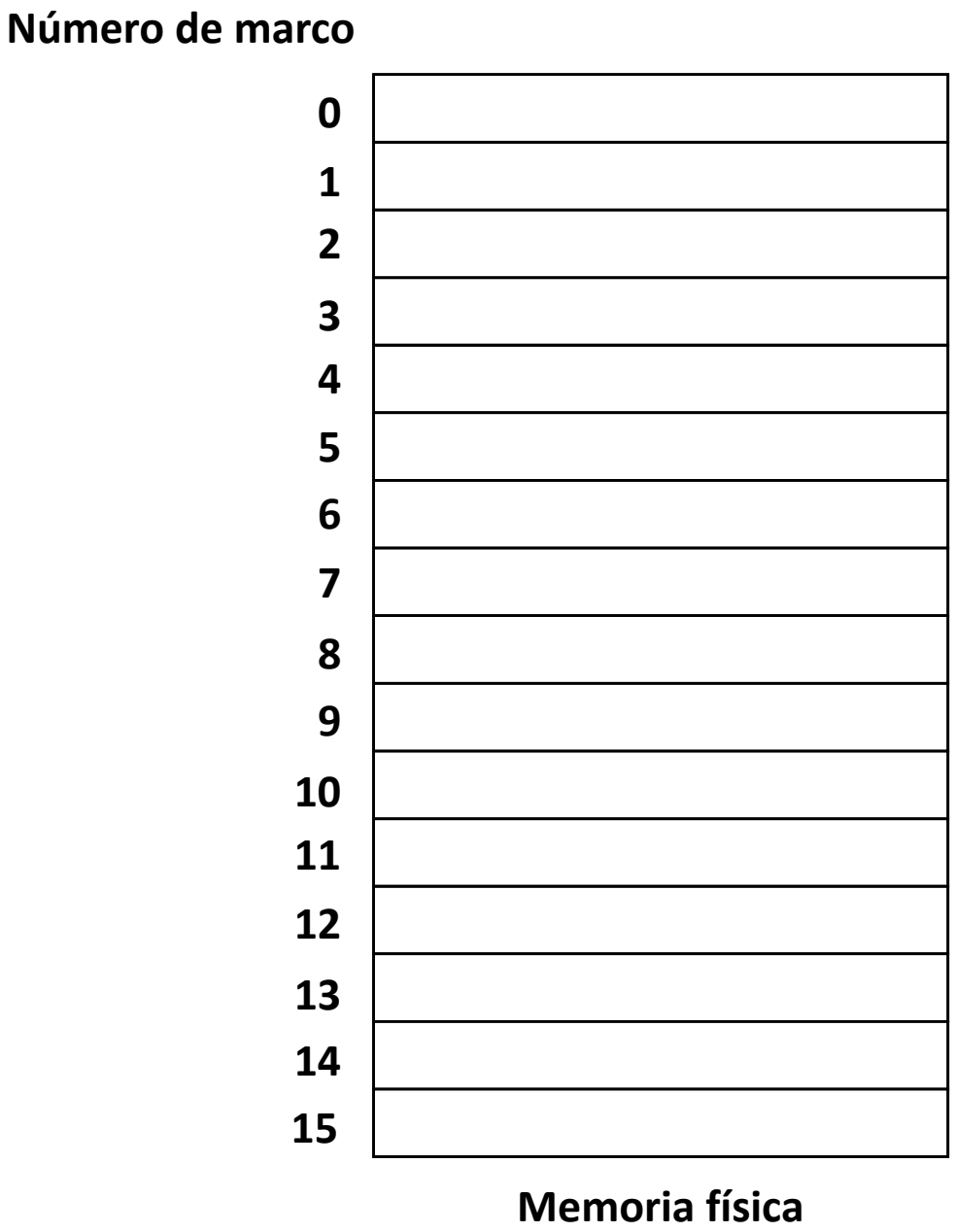
Paginación de memoria

Adaptación (ver referencias al final)

Paginación

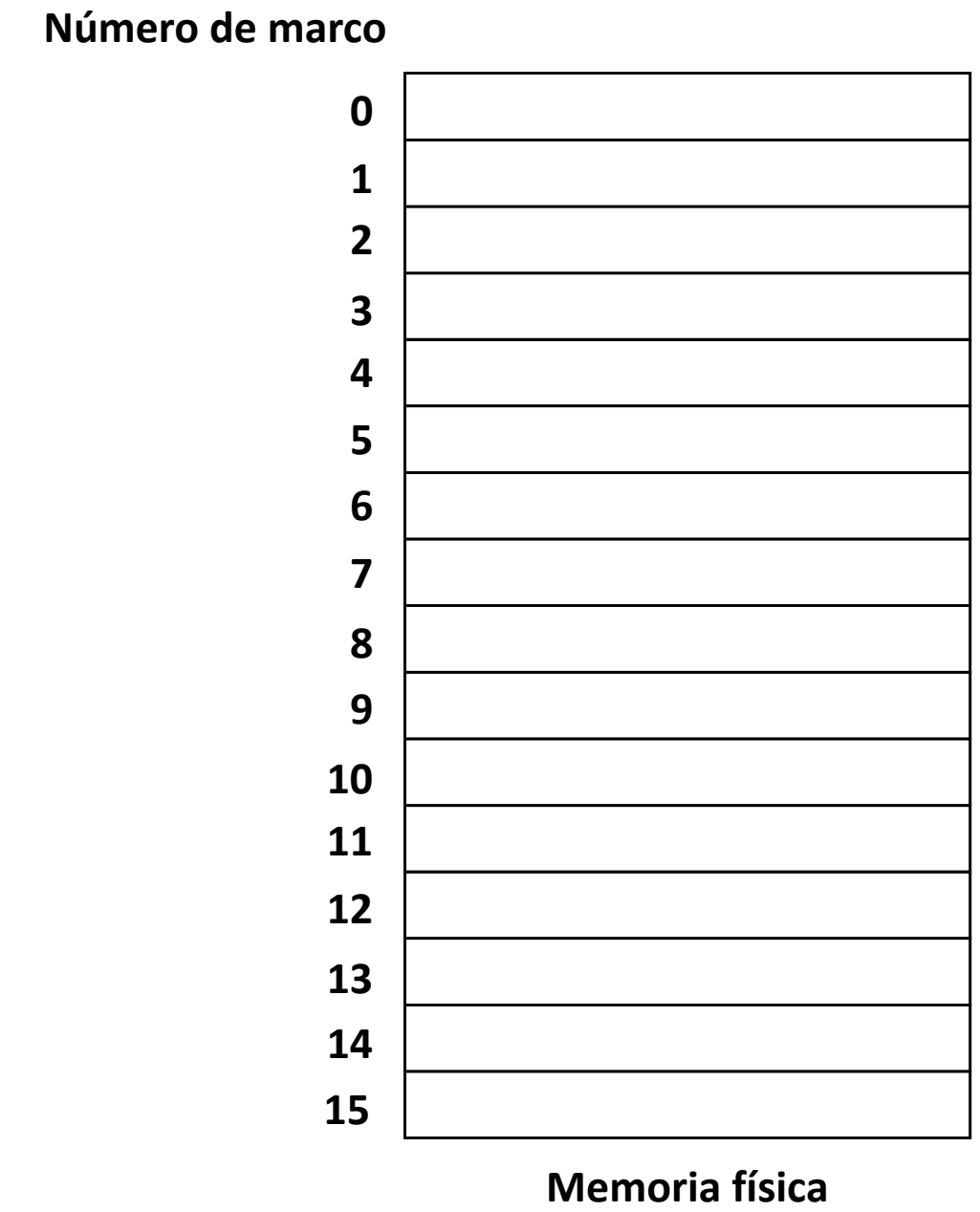
- Evita la fragmentación externa
- Técnica usada en la mayoría de sistemas operativos modernos
- Dividir la **memoria física** en bloques de tamaño fijo llamados **marcos**.
- Tamaño del bloque es una potencia de 2: 512 Bytes a 8192 bytes



Paginación

- La **memoria lógica** también se va a dividir en bloques **del mismo tamaño** de los marcos
- Estos bloques se llaman **páginas**.

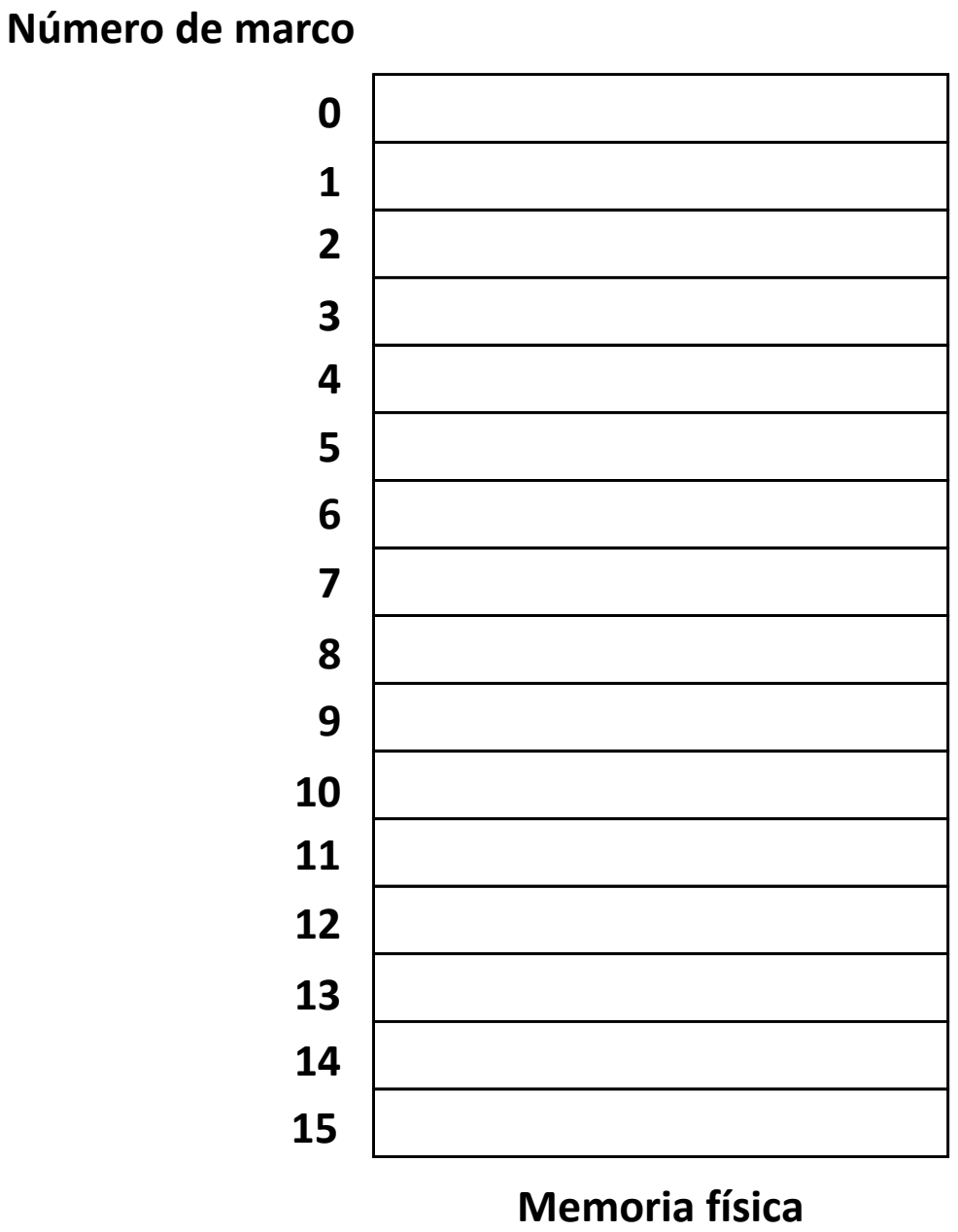
Proceso A
A1
A2
A3
A4
A5



Paginación

- Las páginas se cargan en los marcos.
- El espacio de direcciones lógicas de un proceso puede no estar contiguo en memoria física.

Proceso A
A1
A2
A3
A4
A5

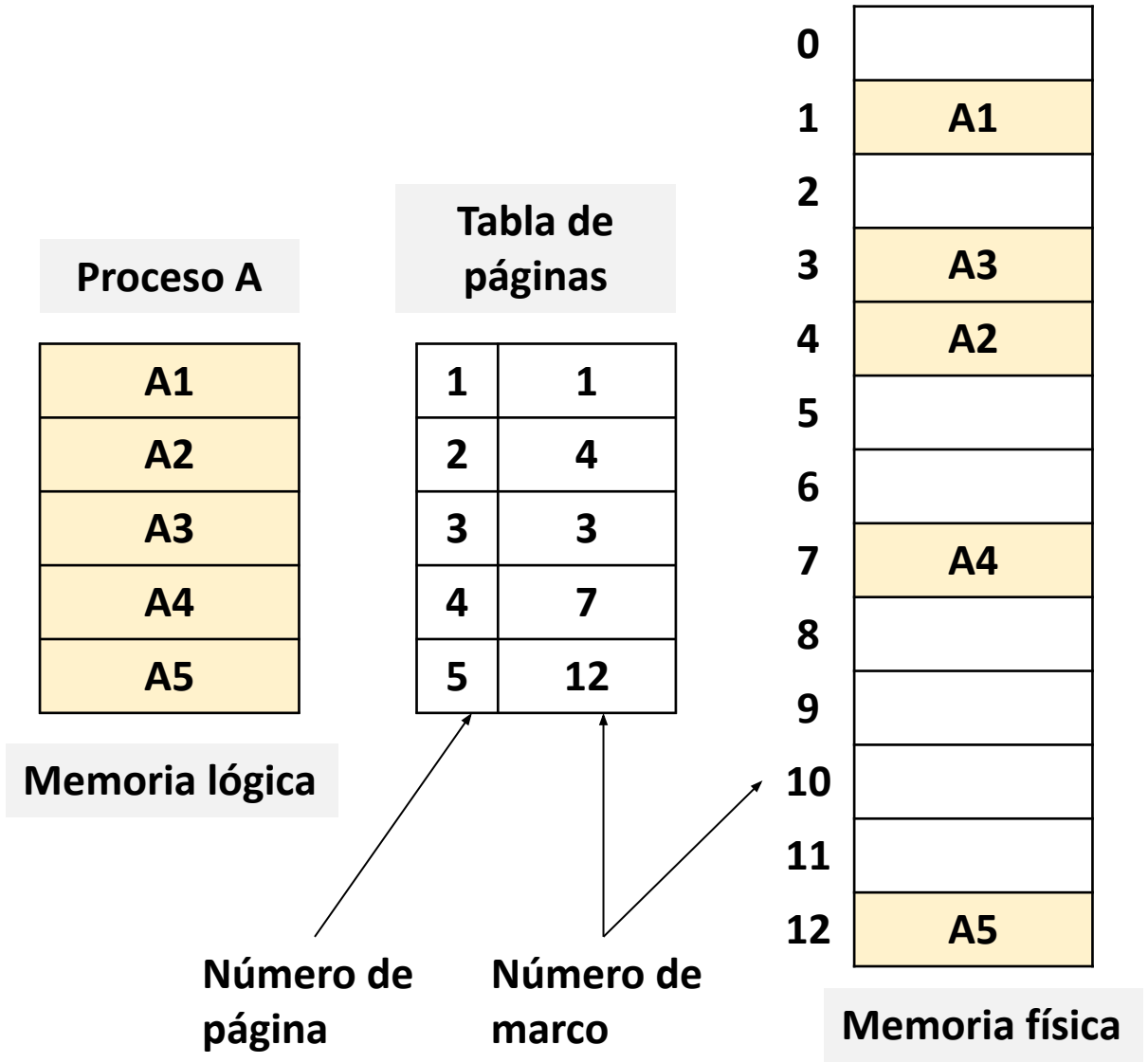


Paginación

- Se logra separar el espacio lógico del espacio físico.
- Podría existir más espacio lógico que físico.
  - Proceso podría pensar que tiene más de 2^{64} bytes direccionables.
  - Memoria física podría ser de menos de 2^{64} bytes.

Traducción de las direcciones

- Las direcciones virtuales generadas por la CPU necesitan traducción.
- Para la traducción se usa una estructura llamada **tabla de páginas**.
- La **tabla de páginas** es **por cada proceso**.



Traducción de las direcciones

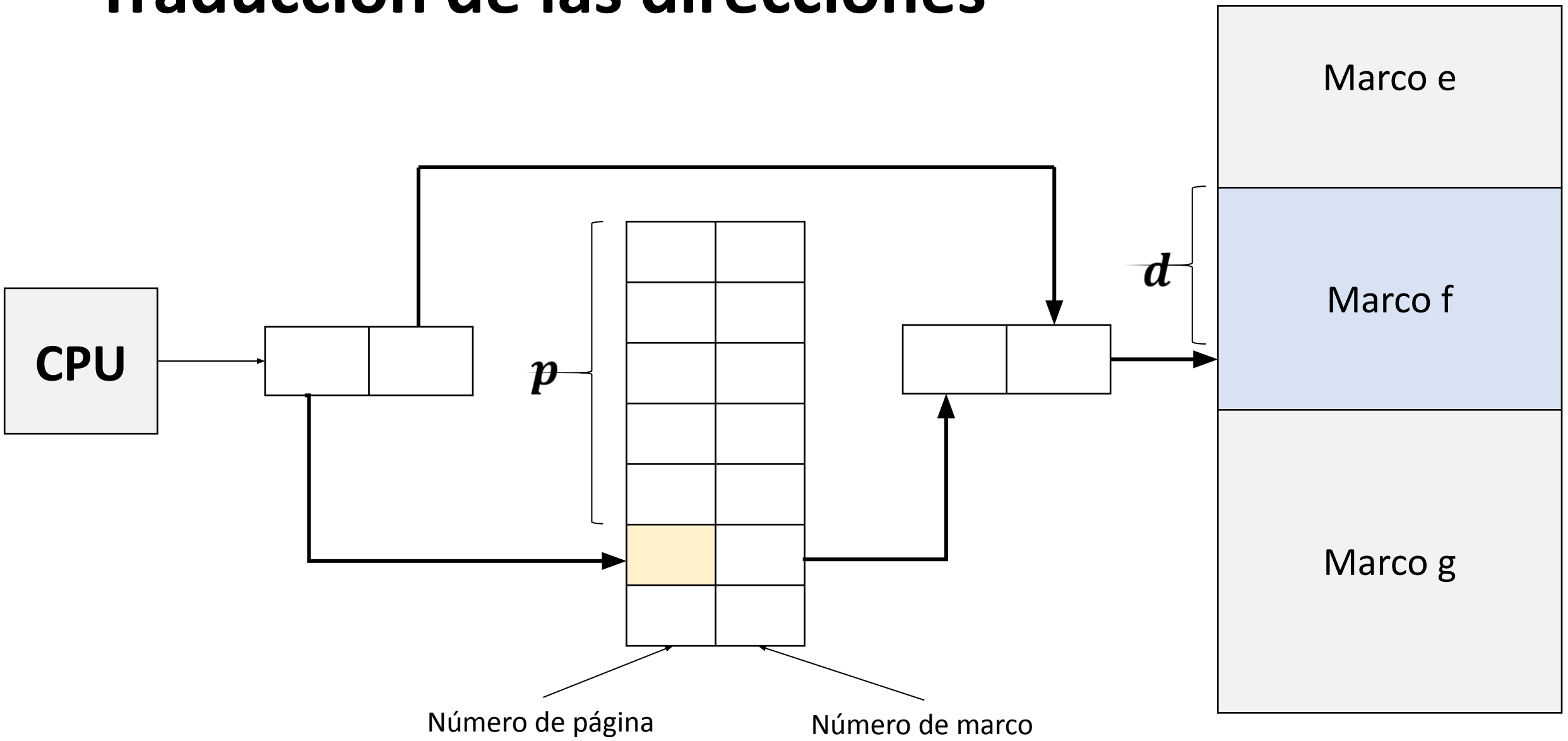
- Las direcciones **lógicas** se dividen en
  - Número de página (p)
  - Desplazamiento (d)
- p se usa como índice en la tabla de páginas de cada proceso.
- d se usa para referenciar la información al interior de cada marco.

Número de página	Desplazamiento

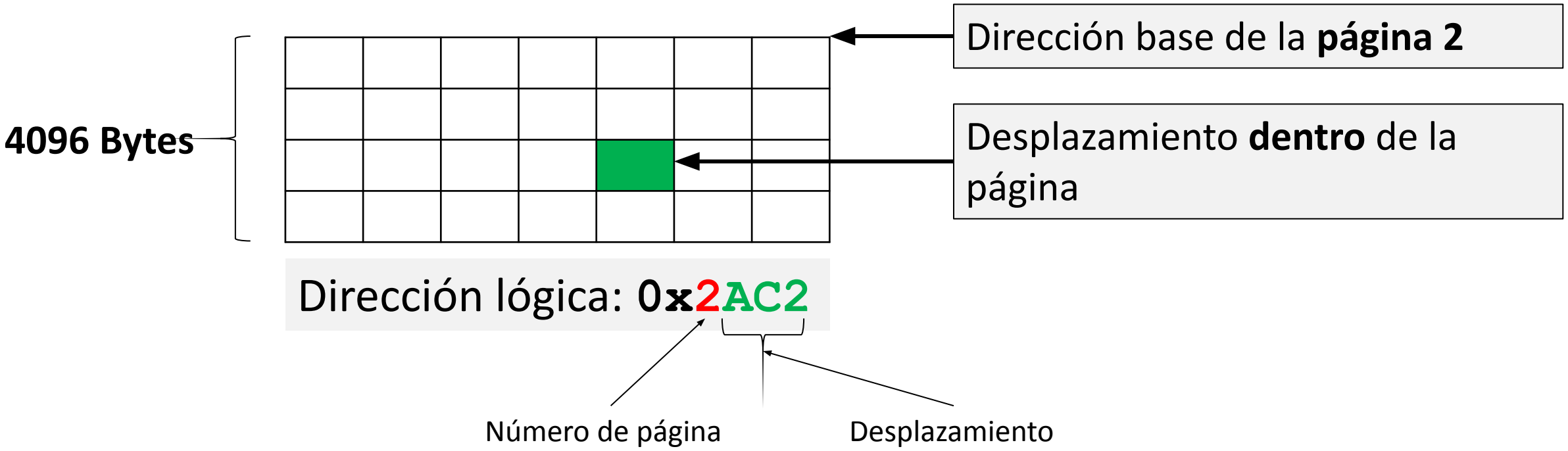
1. Obtener el número de página y usarlo como índice en la tabla de páginas.
2. Obtener el número de marco (f).
3. Reemplazar p por f. Esta es la dirección física.

- d se mantiene igual en la dirección.

Traducción de las direcciones



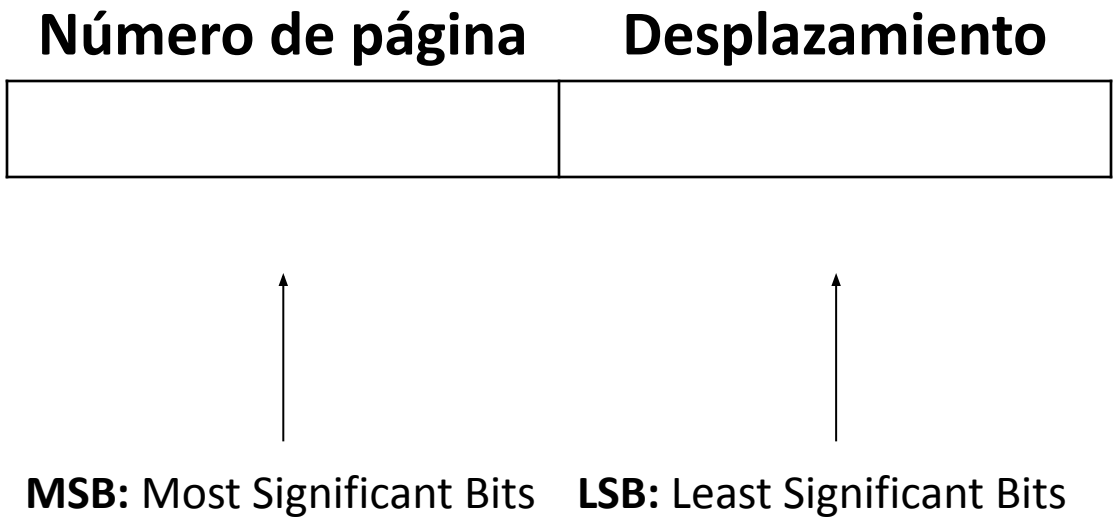
¿Qué es el desplazamiento?



- El desplazamiento indica cuánto hay que desplazarse desde la dirección base de la página hasta el byte que se está referenciando.
- La memoria se puede ver como un arreglo de bytes donde cada byte tiene una dirección de memoria.

¿Cuáles bits son de p y cuáles de d?

- Suponga espacio **lógico** de direccionamiento de 2^m
- Suponga tamaño de páginas de 2^n
- Dirección lógica



Proceso A

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

4 Bytes

Memoria lógica

Tabla de páginas

0	5
1	6
2	1
3	2

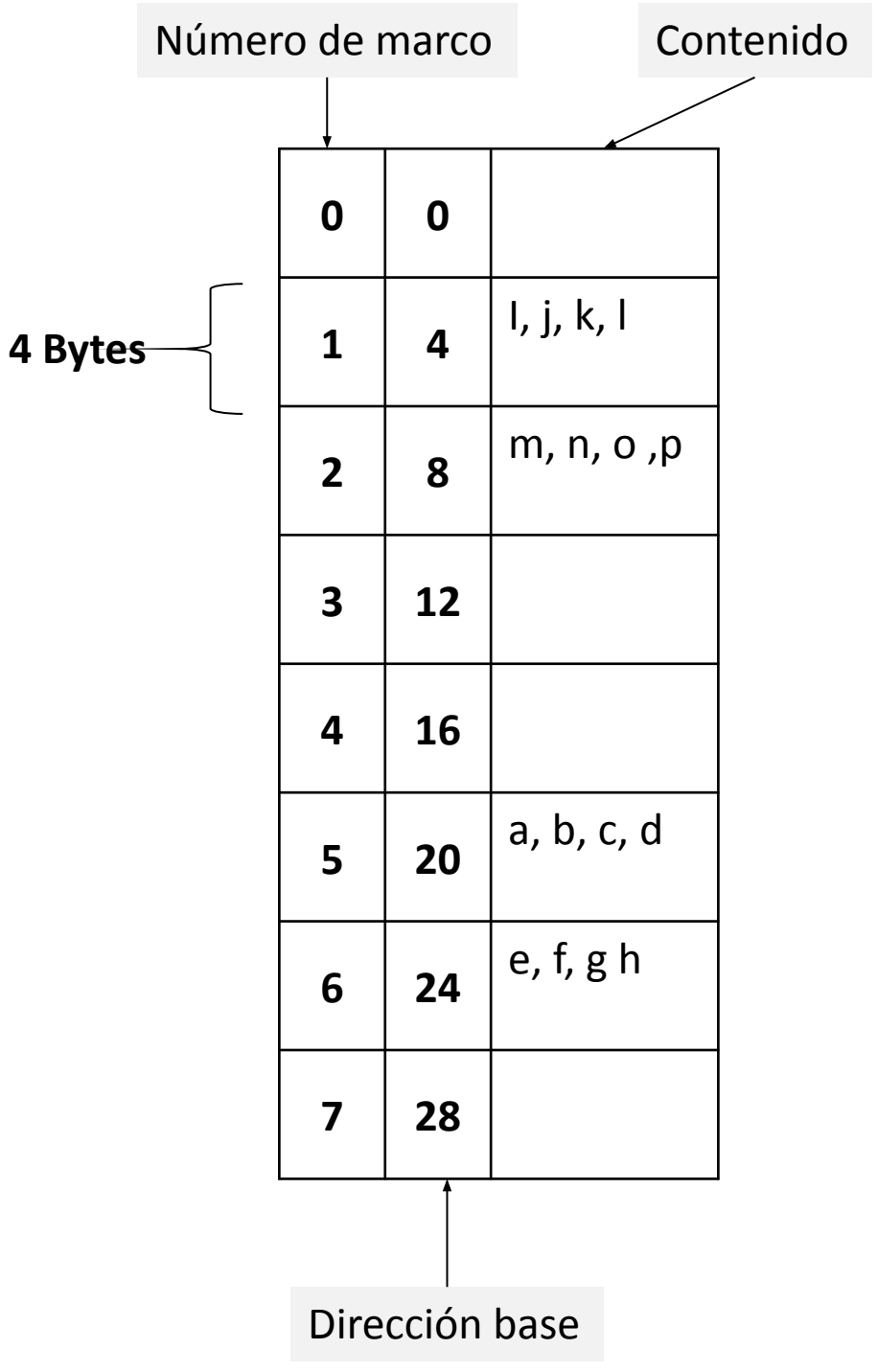
$n = 2 \rightarrow 2^2 = 4$ Bytes

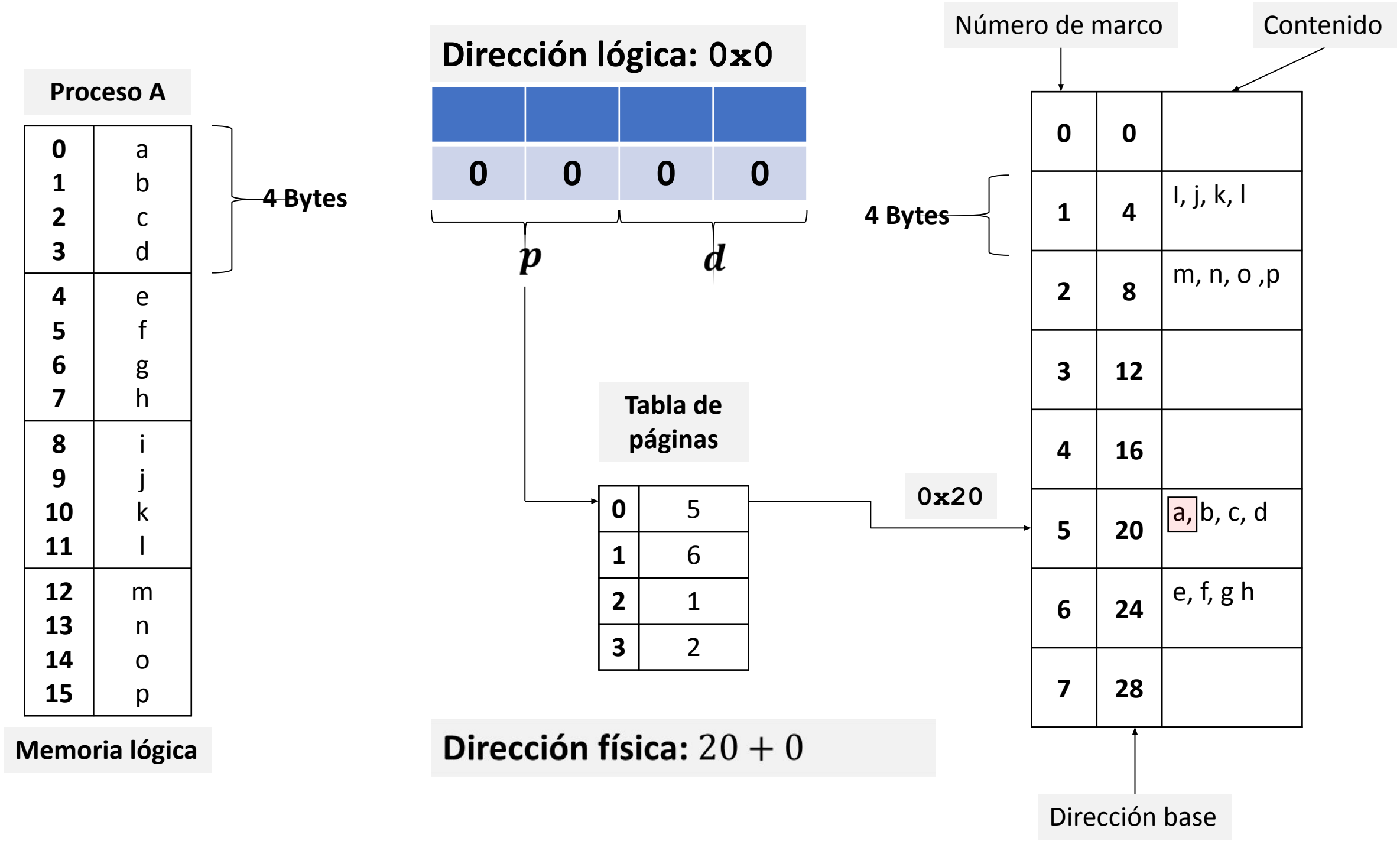
$m = 4 \rightarrow 2^4 = 16$ Bytes

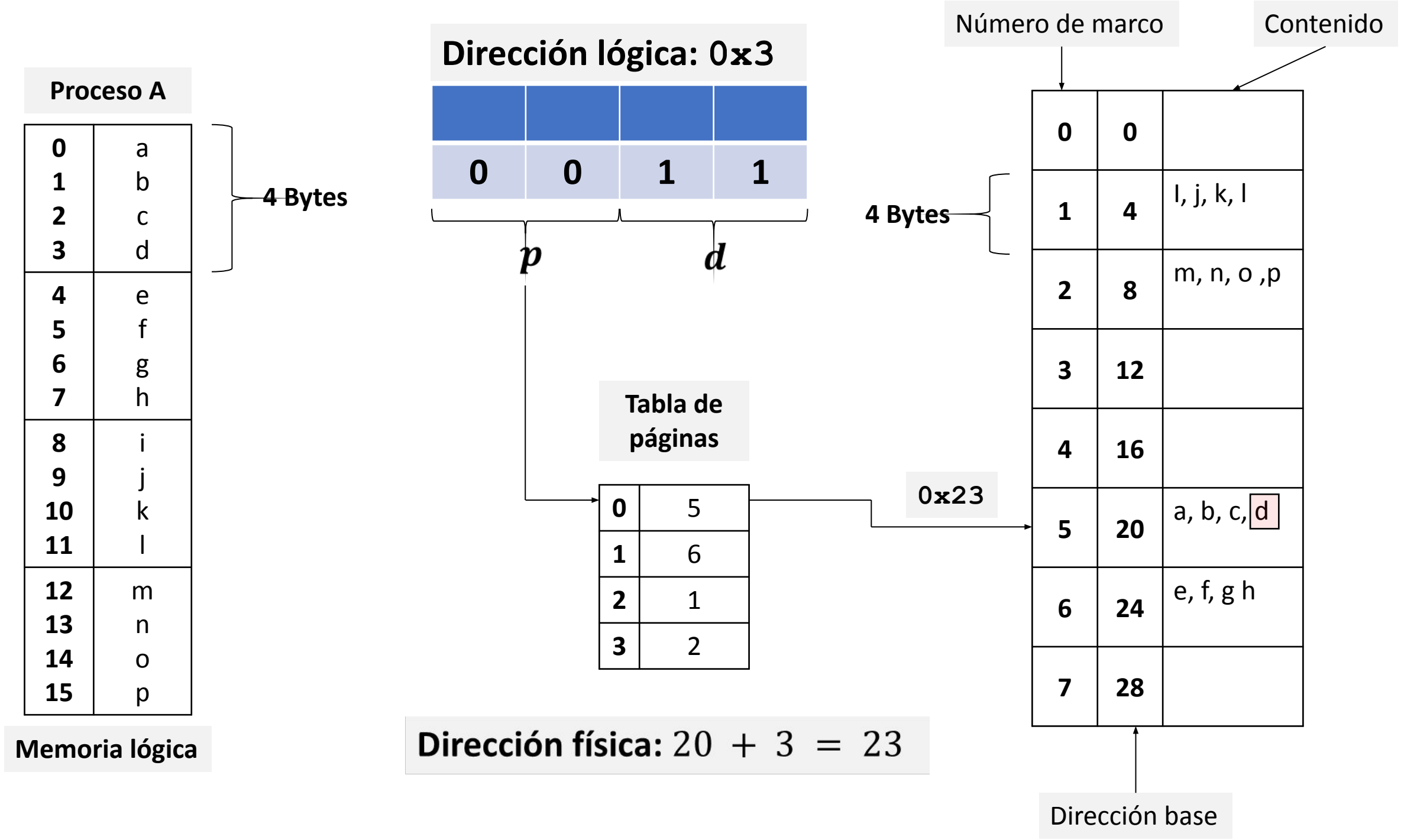
- **Dirección lógica de 4 bits, ya que**

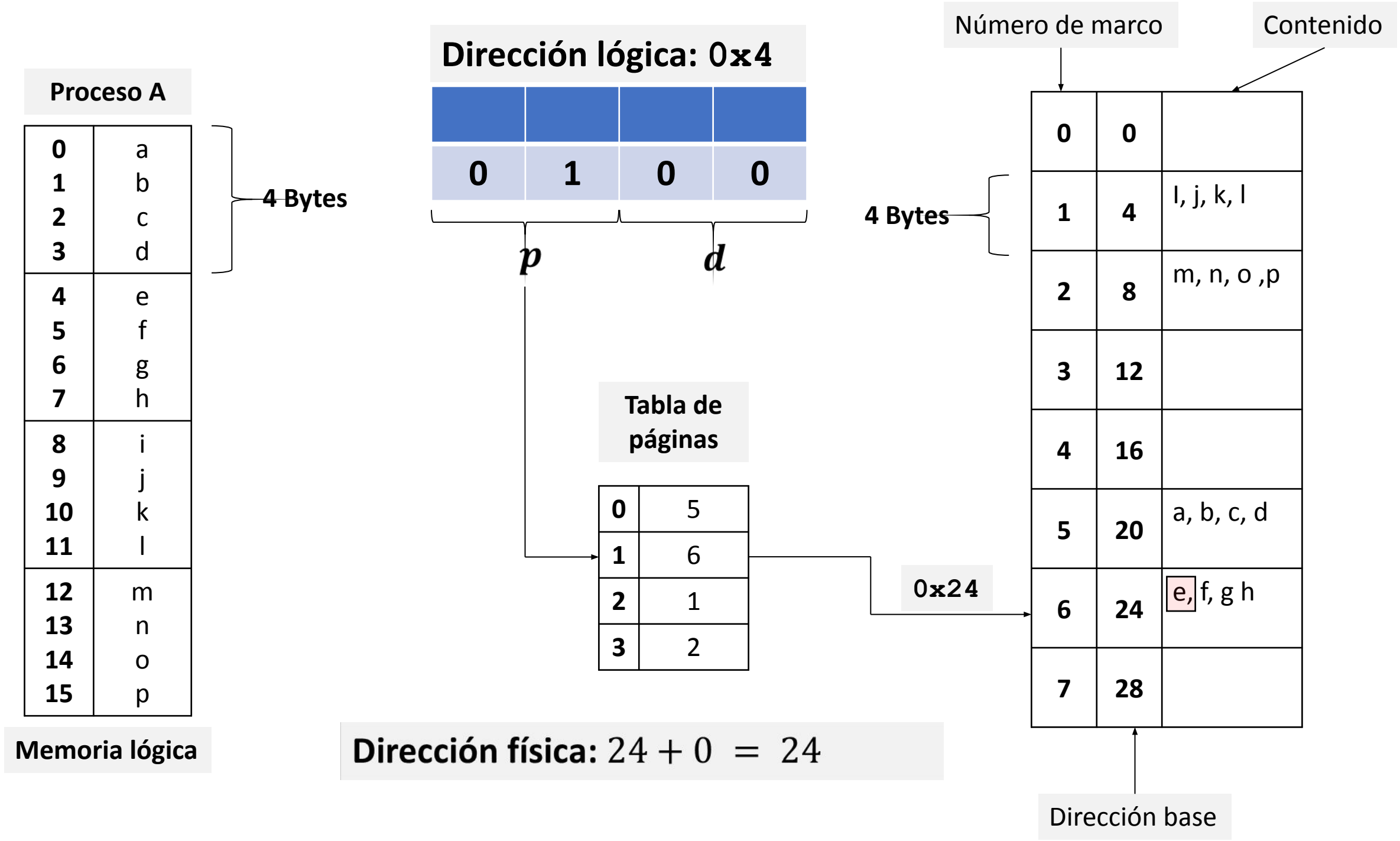
$p: m - n = 2$ Bits

$d: n = 2$ Bits




Proceso A
0 a
1 b
2 c
3 d
4 e
5 f
6 g
7 h
8 i
9 j
10 k
11 l
12 m
13 n
14 o
15 p
4 Bytes
Memoria lógica
Dirección lógica: 0x0
0 0 0 0
p
d
Tabla de páginas
0 5
1 6
2 1
3 2
Número de marco
Contenido
4 Bytes
0 0
1 4 l, j, k, l
2 8 m, n, o ,p
3 12
4 16
5 20 a, b, c, d
6 24 e, f, g h
7 28
0x20
Dirección física: 20 + 0
Dirección base


Proceso A
0 a
1 b
2 c
3 d
4 e
5 f
6 g
7 h
8 i
9 j
10 k
11 l
12 m
13 n
14 o
15 p
4 Bytes
Memoria lógica
Dirección lógica: 0x3
0 0 1 1
p
d
Tabla de páginas
0 5
1 6
2 1
3 2
Número de marco
Contenido
4 Bytes
0 0
1 4 l, j, k, l
2 8 m, n, o ,p
3 12
4 16
5 20 a, b, c, d
6 24 e, f, g h
7 28
0x23
Dirección física: 20 + 3 = 23
Dirección base


Proceso A
0 a
1 b
2 c
3 d
4 e
5 f
6 g
7 h
8 i
9 j
10 k
11 l
12 m
13 n
14 o
15 p
4 Bytes
Memoria lógica
Dirección lógica: 0x4
0 1 0 0
p
d
Tabla de páginas
0 5
1 6
2 1
3 2
Número de marco
Contenido
4 Bytes
0 0
1 4 l, j, k, l
2 8 m, n, o ,p
3 12
4 16
5 20 a, b, c, d
6 24 e, f, g h
7 28
0x24
Dirección base
Dirección física: 24 + 0 = 24

Proceso A

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

4 Bytes

Memoria lógica

Dirección lógica: 0xD

1	**1**	**0**	**1**

p d

Tabla de páginas

0	5
1	6
2	1
3	2

Dirección física: $8 + 1 = 9$

Número de marco

Contenido

4 Bytes

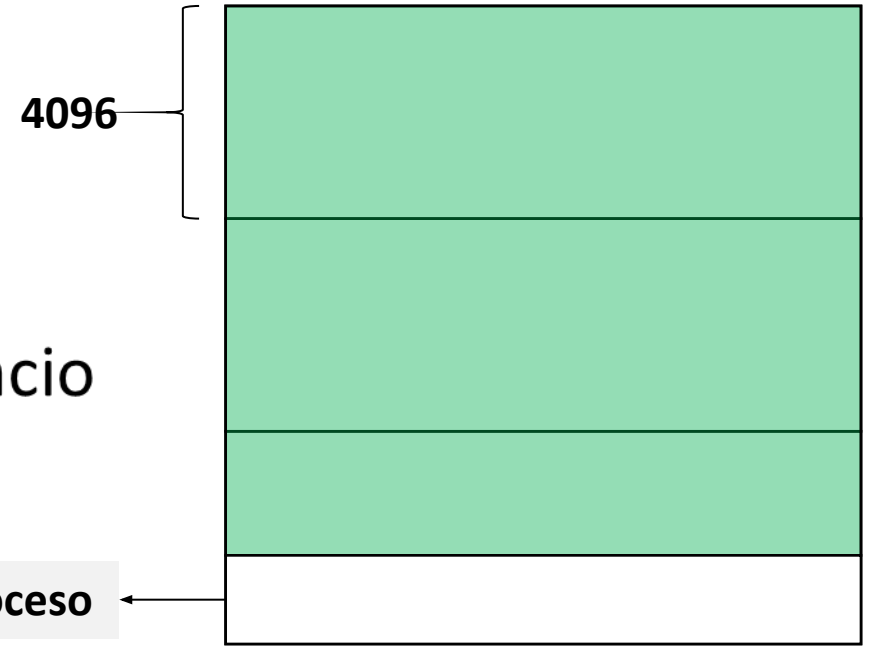
0x9

0	**0**	
1	**4**	l, j, k, l
2	**8**	m, n, o ,p
3	**12**	
4	**16**	
5	**20**	a, b, c, d
6	**24**	e, f, g h
7	**28**	

Dirección base

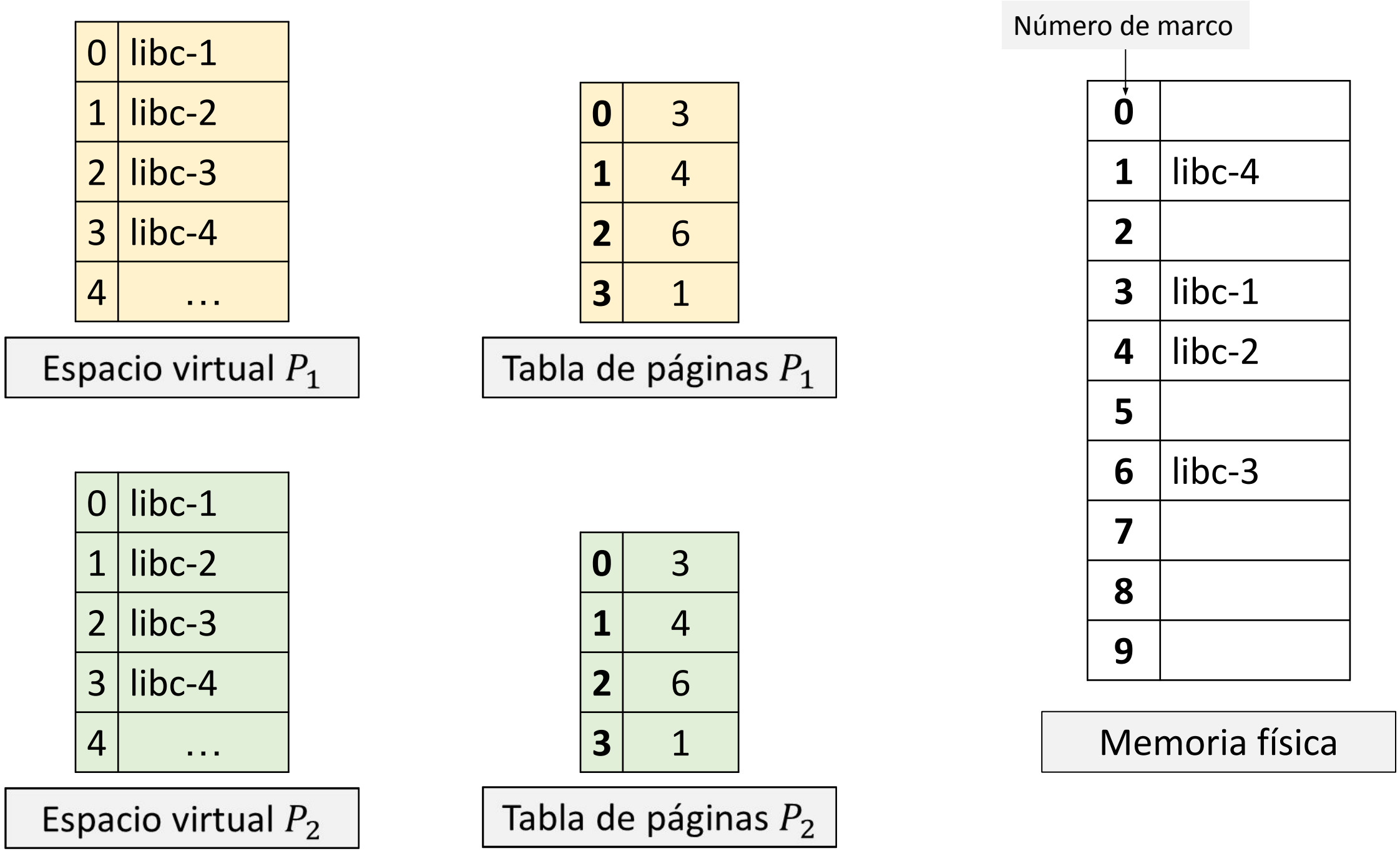
Fragmentación

- No ocurre fragmentación externa pero **SI** podría ocurrir fragmentación interna
- Tamaño de página: **4096 Bytes**
- Proceso mide **10240 Bytes**, necesita
  - $\frac{10240}{4096} = 2.5$ páginas
- Se le asignan tres páginas así le sobre espacio
  - Espacio que se le podría asignar si lo pide



Páginas compartidas

- Se posibilita compartir código entre los procesos
  - Shared Objects (.so)
  - Dynamic Link Libraries (.dll)
- Caso librería estándar de C (P. ej.: `libc-2.17.so`) en Linux
  - Una sola copia en memoria física


0 libc-1
1 libc-2
2 libc-3
3 libc-4
4 …
Espacio virtual P_1
0 3
1 4
2 6
3 1
Tabla de páginas P_1
Número de marco
0
1 libc-4
2
3 libc-1
4 libc-2
5
6 libc-3
7
8
9
Memoria física
0 libc-1
1 libc-2
2 libc-3
3 libc-4
4 …
Espacio virtual P_2
0 3
1 4
2 6
3 1
Tabla de páginas P_2

```
0000000000400000      4K r-x-- va
0000000000600000      4K r---- va
0000000000601000      4K rw--- va
0000000001df8000    132K rw---   [ anon ]
00007f851ee9e000   1808K r-x-- libc-2.17.so
00007f851f062000   2044K ----- libc-2.17.so
00007f851f261000     16K r---- libc-2.17.so
00007f851f265000      8K rw--- libc-2.17.so
00007f851f267000     20K rw---   [ anon ]
00007f851f26c000    136K r-x-- ld-2.17.so
00007f851f481000     12K rw---   [ anon ]
00007f851f48b000      8K rw---   [ anon ]
00007f851f48d000      4K r---- ld-2.17.so
00007f851f48e000      4K rw--- ld-2.17.so
00007f851f48f000      4K rw---   [ anon ]
00007fff38715000    132K rw---   [ stack ]
00007fff3878c000      8K r-x--   [ anon ]
ffffffffff600000      4K r-x--   [ anon ]
 total             4352K
```

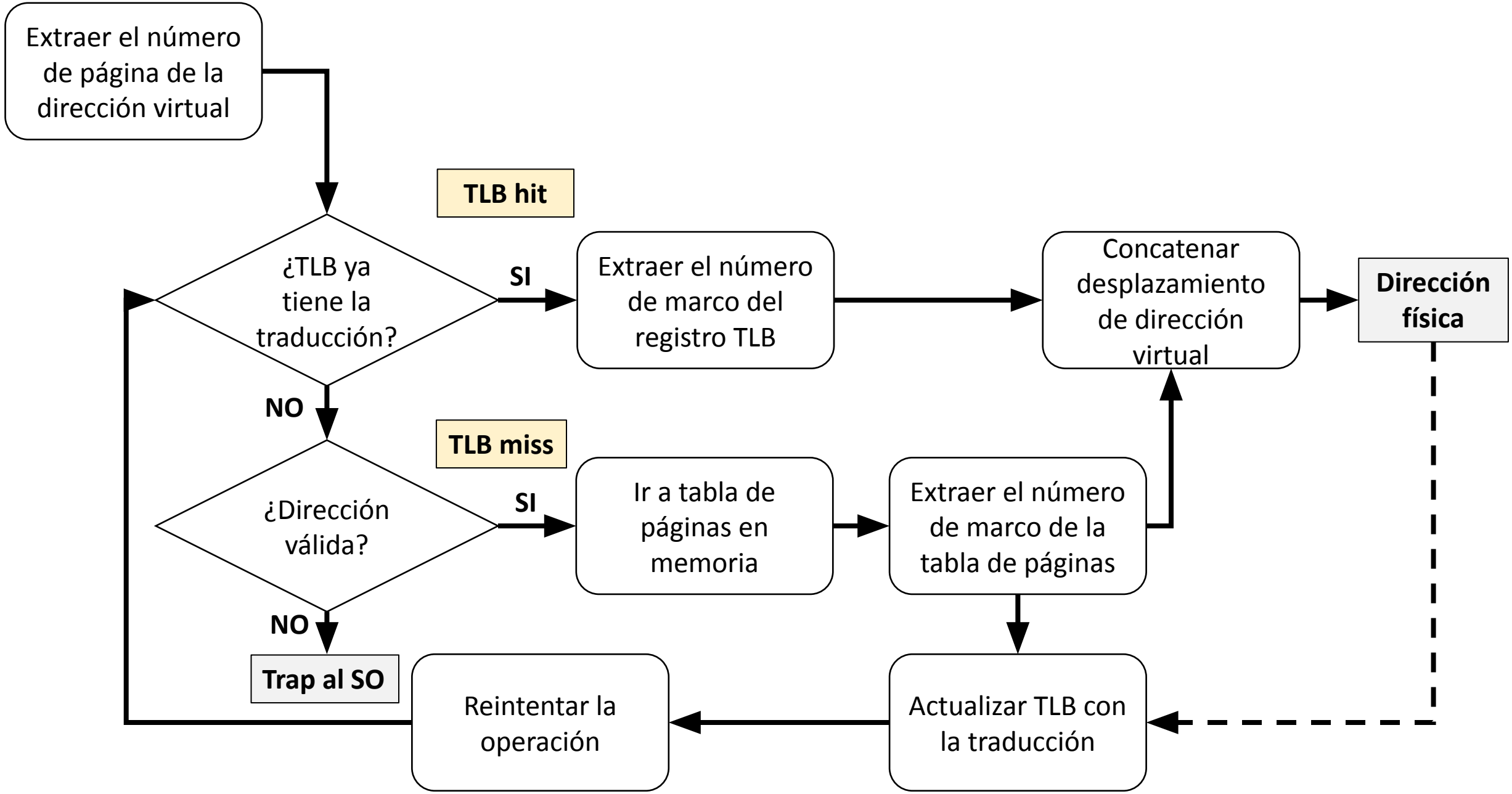

Heap

Librería estándar de C

Stack (gracias Faryd)

Translation-lookaside buffer (TLB)

- La tabla de páginas está implementada en memoria principal
- Cada referencia a memoria necesita una traducción a física
  - Esto implica un acceso adicional a memoria para consultar la tabla de páginas.
- TLB es un mecanismo de la MMU para caché de la traducción virtual a física.
  - Evita ir a consultar a memoria la tabla de páginas.
- Es único para cada proceso
  - En cambios de contexto se debe limpiar TLB o si hay memoria compartida se pueden tener las mismas entradas con un identificador único.


Extraer el número de página de la dirección virtual
TLB hit
¿TLB ya tiene la traducción?
SI
Extraer el número de marco del registro TLB
Concatenar desplazamiento de dirección virtual
Dirección física
NO
TLB miss
¿Dirección válida?
SI
Ir a tabla de páginas en memoria
Extraer el número de marco de la tabla de páginas
NO
Trap al SO
Reintentar la operación
Actualizar TLB con la traducción

Dirección	00	04	08	12	16
0	Página 00				
16	Página 01				
32	Página 02				
48	Página 03				
64	Página 04				
80	Página 05				
96	Página 06		**A[0]**	**A[1]**	**A[2]**
112	Página 07	**A[3]**	**A[4]**	**A[5]**	**A[6]**
128	Página 08	**A[7]**	**A[8]**	**A[9]**	
144	Página 09				
160	Página 10				
176	Página 11				
192	Página 12				
208	Página 13				
224	Página 14				
240	Página 15				

```
int A[10] = {0,1,2,3,4,5,6,7,8,9}
```

- Espacio virtual: $2^8 = 256$ Bytes
- Tamaño de paginas $2^4 = 16$ Bytes
- Dirección de 8 bits: $(8 - 4) + 4 = 8$ Bits

```
int sum = 0;
for (i = 0; i < 10; i++) {
    sum += a[i];
}
```

- `A[0]` está en la dirección virtual **100**
- Hardware extrae número de página desde dirección virtual 100
- Se verifica si **TLB** ya tiene la traducción para la dirección 100
  - Como no la tiene, **TLB miss**
  - Buscar tabla de páginas
- `A[1]` y `A[2]` están en la misma página, **TLB hit**
- `A[3]` produce de nuevo **TLB miss**
- `A[4]`,`A[5]`,`A[6]` Están en la misma página, **TBL hit**

- Tasa de aciertos (**TLB hit**): $7/10 = 70\%$
- Con páginas más grandes se puede mejorar la tasa de aciertos

Tabla de páginas

Página virtual	Marco de página	Otros bits

- Puede verse como un arreglo asociativo
- Otros bits
  - **Válido:** indica si la traducción es válida o no. Espacio no usado entre *heap* y *stack* se marca como no válido.
  - **Permisos:** lectura, escritura, ejecución
  - **Presente:** indica si la página está en memoria o está en disco
  - **Modificada:** indica si la página ha sido modificada desde que se llevó a memoria
  - **Referencia:** indica si la página ha sido referenciada para determinar qué páginas son populares (las no populares podrían ser llevadas a disco)

Referencias

- Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. (2018). Paging: Faster Translations (TLBs). In *Operating Systems. Three Easy Pieces* (pp. 1–16). Arpaci-Dusseau Books.
- Silberschatz, A., Baer Galvin, P., & Gagne, G. (2018). Paging. In *Operating Systems Concepts* (10th ed., pp. 360–363). John Wiley & Sons, Inc.