SISTEMAS OPERATIVOS

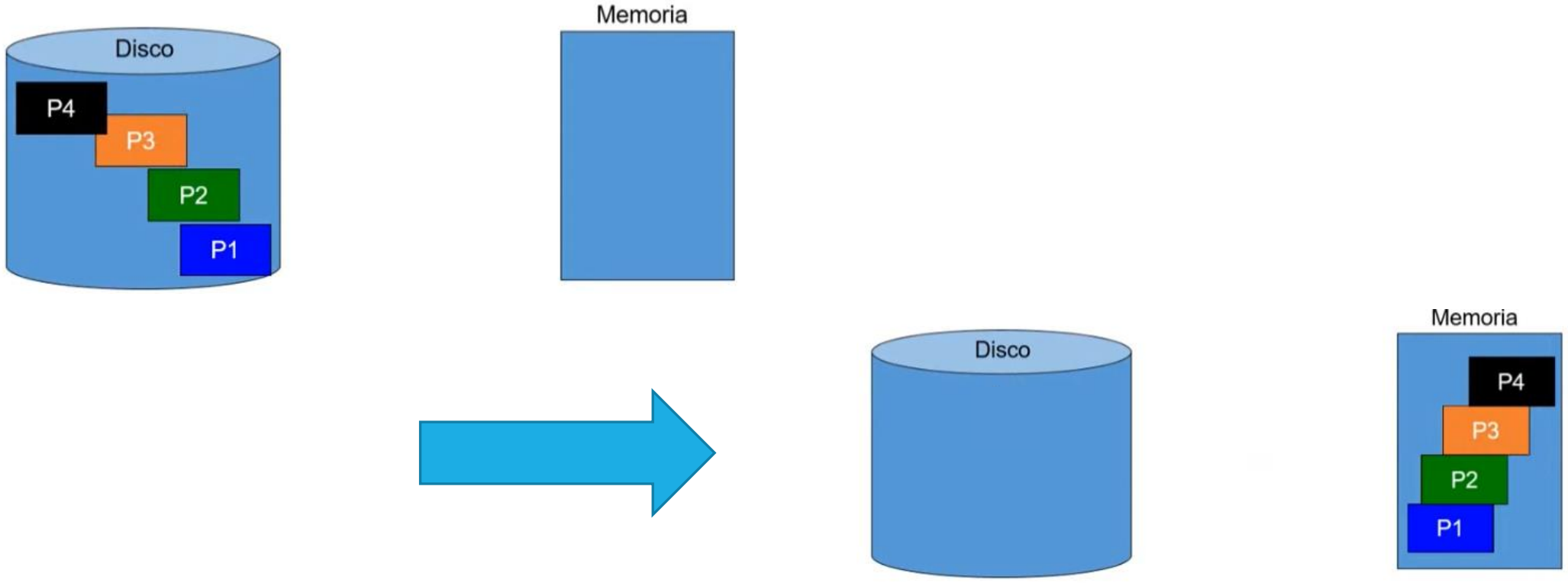
Ingeniería civil informática

GONZALO CARREÑO

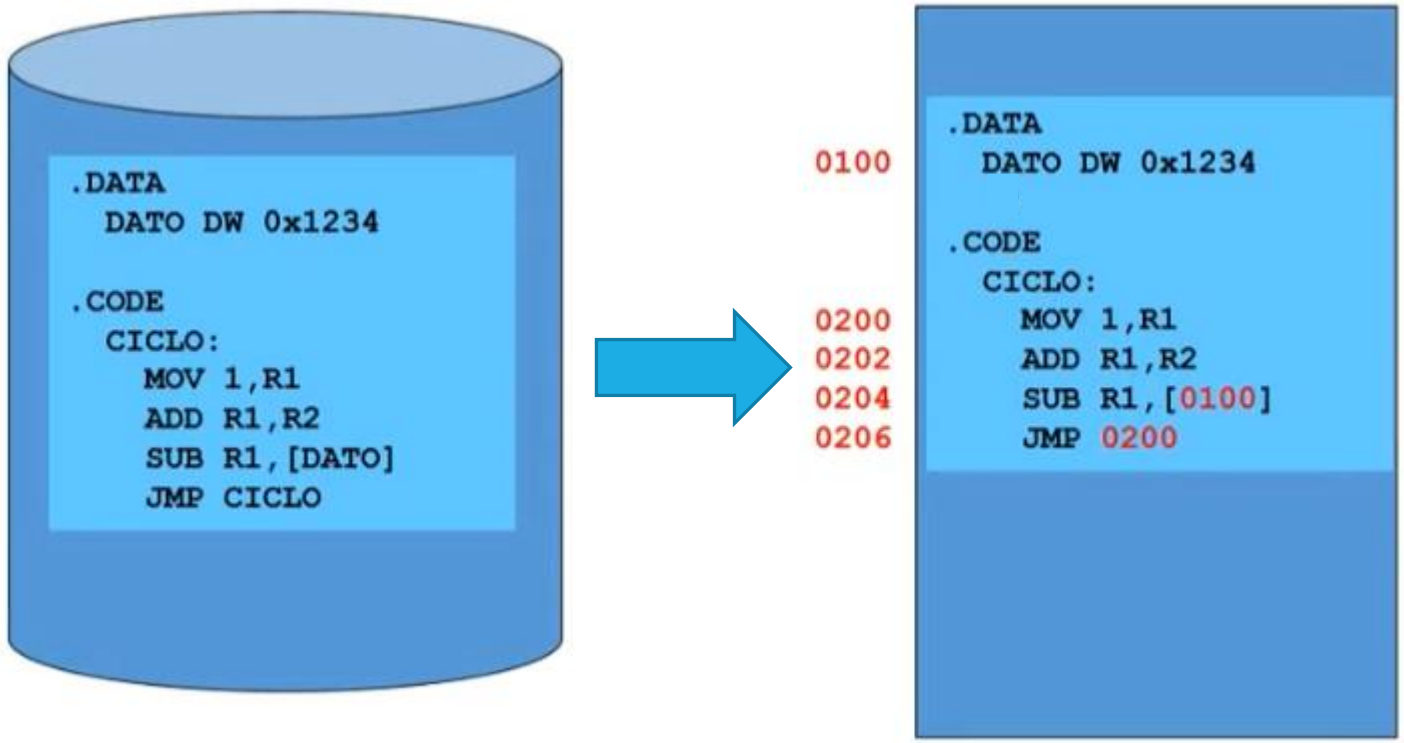
GONZALOCARRENOB@GMAIL.COM

Introducción

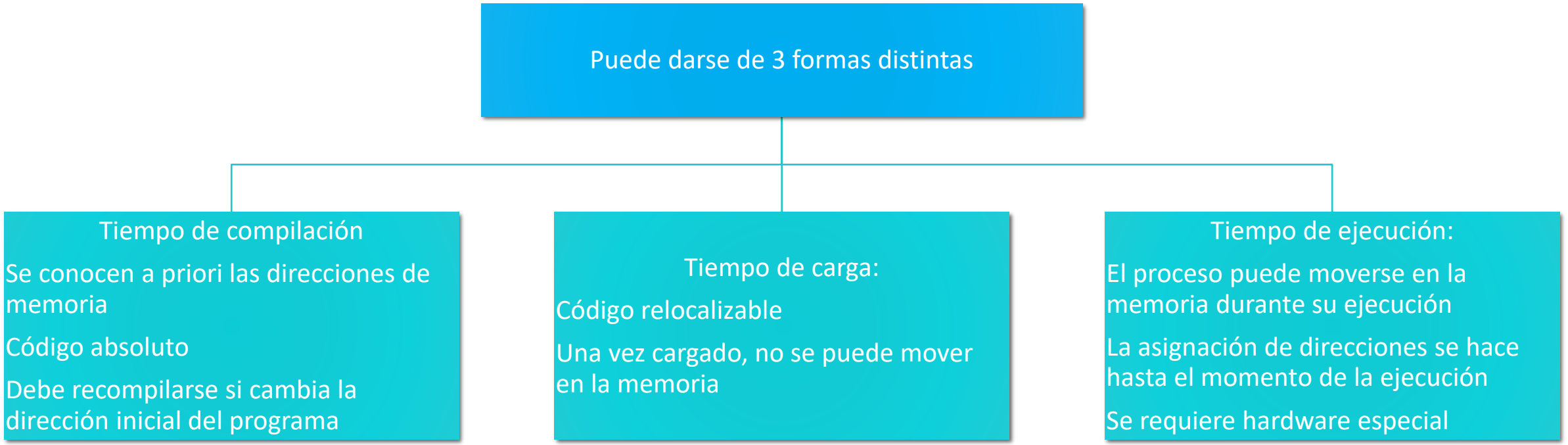
Un programa debe traerse a memoria y establecerse dentro de un proceso a ser ejecutado.



Asociando instrucciones y datos a direcciones de memoria



Asociando instrucciones y datos a direcciones de memoria

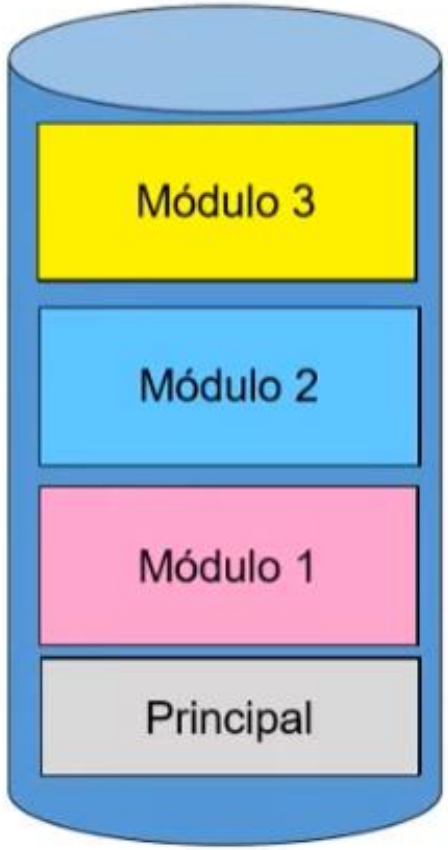


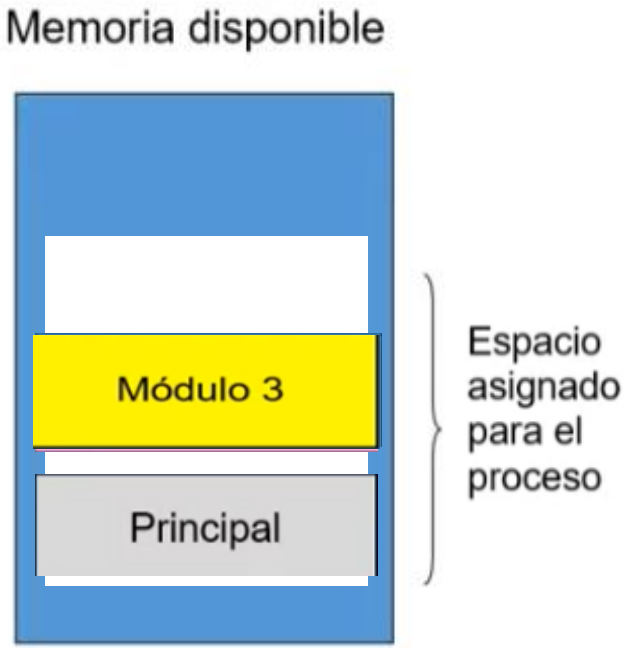
Asociando instrucciones y datos a direcciones de memoria

Carga dinámica

- Una rutina no se carga hasta que es llamada
- Una rutina no usada nunca se carga. Mejor utilización del espacio en memoria
- Útil cuando grandes cantidades de código se necesitan para manejar casos que no ocurren frecuentemente
- No se requiere soporte especial del sistema operativo, puede implementarse por diseño del programa

Overlays
Ejemplo MSDOS





Encadenamiento dinámico

El encadenamiento dinámico se pospone hasta el tiempo de ejecución

Stub: Pequeña pieza de código usada para localizar la rutina de librería residente en memoria.

El Stub se reemplaza con la dirección de la rutina, y ejecuta la rutina.

El sistema operativo se necesita para verificar si la rutina esta en direcciones de memoria del proceso

Espacio de direcciones lógico v/s espacio de direcciones físico

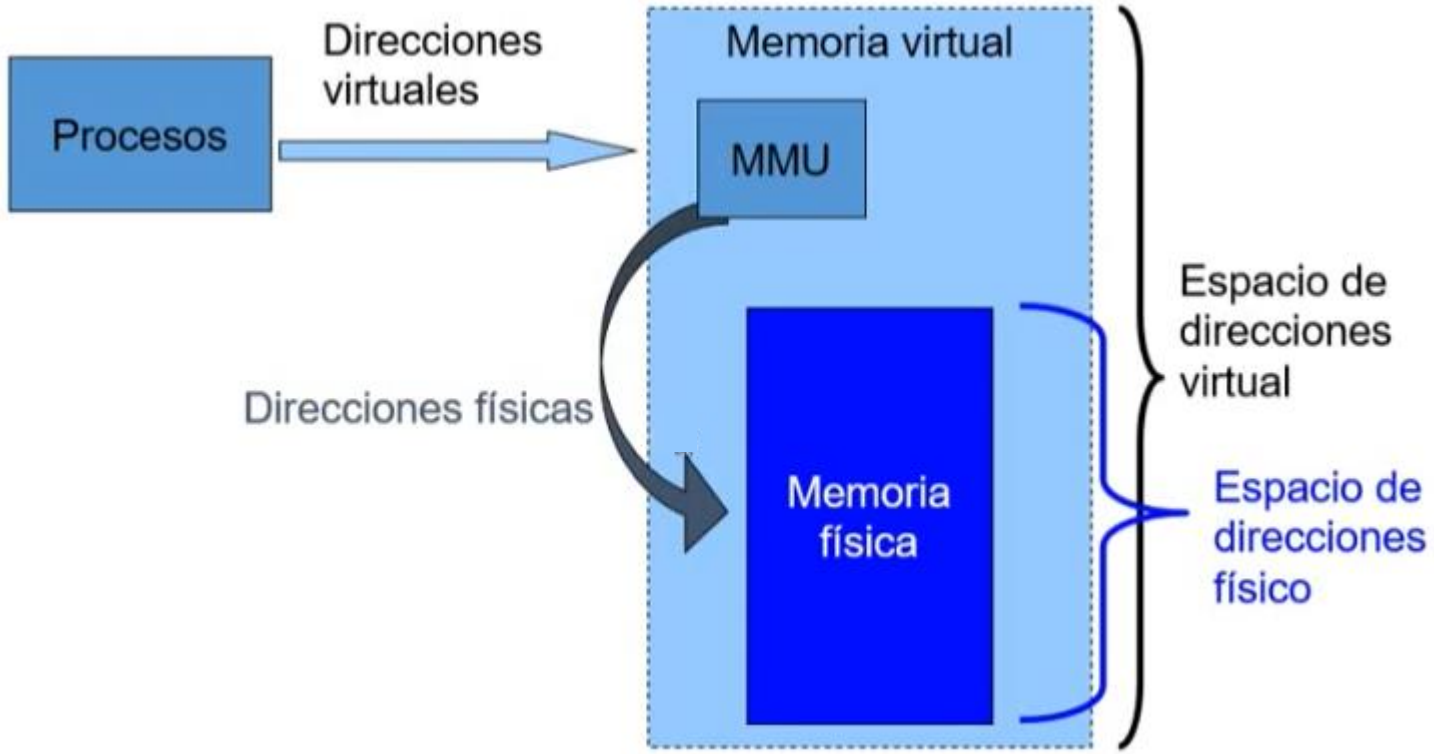
Dirección lógica

- Generada por el CPU
- También se refiere a una dirección virtual

Dirección física

- Direcciones vistas por la unidad de manejo de memoria (mmu)

Direcciones lógicas y físicas



Unidad de manejo de memoria MMU

Es el dispositivo de hardware que mapea direcciones virtuales a direcciones físicas

Registro de relocalización

- Se añade a cada dirección generada por un proceso de usuario a la vez que se manda a memoria

El programa de usuario trata con direcciones lógicas
Nunca ve el espacio de direcciones físicas.

Intercambio (Swapping)

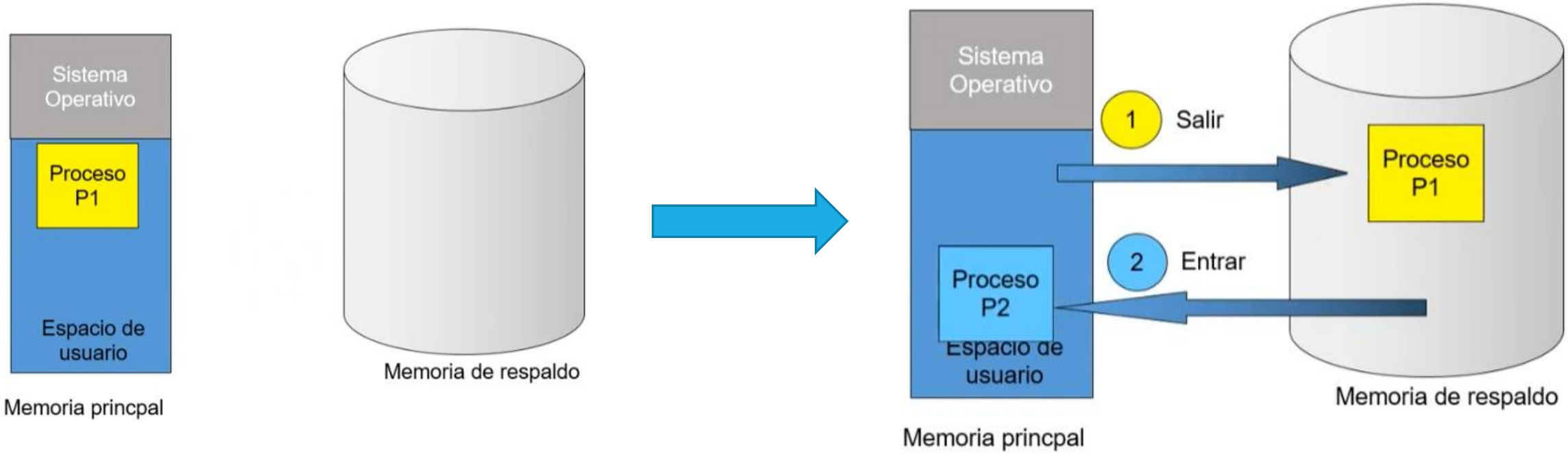
Un proceso puede salir temporalmente de la memoria a un **almacenamiento de respaldo**, y luego traerse de regreso a memoria para continuar en ejecución.

Almacenamiento de respaldo

- Disco grande y rápido para acomodar copias de las imágenes de memoria de todos los usuarios
- Debe proveer acceso directo a esas imágenes de memoria

Intercambio (Swapping)

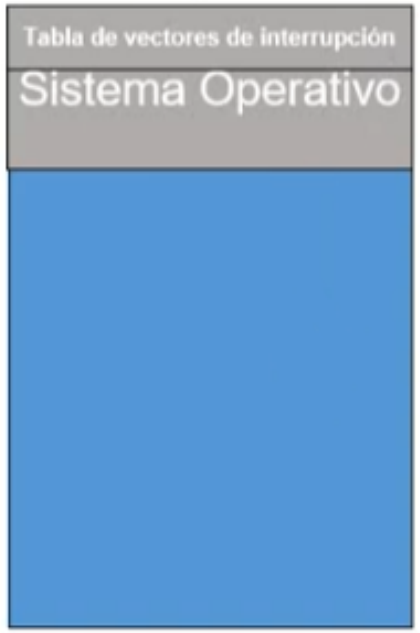
Visión esquemática del intercambio



Asignación contigua

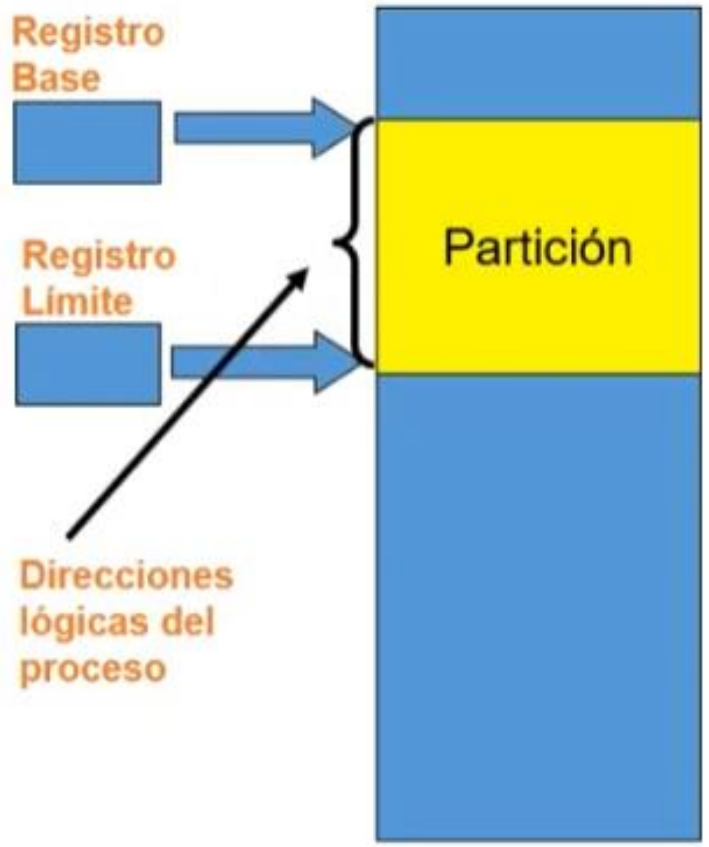
La memoria principal en dos particiones:

- El sistema operativo residente, se mantiene en la parte baja de la memoria con la tabla de vectores de interrupción.



Asignación contigua

- Asignación de una partición
  - El esquema de registro de recolocación usado para la protección
    - El **registro base** contiene el valor de la dirección física más pequeña
    - El **registro límite** contiene el rango de direcciones lógicas
    - Cada dirección lógica debe ser más pequeña que el registro límite



Asignación contigua

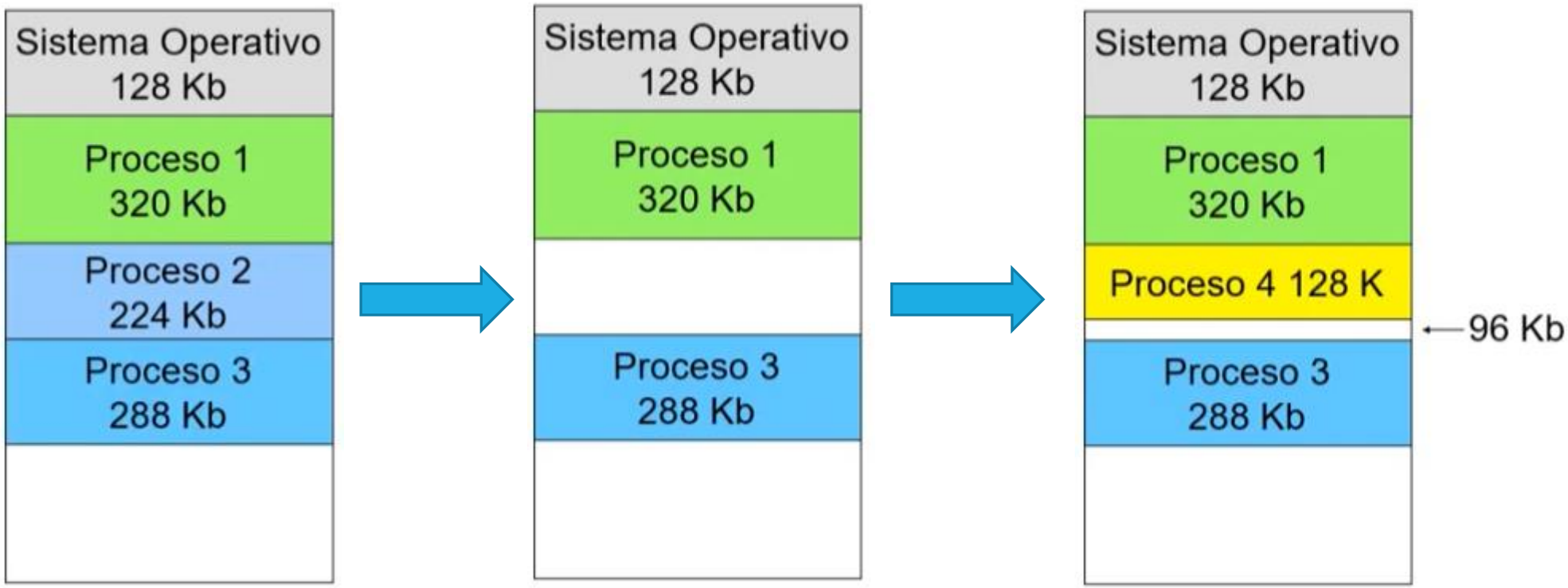
Asignación de particiones múltiples

Hueco
- Bloque de memoria disponible
- Se van creando huecos en la memoria de diferentes tamaños

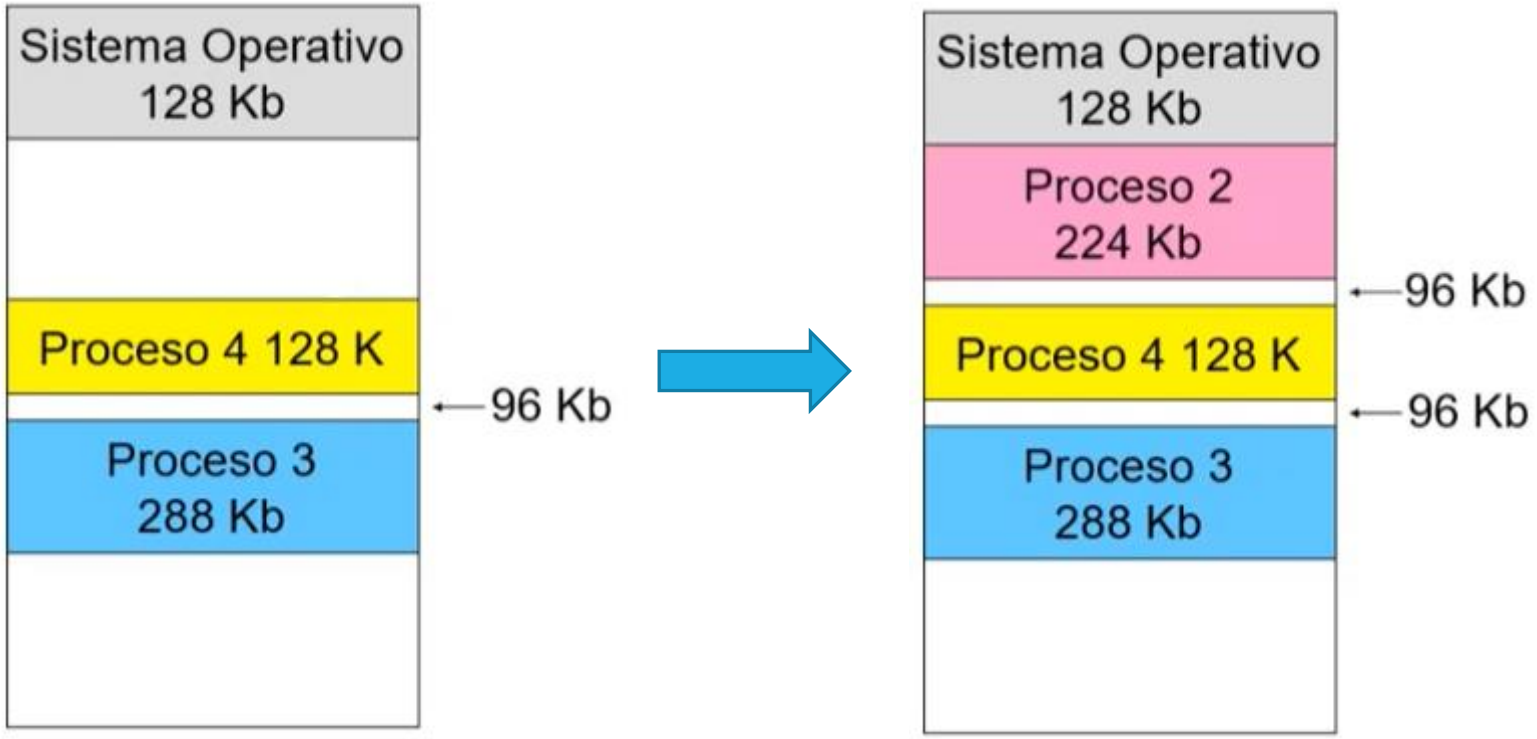
Cuando llega un proceso, se asigna a memoria a un hueco lo suficientemente grande para acomodarlo

El sistema operativo mantiene información sobre:
a) Particiones asignadas
b) Particiones libres

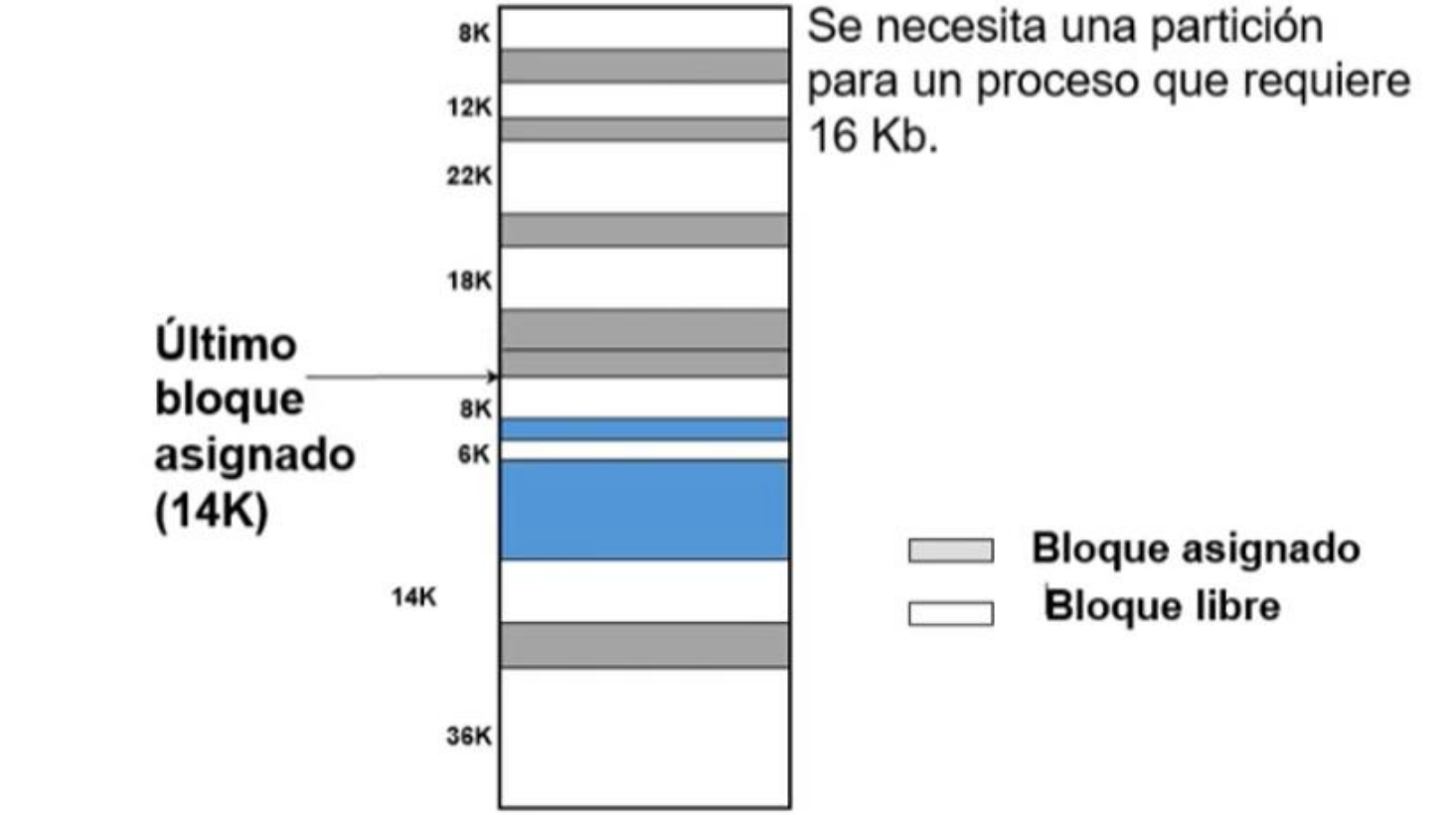
Ejemplo de particiones dinámicas



Ejemplo de particiones dinámicas



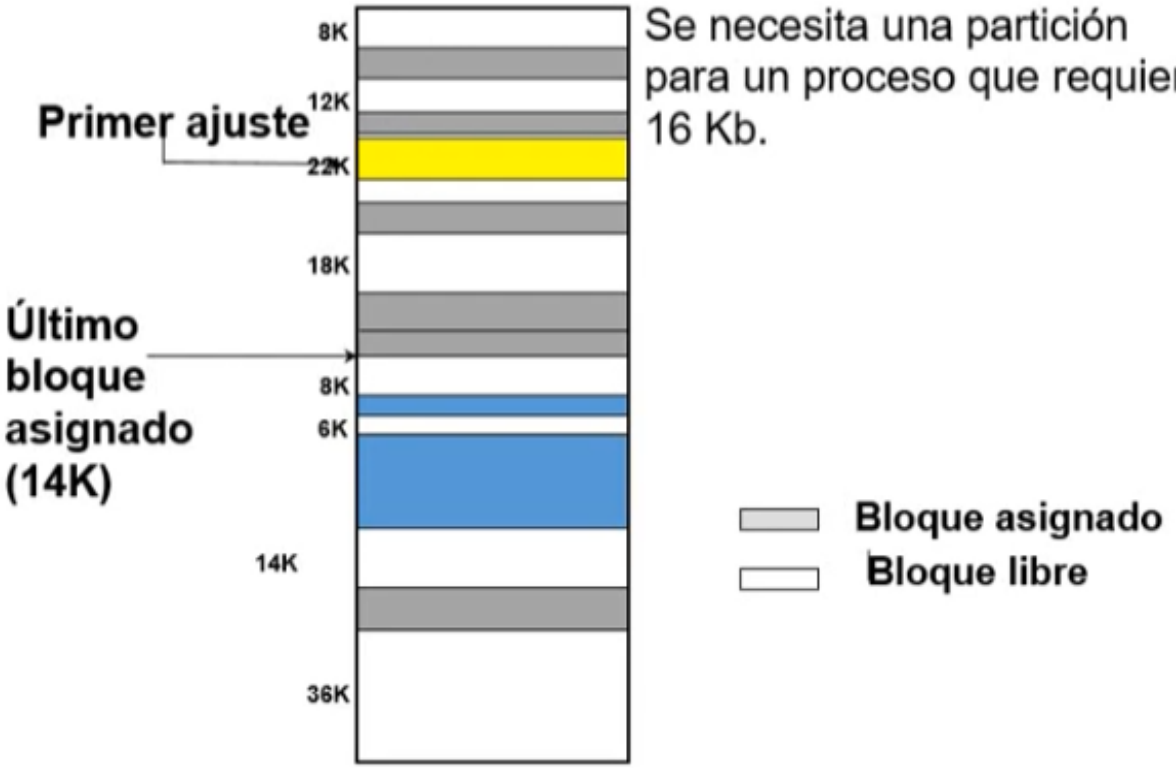
Partición dinámica y algoritmos de ubicación



¿Cómo satisfacer una solicitud de tamaño n de una lista de bloques libres?

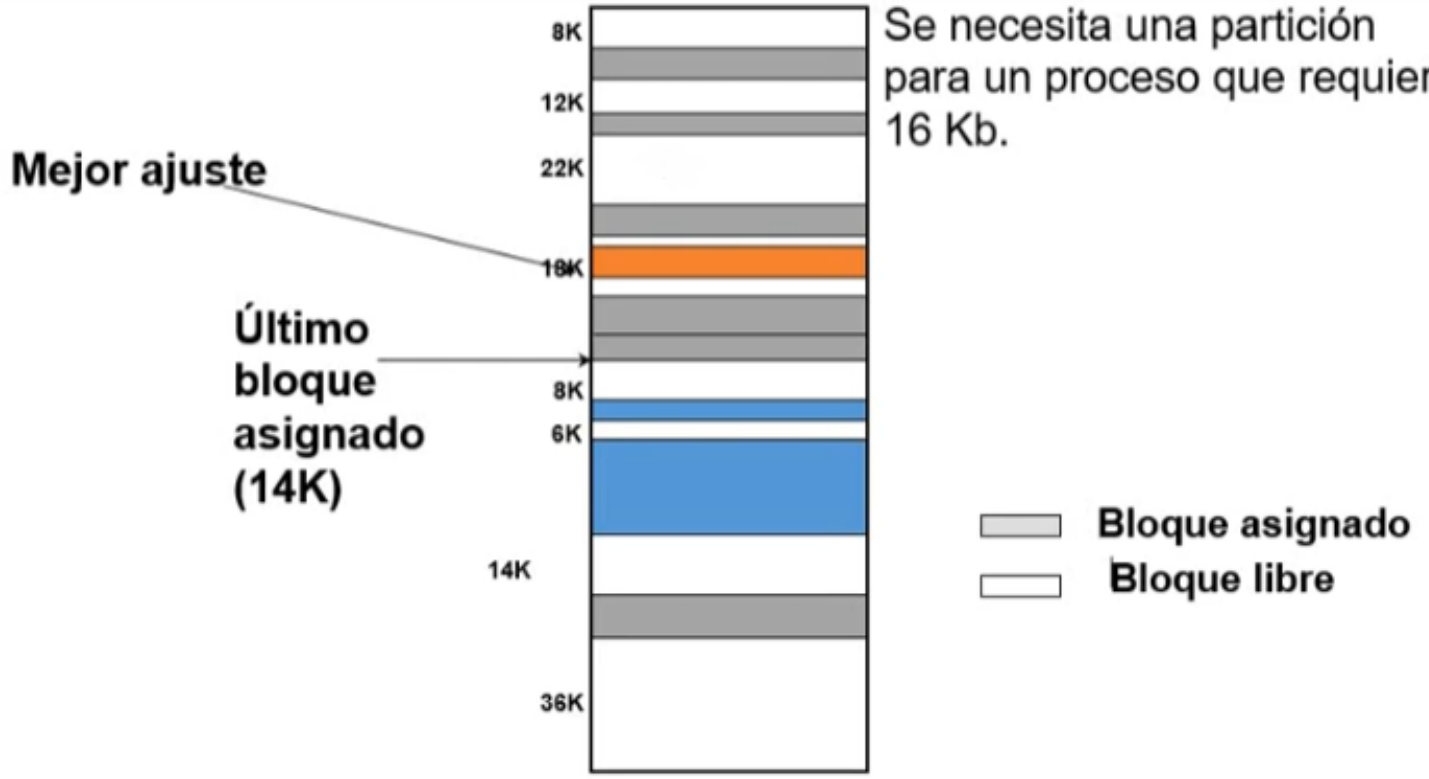
Partición dinámica y algoritmos de ubicación

Primer ajuste: Asigna el primer bloque donde puede caber el proceso.



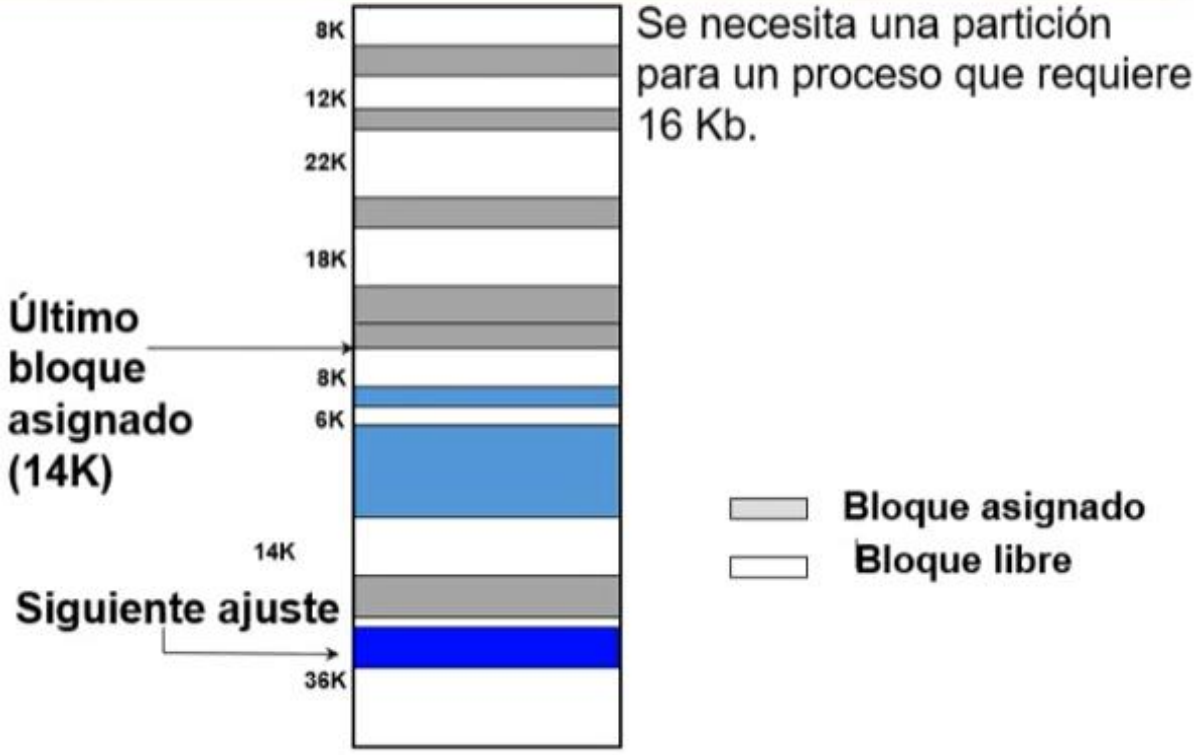
Partición dinámica y algoritmos de ubicación

Mejor ajuste: Asigna el bloque más pequeño donde puede caber el proceso, produce los fragmentos más pequeños



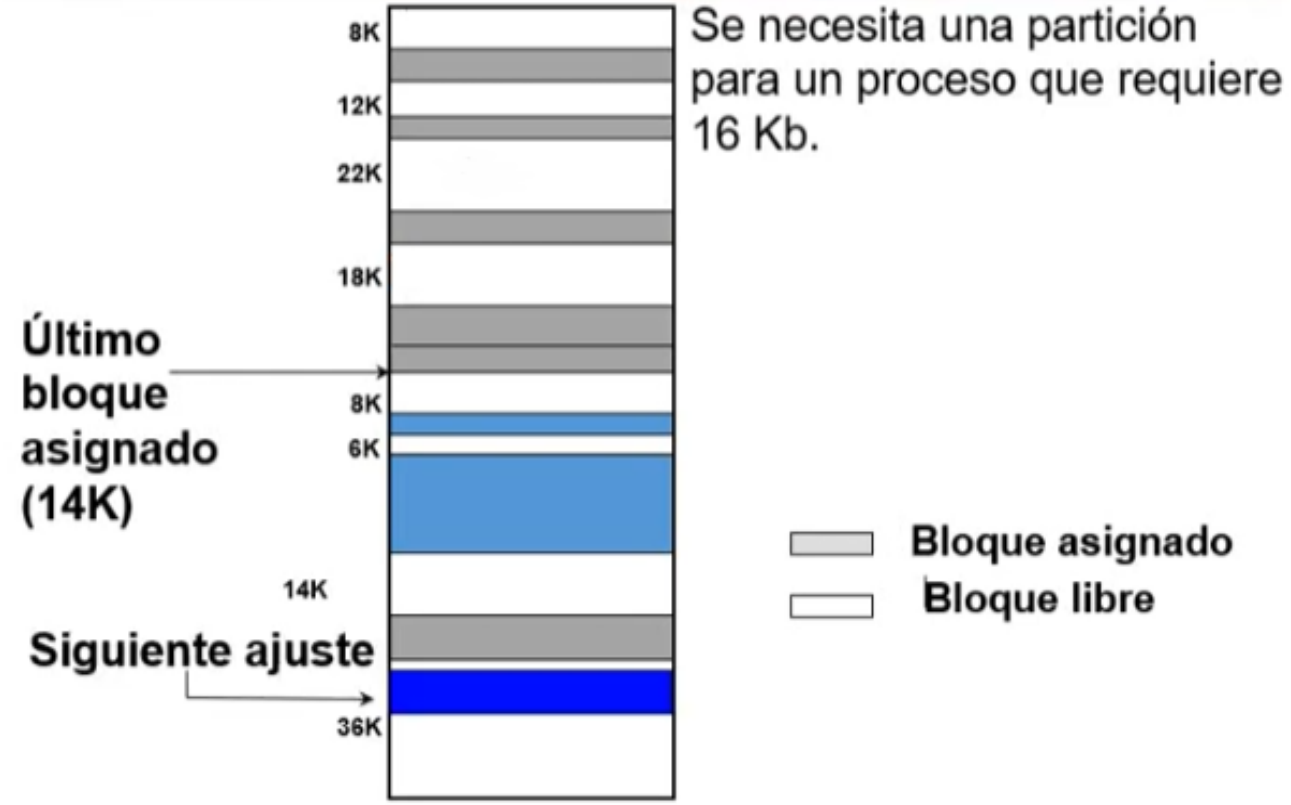
Partición dinámica y algoritmos de ubicación

Siguiente ajuste: Asigna el siguiente bloque lo suficientemente grande donde pueda caber el proceso



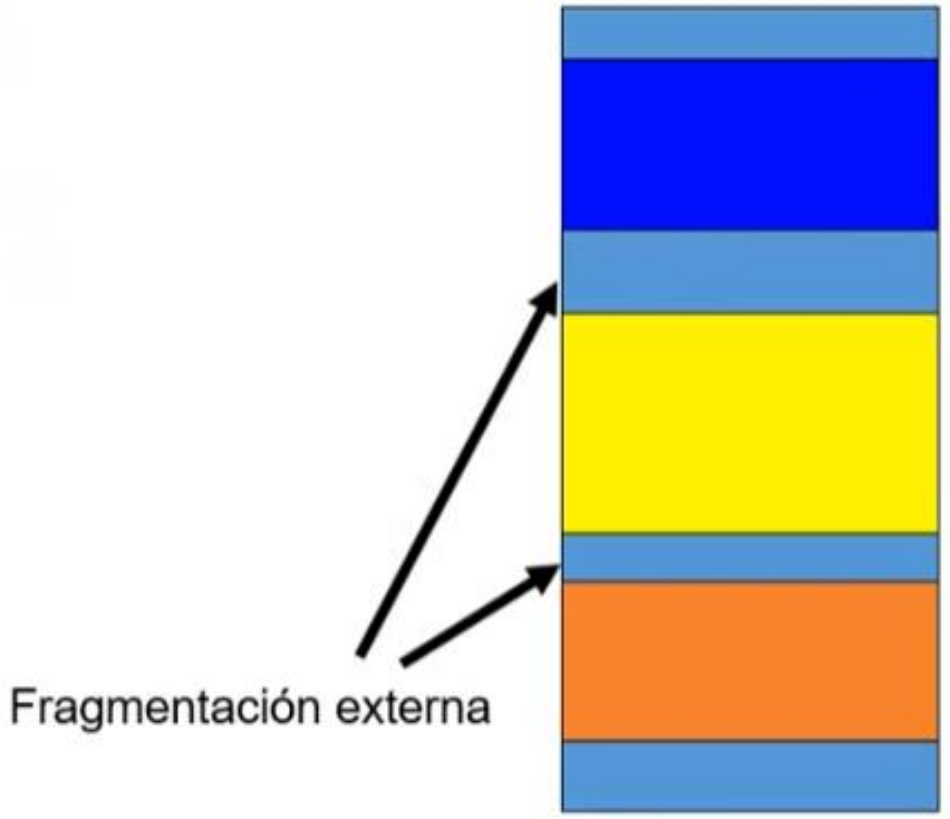
Partición dinámica y algoritmos de ubicación

Peor ajuste: Asigna el hueco más grande, produce fragmentos muy grandes



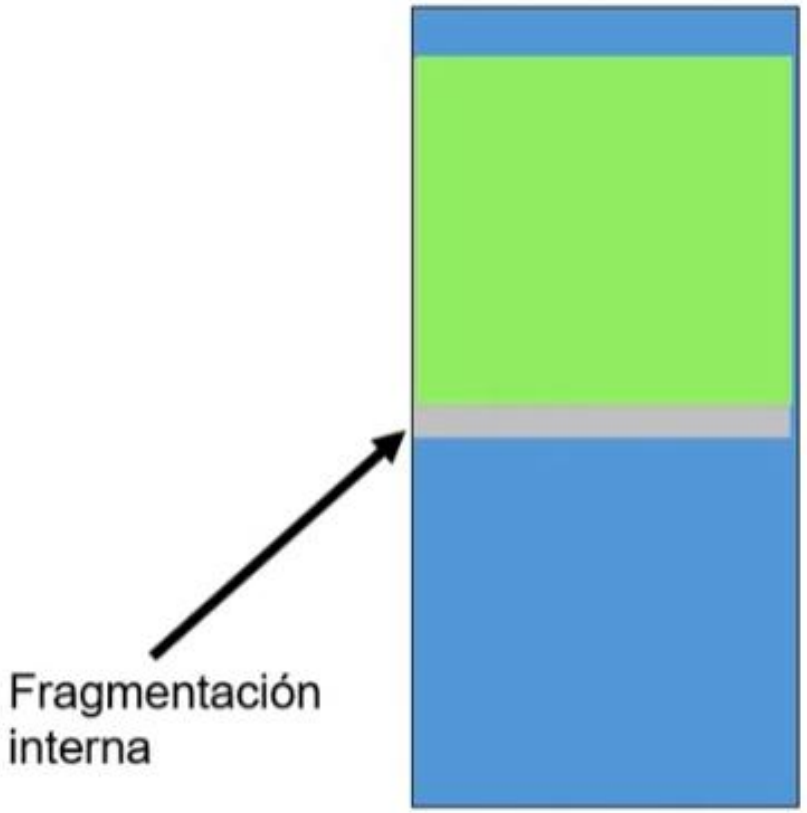
Fragmentación

- Fragmentación externa
  - Existe un espacio de memoria para satisfacer una solicitud, pero no es contiguo



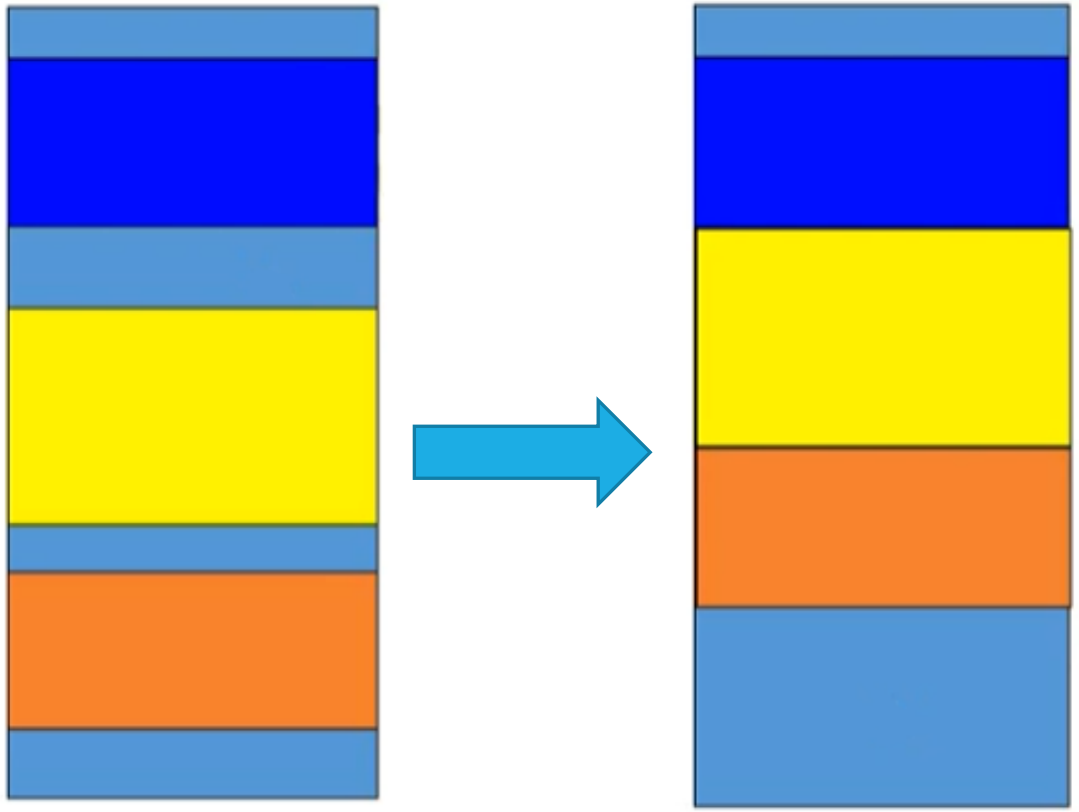
Fragmentación

- Fragmentación interna
  - La memoria asignada es un poco mayor a la memoria solicitada
  - Esta pequeña cantidad de memoria que sobra dentro de la partición que no se usa



Fragmentación

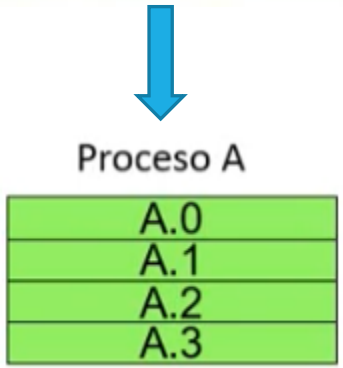
- Reducir la fragmentación externa por compactación
  - Desplazar el contenido de la memoria para tener toda la memoria disponible en un solo bloque



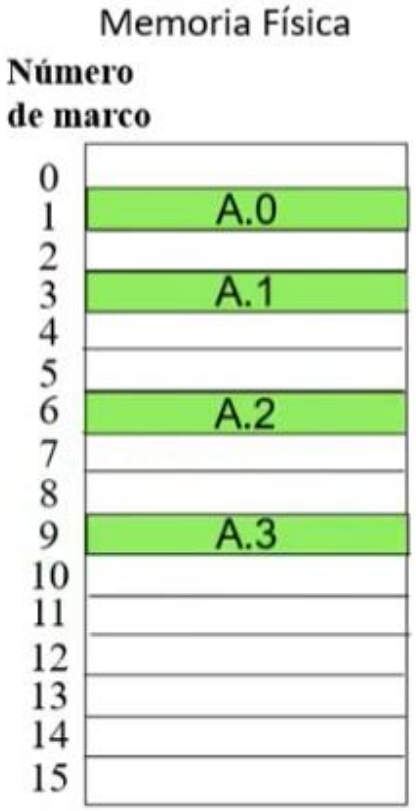
Paginación

Divide la memoria física en bloques de tamaño fijo llamados **marcos**
El tamaño de los marcos es potencia de 2, usualmente entre 512 y 8192 bytes

Divide los procesos en bloques del mismo tamaño llamados páginas



Paginación



El espacio de direcciones lógico de un proceso puede no estar contiguo

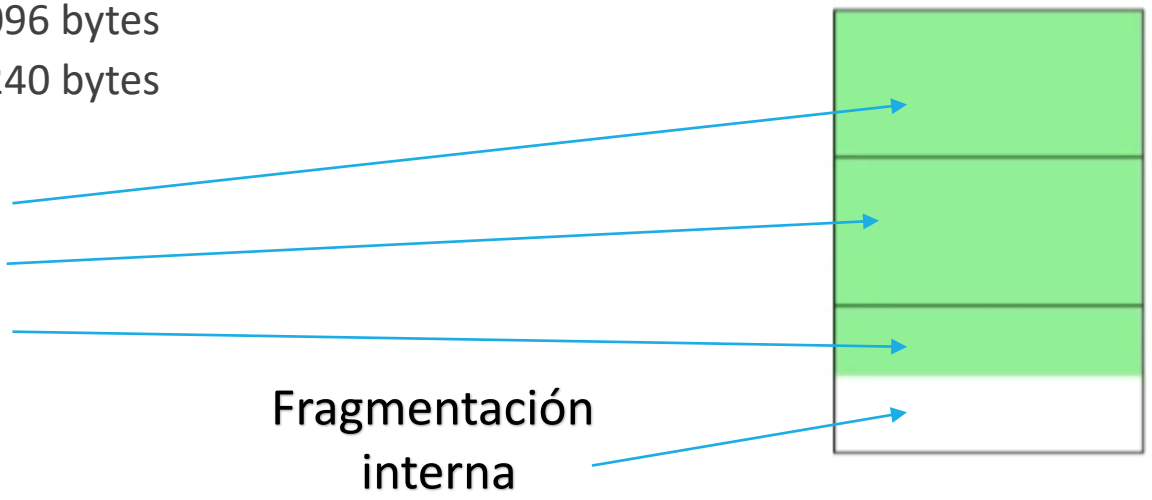
Paginación

Para ejecutar un programa de n paginas, se necesitan n marcos libres para cargar el programa.

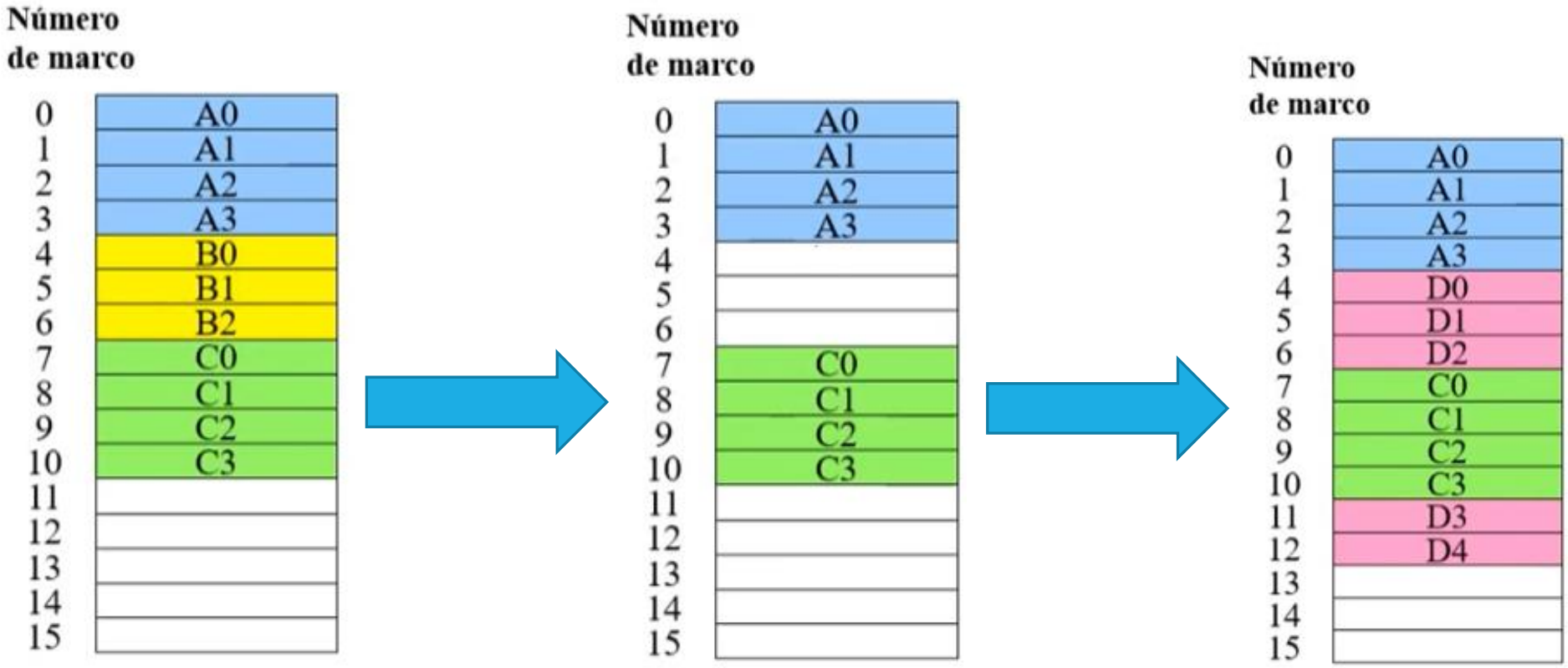
Una tabla de paginas para traducir direcciones lógicas a direcciones físicas

Hay fragmentación interna

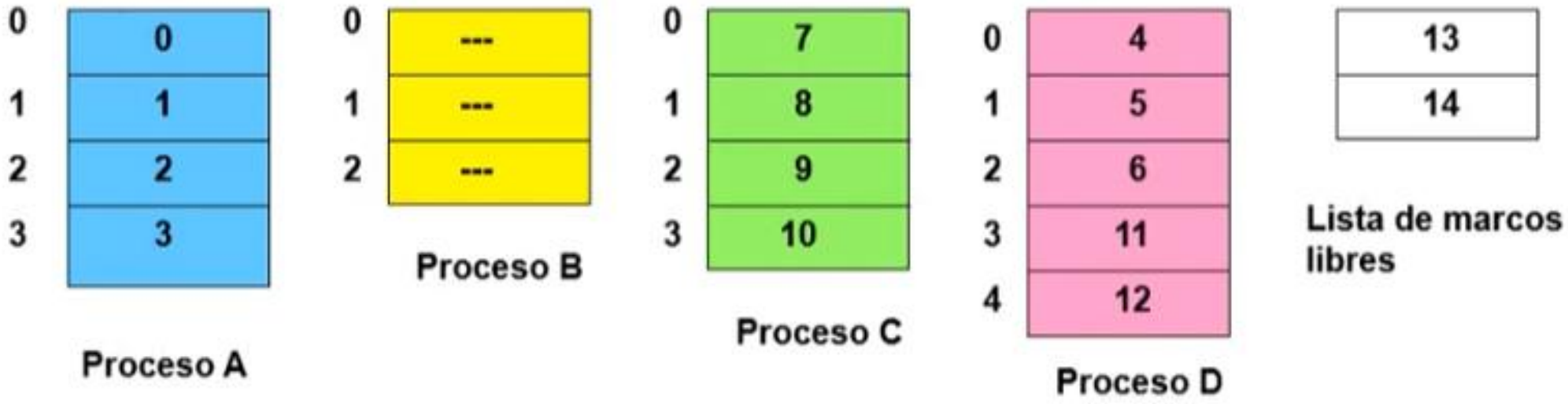
- Ejemplo:
  - Las paginas miden 4096 bytes
  - Un proceso mide 10240 bytes
  - Requiere 3 paginas
    - Pág. 0: 4096 bytes
    - Pág. 1: 4096 bytes
    - Pág. 2: 2048 bytes



Paginación

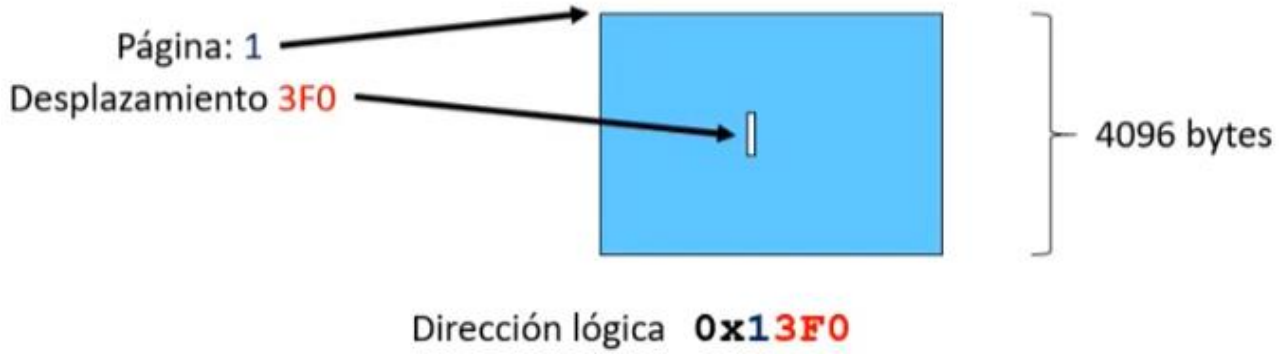

Número de marco
0 A0
1 A1
2 A2
3 A3
4 B0
5 B1
6 B2
7 C0
8 C1
9 C2
10 C3
11
12
13
14
15
Número de marco
0 A0
1 A1
2 A2
3 A3
4
5
6
7 C0
8 C1
9 C2
10 C3
11
12
13
14
15
Número de marco
0 A0
1 A1
2 A2
3 A3
4 D0
5 D1
6 D2
7 C0
8 C1
9 C2
10 C3
11 D3
12 D4
13
14
15

Ejemplo de tablas de páginas



Direcciones lógicas

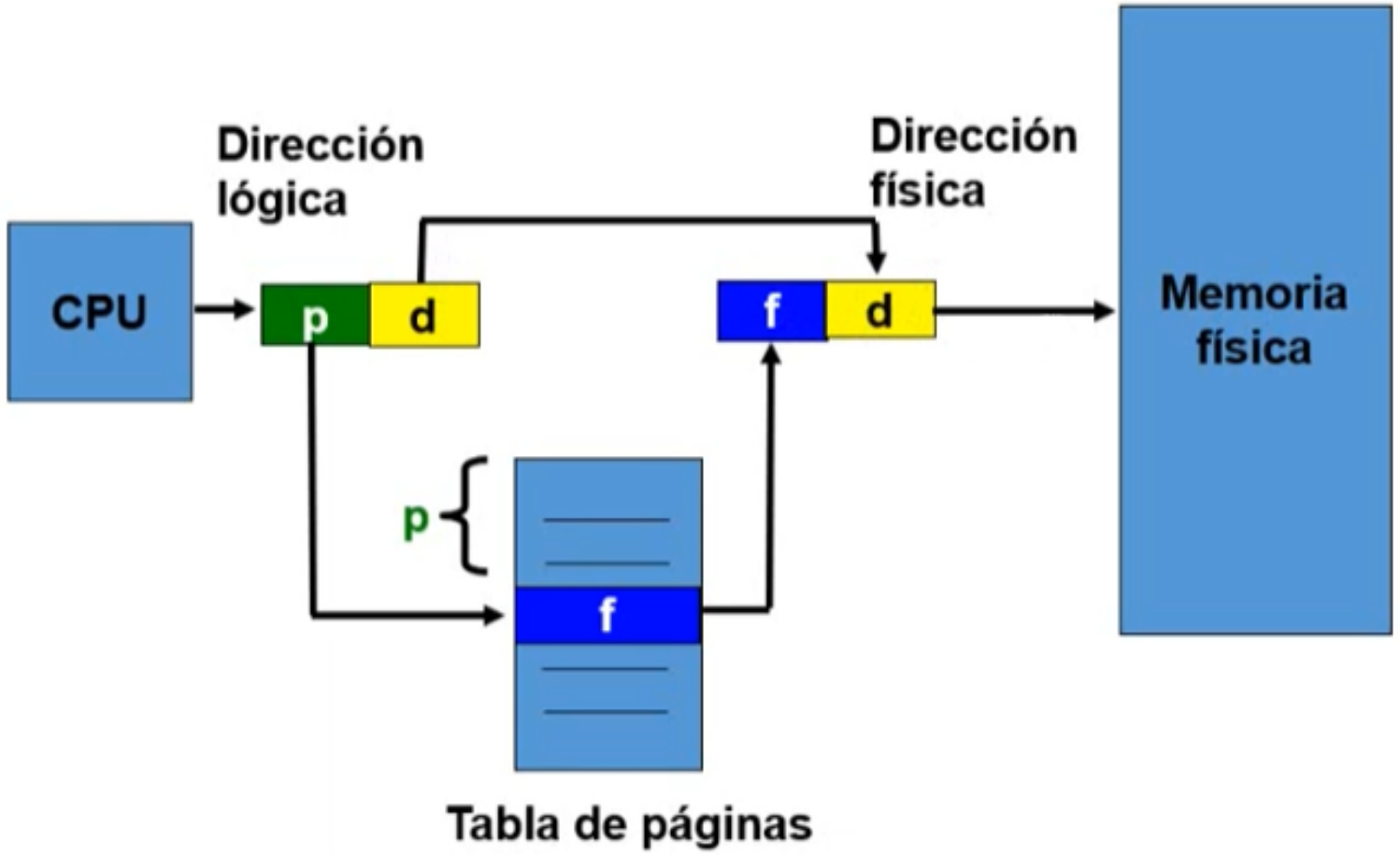
- Las direcciones lógicas se componen de:
  - Número de página (p)
  - Desplazamiento (d)



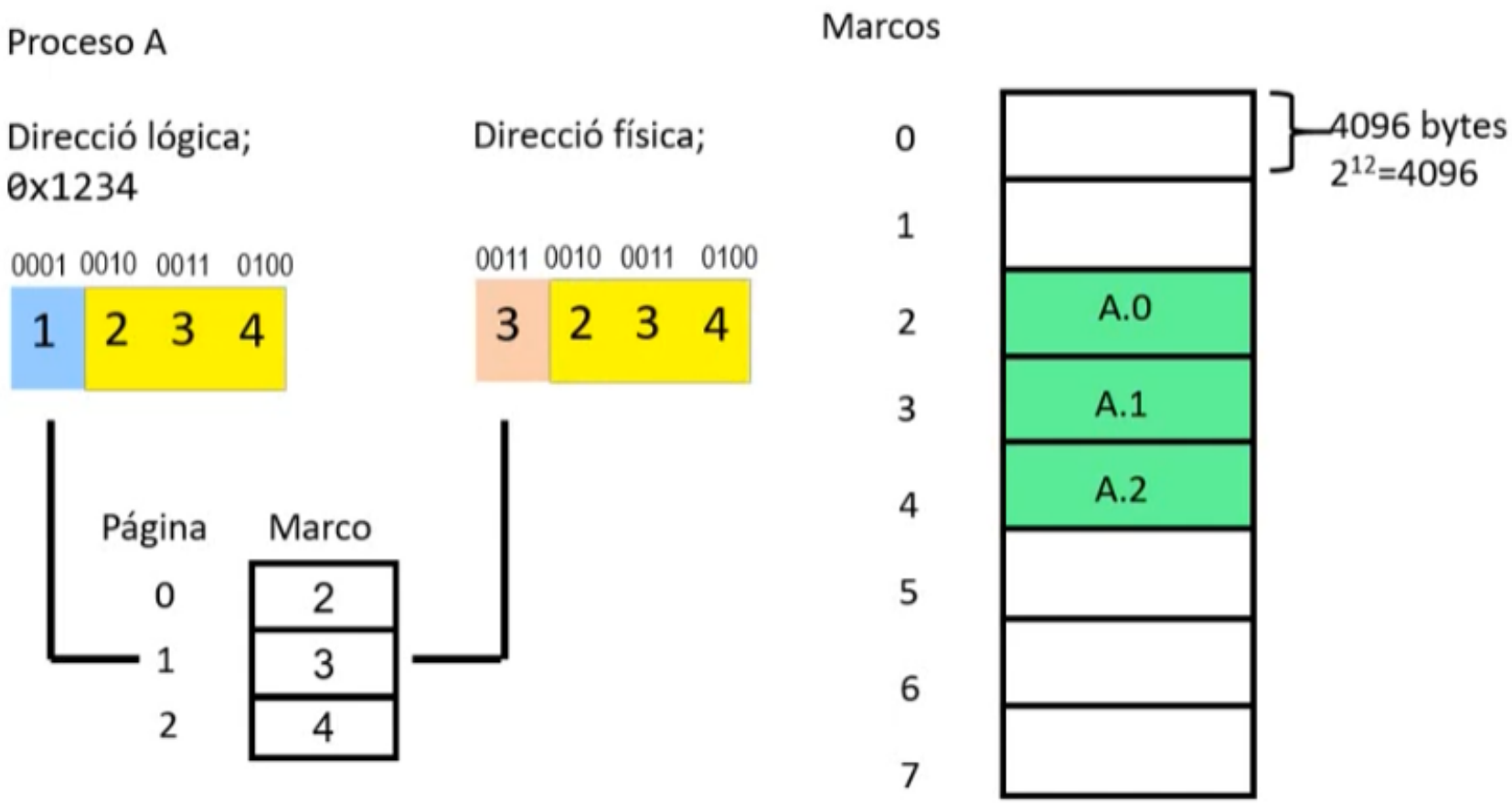
Esquema de traducción de direcciones

- Las direcciones generadas por el CPU se dividen en:
  - Número de página (p)
    - Se usa como un índice en la tabla de páginas que contiene la dirección base de cada página en memoria física
  - Desplazamiento (d)
    - Combinado con la dirección base define la dirección física que se envía a la unidad de manejo de memoria (mmu)

Arquitectura de traducción de direcciones

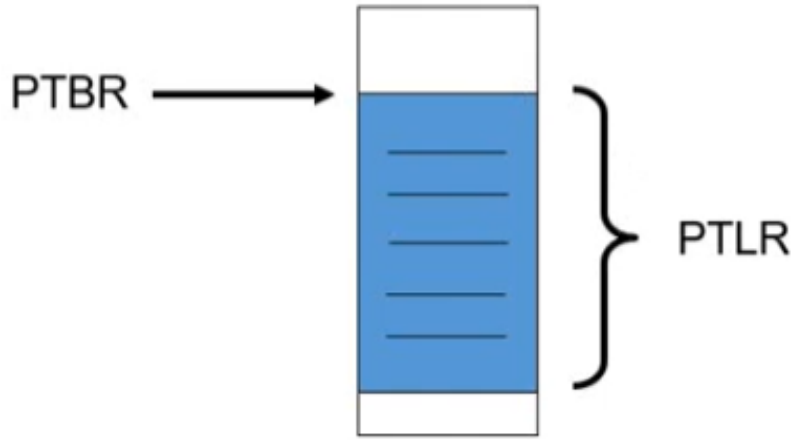


Traducción de dirección



Implementación de tabla de páginas

- La tabla de páginas se mantiene en memoria principal
- *Page-table base register (*PTBR)
  - Apunta a la tabla de páginas
- *Page-table length register* (PTLR)
  - Indica el tamaño de la tabla de páginas



Implementación de tabla de páginas

- En este esquema cada acceso de datos/instrucciones requiere dos accesos a memoria.
  - Uno para la tabla de páginas
  - Otro para los datos/instrucciones

- El problema de los dos accesos a memoria se resuelve con el uso de un caché de hardware
  - Registros asociativos o buffer de traducción adelantada
  - TLB=Translation Look-aside Buffer

Registro asociativo

- Registros asociativos – búsqueda en paralelo

- Traducción de direcciones (A', A'')
  - Si A' está en el registro asociativo
    - Obtener el número de marco
  - Si no
    - Obtener el número de marco de la tabla de páginas en memoria

Tiempo de acceso efectivo

- Búsqueda asociativa = ε unidades de tiempo
- Asume el tiempo del ciclo de memoria es 10 nanosegundos
- Tasa de aciertos
  - Porcentaje de veces que un número de página es encontrado en registros asociativos
  - La tasa se relaciona con el número de registros asociativos
  - Tasa de aciertos = α
- Effective Access Time (EAT)
  - EAT = $(10ns + \varepsilon)\,\alpha + (20ns + \varepsilon)(1 - \alpha) = 20ns + \varepsilon - 10\alpha$

Paginación de dos niveles

- Problema:
  - Una dirección lógica de 40 bits
    - 28 bits para número de página y 12 bits desplazamiento

    - Un proceso puede tener 2^{28} páginas
      - Más de 256 millones de páginas
    - Las tablas de páginas pueden ser demasiado grandes
      - No caber en memoria
      - Tener que salir a memoria secundaria

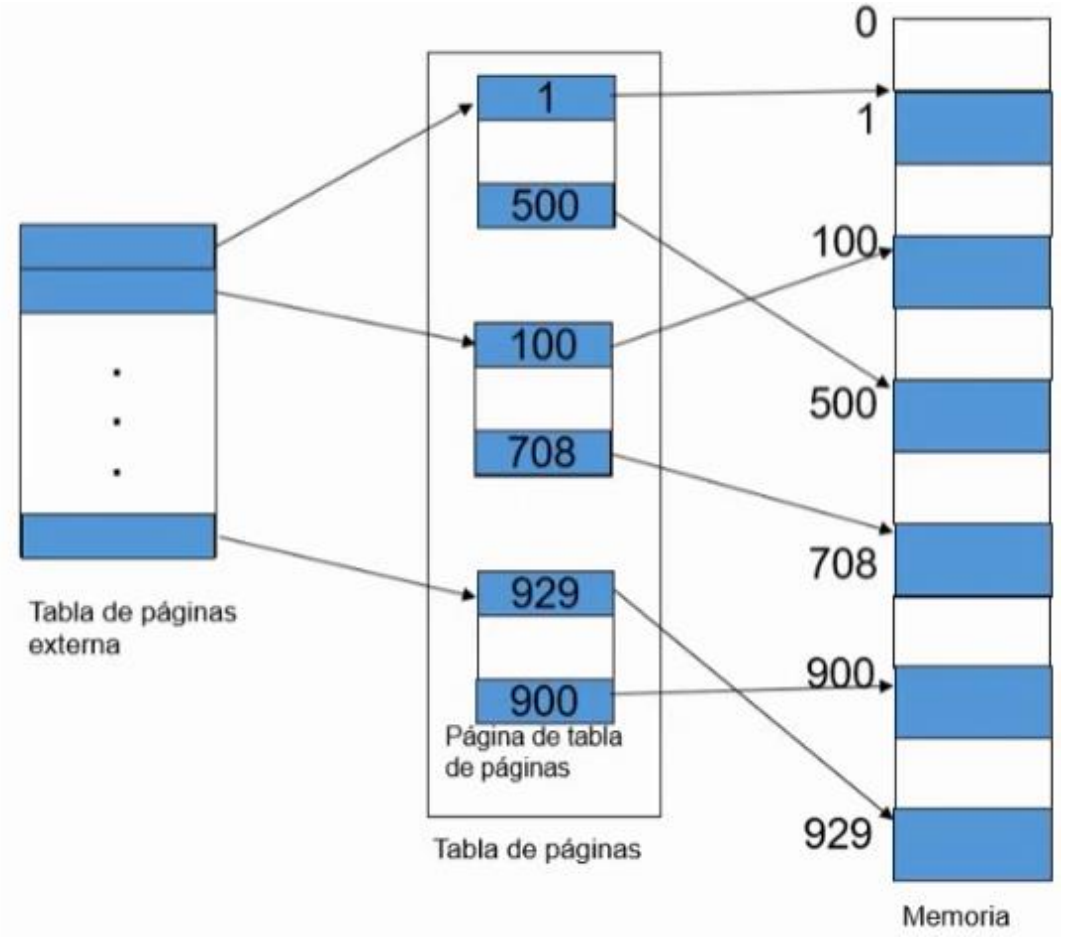
Paginación de dos niveles

- Una dirección lógica (en una máquina de 32 bits con páginas de 4 Kb) se divide en:
  - Numero de página (20 bits)
  - Desplazamiento (12 bits)

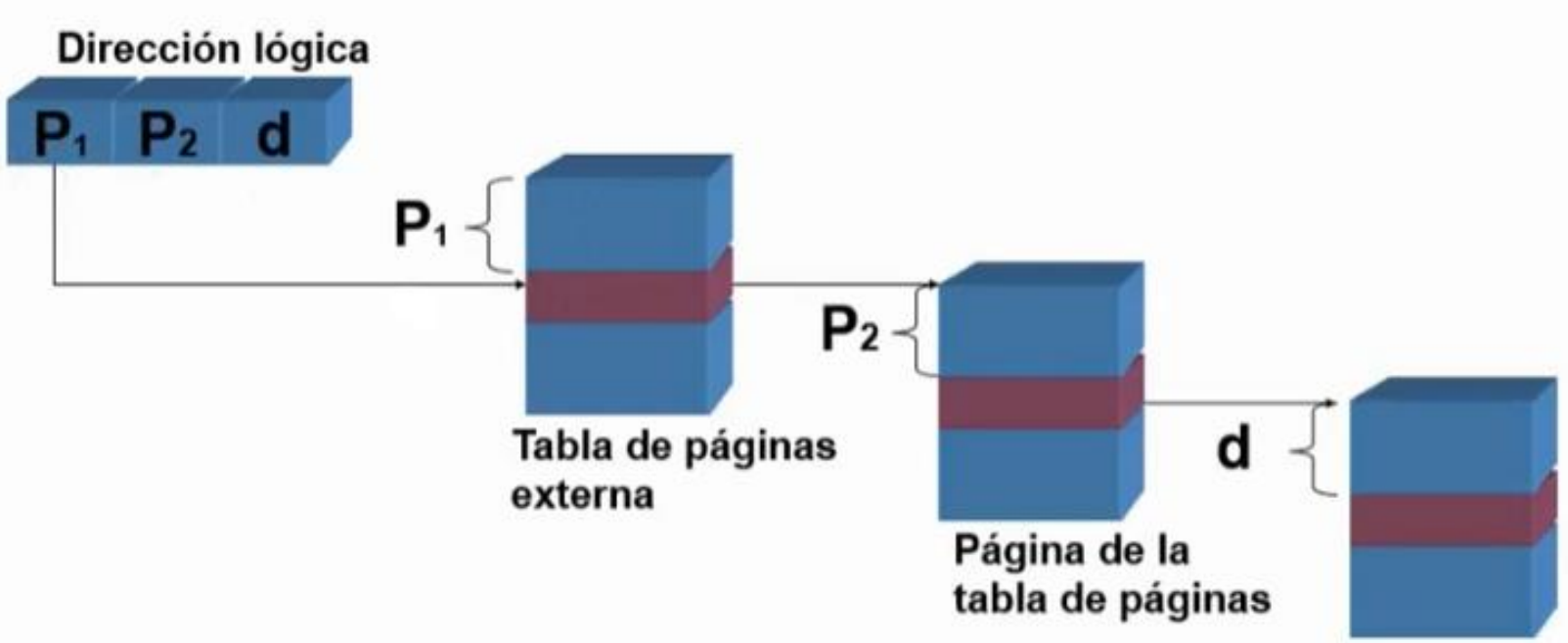
Paginación de dos niveles

- Una dirección lógica (en una máquina de 32 bits con páginas de 4 Kb) se divide en:
  - Numero de página (20 bits)
  - Desplazamiento (12 bits)
- Una dirección puede dividirse en dos niveles de paginación
  - 10 bits para la tabla de páginas externa
  - 10 bits para la tabla de páginas interna
  - 12 bits para el desplazamiento

Esquema de paginación de dos niveles



Paginación de dos niveles

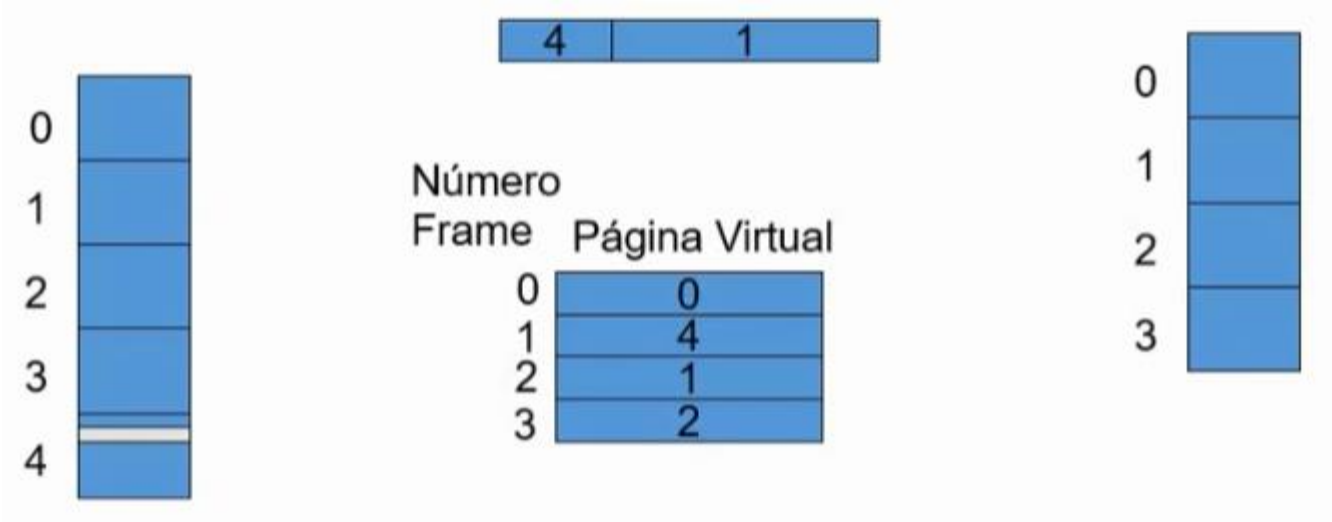


- Una dirección lógica se forma de la siguiente manera
  - P_1 es un índice en la tabla de páginas externa
  - P_2 es el desplazamiento dentro de la página de la tabla de páginas interna

Paginación multinivel y rendimiento

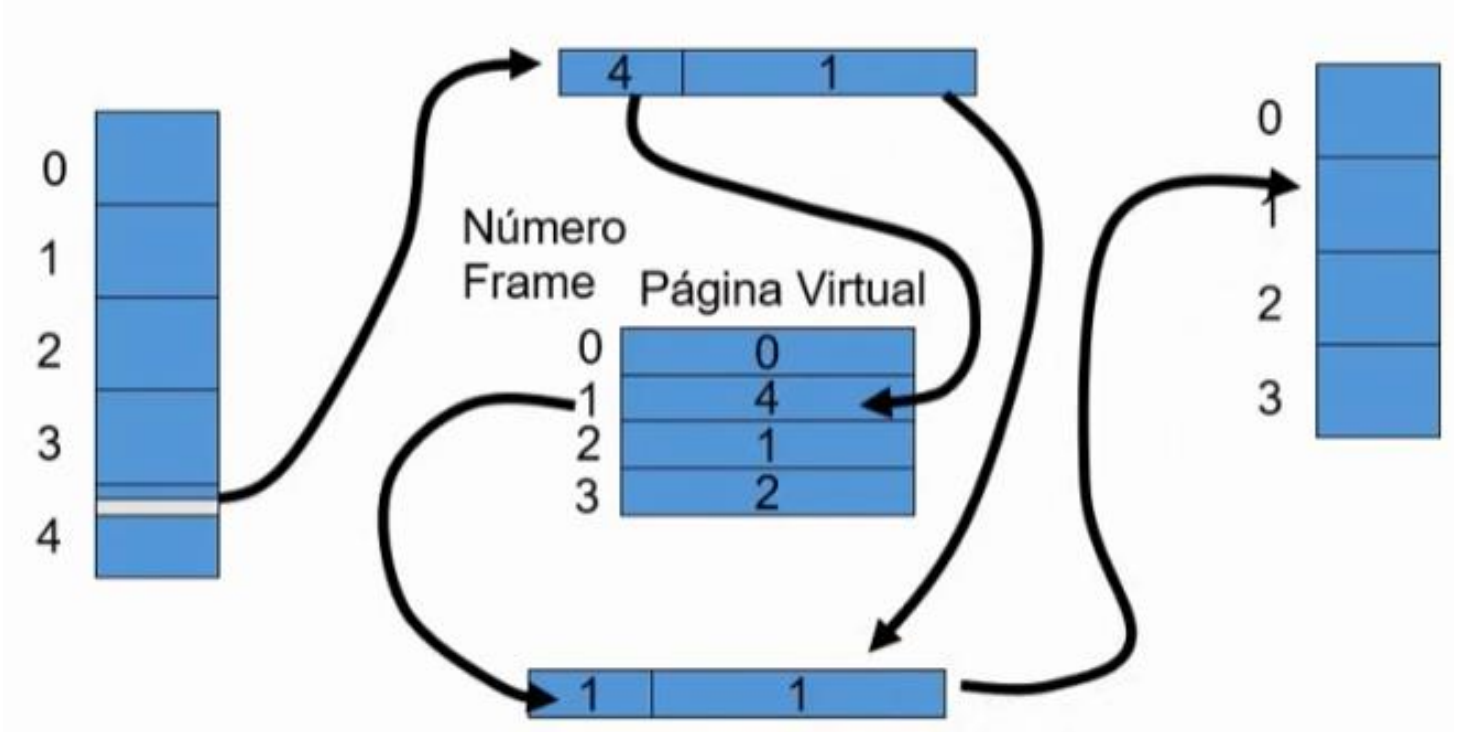
- Cada nivel es almacenado como una tabla separada en memoria
  - Convertir una dirección lógica a una dirección física puede tomarse hasta 4 accesos a memoria
- El tiempo necesario para un acceso a memoria puede quintuplicarse
  - Usar caché permite que el rendimiento sea razonable.

Tabla de pagina invertida



- Una entrada por cada página real de la memoria
- Cada entrada consiste en el número de la página almacenada en esa marco, con información del proceso dueño de esa página

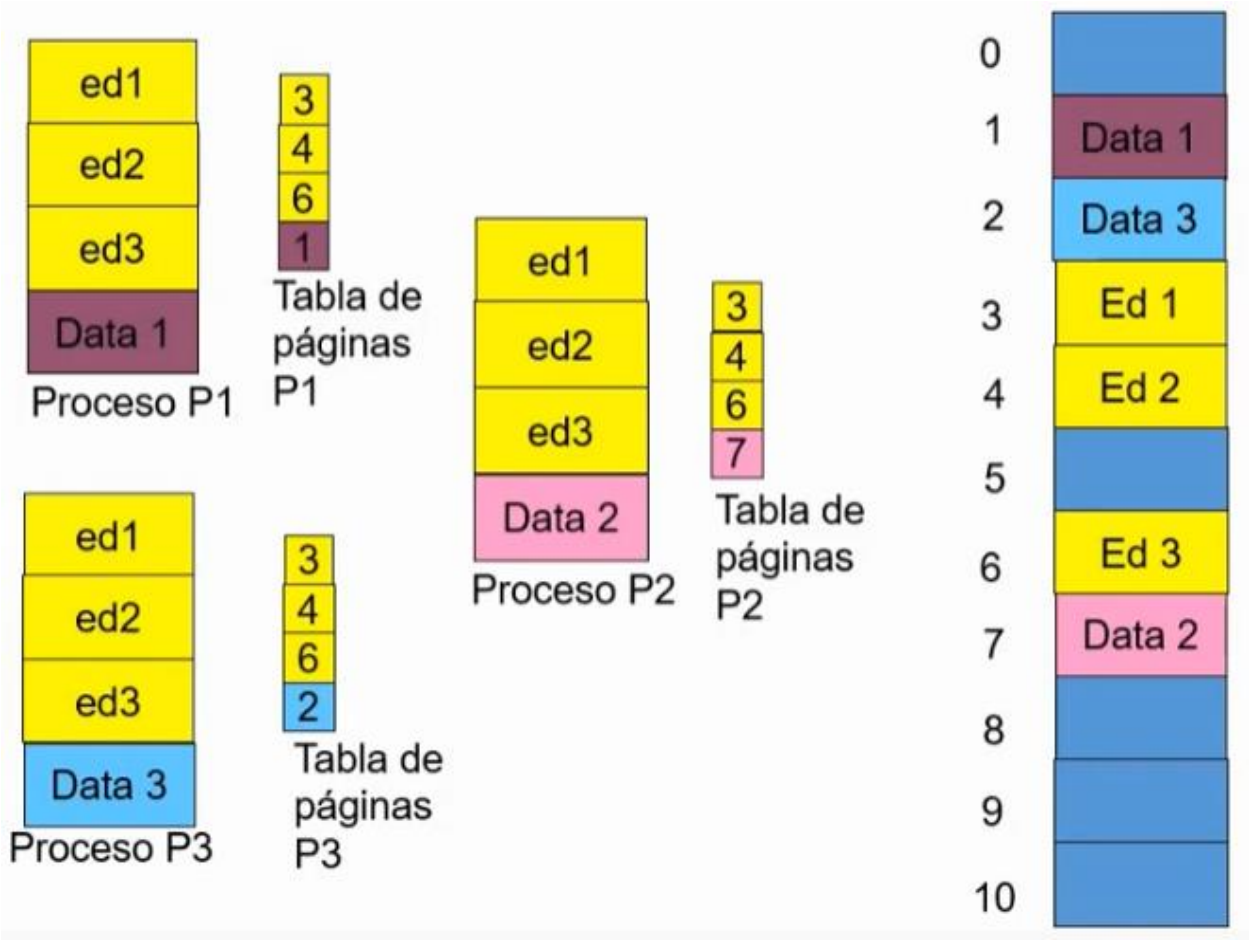
Tabla de pagina invertida



Páginas compartidas

- Código compartido
  - Una copia de código de solo lectura (reentrante) entre procesos
    - Editores de texto
    - Compiladores
    - Sistemas de ventanas
  - El código compartido debe aparecer en la misma posición en el espacio de direcciones lógico de todos los procesos
- Código y datos privados
  - Cada proceso mantiene una copia separada del código y datos
  - Las páginas para el código privado y datos pueden aparecer en cualquier parte en el espacio de direcciones lógico

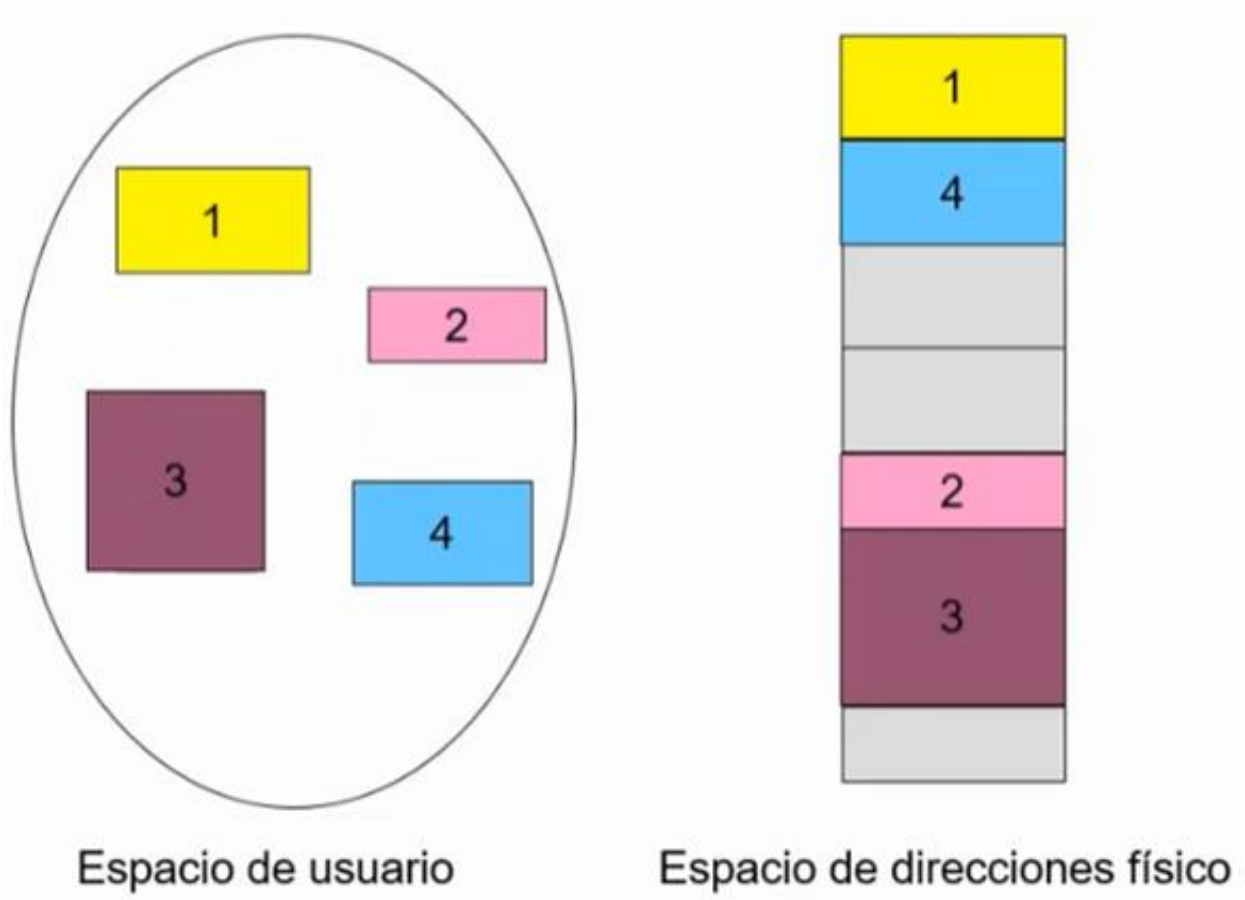
Ejemplo de páginas compartidas



Segmentación

- Esquema de manejo de memoria que es visible para el programador
- Un programa es una colección de segmentos.
- Un segmento es una unidad lógica como:
  - Código
    - Programa principal, funciones.
  - Stack
    - Para retorno de las llamadas a funciones, paso de parámetros y variables locales
  - Datos
    - Variables globales

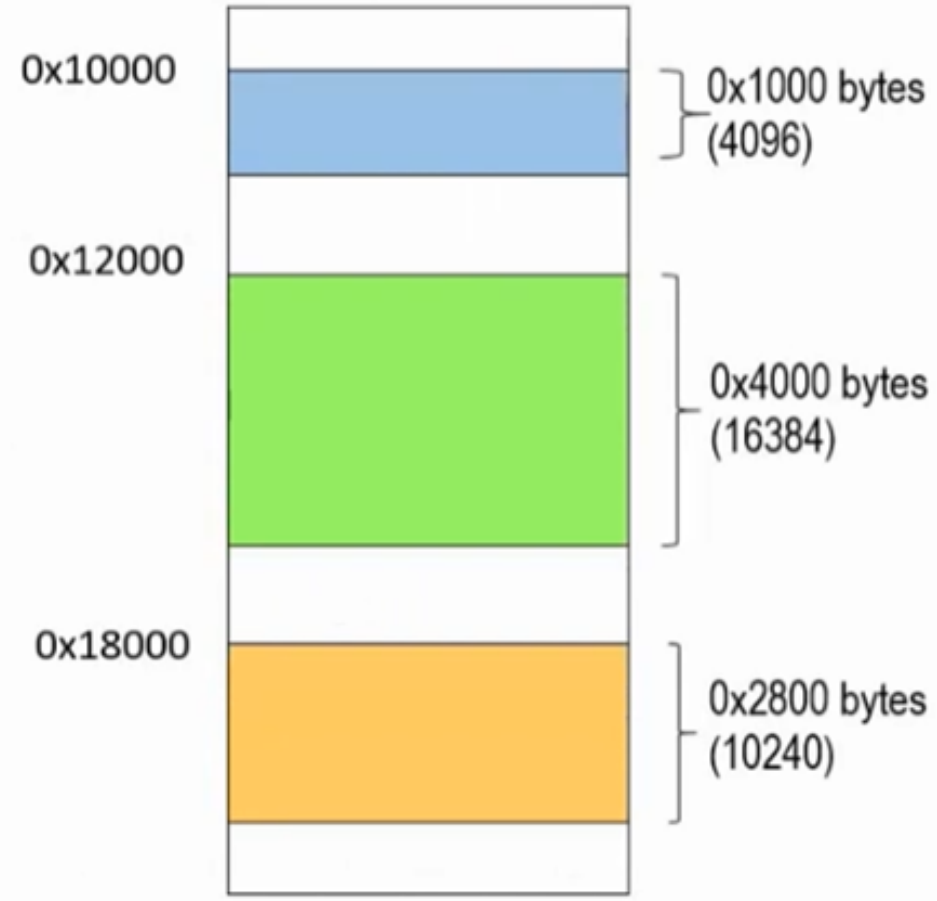
Visión lógica de la segmentación



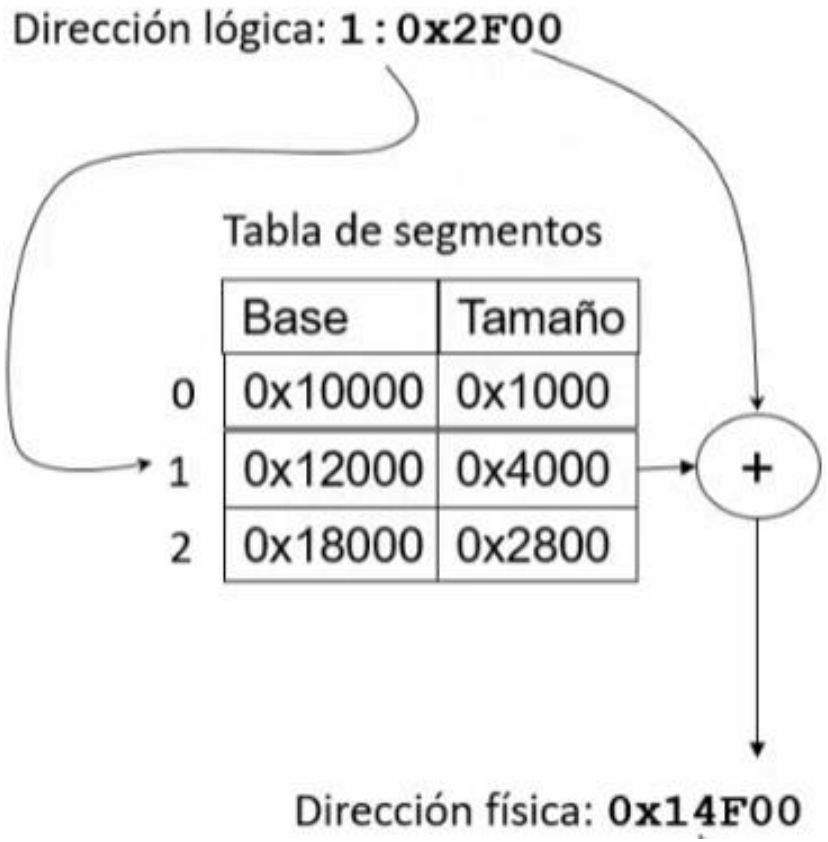
Traducción de direcciones lógicas a físicas en la segmentación

Tabla de segmentos

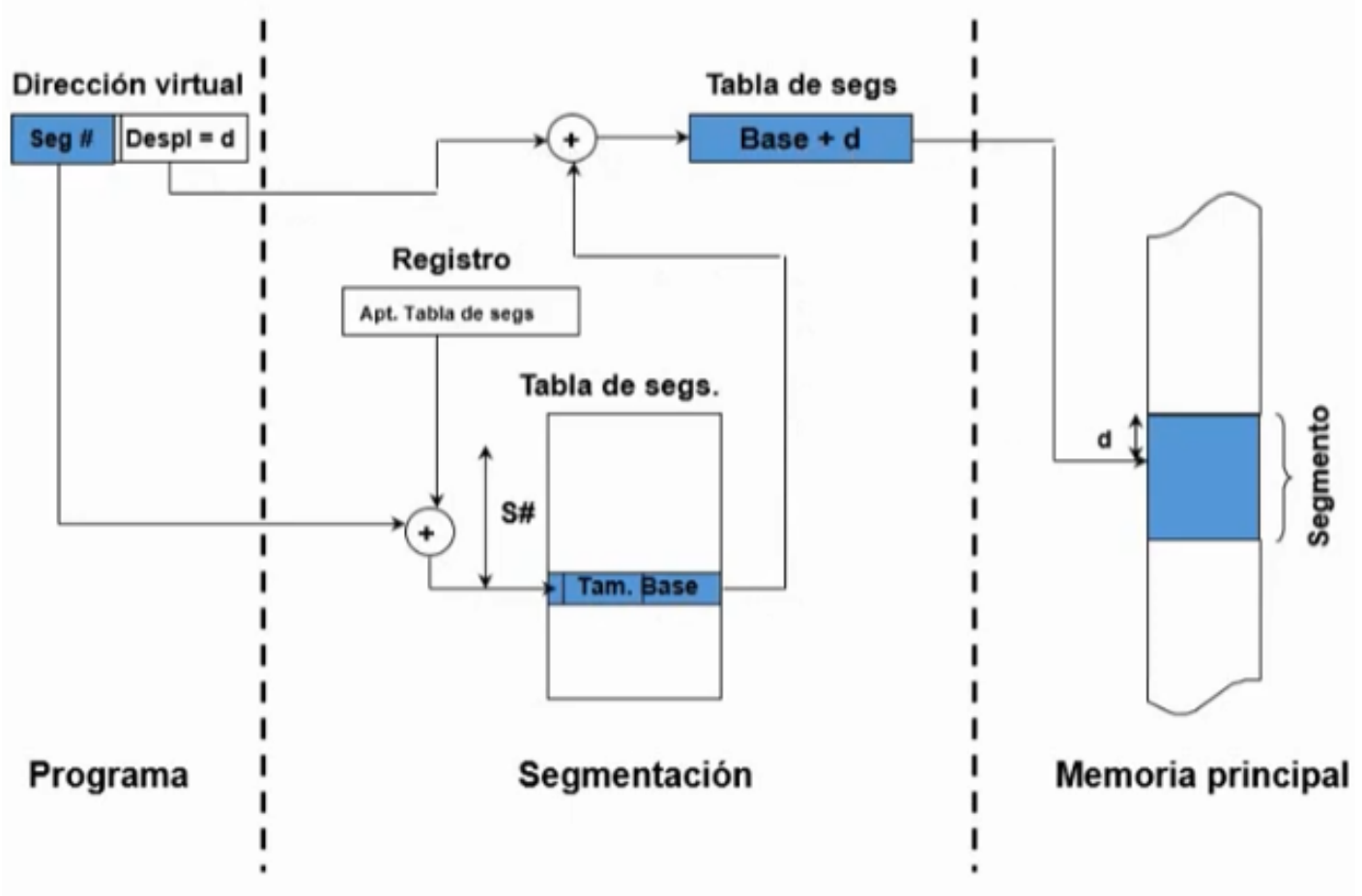
	Base	Tamaño
0	0x10000	0x1000
1	0x12000	0x4000
2	0x18000	0x2800



Traducción de direcciones lógicas a físicas en la segmentación



Traducción de direcciones en un sistema de segmentación

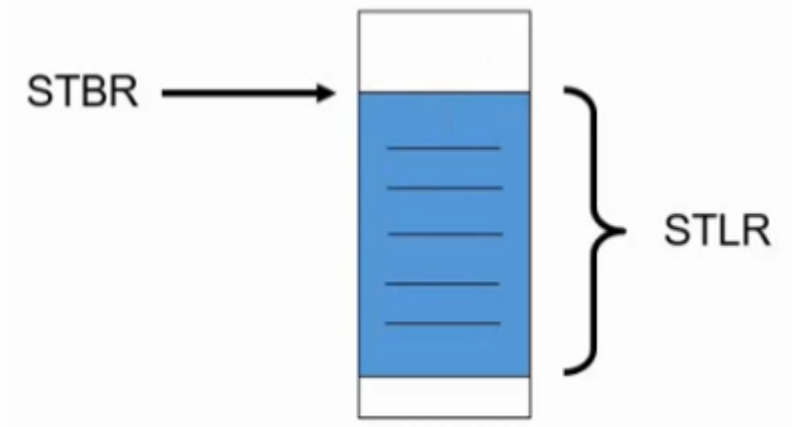


Arquitectura de la segmentación

- Una dirección lógica consiste de un par
  - número de segmento:desplazamiento
- *La tabla de segmentos*
  - Mapea de forma bi-dimensional direcciones físicas
  - Cada entrada de la tabla tiene:
    - **Base:** contiene la dirección física inicial donde reside el segmento en memoria.
    - **Límite:** Especifica el tamaño del segmento

Arquitectura de la segmentación

- *Segment-table base register (STBR)*
  - Apunta a la dirección de la tabla de segmentos en memoria.
- *Segment-table length register (STLR)*
  - Indica el número de segmentos usados por el programa.
  - La dirección es válida si el número de segmento<STLR



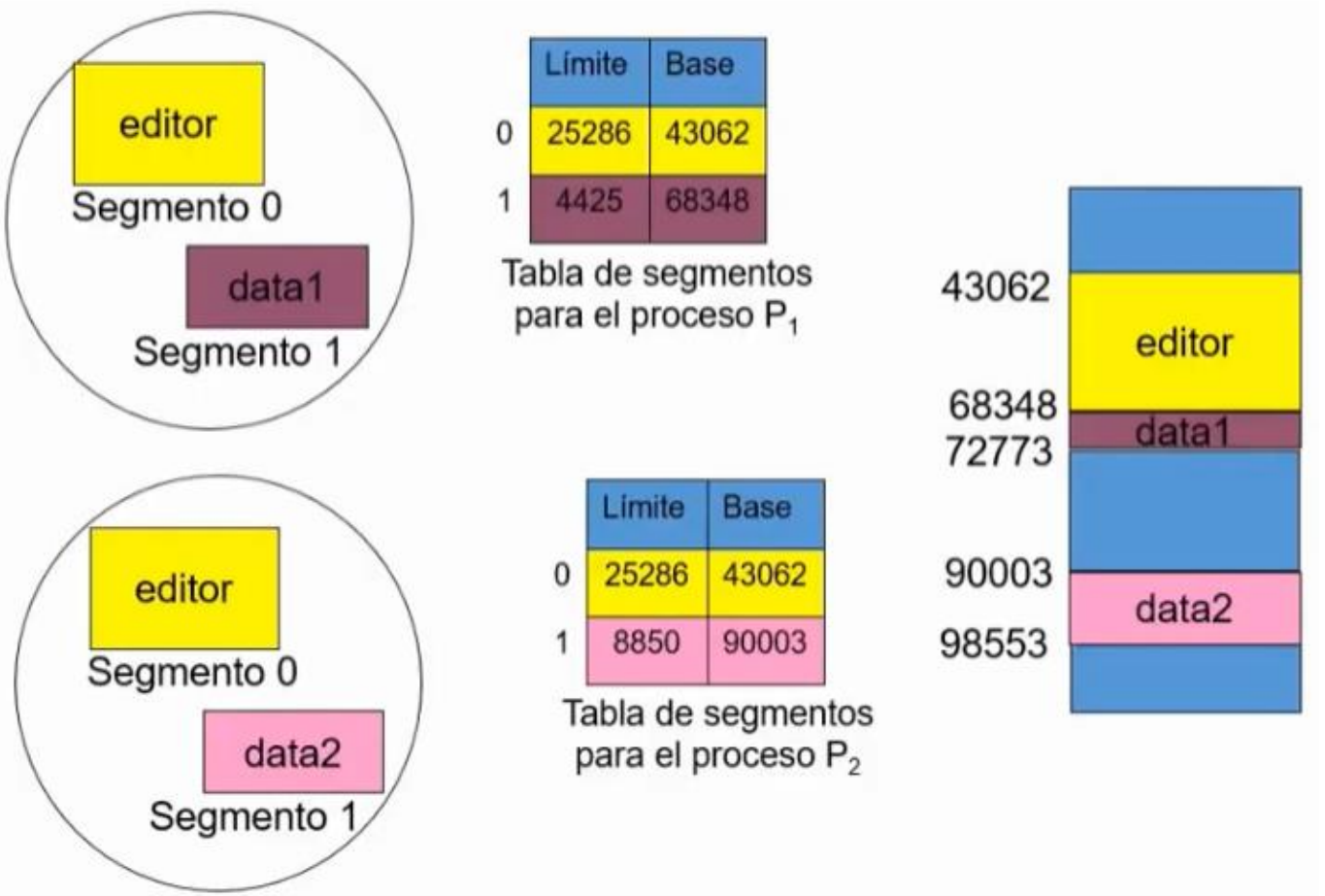
Arquitectura de la segmentación

- Recolocación.
  - Dinámica
  - Por tabla de segmentos
- Compartición.
  - Segmentos compartidos
  - El mismo número de segmento
- Asignación.
  - Primer ajuste/Mejor ajuste
  - Fragmentación externa

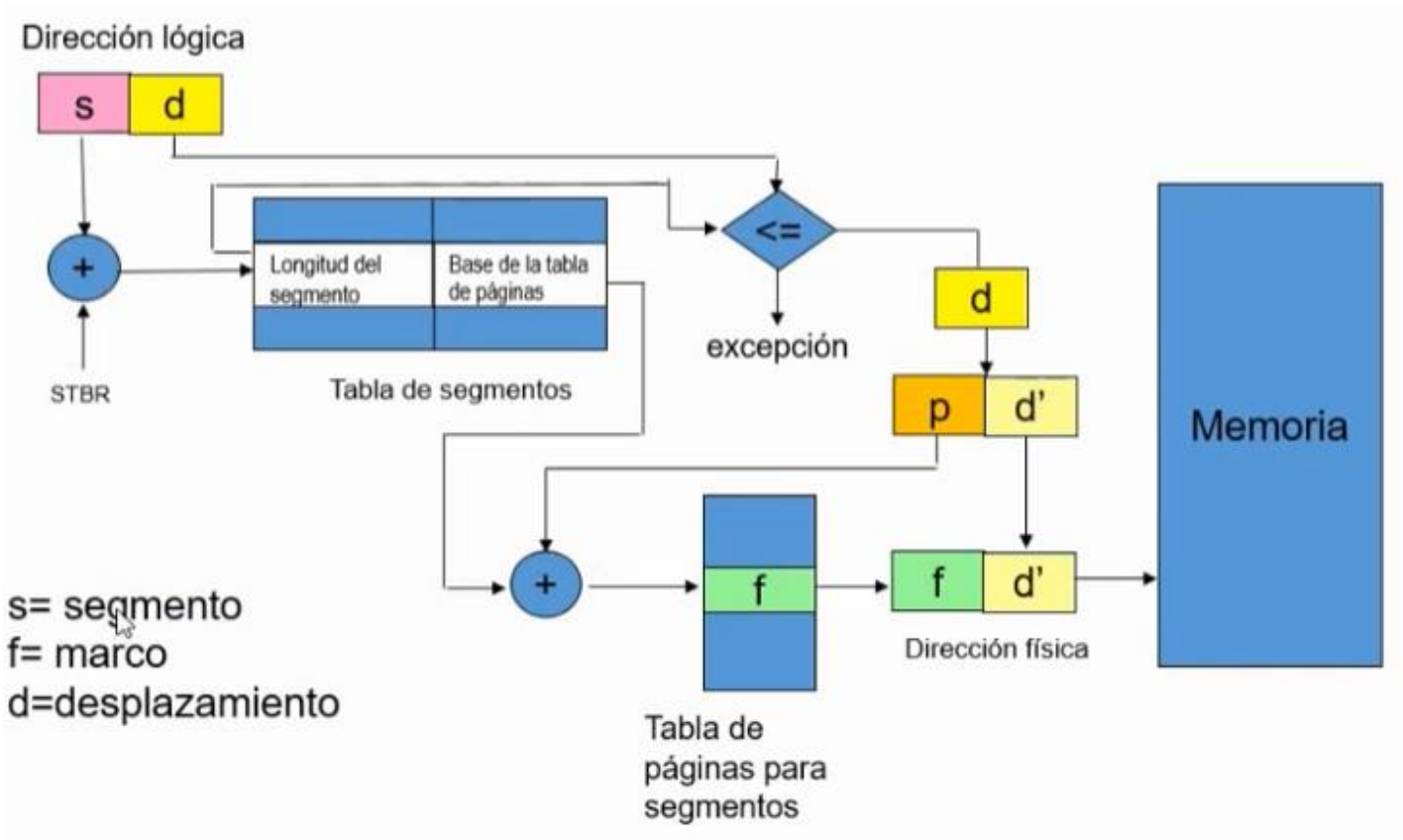
Arquitectura de la segmentación

- Bits de protección asociados a segmentos
  - La compartición de memoria ocurre al nivel de segmentación
- Ya que los segmentos varían en longitud, la asignación de memoria es un problema de asignación dinámica

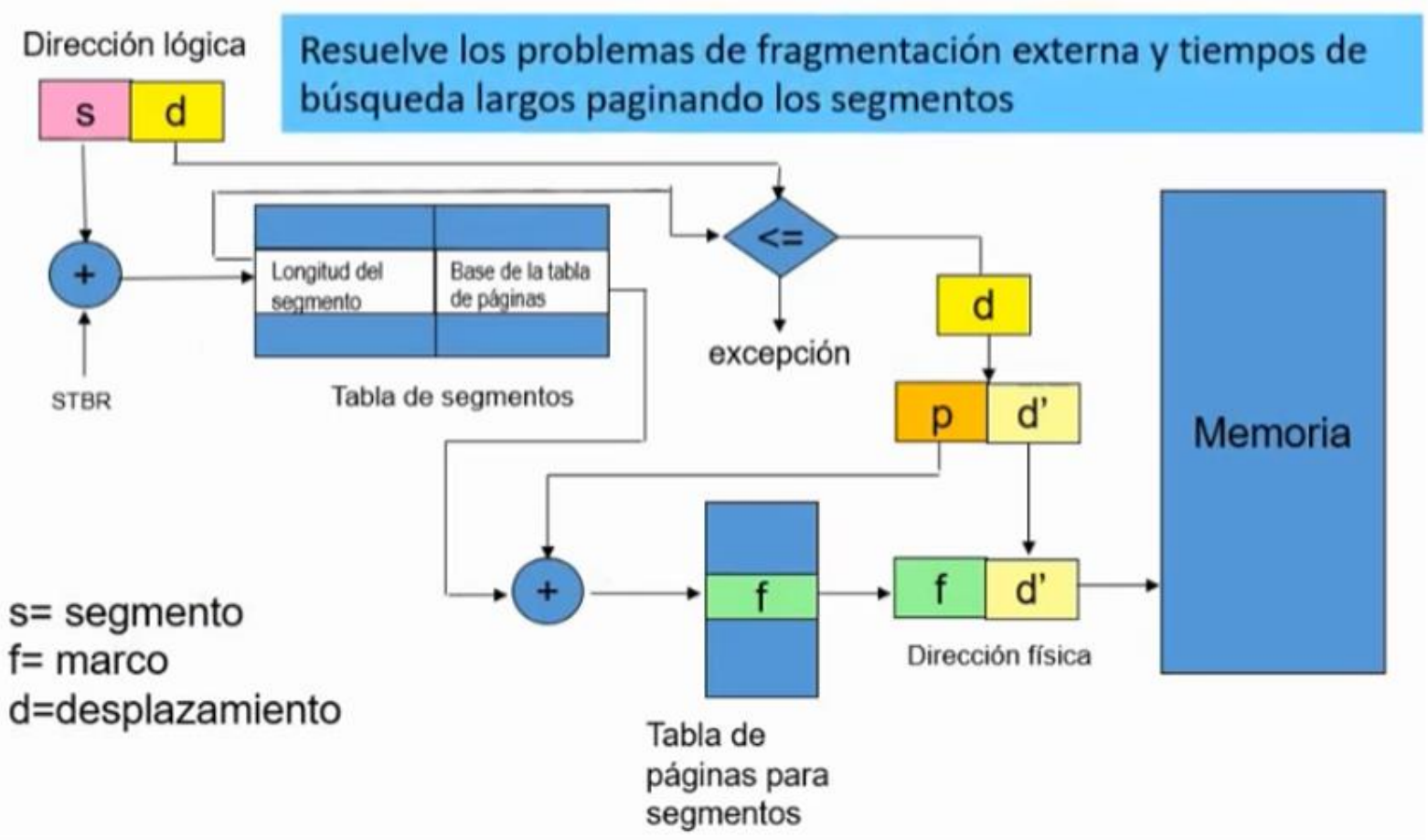
Compartición de segmentos



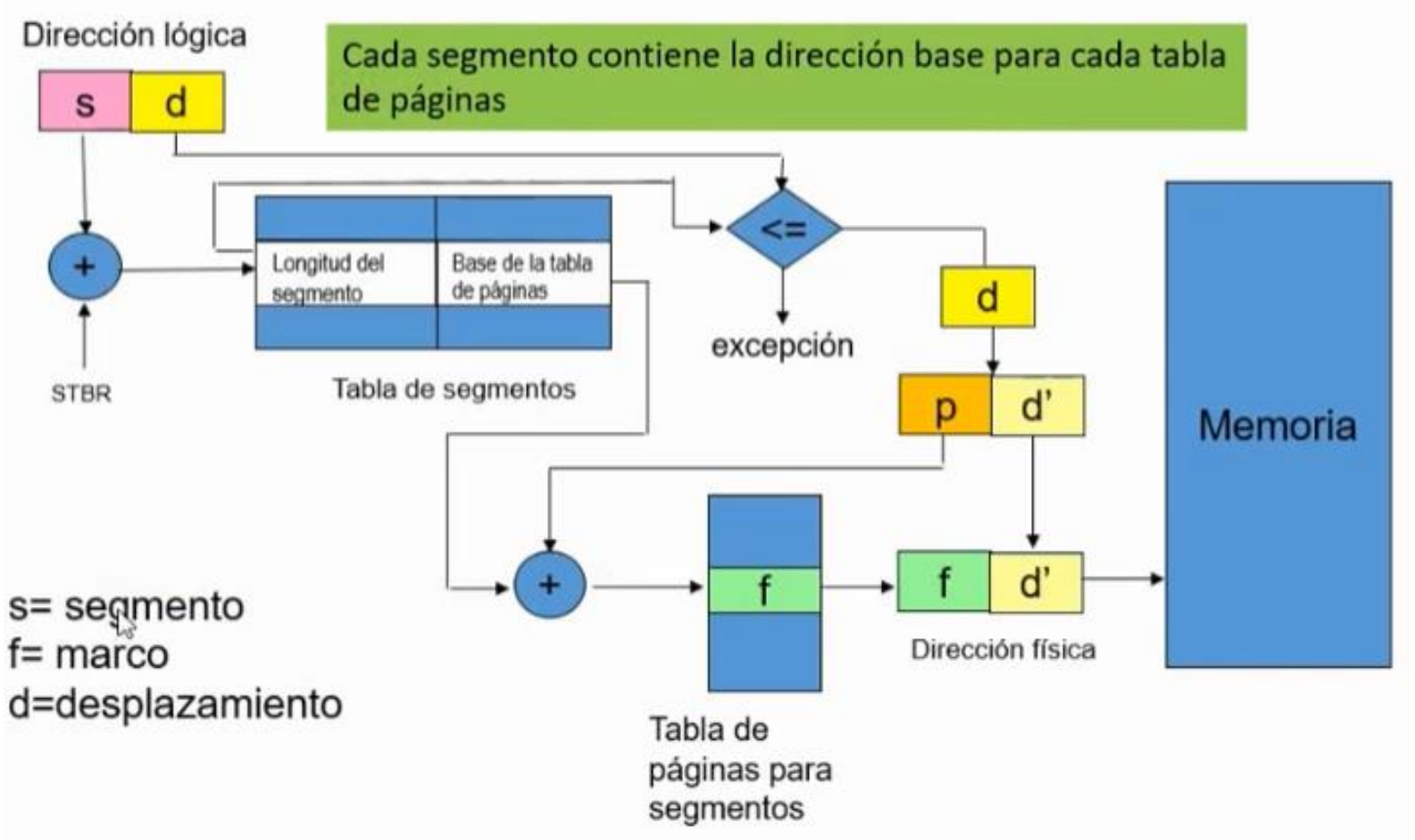
Esquema de traducción de direcciones de MULTICS



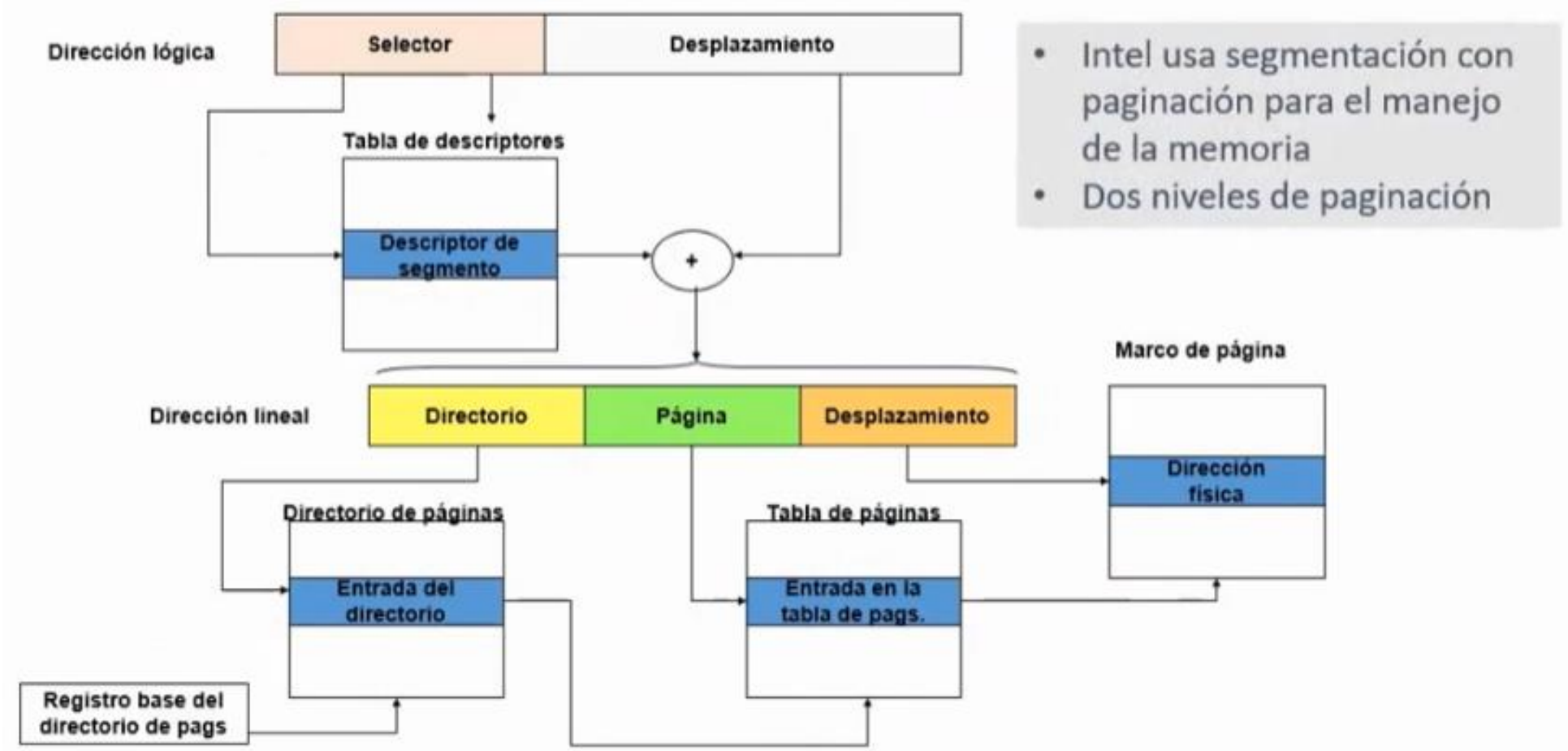
Esquema de traducción de direcciones de MULTICS



Esquema de traducción de direcciones de MULTICS



Intel, traducción de direcciones



- Intel usa segmentación con paginación para el manejo de la memoria
- Dos niveles de paginación

Memoria virtual

Introducción

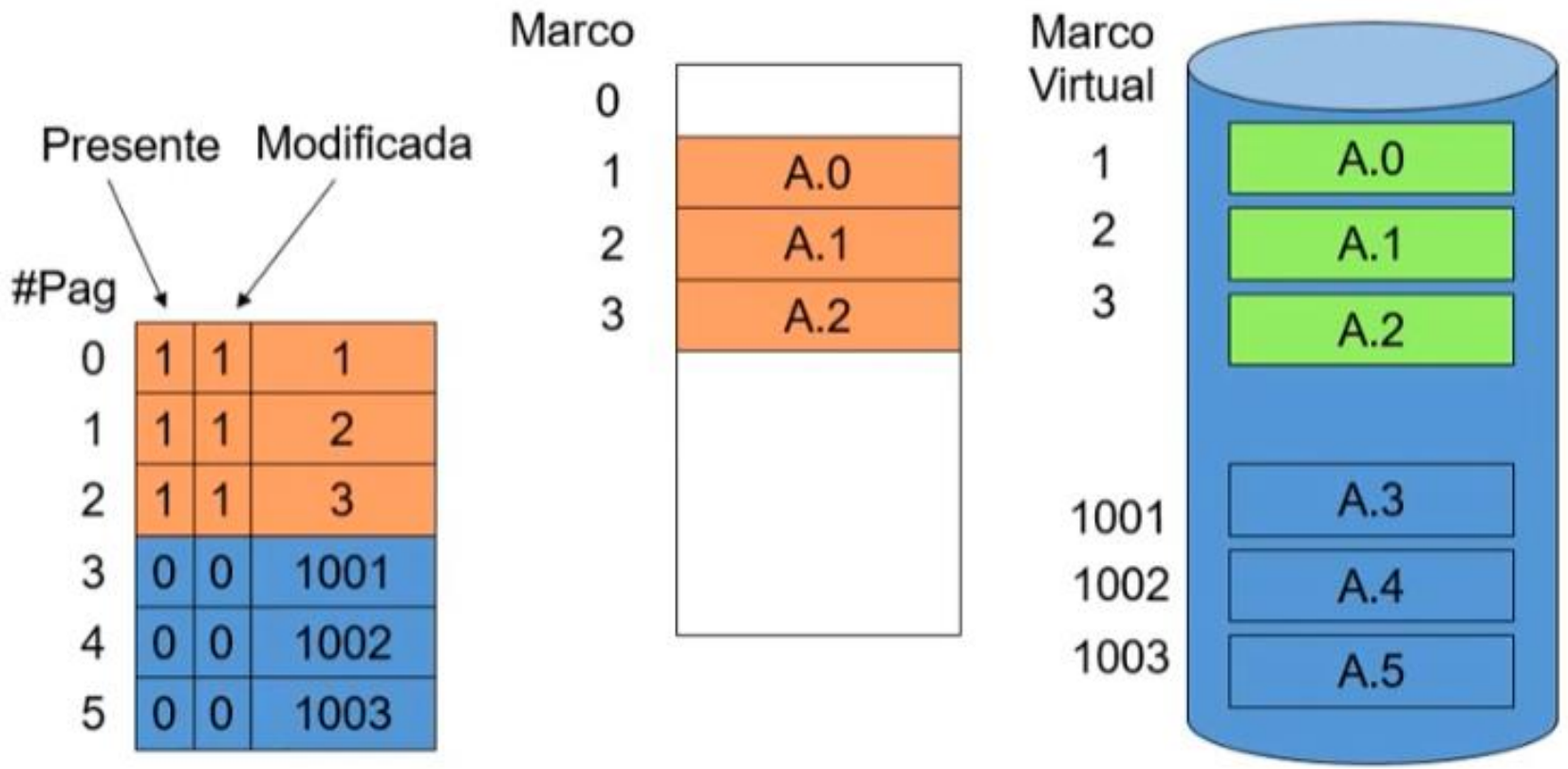
- Memoria virtual – separación de la memoria lógica de usuario de la memoria física.
- Solo una parte del programa se necesita que esté en memoria para su ejecución.
- El espacio de direcciones lógicos puede ser más grande que el espacio de direcciones físico.
- Se necesita permitir que las paginas salgan y entren. Memoria a disco – disco a memoria.

Introducción

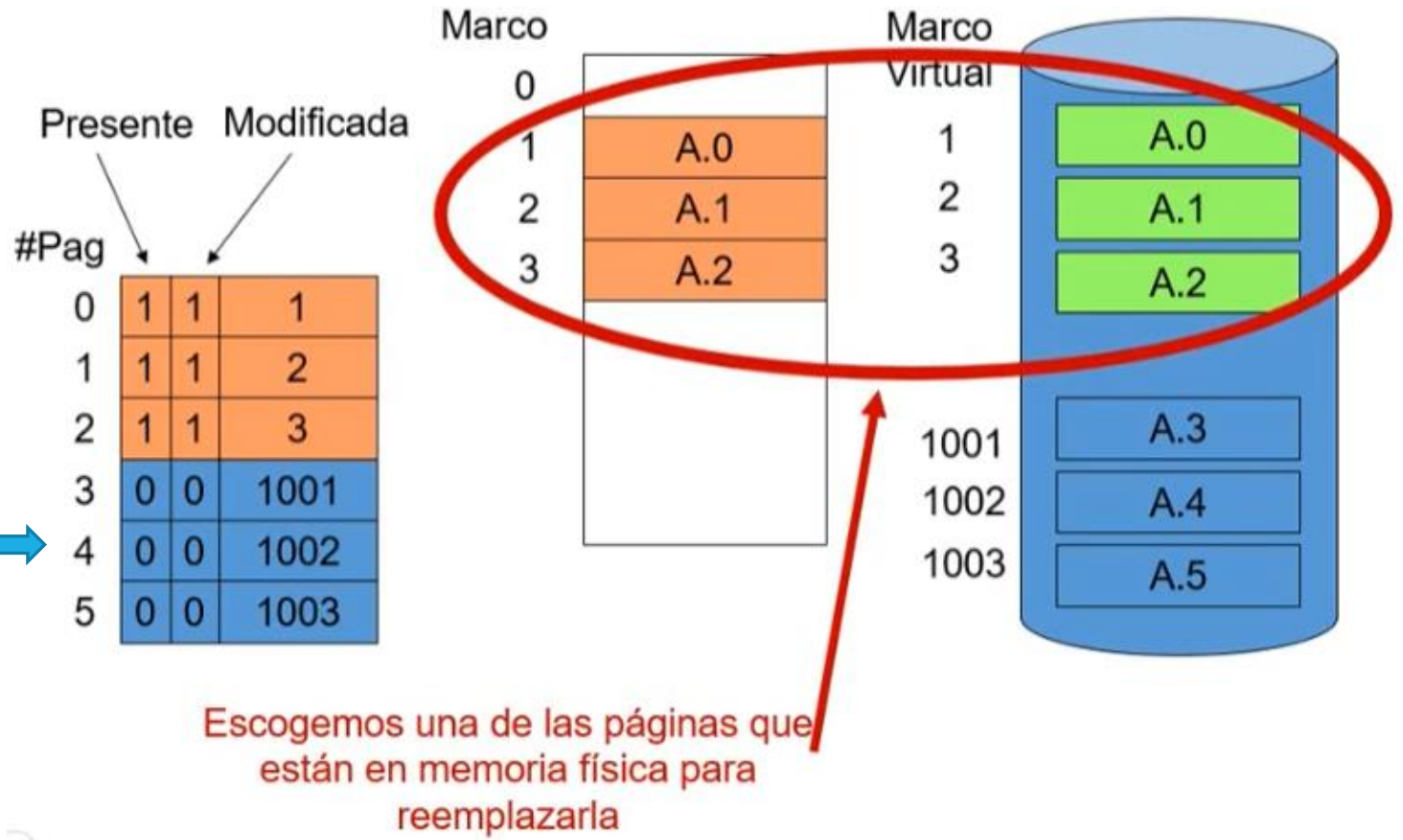
La memoria virtual puede implementarse vía:

- Paginación por demanda
- Segmentación por demanda

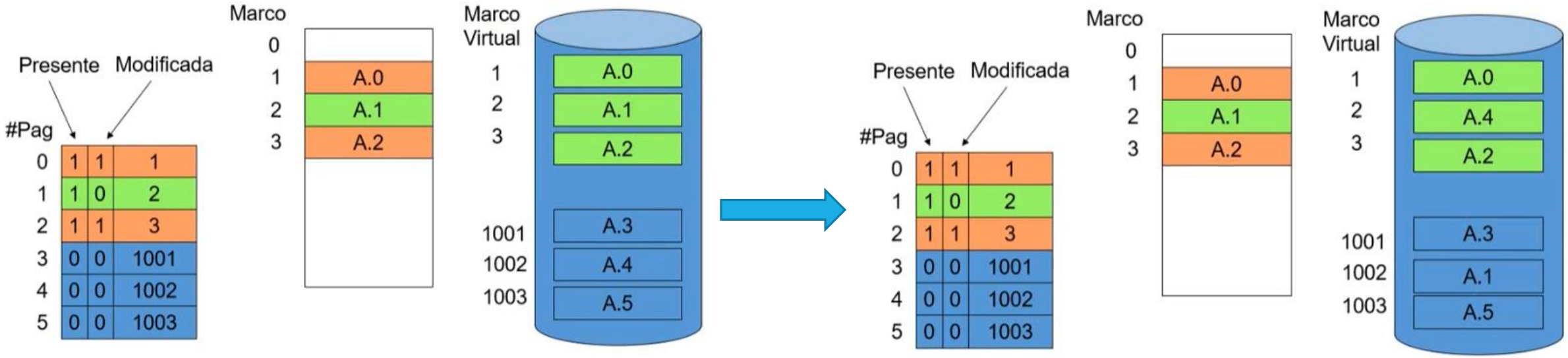
Memoria virtual

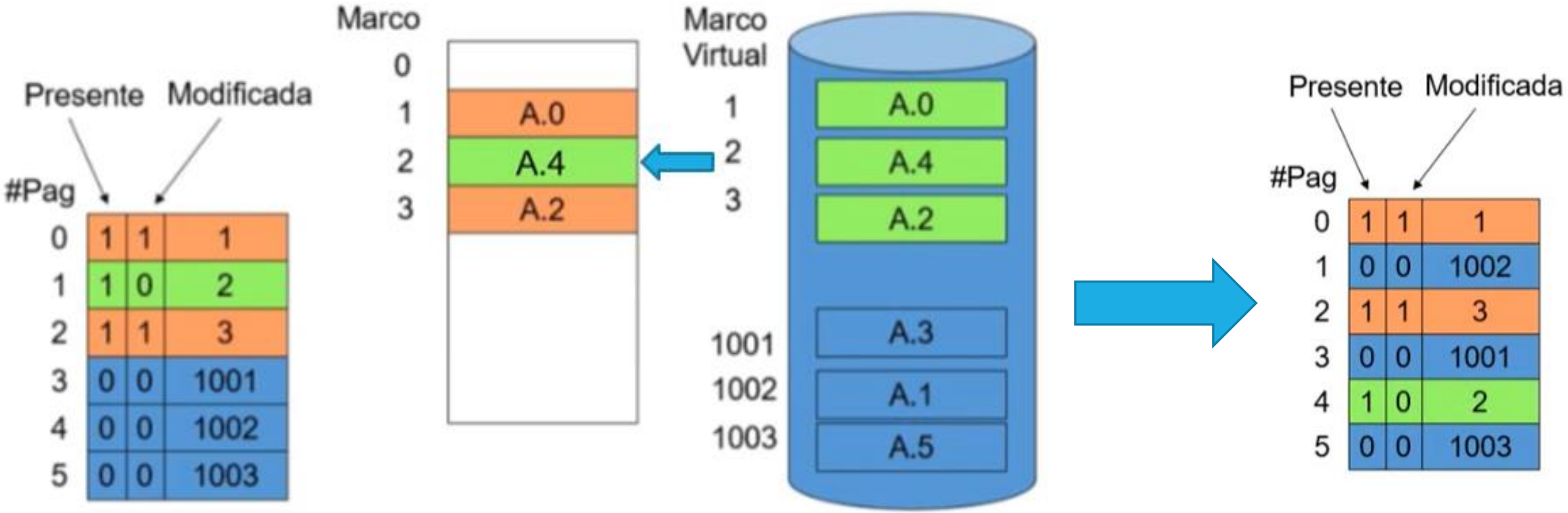


Se referencía una dirección que corresponde a la página 4



Memoria virtual




Presente
Modificada
#Pag
0 1 1 1
1 1 0 2
2 1 1 3
3 0 0 1001
4 0 0 1002
5 0 0 1003
Marco
0
1
2
3
A.0
A.4
A.2
Marco Virtual
1
2
3
1001
1002
1003
A.0
A.4
A.2
A.3
A.1
A.5
Presente
Modificada
#Pag
0 1 1 1
1 0 0 1002
2 1 1 3
3 0 0 1001
4 1 0 2
5 0 0 1003