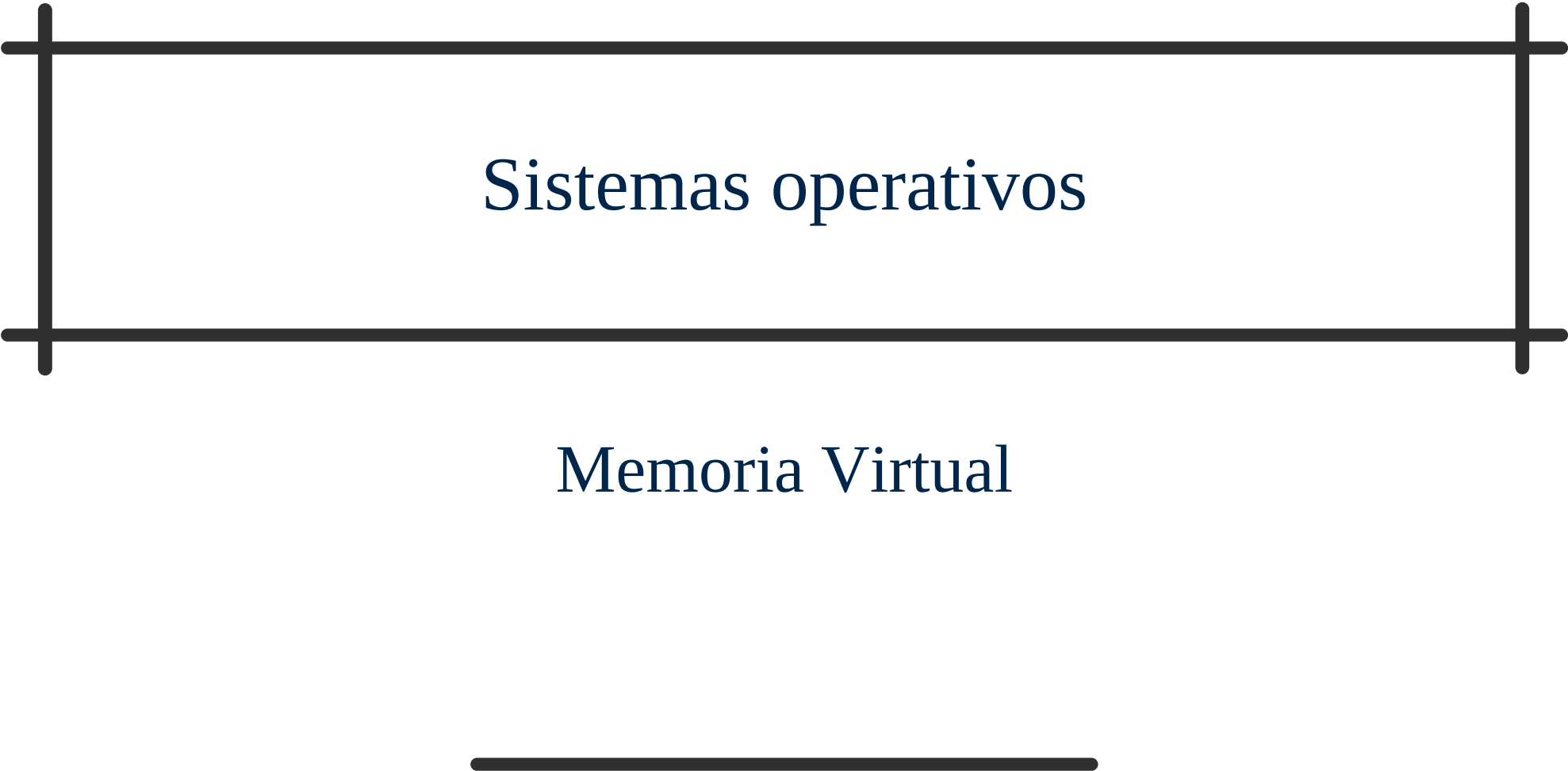




Sistemas operativos

The diagram consists of a large rectangle defined by two horizontal lines and two vertical lines. The text 'Sistemas operativos' is centered within this rectangle. Below the rectangle, the text 'Memoria Virtual' is centered. At the bottom of the diagram, there is a single horizontal line.

Memoria Virtual

Memoria Virtual

■ *Características principales de la Paginación y la Segmentación*

- Como las direcciones lógicas se traducen a direcciones físicas durante la ejecución, se puede cargar y descargar de la memoria principal un proceso de forma que en momentos diferentes ocupe posiciones diferentes en la memoria.
- Un proceso puede dividirse en varios fragmentos (paginas o segmentos) y no es necesario que estos fragmentos estén contiguos en memoria durante la ejecución, lo que es posible gracias a la traducción dinámica de direcciones en tiempo de ejecución y las tablas de páginas o segmentos.

Memoria Virtual

- # La aportación fundamental de la Gestión de Memoria Virtual es la de hacer uso del disco como extensión de la memoria principal.
- # Los procesos solo se ejecutan en la memoria principal o **real**, pero el programador tiene a su disposición una zona de disco gestionada por el sistema de forma que percibe una memoria mucho mayor para sus programas.
- # Esta es la memoria **virtual**.
- # Con la memoria virtual, no es necesario que todas las páginas o segmentos de un proceso residan en memoria real para su ejecución.
- # El conjunto de fragmentos (paginas o segmentos) que en un momento dado se encuentran en memoria real se llama **conjunto residente**.

Memoria Virtual

- ✦ Mientras todas las referencias del proceso se realicen al conjunto residente todo irá bien. El procesador comprobará que la dirección real pertenece al conjunto residente a través de la tabla de páginas o segmentos.
- ✦ Si la referencia sale del conjunto residente, el procesador provocará una interrupción de fallo de acceso a memoria, que tendrá que atender el S.O.
- ✦ La atiende pasando a bloqueado al proceso, lanzando una operación de E/S para cargar el fragmento afectado y pone otro proceso en ejecución.
- ✦ Cuando se produzca la interrupción de fin de E/S que indica la carga del fragmento, pasa a listo al proceso. Cuando le toque seguirá su ejecución.

Memoria Virtual

⚙️ Ventajas de esta estrategia

- **Se pueden mantener más procesos en memoria real**, puesto que no se cargan completos. Esto hará que haya más procesos en Listos y por tanto se incrementa la eficiencia del procesador: Es más probable encontrar un proceso listo para ejecución.
- **Un proceso puede ser más grande que la memoria real disponible.** Los mecanismos de memoria virtual con paginación o segmentación encapsulan la tarea de cargar y descargar datos y código referenciados de forma que sea transparente para el programador, que dispone ahora de una memoria enorme a su disposición.

⚙️ Se necesitan dos ingredientes para que la estrategia funcione

- Soporte del Hardware
- Software del S.O. para gestionar el movimiento de páginas o segmentos entre memoria real y virtual.



Gestión de Memoria

Memoria Virtual

HARDWARE

PAGINACIÓN

Memoria Virtual: Paginación

Dirección virtual



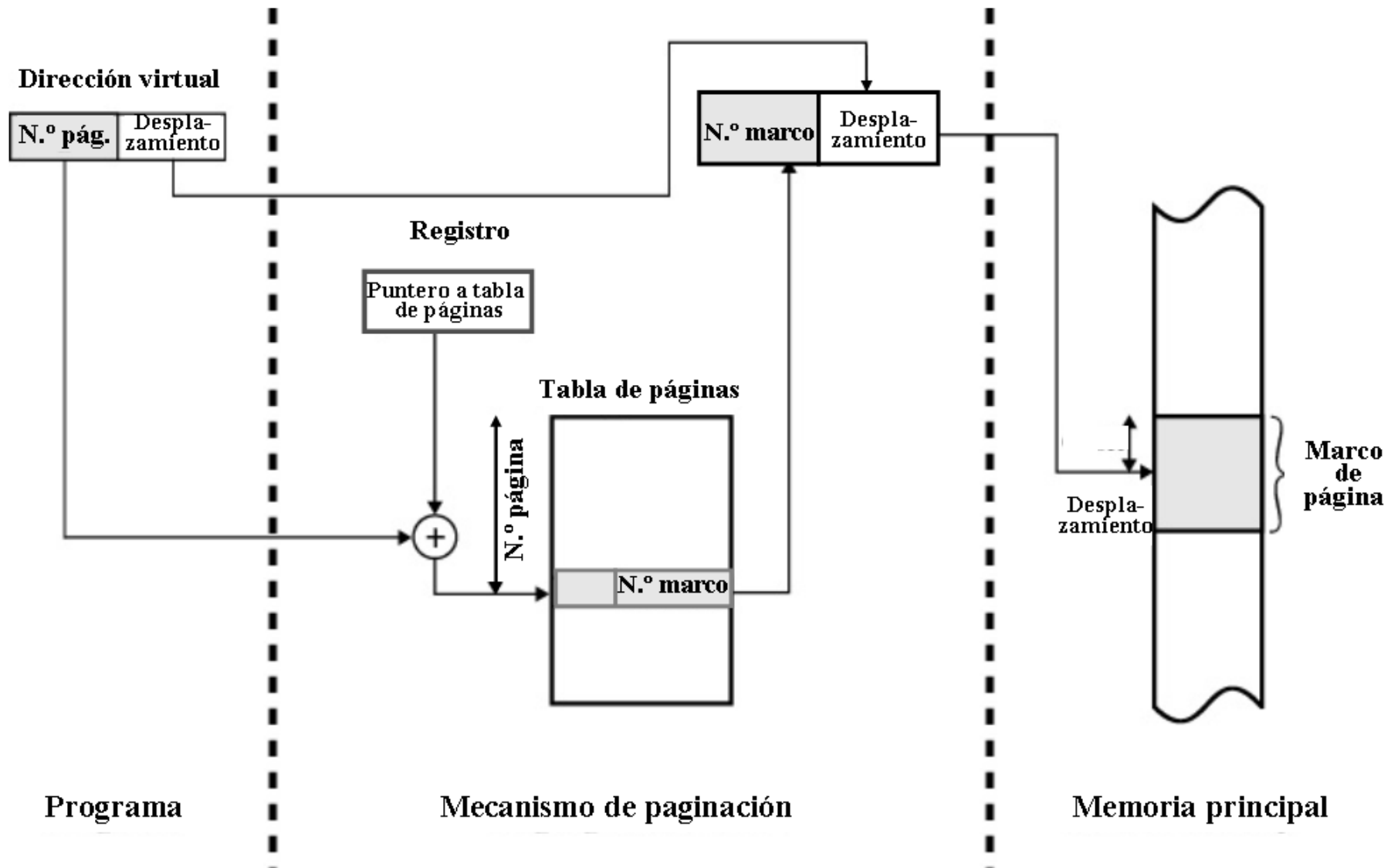
Entrada de la tabla de páginas



- Cada proceso tiene su propia tabla de página.
- Cada entrada de la tabla de páginas contiene el número de marco de la página correspondiente en la memoria principal.
- Se necesita un bit para indicar si la página correspondiente está presente en la memoria principal o no lo está.
- Otro bit de control necesario en la entrada de la tabla de páginas es el bit de modificación para indicar si el contenido de la página se ha alterado desde que la página se cargó en la memoria principal.
- Si no ha habido cambios, no es necesario escribir la página cuando sea sustituida en el marco que ocupa actualmente.

Memoria Virtual: Paginación

Traducción de direcciones en un sistema con Paginación



Memoria Virtual: Paginación

Problema del tamaño de la tabla de páginas

- Los procesos pueden ser enormes, por ejemplo 2 Gbytes en la arquitectura VAX o 4 Gbytes en Pentium.
- $2 \text{ Gbytes} = 2^{31}$
- Con páginas 512 bytes $= 2^9$
- Tendremos 2^{22} paginas, es decir 4 Mbytes por tabla de páginas
- Con 100 procesos deberíamos tener 400 Mbytes para la tabla de páginas en memoria principal.

En algunos sistemas la tabla de páginas se mantiene en memoria virtual, con lo cual sufre de paginación como el resto del proceso.

En otros sistemas se utilizan tablas de páginas jerárquicas de varios niveles.

Memoria Virtual: Paginación

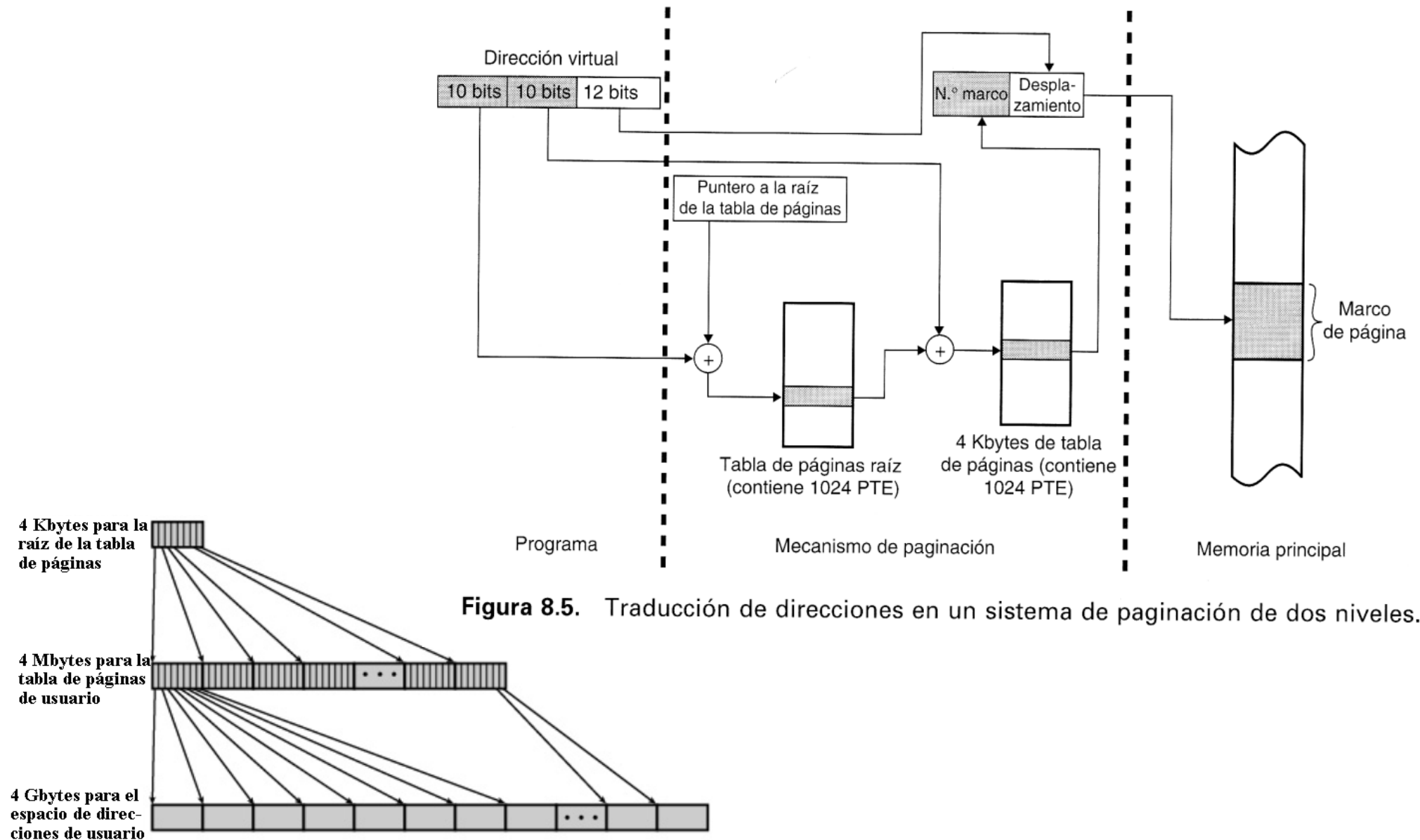


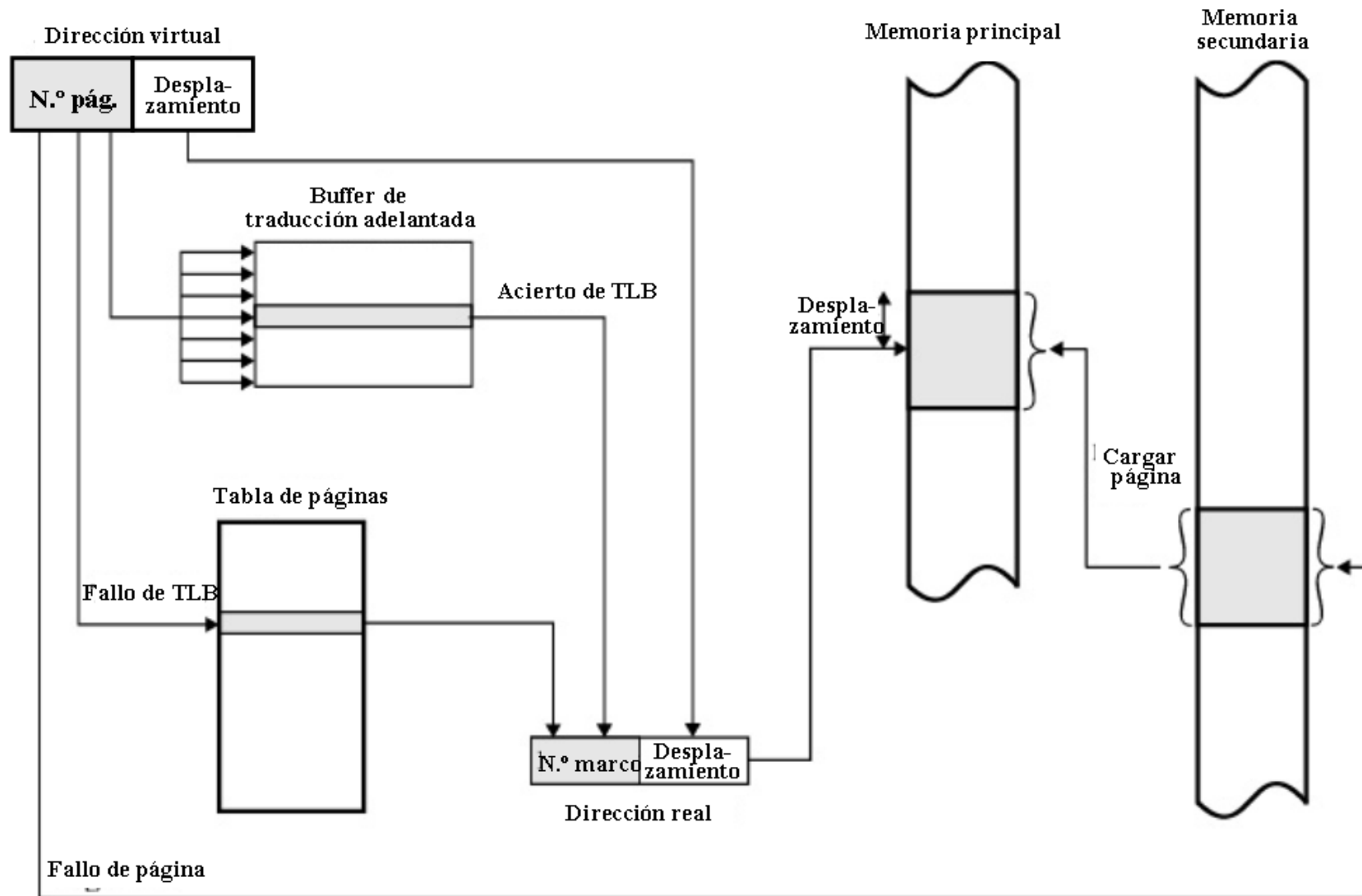
Figura 8.5. Traducción de direcciones en un sistema de paginación de dos niveles.

Memoria Virtual: Paginación

- ✚ Cada referencia a la memoria virtual puede generar dos o mas accesos a la memoria:
 - Uno para obtener la entrada de la tabla de páginas correspondiente.
 - Otro para obtener el dato deseado.
- ✚ Para solucionar este problema, los esquemas de memoria virtual hacen uso de una caché especial para las entradas de la tabla de páginas:
 - Se trata del buffer de traducción adelantada (*TLB, Translation Lookaside Buffer*).
- ✚ Contiene aquellas entradas de la tabla de páginas usadas hace menos tiempo.
- ✚ Funciona del mismo modo que una memoria caché.

Memoria Virtual: Paginación

Uso del buffer de traducción adelantada



Memoria Virtual: Paginación

Tamaño de Página

- Cuanto menor sea el tamaño de página, menor será la cantidad de fragmentación interna.
- Cuanto menor sea la página, mayor será el número de páginas que se necesitan por proceso.
- Un número mayor de páginas por proceso significa que las tablas de páginas serán mayores.
- Esto puede significar que una gran parte de las tablas de páginas de los procesos activos deben estar en la memoria virtual.
- La memoria secundaria está diseñada para transferir eficazmente los bloques de datos de mayor tamaño, de manera que es propicia para tamaños de página mayores.



Gestión de Memoria

Memoria Virtual
HARDWARE
SEGMENTACION

Memoria Virtual: Segmentación

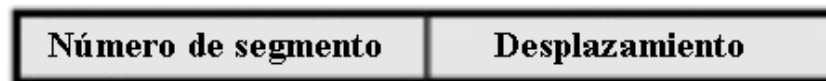
- # Los segmentos pueden ser de distintos tamaños, incluso de forma dinámica.
- # Simplifica la gestión de estructuras de datos crecientes. La estructura de datos se puede ubicar en su propio segmento de tamaño creciente dinámicamente.
- # Permite modificar y recompilar los programas independientemente.
- # Se presta a la compartición entre procesos. Se puede compartir código o datos.
- # Se presta a la protección. Se puede agrupar un conjunto de datos o procedimientos con acceso restringido en un segmento propio al que se le de permisos de acceso.

Memoria Virtual: Segmentación

Tablas de segmentos

- Cada entrada contiene la dirección base del segmento correspondiente en memoria principal.
- Cada entrada de la tabla de segmentos contiene la longitud del segmento.
- Se necesita un bit para indicar si el segmento correspondiente está presente en la memoria principal.
- Otro bit de control necesario es un bit de modificación que indique si el contenido del segmento correspondiente ha sido modificado desde que se cargó por última vez en la memoria principal.
- Pueden ser necesarios otros bits de control para los mecanismos de protección o compartición.

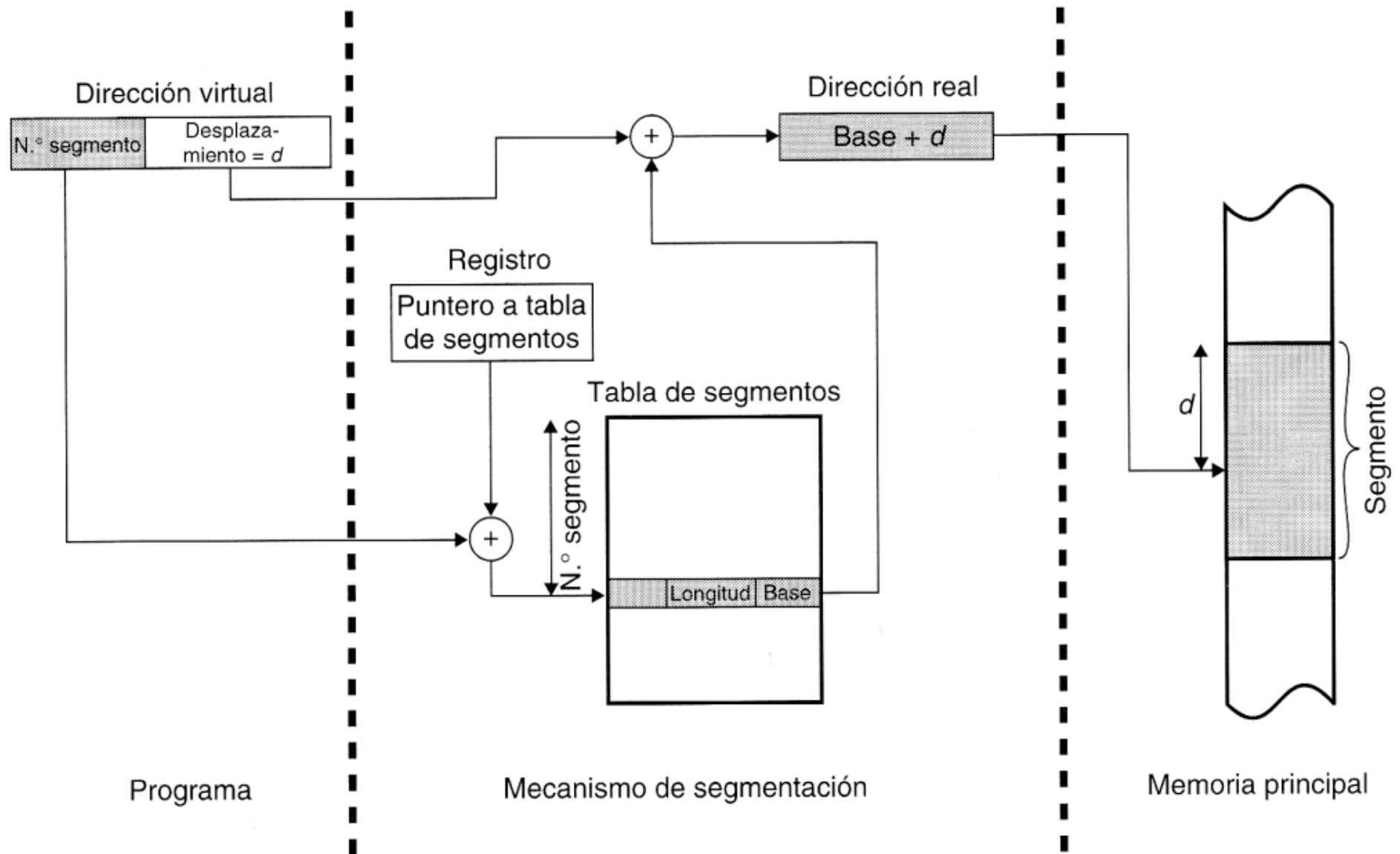
Dirección virtual



Entrada del segmento de tabla



Memoria Virtual: Segmentación





Gestión de Memoria

Memoria Virtual

HARDWARE

PAGINACION & SEGMENTACION



Memoria Virtual: Paginación & Segmentación

- # La paginación es transparente al programador.
- # La paginación elimina la fragmentación externa.
- # La segmentación es visible para el programador.
- # La segmentación permite gestionar estructuras de datos que pueden crecer, la modularidad y el soporte de la compartición y la protección.
- # Cada segmento se divide en varias páginas de tamaño fijo.

Dirección virtual

Número de segmento	Número de página	Desplazamiento
--------------------	------------------	----------------

Entrada de la tabla de segmentos

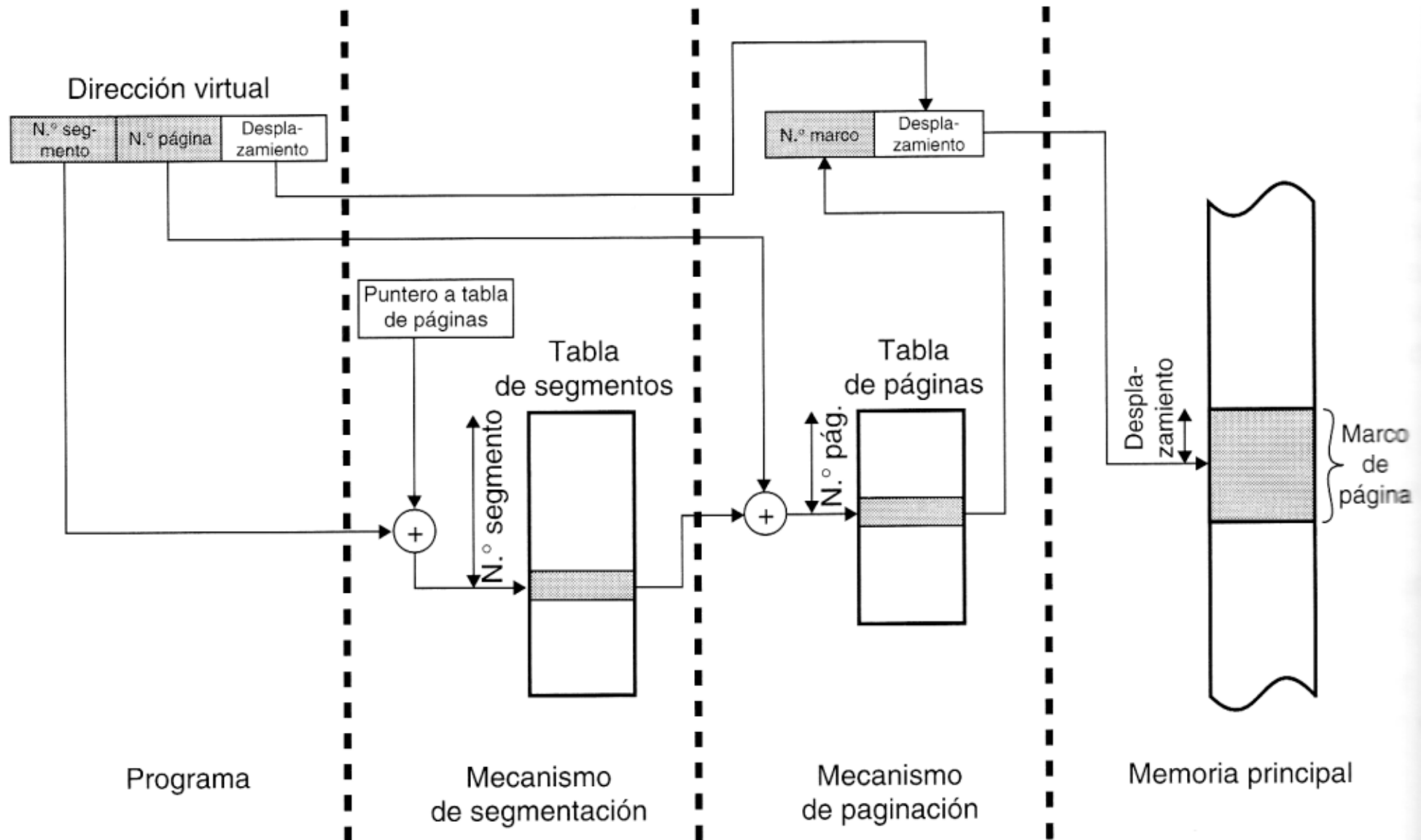
Otros bits de control	Longitud	Base de segmento
-----------------------	----------	------------------

Entrada de la tabla de páginas

P	M	Otros bits de control	Número de marco
---	---	-----------------------	-----------------

P = Bit de presencia
M = Bit de modificación

Memoria Virtual: Paginación & Segmentación





Gestión de Memoria

Memoria Virtual
SOFTWARE

Memoria Virtual: Software

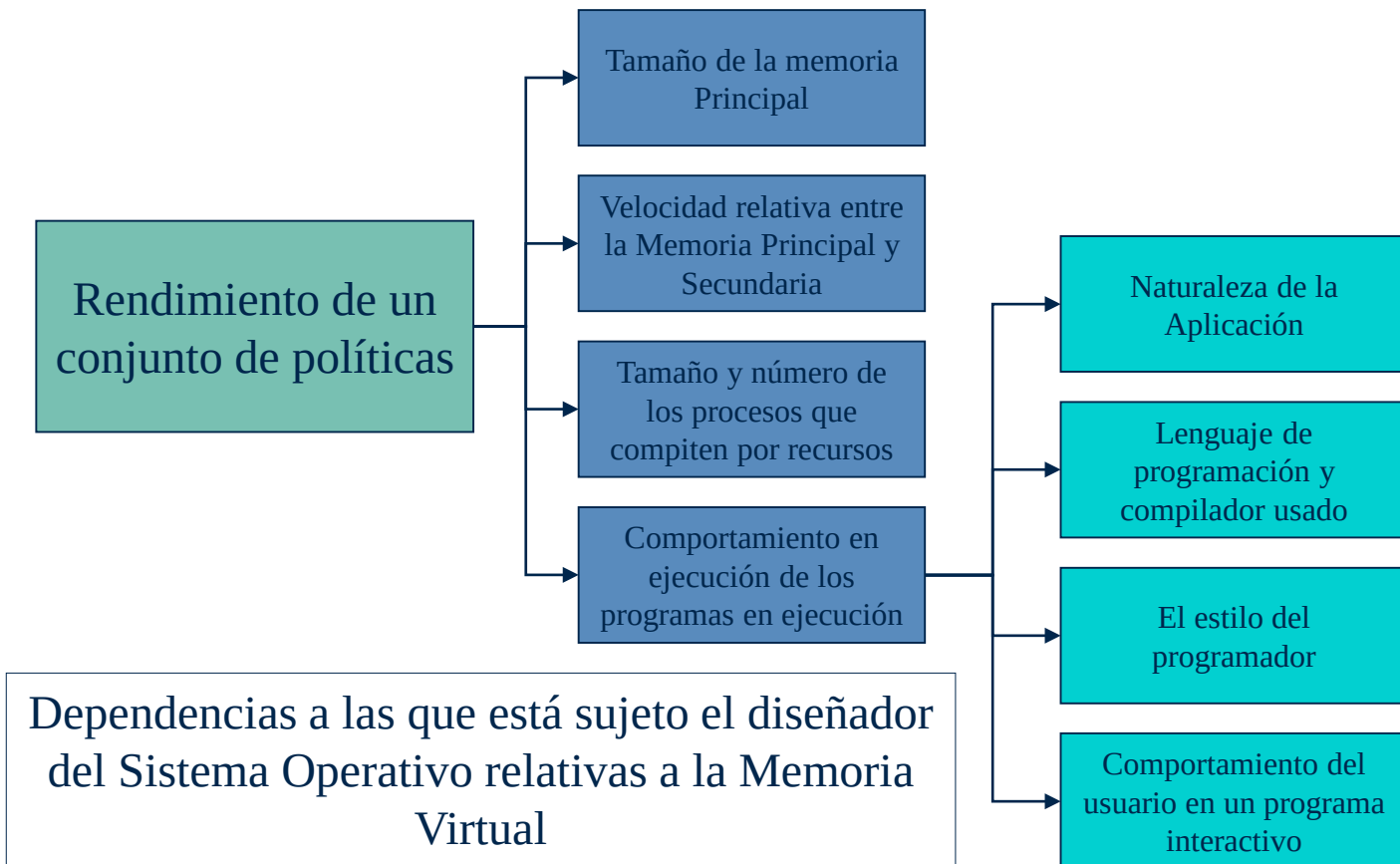
- ⌘ Políticas del sistema operativo sobre la Memoria Virtual.
- ⌘ En todos los elementos clave de diseño se busca maximizar el rendimiento minimizando el porcentaje de fallos de página.
- ⌘ Se pretende organizar las cosas para que la probabilidad de que un proceso en ejecución haga referencia a una palabra de una página ausente sea mínima.

- ⌘ Políticas de Lectura
 - Por Demanda
 - Paginación previa
- ⌘ Políticas de Ubicación
- ⌘ Políticas de Reemplazo
 - Algoritmos básicos
 - Óptimo
 - LRU
 - FIFO
 - Reloj

- ⌘ Gestión del Conjunto Residente
 - Tamaño del conjunto residente
 - Fijo
 - Variable
 - Alcance del reemplazo
 - Global
 - Local
- ⌘ Políticas de Vaciado
 - Por demanda
 - Vaciado Previo
- ⌘ Control de Carga
 - Grado de Multiprogramación

Memoria Virtual: Software

- ⚡ No existe una solución que funcione mejor que otra. El nivel de elementos de los que depende cada solución es tan grande que en función de estas dependencias una estrategia puede ser buena o mala.
- ⚡ La tarea del gestor de memoria es muy compleja por el elevado número de dependencias que condicionan los algoritmos.



Los sistemas pequeños se diseñan eligiendo un conjunto de políticas que parezcan buenas en la mayoría de los casos

En grandes computadoras (mainframes) el sistema operativo se diseña con herramientas de supervisión y control que permiten al administrador configurar o ajustar (tunning) el sistema para obtener buenos resultados en función de las características de cada instalación

Memoria Virtual: Software

■ Políticas de lectura

- Están relacionadas con la decisión de cuándo se debe cargar una página en la memoria principal.
- Con la **paginación por demanda**, se trae una página a la memoria principal sólo cuando se hace referencia a una posición en dicha página:
 - Cuando un proceso se ejecute por primera vez, se producirán muchos fallos de página, a medida que se cargan páginas en memoria el número de fallos de pagina disminuye.
- Con la **paginación previa**, se cargan más páginas de las necesarias:
 - Es más eficiente traer a la memoria un número de páginas adyacentes.
 - Esto está motivado por la forma de acceso por bloques a la memoria secundaria.
 - Si hay suerte y las referencias son a las páginas anticipadas, la política será efectiva.
 - Puede llegar a ser contraproducente si se descargan páginas que estaban en memoria por traer las supuestamente adyacentes, pero luego resultan ser referenciadas las descargadas.

Políticas de ubicación

- Determinar donde va a residir la parte de un proceso en memoria principal
- Con Segmentación pura tiene sentido y se vieron las políticas del mejor ajuste, primer ajuste, etc.
- Cuando se usa paginación es indiferente la ubicación, ya que el hardware de traducción de direcciones y el hardware de acceso a la memoria principal obtienen la misma eficacia indistintamente de la ubicación de la página

Políticas de reemplazo

- ¿Qué página se va a reemplazar de entre el conjunto de páginas consideradas?
- La página que se va a reemplazar tiene que ser la que tenga una menor posibilidad de ser referenciada en un futuro cercano.
- La mayoría de las políticas intentan predecir el comportamiento futuro en función del comportamiento pasado.
- **Bloqueo de marcos:**
 - Cuando un marco está bloqueado, la página cargada en ese marco no puede ser reemplazada.
 - La mayoría del núcleo del sistema operativo está en marcos bloqueados.
 - Estructuras de control.
 - Buffers de E/S.
 - El bloqueo se consigue asociando un bit de bloqueo a cada marco o indicándolo en la tabla de páginas actual.

■ Políticas de reemplazo. Algoritmos básicos.

■ Política óptima:

- Selecciona para reemplazar la página que tiene que esperar una mayor cantidad de tiempo hasta que se produzca la referencia siguiente.
- Es imposible de implementar porque requiere que el sistema operativo tenga un conocimiento exacto de los sucesos futuros.

■ Política de la usada menos recientemente (LRU):

- Reemplaza la página de memoria que no ha sido referenciada desde hace más tiempo.
- Debido al principio de cercanía, ésta sería la página con menor probabilidad de ser referenciada en un futuro cercano.
- Una solución sería etiquetar cada página con el momento de su última referencia.

Memoria Virtual: Software

Políticas de reemplazo. Algoritmos básicos.

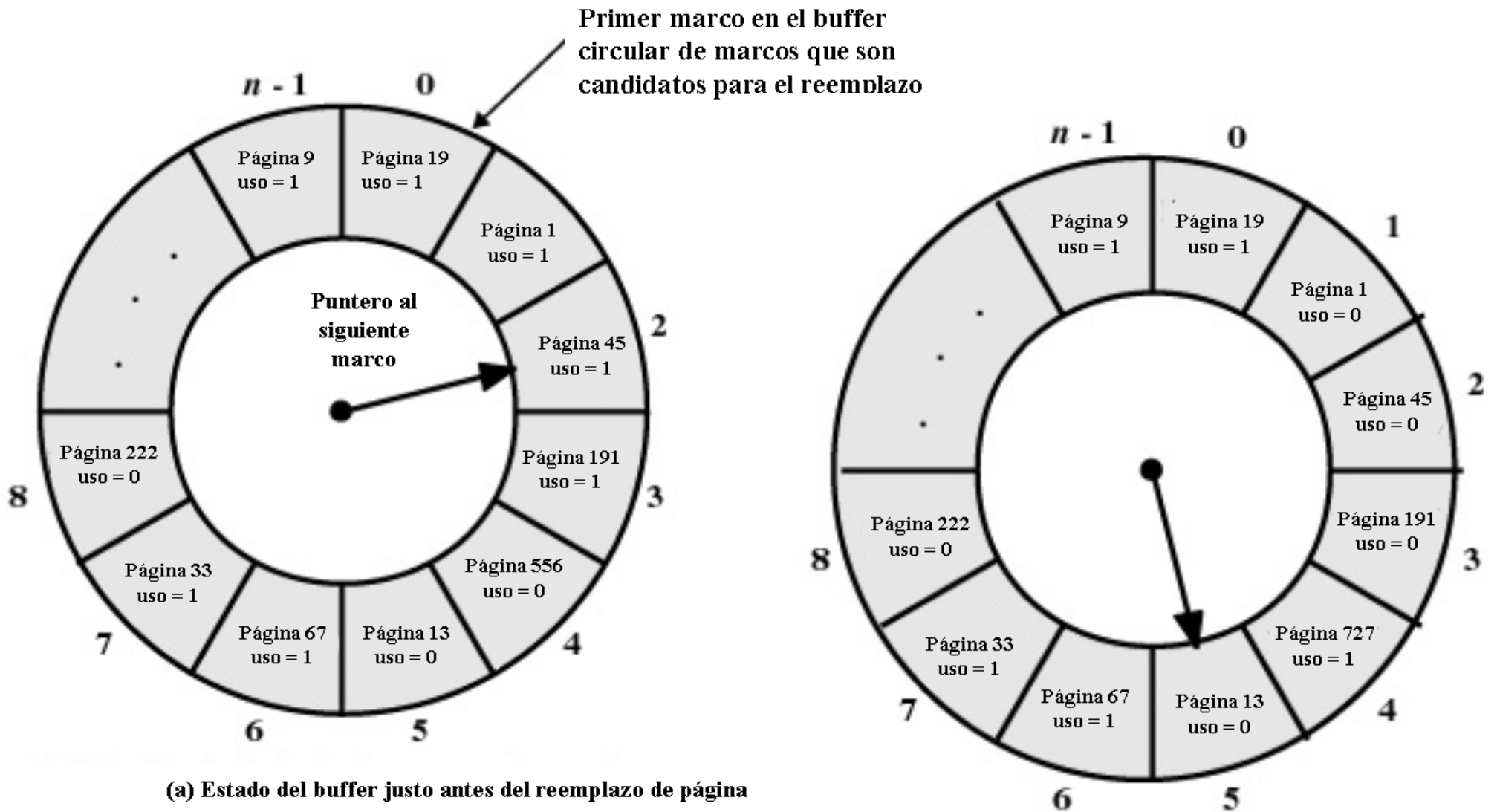
■ Política de primera en entrar primera en salir (FIFO):

- Trata los marcos asignados a un proceso como un buffer circular.
- Las páginas se suprimen de la memoria según la técnica de turno rotatorio (*round-robin*).
- Es una de las políticas de reemplazo más sencillas de implementar.
- Se reemplaza la página que ha estado más tiempo en la memoria.
- Estas páginas pueden necesitarse de nuevo y en un plazo de tiempo corto.

■ Política del reloj:

- Requiere asociar un bit adicional a cada marco, denominado bit de uso.
- Cuando se carga una página por primera vez en un marco de memoria, el bit de uso de dicho marco se inicializa a cero.
- Cuando se referencia una página existente se pone a uno su bit de uso.
- Cuando llega el momento de reemplazar una página, el primer marco encontrado con el bit de uso a 0 es reemplazado.
- Durante la búsqueda para realizar reemplazos cada bit de uso a 1 se cambia a 0.

Memoria Virtual: Segmentación



Ejercicios

Considera un computador con memoria virtual paginada con un esquema de traducción directa en un nivel, donde el espacio virtual es de 128MB, el espacio físico es de 1MB y el tamaño de página es de 4KB.

¿Qué tamaño en bytes tiene la tabla de páginas si esta contiene la información mínima para hacer la traducción (no hay ningún bit de control)?

Ejercicios

Sea un sistema de memoria paginada con un tamaño de página de 2 K sobre un bus de direcciones de 32 bits. Se pide:

- a) ¿Cuál es el espacio de direccionamiento lógico de los procesos si la tabla de páginas ocupa un tamaño fijo fijado de 32 K y cada entrada de dicha tabla son 16 bits?
- b) ¿Cuántos marcos de página puede utilizar el proceso como máximo?
- c) ¿Cuántos bits del bus de direcciones no se utilizan?
- d) ¿Cuál sería el rango de direcciones físicas expresadas en hexadecimal en que se pueden ubicar las direcciones lógicas del proceso?
- e) ¿Cómo se podría aumentar el espacio de direccionamiento lógico sin modificar el tamaño de la tabla de páginas ni los bits de cada entrada?

Ejercicios

Sea un sistema con un bus de direcciones de 16 bits, con toda la memoria direccionable instalada, donde el tamaño de página es dimensionable al arranque y se ha fijado en un tamaño tal que la tabla de página ocupa 32 entradas de 1 byte cada una y el sistema operativo está ocupando los primeros 8Kb de memoria,

- a) Cuál es el tamaño de la memoria instalada
- b) Cuál es el tamaño de página que establece este sistema.
- c) Cuál es el tamaño máximo del conjunto residente de un proceso
- d) Cuál es el tamaño máximo de direccionamiento lógico de un proceso
- e) Si la tabla de páginas está completa, ¿se puede producir un fallo de página? ¿Por qué?

Ejercicios

Sea un sistema de memoria paginada con un tamaño de página de 512 bytes sobre un bus de direcciones de 16 bits con toda la memoria direccionable instalada.

Si la parte residente del sistema operativo son siempre los primeros 6K, se pide:

- a) ¿Qué tamaño ocupará la tabla de páginas si cada entrada es de 2 bytes?
- b) ¿Cuántos marcos de página puede utilizar el proceso como máximo?
- c) ¿Cuál sería la dirección física correspondiente a la dirección lógica 0x05D1 si la tabla de páginas es la siguiente?
- d) ¿Se podría optimizar el tamaño de la tabla de páginas?, si se puede, indica cómo.
- e) Al estar usando memoria virtual paginada ¿Se podría cargar un programa más grande en este sistema? Razona la respuesta.
- f) Si dispusiéramos únicamente de la mitad de la memoria física, ¿Cuántas entradas como máximo de la tabla de página tendrían el bit de presencia a cero?

PAG	MARCO
0	12
1	23
2	127
3	13
4	26
5	45
6	27
7	12
8	33
9	14
...	...

Ejercicios

Sea un sistema de memoria paginada con un tamaño de página de 2 K sobre un bus de direcciones de 32 bits. El sistema operativo puede gestionar toda la memoria direccionable con 32 bits que está instalada y se reserva los primeros 64MB de memoria.

- a)¿Cuál es el espacio de direccionamiento lógico de los procesos si la tabla de páginas ocupa 4 MB y cada entrada de la tabla de páginas tiene 4 bytes siendo 5 bits de control?
- b)¿Cuántos bits del bus de direcciones no se utilizan ?
- c)¿Cuál es el tamaño máximo direccionable por el sistema?
- d)¿Cuál es el tamaño máximo del conjunto residente de un proceso?
- e)¿Puede ser el grado de multiprogramación mayor que 1 sin que se produzcan fallos de página?



Memoria Virtual

