

# Fonaments dels Sistemes Operatius (FSO)

Departament d'Informàtica de Sistemes i Computadores (DISCA)  
*Universitat Politècnica de València*

Bloc Temàtic 4: Gestió de Memòria  
Unitat Temàtica 10

## Assignació Dispersa de Memòria

f SO

DISCA



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

- **Objectius**

- Descriure l'**estratègia d'assignació dispersa** de memòria
- Estudiar de les **tècniques** bàsiques d'assignació dispersa:
  - Paganació
  - Segmentació
- Comprendre la paginació multinivell i la seu aplicabilitat

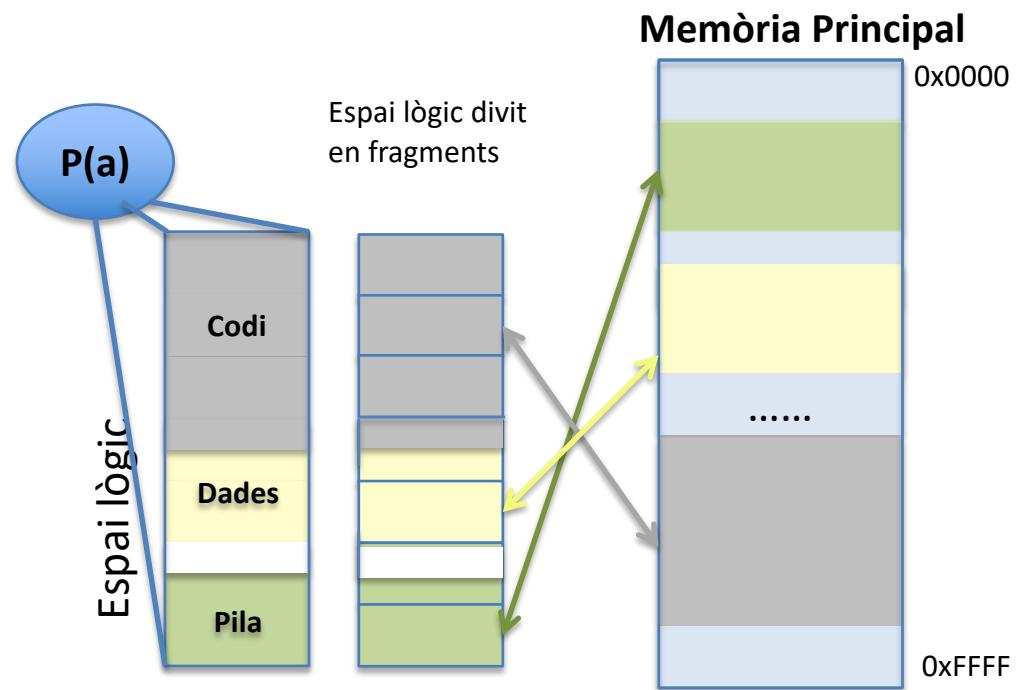
- **Bibliografia**

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

- **Concepte d'Assignació Dispersa**
- Paginació
- Segmentació
- Paginació Multinivell

## • Assignació dispersa

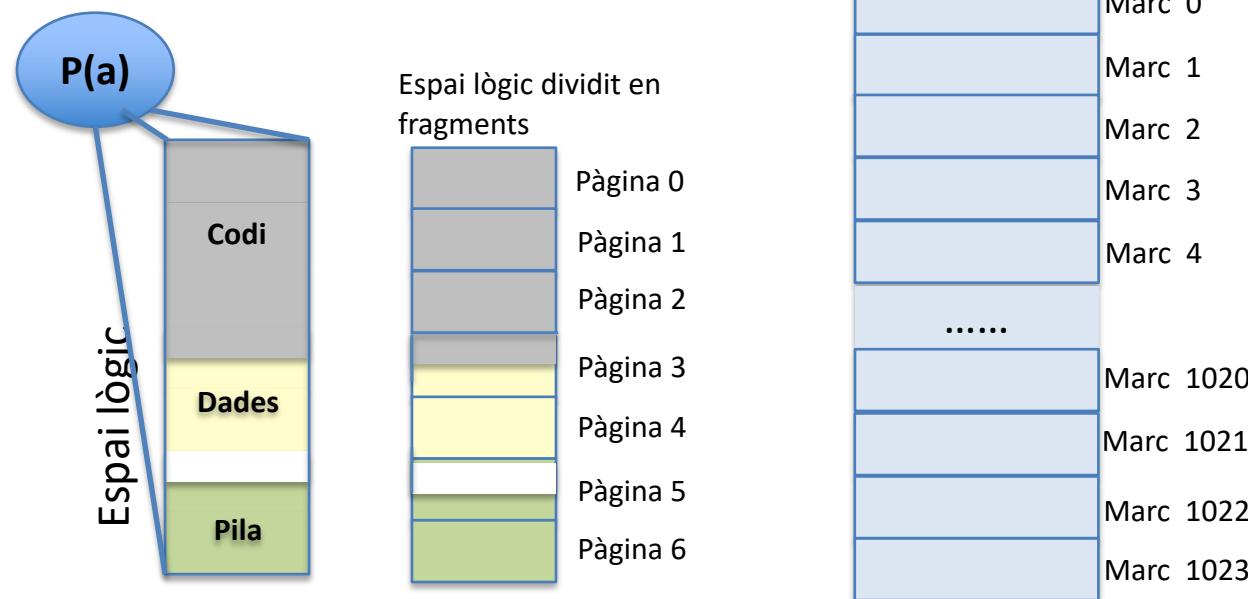
- Considera l'espai d'adreses lògiques d'un procés dividit en **fragments**
- Cada fragment s'ubica en memòria física de forma independent



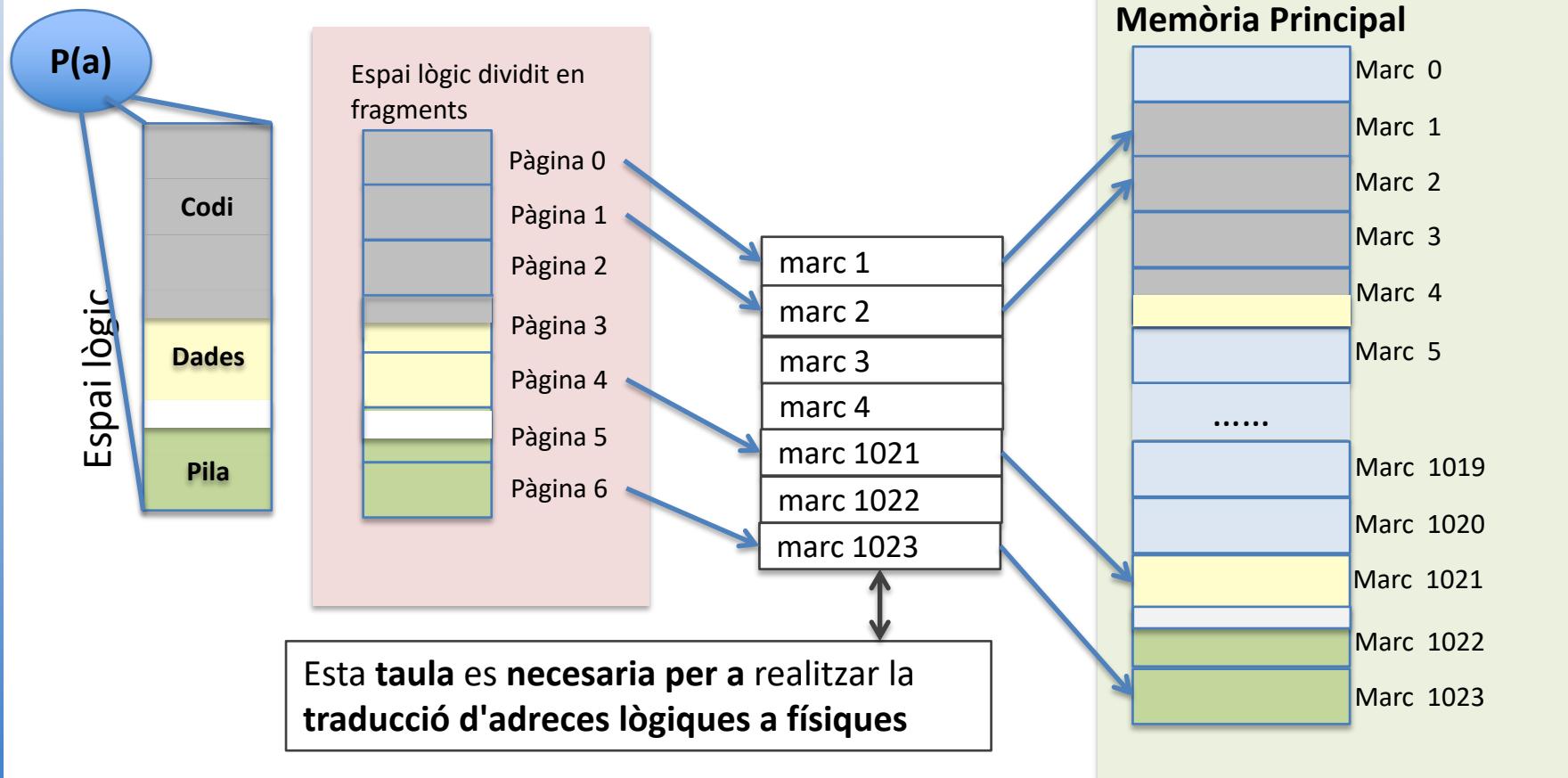
- L'espai d'adreces físiques d'un procés no ha de ser contigu
  - **Paginació**: els fragments són de grandària fixa
  - **Segmentació**: els fragments són de grandària variable
- Per a traduir adreces lògiques a adreces físiques, la MMU necessita conèixer la ubicació de cada fragment i la seua grandària
  - **Taula de pàgines**
  - **Taula de segments**

- Concepte d'Assignació Dispersa
- **Paginació**
- Segmentació
- Convinació de tècniques
  - Segmentació Paginada
  - Paginació Multinivell

- **Esquema de gestió de memòria amb paginació permet:**
  - L'espai d'adreces de memòria física d'un procés siga no contigu
  - Els espais d'adreces d'un procés els considera dividits en fragments de grandària fixa
    - L'espai d'adreces lògiques → dividit en pàgines
    - L'espai d'adreces físiques → dividit en marcs o frames



- Quan es va a executar un procés el sistema **carrega** totes les **seues pàgines en marcs** de memòria principal
  - Cada pàgina ocupa un marc
  - Es construeix **una taula** per a emmagatzemar el **nombre de marc** on s'ha ubicat cada pàgina del procés
  - Cada procés té associada una taula de pàgines



- **Taula de Pàgines**

- Cada entrada de la taula es denomina **descriptor de pàgina** i conté
  - El **nombre de marc** on se ha ubicado la pàgina
  - Un conjunt de bits: **bit de validesa**, **bits de protecció**, **bit de modificat**, ....

Descriptor Pàgina 0	Nombre de marc	Bit validesa	Bit modificat	Bits Protecció rwx
Descriptor Pàgina 1	Nombre de marc	Bit validesa	Bit modificat	Bits Protecció rwx
	.....	.....	.....	.....
Descriptor Pàgina n-1	Nombre de marc	Bit validesa	Bit modificat	Bits Protecció rwx

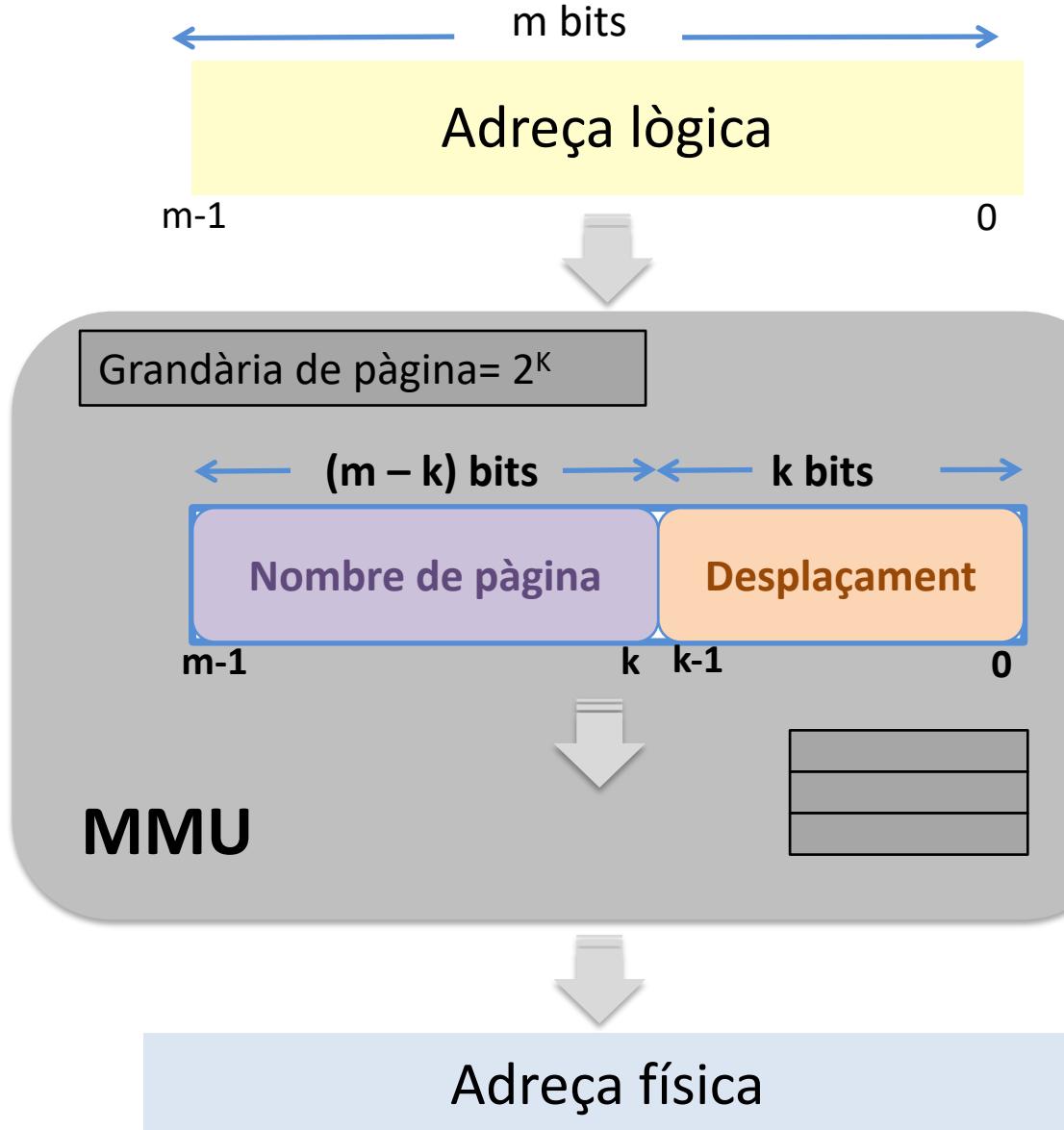
**Nombre de marc de Memòria Principal on esta cargada la pàgina**

Indica si la pàgina es **vàlida** per al procés

Indica si la pàgina ha sigut **modificada**

Indica el tipus d'accés **permés** a la pàgina

- Estructura d'una adreça lògica**

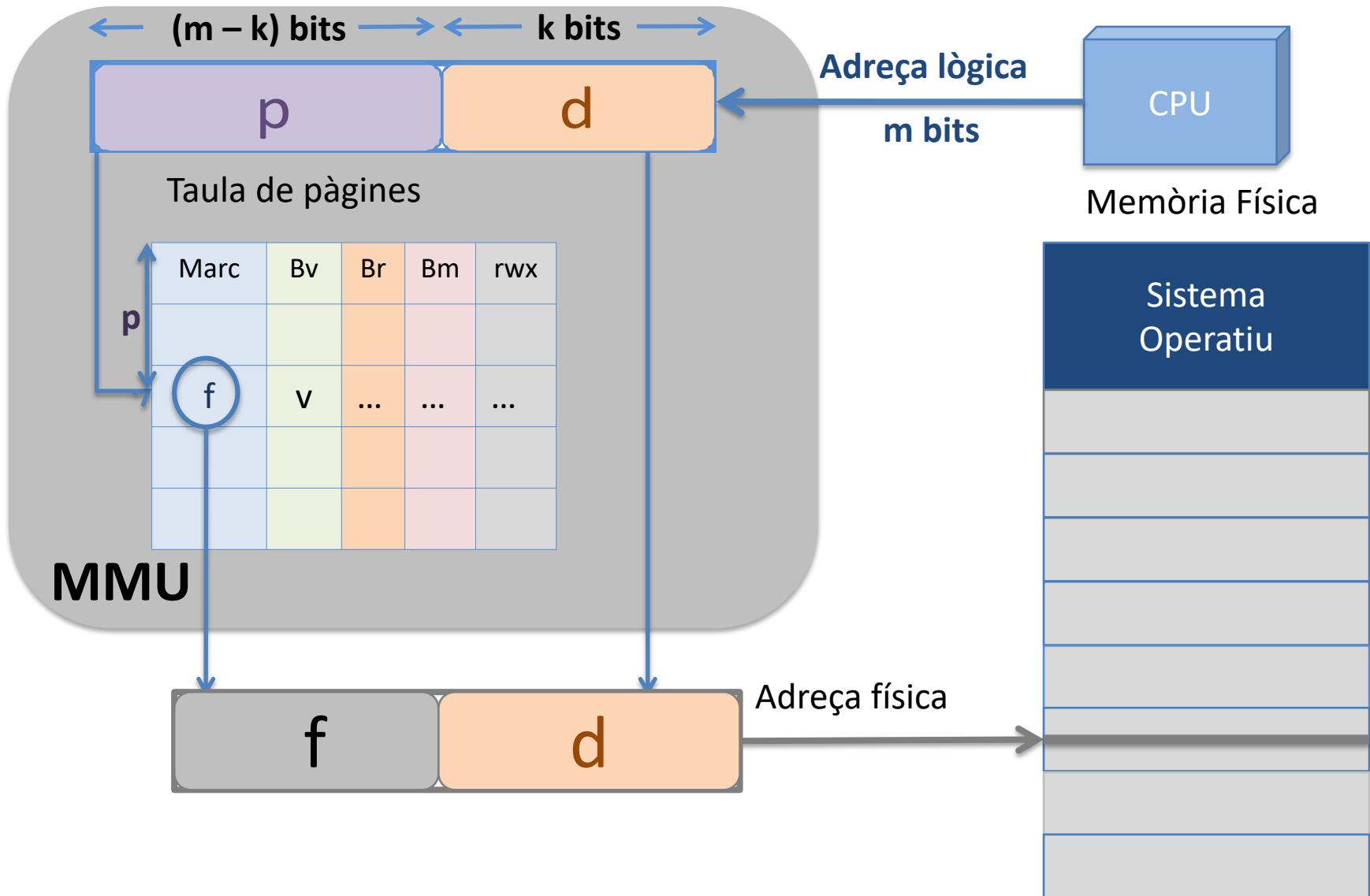


Espai lògic de  $2^m$  paraules

Sistema amb **pàgines de  $2^k$  paraules**, espai lògic de  $2^m$  paraules

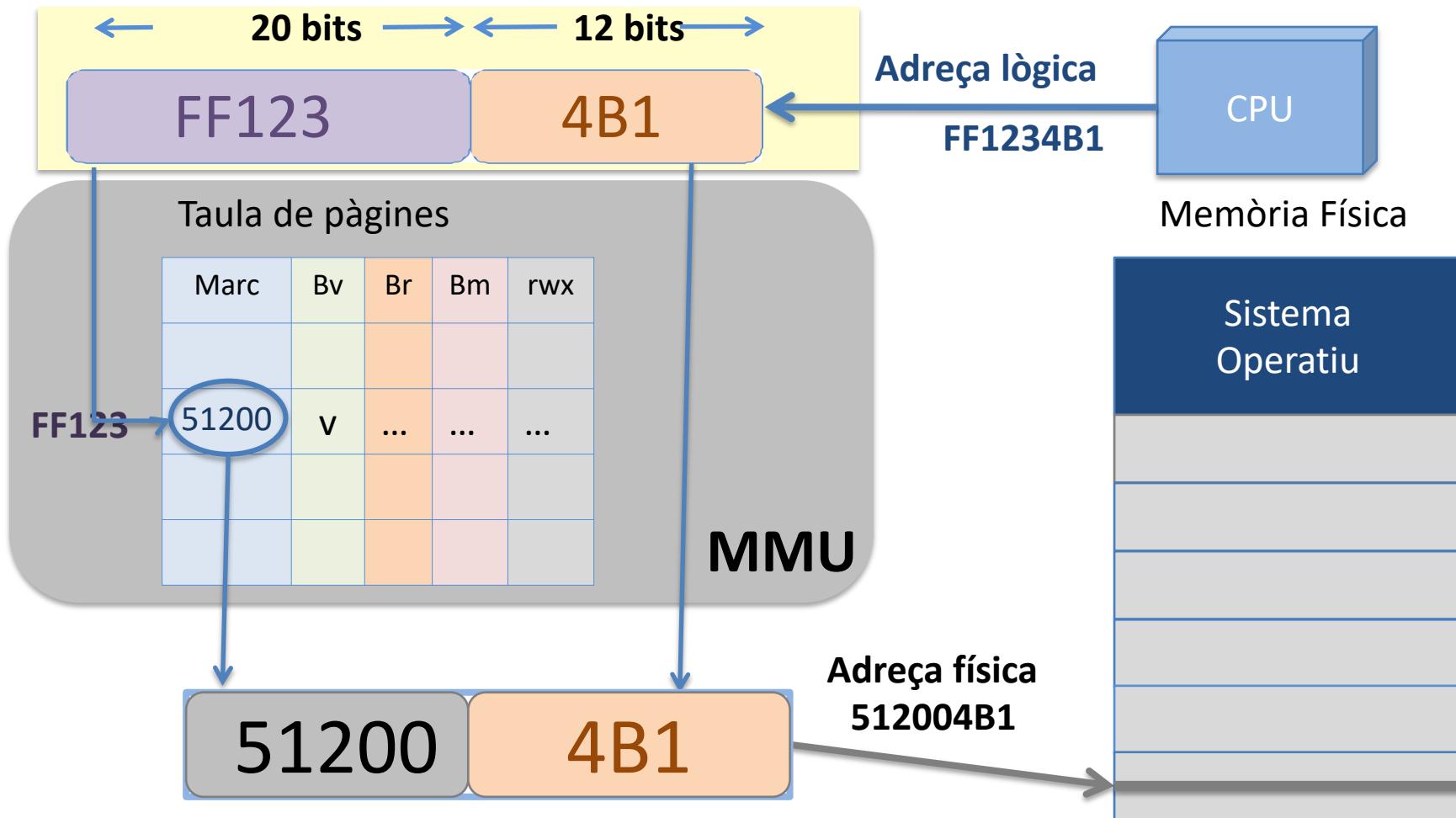
En la **MMU** ve l'espai lògic format per  $2^{(m-k)}$  pàgines

- Traducció d'adreces



- **Exemple paginació micro 32 bits.**

- Grandària pàgina 4K:  $k=12$  bits ( $2^{12}=4096$ )
- Pàgines =  $2^{20} = 1M = 1048576$  ( $m-k = 32-12=20$  bits)
- Adreça lògica = **FF1234B1**



- **Paginació: Anàlisi de eficiència**

- Avantatges

- No apareix **fragmentació externa**
    - Facilita la reubicació
    - Proporciona protecció
    - Compartició de pàgines de codi entre processos

- Inconvenients

- **Fragmentació interna**

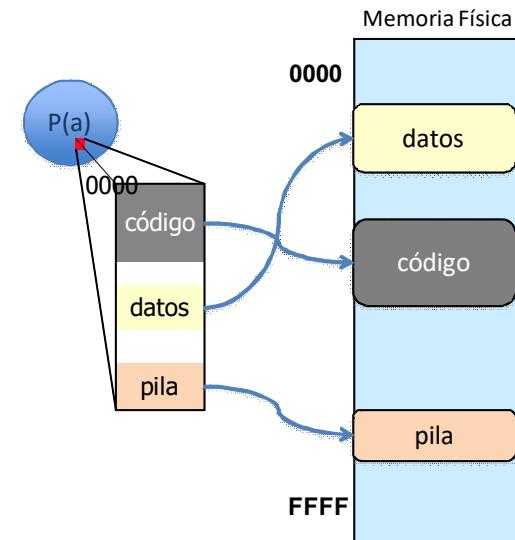
- Grandàries de pàgina

- El **grandària de pàgina ha de ser potència de 2** per a facilitar l'obtenció del nombre de pàgina i desplaçament
      - Grandàries actuals de pàgines 4K,8K
    - Pàgines de grandària gran: molta fragmentació interna
    - Pàgines de grandària xicoteta: taules de pàgines molt grans

- **Implementació de les taules de pàgines**
  - **Registres de la MMU**
    - Només per a espais lògics molt xicotets (poques pàgines)
  - **Memòria**
    - Es manté un registre (PTBR=registre base taula de pàgines) amb l'adreça física base en la que comença la taula en memòria
    - Es necessita un accés extra a memòria principal (accés a la taula de pàgines) per a poder traduir una adreça lògica
  - **TLB (*translation look-aside buffer*)**
    - La TLB conté només unas poques entrades de la taula de pàgines (les d'ús recent) de la taula de pàgines
      - Accés mes ràpid que una implantació en memòria
      - Taxa d'éxito alta amb poques entrades

- Concepte d'Assignació Dispersa
- Paginació
- **Segmentació**
- Paginació Multinivell

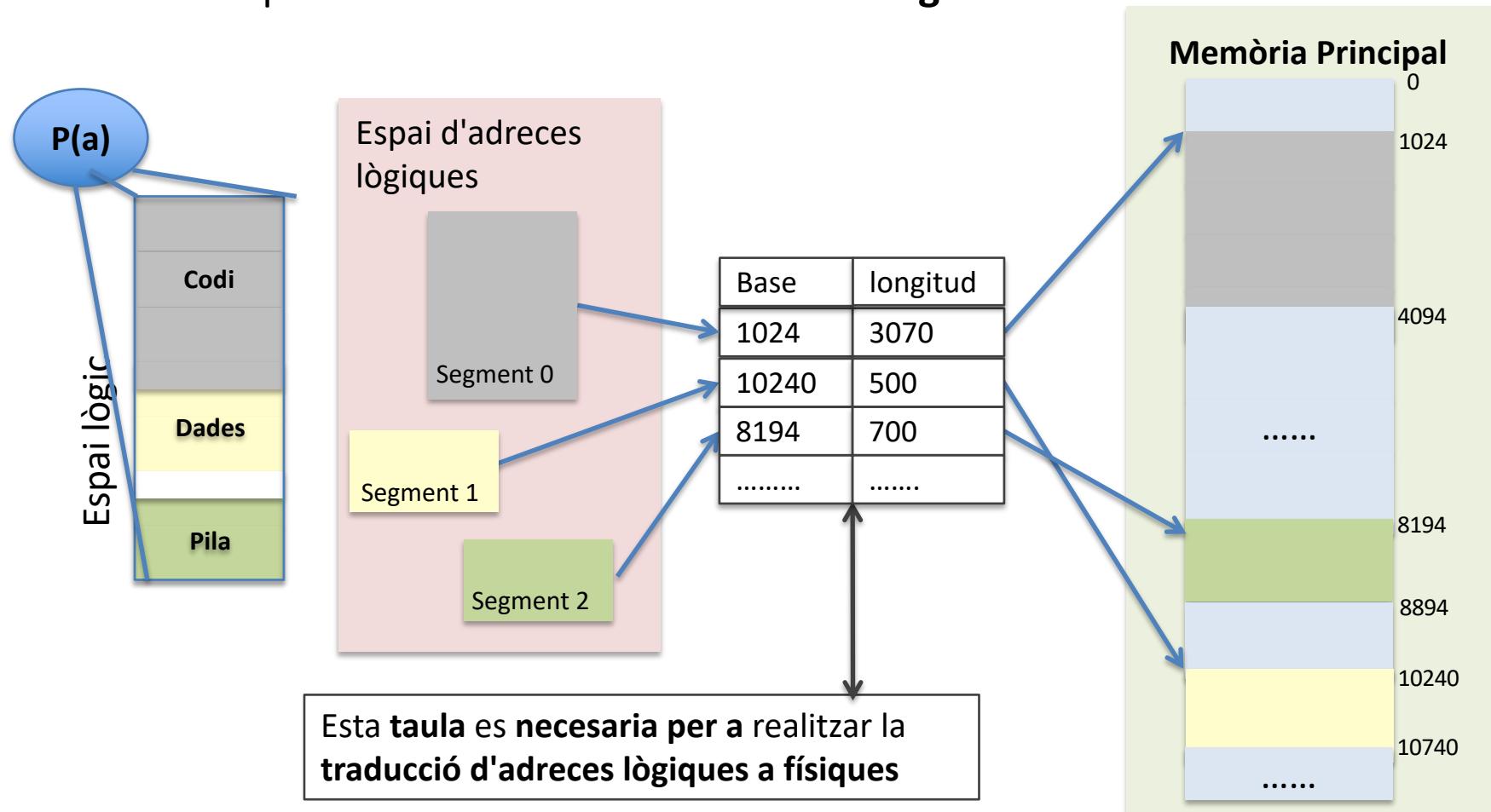
- Esquema de gestió de memòria amb Segmentació
  - L'espai d'adreces físiques en memoria princiàr d'un procés pot ser **no contigu**
  - L'espai d'adreces físiqueses troba dividit en **fragments de longitud variable → Segment**
  - L'espai lògic es un conjunt de **segments**
    - Cada segment te **un nombre i una longitud**
  - El programa es compilat i automàticament el compilador construeix un o diversos segments reflexant el contingut del programa font
    - Segment de Codi**
    - Segment de Dades**
    - Segment de Pila**



Un segment sempre s'ubica de forma contigua en memòria principal

# Segmentació

- Quan es va a executar un procés el sistema **carrega** tots els **seus segments en memòria principal**
  - Es construeix **una taula** per a emmagatzemar l'**adreça física base de cada segment i la seu longitud**
  - Cada procés té associada **una taula de segments**



- **Taula de Segments**

- Cada entrada de la taula es denomina **descriptor de segments** i conté
  - La adreça física base on s'ha ubicat el segment
  - La grandària del segment
  - Un conjunt de bits: **bits de protecció, bit de validesa**

Descriptor Segment 0	Base del segment	Límit del segment	Bits Protecció rwx	Bit validesa
Descriptor Segment 1	Base del segment	Límit del segment	Bits Protecció rwx	Bit validesa
	.....	.....	.....	.....
Descriptor Segment n-1	Base del segment	Límit del segment	Bits Protecció rwx	Bit validesa

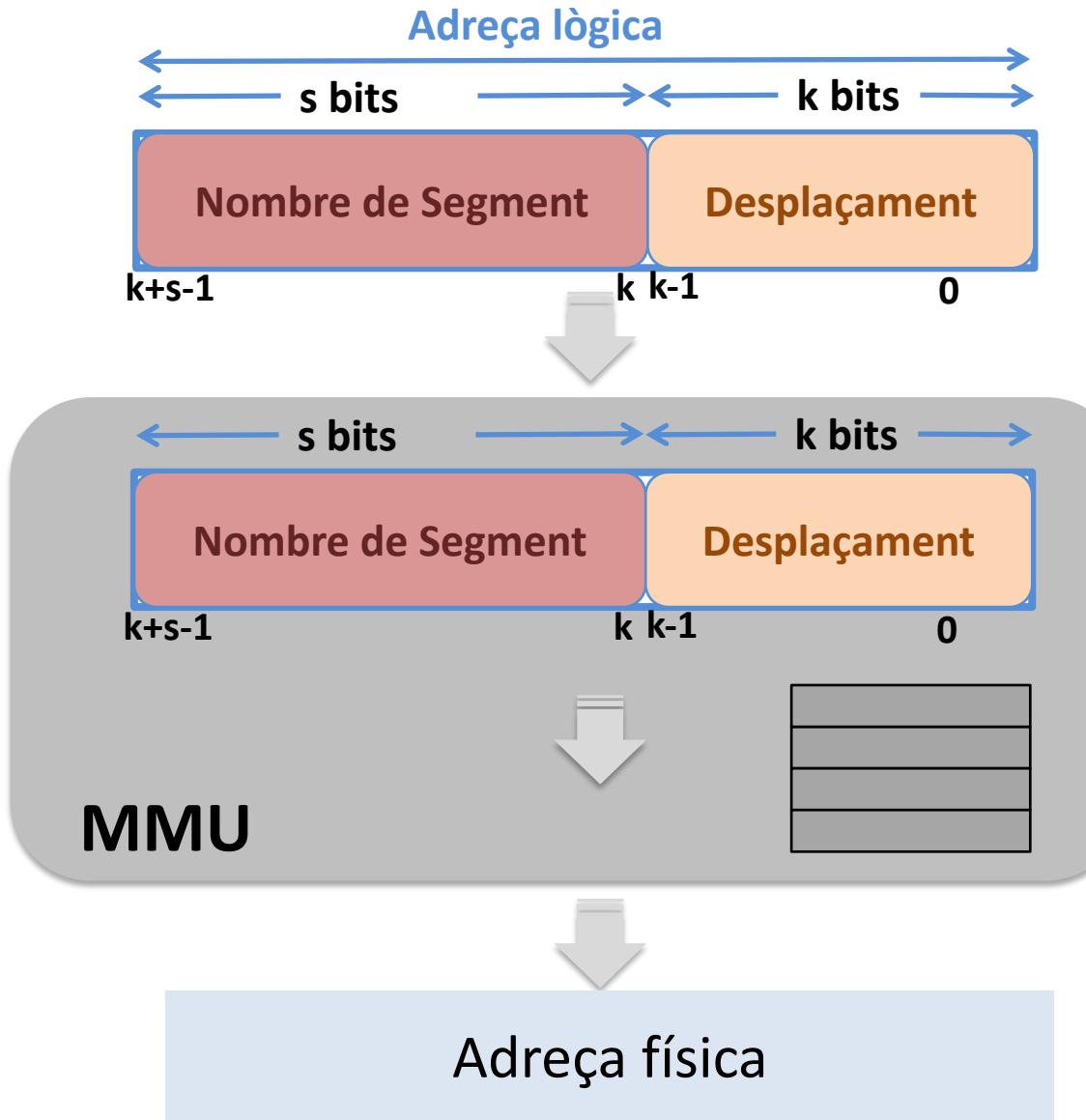
Conté la **primera adreça física** a partir de la qual s'ha ubicat el **segment en memòria**

Especifica la **longitud del segment**

Indica si el **segment** és **vàlid** per al procés

Indica el **tipus d'accés permés** per al segment

- Estructura d'una adreça lògica

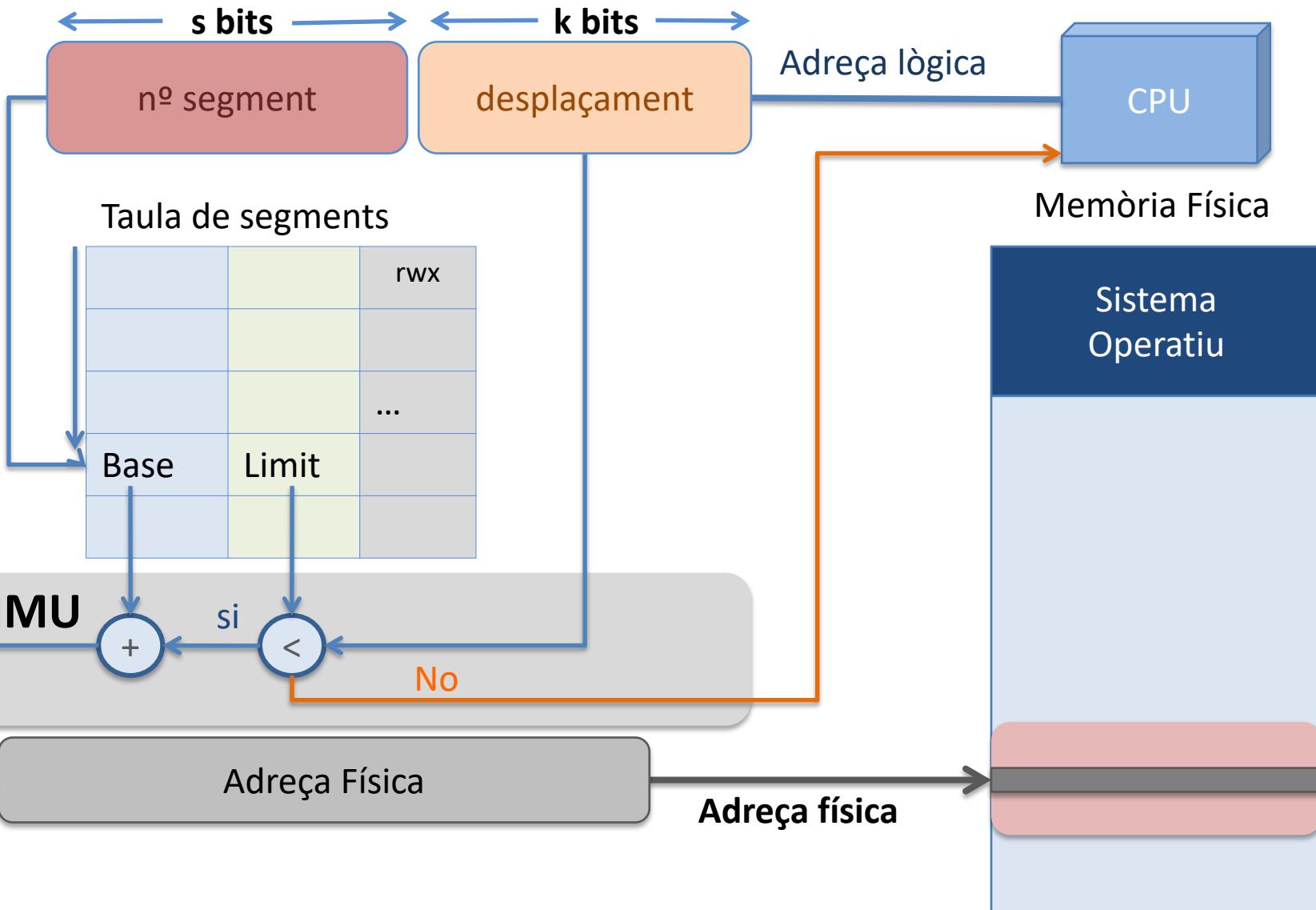


**Espai lògic** està format per  **$2^s$  segments**. Cada segment té  $2^k$  paraules

La **MMU** ve l'**espai lògic** format per  **$2^s$  segments** de  $2^k$  **paraules**  
Los segments s'identifiquen per un nombre i la seu grandària

# Segmentació

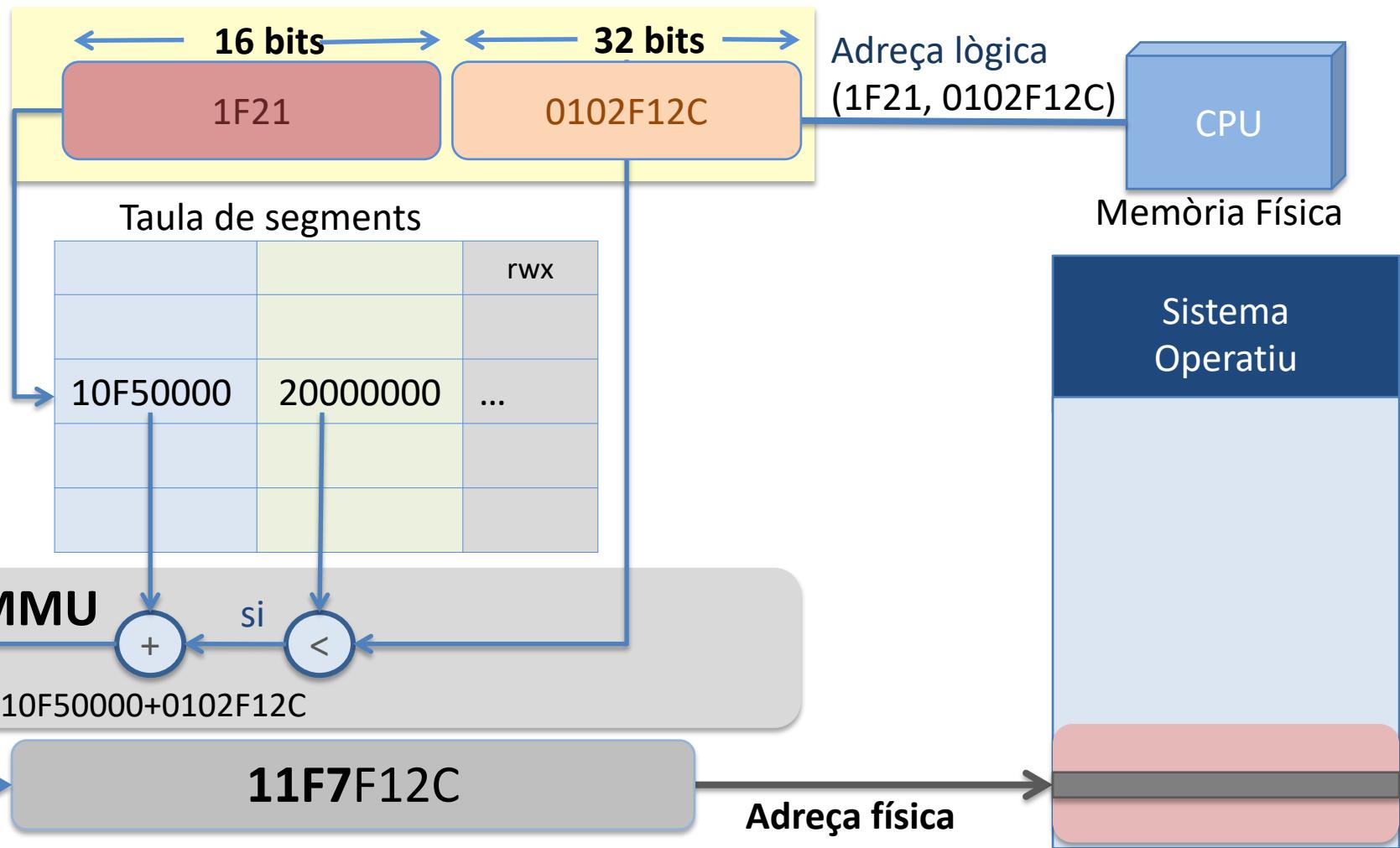
- Traducció d'adreces



# Segmentació

- Exemple segmentació micro 32 bits (Intel x86)

- Desplaçament k = 32 bits
- Segmentació s = 16 bits
- Adreça lògica (seg, desp) : (1F21, 0102F12C)



- **Segmentació: Anàlisi d'eficiència**

- Avantatges

- No apareix **fragmentació interna**
    - Facilita la reubicació
    - Proporciona protecció

- Inconvenients

- Fragmentació externa

- Grandàries de segment

- Molt Grans → aproximació a particions variables
    - Molt xicotets → eliminaria la fragmentació externa, però augmentaria la grandària destinat a registres
    - Grandària Fixa → Paganació

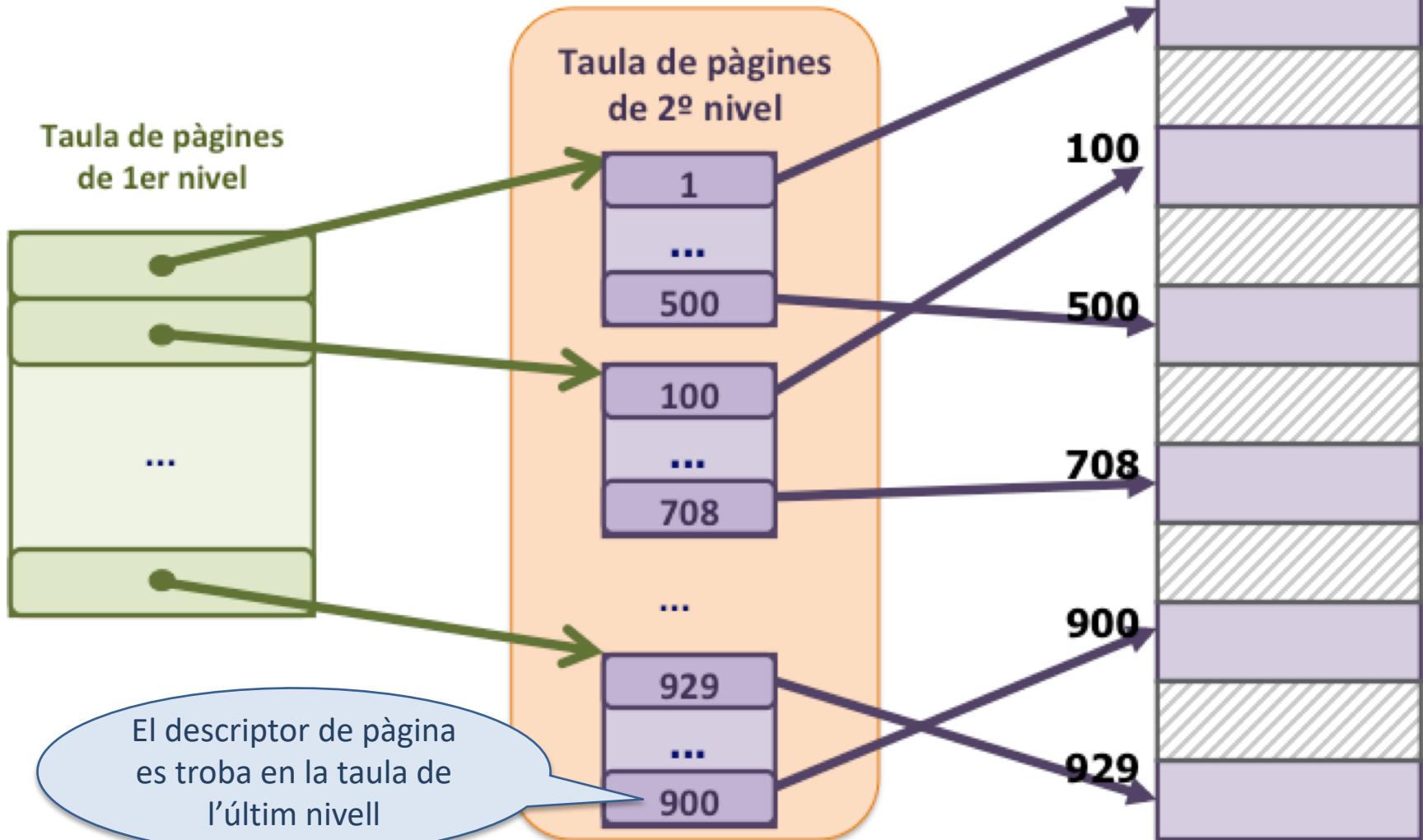
- Concepte d'Assignació Dispersa
- Paginació
- Segmentació
- **Paginació Multinivell**

- Motivació
  - Per a espais d'adreces molt grans, la taula de pàgines pot ser excessivament gran, i el fet de tindre que ubicar-la de forma continua resulta ineficient
- Solució:
  - Paginar la pròpia taula de pàgines, es a dir, partir-la de forma que cada fragment ocupe un marc en memòria física.



Paginació multinivell

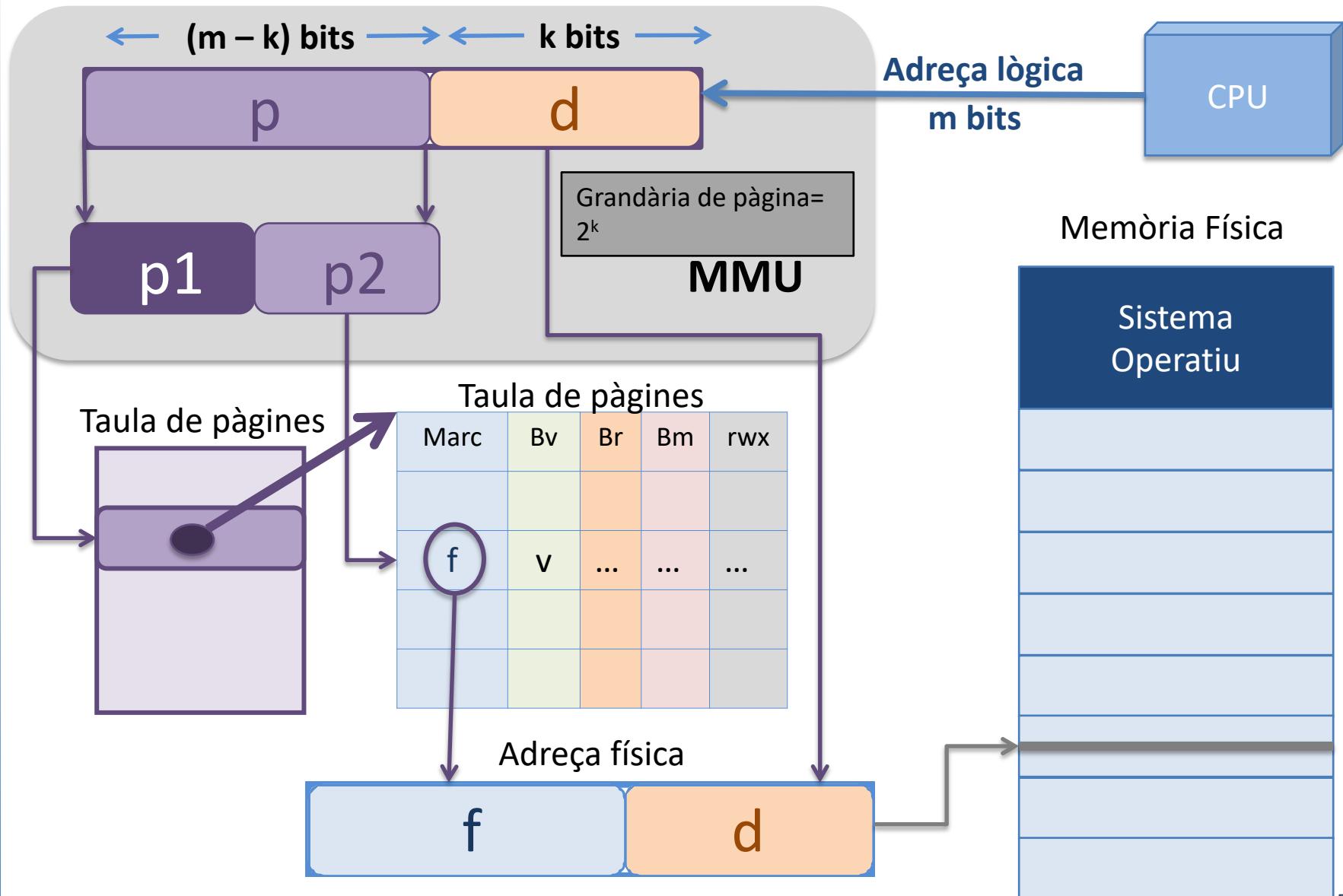
## Taules de pàgines



# Paginació Multinivell

fSO

- Traducció d'adreces



# Paginació Multinivell

fSO

- Exemple paginació multinivell micro 32 bits

- Grandària pàgina 4K:  $k=12$  bits ( $2^{12}=4096$ )
- Pàgines (primer nivell 8 bits/segon nivell 12 bits)  $8+12=20=m-k$
- Adreça lògica = **FF1234B1**

