



75-08 Sistemas Operativos

Lic. Ing. Osvaldo Clúa

2011

Facultad de Ingeniería
Universidad de Buenos Aires

Memoria Virtual

Objetivos

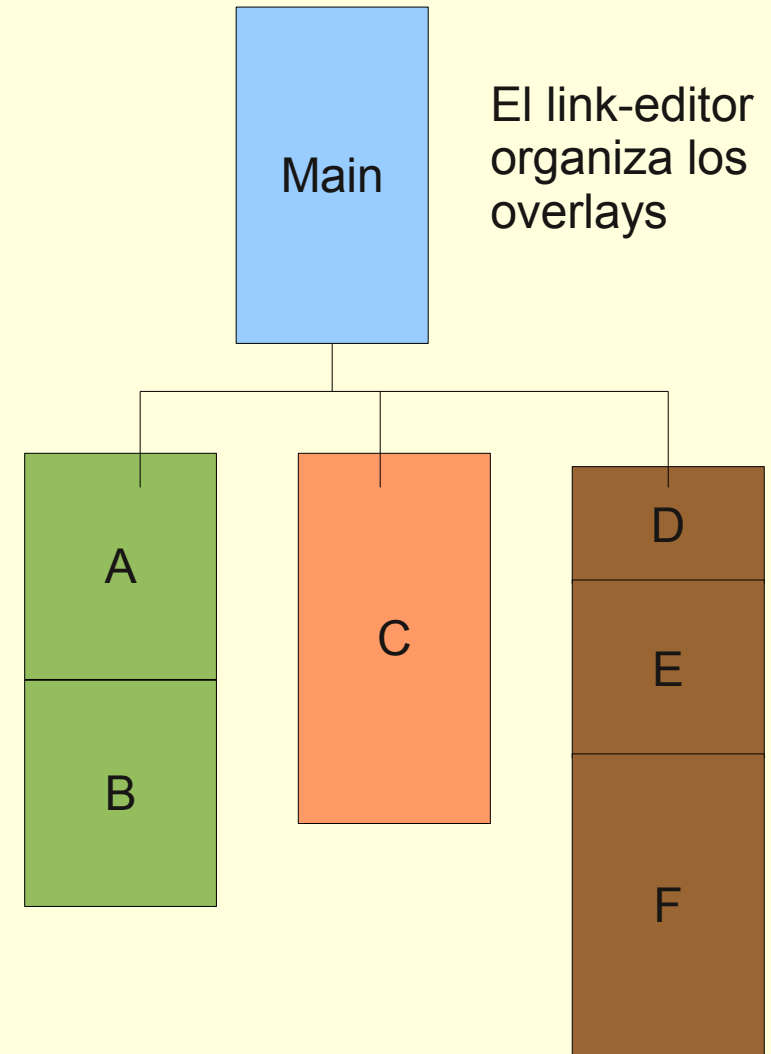
- Permite separar el direccionamiento del almacenamiento.
 - Soluciona los problemas de fragmentación
 - Soluciona la reubicación de código y datos.
 - Permite direccionar mas memoria que la instalada
- Es parte de la arquitectura - requiere soporte de hardware

Memoria instalada y Memoria requerida

- Como correr programas demasiado grandes para la cantidad de memoria disponible.
- Una solución fue la de usar **Overlays** para compartir partes de la memoria.
 - Reemplaza un bloque de código por otro.
 - El programador debe planificar el uso de overlays según el uso de las rutinas del sistema.
- Aún se usa en PDAs, Celulares y Embedded.

Overlays

- Main puede llamar a cualquier procedimiento.
- A puede llamar a B
 - No a C ni a D, E o F.
- C no puede llamar a ninguno
- Cada una de las tres partes son excluyentes.



Páginas (Page)

- Las direcciones generadas por la CPU se dividen bloques de tamaño fijo llamados páginas.
 - Estas direcciones se llaman direcciones virtuales.
 - Generalmente de 2 o 4 KB para evitar fragmentación interna.
 - Esta técnica permite el uso de memoria no contigua.

Marcos de Páginas (Frames)

- La memoria principal se divide en frames.
 - Del mismo tamaño que las páginas.
- Las direcciones de los frames se llaman direcciones reales
- Una unidad de Hardware mapea las direcciones virtuales en reales.
 - La MMU (Memory Management Unit) hace esta traducción de direcciones.

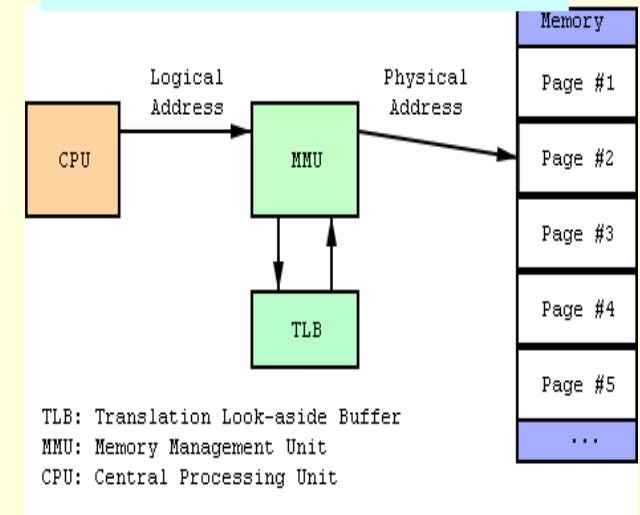
La MMU

Memory management unit

- Responsable de traducir las direcciones virtuales (o lógicas) a direcciones reales (o físicas)
- En general hace uso de una cache asociativo, el Translation Lookaside Buffer o TLB.



MMU 68451 usado con el Motorola 68010

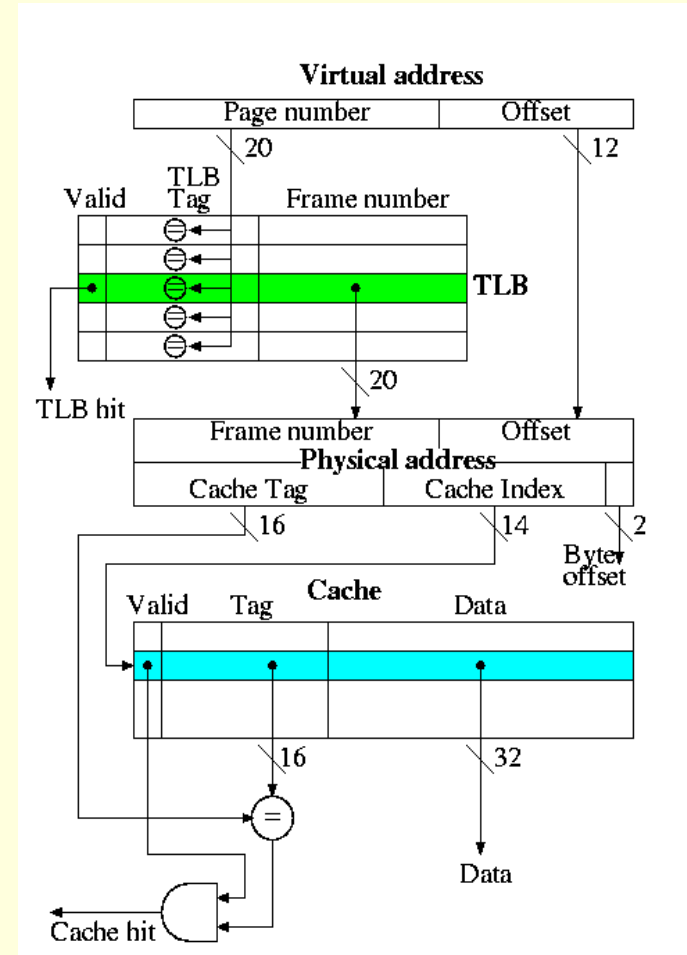


Haciendo números

- Por cada instrucción hay al menos 3 accesos a memoria.
- Con paginado, cada acceso a memoria consiste en una búsqueda en tabla y un acceso.
- ¿En cuanto se aumenta el tiempo de ejecución de la instrucción?
 - Más del 30% sería inadmisibile.

Usando Context Addressable Memory (CAM)

- Como cache de la tabla se usa un **TLB** (Translation Lookaside Buffer).
- La penalización por acceso depende si la dirección se encuentra o no en la tabla.
 - La figura muestra una Tabla de Páginas Invertida (a ver mas adelante)



Hit/Miss ratio

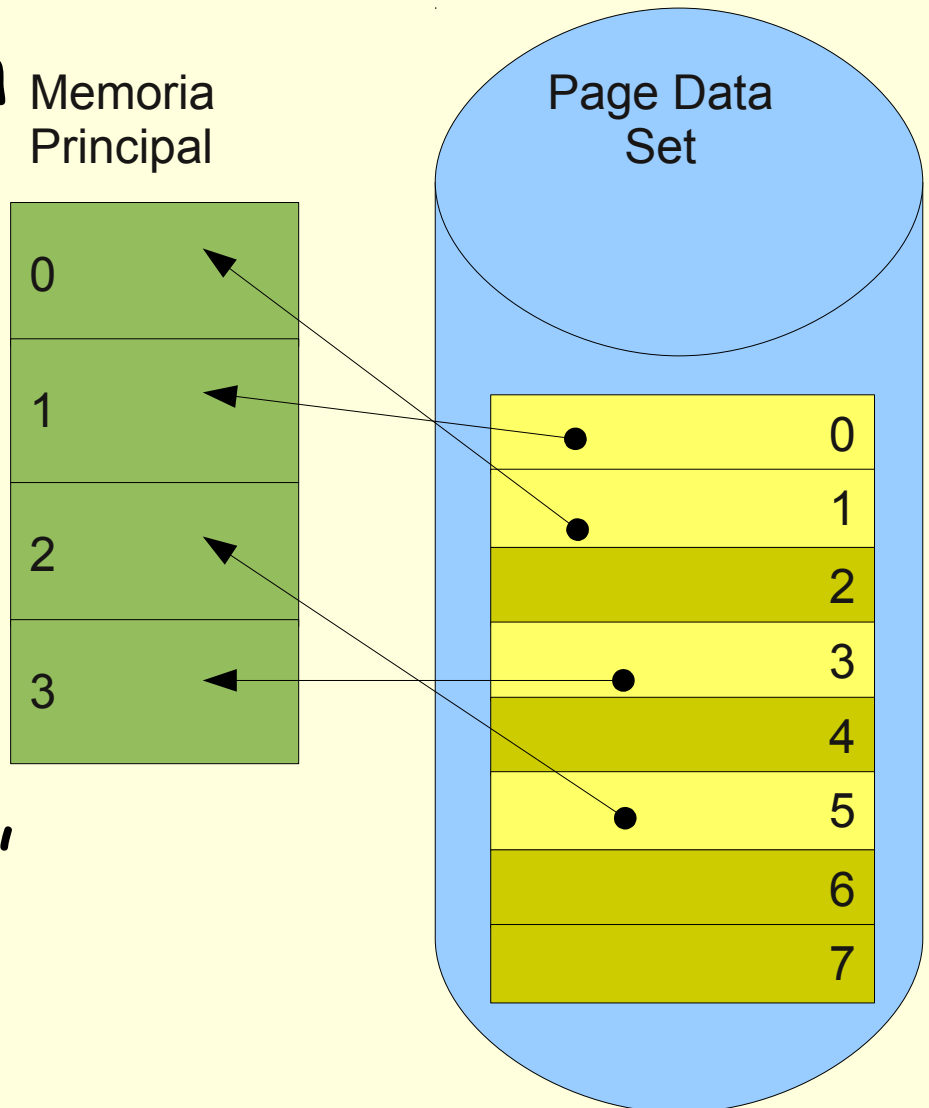
- La relación entre aciertos y fracasos se llama Hit/Miss.
 - 95% de éxitos
 - 5 ns de acceso a TLB
 - 100 ns de acceso a memoria.
 - $0.95 \cdot 105 + 0.05 \cdot 205 = 110$ ns (10 % de penalidad)
 - Son típicas relaciones de 0.995 a 0.998

Manejo del Miss de TLB

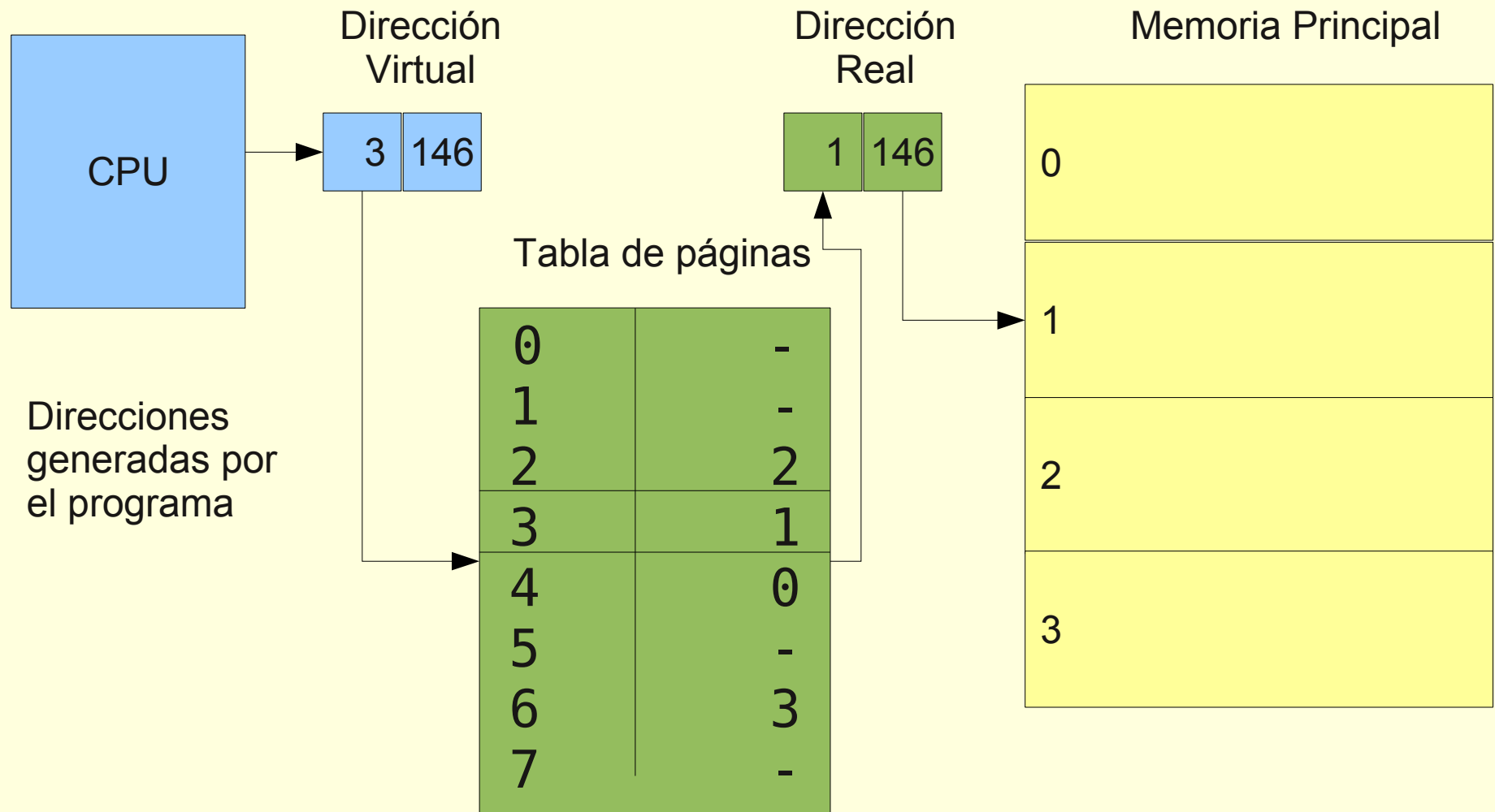
- Por Hardware (ej. **IA 32/X86-64**).
 - La búsqueda en las tablas forma parte de la **Microarquitectura** de la traducción.
 - Si se encuentra la página, se sube la entrada al TLB.
 - Si no hay Page Fault.
- Por Software (ej. **MIPS**)
 - La arquitectura provee una TLB-MISS exception.
- Hay esquemas mixtos (**SPARC, Itanium**)

Memoria Virtual

- En la memoria no caben todas las páginas.
- Las páginas se guardan en disco en un "page data set".
- Si la página no está, se produce un "page fault"



Traducción de direcciones (Memoria real < Memoria virtual)



¿Qué se logra con la Memoria Virtual?

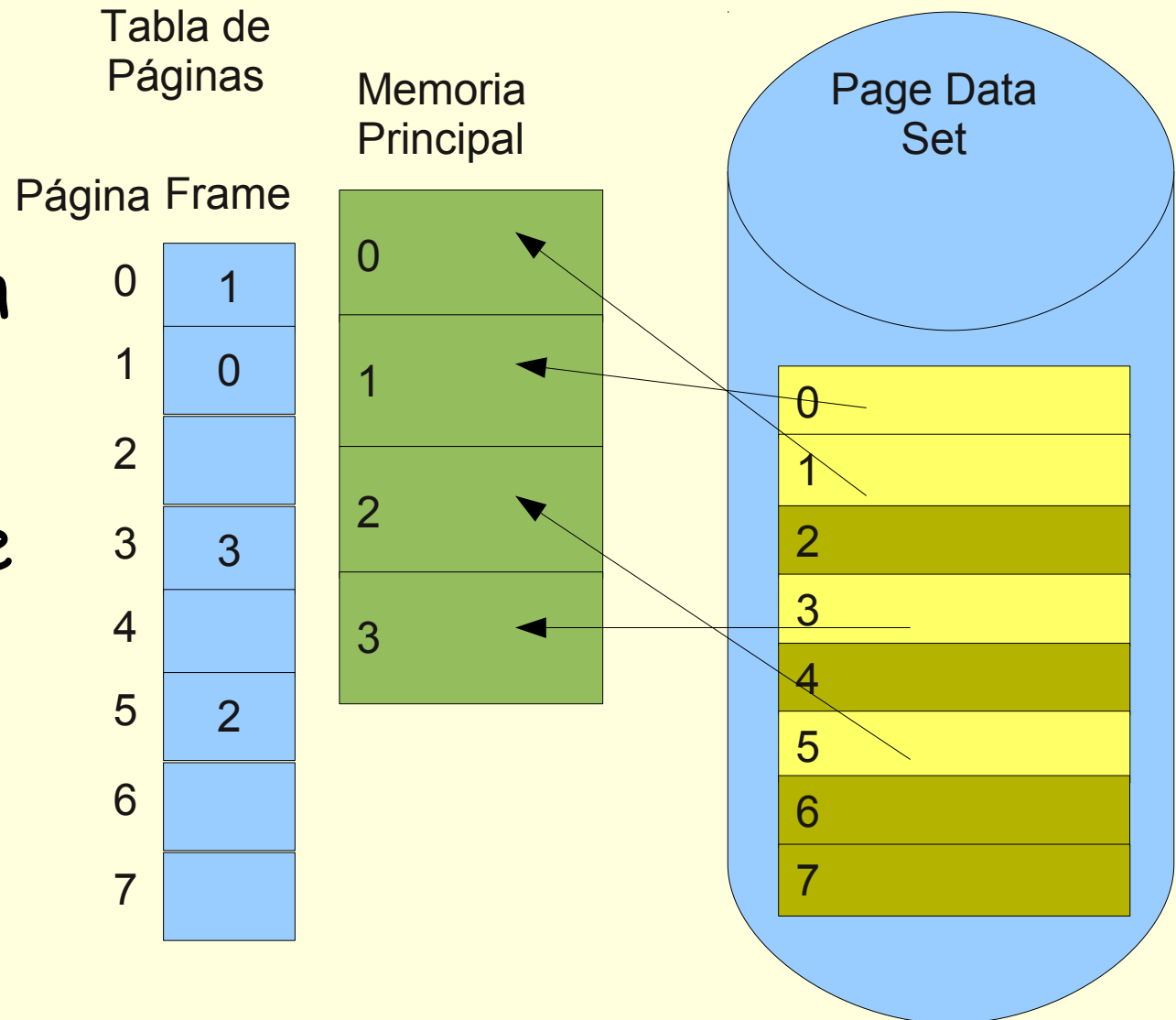
- La memoria del usuario queda separada de la memoria física.
 - Solo una parte del programa debe estar en la memoria para su ejecución.
 - El espacio de direcciones de los procesos pueden ser mas grandes que la memoria disponible.
 - Permite compartir los espacios de memoria.

Oportunidades de optimización

- Algoritmos de paginado.
 - Y que partes pasar al Hard.
 - Y como evitar los page faults.
- Estructura de las tablas de páginas.
- Organización del Page Data Set.
- Algunas otras cosas que se resuelven con el paginado.

Paginado por demanda (1)

- No todas las páginas se encuentran en memoria
- Las páginas se traen a memoria cuando se las necesita



Paginado por demanda (2)

- Solo se traen a memoria las páginas que se van a usar.
- Un bit adicional en la Page Table (y en el TLB) indica si la página está cargada o nó.
 - Bit de página válida.
- Un acceso a una página inválida produce un page fault.
 - Es una software trap que interrumpe al proceso.

Proceso de un Page Fault (1)

- Al no encontrarse la página en memoria, se debe ingresar la copia del disco.
- Para eso se debe buscar una página a sacar.
 - Si está modificada, se debe copiar primero a disco.
- El proceso que sufre el page fault se bloquea.

Proceso de un Page Fault (2)

- Si hay frames libres
 - se copia la página a ese frame.
 - se actualiza la page table.
 - el proceso pasa a ready para continuar su ejecución.
- Si no hay frames libres
 - se debe desalojar una página de la memoria

Algoritmos de reemplazo de páginas

- Siempre se trata de reemplazar primero una página que no fue modificada (clean).
 - Si fue modificada (dirty) debe escribirse primero a disco.
 - Un bit más en la page table (y en el TLB) marca si es clean o no.
- Con las páginas limpias se pueden usar distintos algoritmos

Óptimo

- Reemplazar la página cuyo uso está mas lejano en el futuro.
- Debe conocerse a priori el comportamiento del programa.
 - En general no es posible
 - Si lo es, conviene usar overlays.
- Sirve como comparación para el resto de los algoritmos.

FIFO

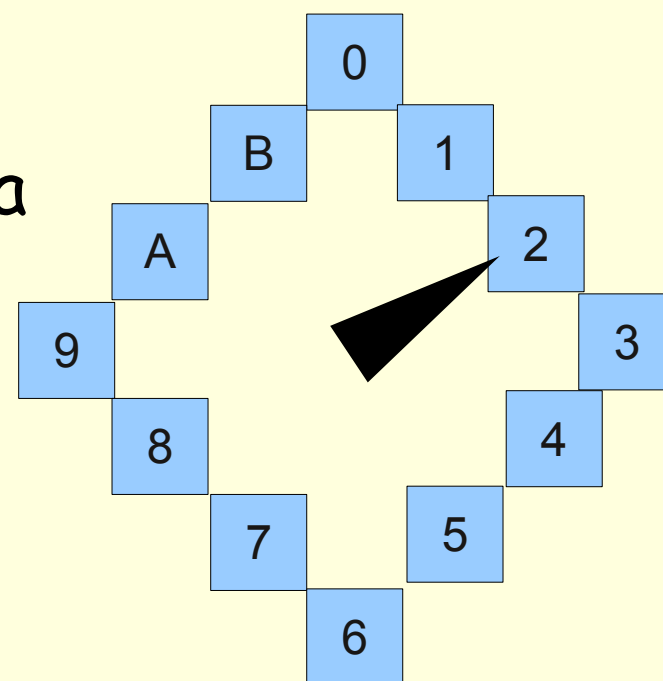
- Se reemplazan las páginas que hace mas tiempo que están en la memoria.
 - Requiere mantener una cola de páginas.
 - No es buena idea
 - reemplaza las páginas del scheduler o del Kernel.
- Este algoritmo experimenta la anomalía de Belady
 - En algún caso, con mas frames, pagina mas

No Usado Recientemente (NRU)

- Se agrega un bit de referencia (r) al de dirty (modificado) al TLB.
- Con cada acceso a una página, se marca este bit r.
- Cada tanto, se limpian todos los bits de referencia.
- La preferencia de la elección está en las páginas:
 - no referenciadas no modificadas.
 - referenciada y no modificada
 - no referenciadas, modificadas (el reloj limpió el bit r)
 - referenciada y modificada

2da Oportunidad/Reloj

- Son mejoras al FIFO.
 - 2nd chance - usa el bit r y una cola para decidir si se reemplaza o no.
 - Clock - usa una lista circular
 - la "aguja" indica la página mas vieja.
 - Ambos limpian el bit r en cada pasada.



Si $r == 0$ reemplaza
sino, limpia r y avanza

Menos Recientemente Usado LRU

- Cada referencia a memoria actualiza un time stamp en la page table.
 - Se reemplaza la página cuyo time stamp sea el mas antiguo.
- Requiere bastante auxilio de hard para tener una performance adecuada.
- Hay variantes en software
 - **NFU** (Non Frequently Used) o **Aging**

Working Set

- Los programas exhiben un comportamiento conocido como **localidad** de referencia.
 - En cada fase de su ejecución, el proceso referencia solo a un pequeño número de páginas (no necesariamente contiguas).
 - Ese conjunto se llama Working Set (aunque la definición cambia con la implementación).
- El Working Set va cambiando a medida que progresa la ejecución

Pre-paginado

- Si un proceso pagina durante mas tiempo que el que ejecuta se dice que hace **thrashing**.
 - thrash= sacudir
 - no confundir con trash=basura
- Se pre-pagina (cargan en memoria) todas las páginas del último Working Set del proceso.
- Se re-calcula el Working Set a intervalos.
- Ejemplo WSClock.

Tablas de páginas Locales y Globales

- Algunos Sistemas permiten el reemplazo de cualquier página (paginado global).
- Otros solo permiten que un proceso pagine sobre si mismo, evitando efectos en la performance del resto.
 - Esta estrategia es la que se usaría en un Sistema Operativo orientado a Objetos.

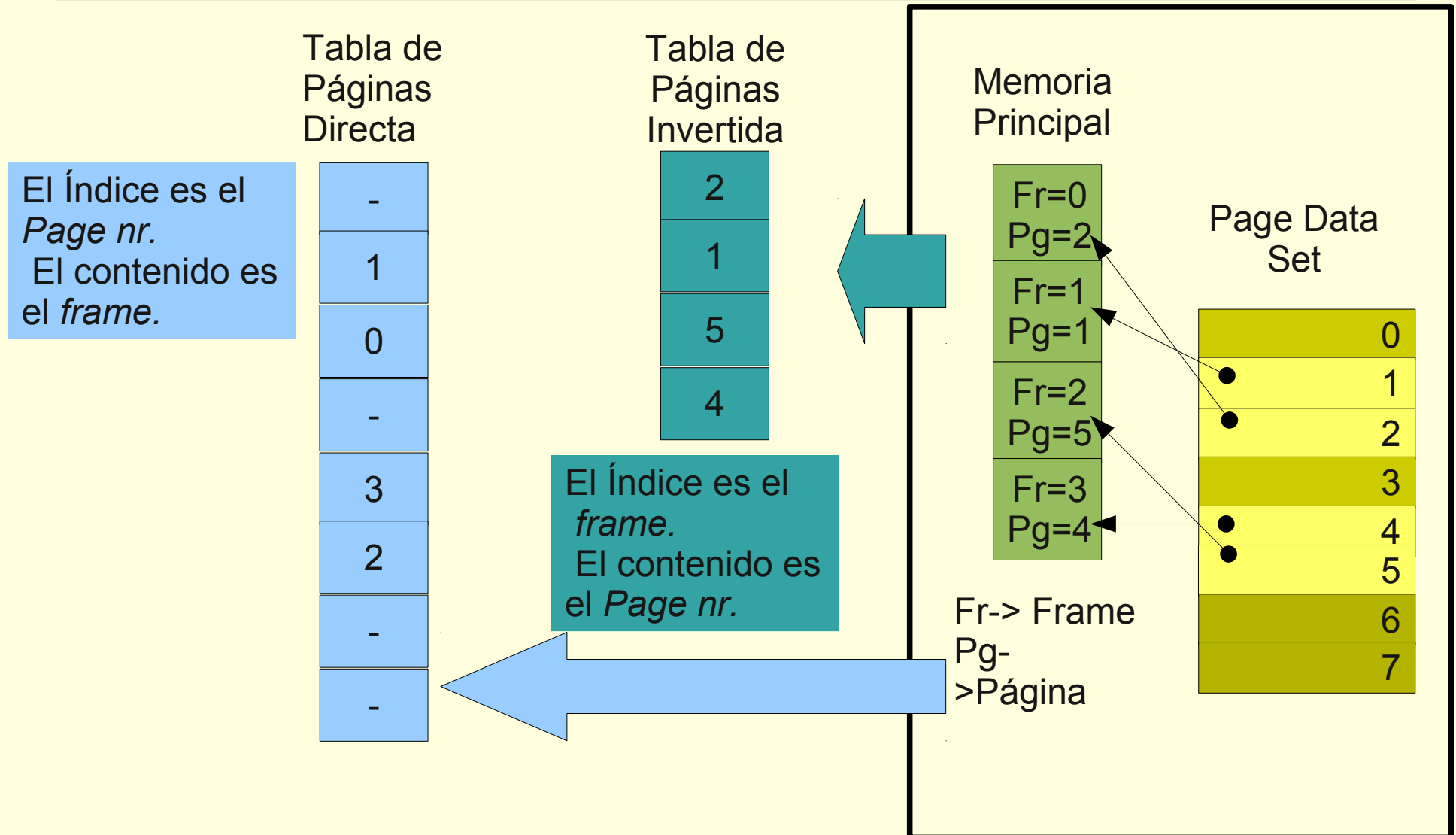
Estructura de las page tables

- 32 bits direccionan 4.294.967.296 bytes.
 - Con páginas de 4K son 1.048.575 entradas.
 - ¿Con 64 bits?
- Una idea es usar varios niveles de páginas.
 - El primer nivel indica en que tabla buscar.
 - El segundo indica el frame.
 - Solo hay tablas secundarias para las páginas que están en memoria.

Inverted Page Tables

- Combina una tabla de frames y una de páginas en una sola estructura.
 - Una entrada por cada frame.
 - El índice es el número de frame.
 - Contiene el número de la página en memoria.
- La búsqueda es asociativa por contenido (TLB).

Tabla de Páginas Directa e Invertida



Archivos de paginado

- Una página no es una buena unidad de transferencia.
 - Se usan entonces láminas (slab) de páginas en cada transferencia.
 - La ubicación de una página es entonces $\#slab + offset$.
- Las slabs se acomodan en el page data set (o partición de paginado).
 - Que se formatea y ubica por anticipado.

Paginado de código

- Las páginas de código son read-only.
 - Se paginan directamente desde el archivo del programa ejecutable.
 - El elf tiene previsiones para ello.
 - Es una forma muy eficiente de cargar un programa a memoria.
- El resto de las páginas tiene su "shadow" en disco.

Otros tópicos

- El paginado interactúa con la I/O.
 - Se deben fijar las páginas donde hay transferencia desde memoria secundaria.
- Los archivos pueden accederse como memoria virtual.
 - más en la clase de archivos...
- Algunos sistemas permiten tamaños de página distintos (**Intel**).