

Memoria virtual

Introducción

- Memoria virtual es un mecanismo que permite que la memoria principal parezca mas grande que su tamaño físico.
- Permite ejecutar programas mas grandes que la memoria física disponible.
- La memoria principal actúa como caché de la memoria secundaria (disco duro).

Definiciones

- Espacio de direcciones. Rango de localidades de la memoria virtual accesibles por un programa.
- Dirección virtual. Dirección en el espacio de direcciones del programa.
- Dirección física. Dirección en la memoria principal.
- La memoria virtual incluye mecanismos para traducir direcciones virtuales a direcciones físicas.

Definiciones

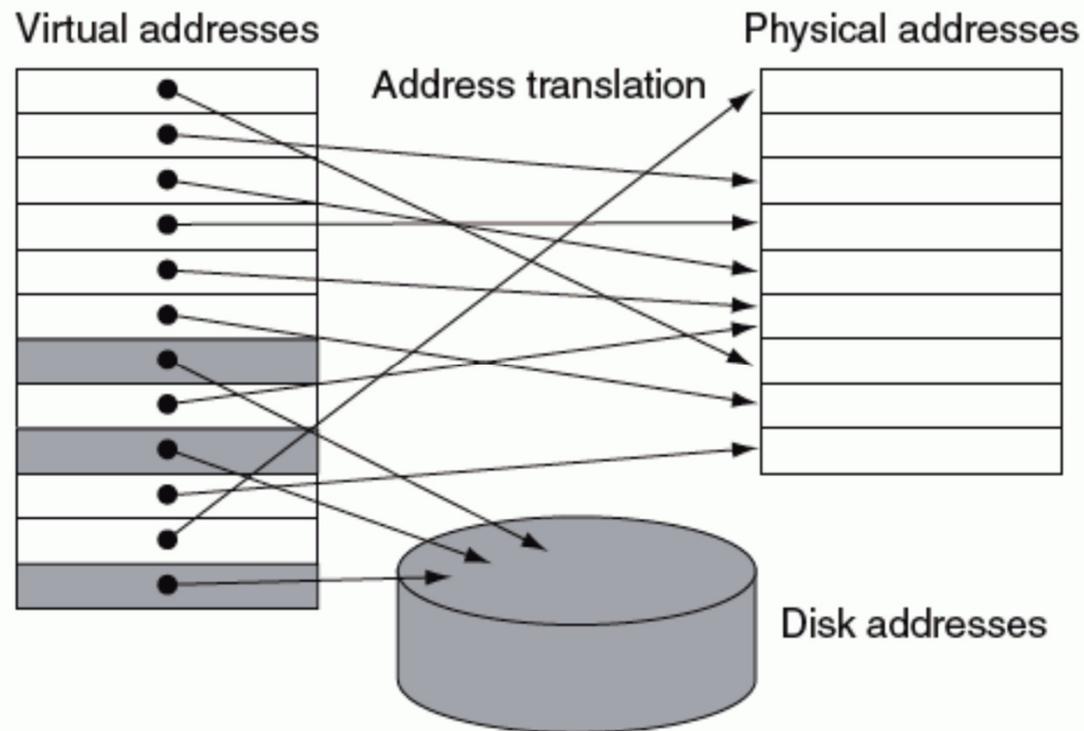
- Protección. Mecanismos para asegurar que múltiples procesos que compartan la CPU, memoria o dispositivos de I/O no interfieran uno con otro.
- La protección también aísla los procesos del usuario de los procesos del sistema operativo.

Memoria y cachés

- Los conceptos en memoria virtual y cachés son los mismos pero con nombres diferentes.
- Un bloque en memoria virtual se llama “página”.
- Una falla en memoria virtual se llama “falta de página”.
- En memoria virtual la CPU produce una dirección virtual que es traducida por hardware y/o software a una dirección física que se usa para acceder la memoria principal.

Mapeo de direcciones

- Esta traducción se le llama “mapeo de direcciones” o “traducción de direcciones”.

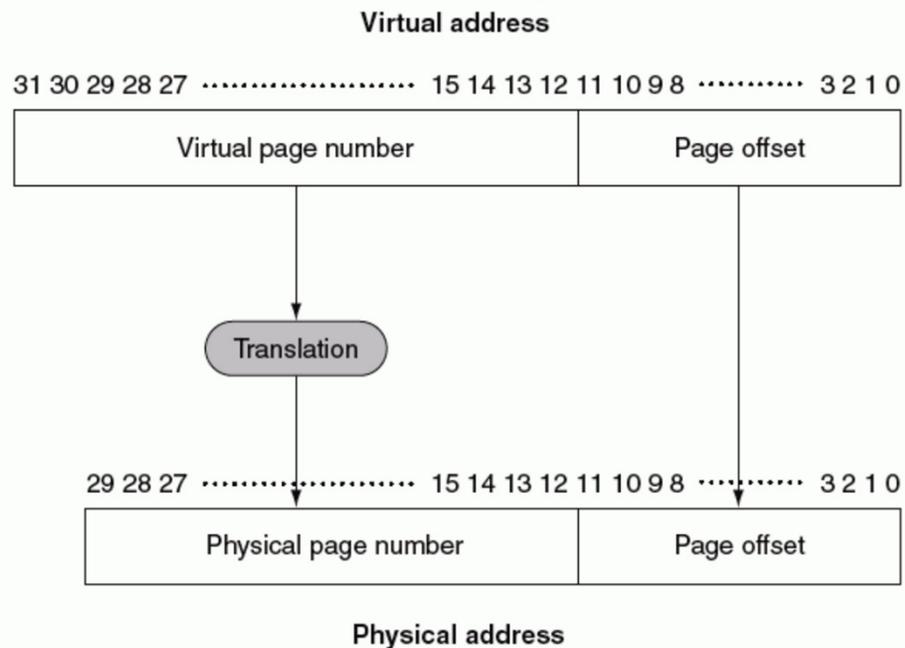


Páginas

- La memoria física y la memoria virtual están divididas en páginas de tamaño fijo.
- Un programa se compone de una o más páginas.
- El sistema operativo carga en memoria principal un número suficiente de páginas del programa.

Direcciones virtuales y físicas

- En memoria virtual, una dirección se compone de un número de página virtual y un offset.
- El número de página virtual se traduce a un número de página física.



Direcciones virtuales y físicas

- El número de bits del offset determina el tamaño de la página.
- El número de bits del número de página virtual es mayor o igual al número de bits del número de página física.
- La idea es que la memoria virtual sea más grande que la memoria física.

Consideraciones de diseño

- La falta de página es muy costosa.
- La memoria principal es 100,000 veces más rápida que el disco duro.

Feature	Typical values for L1 caches	Typical values for L2 caches	Typical values for paged memory	Typical values for a TLB
Total size in blocks	250–2000	2,500–25,000	16,000–250,000	40–1024
Total size in kilobytes	16–64	125–2000	1,000,000–1,000,000,000	0.25–16
Block size in bytes	16–64	64–128	4000–64,000	4–32
Miss penalty in clocks	10–25	100–1000	10,000,000–100,000,000	10–1000
Miss rates (global for L2)	2%–5%	0.1%–2%	0.00001%–0.0001%	0.01%–2%

Fuente: COD 5, p. 454.

Consideraciones de diseño

- Las páginas deben ser suficientemente grandes para amortizar el costo de acceso.
- El tamaño de página es típicamente entre 512 y 8192 bytes.
- Las páginas de 4096 bytes (4KB) son comunes.
- Para reducir la tasa de faltas, los bloques pueden ir en cualquier parte de la memoria principal (i.e. la memoria virtual es fully associative).

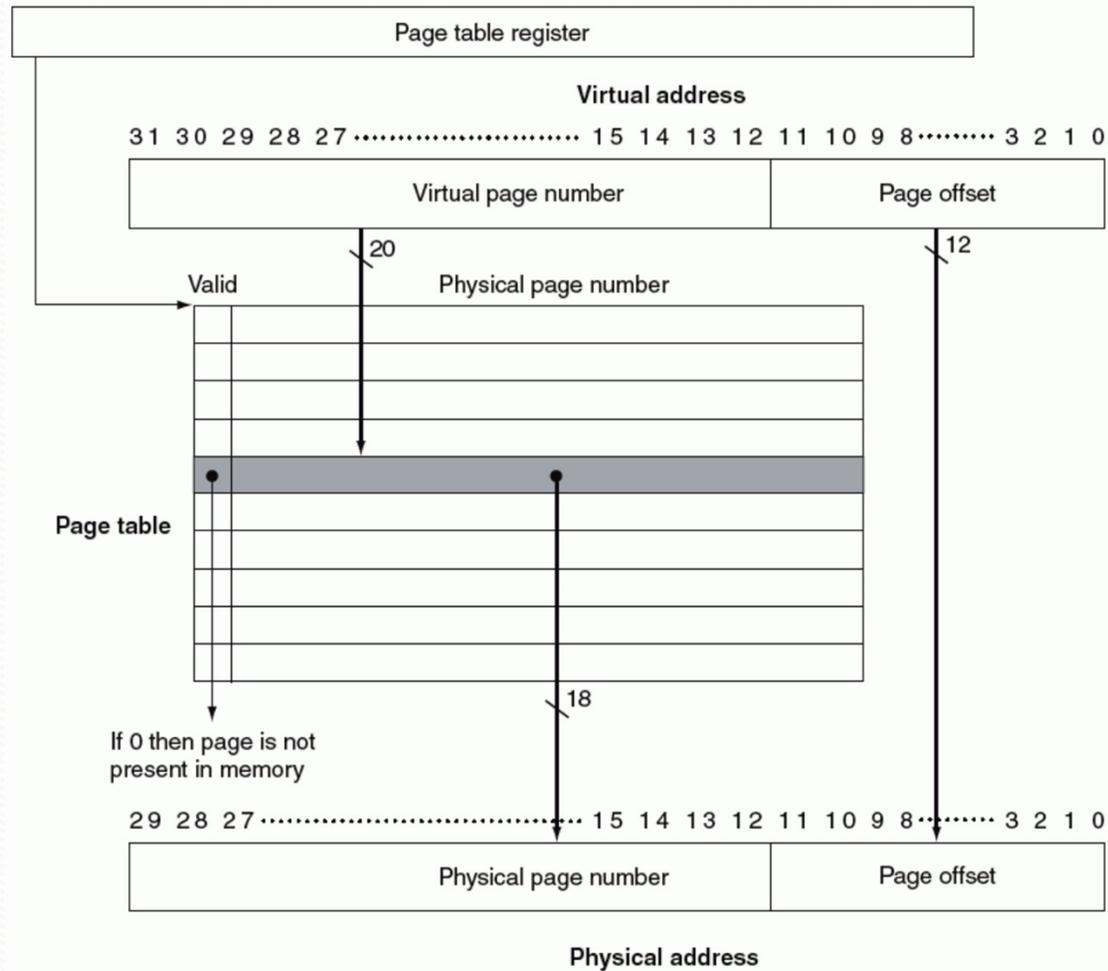
Consideraciones de diseño

- Las faltas de página se manejan por software para permitir algoritmos “inteligentes”.
- Cualquier reducción en la tasa de faltas vale la pena el esfuerzo de implementación.
- Write-through no funciona para la memoria virtual. Los sistemas de memoria virtual usan write-back.

Buscando una página

- Una página puede estar en cualquier lugar de la memoria.
- Se usa una tabla para acceder la memoria llamada “tabla de páginas”.
- La tabla de páginas se indexa con el número de página virtual y regresa el número de página física.
- La CPU incluye un registro para apuntar a la tabla de páginas del programa que está corriendo.

Tabla de páginas



Explicación

- El registro de tabla de páginas apunta al comienzo de la tabla de páginas.
- El tamaño de página es $2^{12} = 4$ KB.
- El espacio de direcciones virtuales es $2^{32} = 4$ GB.
- El espacio de direcciones físicas es $2^{30} = 1$ GB.
- El número de entradas en la tabla de páginas es $2^{20} = 1,048,576$.
- Cada entrada tiene un bit válido para indicar si el mapeo es legal.
- Si el bit es falso, se genera una falta de página.

Tabla de páginas

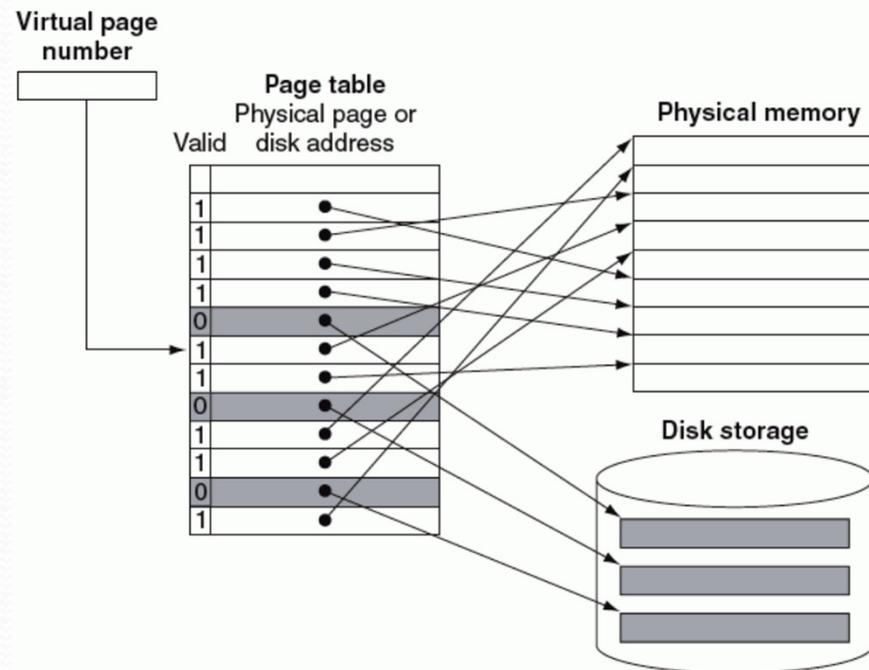
- Cada programa tiene su propia tabla de páginas.
- Varios programas pueden tener el mismo espacio de direcciones virtuales.
- El sistema operativo se encarga de:
 - Asignar la memoria física.
 - Actualizar las tablas de páginas para que los espacios de direcciones virtuales de los distintos programas no colisionen.

Falta de página

- Ocurre cuando el bit válido es falso.
- El sistema operativo:
 - Toma el control.
 - Interrumpe el proceso y guarda su estado.
 - Busca la página en el *área de swap* del disco duro y la carga en la memoria principal.
 - Si la memoria está llena, se necesita reemplazar una página.
 - Típicamente, la estrategia de reemplazo es LRU (o una aproximación).

Área de swap

- Área del disco reservada para todo el espacio de memoria virtual de un proceso.
- Incluye un registro de donde se guarda en disco cada página virtual.



Tamaño de la tabla de páginas

- Suponer lo siguiente:
 - Direcciones virtuales de 32 bits.
 - Número de página virtual ocupa 20 bits y el offset los otros 12 bits.
- El número de entradas en la tabla de páginas es $2^{20} = 1,048,576$.
- Si cada entrada ocupa 4 bytes, la tabla ocupa 4 MB.
- Si hay cientos de procesos, la memoria puede ser insuficiente.
- Es deseable reducir el tamaño de la tabla.

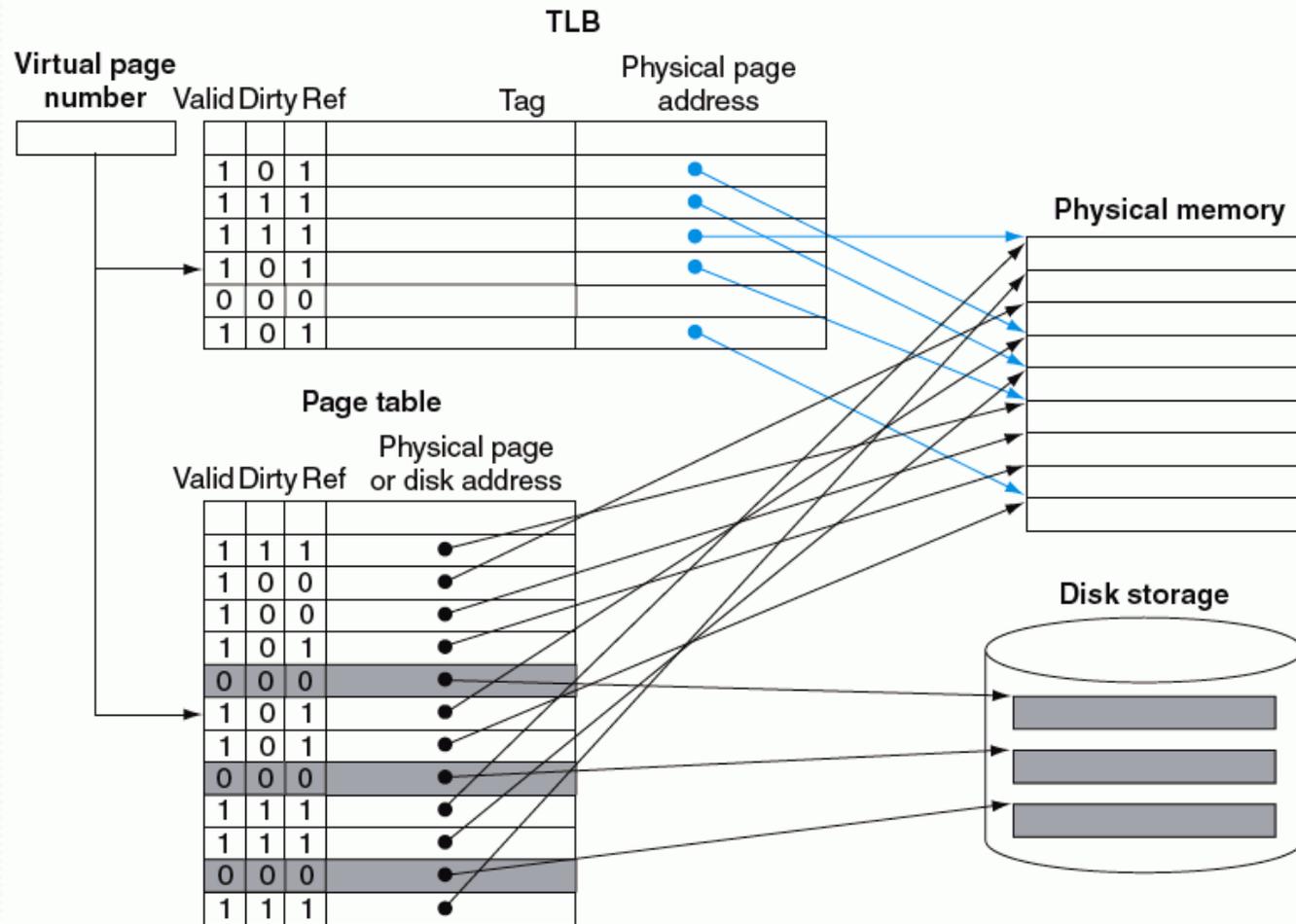
Tamaño de la tabla de páginas

- Estrategias:
 1. La tabla crece según el proceso consume memoria (tabla dinámica).
- No es práctico en los esquemas modernos de usar un área dinámica de pila y otra de heap.
- 2. Usar dos tablas dinámicas, una para la pila y otra para el heap.
- Las tablas crecen en direcciones opuestas.
- Usado en varias arquitecturas incluyendo MIPS.

TLB

- Aunque las tablas de páginas residan en memoria principal, cada acceso toma el doble de tiempo:
 - Un acceso para obtener la dirección física.
 - Un segundo acceso para obtener los datos.
- Para mejorar el rendimiento se aprovecha el concepto de “locality”.
- Si una traducción de una dirección virtual se acaba de usar, es posible que se vuelva a usar.
- Se usa un caché especial de traducciones llamado TLB (translation-lookaside buffer).

TLB



Explicación

- La TLB actúa como caché para las entradas que mapean solo a páginas físicas.
- La TLB contiene un subconjunto de los mapeos de página virtual a página física que están en la tabla de páginas.
- Como el TLB es un caché tiene campo de etiqueta.
- Si la página no está en el TLB se busca en la tabla de páginas.

Explicación

- Se busca la página virtual en la tabla de páginas y sucede una de dos cosas:
 - a) Regresa el número de página física de la página (que se usa para construir una entrada en la TLB).
 - b) Indica que la página reside en disco, en cuyo caso se genera una falta de página.
- La tabla de páginas no tiene campo de etiqueta porque **no** es un cache.

Valores típicos de TLB

- Tamaño: 16 – 512 bloques.
- Tamaño de bloque: 4 – 32 bytes.
- Hit time: 0.5 – 1 ciclo de reloj.
- Castigo por falla: 10 – 1000 ciclos de reloj.
- Tasa de fallas: 0.01% – 2%.
- Estrategia de escritura: write-back.
- Organización típica: fully associative con reemplazo aleatorio de bloques.

Comparación

Characteristic	ARM Cortex-A8	Intel Core i7
Virtual address	32 bits	48 bits
Physical address	32 bits	44 bits
Page size	Variable: 4, 16, 64 KIB, 1, 16 MiB	Variable: 4 KIB, 2/4 MiB
TLB organization	<p>1 TLB for instructions and 1 TLB for data</p> <p>Both TLBs are fully associative, with 32 entries, round robin replacement</p> <p>TLB misses handled in hardware</p>	<p>1 TLB for instructions and 1 TLB for data per core</p> <p>Both L1 TLBs are four-way set associative, LRU replacement</p> <p>L1 I-TLB has 128 entries for small pages, 7 per thread for large pages</p> <p>L1 D-TLB has 64 entries for small pages, 32 for large pages</p> <p>The L2 TLB is four-way set associative, LRU replacement</p> <p>The L2 TLB has 512 entries</p> <p>TLB misses handled in hardware</p>

Jerarquía de memoria

- La memoria virtual, el TLB y los cachés forman una jerarquía.
- Un dato no puede estar en el caché sin estar en la memoria principal.
- El sistema operativo mantiene la jerarquía borrando una página del caché si decide migrar la página de la memoria principal al disco duro.
- Al mismo tiempo, modifica el TLB y la tabla de página de modo que una referencia a un dato en la página produce una falta de página.

Combinaciones

- Posibles combinaciones de eventos en el TLB, memoria virtual y caché.

TLB	Page table	Cache	Possible? If so, under what circumstance?
hit	hit	miss	Possible, although the page table is never really checked if TLB hits.
miss	hit	hit	TLB misses, but entry found in page table; after retry, data is found in cache.
miss	hit	miss	TLB misses, but entry found in page table; after retry, data misses in cache.
miss	miss	miss	TLB misses and is followed by a page fault; after retry, data must miss in cache.
hit	miss	miss	Impossible: cannot have a translation in TLB if page is not present in memory.
hit	miss	hit	Impossible: cannot have a translation in TLB if page is not present in memory.
miss	miss	hit	Impossible: data cannot be allowed in cache if the page is not in memory.

Conclusión

- Memoria virtual es el nivel de memoria que actúa como caché entre la memoria principal y el disco.
- Permite a un programa expandir su espacio de direcciones mas allá de los límites de la memoria principal.
- Permite compartir la memoria principal entre varios procesos activos.
- Para apoyar la compartición, la memoria virtual debe ofrecer mecanismos para protección de la memoria.

Conclusión

- El principal problema es el alto costo de las faltas de página.
- Técnicas para reducir la tasa de faltas:
 - Usar páginas grandes para tomar ventaja del locality espacial.
 - El mapeo entre direcciones virtuales y físicas es fully associative. Una página virtual puede estar donde sea en la memoria principal.
 - El sistema operativo usa técnicas, como LRU (o su aproximación) para escoger la página a reemplazar.

Conclusión

- Las escrituras son caras. Se usa write-back y un bit sucio para evitar escribir páginas que no cambiaron.
- Para asegurar que los procesos estén protegidos unos de otros, el sistema operativo es el único que puede alterar las tablas de páginas.
- La tabla de páginas incluye un bit de acceso de escritura para permitir que otro proceso escriba en alguna página.
- El TLB actúa como caché de la tabla de páginas.

Conclusión

- Las 4 preguntas para la memoria virtual:
 1. ¿Dónde se puede poner una página?
En cualquier parte (fully associative).
 2. ¿Cómo se encuentra una página?
Buscando en toda la memoria con la ayuda del TLB y la tabla de páginas.
 3. ¿Qué página se reemplaza en una falla?
Típicamente LRU o aleatorio.
 4. ¿Qué pasa con las escrituras?
Write-back.