Arquitectura de las Computadoras

Unidad 5 Memoria

Objetivos

- Definir los tipos de memorias, caracterizarlas y establecer sus principales usos
- Entender los conceptos relativos a memoria cache
- Observar tipos y políticas de cache, presentando sus algoritmos que son comunes a otras disciplinas
- Presentar la influencia que tienen dichos conceptos, sobre las arquitecturas y las ejecuciones de los programas
- Introducir el concepto de memoria virtual, administración y posibles mejoras qeu se pueden hacer para elevar sus prestaciones

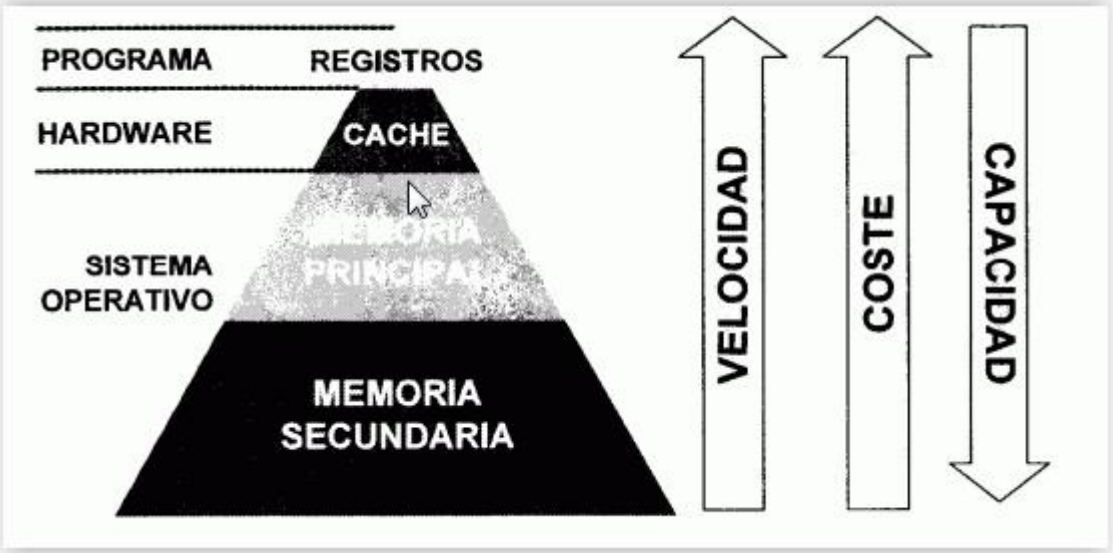
Rendimiento

El sistema de memoria almacena los programas y datos que requiere la CPU
Sus prestaciones marcan las del ordenador completo: es un cuello de botella
Criterios de diseño de memorias

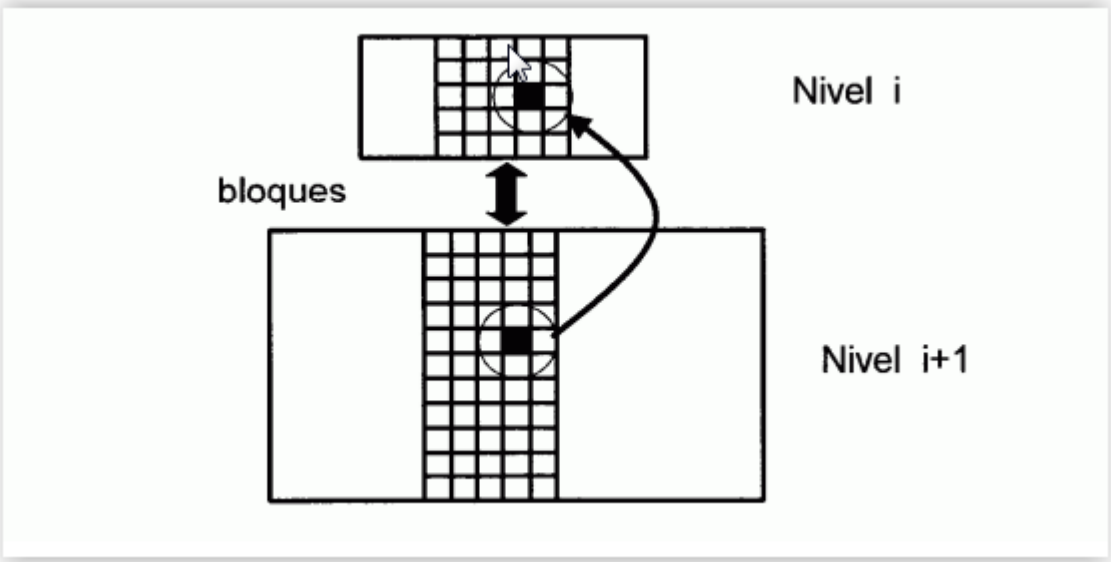
- Coste por bit
- Velocidad
- Capacidad
- Consumo
- Fiabilidad

Estos criterios son incompatibles entre sí, con lo que se llega a una solución de compromiso
La jerarquía de memoria

Jerarquía de memoria



La información fluye de un nivel inferior a uno superior según va siendo necesaria



Introducción

Principio de localidad de las referencias

- Localidad temporal: Alta probabilidad de volver a hacer referencia a un mismo dato en un corto espacio de tiempo
- Localidad espacial: Alta probabilidad de que la siguiente referencia sea a un dato almacenado en una posición cercana a la anterior

Ejemplo sencillo: Ejecución de una rutina

Propiedades de la jerarquía de memoria

Todos los datos contenidos en un nivel se encuentran en un nivel superior
Las copias de un mismo dato en diferentes niveles deben ser coherentes

Características de la memoria

Las memorias se caracterizan por

Ubicación

- Interna: Conectada directamente a la CPU. Ejemplo memoria principal MP
- Externa: Conectada a la CPU a través de E/S. Ejemplo disco rigido

Capacidad

- En las memorias externas se expresa en bytes
- En las memorias internas se expresa en bytes o palabras

Unidad de transferencia

- Palabras para memoria interna
- Bloques para la memoria externa

Recuerde que la memoria interna se organiza en palabras pero la unidad direccionable suele ser el byte

Método de acceso

- Secuencial: Cintas
- Directo: Discos
- Aleatorio: Memoria principal y cache
- Asociativo: Tablas de etiquetas para cache

Prestaciones

- Tiempo de acceso (Ta)
  - Variable en secuencial directo
  - Fijo en aleatorias
- Tiempo de ciclo (Tc): Se define para las memorias de acceso aleatorio como el tiempo mínimo entre dos accesos consecutivos
- Velocidad de transferencia: Se define como 1/Tc para las aleatorias

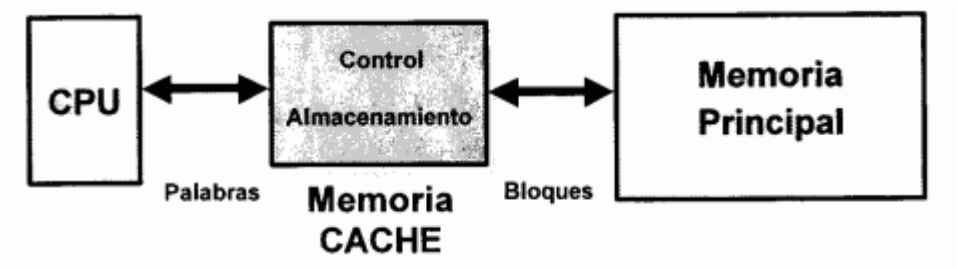
Tipos de Memorias

Tipos

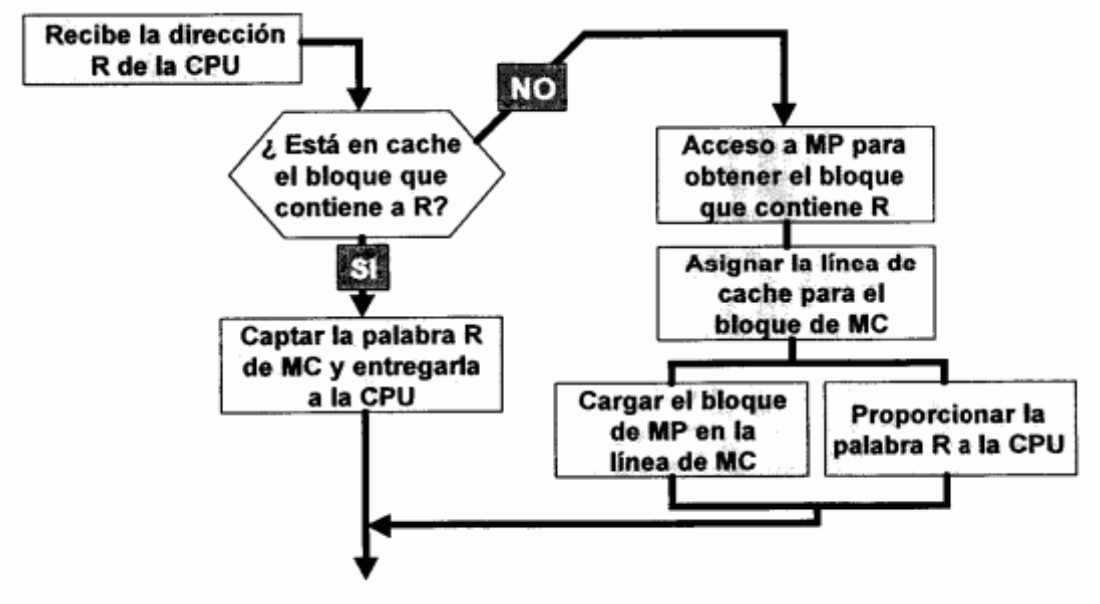
- Registros internos
- SRAM (cache)
- DRAM (memoria principal)
- ROM (memoria no volátil) BIOS
- Flash
- Disco

Memoria cache

Estructura y funcionamiento



- La unidad de transferencia entre la memoria principal (MP) y la memoria cache (MC) es el bloque o línea
- Cada bloque está constituido por un conjunto de 2^r palabras
- La memoria cache y la memoria central está divididas en líneas o bloques de igual tamaño.



Organización
Como se sabe si una posición de MP se encuentra copiada en una posición de MC?
Donde se copian las posiciones de MP en MC

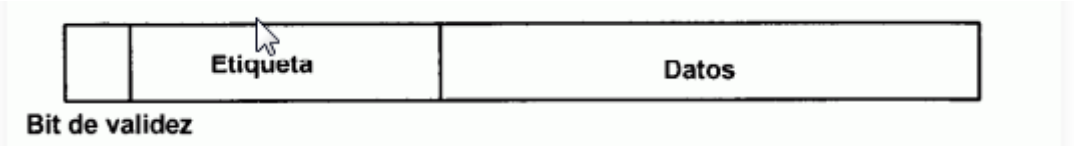
Adquisición
Cuando se lleva una línea de MP a MC

Actualización
Cuando se actualiza el contenido de MC en MP?

Reemplazo
Que línea de MC se desprecia cuando se necesita espacio en MC para otra diferente

Organización
Cada línea de la cache está compuesta por tres campos

- Bit de validez
- Información de etiqueta: Para realizar la correspondencia entre la línea de MC y el bloque de MP que almacena
- Datos



Localización de un bloque de MP en una línea de MC

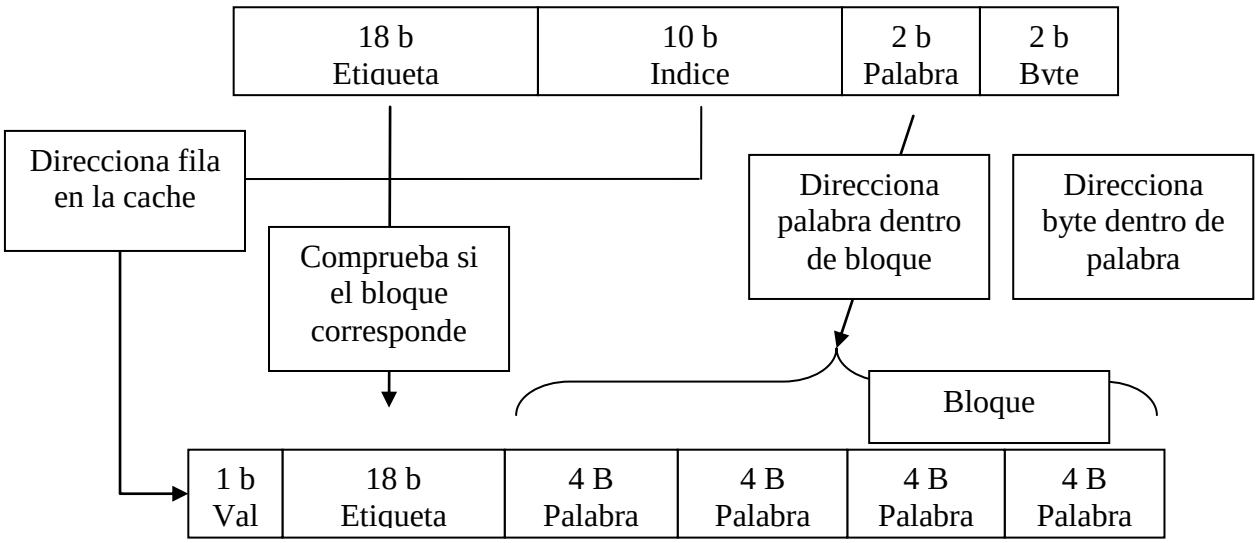
- Mapeo directo
- Completamente asociativa
- Asociativa por conjuntos

Cache de mapeo directo

Organización
Cada bloque de MP se puede almacenar sólo en una determinada línea de MC
Si se tienen m líneas de cache

Si se elige m como potencia de 2

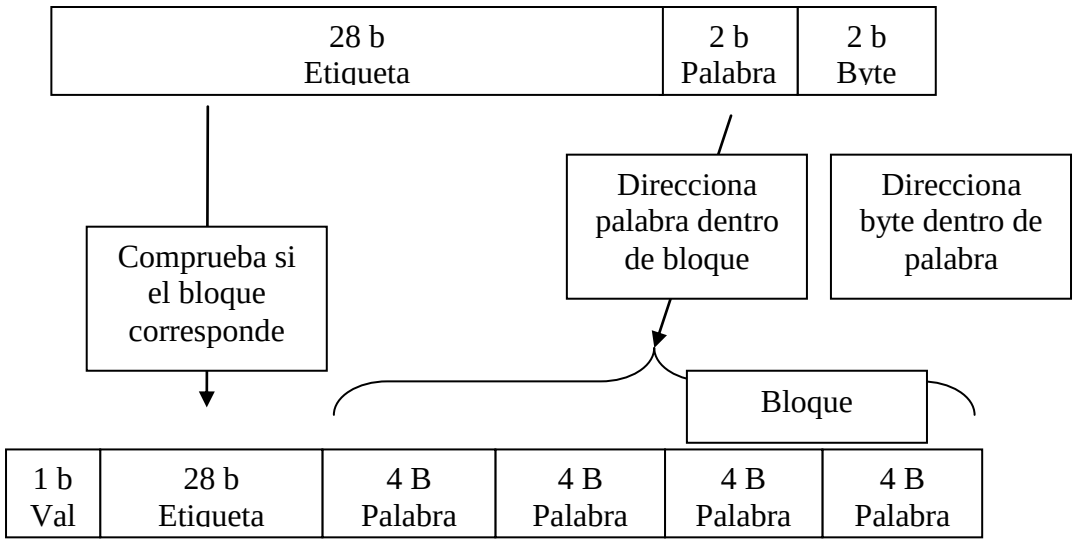
Como se sabe, dada una dirección de MP en que línea de MC se encuentra?



Simple, fácil de implementar
Posición concreta de cache para cada bloque dado

Cache asociativa

Organización
Los bloques de la MP se pueden almacenar en cualquier línea de la MC.



Como se sabe, dada una dirección de MP en que línea de MC se encuentra?
Flexibilidad para que cualquier bloque se encuentre en cualquier línea de MC vs circuitería muy compleja examinar en paralelo todas las etiquetas.

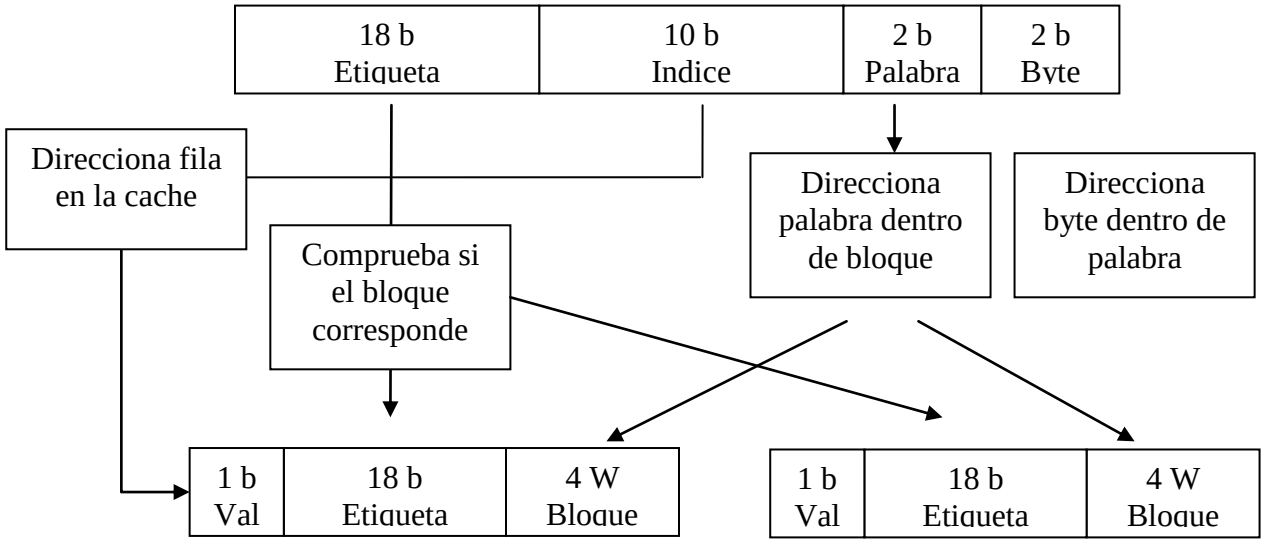
Cache asociativa por conjuntos

Organización
Solución de compromiso entre la organización de mapeo directo y la completamente asociativa.
La cache k asociativa o asociativa por conjunto de k vías se divide en v conjuntos, cada uno con k líneas, cumpliéndose que

m = v x k
m: número de líneas de la cache
v: número de conjuntos
k: número de líneas por conjunto

i = j modulo v
i: número de conjuntos de la cache
j: número de bloques de la MP
v: número de conjuntos

Un determinado bloque de MP se puede cargar en cualquiera de las líneas de cache de conjunto de líneas de cache asociado.



Funcionamiento de la cache

Adquisición
Cuando se lleva un conjunto de MP a una línea de cache

- En fallo
  Se trae la línea solo cuando se quiere leer o escribir

- En fallo de lectura (no write allocate)
  Se trae la línea solo cuando se quiere leer

- En fallo de escritura (write allocate)
  Se trae la línea cuando se quiere escribir

Actualización
Cuando se actualiza el bloque en MP

- Escritura directa (write trough)
  Simultáneamente se escribe en MC y MP
  Optimización de la escritura
  Buffer de escritura: Evita las paradas de la CPU

- Post escritura (write back)
  Se actualiza la MP cuando se reemplaza la línea de la MC
  Mecanismo para indicar inconsistencia entre MC y MP
  Bit extra para indicar línea modificada (M)

Reemplazo
Qué línea se desecha cuando hace falta espacio para otra

- Aleatoria (random)
  La línea a reemplazar se elige en forma aleatoria.

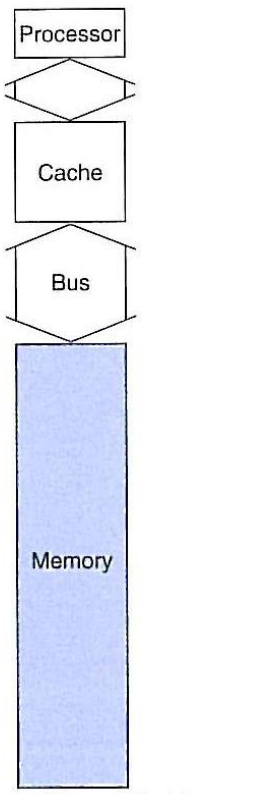
- Menos recientemente utilizada (LRU o pseudo LRU)
  Siguiendo el principio de localidad se reemplaza la línea que más tiempo tiene sin ser referenciada.

- Menos frecuentemente usada (LFU)
  Se reemplaza la línea que haya sido menos referenciada.

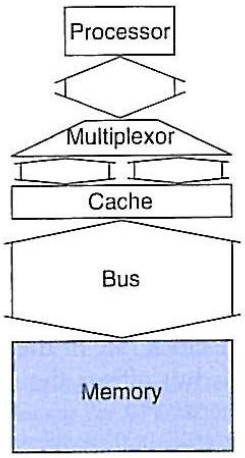
- FIFO (no se usa normalmente)
  Se sigue un orden de reemplazo atendiendo al orden en el que se han ido utilizando las líneas

Conexión con la Memoria Principal

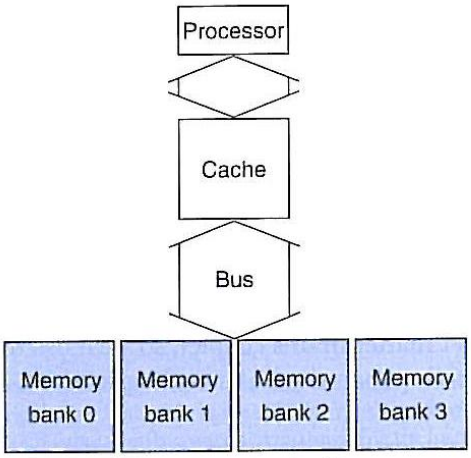
- Memoria de una palabra
- Memoria de N palabras
- Memoria entrelazada



a. One-word-wide memory organization



b. Wider memory organization



c. Interleaved memory organization

Memoria cache multinivel

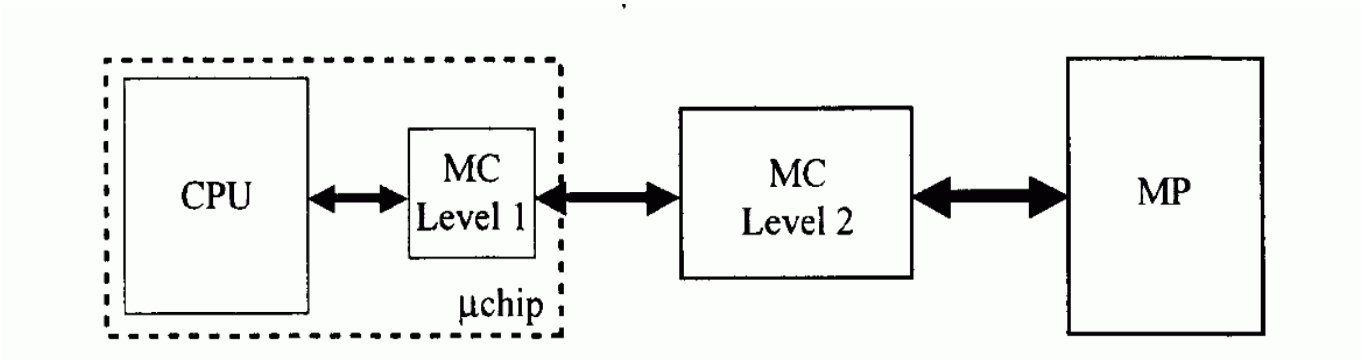
Esquema básico

- La diferencia entre la velocidad de la CPU y la MP crece día a día.
- Se pueden crear sistemas de memoria con varios niveles de MC que tienen normalmente distintos tamaños y velocidades.
- Los sistemas actuales vienen equipados con dos niveles de MC
- El primero integrado dentro del propio procesador con tamaño pequeño. Los accesos a la cache tienen un ciclo de reloj, igual que la CPU.
- El segundo nivel tiene gran tamaño para conseguir que el número de accesos a la MP sea el menor posible.

Organización de los datos

Las MC pueden almacenar tantos datos como instrucciones
Existen diferentes tipos de MC

- Cache de datos: Solo almacena datos
- Cache de instrucciones: Solo almacena instrucciones
- Cache unificada o mixta: Simultáneamente existen datos e instrucciones.



Rendimiento de la cache

- $T_{(i)}$: Tiempo de acceso en el nivel i
- $T_{(i+1)}$: Tiempo de acceso en el nivel i+1
- α es la tasa de acierto (hit ratio)
- $T_{ap(i)}$: Tiempo de acceso aparente en el nivel i

$$T_{ap(i)} = \alpha\, T_{(i)} + (1-\alpha)\, T_{fallo(i)}$$

- $T_{fallo(i)} = T_{penalización(i)} + T_{(i)}$
- $T_{penalización(i)} = T_{ap(i+1)}$
- $\theta = (1-\alpha)$ es la tasa de fallo (miss ratio)

$$T_{ap(i)} = T_{(i)} + \theta T_{penalización(i)}$$

- A nivel del procesador

$$T_{CPU} = \left[\begin{array}{c}\text{Ciclos de}\\ \text{ejecución}\end{array} + \begin{array}{c}\text{Ciclos de}\\ \text{bloqueo}\end{array}\right] \times \text{Tiempo de ciclo}$$

- Ciclos de ejecución = I x CPI
- Ciclos de bloqueo = $I \times \left[\dfrac{\text{Referencias a memoria}}{\text{Instrucción}}\right] \times \left[\text{tasa de fallos}\right] \times \left[\text{Penalización por fallo}\right]$

Memoria Virtual

Una computadora utiliza memoria virtual cuando

- El espacio de direcciones que utilizan los programas durante su ejecución es mayor que el de las direcciones físicas de la MP.
- El SO debe administrar la memoria y brindar un mapa de memoria único para cada proceso

Funcionamiento general

- El espacio de direcciones virtuales (instrucciones y datos) que maneja un programa se divide en bloques.
- En un instante determinado en MP solo se encuentran unos pocos bloques del programa.
- El resto de los bloques se mantiene en memoria secundaria (area swap del disco).
- Se van trayendo nuevos bloques de la MP a medida que se van necesitando.

Objetivos

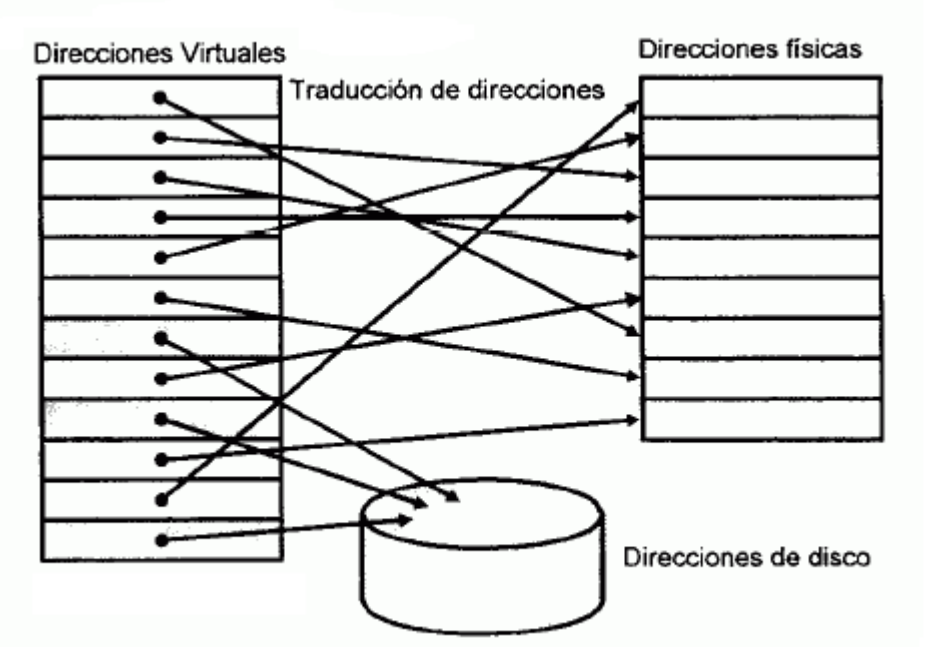
- Permitir tener un espacio de direcciones superior al real.
- Permitir compartir eficientemente memoria (entornos multiproceso)

Conceptos

La CPU produce direcciones virtuales que se traducen en una direccion fisica que se usa para acceder a la MP.

Según el tipo de traducción

- Memoria virtual paginada
- Memoria virtual segmentada
- Memoria virtual segmentada/paginada

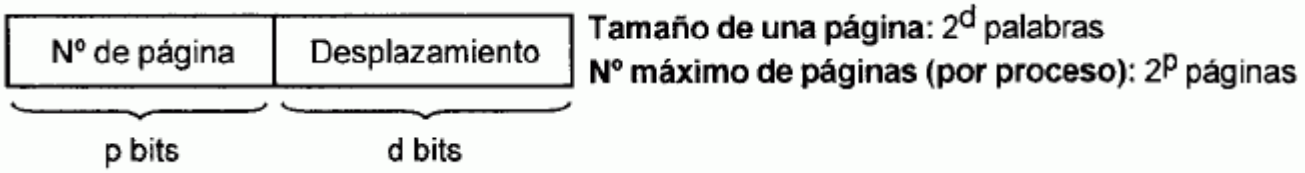


Memoria virtual paginada

Espacio virtual de direcciones

Se divide en páginas de tamaño fijo

Dirección virtual



Espacio físico de direcciones

Se divide en marcos de página de igual tamaño que una pagina.

Dirección física

Traducción de direcciones

Es necesario un mecanismo de correspondencia para conocer en que marco de página M de MP esta ubicada una determinada página de memoria virtual P.

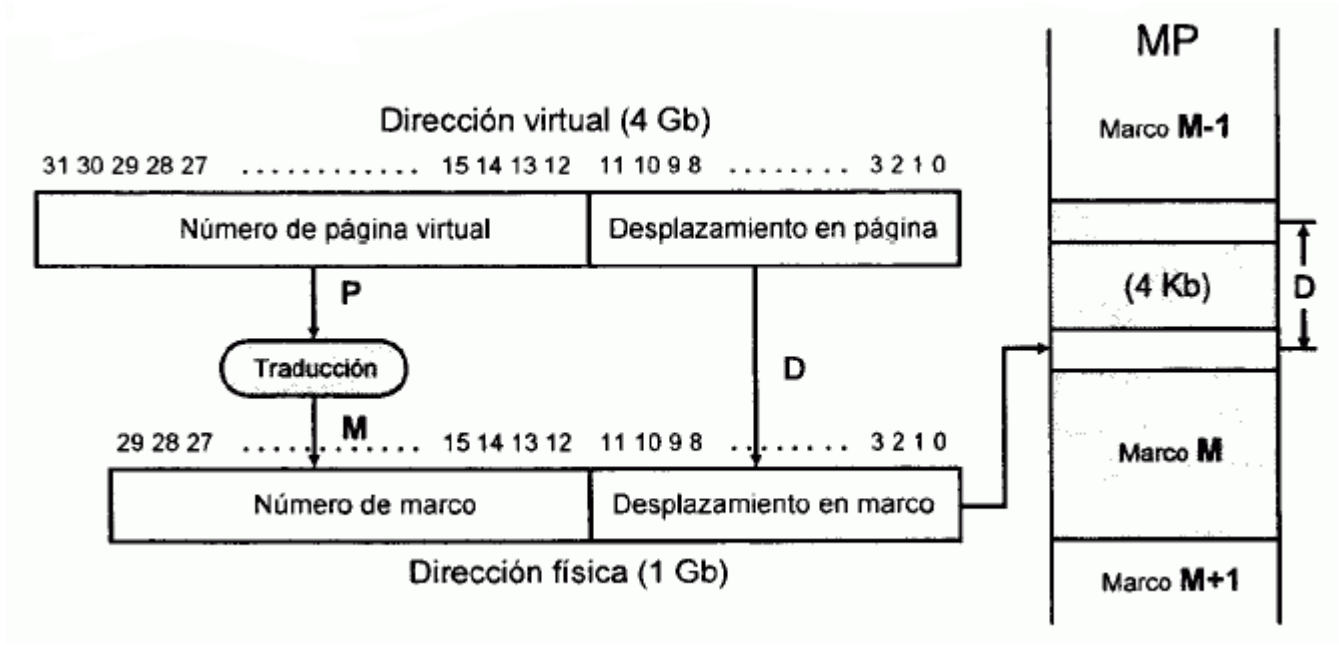
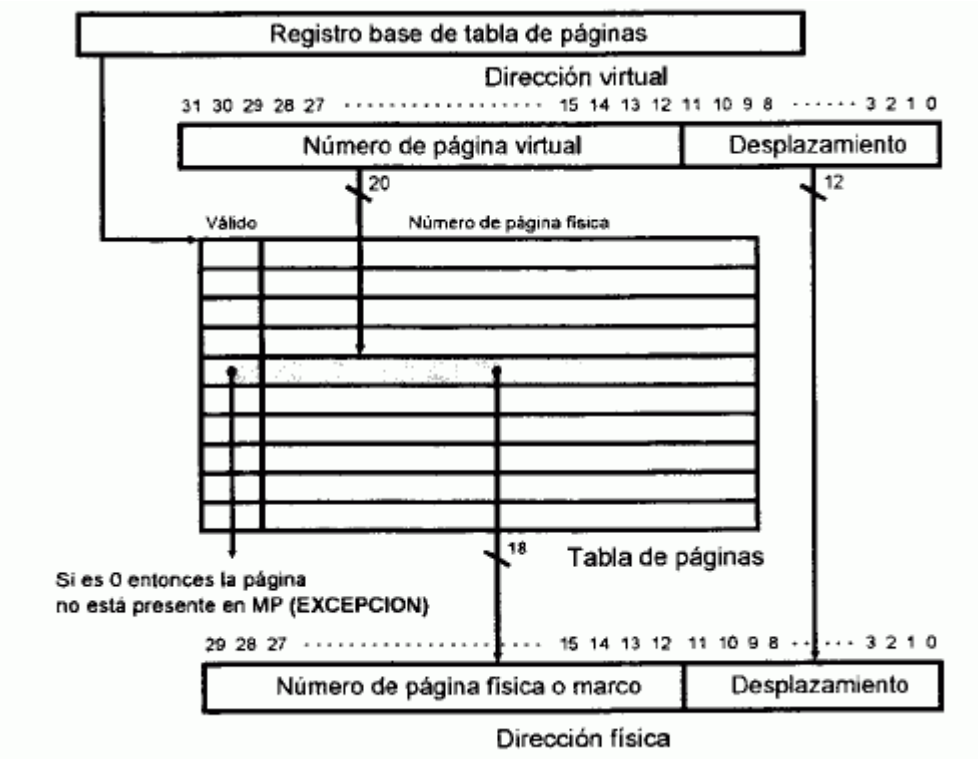


Tabla de páginas

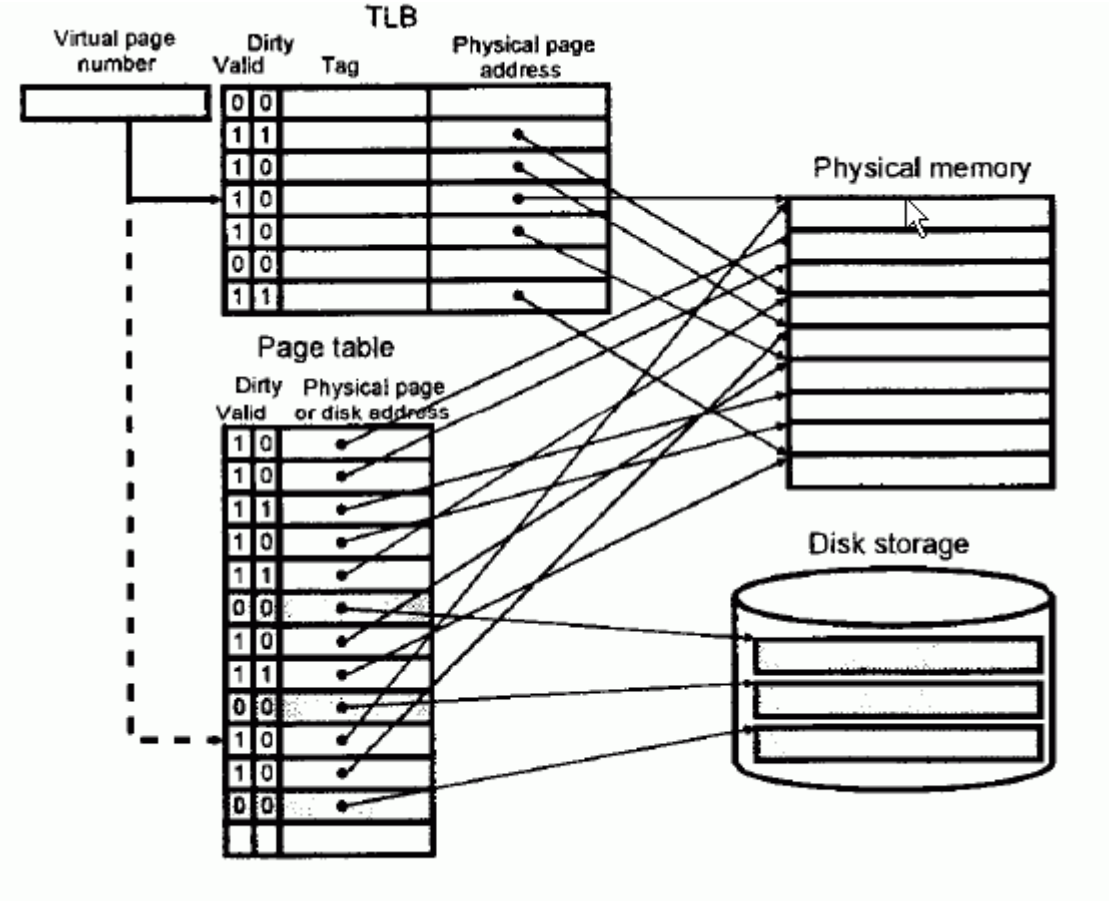
- Existe una tabla de páginas TP por cada proceso J
- La TP contiene una entrada por cada posible página del proceso J: 2P entradas por proceso
- La entrada p-esima de TP contiene el marco de página M donde esta ubicada la pagina P (si esta en MP).
- La TP de un determinado proceso J esta apuntada por un registro base asignado a se proceso



Problemas al almacenar la TP en MP

- La TP puede ser de gran tamaño (4 Mb en el ejemplo)
  - Uso de registro limites
  - Tabla de hash
  - Varios niveles de tablas
  - Paginar tablas
- Se requieren dos accesos a MP por cada referencia: Se duplica el tiempo de acceso.
  - Cache especial que guarda las traducciones mas recientes TLB (table lookaside buffer)

TLB



Gracias al principio de localidad las traducciones se vuelven a utilizar en breve
El TLB es una memoria cache especial que permite mejorar las prestaciones, al almacenar las traducciones mas recientes.

Cada entrada en la TLB consta de tres partes

- Número de página virtual
- Número de marco
- Bits de control

Funcionamiento

- Política de emplazamiento: Todas las páginas son de igual tamaño y coinciden en tamaño con los marcos de página. Una página se puede ubicar en cualquier marco libre
- Política de reemplazo de página: Cuando no hay ningún marco de página libre se debe reemplazar algunas de las páginas de MP. Los principales algoritmos FIFO, LFU, LRU
- Políticas de actualización de la MS: Es necesario mantener la coherencia entre MP y MS, cuando se modifica una página en MP es necesario actualizar esa misma página en MS. Se emplea la política de post-escritura, una página modificada se actualiza en MS solo cuando se reemplaza.