

Tema 6: Jerarquía de memoria — Memoria virtual

Febrero de 2010

- 1 Introducción
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

Agenda

- 1 **Introducción**
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

Memoria virtual

La memoria principal puede actuar como una «caché» para el almacenamiento secundario. Razones:

1. Eliminar el inconveniente de tener un espacio de memoria principal pequeño y limitado:

- Sin el sistema de memoria virtual: los programadores dividían los programas en partes mutuamente exclusivas (*overlays*) cargadas en memoria o salvadas a disco bajo el control del programa.
- Con el sistema de memoria virtual: un programa manejará un espacio de direcciones de una memoria virtual como si se tratase de una gran memoria principal. Sólo una parte de ella estará en la memoria principal (*memoria física*).

Memoria virtual

2. Permitir la compartición eficiente y sin peligros de memoria entre múltiples programas:

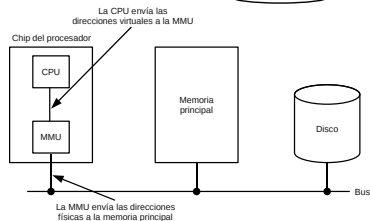
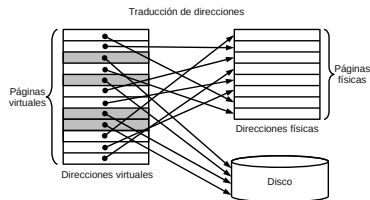
- Varios programas se están ejecutando al mismo tiempo.
- Sólo una parte de esa memoria se usa activamente en un instante de tiempo.
- La memoria principal sólo tiene que contener las partes activas de los programas en ejecución.
- Hay que asegurar que un programa sólo pueda leer y escribir las partes de la memoria que tiene asignadas.

Agenda

- 1 **Introducción**
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

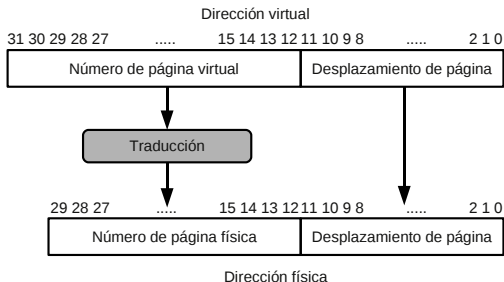
Conceptos de memoria virtual

- **Página:** bloque de memoria virtual.
- **Fallo de página:** fallo de memoria virtual.
- **Traducción de direcciones:**
 - La CPU genera **direcciones virtuales**.
 - La **MMU** las traduce a **direcciones físicas**.
 - A memoria se accede con direcciones físicas.
- Memoria virtual y memoria física dividida en páginas.
- Es posible que una página virtual no esté en la memoria principal y que solamente esté en disco.



Conceptos de memoria virtual

- Ejemplo:



- Campo «desplazamiento dentro de la página» de 12 bits → Tamaño de la página es 2^{12} bytes = 4 KB.
- Campo «número de página física» de 18 bits → 2^{18} páginas físicas.
- La memoria principal es de 2^{30} bytes = 1 GB.
- La memoria virtual es de 2^{32} bytes = 4 GB.

Agenda

- 1 **Introducción**
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

Consideraciones de diseño de un sistema de memoria

Gran coste de los fallos de página debido a la gran diferencia de velocidades de la memoria principal y la secundaria:

- Páginas grandes para amortizar el tiempo de acceso.
- Esquema totalmente asociativo → Reducir al máximo la tasa de fallos.
- Los fallos de página se pueden gestionar con software porque:
 - La sobrecarga de usar software en lugar de hardware es poca comparada con el tiempo de acceso a disco.
 - Se pueden usar algoritmos más inteligentes → Reducir al máximo la tasa de fallos.
- Política de postescritura: en escritura directa cada escritura sería muy lenta.

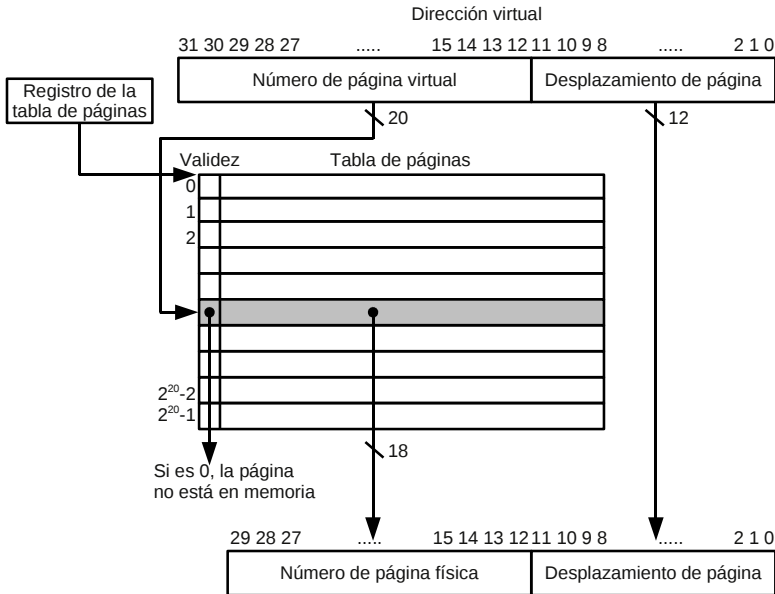
Agenda

- 1 Introducción
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

Tabla de páginas

- **Tabla de páginas:**
 - Reside en memoria.
 - Se indexa con el número de página virtual.
 - Contiene el correspondiente número de página física.
- Cada programa tiene su propia tabla de páginas.
- **Registro de la tabla de páginas:** apunta al inicio de la tabla de páginas del programa que esté en funcionamiento.
- Fallo de página → **Excepción por fallo de página:** el control pasa al sistema operativo, que ha de encontrar la página solicitada en la memoria secundaria y decidir dónde colocarla en la memoria principal.

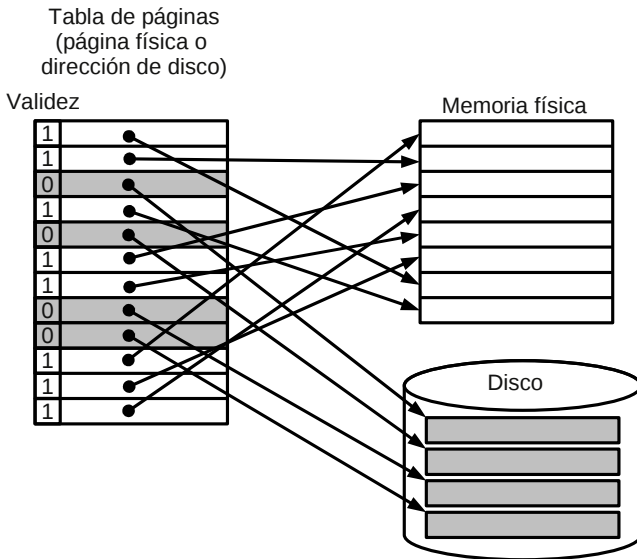
Tabla de páginas



Agenda

- 1 Introducción
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

Tabla de páginas unificada



Tratamiento de los fallos de página

- Política de reemplazo: **LRU**. LRU perfecto es demasiado caro → Pseudo-LRU con un **bit de uso**:
 - Se activa cuando se accede a la página.
 - El sistema operativo lo apaga periódicamente.
 - El sistema operativo puede seleccionar una página entre las que tengan el bit de uso apagado.
- Política de escritura: **postescritura**
 - Las escrituras individuales se realizan en la página que hay en la memoria principal y se copia la página a memoria secundaria cuando ésta se reemplaza.
 - **Bit de modificación**: para conocer si una página ha sido modificada desde que se leyó de memoria secundaria.

Agenda

- 1 Introducción
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

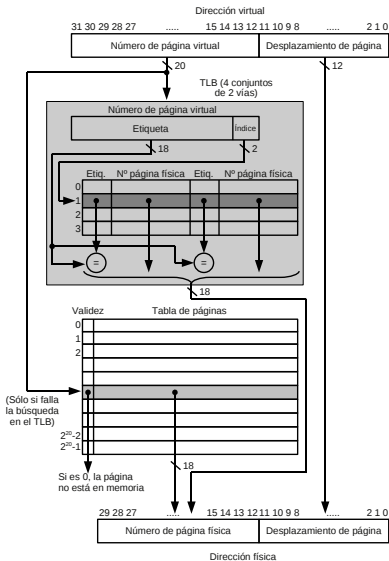
TLB (Translation–Lookaside Buffer)

- Cada lectura/escritura a memoria provoca realmente **dos accesos a memoria**:
 - 1 Acceso a la tabla de páginas para traducir la dirección virtual a dirección física.
 - 2 Con la dirección física obtenida, acceder físicamente a la palabra solicitada.
- Clave para solucionarlo: la **localidad de las referencias a la tabla de páginas**. Cuando un programa accede a una palabra que pertenece a una página virtual X, es muy probable que próximamente acceda a palabras de direcciones cercanas y que, por tanto, pertenezcan a la misma página X.
- **Solución** → **TLB**: caché especial que guarda las traducciones «número de página virtual» → «número de página física» más frecuentemente usadas.

TLB (Translation–Lookaside Buffer)

- El TLB es una caché que trata la tabla de páginas como una mini–memoria principal de bloques monopalabra.
- El índice de la tabla de páginas (número de página virtual) viene a ser como la dirección de la palabra y el contenido de la posición de la tabla (básicamente, el número de página física) viene a ser el contenido de la palabra.
- El índice de la tabla de páginas (= número de página virtual) será usado para acceder al TLB:
 - Los bits menos significativos como índice.
 - El resto como etiqueta.

TLB (Translation-Lookaside Buffer)



TLB (Translation–Lookaside Buffer)

En cada referencia a memoria:

- 1 La MMU descompone la dirección virtual en NPV + desplaza.
- 2 Se busca el número de página virtual en el TLB.
 - Si hay **acierto de TLB**, se salta al paso 6.
- 3 Como hay un fallo de TLB, hay que ir a la tabla de páginas.
 - Si bit de validez está encendido, hay **acierto de página**. Se salta al paso 5.
- 4 Como hay fallo de página, entra a funcionar el S.O.:
 - Coloca la página solicitada en la memoria principal.
 - Actualiza la entrada de la tabla de páginas (el n^o de página física y el bit de validez).
- 5 El contenido de la entrada de la tabla de páginas va al TLB. Se actualiza la etiqueta en el TLB.
- 6 Usando el TLB: n^o de página virtual → n^o de página física.
 - Se actualizan los bits de control de esa página.

TLB (Translation–Lookaside Buffer)

- Los fallos de TLB son más frecuentes que los fallos de páginas, por tener el TLB menos entradas que la tabla de páginas.
 - Consecuencia: muchos sistemas escogen **aleatoriamente** la entrada del TLB a reemplazar en vez de utilizar un esquema LRU.
- La tasa de fallos del TLB es pequeña (aunque mayor que la de la tabla de páginas). Consecuencia: el TLB suele emplear una estrategia de **postescritura**. Esto conlleva que:
 - En el paso 6 del algoritmo anterior, la actualización de los bits de control se realice solamente en el TLB.
 - Cuando una entrada en el TLB va a ser sustituida, toda su información debe copiarse en la tabla de páginas.

Agenda

- 1 Introducción
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

Implementación de la protección con mem. virtual

- El mecanismo de protección tiene que asegurar que aunque muchos procesos estén compartiendo la misma memoria principal, un proceso no pueda leer/escribir en el espacio de direcciones de otro proceso.
- Necesario que un proceso de usuario no tenga capacidad para modificar por sí mismo los valores de su tabla de páginas, siendo el sistema operativo el único proceso capaz de hacerlo.
- Cuando el sistema operativo cambia del proceso en ejecución P_1 al proceso en ejecución P_2 (*cambio de contexto*):
 - Cambiar el valor del registro que apunta al inicio de la tabla de páginas del proceso que está en funcionamiento.
 - Vaciar el TLB de las entradas que pertenezcan a P_1 .

Implementación de la protección con mem. virtual

● Características necesarias del hardware:

- 1 Tener de dos modos de funcionamiento en la CPU que indiquen si un proceso es un:
 - Proceso de usuario (con permisos de lectura/escritura limitados).
 - Proceso del sistema operativo (también llamado proceso del *núcleo*, proceso *supervisor* o proceso *ejecutivo*).
- 2 Impedir que la información sobre el estado en que se encuentra la CPU pueda ser modificada por un proceso de usuario. Esta información incluye:
 - El bit que indica si el modo de proceso es de usuario/supervisor.
 - El registro que apunta al inicio de la tabla de páginas.
 - El contenido de la tabla de páginas.
 - El contenido del TLB.
- 3 Disponer de mecanismos que permitan a la CPU pasar del modo usuario a modo supervisor (*syscall*) y viceversa (*retorno de excepción*).

Agenda

- 1 Introducción
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 **Un marco común para las jerarquías de memoria**
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

Un marco común para las jerarquías de memoria

Tabla: Posibles combinaciones de sucesos en el TLB, memoria virtual y caché ordenados de mayor a menor rapidez de acceso.

TLB	TP	Caché	¿Posible?	Nº Mínimo accesos MP	Nº Mínimo accesos MS
A	A	A	Lo más deseado. No se comprobaría la tabla de páginas porque hay acierto de TLB. La palabra solicitada sí está en caché.	0	0
A	A	F	Posible. No se comprobaría la tabla de páginas porque hay acierto de TLB. La palabra solicitada no está en caché; habrá que acceder a memoria principal, donde se encuentra la página a la que pertenece.	1	0

Un marco común para las jerarquías de memoria

TLB	TP	Caché	¿Posible?	Nº Mínimo accesos MP	Nº Mínimo accesos MS
F	A	A	Posible. Hay que acceder a la tabla de páginas para obtener la dirección física. La palabra solicitada sí está en caché.	1	0
F	A	F	Posible. Hay que acceder a la tabla de páginas para obtener la dirección física. La palabra solicitada no está en caché; habrá que acceder a memoria principal, donde se encuentra la página a la que pertenece, para obtener la palabra.	2	0
F	F	F	Lo menos deseado. Hay que acceder a la tabla de páginas para obtener la dirección física. La página a la que pertenece la palabra solicitada no está en memoria principal; necesario acceso a memoria secundaria para traerla. Tras ello, como la palabra solicitada no está en caché, habrá que acceder a memoria principal, donde ya se encuentra la página a la que pertenece, para obtener la palabra.	2	1

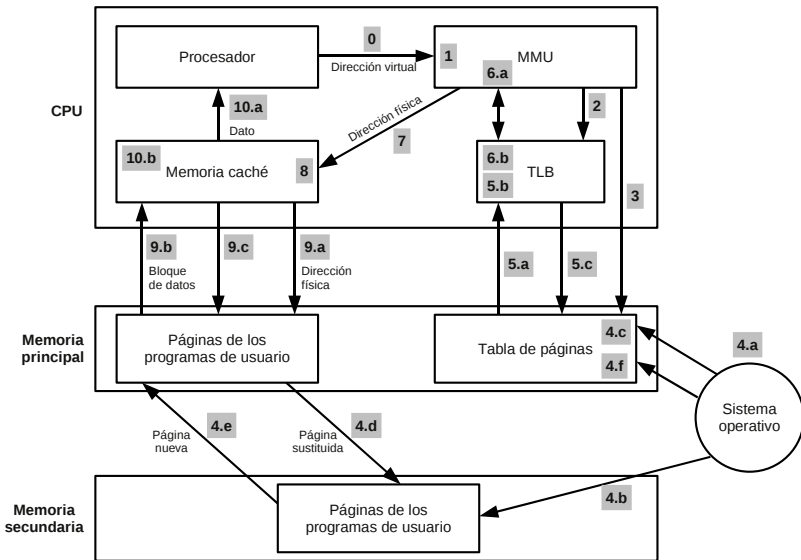
Un marco común para las jerarquías de memoria

TLB	TP	Caché	¿Posible?	Nº Mínimo accesos MP	Nº Mínimo accesos MS
A	F	X	Imposible. No se puede tener una traducción en el TLB de una página que no está en memoria principal.		
X	F	A	Imposible. No se puede tener un dato en la caché cuya página no está en memoria principal.		

Un marco común para las jerarquías de memoria

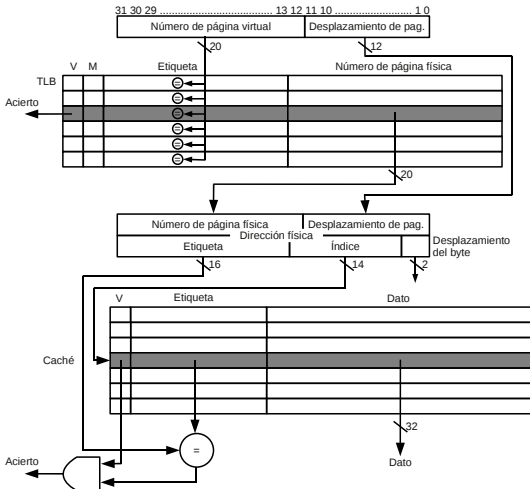
TLB	TP	Caché	¿Posible?	Nº Mínimo accesos MP	Nº Mínimo accesos MS
A	A	A	Lo más deseado. Acierto TLB → No Acceso TP (MP) Acierto caché → No Acceso MP	0	0
A	A	F	Posible. Acierto TLB → No Acceso TP (MP) Fallo caché → Acceso MP	1	0
F	A	A	Posible. Fallo TLB → Acceso TP (MP) Acierto caché → No Acceso MP	1	0
F	A	F	Posible. Fallo TLB → Acceso TP (MP) Fallo caché → Sí Acceso MP	2	0
F	F	F	Lo menos deseado. Fallo TLB → Acceso TP (MP) Fallo caché → Acceso MP, Fallo → Acceso MS	2	1
A	F	X	Imposible. No se puede tener una traducción en el TLB de una página que no está en memoria principal.		
X	F	A	Imposible. No se puede tener un dato en la caché cuya página no está en memoria principal.		

Un marco común para las jerarquías de memoria



Un marco común para las jerarquías de memoria

Ejemplo: caso más deseable (aciertos de TLB y caché).



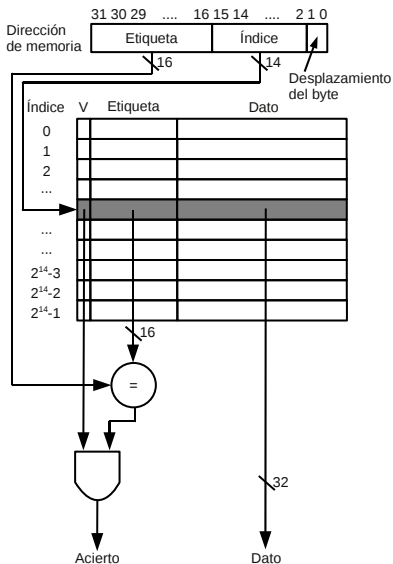
Agenda

- 1 Introducción
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

Agenda

- 1 Introducción
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

Memoria caché de la DECSTATION 3100



Agenda

- 1 Introducción
 - Conceptos generales
 - Consideraciones de diseño de un sistema de memoria virtual
- 2 La tabla de páginas
- 3 Tratamiento de los fallos de página
- 4 TLB (Translation–Lookaside Buffer)
- 5 Implementación de la protección con memoria virtual
- 6 Un marco común para las jerarquías de memoria
- 7 Jerarquía de memoria para la DECSTATION 3100
 - Memoria caché
 - Memoria virtual

Memoria virtual de la DECSTATION 3100

- Direcciones virtuales de 32 bits.
- Direcciones físicas de 32 bits.
- Páginas de 4 KB → 12 bits para el desplazamiento y 20 bits para el número de página virtual.
- TLB:
 - Totalmente asociativo de 64 entradas.
 - Compartido entre las referencias a instrucciones y a datos.
 - Entradas de 64 bits: 20 bits de etiqueta (coincide con el NPV asociado a la entrada), 20 bits de NPF, 1 bit de validez y otros bits de control.