

Fundamentos de los Sistemas Operativos (FSO)

Departamento de Informàtica de Sistemes y Computadoras (DISCA)
Universitat Politècnica de València

Bloque Temático 4: Gestión de Memoria
Unidad Temática 10

Asignación Dispersa de Memoria

fSO

DISCA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

- **Objetivos**

- Describir la **estrategia de asignación dispersa** de memoria
- Estudiar las **técnicas** básicas de asignación dispersa:
 - **Paginación**
 - **Segmentación**
- Comprender la paginación multinivel y su aplicabilidad

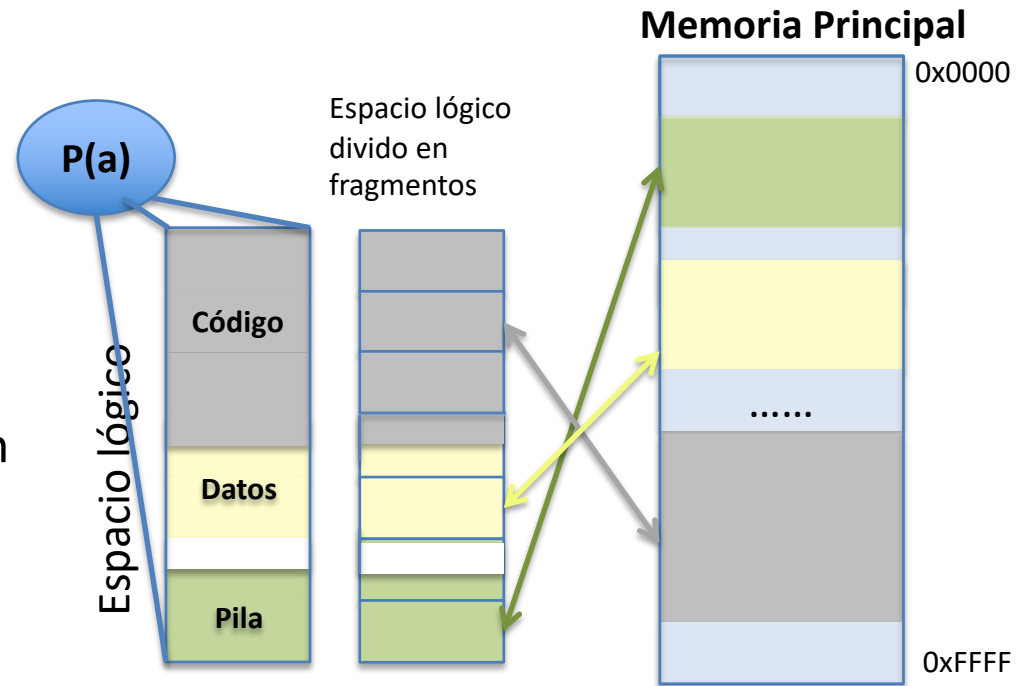
- **Bibliografía**

- “Fundamentos de sistemas operativos” A. Silberschatz 7ª Ed, capítulo 8
- “Sistemas operativos: una visión aplicada” Jesús Carretero 2º Ed, capítulo 5

- **Concepto de Asignación Dispersa**
- Paginación
- Segmentación
- Paginación Multinivel

- **Asignación dispersa**

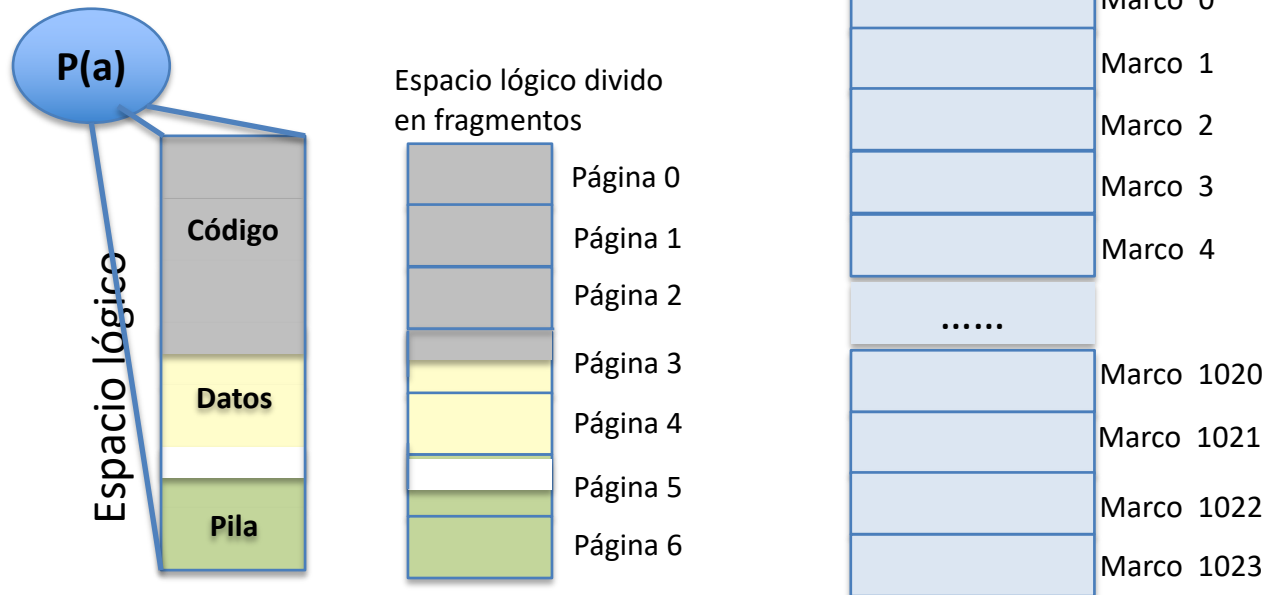
- Considera el espacio de direcciones lógicas de un proceso dividido en **fragmentos**
- Cada fragmento se ubica en memoria física de forma independiente



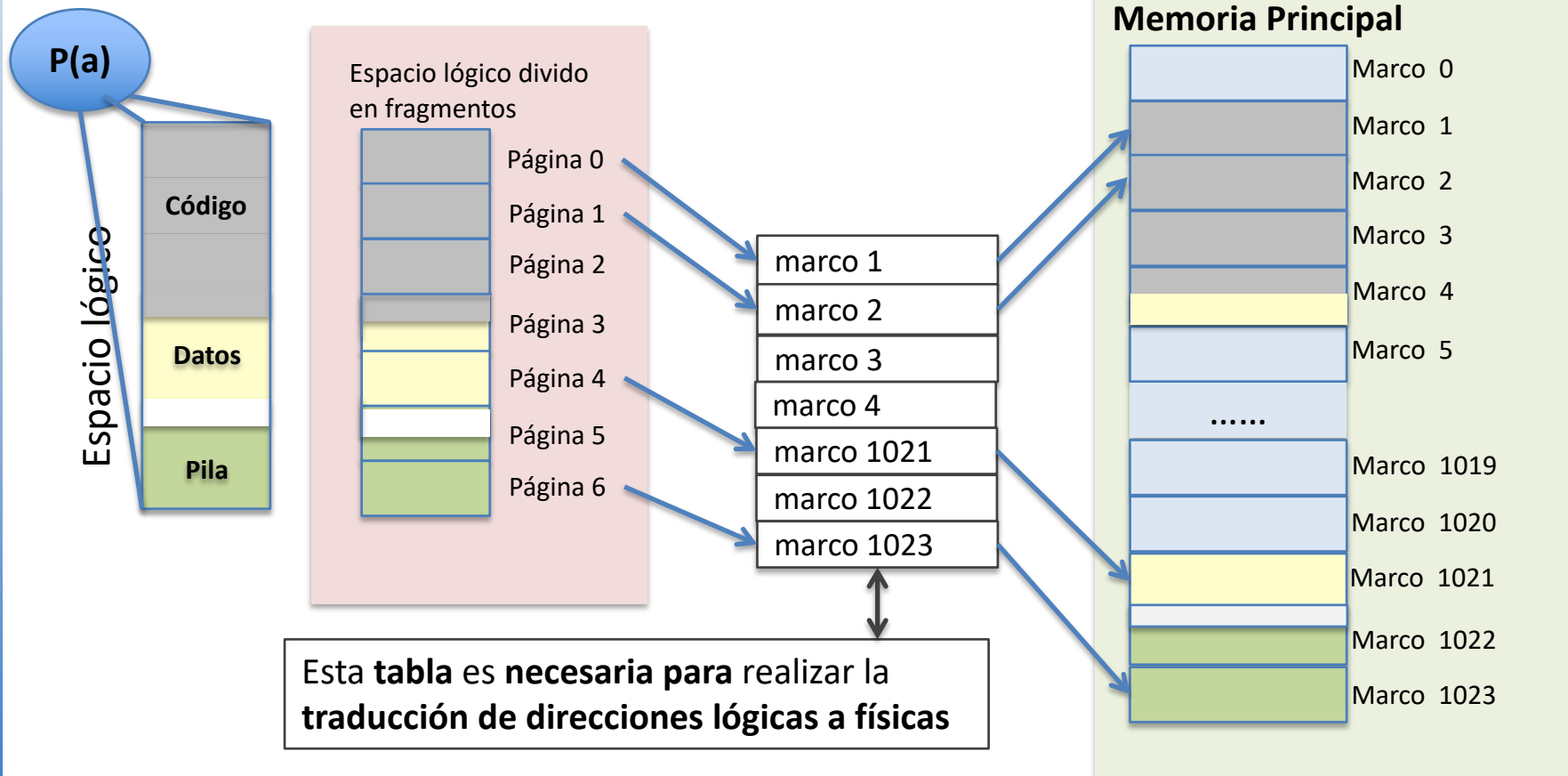
- El espacio de direcciones físicas de un proceso puede no ser contiguo:
 - **Paginación:** los fragmentos son de tamaño fijo
 - **Segmentación:** los fragmentos son de tamaño variable
- La MMU necesita conocer la ubicación de cada fragmento y su tamaño para traducir direcciones lógicas a direcciones físicas,
 - **Tabla de páginas**
 - **Tabla de segmentos**

- Concepto de Asignación Dispersa
- **Paginación**
- Segmentación
- Paginación Multinivel

- **Esquema de gestión de memoria con paginación**
 - El espacio de direcciones físicas en memoria principal de un proceso puede ser no contiguo
 - Los espacios de direcciones de un proceso los considera dividido en fragmentos de tamaño fijo
 - El espacio de **direcciones lógicas** → dividido en **Páginas**
 - El espacio de **direcciones físicas** → divididos en **Marcos** o frames



- Cuando se va a ejecutar un proceso el sistema **carga** todas **sus páginas en marcos** de memoria principal
 - Cada página ocupa un marco
 - Se construye **una tabla** para almacenar el **número de marco** donde se ha ubicado cada página del proceso
 - Cada proceso tiene asociada una **tabla de páginas**



- **Tabla de Páginas**

- Cada entrada de la tabla se denomina **descriptor de página** y contiene

- El **número de marco** donde se ha ubicado la página
- Un conjunto de bits: **bit de validez**, **bits de protección**, **bit de modificado**,

| | | | | |
|-----------------------|-----------------|-------------|----------------|---------------------|
| Descriptor Página 0 | Número de marco | Bit validez | Bit modificado | Bits Protección rwx |
| Descriptor Página 1 | Número de marco | Bit validez | Bit modificado | Bits Protección rwx |
| | | | | |
| Descriptor Página n-1 | Número de marco | Bit validez | Bit modificado | Bits Protección rwx |

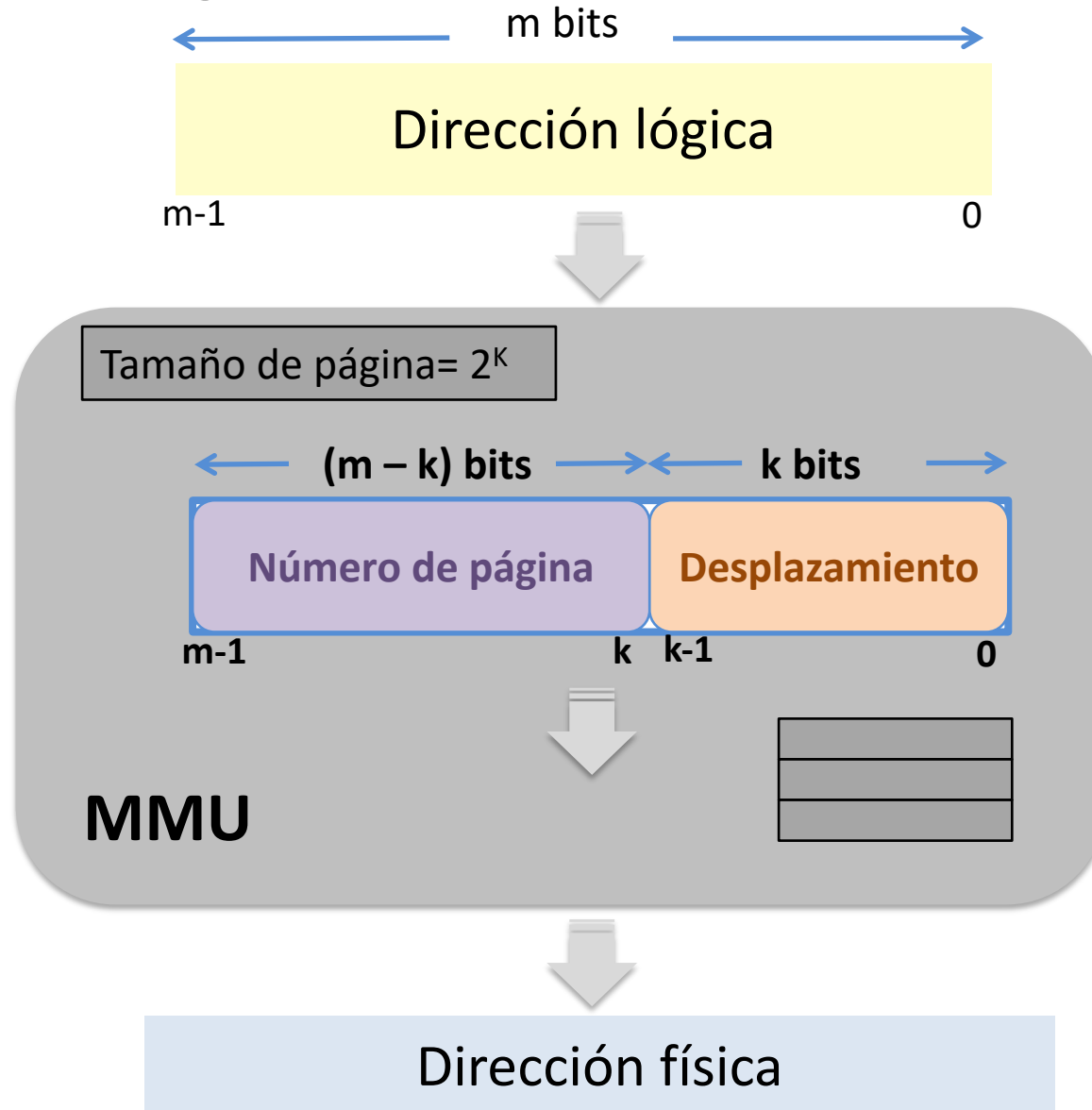
Número de marco de Memoria Principal donde esta cargada la página

Indica **si** la **página** es **válida** para el proceso

Indica si la **página** ha sido **modificada**

Indica el tipo de **acceso permitido** a la página

- Paginación: Estructura de la dirección lógica**

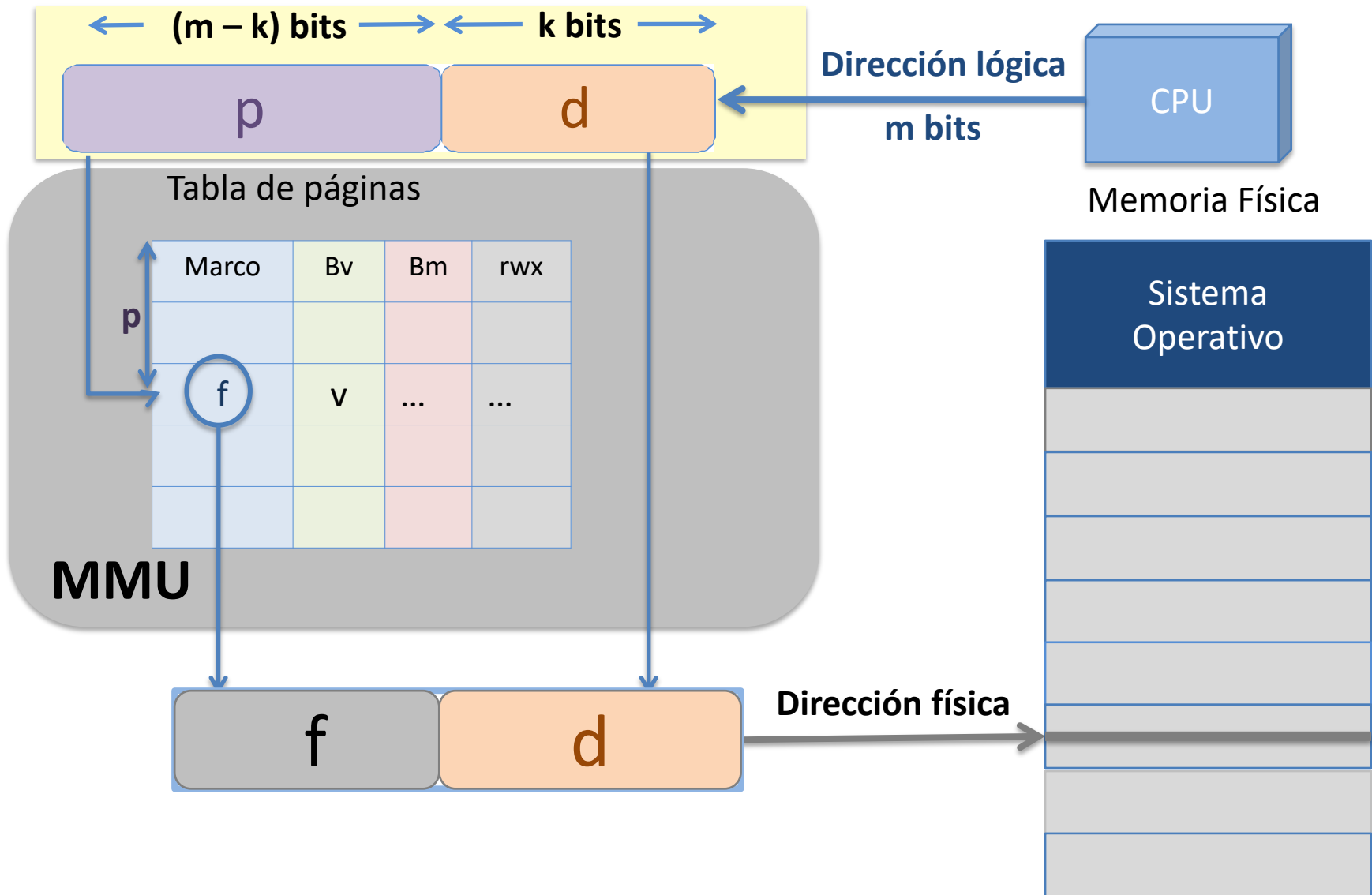


Espacio lógico de 2^m palabras

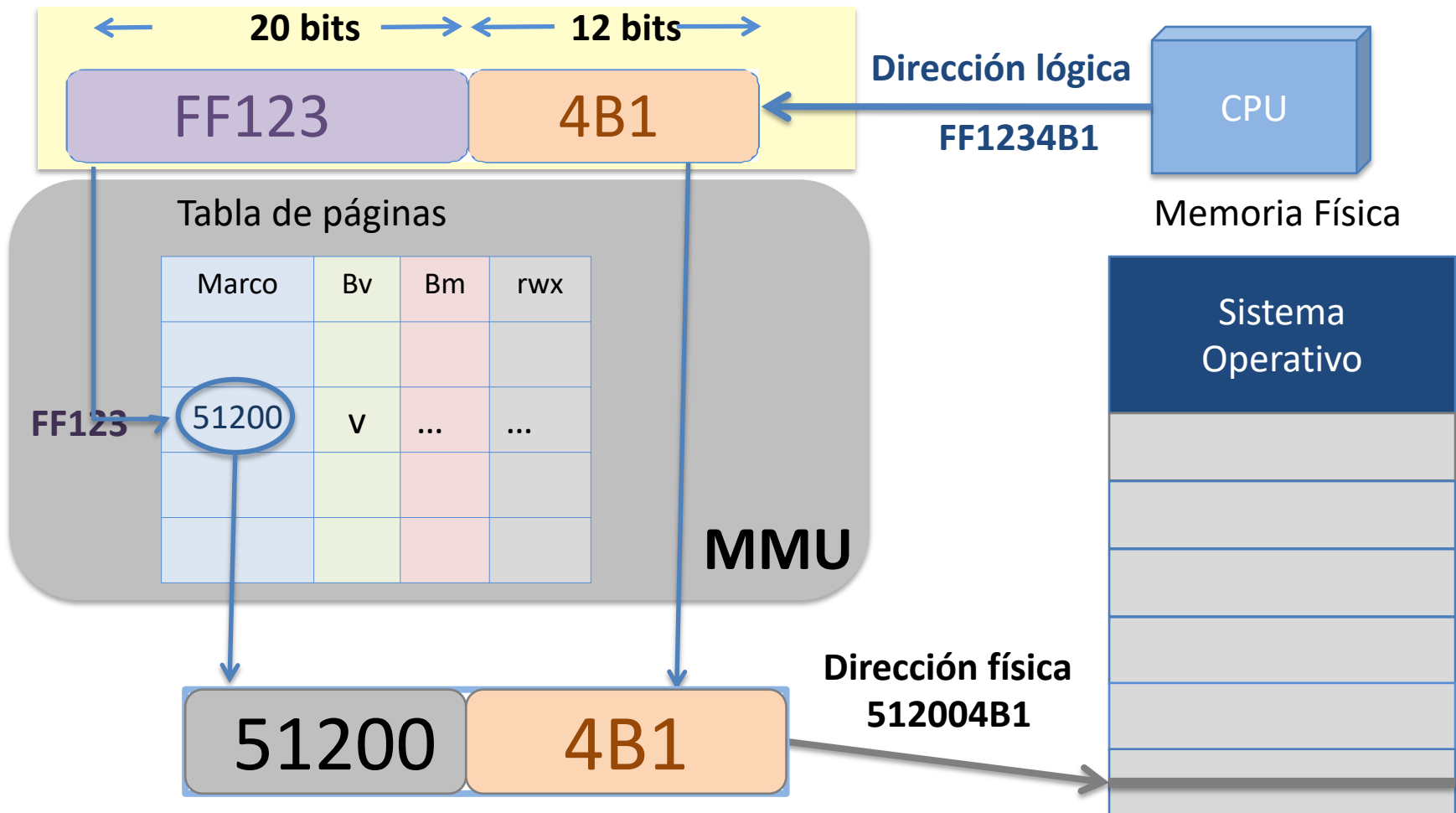
Sistema con **páginas de 2^k palabras** y espacio lógico de 2^m palabras

La **MMU** ve el espacio lógico formado por $2^{(m-k)}$ páginas

- **Paginación: Traducción de direcciones**



- **Ejemplo paginación micro 32 bits.**
 - Tamaño página 4K: $k=12$ bits ($2^{12}=4096$)
 - Páginas = $2^{20} = 1M = 1048576$ ($m-k = 32-12=20$ bits)
 - Dirección lógica = **FF1234B1**



- **Paginación: Análisis de eficiencia**

- Ventajas

- **No** aparece **fragmentación externa**
 - Facilita la reubicación
 - Proporciona protección
 - Compartición de páginas de código entre procesos

- Inconvenientes

- Fragmentación interna

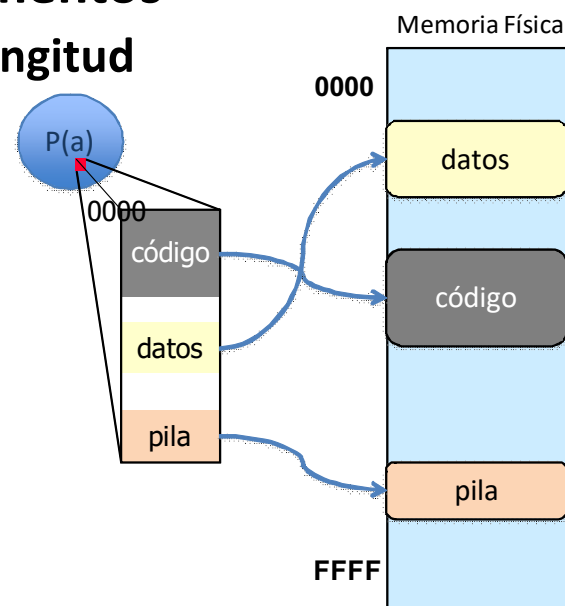
- Tamaños de página

- El **tamaño de página debe ser potencia de 2** para facilitar la obtención del número de página y desplazamiento
 - Tamaños actuales de páginas 4K,8K
 - Páginas de tamaño grande: mucha fragmentación interna
 - Páginas de tamaño pequeño: tablas de páginas muy grandes

- **Implementación de Tabla de páginas en:**
 - **Registros de la MMU**
 - Sólo para espacios lógicos muy pequeños (pocas páginas)
 - **Memoria**
 - Se mantiene un registro (PTBR=registro base tabla de páginas) con la dirección física base en la que empieza la tabla de páginas en memoria
 - Necesario un acceso extra a memoria principal para acceder a la tabla de páginas y traducir una dirección lógica
 - **TLB (*translation look-aside buffer*)**
 - La TLB contiene sólo unas pocas entradas de la tabla de páginas (las de uso reciente) de la tabla de páginas
 - Acceso más rápido que una implantación en memoria
 - Tasa de éxito alta con pocas entradas

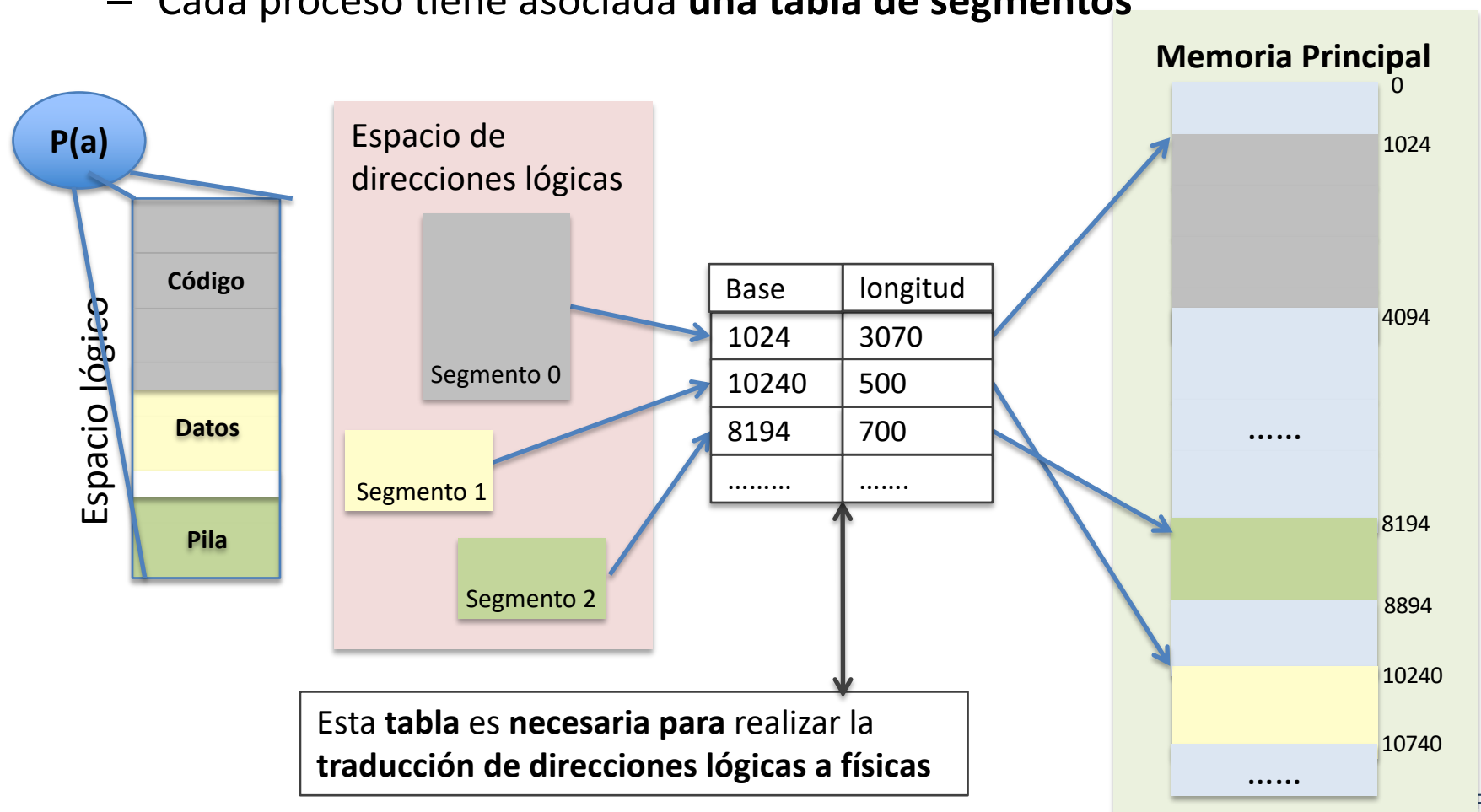
- Concepto de Asignación Dispersa
- Paginación
- **Segmentación**
- Paginación Multinivel

- **Esquema de gestión de memoria con Segmentación**
 - El espacio de direcciones físicas en memoria principal de un proceso puede ser no contiguo
 - Espacio de direcciones dividido en **fragmentos de longitud variables** → **Segmento**
 - El espacio lógico es un conjunto de **segmentos**
 - Cada segmento tiene **un nombre y una longitud**
 - El programa es compilado y automáticamente el compilador construye uno o varios segmentos reflejando el contenido del programa fuente
 - **Segmento de Código**
 - **Segmento de Datos**
 - **Segmento de Pila**



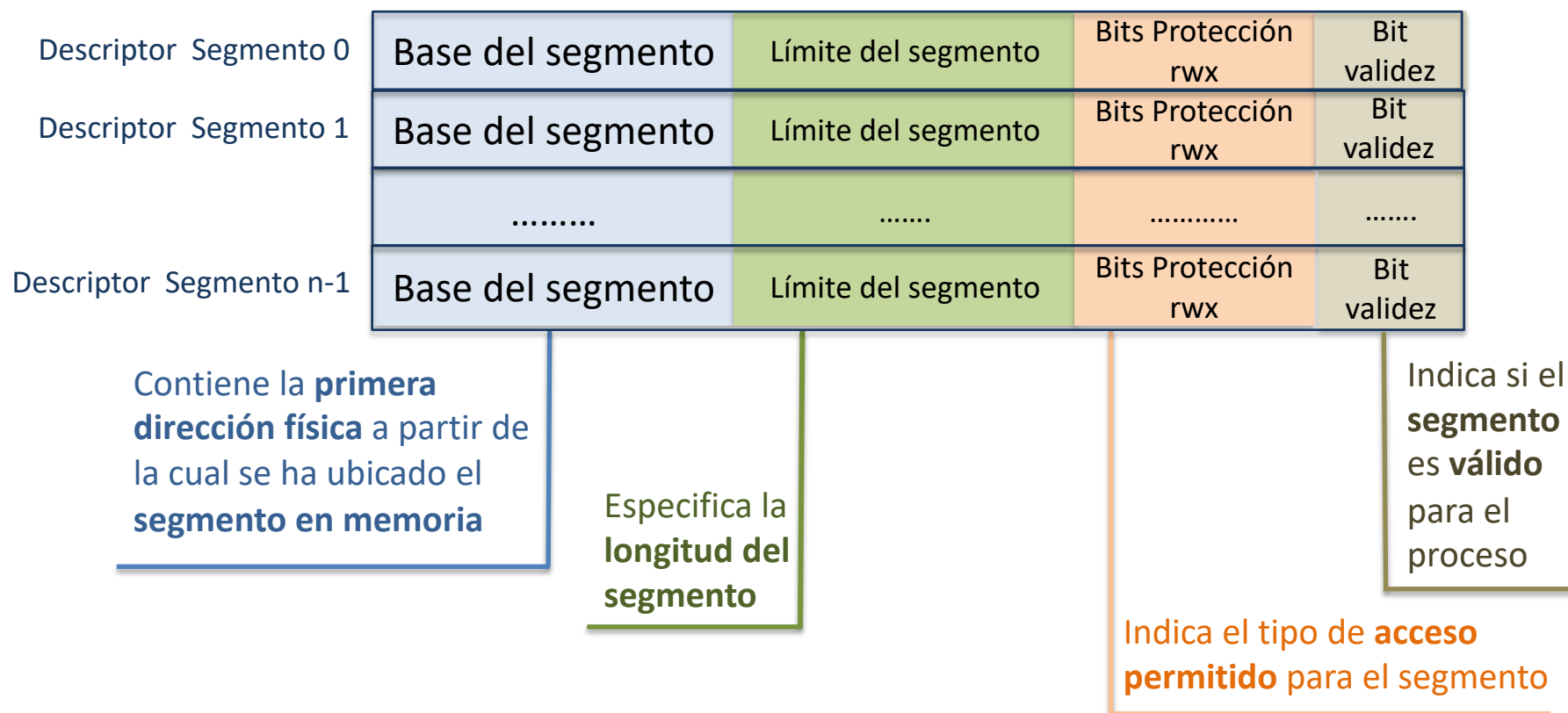
Un segmento siempre se ubica de forma contigua en memoria principal

- Cuando se va a ejecutar un proceso el sistema **carga** todos **sus segmentos en** memoria principal
 - Se construye **una tabla** para almacenar la **dirección física base de cada segmento y su longitud**
 - Cada proceso tiene asociada **una tabla de segmentos**

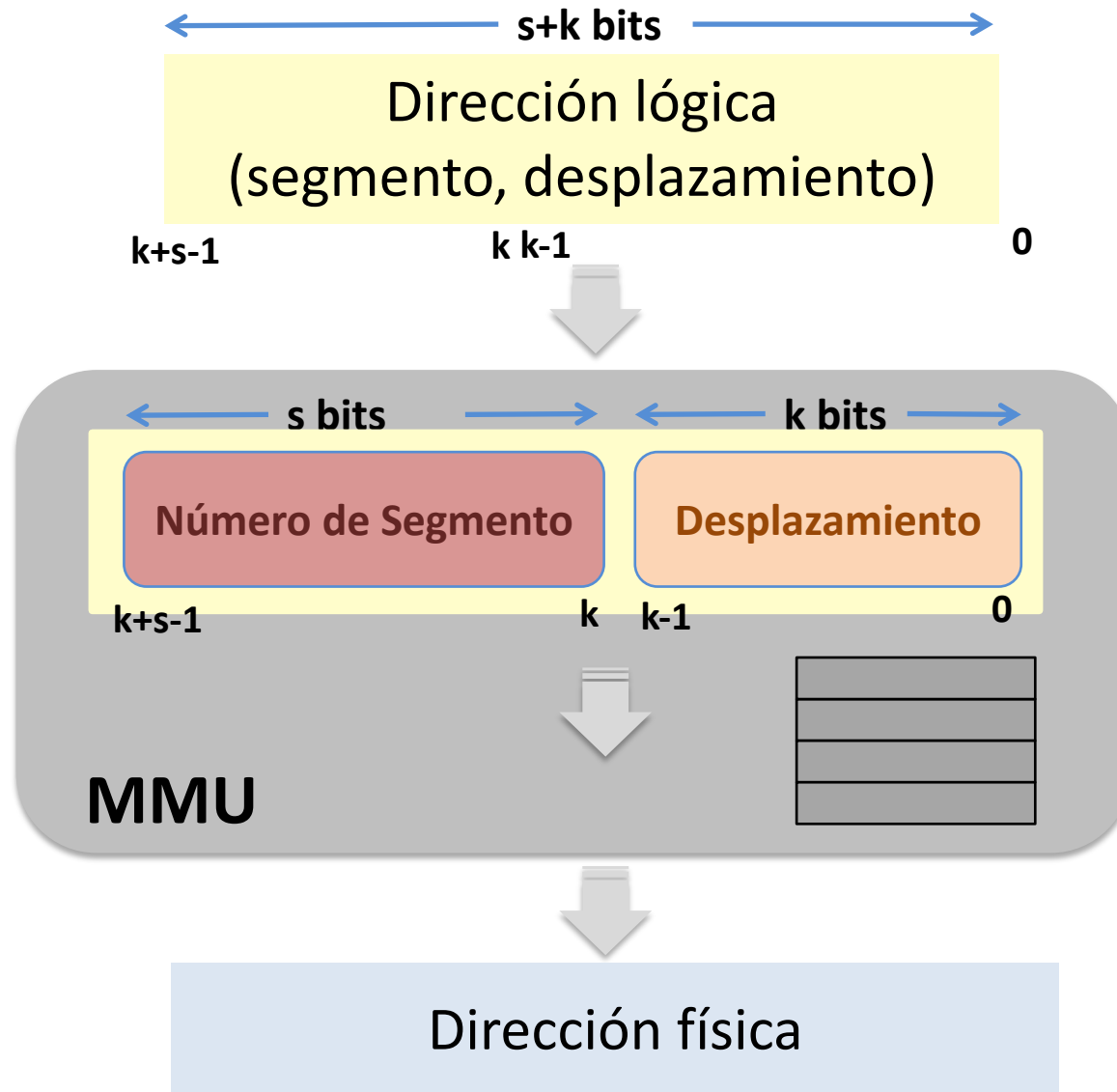


• Tabla de Segmentos

- Cada entrada de la tabla se denomina **descriptor de segmentos** y contiene:
 - **Dirección física base** donde se ha ubicado el segmento
 - Tamaño del segmento
 - Un conjunto de bits: **bits de protección, bit de validez**



- Segmentación: Estructura de la dirección lógica

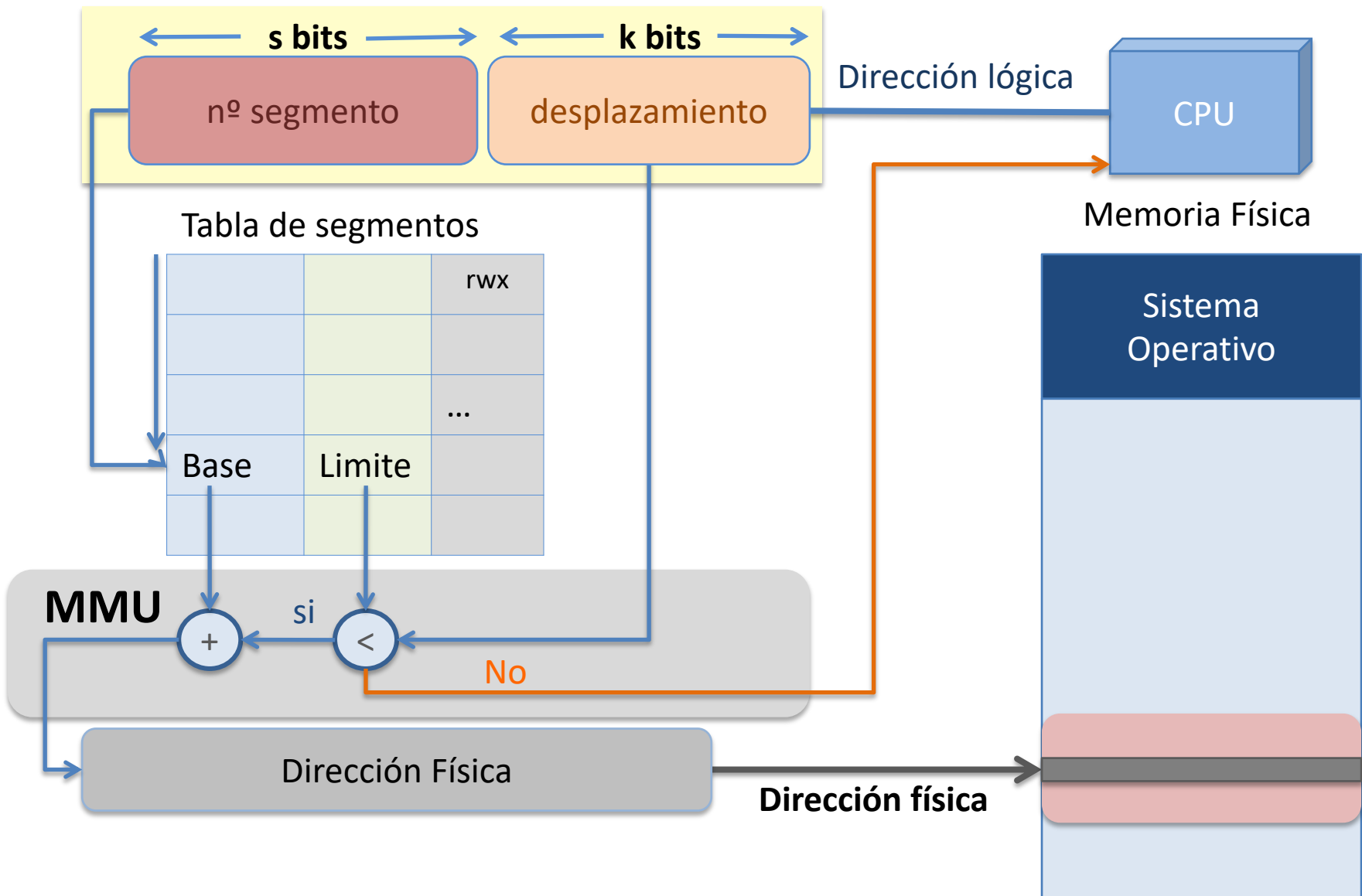


Dirección lógica formada por **segmento, desplazamiento**. Hay 2^s segmentos y cada segmento puede tener 2^k palabras

La **MMU** ve el **espacio lógico** formado por 2^s segmentos de 2^k palabras

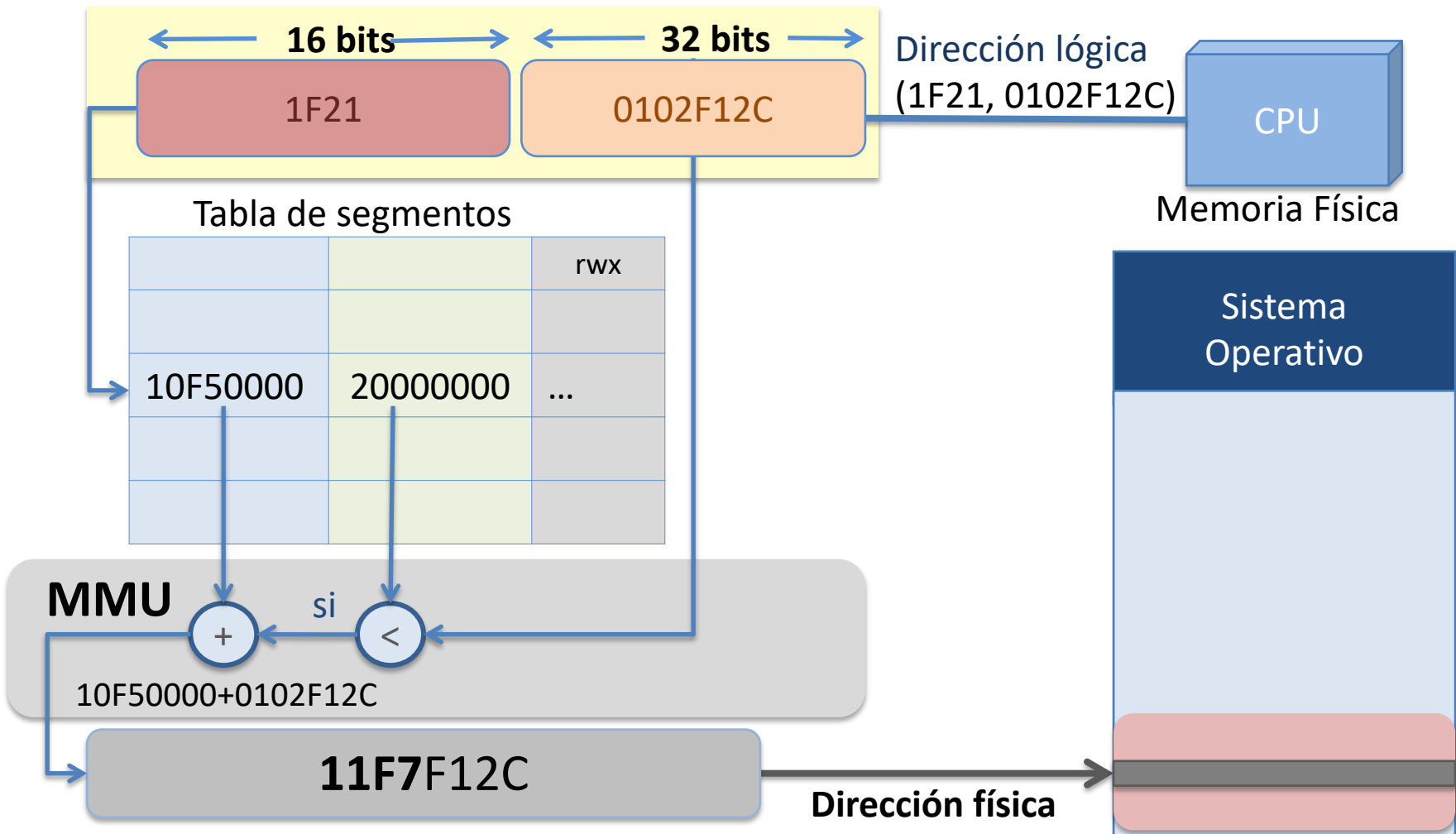
Los segmentos se identifican por un número y su tamaño

- Segmentación: Traducción de direcciones



- **Ejemplo segmentación micro 32 bits (Intel x86)**

- Desplazamiento $k = 32$ bits
- Segmentación $s = 16$ bits
- Dirección lógica (seg, desp) : (1F21, 0102F12C)



- **Segmentación: Análisis de eficiencia**

- **Ventajas**

- No aparece **fragmentación interna**
 - Facilita la reubicación
 - Proporciona protección

- **Inconvenientes**

- Fragmentación externa

- **Tamaños de segmento**

- Muy Grandes → aproximación a particiones variables
 - Muy Pequeños → eliminaría la fragmentación externa, pero aumentaría el tamaño destinado a registros
 - Tamaño Fijo → Paginación

- Concepto de Asignación Dispersa
- Paginación
- Segmentación
- **Paginación Multinivel**

- Concepto de Asignación Dispersa
- Paginación
- Segmentación
- **Paginación Multinivel**

- **Motivación**

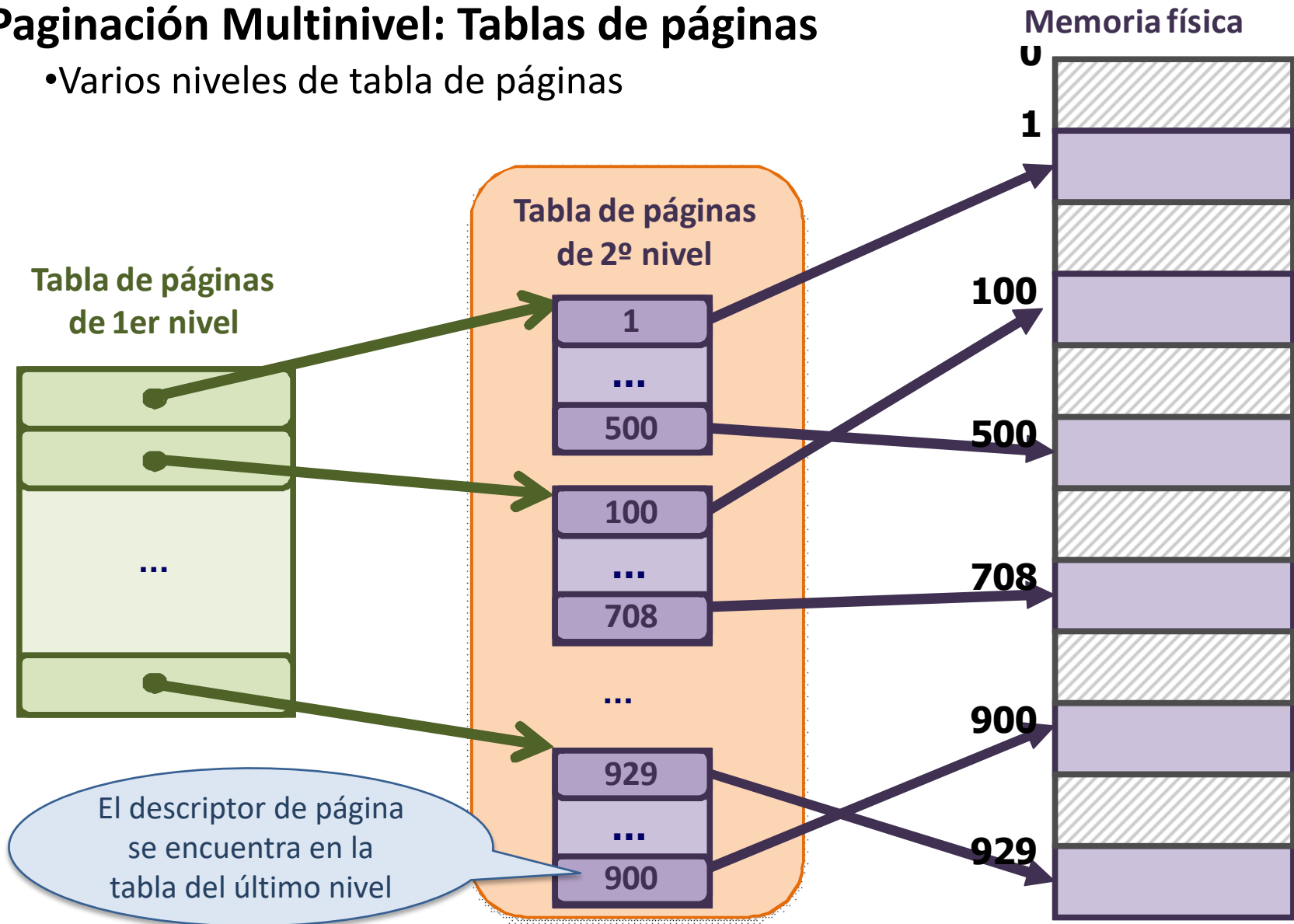
- Procesos con espacios lógicos muy grandes
 - Las tablas de páginas puede ser excesivamente grande
 - El hecho de tener que ubicar la tabla de páginas de forma contigua resulta ineficiente
- Solución: Pagar la propia tabla de páginas
 - Se fragmenta o divide en trozos la tabla de páginas de forma que cada fragmento ocupe un marco en memoria física.



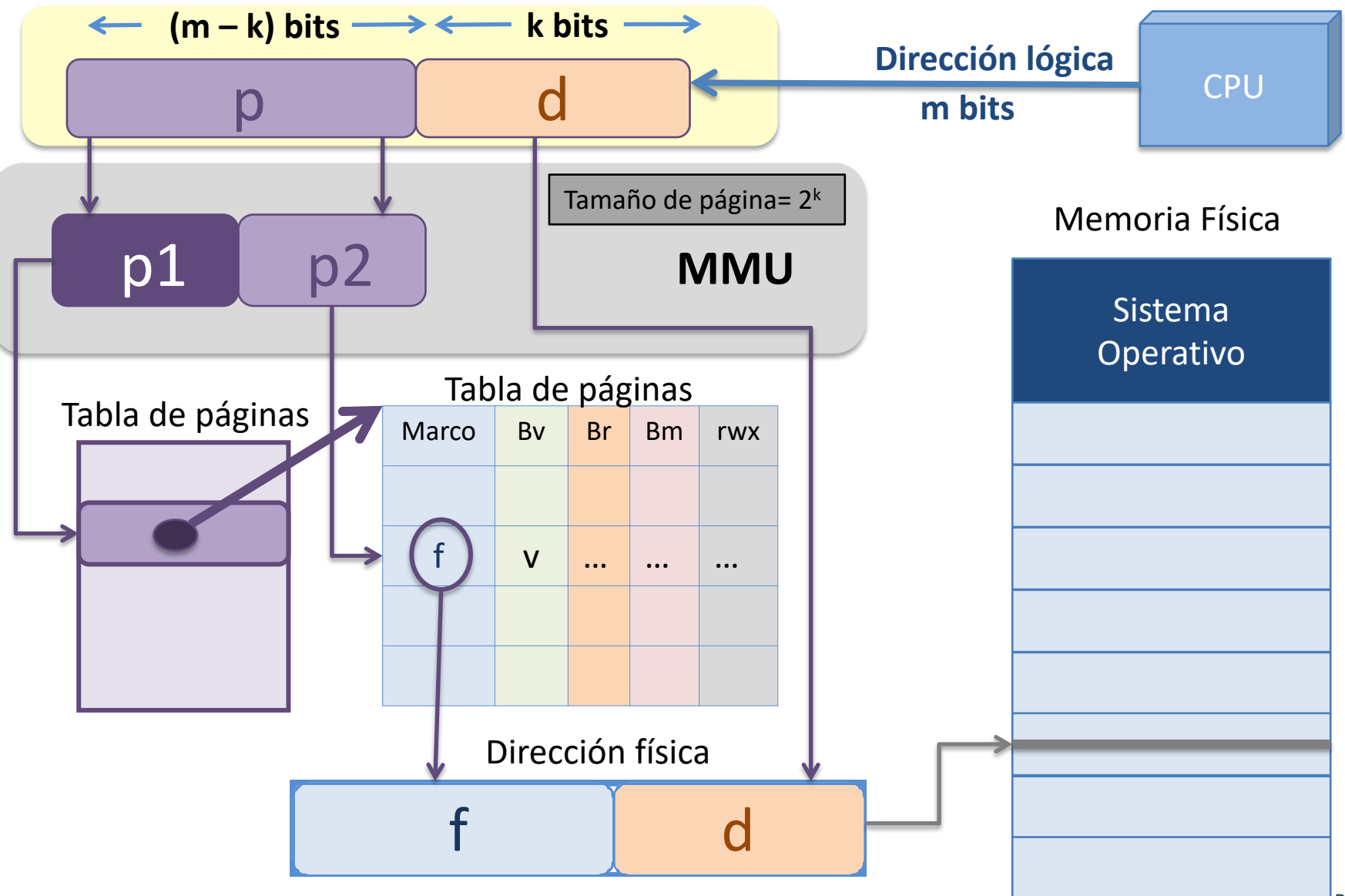
Paginación multinivel

- **Paginación Multinivel: Tablas de páginas**

- Varios niveles de tabla de páginas



- Paginación multinivel: Traducción de direcciones



- Ejemplo paginación multinivel micro 32 bits

- Tamaño página 4K: $k=12$ bits ($2^{12}=4096$)
- Páginas (primer nivel 8 bits/segundo nivel 12 bits) $8+12=20=m-k$
- Dirección lógica = **FF1234B1**

