

# SISTEMAS OPERATIVOS

## UNIDAD 6B

### ADMINISTRACION DE MEMORIA

**Los espectadores no encuentran aquello que desean,  
sino que desean aquello que encuentran  
(San Agustin)**



## MEMORIA VIRTUAL

- ❖ Todas las referencias a la memoria se traducirán dinámicamente a direcciones físicas durante la ejecución:
  - Un proceso puede cargarse y descargarse de la memoria principal de tal forma que ocupe regiones diferentes.
- ❖ Un proceso puede dividirse en varias partes y no es necesario que estas partes se encuentren contiguas en la memoria principal durante la ejecución:
  - No será necesario que todas las páginas o todos los segmentos de un proceso estén en la memoria durante la ejecución.



## MEMORIA VIRTUAL: EJECUCION DE UN PROGRAMA

- ❖ El sistema operativo comienza trayendo sólo unos pocos fragmentos del programa.
- ❖ El conjunto residente es la parte de un proceso que está realmente en la memoria principal.
- ❖ Si el procesador encuentra una dirección lógica que no está en la memoria principal, genera una interrupción que indica un fallo de acceso a la memoria.
- ❖ El sistema operativo pone al proceso interrumpido en estado Bloqueado.



## MEMORIA VIRTUAL: EJECUCION DE UN PROGRAMA

- ❖ El sistema operativo necesita traer a la memoria principal el fragmento del proceso que contiene la dirección lógica que provocó el fallo de acceso:
  - El sistema operativo emite una solicitud de Lectura de E/S al disco.
  - El sistema operativo puede expedir otro proceso para que se ejecute mientras realiza la operación de E/S.
  - Una vez que el fragmento deseado se ha traído a la memoria principal y se ha emitido la interrupción de E/S, se devuelve el control al sistema operativo, que coloca el proceso afectado en el estado de Listo.



## MEMORIA VIRTUAL: DIVISION DE UN PROCESO

- ❖ Se pueden mantener más procesos en la memoria principal:
  - Se cargan sólo algunos fragmentos de un proceso particular.
  - Con tantos procesos en la memoria principal es muy probable que uno de los procesos esté en estado Listo en un instante determinado.
- ❖ Es posible que un proceso sea más grande que toda la memoria principal.



## MEMORIA VIRTUAL: TIPOS

- ❖ Memoria real:
  - Memoria principal.
- ❖ Memoria virtual:
  - Memoria situada en el disco.
  - Permite una multiprogramación muy efectiva y releva al usuario de las rígidas e innecesarias restricciones de la memoria principal.



## MEMORIA VIRTUAL: TRASHING

- ❖ El sistema operativo expulsa un fragmento de un proceso justo antes de ser usado.
- ❖ El procesador consume más tiempo intercambiando fragmentos que ejecutando instrucciones de usuario.

## CRITERIO DE LOCALIDAD

- ❖ Las referencias a los datos y al programa dentro de un proceso tienden a agruparse.
- ❖ Sería posible hacer predicciones inteligentes sobre qué fragmentos de un proceso se necesitarán en un futuro cercano.



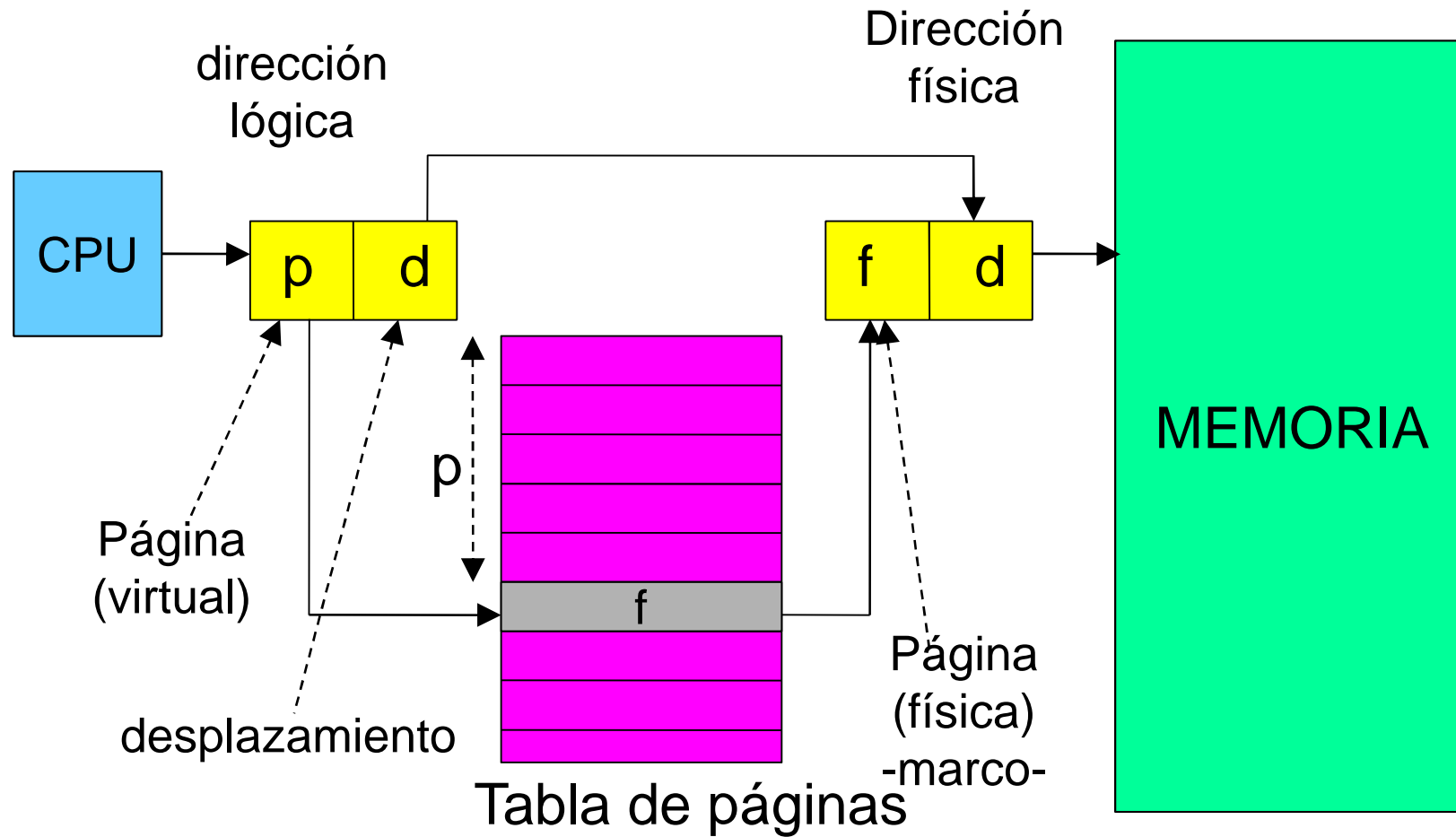
## PAGINACION – CONCEPTOS GENERALES -

- ❖ La memoria física se divide en bloques de tamaño fijo que llamamos marcos. La memoria virtual se divide en bloques del mismo tamaño llamados páginas.
- ❖ El sistema operativo mantiene una tabla de páginas para cada proceso:
  1. Muestra la posición del marco de cada página del proceso.
  2. La dirección de la memoria consta de un número de página y de un desplazamiento dentro de la página.
- ❖ Al ejecutar un proceso se cargan sus páginas en los marcos disponibles. La vinculación de direcciones requiere soporte por hardware (Manejador de Memoria)





PAGINACION – CONCEPTOS GENERALES -



## PAGINACION – CONCEPTOS GENERALES -

- ❖ - La paginación remedia la fragmentación externa, pero no la fragmentación interna.
- ❖ - Un intento de acceso a una página virtual que no esté asociada a un marco produce un señalamiento al SO (trap), llamado fallo de página.

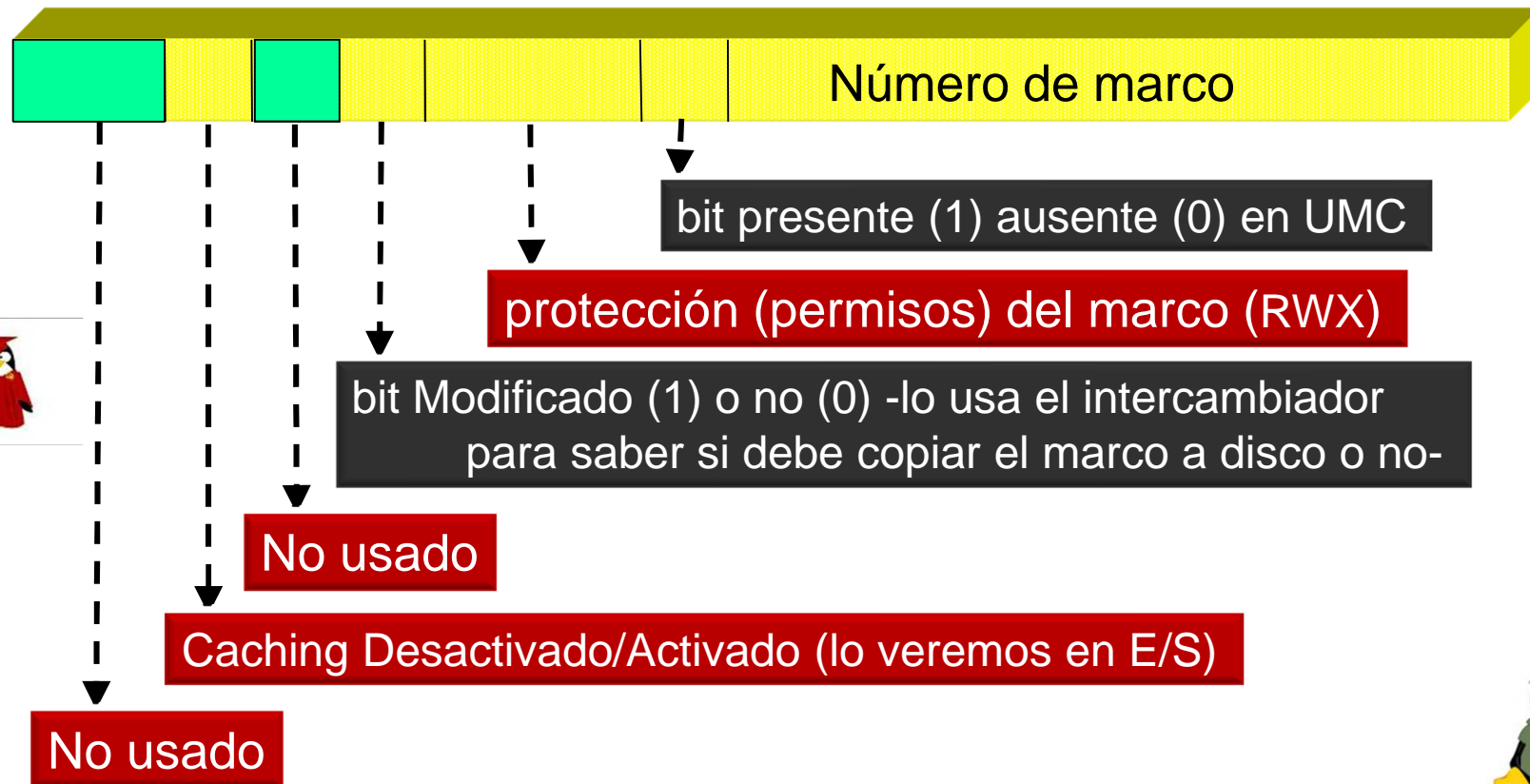
Como respuesta al fallo de página, el SO:

1. selecciona una página poco usada del proceso
  2. Intercambia la página a disco
  3. Asigna el marco de la página liberada a la página virtual que se intenta acceder
- Esto supone una forma de reasignación dinámica por bloques de las direcciones de memoria del proceso

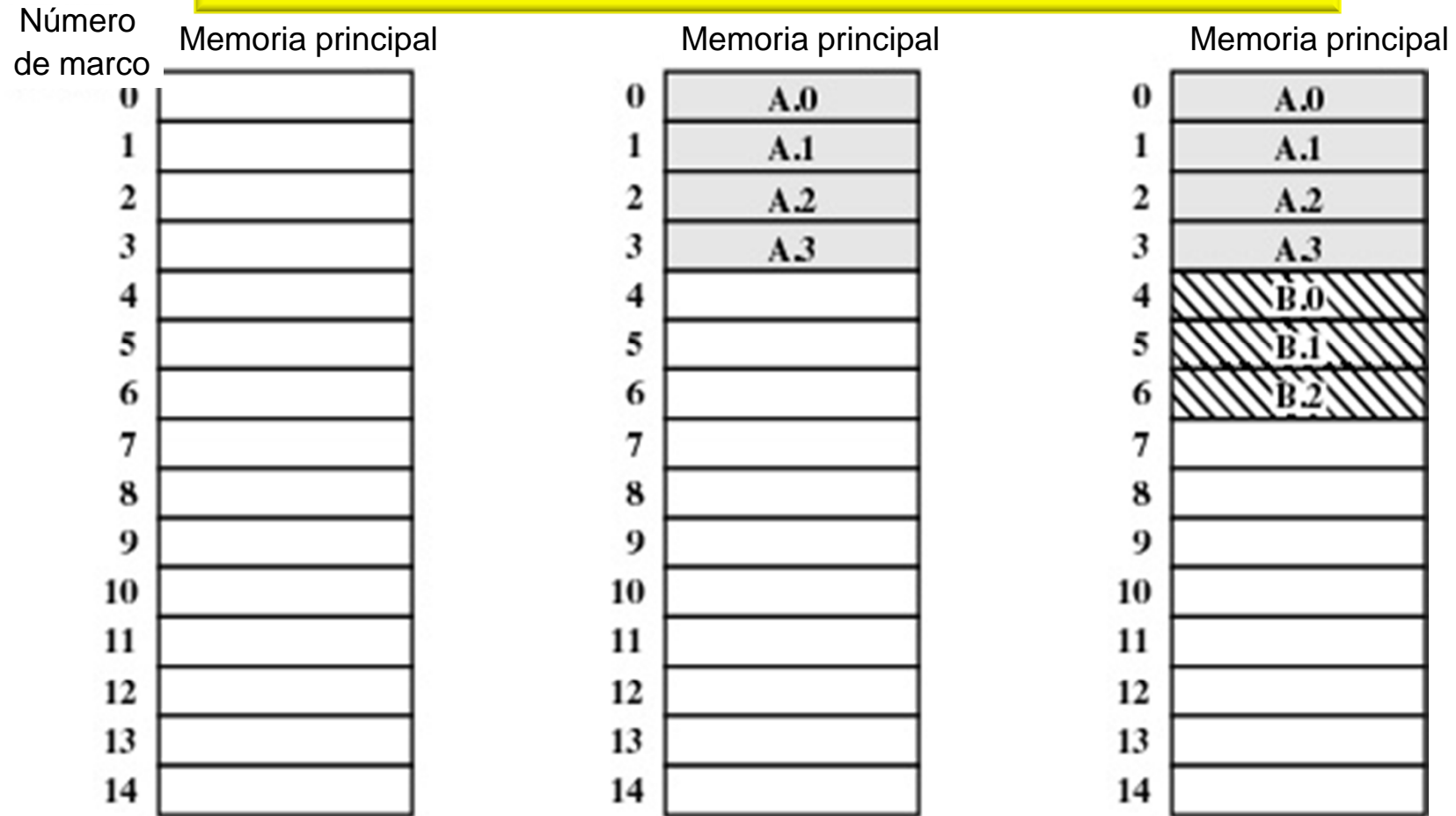


## PAGINACION – TMP-

Son tablas que contienen (para cada proceso) el número de marco que corresponde a cada página virtual del proceso



**PAGINACION – ASIGNACION EN UMC-**



(a) Quince marcos libres

(b) Carga del proceso A

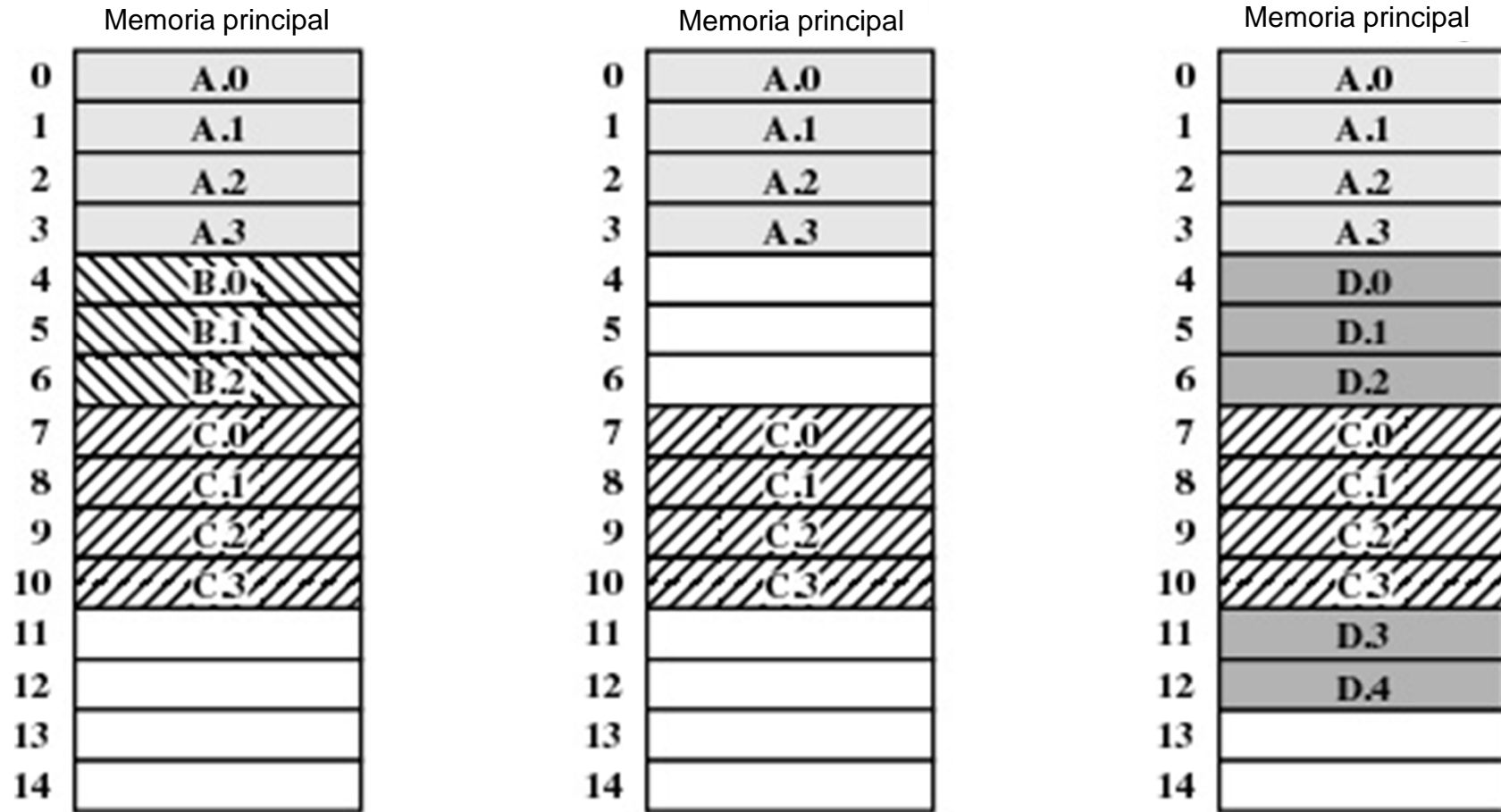
(c) Carga del proceso B

Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4

Figura 7.9. Asignación de páginas de procesos a marcos libres.



PAGINACION – ASIGNACION EN UMC-



(d) Carga del proceso C

(e) Descarga del proceso B

(f) Carga del proceso D

Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4



Figura 7.9. Asignación de páginas de procesos a marcos libres.

PAGINACION – ASIGNACION EN UMC-

0	0
1	1
2	2
3	3

Tabla de páginas del proceso A

0	—
1	—
2	—

Tabla de páginas del proceso B

0	7
1	8
2	9
3	10

Tabla de páginas del proceso C

0	4
1	5
2	6
3	11
4	12

Tabla de páginas del proceso D

13
14

Lista de marcos libres

Figura 7.10. Estructuras de datos para el ejemplo de la Figura 7.9 en el instante de tiempo (f).



Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4



**PAGINACION – ASIGNACION EN UMC- CONCLUSIONES**

1. Cuanto menor sea el tamaño de página, menor será la cantidad de fragmentación interna.
2. Cuanto menor sea la página, mayor será el número de páginas que se necesitan por proceso.
3. Un número mayor de páginas por proceso significa que las tablas de páginas serán mayores.
4. Esto puede significar que una gran parte de las tablas de páginas de los procesos activos deben estar en la memoria virtual.
5. 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.



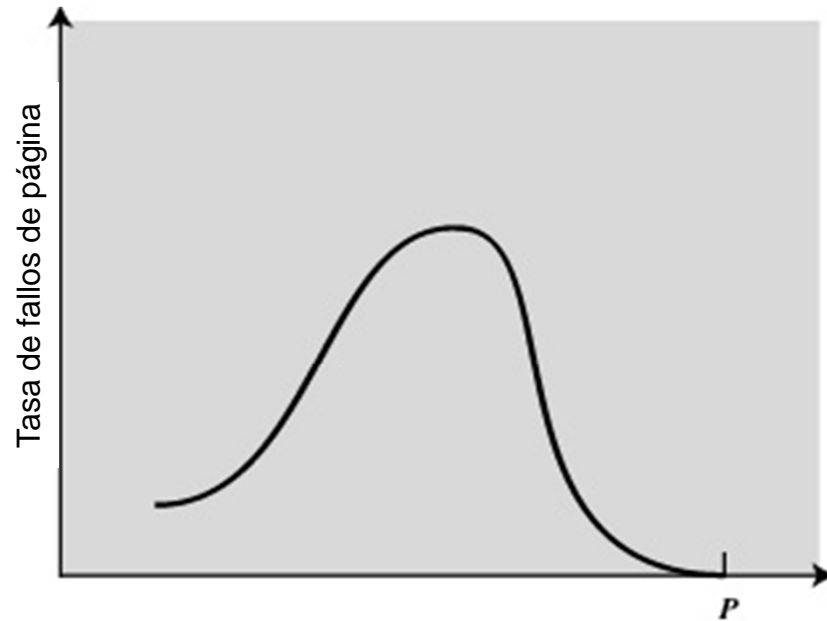
**PAGINACION – ASIGNACION EN UMC- CONCLUSIONES**

6. Si el tamaño de página es muy pequeño, estarán disponibles en la memoria principal un gran número de páginas para cada proceso.
7. Después de un tiempo, todas las páginas de la memoria tendrá parte de las referencias más recientes del proceso. La tasa de fallos de página será menor.
8. Cuando se incrementa el tamaño de la página, cada página individual contendrán posiciones cada vez más distantes de cualquier referencia reciente. La tasa de fallos será mayor.

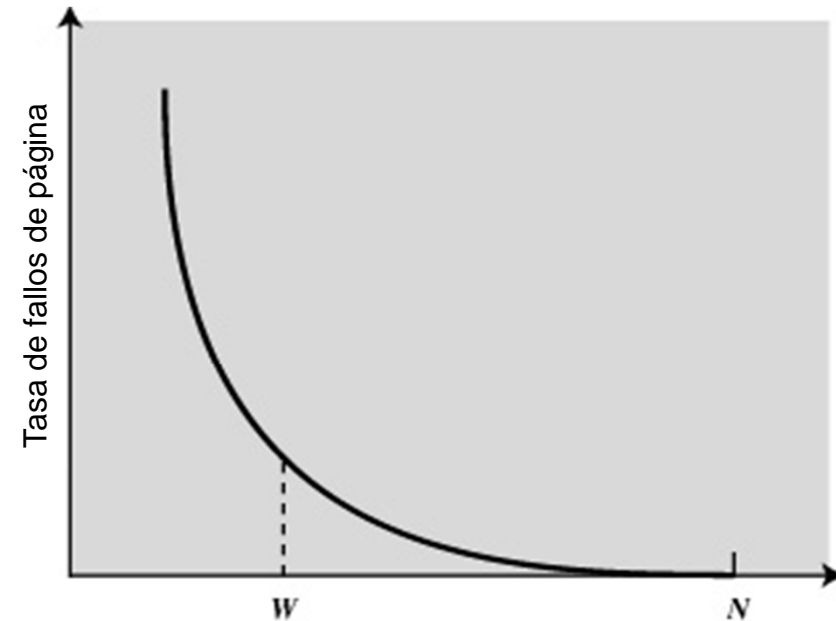




PAGINACION – TAMAÑO DE PAGINA



(a) Tamaño de página



(b) Número de marcos de página asignados

$P$  = Tamaño del proceso completo  
 $W$  = Tamaño del conjunto de trabajo  
 $N$  = Número total de páginas del proceso

Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4



Figura 8.11. Comportamiento típico de la paginación en un programa.



**PAGINACION – TAMAÑO DE PAGINA**

Tabla 8.2. Ejemplos de tamaños de páginas.

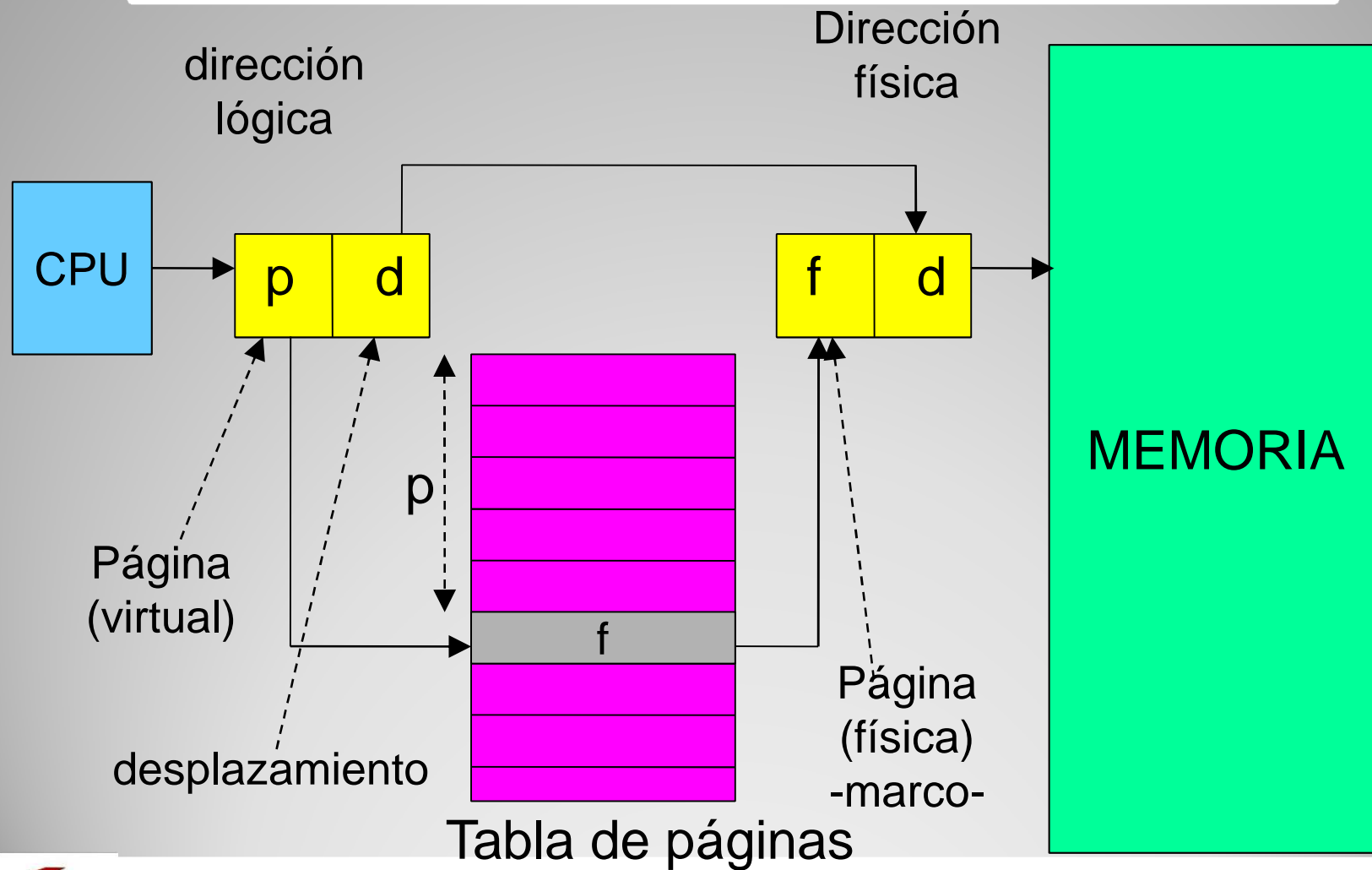
Computadora	Tamaño de página
Atlas	512 palabras de 48 bits
Honeywell-Multics	1.024 palabras de 36 bits
IBM 370/XA y 370/ESA	4 Kbytes
Familia VAX	512 bytes
IBM AS/400	512 bytes
DEC Alpha	8 Kbytes
MIPS	de 4 Kbytes a 16 Mbytes
UltraSPARC	de 8 Kbytes a 4 Mbytes
Pentium	de 4 Kbytes a 4 Mbytes
Power Pc	4 Kbytes



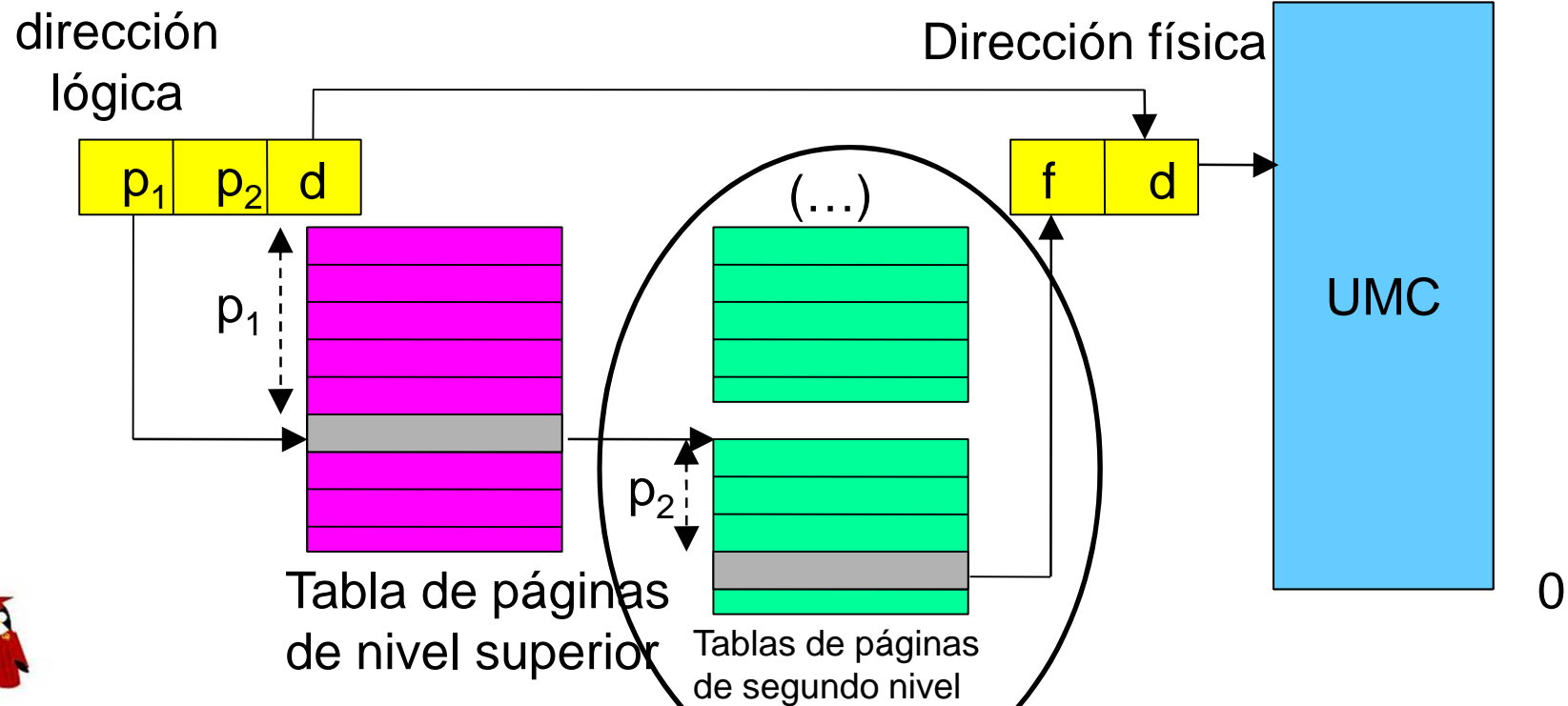
Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4



PAGINACION – CONVERSION DE DIRECCIONES



PAGINACION – CONVERSION DE DIRECCIONES

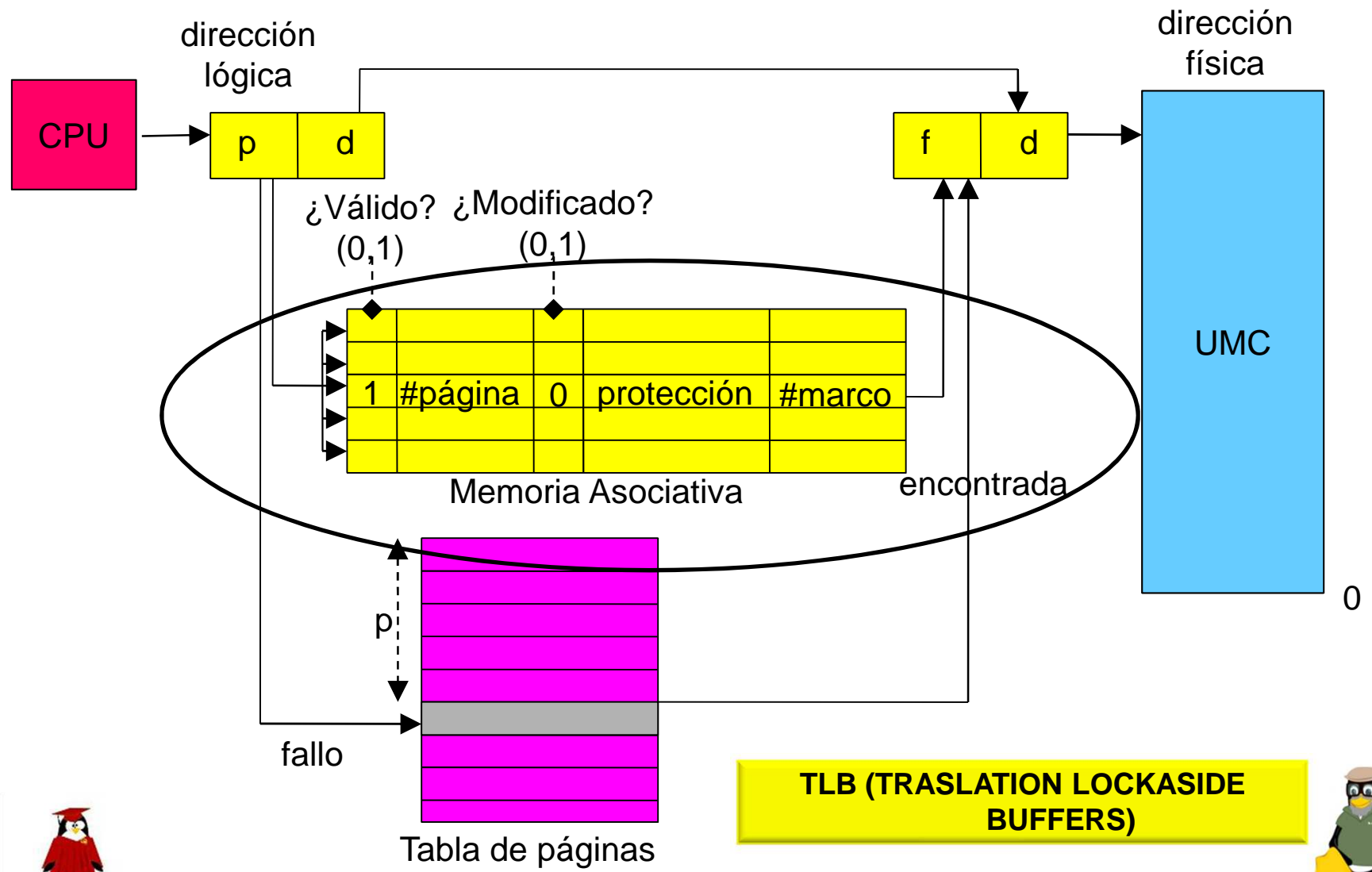


**Multinivel**

- ❖ - Objetivo: evitar tener siempre en memoria tablas de páginas completas
- ❖ - Solución: dividir la tabla en sub-tablas y mantener en memoria sólo las que sean necesarias en cada momento



**PAGINACION – CONVERSION DE DIRECCIONES**



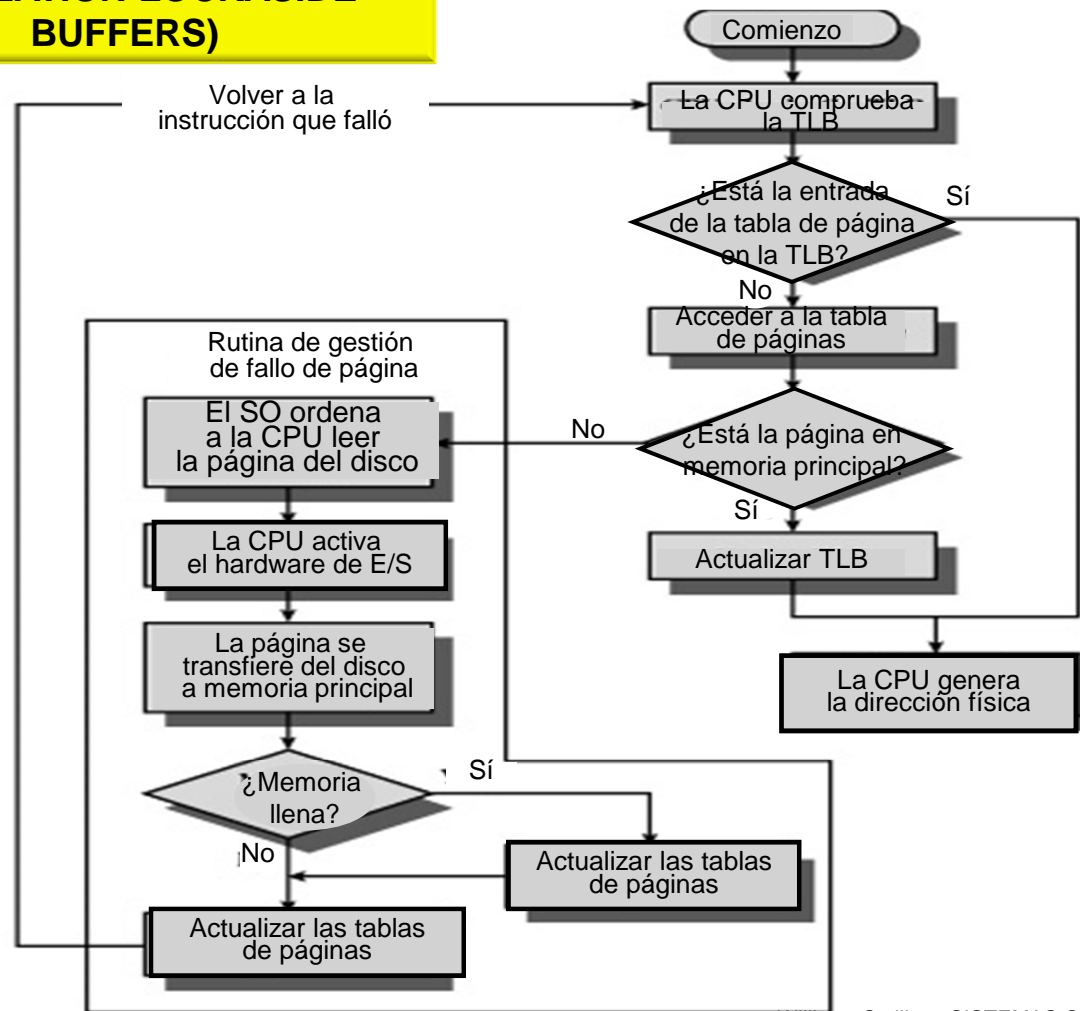
## PAGINACION – CONVERSION DE DIRECCIONES

### TBL:

- ❖ - Solución para acelerar el acceso a los marcos cuando las tablas de paginas son muy grandes (búsqueda lenta) y/o están organizadas en niveles (requiere múltiples accesos a memoria)
- ❖ - Se basa en la observación de que los procesos acceden normalmente a un número pequeño de páginas (y esporádicamente al resto)
- ❖ - Solución: dotar a los ordenadores con hardware (Memoria Asociativa) para asociar algunas páginas de uso frecuente con páginas físicas sin necesidad de acceder a la tabla de páginas. El tamaño de la memoria asociativa suele ser de 8 a 32 entradas.
- ❖ - *Proporción de encuentros*: Proporción de accesos a la Memoria Asociativa que son exitosos (la Página Virtual buscada se encuentra en la Memoria)



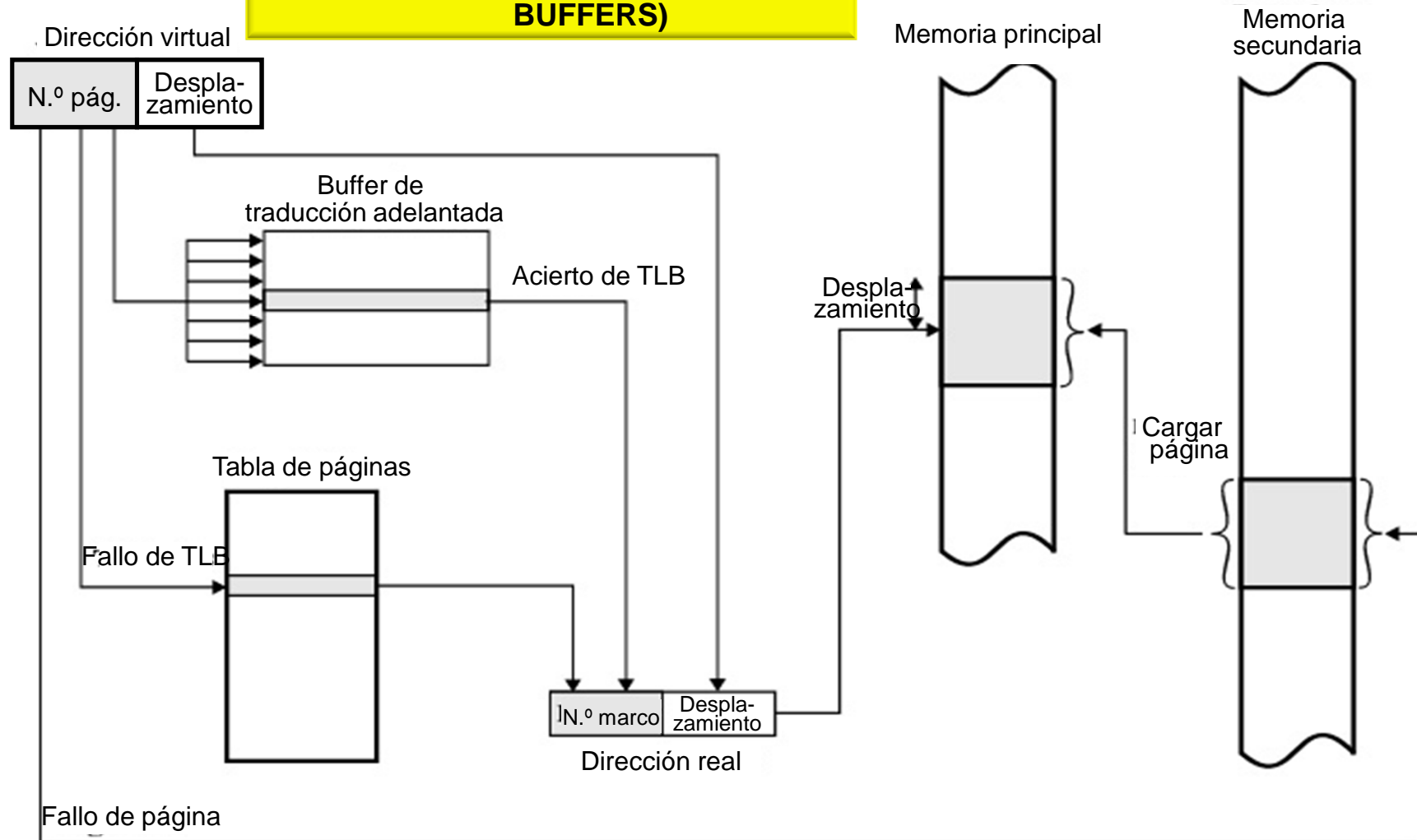
**TLB (TRASLATION LOCKASIDE BUFFERS)**



Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4

Figura 8.8. Funcionamiento de la paginación con buffer de traducción adelantada (TLB) [FUTH87].

**TLB (TRASLATION LOCKASIDE BUFFERS)**



Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4

Figura 8.7. Uso de un Buffer de Traducción Adelantada.



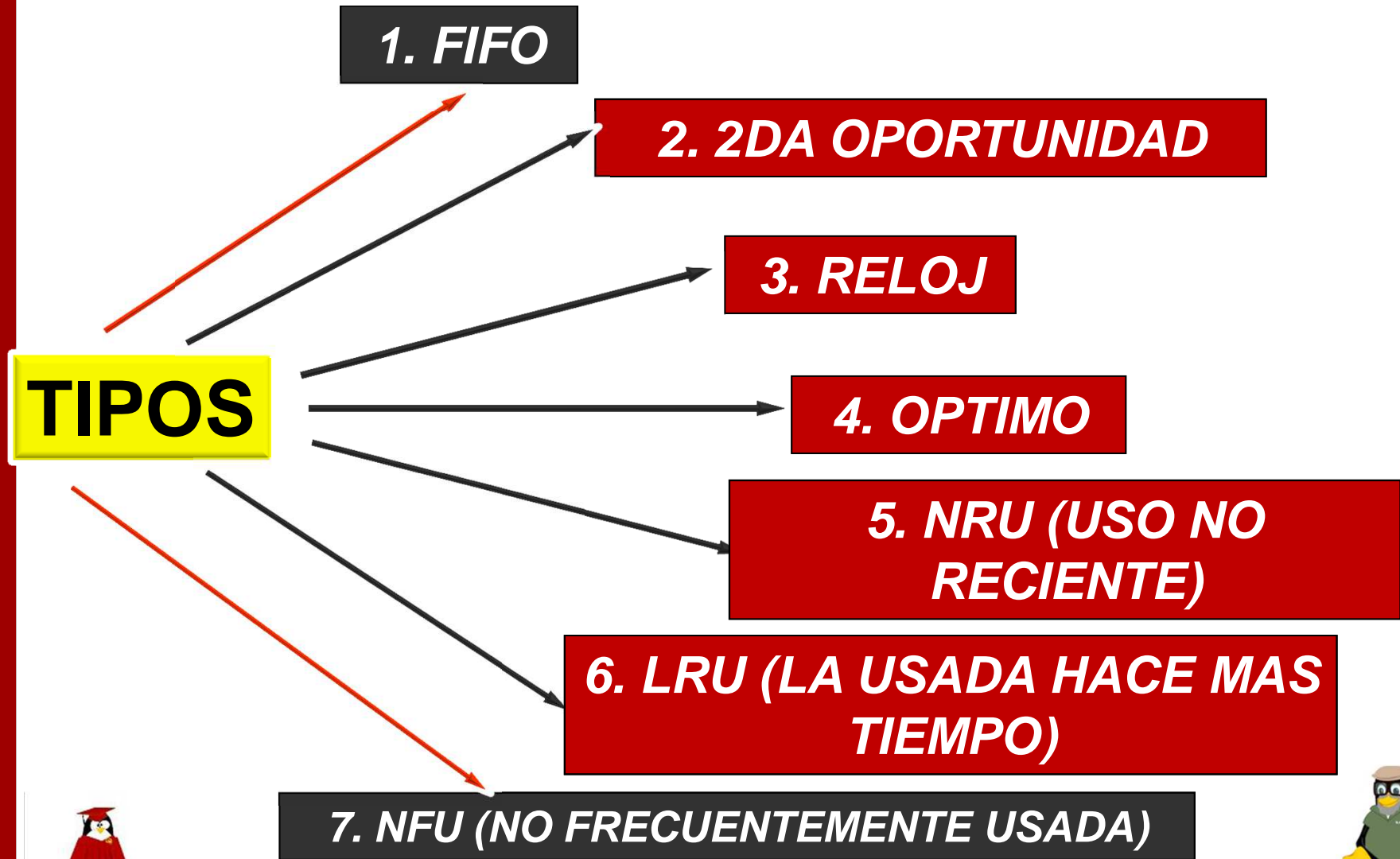


**PAGINACION – ALGORITMOS DE SELECCION DE  
PAGINA A REEMPLAZAR**

- ❖ **Tras un fallo de página, el SO debe elegir qué página de memoria deber ser intercambiada disco para hacer sitio a la nueva página que se está solicitando**
- ❖ **Criterio general: eliminar páginas poco usadas.**
- ❖ **Ejemplo. Secuencia de accesos:**



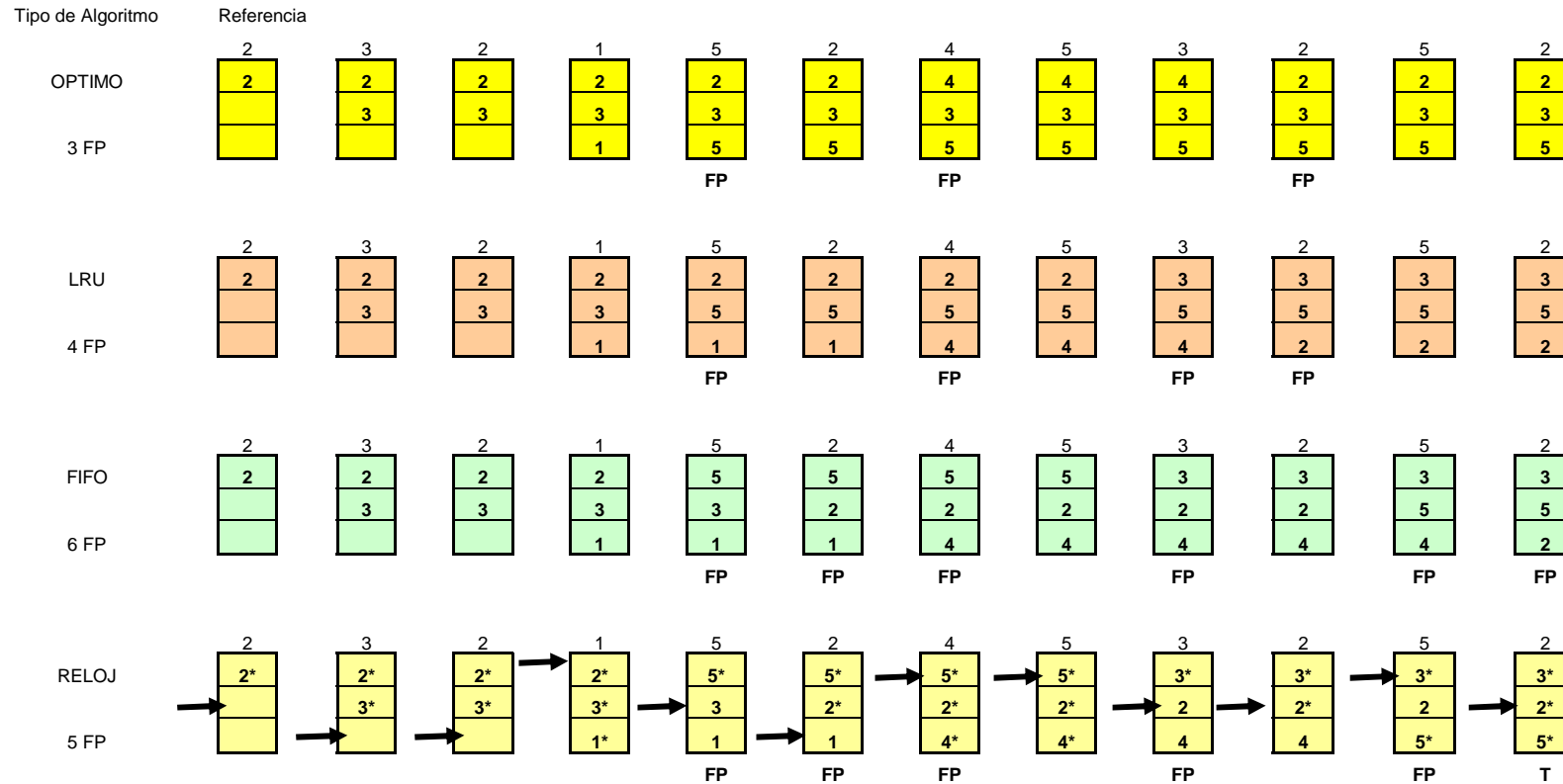
PAGINACION – ALGORITMOS DE SELECCION DE PAGINA A REEMPLAZAR



## ALGORITMOS DE REEMPLAZO DE PAGINA

Dada la siguiente secuencia de Referencia:

2 3 2 1 5 2 4 5 3 2 5 2



Bit de Uso: 1

Flecha corresponde al puntero



## ALGORITMOS DE REEMPLAZO DE PAGINA

Mi proceso tiene 8 paginas asignadas y dispongo de un espacio de memoria de tres marcos, desarrollar según :

Dada la siguiente secuencia de Referencia:

1 5 2 3 3 3 1 4 5 1 2 2 7 4 2 1

Tipo de Algoritmo	Referencia	1	5	2	3	3	3	1	4	5	1	2	2	7	4	2	1
OPTIMO		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
LRU		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
FIFO		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
RELOJ		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Bit de Uso: 1

Flecha corresponde al puntero





## SEGMENTACION – CONCEPTOS GENERALES -



- ❖ No es necesario que todos los segmentos de todos los programas tengan la misma longitud.
- ❖ Existe una longitud máxima de segmento.
- ❖ Un dirección lógica segmentada consta de dos partes, un número de segmento y un desplazamiento.
- ❖ Como consecuencia del empleo de segmentos de distinto tamaño, la segmentación resulta similar a la partición dinámica.

## SEGMENTACION – CONCEPTOS GENERALES -

- ❖ 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.



**SEGMENTACION – CONCEPTOS GENERALES -**



Dirección virtual



Entrada del segmento de tabla



(b) Sólo segmentación

Figura 8.2. Formatos típicos de gestión de memoria.



Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4

## SEGMENTACION PAGINADA

- ❖ 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.





**SEGMENTACION PAGINADA**

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

(c) Segmentación y paginación combinadas

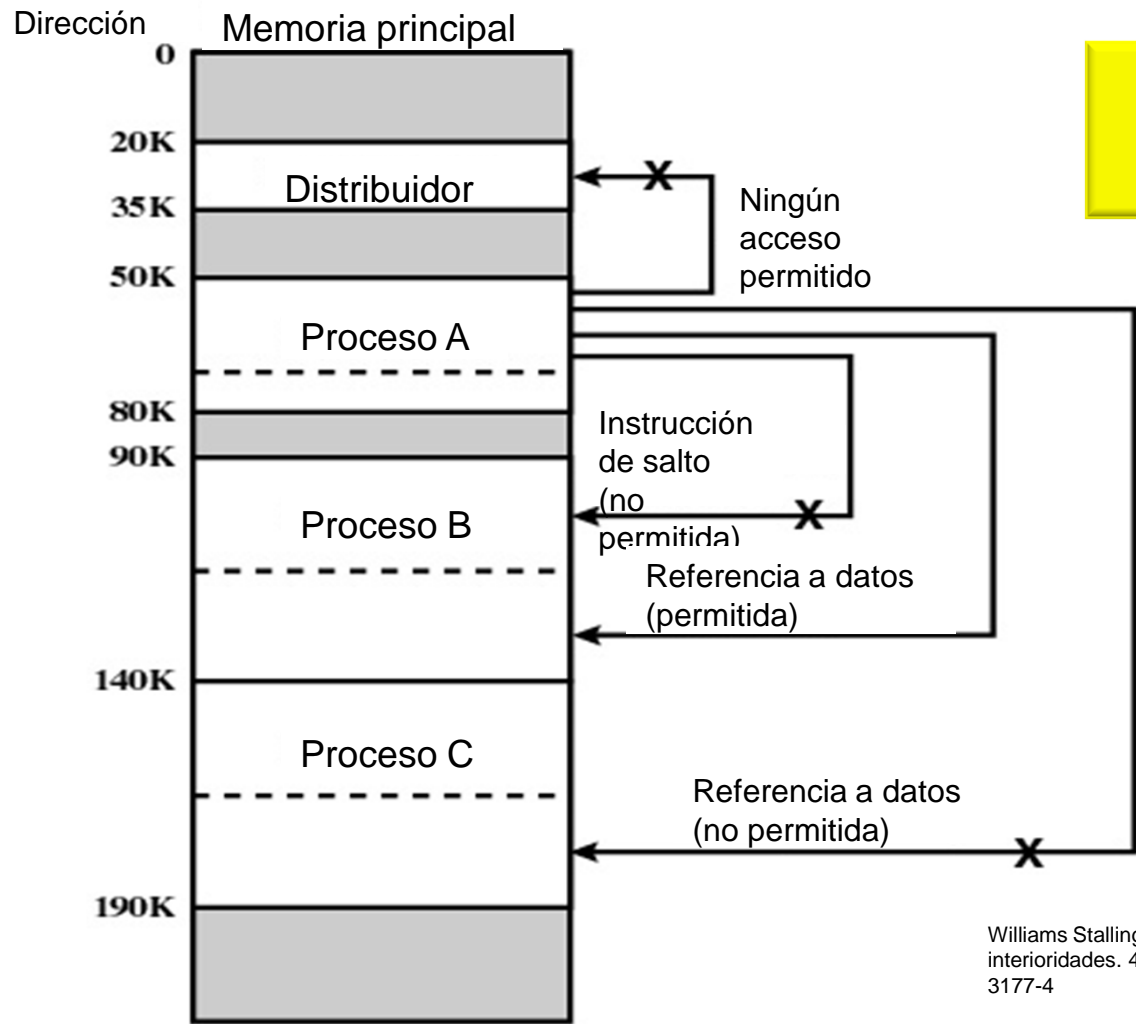
Figura 8.2. Formatos típicos de gestión de memoria.



Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4



**SEGMENTACION  
PAGINADA**



Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4

Figura 8.14. Relaciones de protección entre segmentos.



## Bibliografía

1. Programación en Linux, con ejemplos. Kurt Wall. QUE, Prentice Hall. Madrid. 2000.
2. Sistemas Operativos. 5ta Ed. William Stalling. Pearson Prentice Hall. Madrid. 2006
3. Sistemas Operativos. 7ma Ed. William Stalling. Pearson Prentice Hall. Madrid. 2012
4. Sistemas Operativos Modernos. Andrew. S. Tanenbaum. Prentice-Hall. Interamericana S.A. Madrid, 2009.
5. Unix, Sistema V Versión 4. Rosen,Rozinsky y Farber.McGraw Hill. NY 2000.
6. Lunix, Edición especial. Jack Tackett, David Guntery Lance Brown. Ed. Prentice Hall. 1998.
7. El Libro de Linux. Syed M. Sarwar, Robert Koretsky y Syed. A. Sarwar. Ed. Addison Wesley. 2007. España.



**FIN UNIDAD 6B**  
**ADMINISTRACION DE**  
**MEMORIA**



**May the force be with you**