

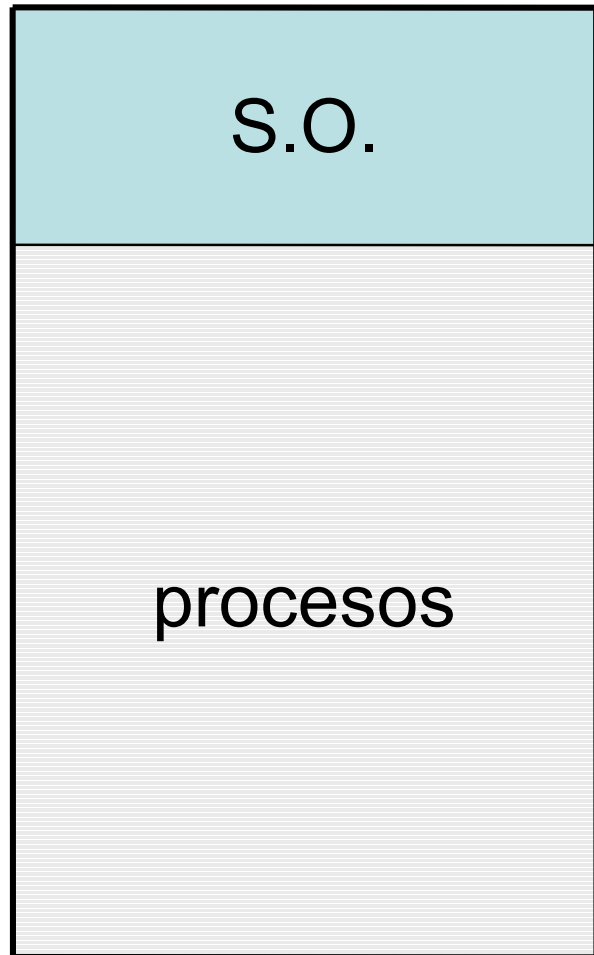
Administración de la memoria

Administración de la Memoria

Funciones:

- administrar el recurso memoria realizando la asignación del mismo a los procesos
- proveer mecanismos que eviten la interferencia entre los procesos en el acceso a su espacio de memoria asignado y en su acceso a la memoria ocupada por el Sistema Operativo.

Adm. memoria



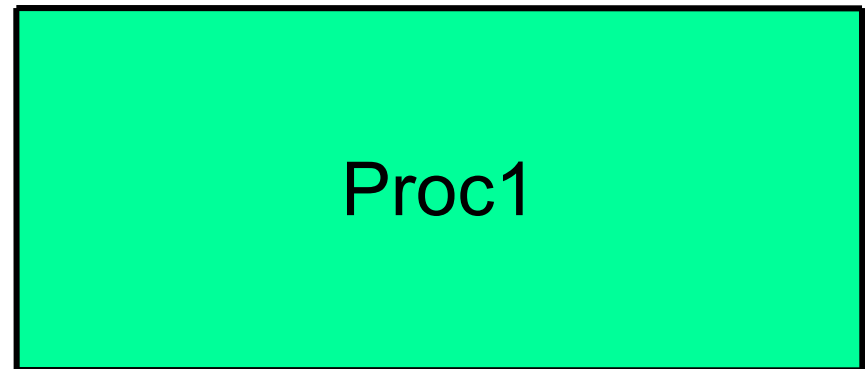
- El Sistema Operativo y los procesos comparten la memoria.
- Es necesario un mecanismo de protección para evitar accesos de los procesos al área del Sistema Operativo

Administraciones de Memoria (tipos)

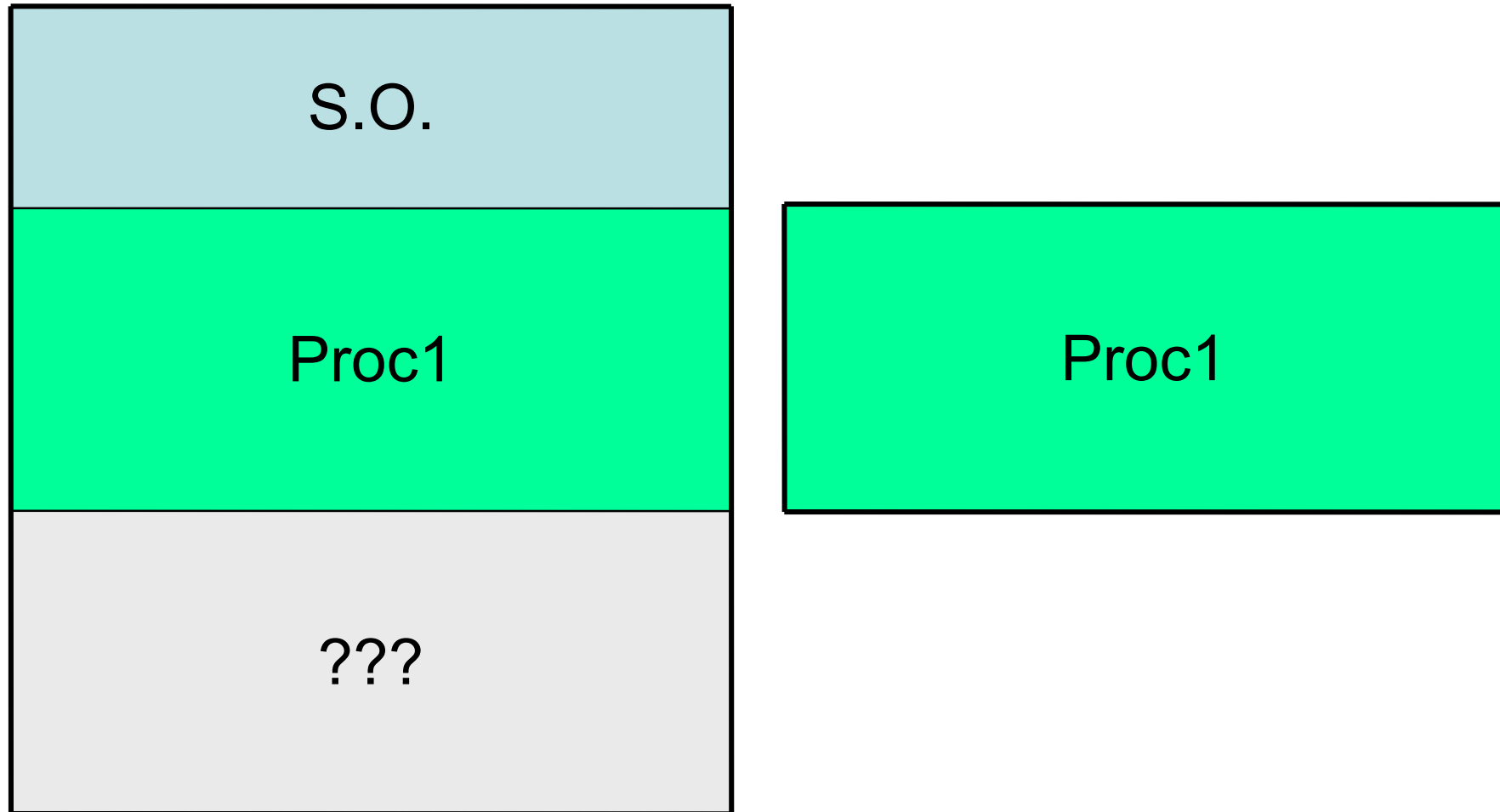
- Simple contigua
- Particionado fijo
- Particionado variable (con/sin compactación)
- Paginación
- Paginación por demanda (con memoria virtual)
- Segmentación
- Segmentación paginada

Simple Contigua

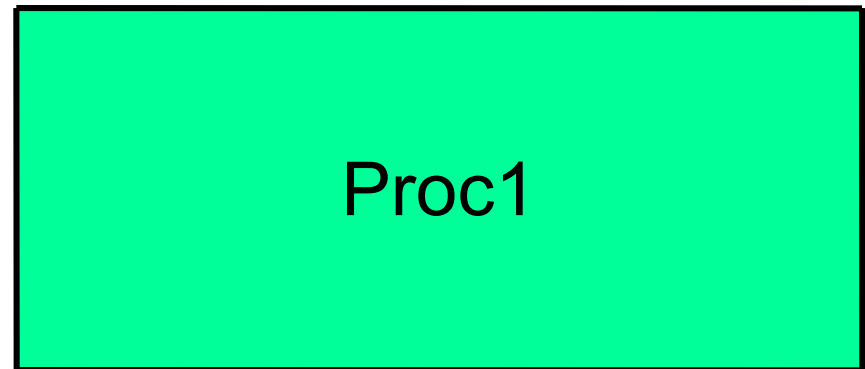
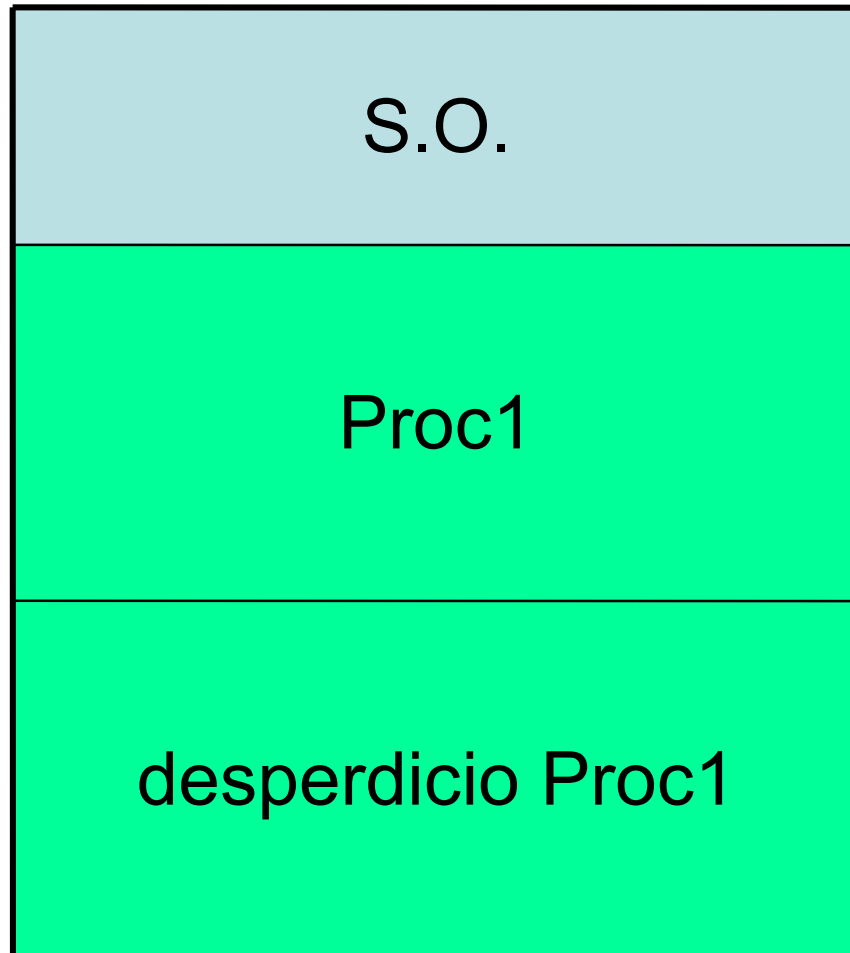
Simple contigua



Simple contiguous (cont.)



Simple contigua (cont.)



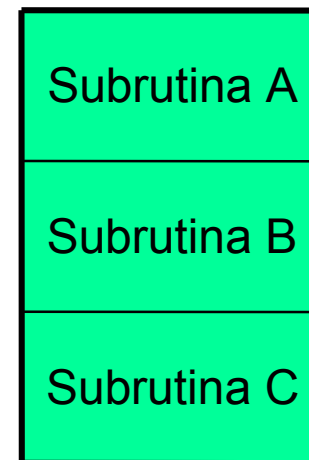
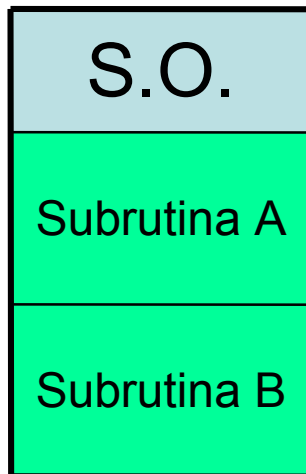
Existe fragmentación externa!!

Simple contigua (cont.)

- Administración trivial
- No permite multiprogramación

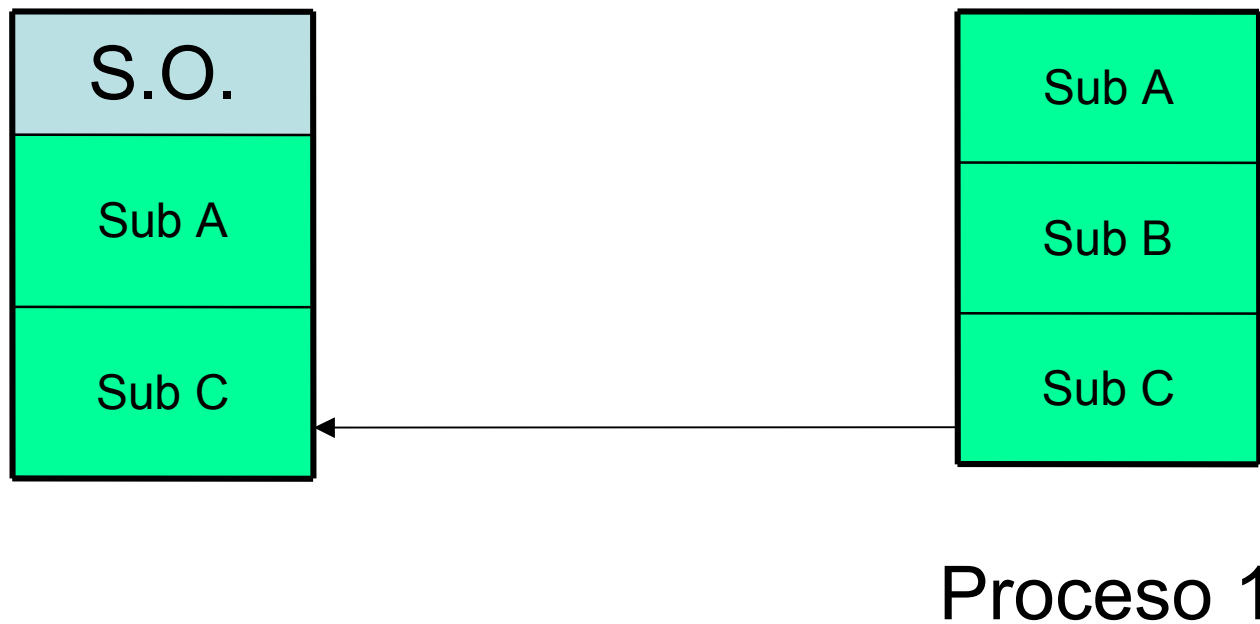
Variantes: Overlay

- Simula más memoria física



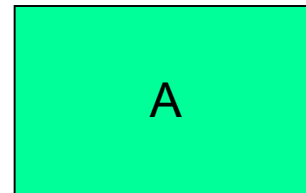
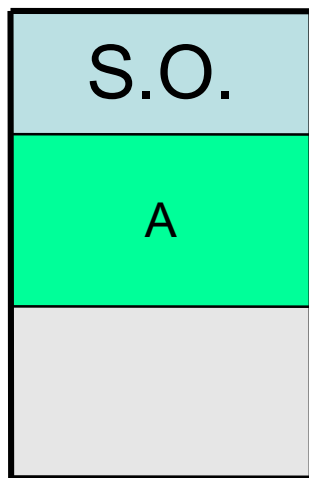
Proceso 1

Overlay (cont.)

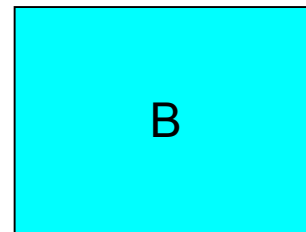


Variantes: Swapping

- Simula multiprogramación



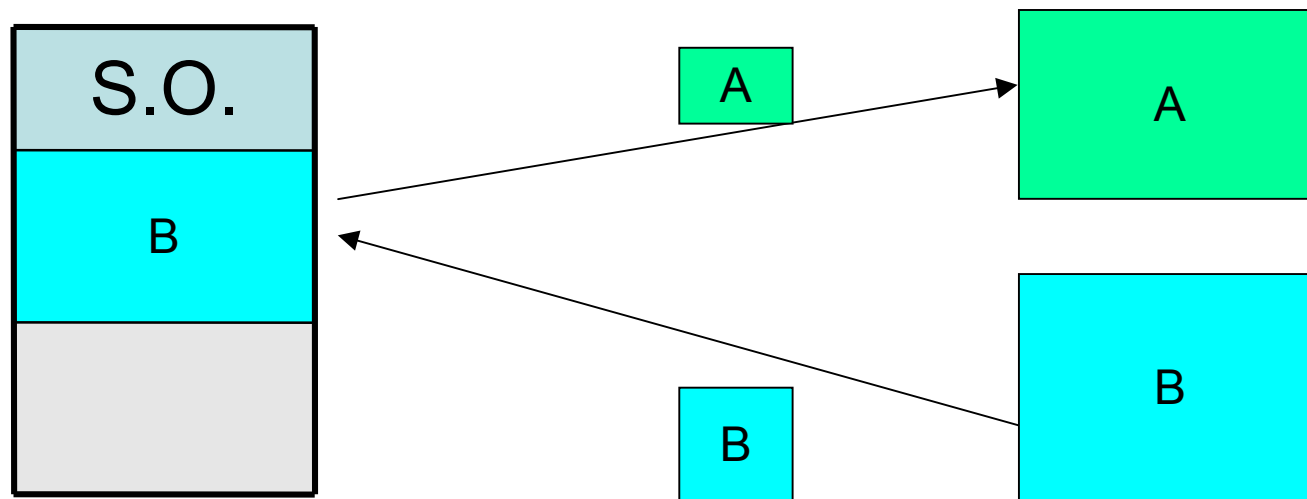
Proceso A



Proceso B

Swapping (cont.)

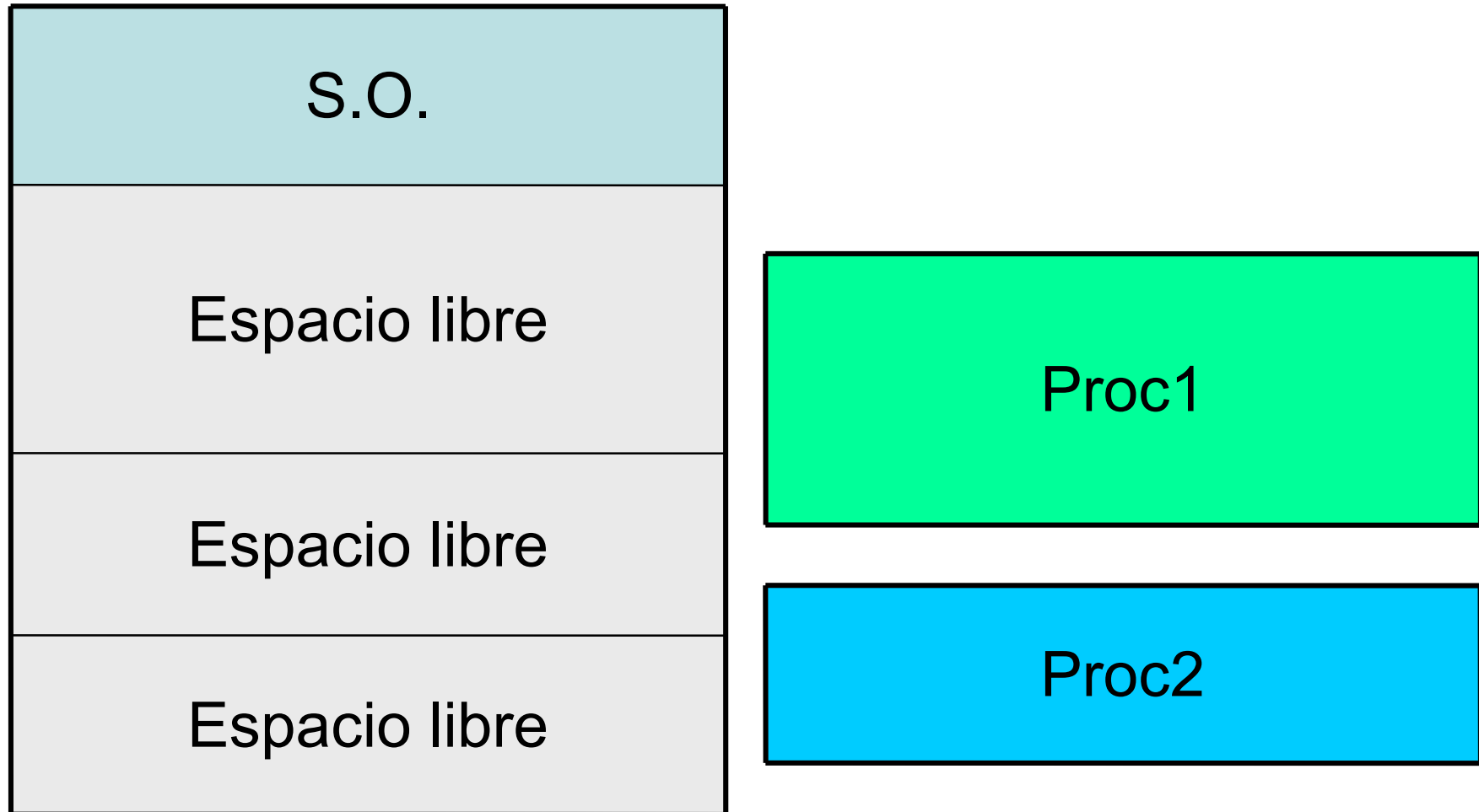
- Simula multiprogramación



- Mejora posible:
 - Particionar la memoria y asignar un proceso distinto a cada partición .

Particionada Fija

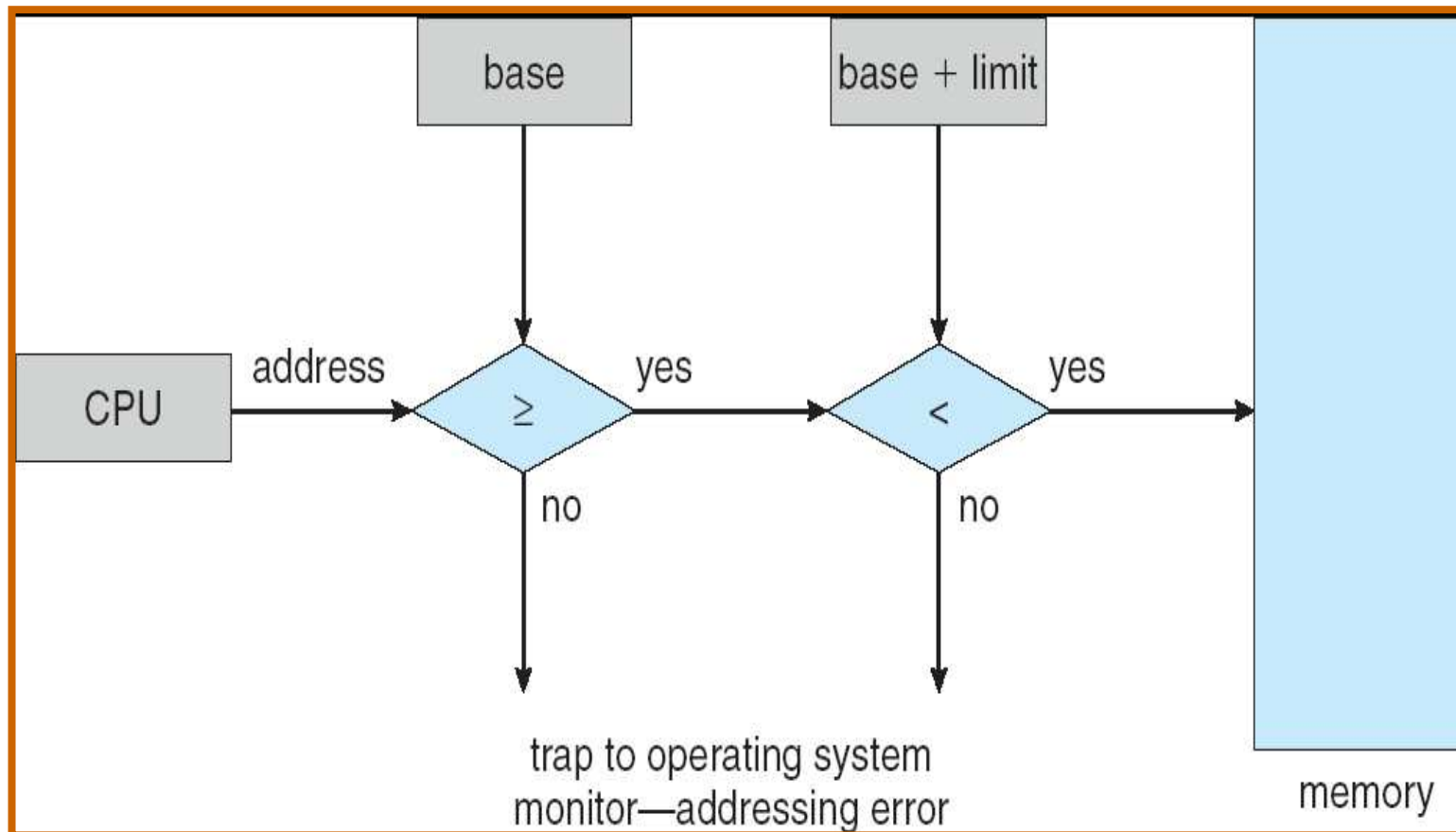
Particionada fija



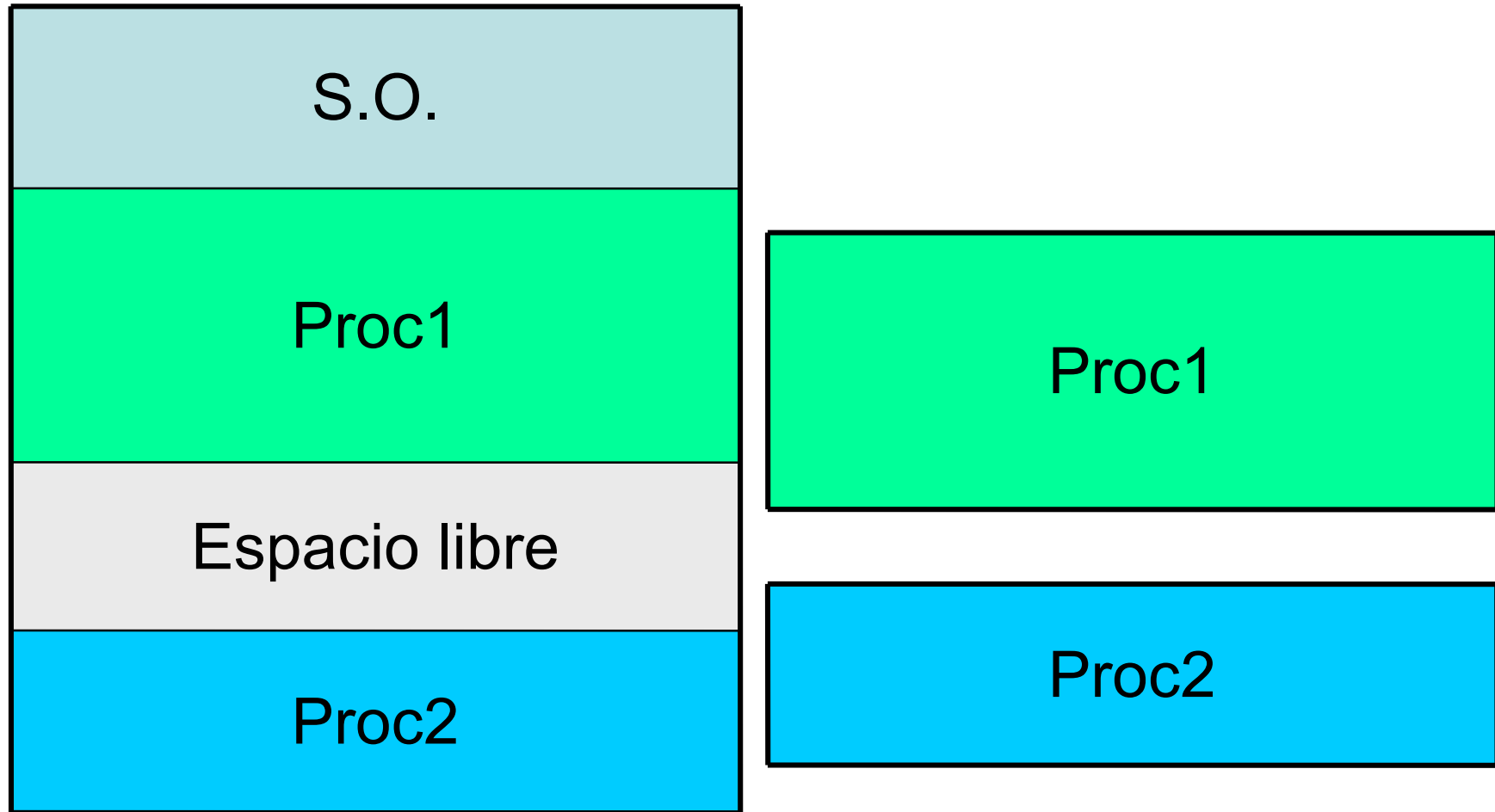
Particionada fija (cont.)

- Necesita guardar en una tabla de particiones:
 - Dirección de comienzo de la partición
 - Tamaño de la partición

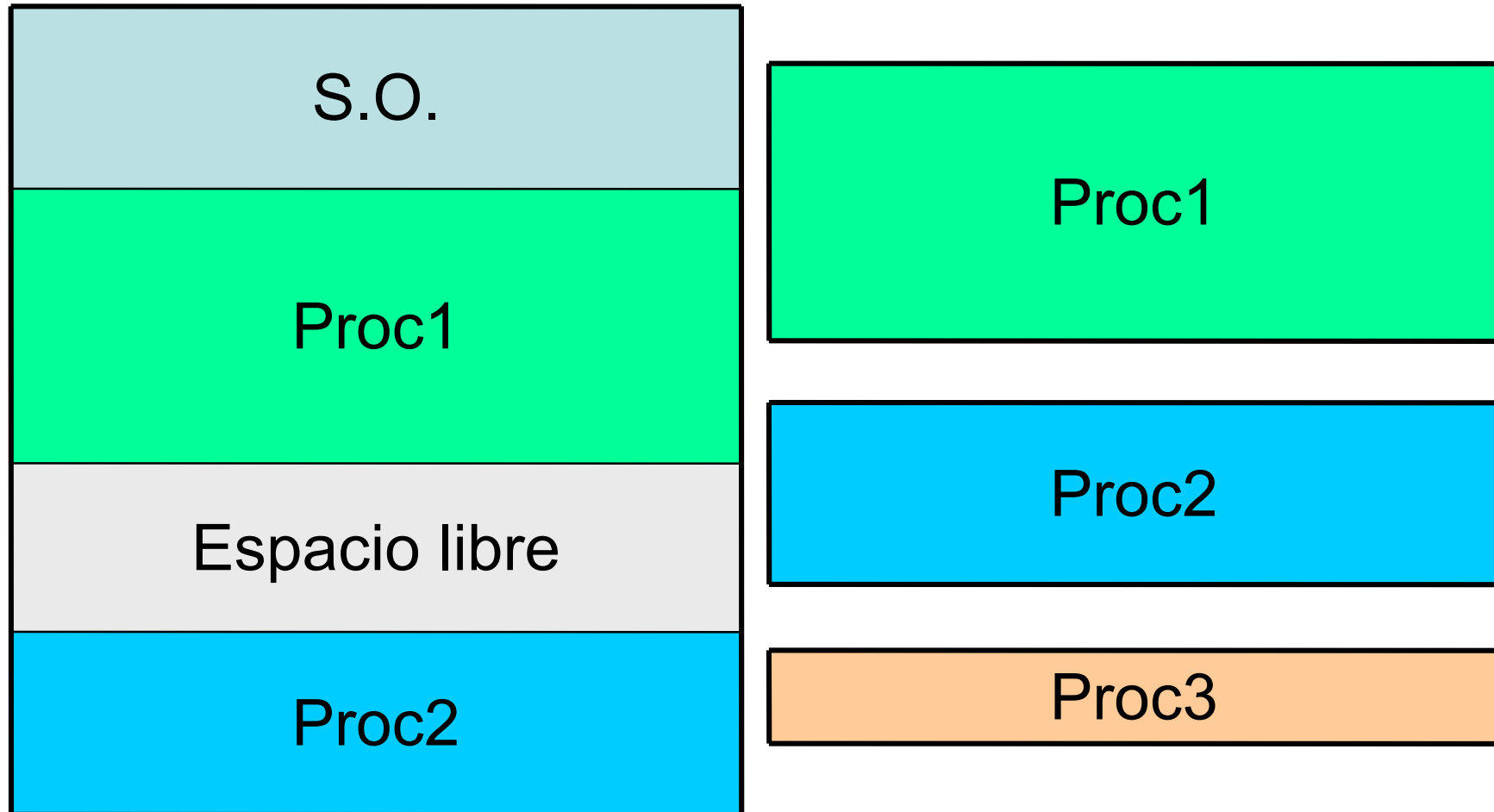
Particionada fija (cont.)



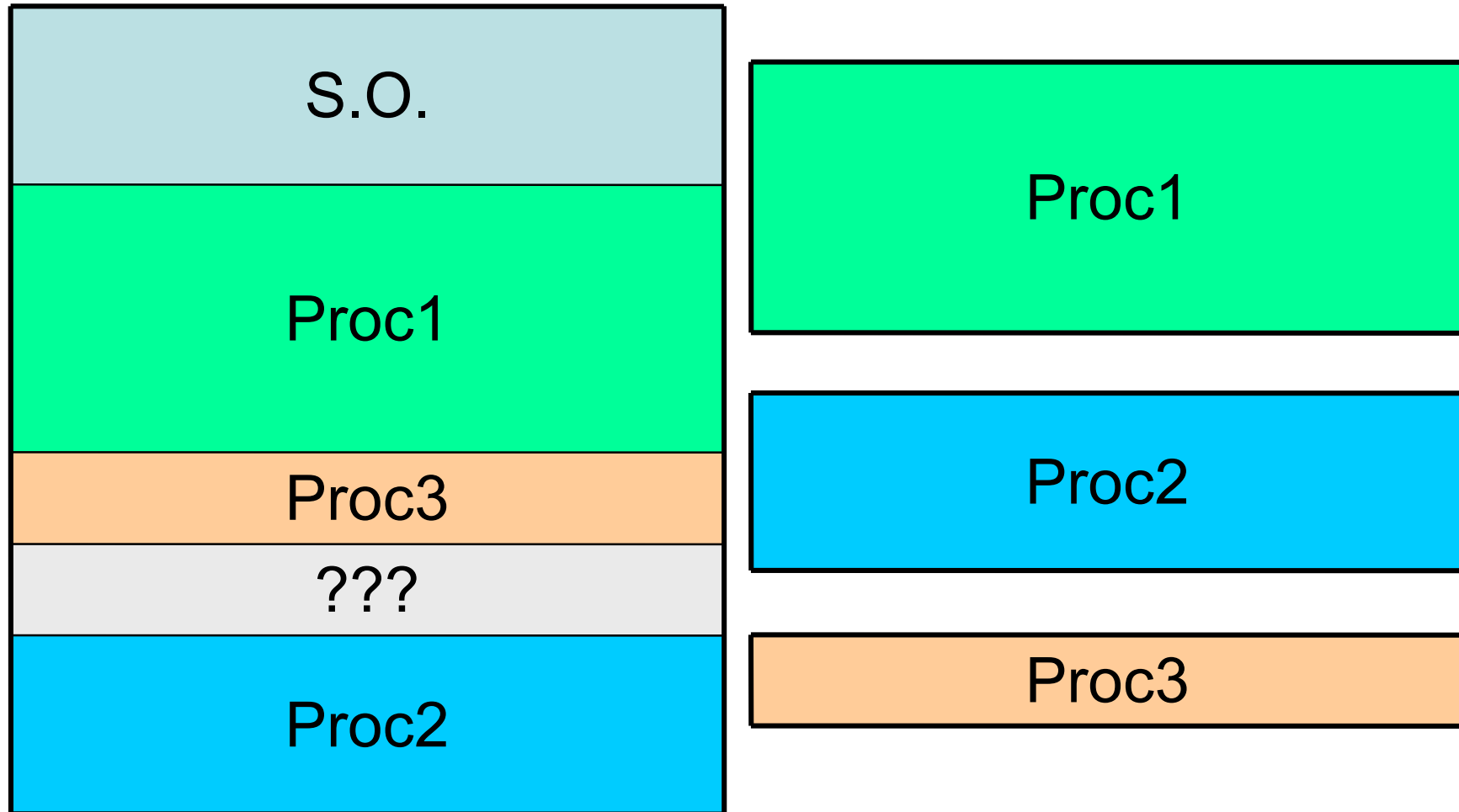
Particionada fija (cont.)



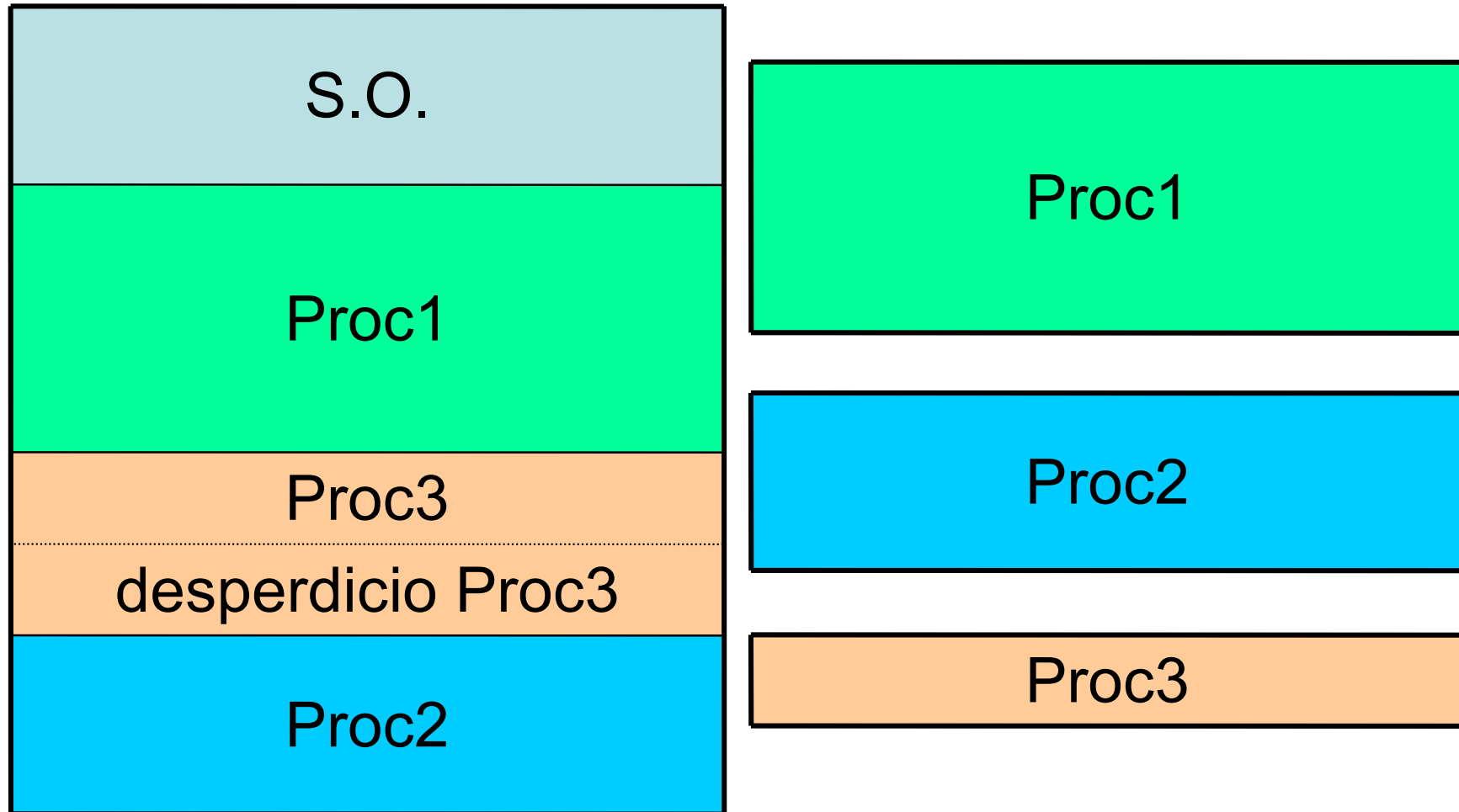
Particionada fija (cont.)



Particionada fija (cont.)



Particionada fija (cont.)



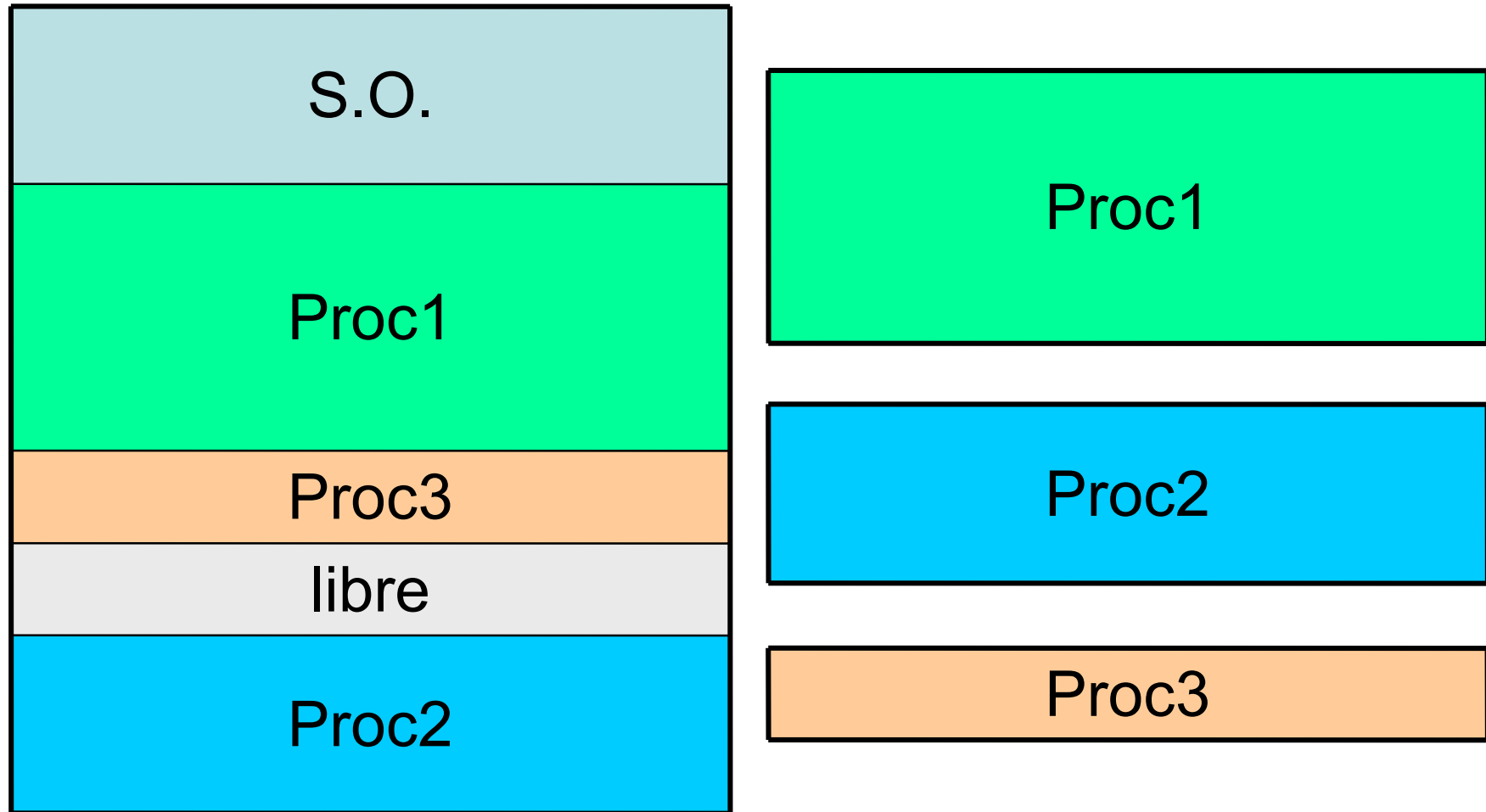
Particionada fija (cont.)



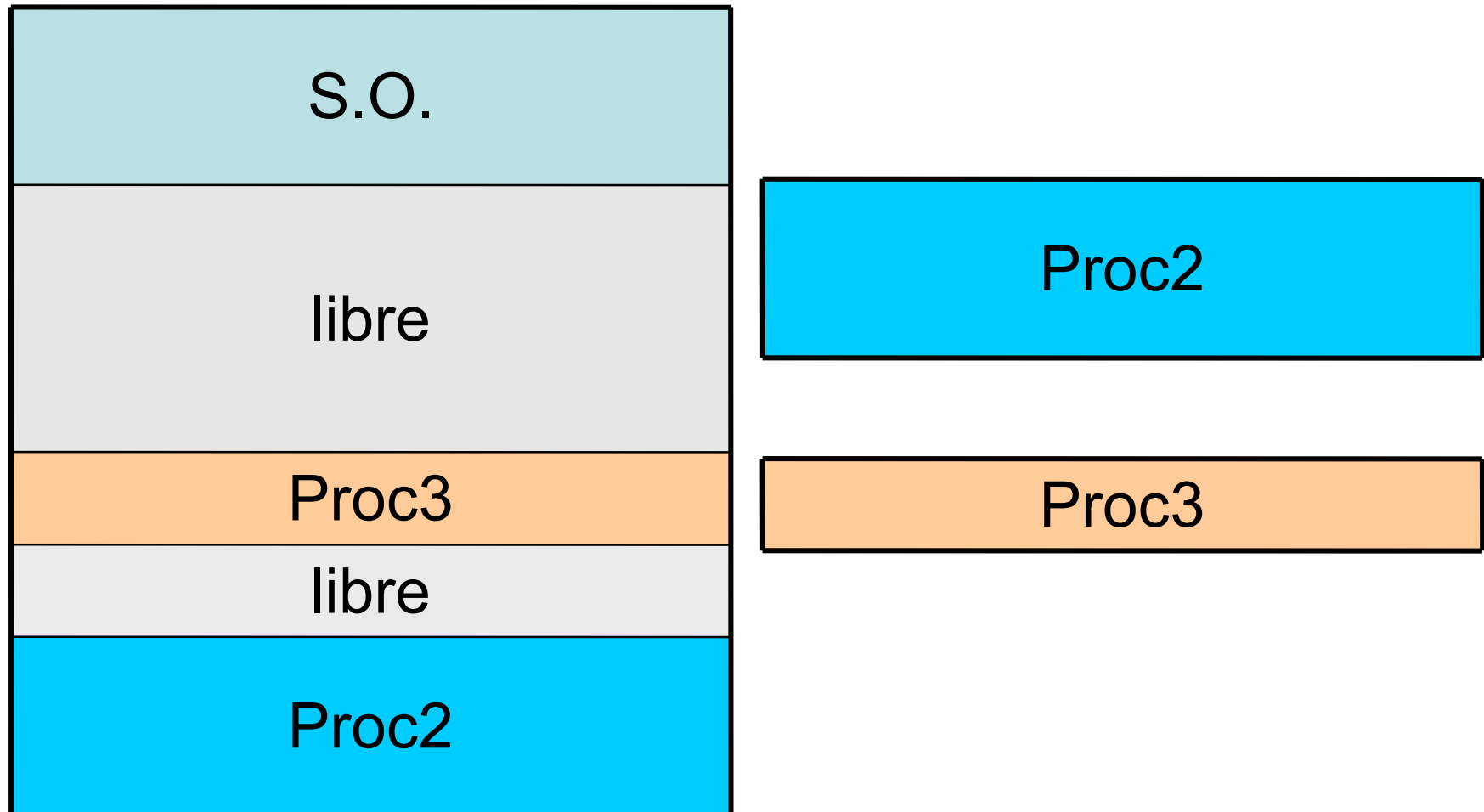
Fragmentación Interna
a la partición

Particionada Variable

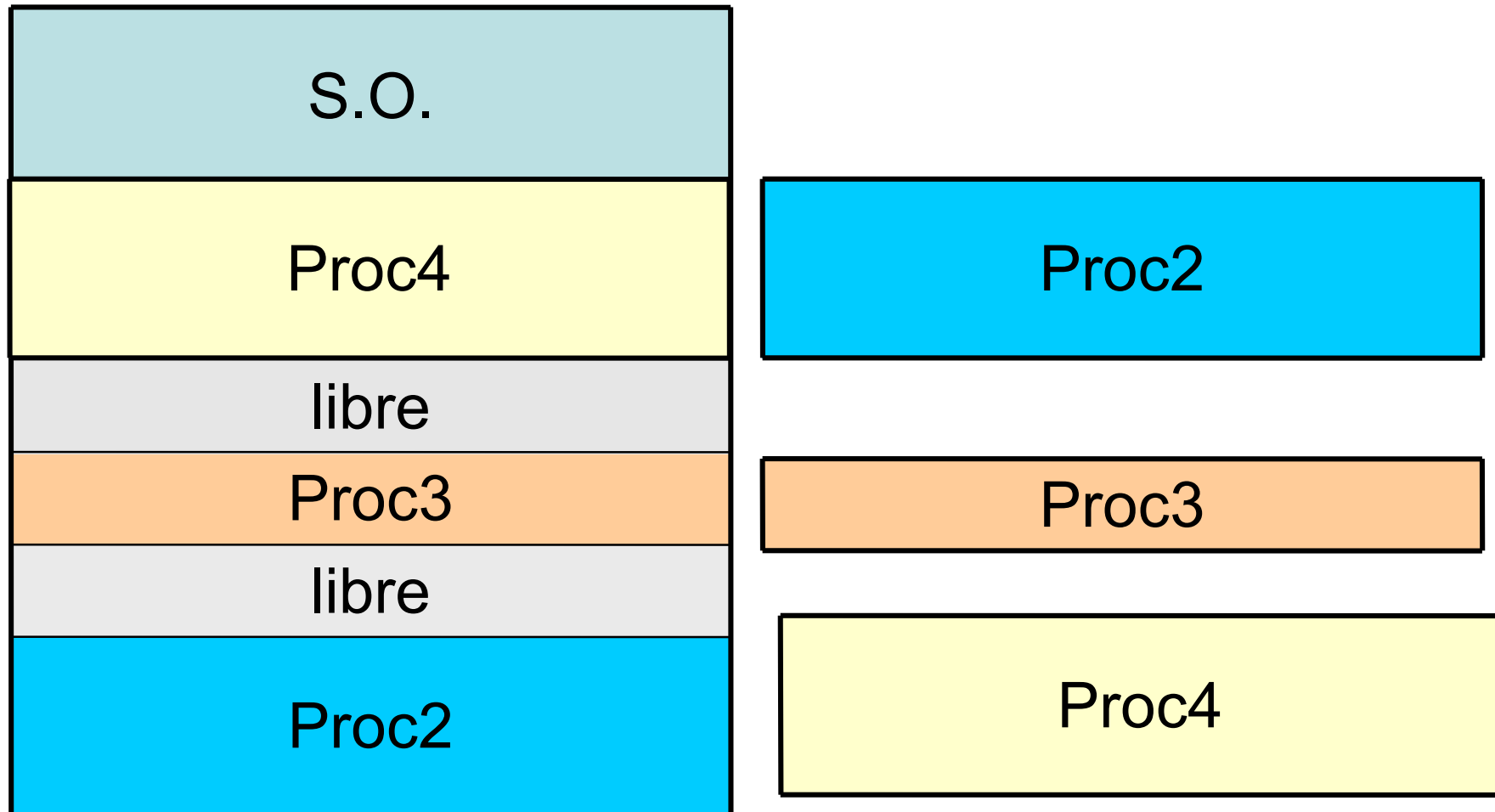
Particionada variable



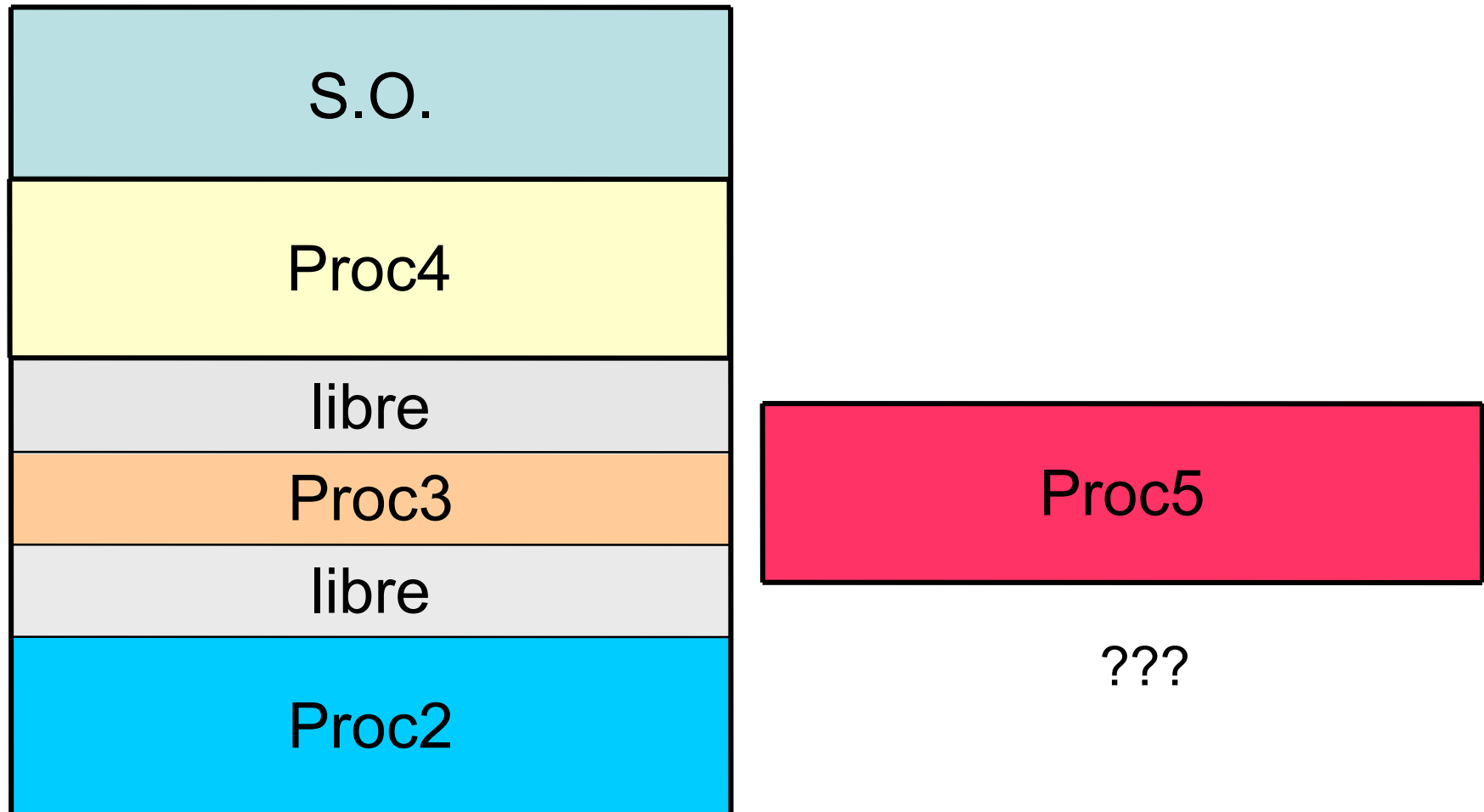
Particionada variable (cont.)



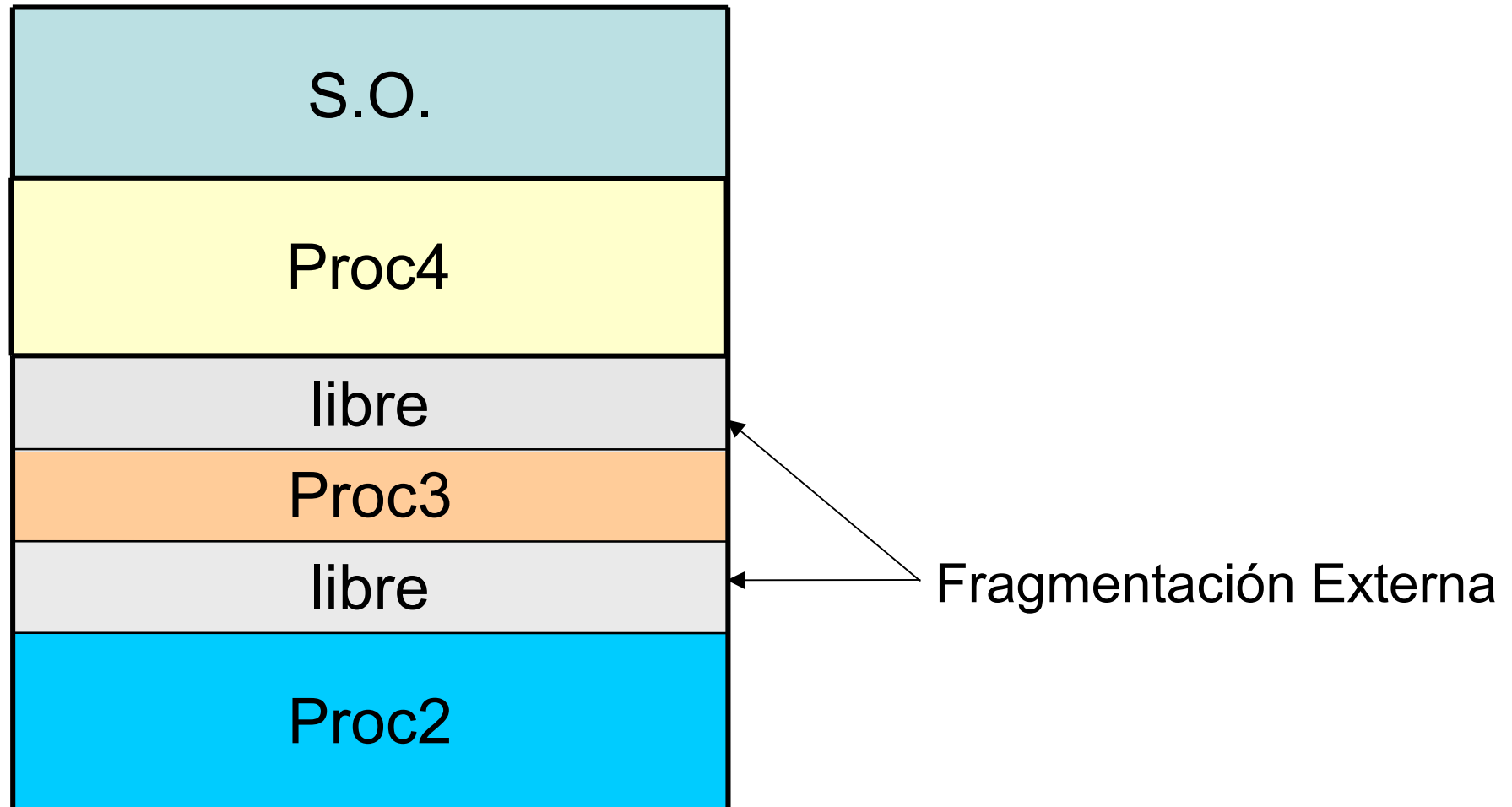
Particionada variable (cont.)



Particionada variable (cont.)



Particionada variable (cont.)



Particionada variable (cont.)

- Una posible solución a la fragmentación externa:
 - Compactación (Memoria-a-Memoria)
- CUIDADO! Es MUY caro:
 - copiar toda la memoria al disco (Memoria-a-Disco-a-Memoria), o
 - ejecutar algoritmos complejos (Mem-a-Mem)

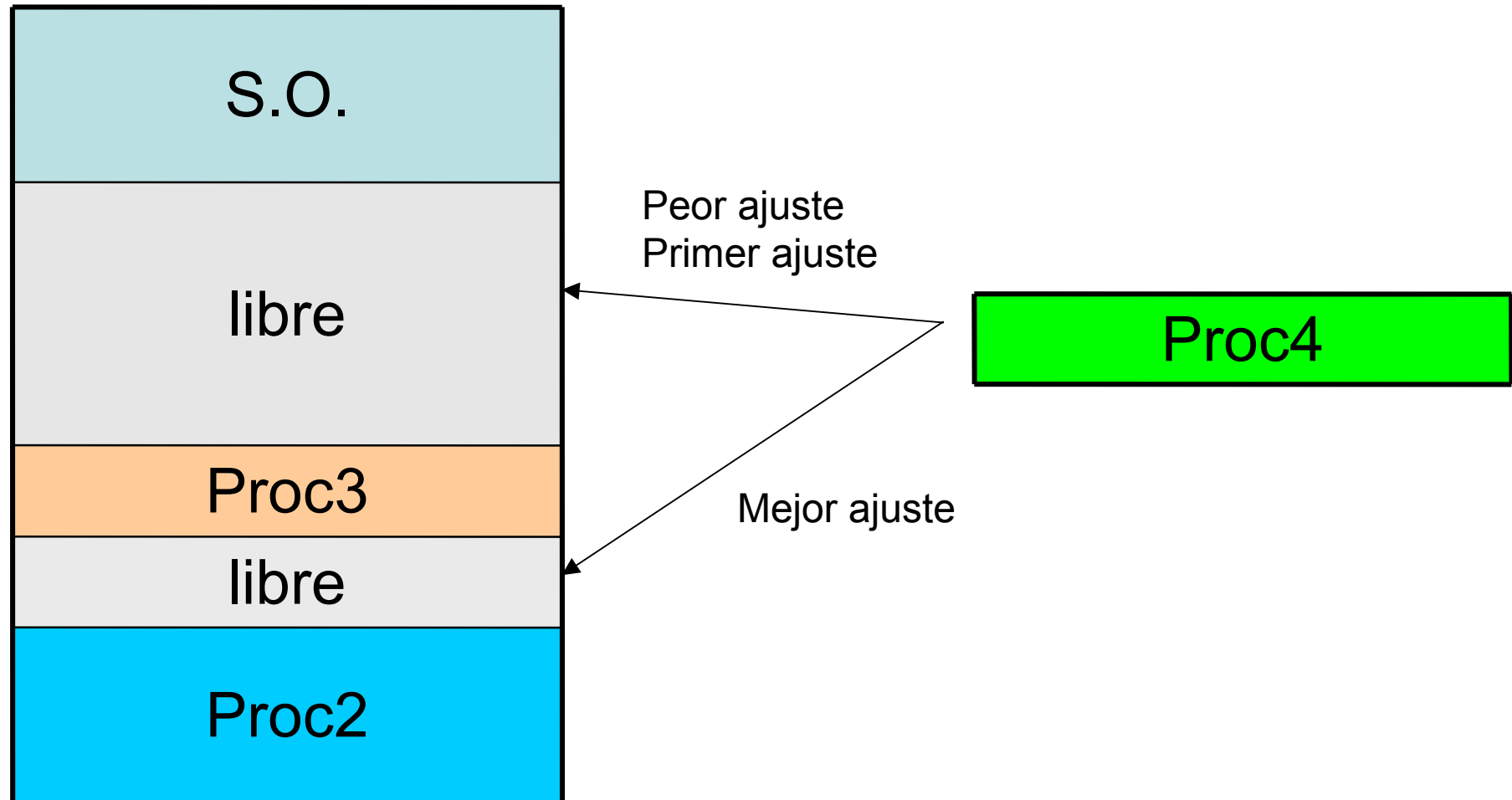
Particionada variable (cont.)

- En la compactación Memoria-Disco-Memoria la CPU queda libre para ejecutar otros procesos mientras el canal de E/S robando ciclos va extrayendo los procesos de la memoria y depositandolo en el disco para luego reingresarlo al sistema en otra ubicación.
- Cuando la compactación se realiza en Memoria la CPU debe estar dedicada a esta tarea no pudiendo ejecutar otros procesos de usuario!!

Particionada Variable (cont.)

- Estrategia de asignación
 - Mejor ajuste
 - Peor ajuste
 - Primer ajuste

Estrategia de asignación



- Mejora posible:
 - Particionar la memoria para poder asignar una parte a cada proceso (ya visto)

Y, además,

- Partir los procesos en unidades más pequeñas

Paginación

Paginación

- Idea:
 - partir la memoria en bloques (frames)
 - partir el proceso en páginas
- Idealmente bloques y páginas son de igual tamaño (lo asumimos para los ejercicios prácticos)
- **Todas** las páginas del proceso se deben cargar en la memoria

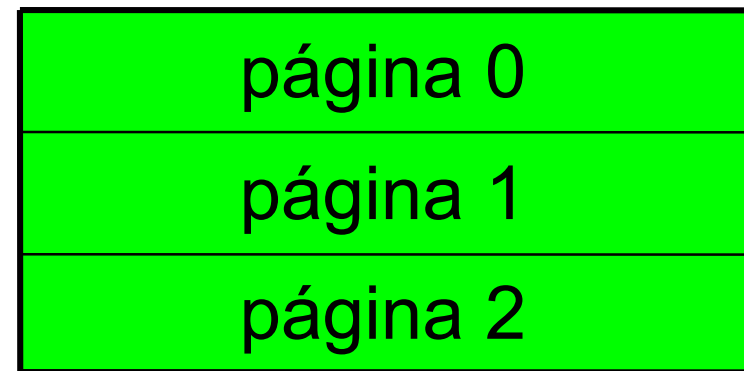
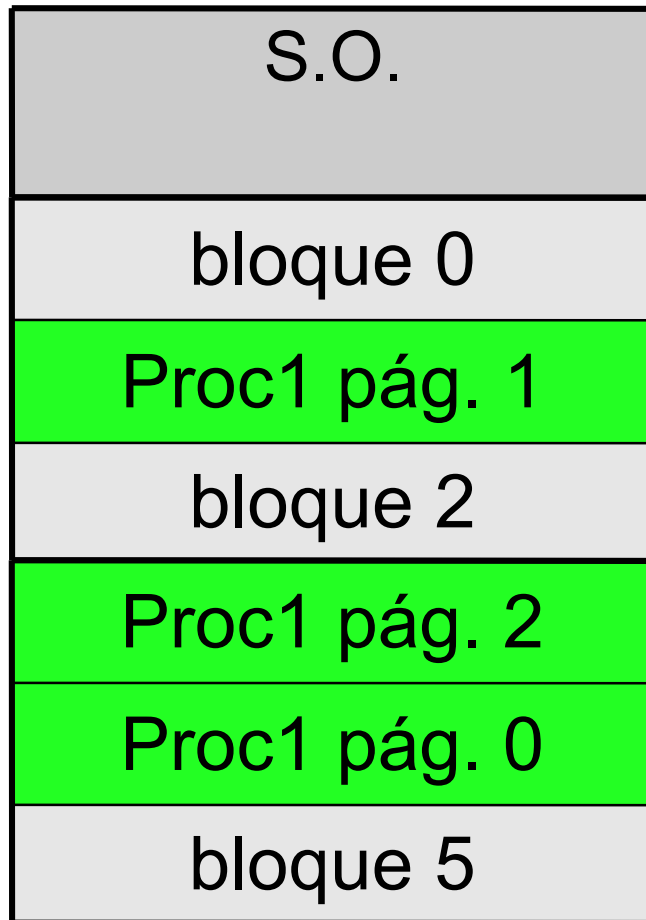
Paginación (cont.)

S.O.
bloque 0
bloque 1
bloque 2
bloque 3
bloque 4
bloque 5

página 0
página 1
página 2

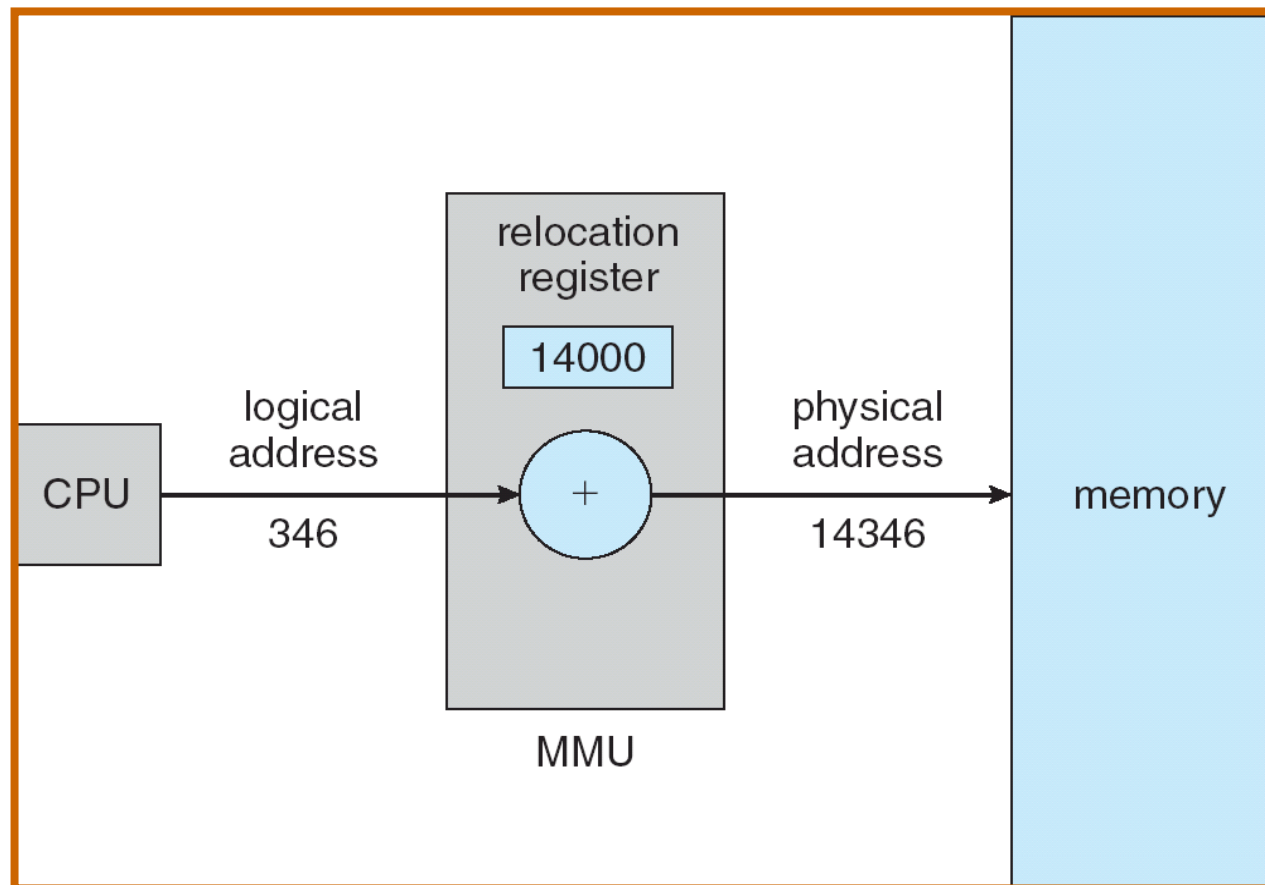
Proc1

Paginación (cont.)



Proc1

Procesos usan direcciones lógicas:
(nunca conocen la dirección real de memoria)



Paginación (cont.)

S.O.
bloque 0
Proc1 pág. 1
bloque 2
Proc1 pág. 2
Proc1 pág. 0
bloque 5

página 0
página 1
página 2

Proc1

página	bloque
0	4
1	1
2	3

Tabla de Distribución de
páginas (TDP)

Paginación (cont.)

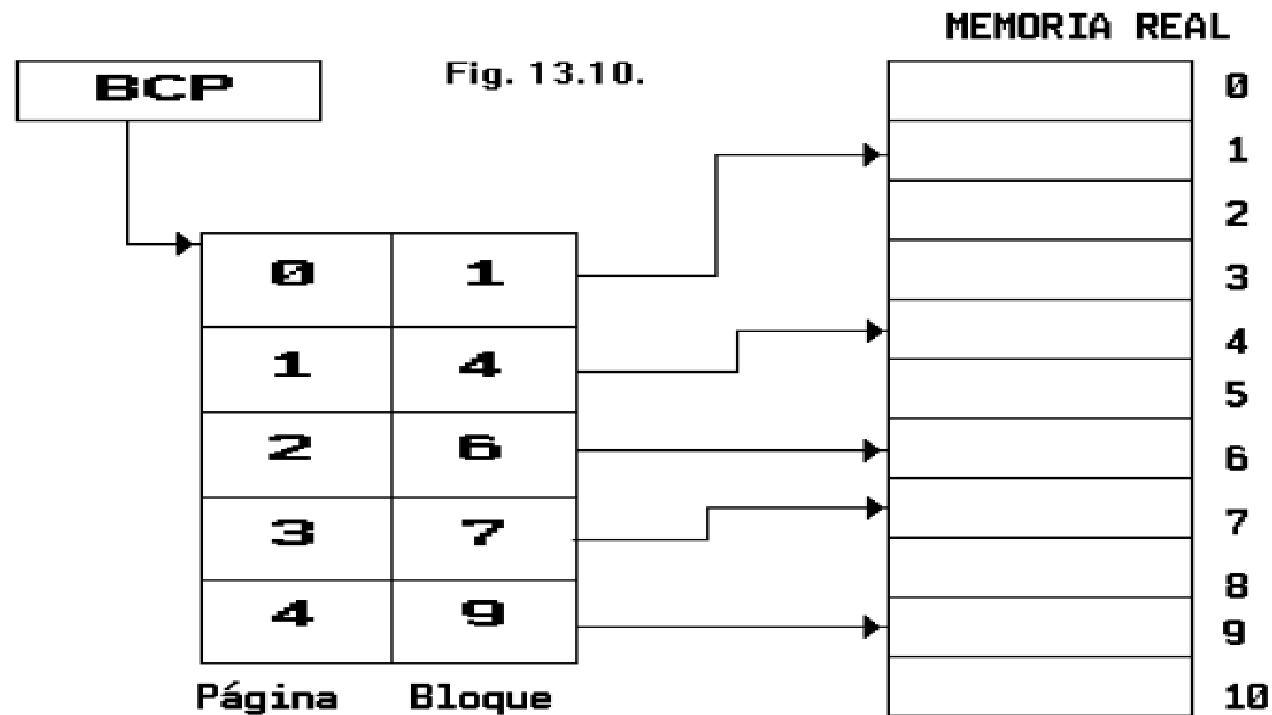
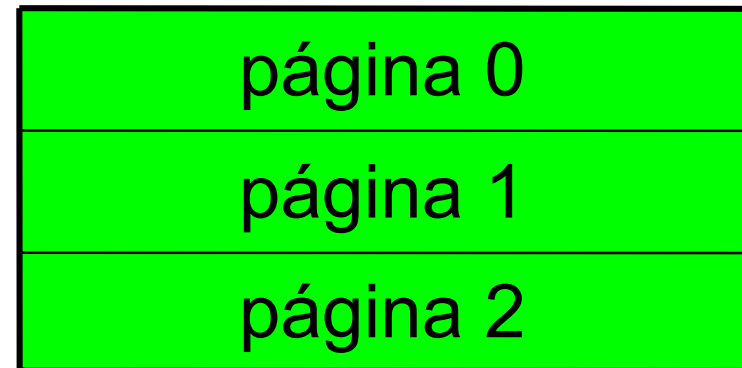
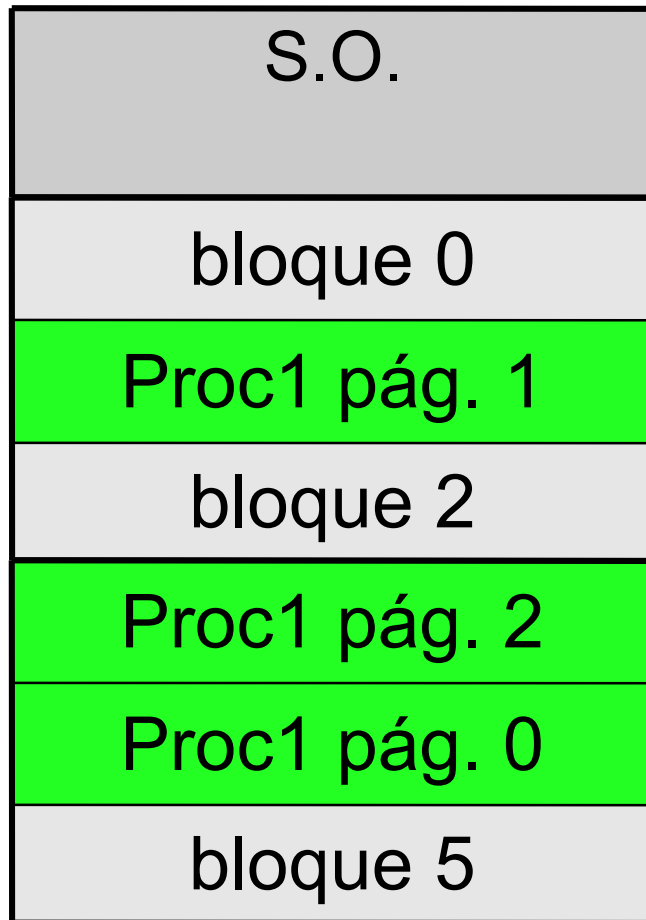


Tabla de distribución de
páginas

Paginación (cont.)



Proc1

Hay fragmentación?

Paginación (cont.)

- Si el tamaño de las páginas y los bloques son iguales, sólo puede haber fragmentación interna al proceso en el último bloque.

Paginación (cont.)

- Direcciones lógicas (virtuales):

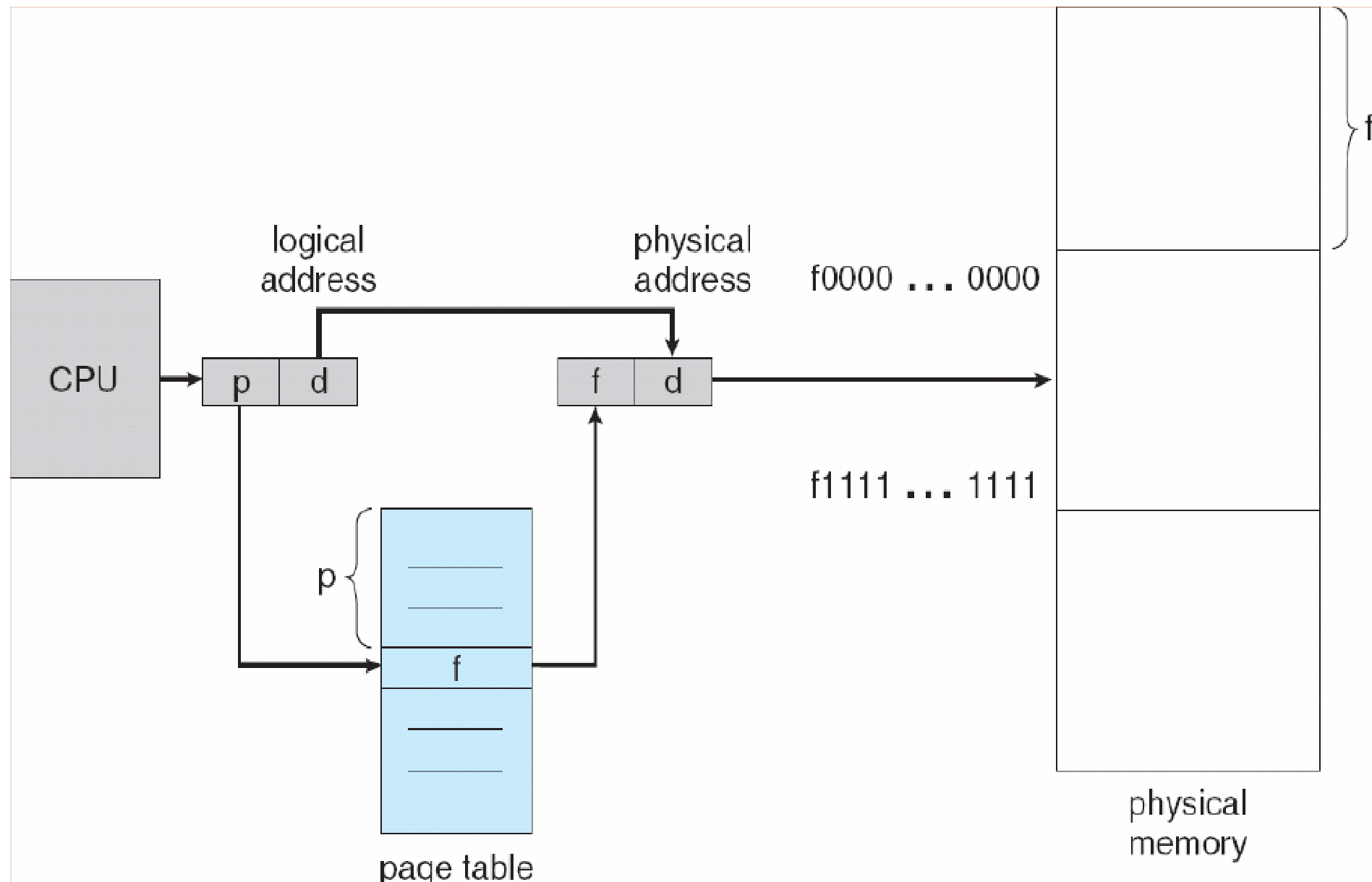
$\underbrace{10 \dots 10}_{\text{página}} \quad \underbrace{011 \dots 1001}_{\text{desplazamiento}}$

El DAT (Direct Address Translator) es el hardware que se encarga de traducir la dirección virtual a dirección en memoria real accediendo a la TDP.

Paginación (cont.)

- La capacidad de direccionamiento del sistema estará dada por el tamaño del bus de direcciones
- En estas administraciones es tarea del compilador partir los programas en las paginas que correspondan según el tamaño requerido por la instalación (tamaño del bloque).

Paginación (cont.)



Paginación (cont.)

Cada acceso a una posición de memoria real implica dos accesos:

- un acceso para ir a la TDP del proceso y ubicar en qué bloque de MR está la página cargada
- un acceso a la dirección de memoria real que se quería acceder

Paginación (cont.)

S.O.
Proc2 pág. 0
Proc1 pág. 1
Proc2 pág. 2
Proc1 pág. 2
Proc1 pág. 0
Proc2 pág. 1

página 0
página 1
página 2

Proc1

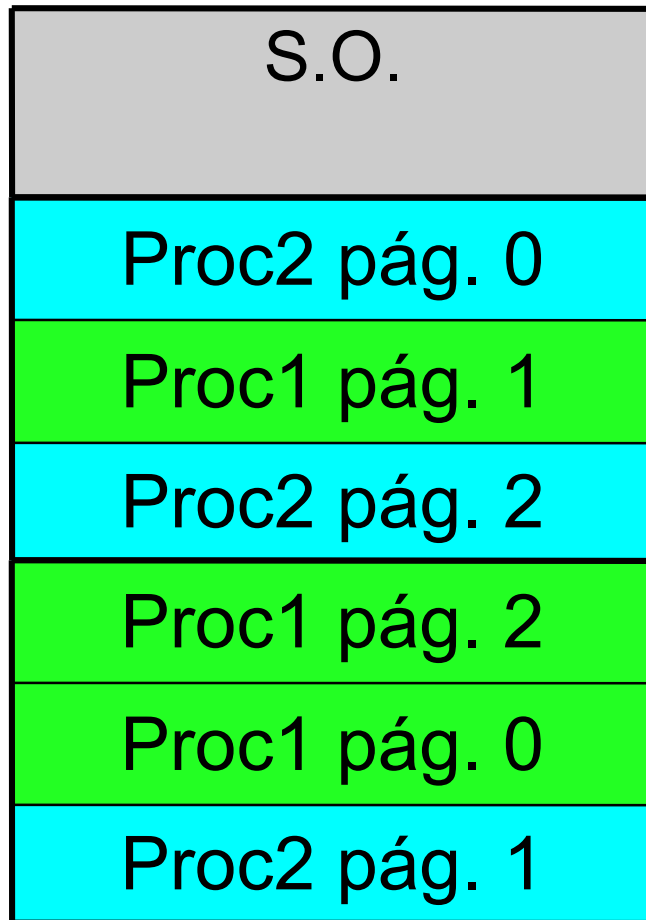
página 0
página 1
página 2

Proc2

página	bloque
0	4
1	1
2	3

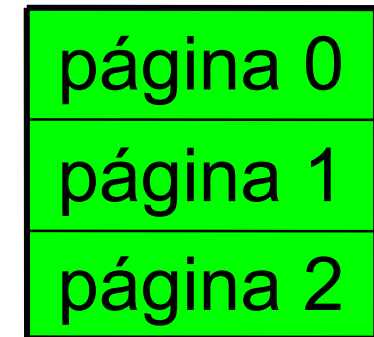
página	bloque
0	0
1	5
2	2

Paginación (cont.)

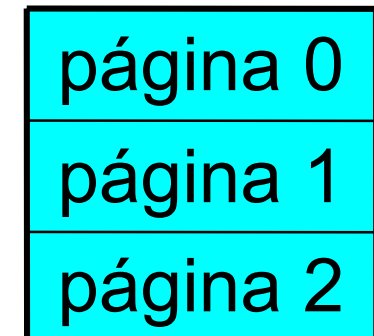


bloque	página	proceso
0	0	2
1	1	1
2	2	2
3	2	1
4	0	1
5	1	2

Tabla de Distribución bloques (TDB). Es el mapa de memoria.



Proc1



Proc2

Paginación (cont.)

- Tabla de páginas (TDP):
 - 1 por proceso
- Tabla de bloques (TDB):
 - 1 para todo el sistema

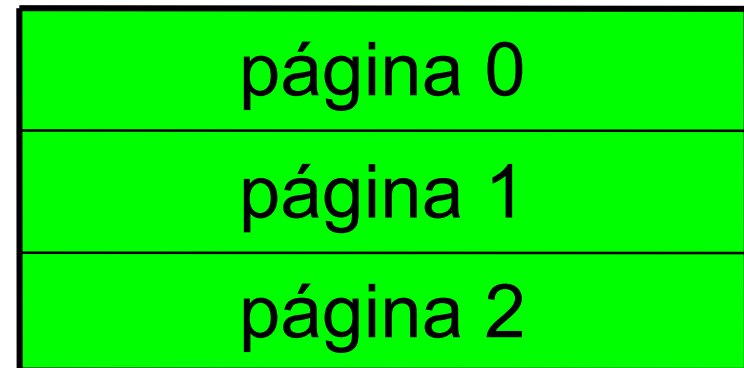
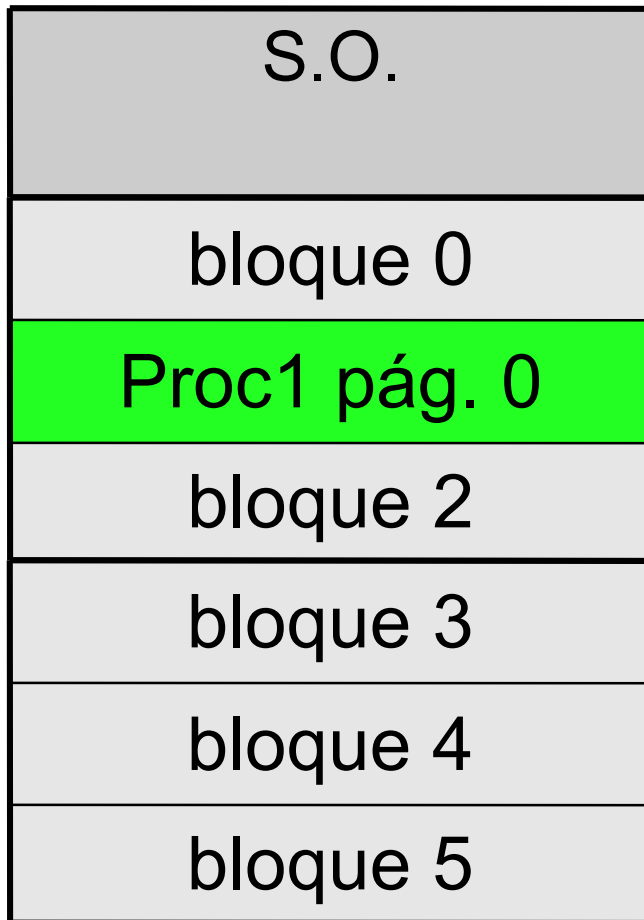
Paginación por demanda o Paginación con memoria virtual

Paginación por demanda

Idea: no todas las páginas se cargan simultáneamente...

No se necesita que estén todas las páginas del proceso cargadas en la memoria, entonces puedo usar ese espacio para páginas de otro proceso (siempre que pueda traer las páginas que necesito).

Paginación por demanda (cont.)



Proc1

Sólo carga la
primera página
(entry point)

Paginación por demanda (cont.)

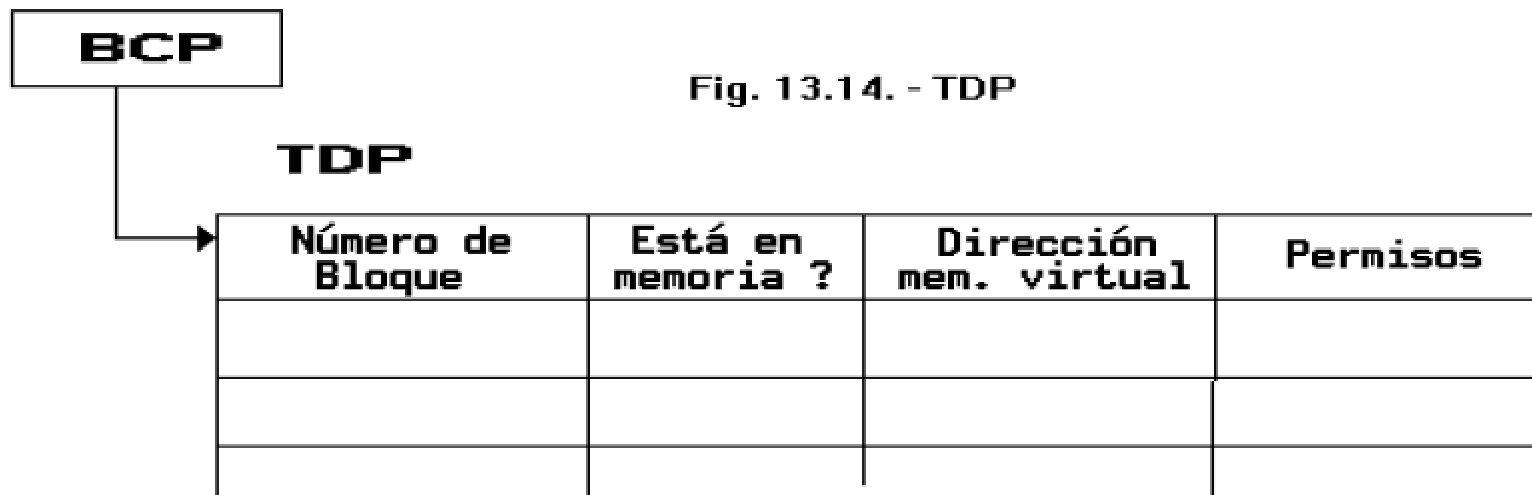
Recordemos la tabla de páginas (TDP)

página	bloque
0	4
1	1
2	3

Es suficiente esta información?

Paginación por demanda (cont.)

Tabla de páginas



El número de página es el índice de la tabla.
El disco en el cual se almacenan las páginas del proceso se suele denominar por extensión Disco de Memoria Virtual.

Paginación por demanda (cont.)

S.O.
Proc 2 pág. 0
Proc3 pág. 0
Proc1 pág. 4
Proc2 pág. 9
Proc3 pág. 2
Proc2 pág. 1

página 0
página 1
página 2

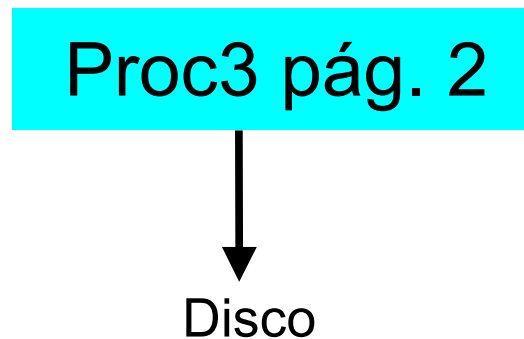
Proc4

Tiene que cargar la primera página pero no hay lugar => remoción

Paginación por demanda (cont.)

S.O.
Proc 2 pág. 0
Proc3 pág. 0
Proc1 pág. 4
Proc2 pág. 9
Proc4 pág. 0
Proc2 pág. 1

La página removida
debe guardarse en
memoria virtual sii
esa página cambió!



Algoritmos de remoción

- FIFO
- LRU (least recently used - la que hace más tiempo que no se referencia)
- LFU (least frequently used - la utilizada menos veces)

Paginación por demanda (cont.)

Recordemos la tabla de distribución de bloques (TDB)

bloque	página	proceso
0	0	2
1	0	3
2	4	1
3	9	2
4	2	3
5	1	2

Es suficiente esta información?

Tabla de distribución de bloques

TDB

Fig. 13.15. - TDB

Id. del programa	Página	Contador	Página cambió ?	Fijo por canal ?	Tránsito?

El número de bloque es el índice de la tabla

Problemas

- Buffer a caballo:
 - Un buffer que cruza el límite de una página a otra
 - Los canales de E/S usan direcciones físicas reales!
- Solución:
 - Se usan varios programas de canal, cada uno opera sobre los datos de una página.
 - Las páginas deben estar fijas en memoria!

Problemas

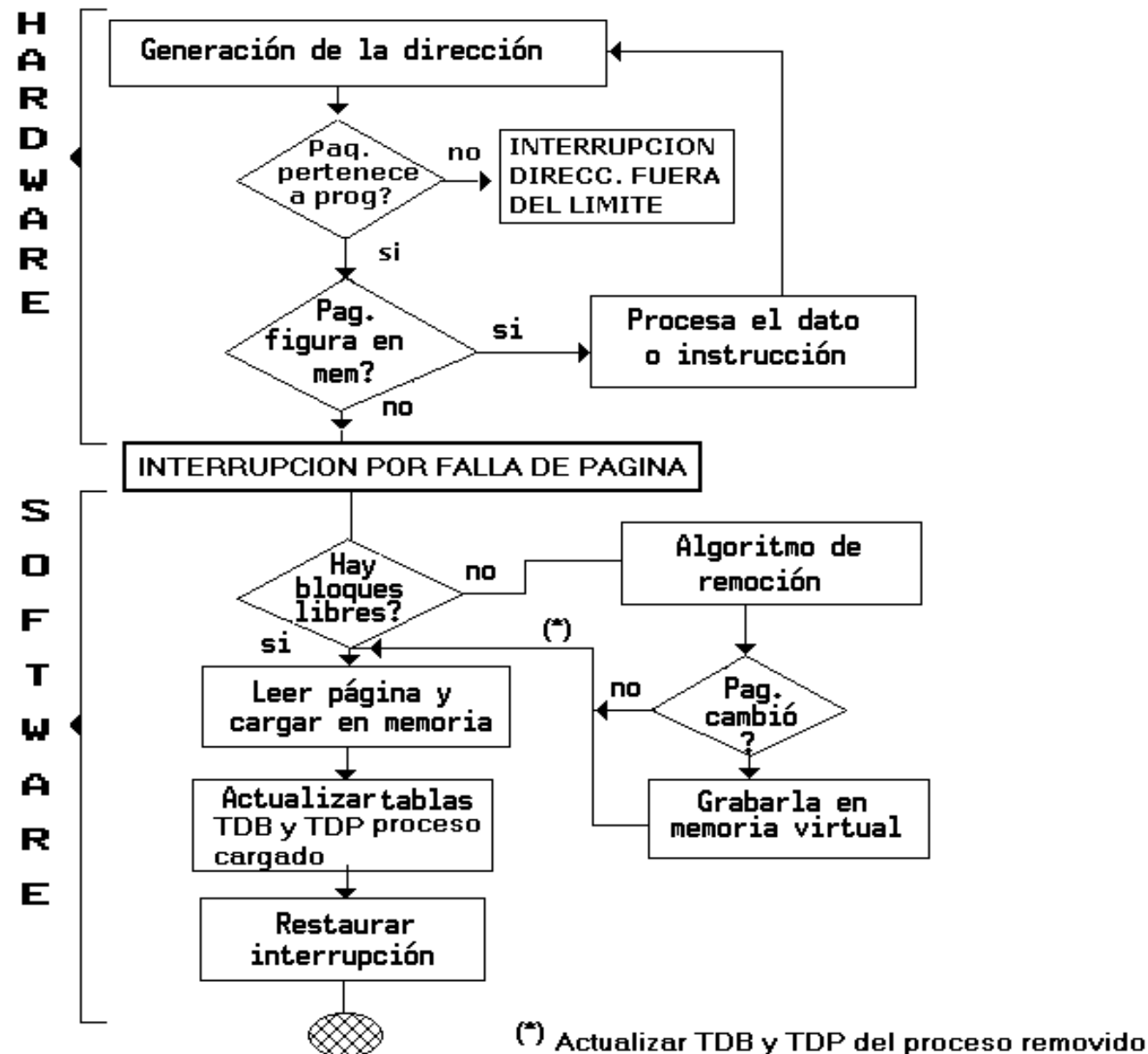
- Thrashing
 - Es el estado en que se encuentra el sistema cuando el S.O. consume más tiempo en gestionar la memoria (cargar/guardar páginas) que en ejecutar instrucciones de los procesos. Alto overhead dedicado a ejecutar las rutinas de paginación.
 - Usualmente provocado por algoritmos malos o una memoria real chica

Qué páginas no pueden removerse??

- páginas atadas a E/S (los canales usan direcciones reales y no virtuales - no usan la TDP-)
- páginas en tránsito (están siendo transferidas desde/hacia memoria virtual)

Existe un “bit de protección de bloque” asociado a cada bloque y que el canal debe proveer para poder grabar en un bloque en memoria real

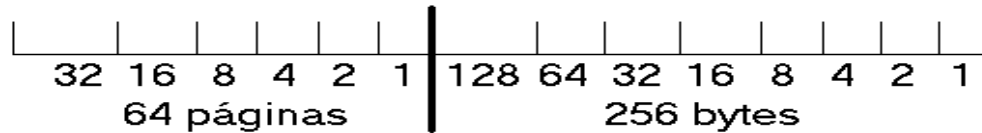
Cómo funciona el algoritmo completo de paginación?



Ejemplo:

- Direccionamiento 14 bits
- Memoria real de 4 K
- Tamaño de página 256 bytes

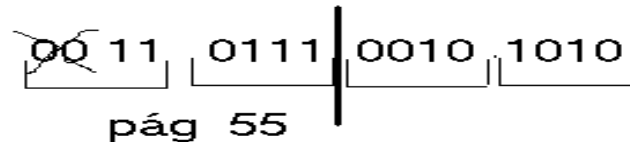
a) Tamaño máximo de un programa a ejecutar en esta instalación??



R: $64 \text{ páginas} * 256 \text{ bytes} = 16 \text{ K}$

b) Un programa genera la dirección virtual X'372A' cuál será la dirección en memoria real?

R:



C) Si la dirección de b) la genera un programa de 10 K de tamaño que ocurre al ejecutarla?

R: $10 \text{ K} \Rightarrow 40 \text{ páginas de } 256 \text{ bytes} \Rightarrow \text{la página } 55 \text{ no existe} \Rightarrow \text{al ejecutarla el programa será cancelado por direccionamiento fuera del límite}$

Traza

Es la sucesión de páginas referenciadas por los procesos.

Si un proceso ejecuta muchas instrucciones dentro de una página sin referenciar a otra, entonces en la traza solamente habrá una referencia para esa página de ese proceso.

En base a una traza dada se puede determinar si es necesario remover una página, cuál será la página elegida para remoción.

Ejemplo de seguimiento de una traza con diferentes algoritmos de remoción.

En todos los ejemplos se supone una capacidad de la memoria real de 4 bloques y en el ultimo renglón se va llevando el control sobre cual es la página candidata a ser removida. En el primer renglón figura la traza analizada que es :

1 – 2 – 3 – 1 – 3 – 1 – 4 – 2 – 1 – 5

LRU : least recently used (la que hace más tiempo que no se referencia))

Mantengo ordenado por el uso en el tiempo. Indice de fracasos 5 / 10.

1	2	3	1	3	1	4	2	1	5	Traza
1	2	3	1	3	1	4	2	1	5	
	1	2	3	1	3	1	4	2	1	
		1	2	2	2	3	1	4	2	
						2	3	3	4	Candidata
X	X	X				X			X	Fracasos

LFU : least frequently used (la que fue menos usada en veces)

Ordeno por cantidad de referencias a la página (si no indica nada la cantidad es 1). A igual cantidad de referencias tomo la página más antigua para salir. Indice de fracasos 5 / 10.

1	2	3	1	3	1	4	2	1	5
1	2	3	1(2)	3(2)	1(3)	1(3)	1(3)	1(4)	1(4)
	1	2	3	1(2)	3(2)	3(2)	2(2)	2(2)	2(2)
		1	2	2	2	4	3(2)	3(2)	3(2)
						2	4	4	5

FIFO : first in first out. Indice de fracasos 5 / 10.

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

Anomalía de Belady

Se puede pensar que incrementar la memoria indefinidamente otorgará el mejor índice de hallazgos pero esto no es cierto en los algoritmos de remoción FIFO.

Belady demostró que incrementarla no sólo no mejora sino que empeora el índice de hallazgos en cierto casos concretos.

Ejemplo de seguimiento de una traza que muestra la anomalía de Belady

Traza: 1 – 2 – 3 – 4 - 1 – 2 – 5 – 1 – 2 – 3 – 4 - 5

FIFO con M = 3

	1	2	3	4	1	2	5	1	2	3	4	5
	1	2	3	4	1	2	5	5	5	3	4	4
		1	2	3	4	1	2	2	2	5	3	3
Candidata			1	2	3	4	1	1	1	2	5	5
Fracasos	X	X	X	X	X	X	X			X	X	

Indice de fracasos = 9 / 12

FIFO con M = 4

	1	2	3	4	1	2	5	1	2	3	4	5
	1	2	3	4	4	4	5	1	2	3	4	5
		1	2	3	3	3	4	5	1	2	3	4
			1	2	2	2	3	4	5	1	2	3
Candidata				1	1	1	2	3	4	5	1	2
Fr.	X	X	X	X			X	X	X	X	X	X

Indice de fracasos = 10 / 12

Anomalía de Belady con LRU

LRU con M = 3

T	1	2	3	4	1	2	5	1	2	3	4	5
	1	2	3	4	1	2	5	1	2	3	4	5
		1	2	3	4	1	2	5	1	2	3	4
Candidata			1	2	3	4	1	2	5	1	2	3
Fracasos	X	X	X	X	X	X	X			X	X	X

Indice de fracasos = 10 / 12

LRU con M = 4

T	1	2	3	4	1	2	5	1	2	3	4	5
	1	2	3	4	1	2	5	1	2	3	4	5
		1	2	3	4	1	2	5	1	2	3	4
			1	2	3	4	1	2	5	1	2	3
Candidata				1	2	3	4	4	4	5	1	2
Fr.	X	X	X	X			X			X	X	X

Indice de fracasos = 8 / 12

Anomalía de Belady con LRU

T	1	2	3	4	1	2	5	1	2	3	4	5
	1	2	3	4	1	2	5	1	2	3	4	5
M=3		1	2	3	4	1	2	5	1	2	3	4
			1	2	3	4	1	2	5	1	2	3
=====												
M=4				1	2	3	4	4	4	5	1	2

Esto se da en todo **algoritmo de stack**, o sea que cumplen con (*), siendo $m = \#$ de frames y r la traza

$$(*) \quad M(m, r) \subseteq M(m + 1, r)$$

Predicción de Tasas de Page Fault

LRU con M = 4

T	1	2	3	4	1	2	5	1	2	3	4	5
	1	2	3	4	1	2	5	1	2	3	4	5
		1	2	3	4	1	2	5	1	2	3	4
			1	2	3	4	1	2	5	1	2	3
Candidata				1	2	3	4	4	4	5	1	2
=====												
							3	3	3	4	5	1
=====												
Fr.	X	X	X	X			X			X	X	X
=====												
Dist.	i	i	i	i	4	4	i	3	3	5	5	5

$F_m = \sum C_k + C_i$ desde $k = m+1$ hasta n (siendo n la cantidad total de páginas virtuales del proceso)

Tamaño de Página

- Sea s el tamaño de un proceso en bytes
- Sea p el tamaño de una página en bytes
- Sea e el tamaño de cada entrada de la TDP en bytes
- Cantidad de páginas de un proceso es s/p
- Tamaño de la TDP = $s/p * e = s * e / p$
- Promedio de fragmentación interna = $p/2$
- Overhead de memoria generado por un proceso es $= s * e / p + p/2$

Tamaño de Página (cont.)

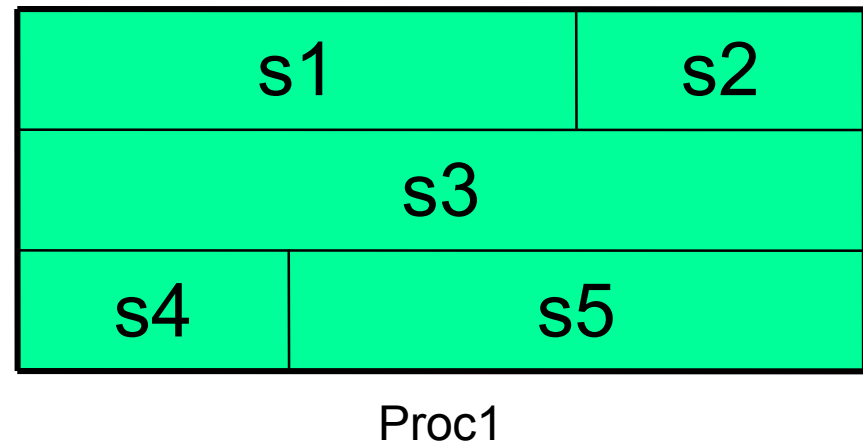
- Overhead de memoria generado por un proceso es $= s \cdot e / p + p / 2 = 0$ (ideal)
- Optimo, se deriva la anterior con respecto a p
- $-s \cdot e / p^2 + 1/2 = 0$ despejando p obtenemos
- $p = (2 \cdot s \cdot e)^{1/2}$ (o sea raíz cuadrada)
- Suponiendo $s = 128$ KB, $e = 8$ bytes, se obtiene un tamaño de página de 1448 bytes
- (Actuales aprox. 8KB)

Segmentación

Segmentación

- Idea:
 - partir el proceso en unidades lógicas
 - usar la memoria necesaria para cada unidad
 - cargar los segmentos a medida que se los necesite

Segmentación (cont.)



Segmentación (cont.)

- Direcciones lógicas:

10 ...10 011 ...1001
└────────┘ └──────────┘
segmento desplazamiento

Segmentación

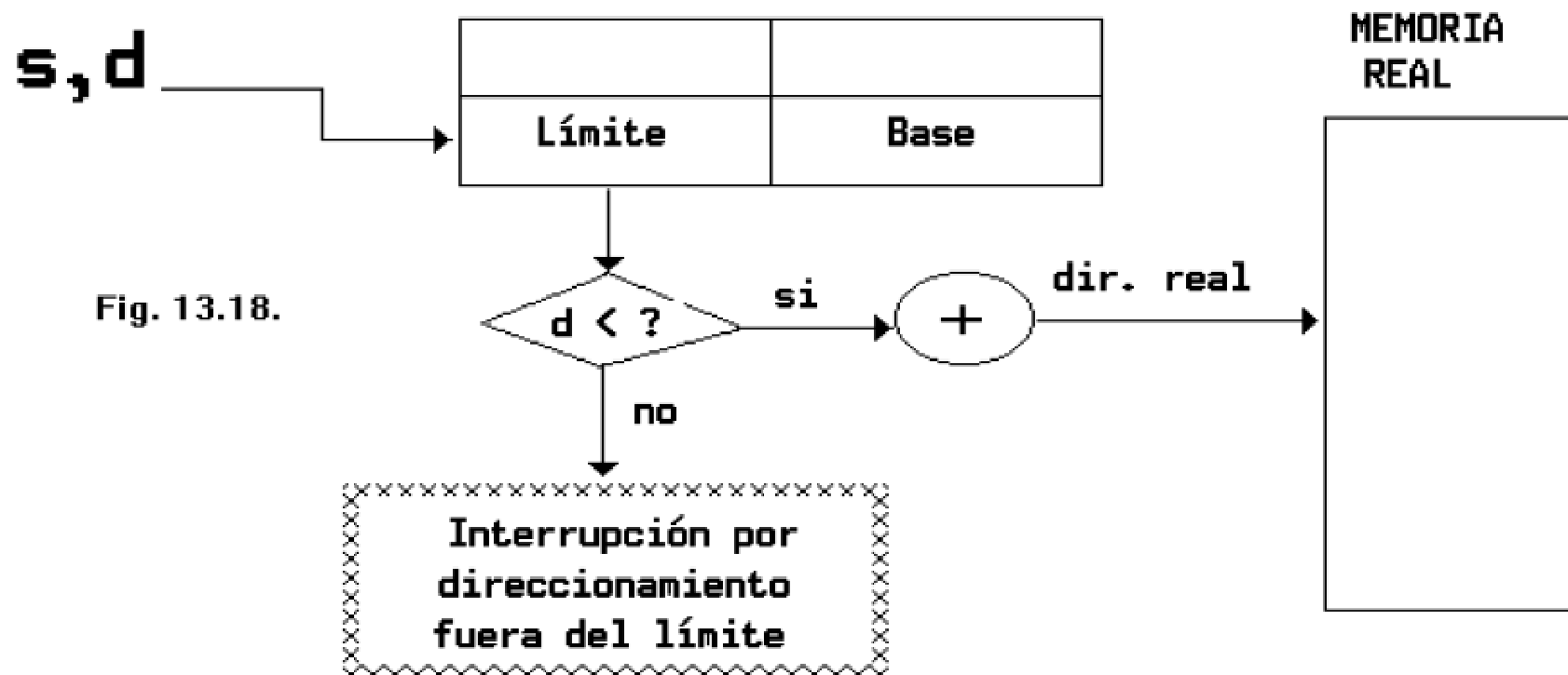


Fig. 13.18.

BCP 1

Número segmento	Nombre segmento	Apuntador segmentos activos
0	P1	3
1	SUB1	4
2	SEN0	9

BCP 2

TABLA DE MAPAS Y NOMBRES DE SEGMENTOS

Número segmento	Nombre segmento	Apuntador segmentos activos
0	P2	1
1	SUB2	5
2	SUB3	6
3	SEN0	9

Espacio de direcciones

TABLA DE ESTADO DE SEGMENTOS ACTIVOS

9

Nombre	Dir	Long.	Presencia en memor.	Permisos	Dir mem. virt	Cambio?	Uso
SEN0							

MEMORIA REAL

S. O.	P1	P2	SEN0	SUB3	SUB2	SUB1	...
-------	----	----	------	------	------	------	-----

Tabla de Permisos

TABLA DE AREAS NO ASIGNADAS

Dir.	Longitud

Segmentación

Importante:

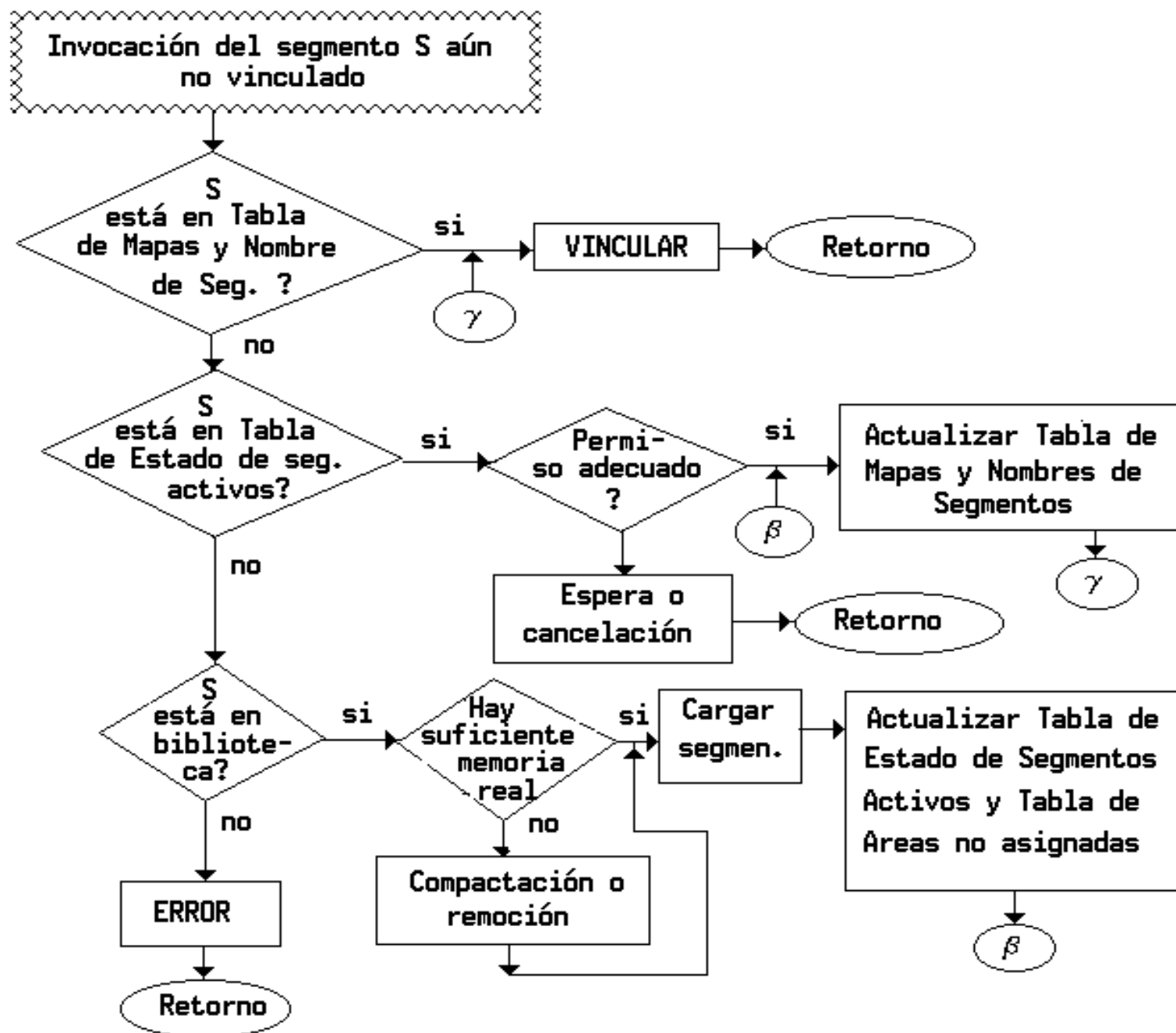
- para poder compartir un segmento entre distintos procesos, debe ser reentrante (no posee variables propias sino que las referencia siempre en base a un direccionamiento indirecto y su código no se automodifica).

Segmentación (cont.)



Proc1

No hay fragmentación interna.
Sí hay fragmentación externa a los procesos.



Segmentación paginada

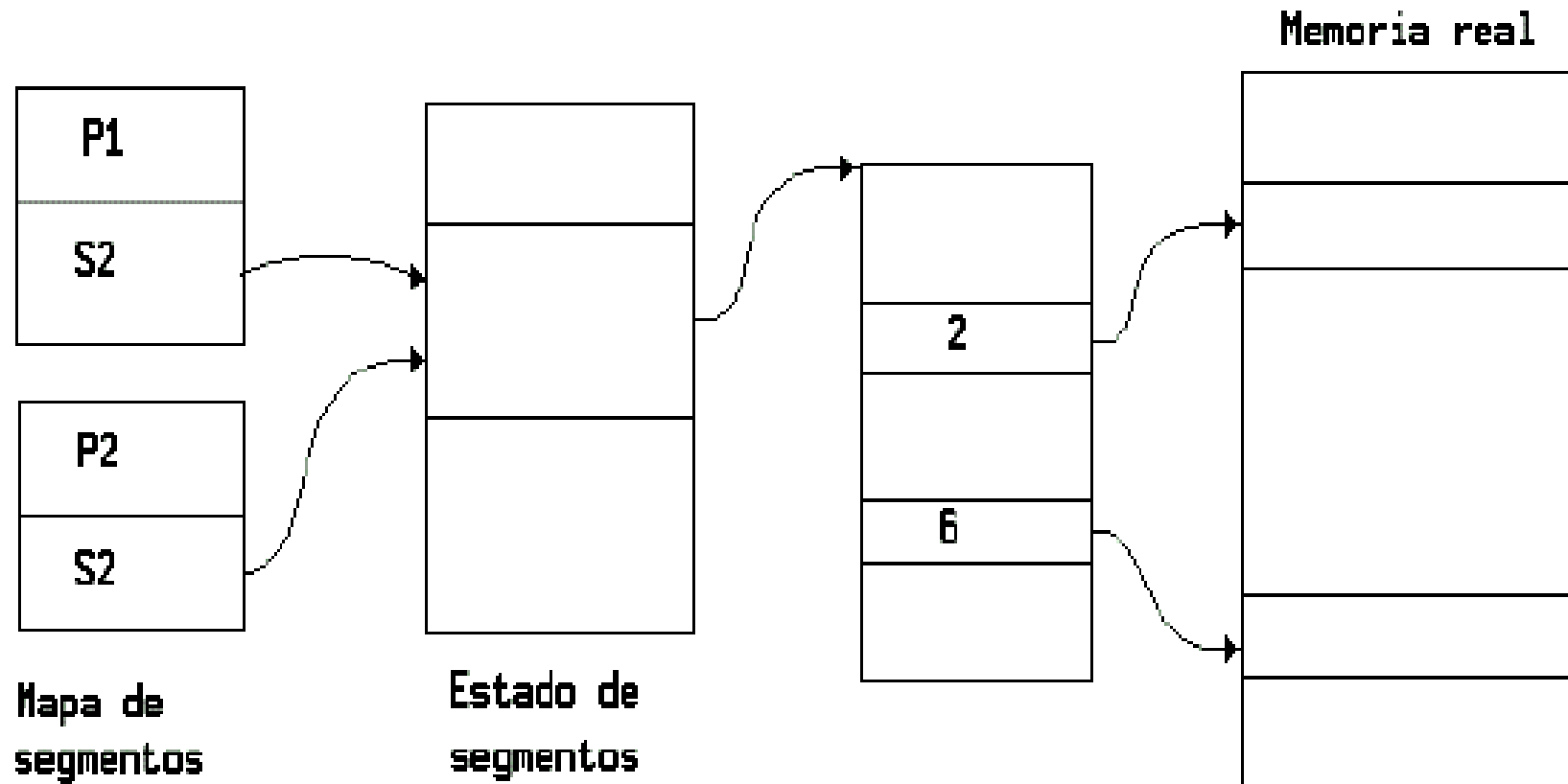


Fig. 13.23. - Administración de Segmentación Paginada.

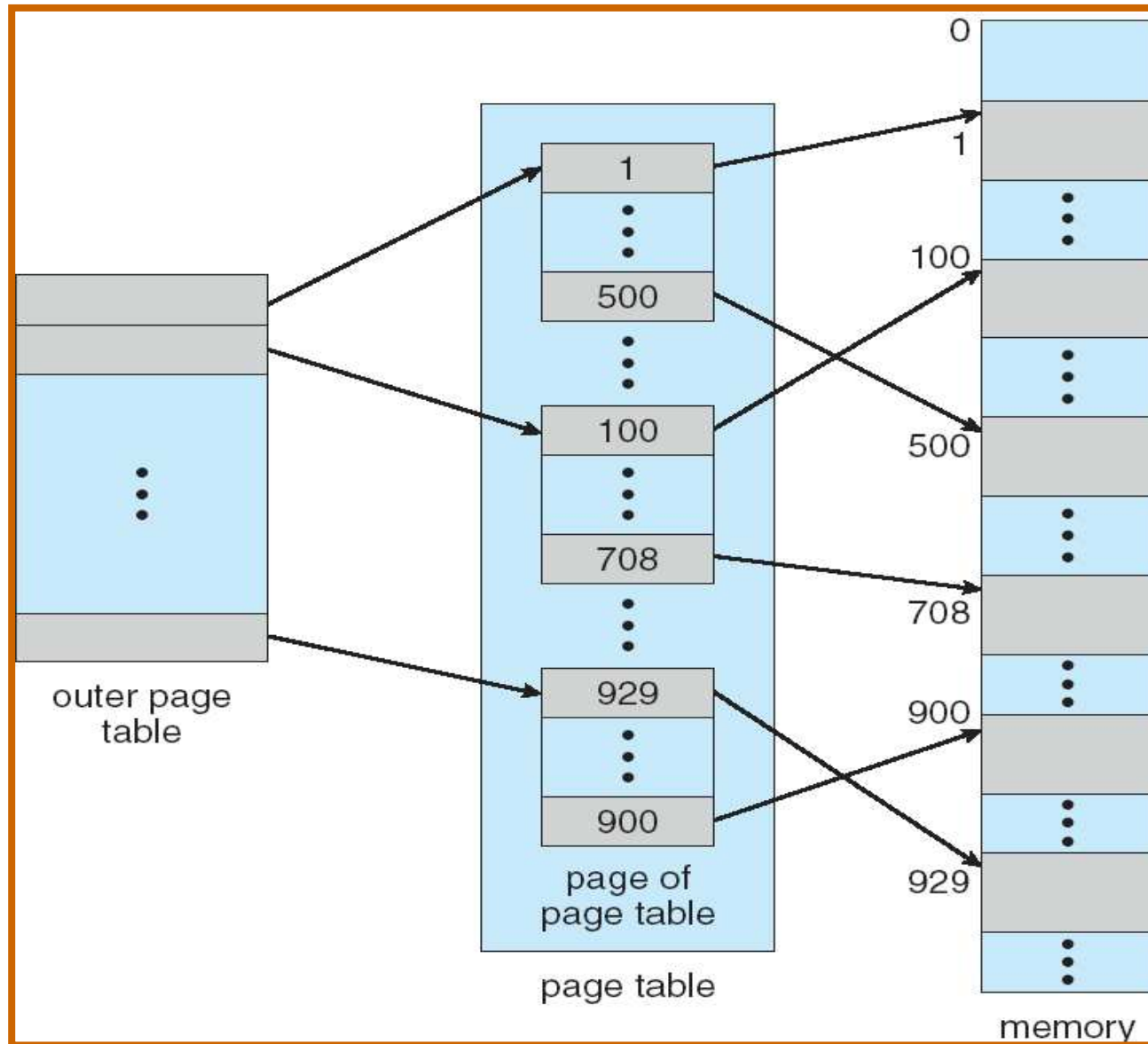
Similar a segmentación, con el agregado que cada segmento es dividido en páginas.

Ventajas y desventajas de Segmentación paginada

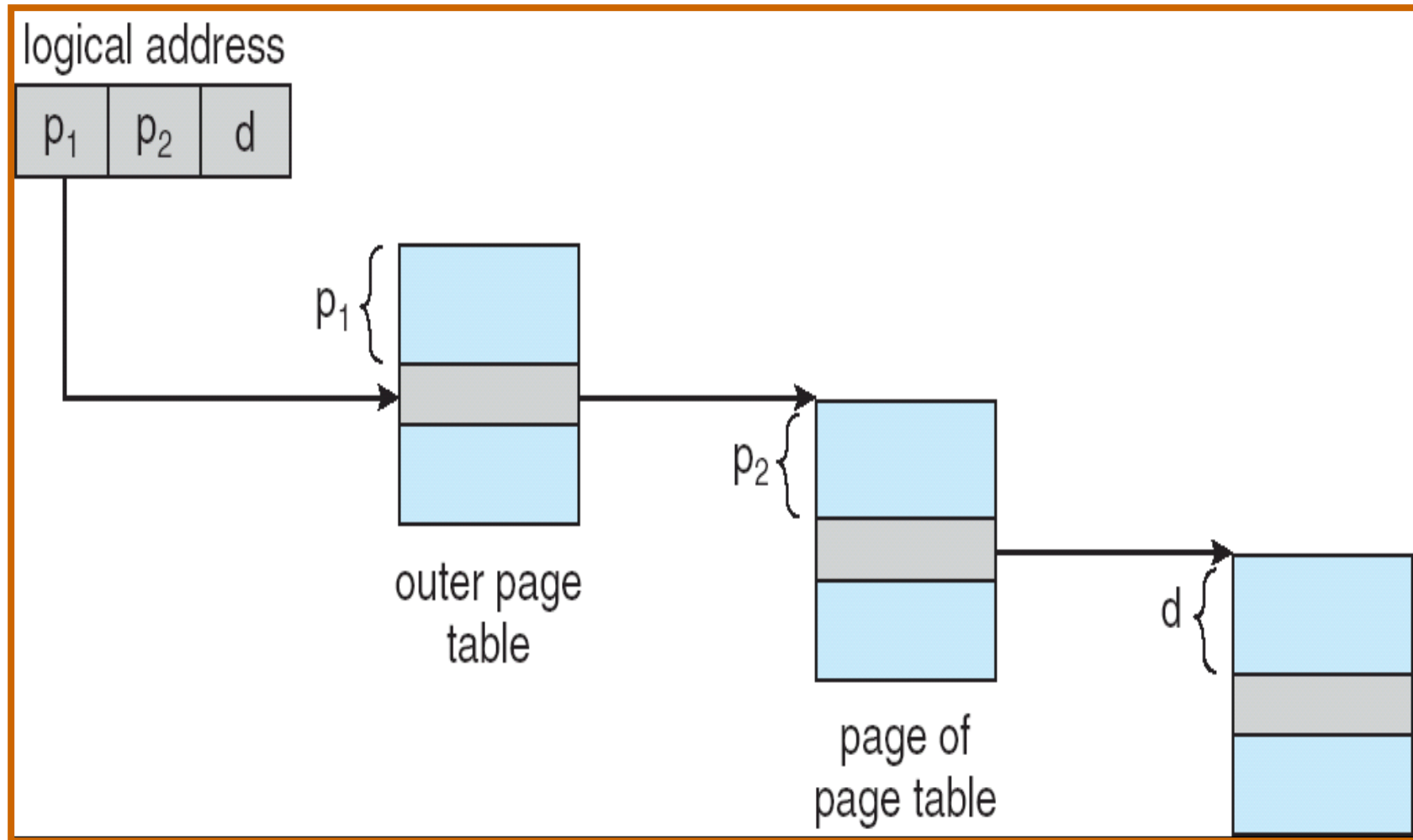
- Existe fragmentación interna al segmento en la ultima página de cada segmento
- El cálculo de las direcciones es más costoso
- Existe un mejor aprovechamiento del espacio. No es necesario compactar

Ejemplos de Sistemas de paginación

Paginación de dos niveles



Paginación de dos niveles



Paginación de tres niveles - Linux

