

Memoria virtual

Introducción

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

Definiciones

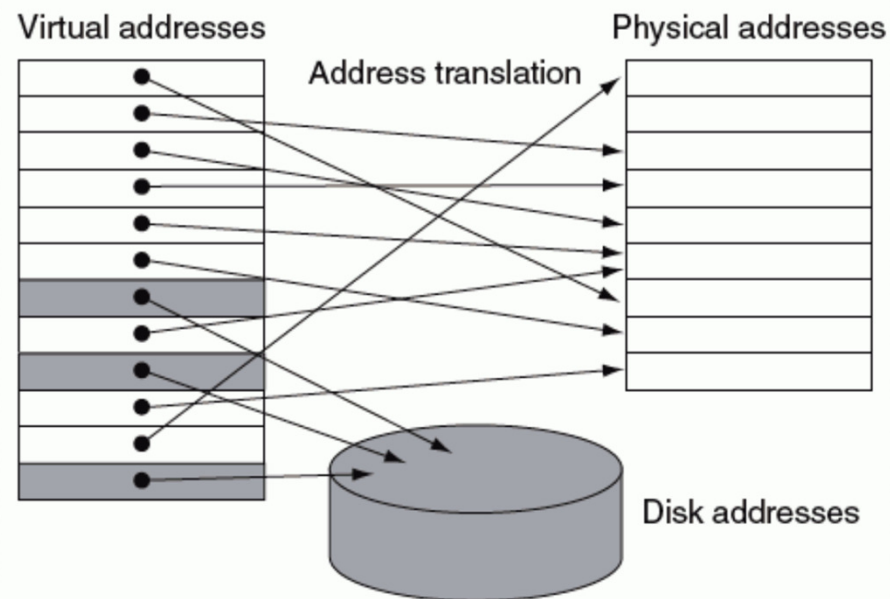
- **Dirección virtual.** Dirección en el programa.
- **Dirección física.** Dirección en la memoria principal.
- La memoria virtual incluye mecanismos para traducir direcciones virtuales a direcciones físicas.
- **Protección.** Mecanismos para asegurar que múltiples procesos que compartan la CPU, memoria 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”.

Mapeo de direcciones

- Es la traducción de una dirección virtual a una dirección física.



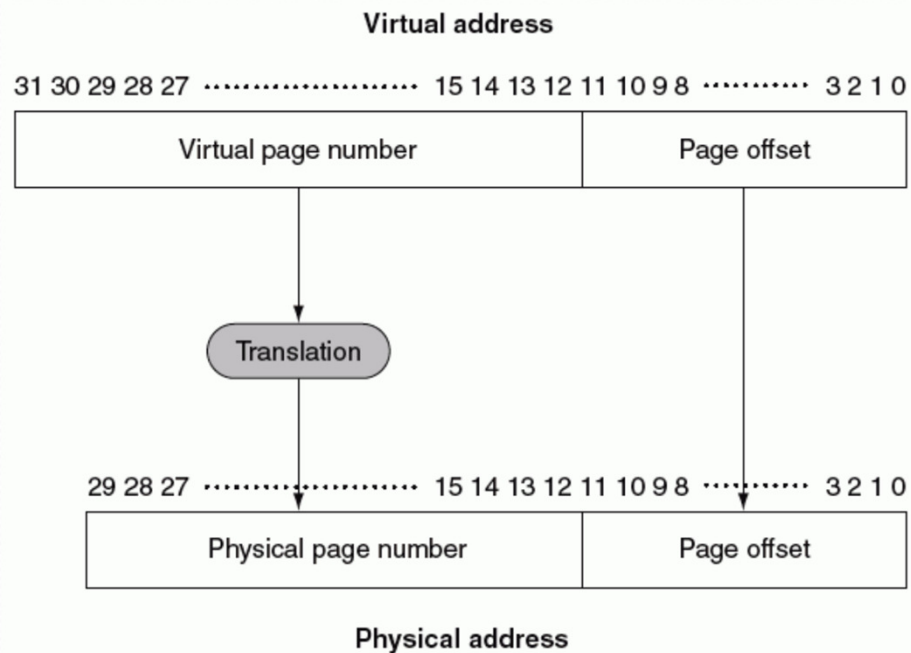
Fuente: COD:HSI₁ p. 420

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.



Fuente: COD:HSI₁ p. 430

Direcciones virtuales y físicas

- El número de bits del offset es el logaritmo base dos del tamaño de página en bytes.
- El resto se usa para el número de página virtual.
- Otra opción para obtener el número de página virtual V :
 - $V = A \text{ DIV } K$
 - A es la dirección virtual en bytes.
 - K es el tamaño de página en bytes.
 - DIV es la división entera.

Direcciones virtuales y físicas

- 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.

Consideraciones de diseño

- Las páginas deben ser suficientemente grandes para amortizar el costo de acceso.
- El tamaño de página es 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).

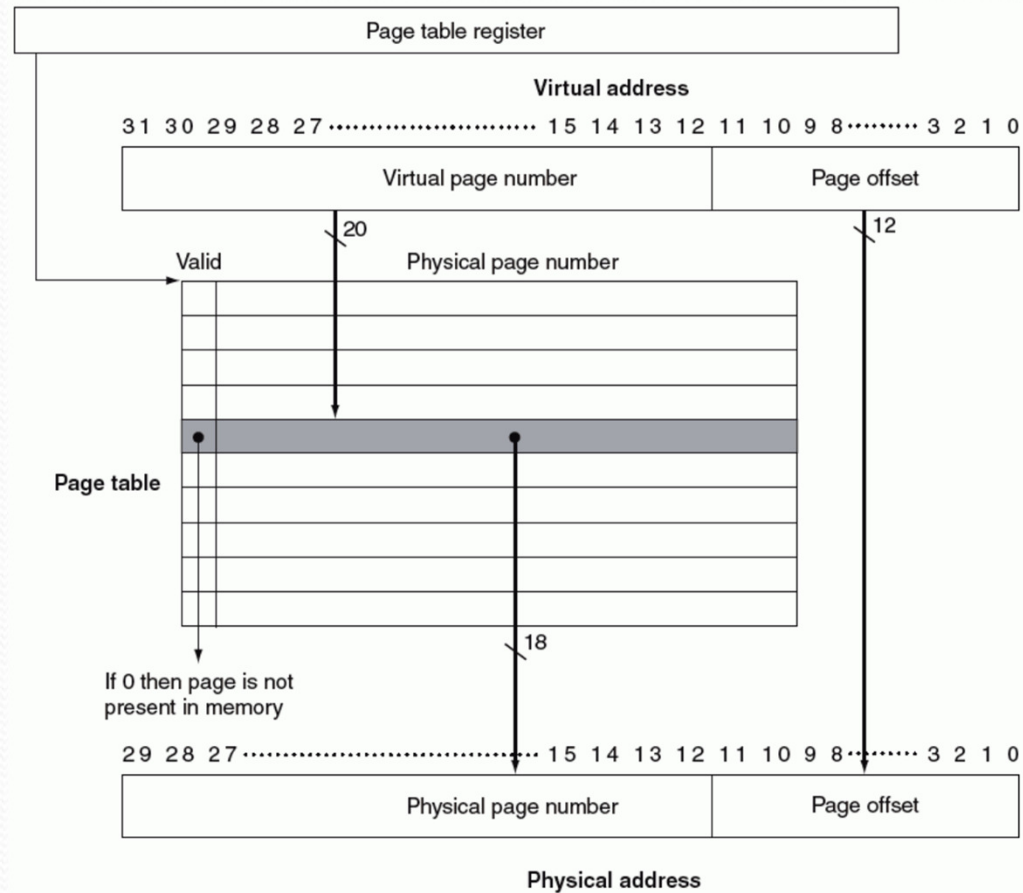
Consideraciones de diseño

- 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.
- El procesador usa un registro para apuntar a la tabla de páginas del programa que está corriendo.

Tabla de páginas



Fuente: COD:HSI, p. 433

Explicación

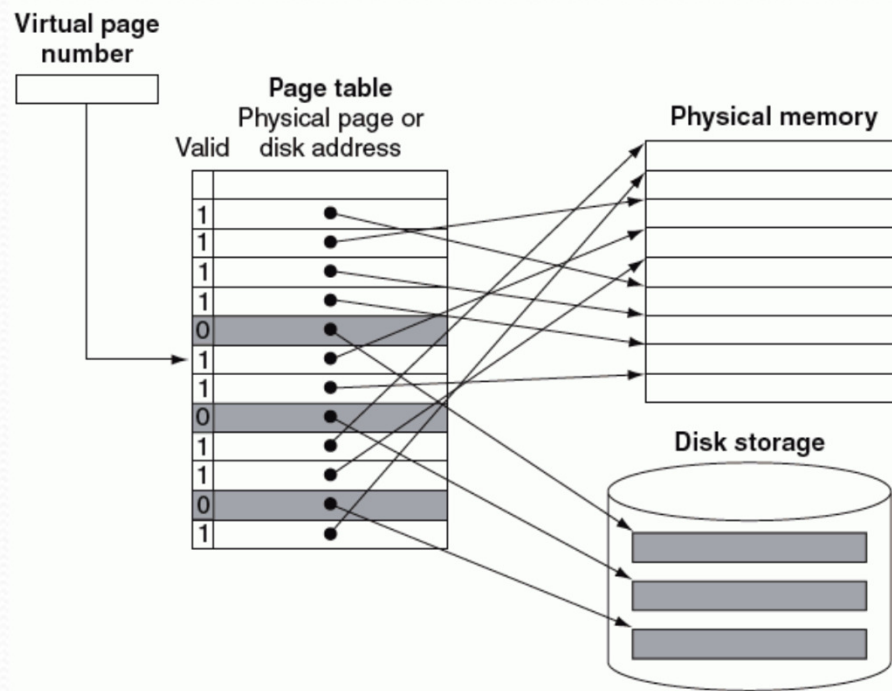
- El registro de tabla de páginas apunta al comienzo de la tabla de páginas.
- El tamaño de página es $2^{12} = 4$ KB.
- El espacio de direcciones virtuales es $2^{32} = 4$ GB.
- El espacio de direcciones físicas es $2^{30} = 1$ GB.
- El número de entradas en la tabla de páginas es $2^{20} = 1,048,576$.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

Falta de página

- Ocurre cuando la página no está en la memoria.
- 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 la memoria virtual de un proceso.



Fuente: COD:HSI, p. 435

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.

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 máximo de entradas en la tabla de páginas es $2^{20} = 1,048,576$.
- Si cada entrada ocupa 4 bytes, la tabla ocupa $2^{20} \times 4$ bytes = 2^{22} bytes = 4 MB.
- Si hay cientos de procesos, la memoria puede ser insuficiente.

Tabla de páginas

- Se utiliza una tabla dinámica que crece según el programa consume memoria.

Estrategias

1. Una sola tabla dinámica.
 - No es práctico en los esquemas modernos de usar un segmento de pila y otro de heap.
2. Dos tablas: una para la pila y otra para el heap.
 - Las tablas crecen en direcciones opuestas.
 - Usado en las arquitecturas modernas.

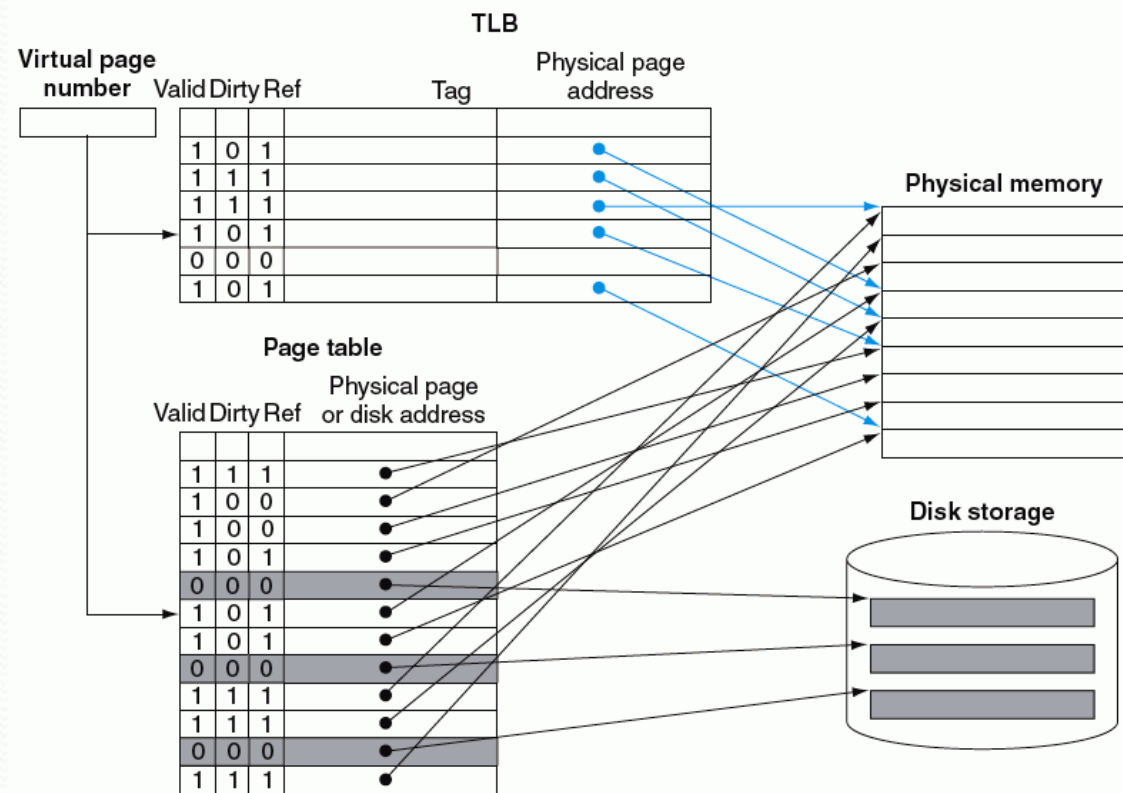
TLB

- Las tablas de páginas residen en la memoria principal.
- Cada acceso toma el doble de tiempo:
 - Un primer acceso para obtener la dirección física.
 - Un segundo acceso para obtener el item.
- Para mejorar el rendimiento se aprovecha el concepto de “locality”.
- Si una traducción de una dirección virtual a una física se acaba de usar, es posible que se vuelva a usar.

TLB

- Se usa un caché especial de traducciones llamado TLB (translation-lookaside buffer).

TLB



Fuente: COD:HSI, p. 438

Explicación

- La TLB es el caché de la tabla de páginas.
- La TLB contiene los últimos mapeos de página virtual a página física que se han realizado.
- Como el TLB es un caché tiene campo de etiqueta.
- Si el número de página virtual está en el TLB, se regresa el número de página física. Se prende el bit de acceso (ref). Se prende el bit sucio (dirty) si es una escritura.
- Si el número de página virtual no está en el TLB se busca en la tabla de páginas.

Explicación

- Después de la búsqueda en la tabla de páginas sucede una de dos cosas:
 - a) Se regresa el número de página física de la página (la traducción se carga en la TLB).
 - b) Se indica que la página reside en disco, en cuyo caso se genera una falta de página.

Valores típicos de TLB

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

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	1 TLB for instructions and 1 TLB for data Both TLBs are fully associative, with 32 entries, round robin replacement TLB misses handled in hardware	1 TLB for instructions and 1 TLB for data per core Both L1 TLBs are four-way set associative, LRU replacement L1 I-TLB has 128 entries for small pages, 7 per thread for large pages L1 D-TLB has 64 entries for small pages, 32 for large pages The L2 TLB is four-way set associative, LRU replacement The L2 TLB has 512 entries TLB misses handled in hardware

Fuente: COD:HSI, p. 471

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.

Fuente: COD:HSI, p. 438

Resumen

- 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, el sistema operativo ofrece mecanismos para protección de la memoria.

Resumen

- El principal problema de la memoria virtual 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.
 - La memoria virtual es fully associative. Una página virtual puede estar donde sea en la memoria principal.
- Para reemplazar páginas típicamente se utiliza LRU o una aproximación.

Resumen

- Las escrituras son caras. La memoria virtual 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 permiso de escritura para indicar si otro proceso puede escribir en una página.

Resumen

- El TLB es el caché de la tabla de páginas.
- Por lo general, el TLB es fully associative con reemplazo aleatorio.