribe en MP al ser reemplazado. Se utiliza un bit de modificación para saber si ese bloque está limpio o sucio.

**TIPOS DE FALLOS**

Por carga inicial o forzosos (Compulsory) : Carga inicial de la caché

De capacidad (Capacity) : Si la caché no puede almacenar todos los bloques de un programa en ejecución.

Por conflicto (Conflict): Aún habiendo líneas libres no se puede cargar un bloque, sucede en cachés con asociatividad.

**OPTIMIZACIÓN**

**Reducir frecuencia de fallos aumentando el tamaño del bloque:** Reduce fallos por carga inicial, pero aumenta los fallos por capacidad y conflicto. Aumenta la penalización por fallos.

**Reducir frecuencia de fallos aumentando el tamaño de la caché:** Se alarga el tiempo de acierto y aumenta el coste.

**Reducir frecuencia de fallos aumentando la asociatividad:** esto puede aumentar el coste e incrementar el tiempo de acierto.

**Reducir la penalización de fallos con cachés multinivel:** la caché de primer nivel pequeña para ajustarse al tiempo de ciclo de reloj del procesador rápido. La caché de segundo nivel lo suficientemente grande para que haya pocos accesos a la MP.

**Reducir la penalización de fallos dando prioridad a las lecturas sobre las escrituras:** Se realiza la lectura antes de que la escritura se haya completado.

**MEMORIA PRINCIPAL**

La memoria principal satisface las demandas de las cachés y unidades vectoriales y sirve como interfaz de E/S.

Las medidas de rendimiento de la MP:

* Latencia: es la preocupación primordial de la caché (PF)
* Ancho de banda: es la preocupación primordial de la E/S y de las unidades vectoriales.

Más fácil mejorar ancho de banda que reducir la latencia.

**Memoria principal más ancha**

La MP tiene la anchura de una palabra para que coincida con la anchura de la caché. Duplicar o Cuadruplicar el ancho de la memoria, duplicará o cuadruplicará el ancho de banda de memoria.

Inconvenientes:

* Mayor coste
* Dificultad en la corrección de errores.

**Memoria entrelazada**

Los chips de memoria se pueden organizar en bancos para leer o escribir múltiples palabras a la vez. Los bancos pueden leer simultáneamente. Los bancos permiten un ciclo de reloj para cada escritura.

La correspondencia entre direcciones y bancos afecta al comportamiento del sistema de memoria. (Factor de entrelazado). El entrelazado optimiza los accesos secuenciales a la memoria.

Inconvenientes:

* Mayor coste
* Dificultad de expansión de la memoria principal

**Bancos de memoria independientes**

Permitir múltiples accesos independientes, múltiples controladores de memoria permiten a los bancos operar de manera independiente.

Cada banco necesita líneas de dirección separadas y posiblemente un bus de datos separado.

Evitar conflictos al acceder al mismo banco:

* Solución Software: El compilador puede realizar las optimizaciones.
* Solución Hardware: tener número primo de bancos.

**MEMORIA VIRTUAL**

Esta memoria gestiona automáticamente los niveles de la jerarquía de memoria representada por la memoria principal y la secundaria.

Permite:

* En un mismo instante de tiempo los computadores estén corriendo múltiples procesos.
* La existencia de un esquema de protección de la memoria para cada proceso.
* Las necesidades de memoria de todos los programas de un sistema sean superiores a la memoria física disponible.
* Compartir una cantidad de memoria física pequeña entre muchos procesos.
* Simplifica la carga del programa para su ejecución

Divide la memoria física en bloques ( páginas ) y los asigna a diferentes procesos. Cada proceso tiene un espacio virtual propio dividido en páginas. Estas páginas pueden estar en memoria principal o en disco. La CPU genera direcciones virtuales que más tarde son traducidas a direcciones físicas. Cuando una página no está en MP se produce un fallo y debe traerse desde disco. El reemplazo de páginas está controlado principalmente por el sistema operativo.

Dos clases:

* Páginas: bloques de tamaño fijo ( 4096 bytes y 8192 bytes)
* Segmentos: Bloque de tamaño variable (Mayor entre 216 bytes 232 bytes; pequeño 1 byte)

**¿Dónde puede ubicarse un bloque en MP?**

La penalización de fallos es alta ya que acceden a disco. Es importante reducir los fallos de página. Los SO permiten asignación completamente asociativa.

**¿Cómo se encuentra un bloque si está en MP?**

La dirección virtual se descompone en : número de página y desplazamiento de página. La dirección física se obtiene concatenando la dirección física de la página con el desplazamiento de la página. Se utilizan las tablas de páginas que contiene la dirección física de la página y es indexada por la dirección virtual. Esta tabla reside en memoria principal.

**¿Qué bloque debe sustituirse en un fallo de memoria virtual?**

Casi todos los SO intentan sustituir el bloque menos recientemente utilizado. Muchas máquinas proporcionan un bit de uso o referencia para este propósito.

**¿Qué ocurre en una escritura?**

Usar escritura directa es demasiado caro por tanto se usará postescritura. Incluyen un bit de modificación (dirty).

**TRADUCCIÓN RÁPIDA DE DIRECCIONES**

Disponer de una caché para la traducción de direcciones (Translation lookaside buffer (TLB)).