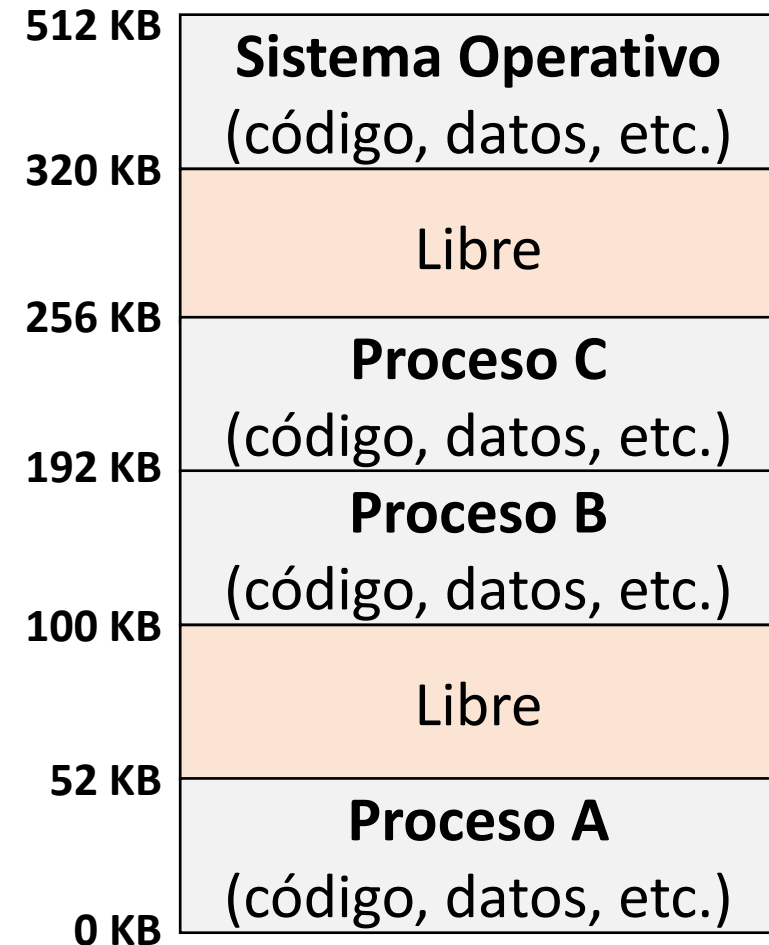


# **Asignación contigua**

Adaptación (ver referencias al final)

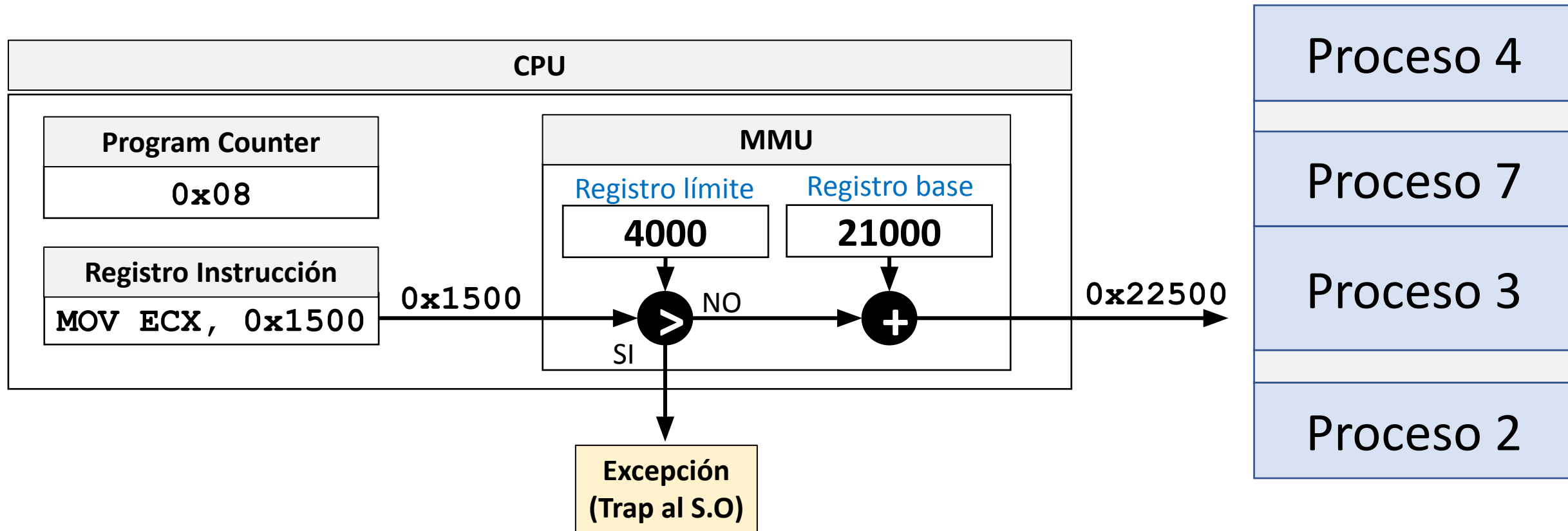
# Asignación contigua

- Se necesita asignar memoria a los procesos y S.O de la mejor manera posible.
- Usualmente la memoria tiene dos grandes particiones
  - Partición para el S.O (usualmente direcciones altas): Linux, Windows
  - Partición para para los procesos
- Se requiere que los procesos residan en memoria

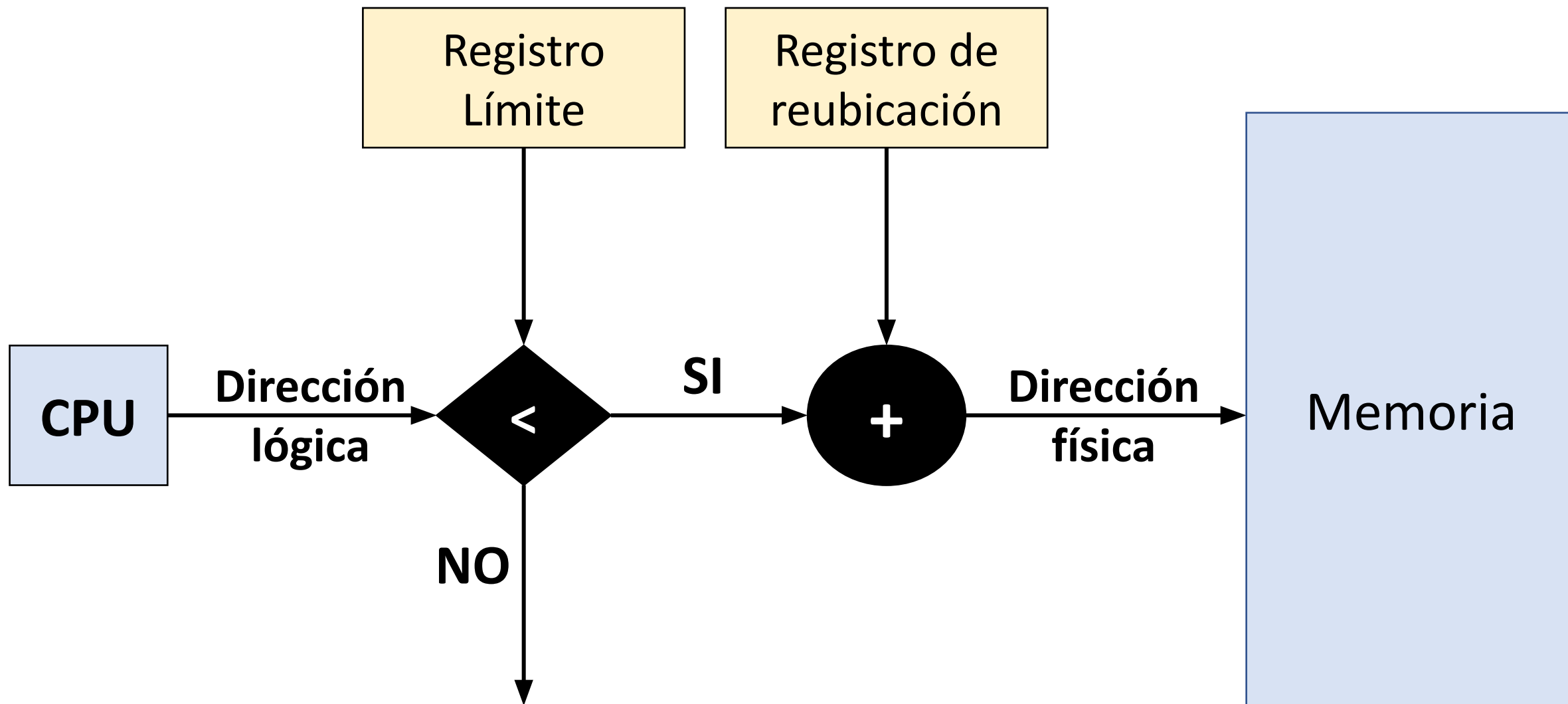


# Protección

- **Registro límite:** comprueba que cada dirección virtual NO es mayor que el valor de este registro.
- **Registro base:** Si dirección virtual no supera el límite, se suma a la dirección el valor del registro base para obtener la dirección física.



- **Registro base y registro límite** se pueden acceder solo en modo privilegiado.

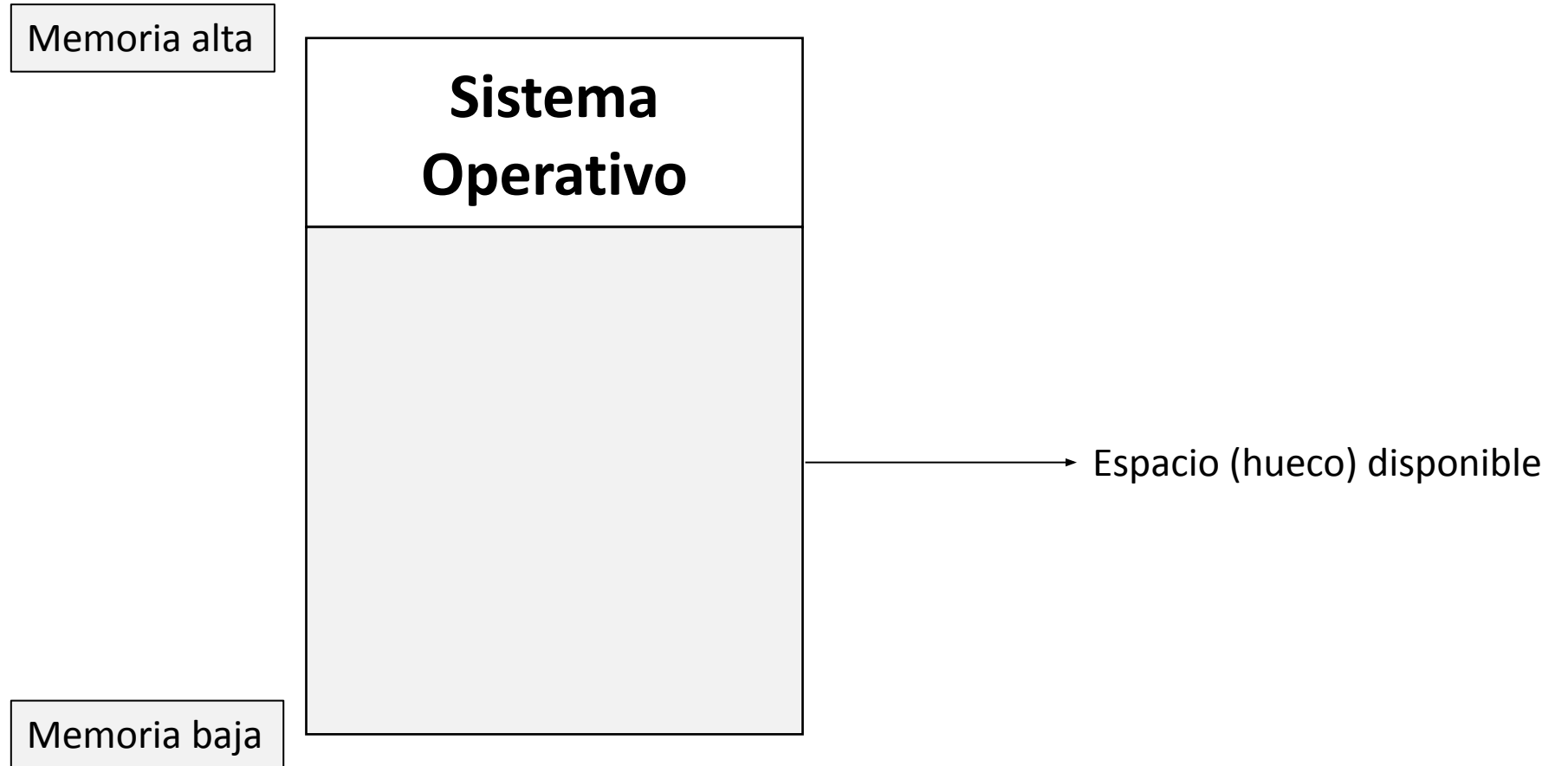


- S.O almacena en el PCB los valores de ambos registros para cada proceso
- En cada cambio de contexto se deben volver a cargar los registros

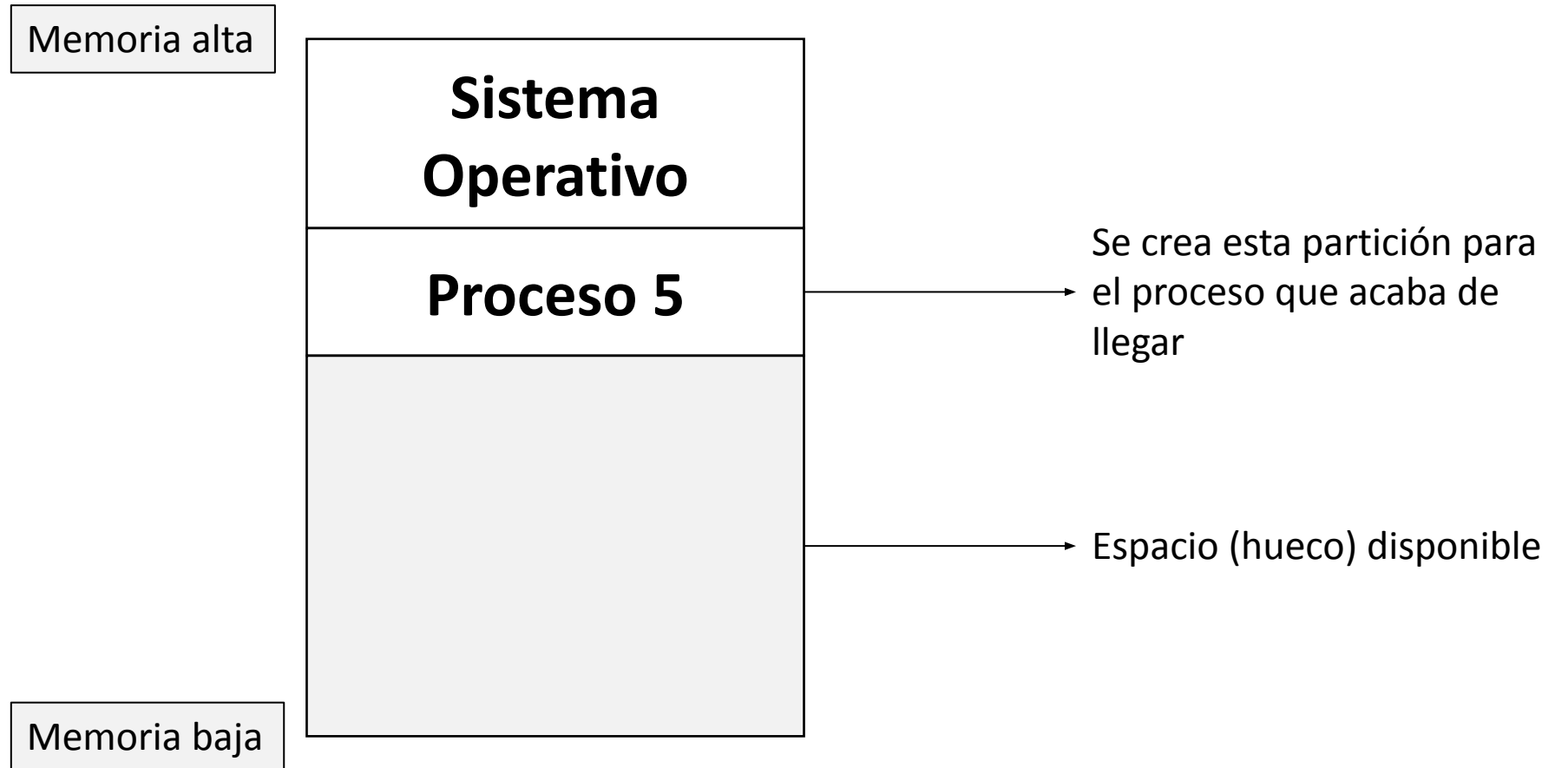
# Asignación contigua

- Cada proceso se asigna a una partición de memoria
- Cada partición (de tamaño variable) es asignada a un solo proceso
  - Particiones variables crean problema de fragmentación (luego...)
- S.O debe llevar registro de:
  - Espacios/particiones de memoria están ocupados (asignados a los procesos)
  - Espacios/particiones (huecos) de memoria disponible

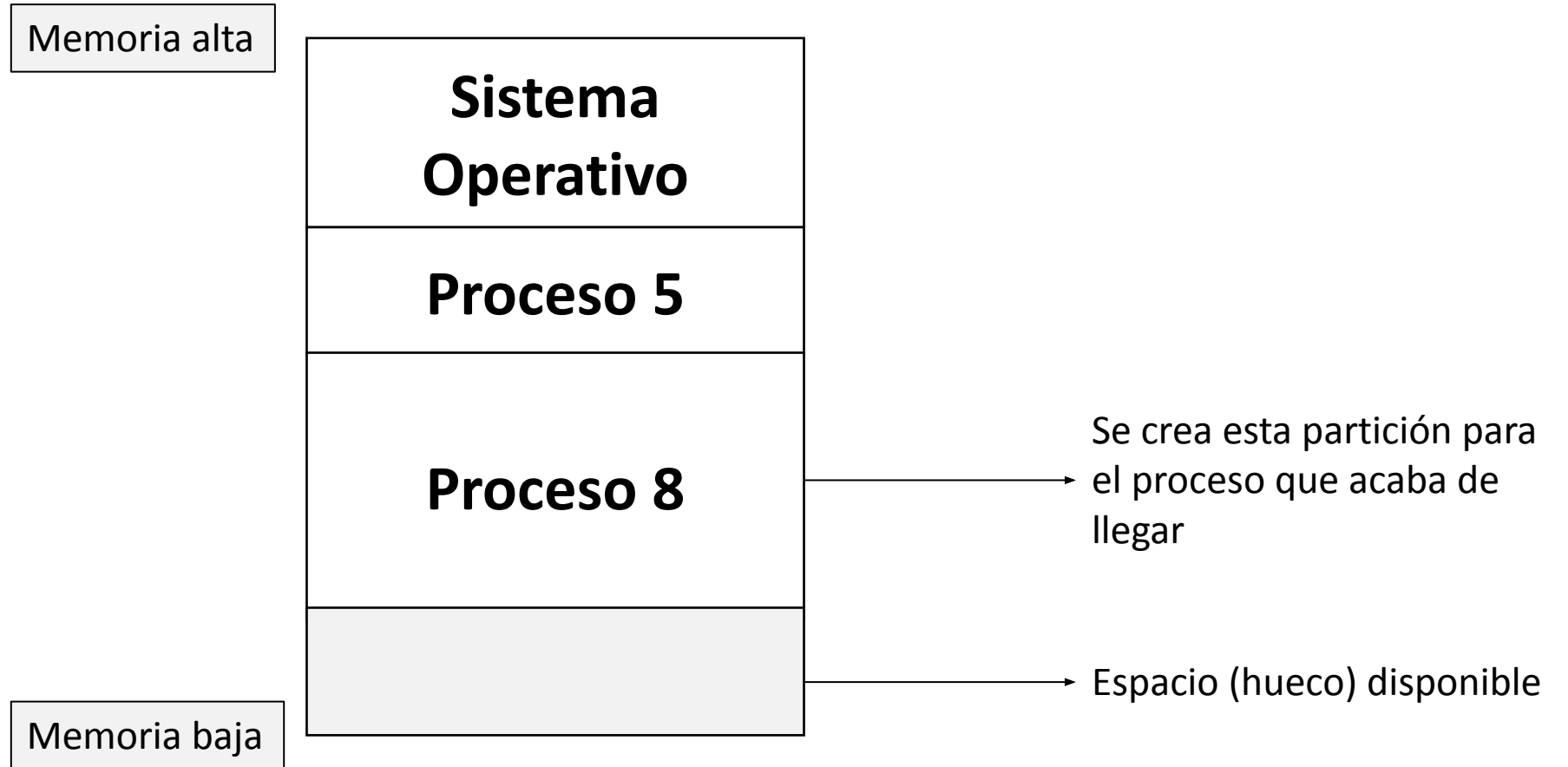
# Asignación contigua



# Asignación contigua

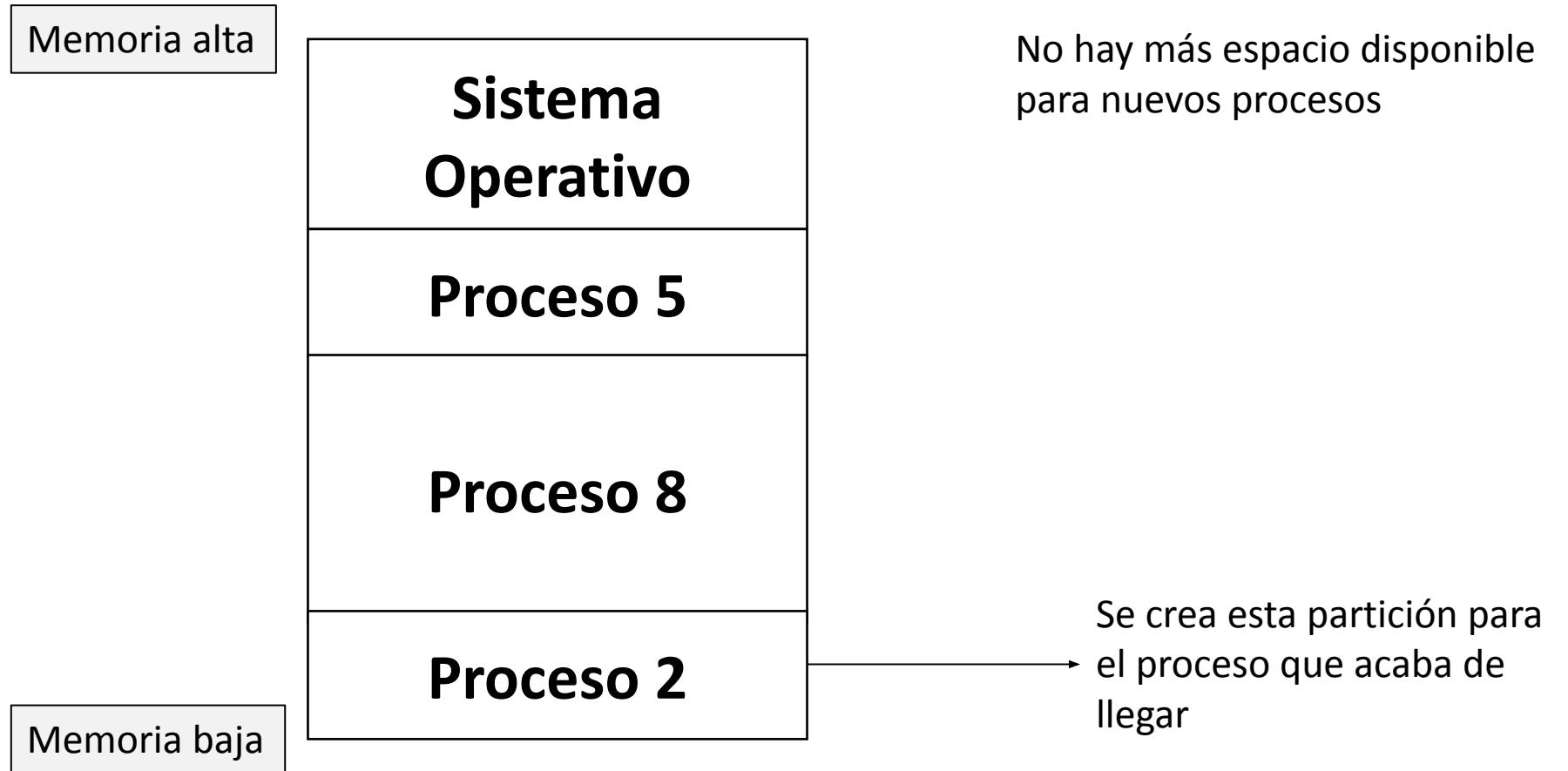


# Asignación contigua

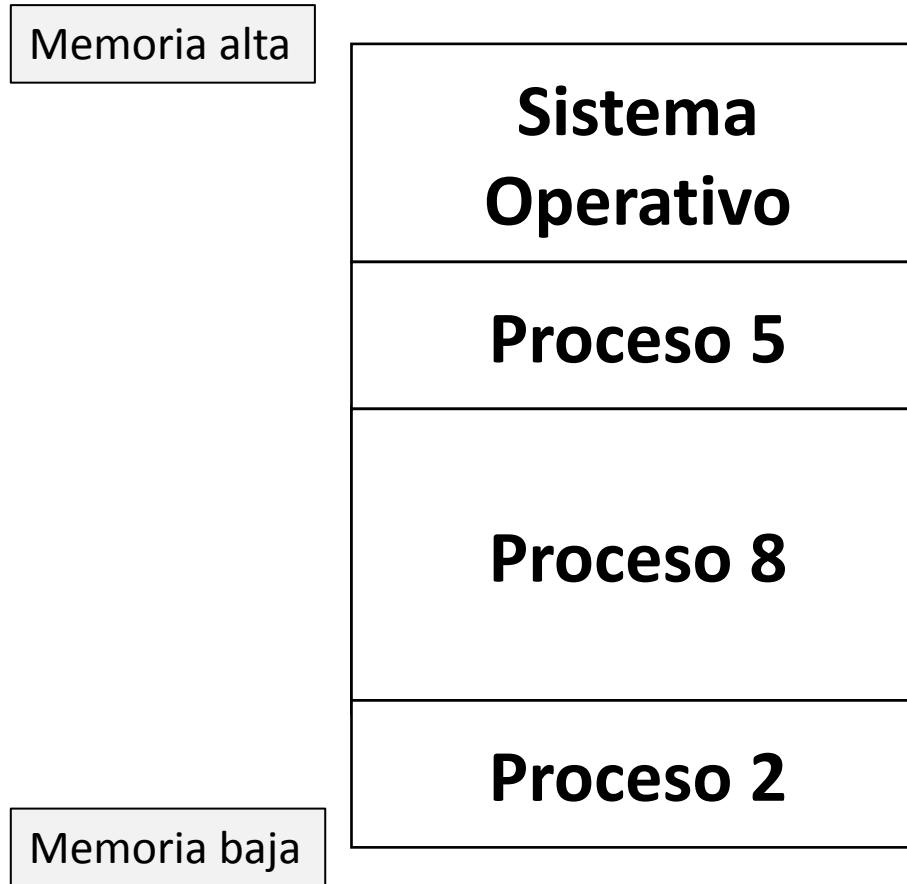




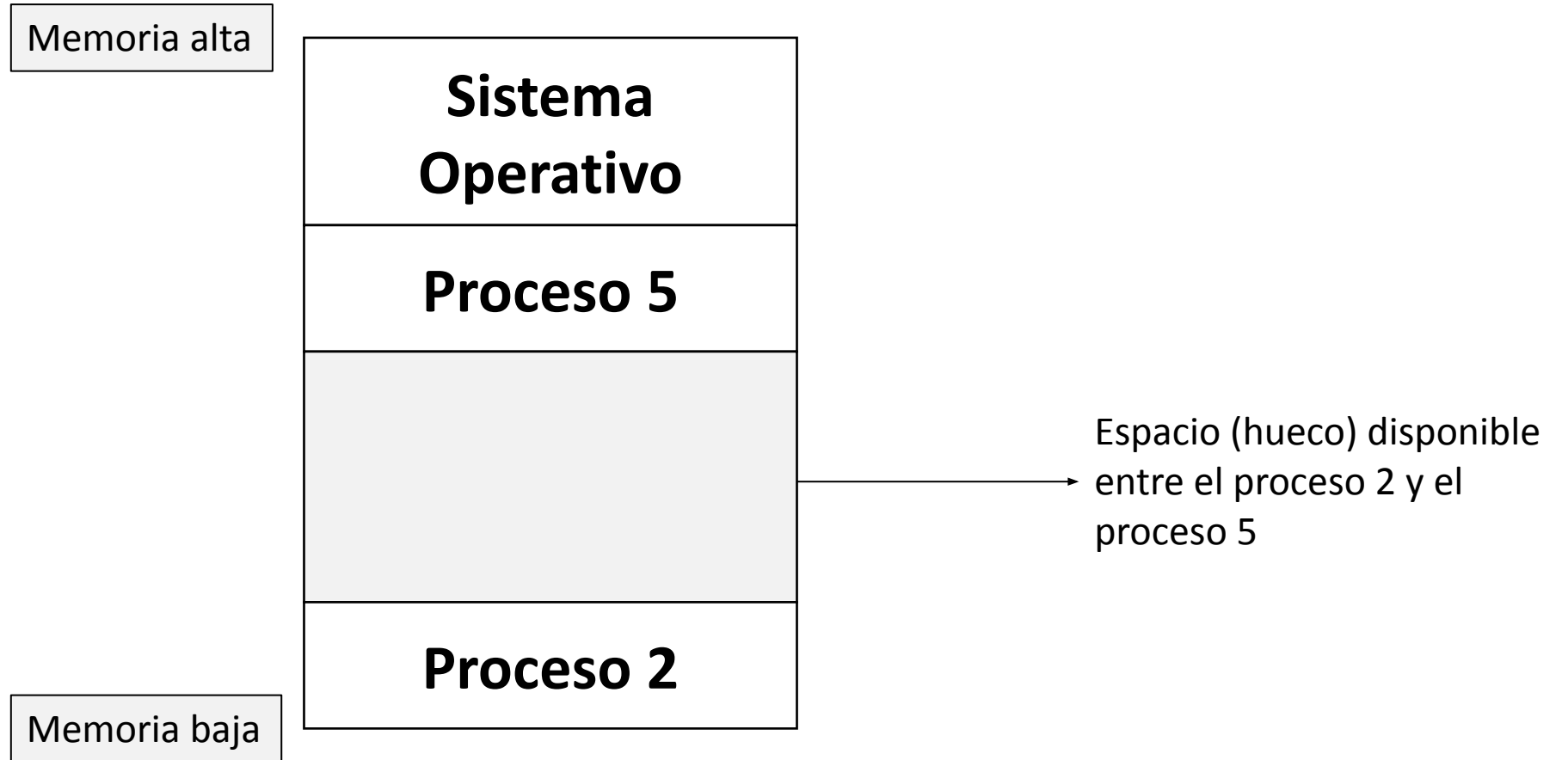
# Asignación contigua



# Asignación contigua



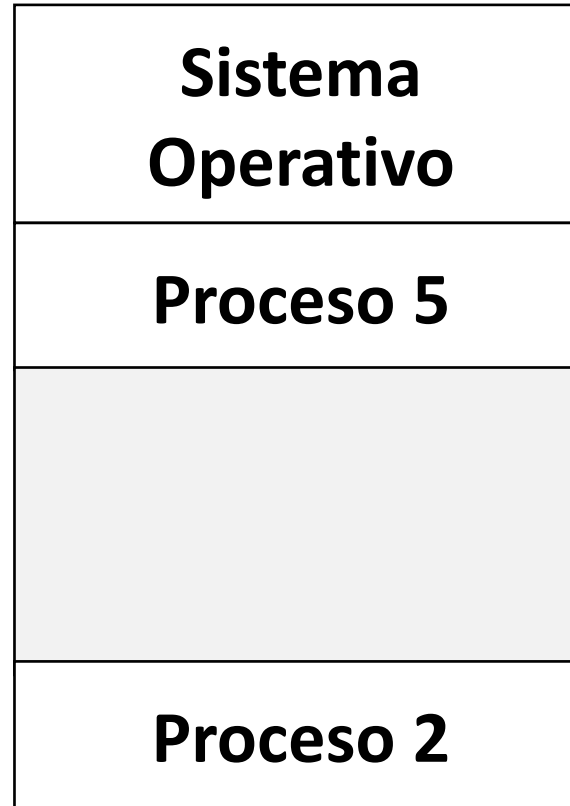
# Asignación contigua



# Asignación contigua

Memoria alta

Memoria baja

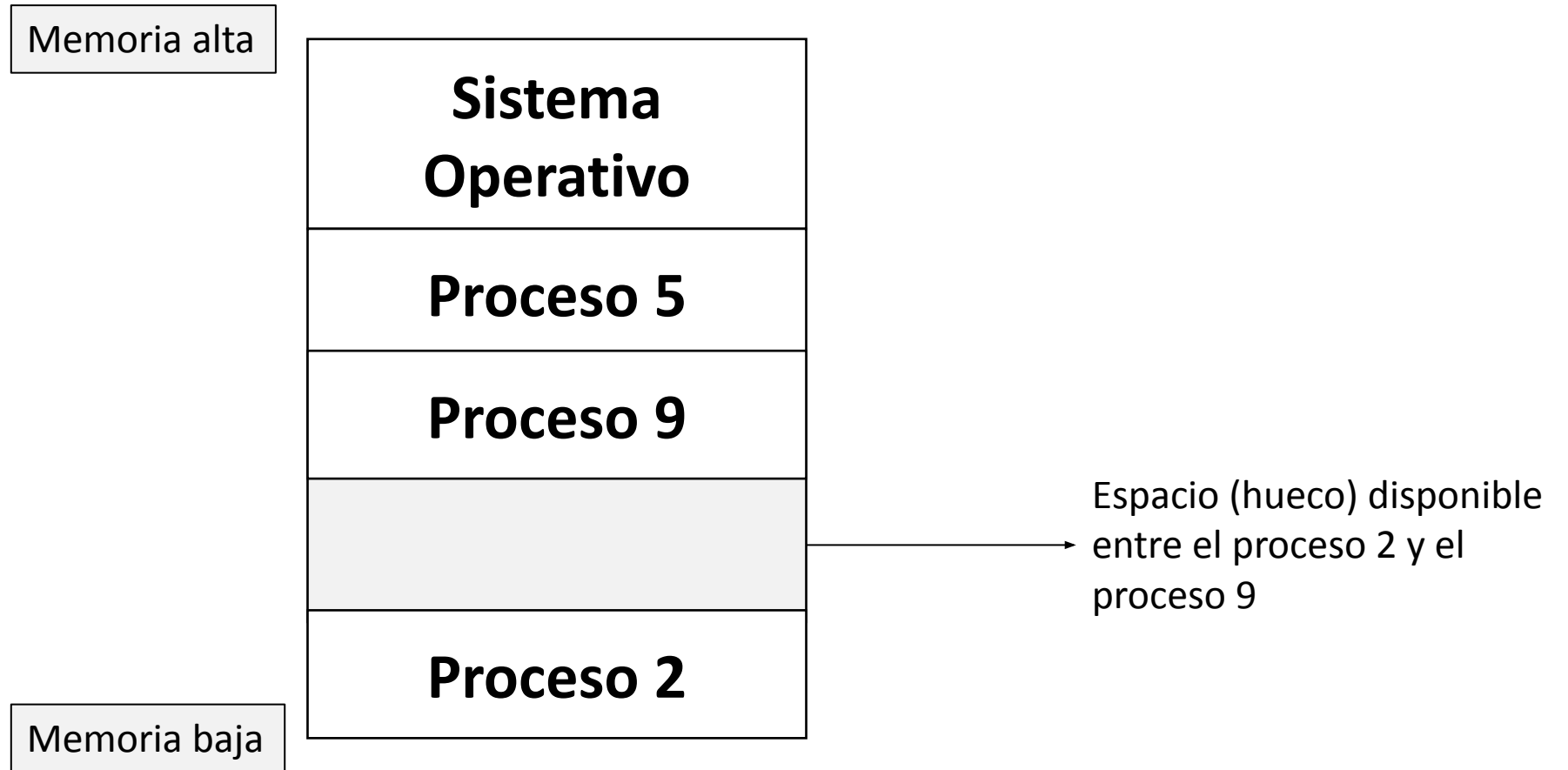


**Proceso 9**

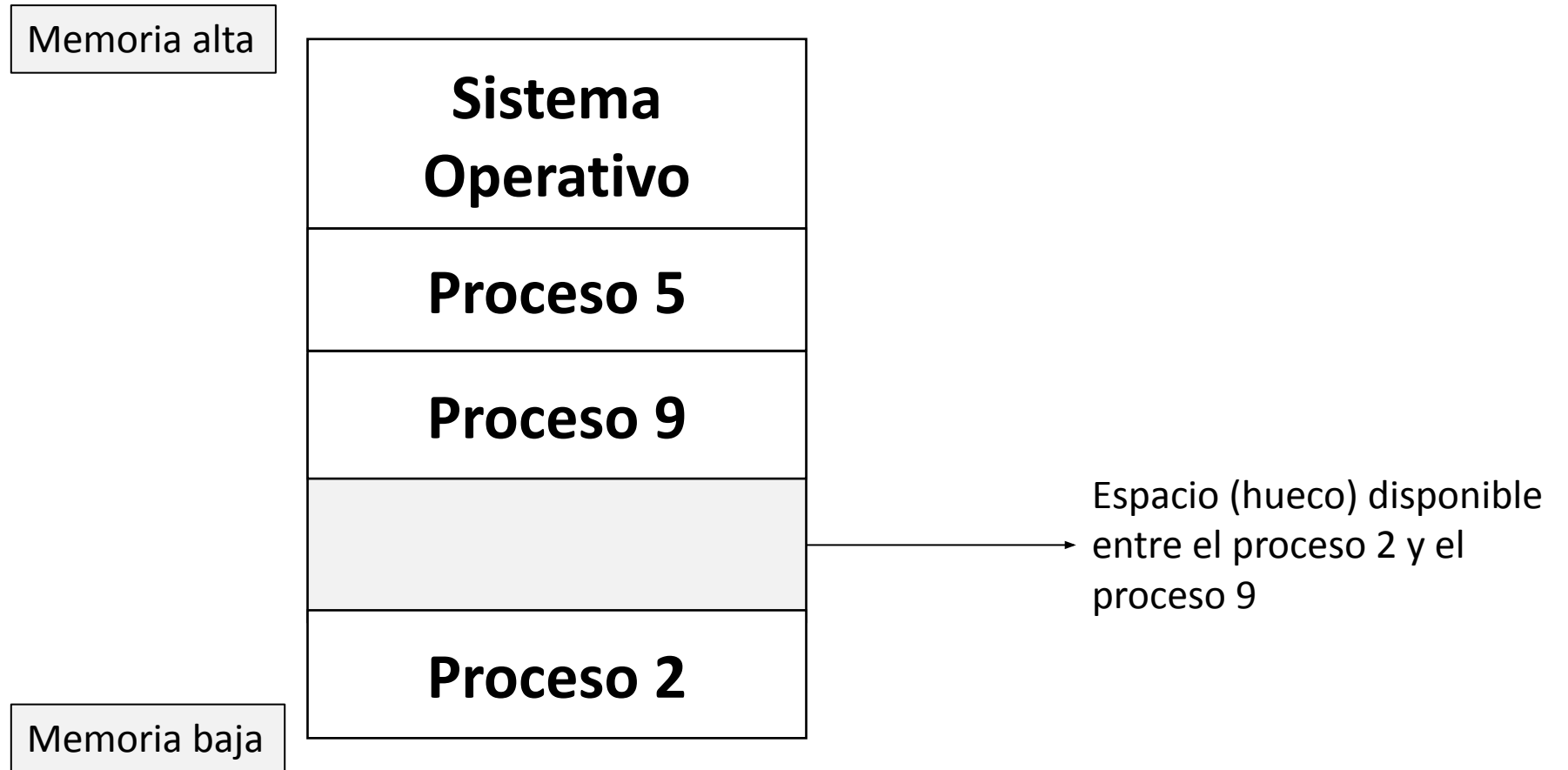
Llega este nuevo proceso

Espacio (hueco) disponible  
entre el proceso 2 y el  
proceso 5

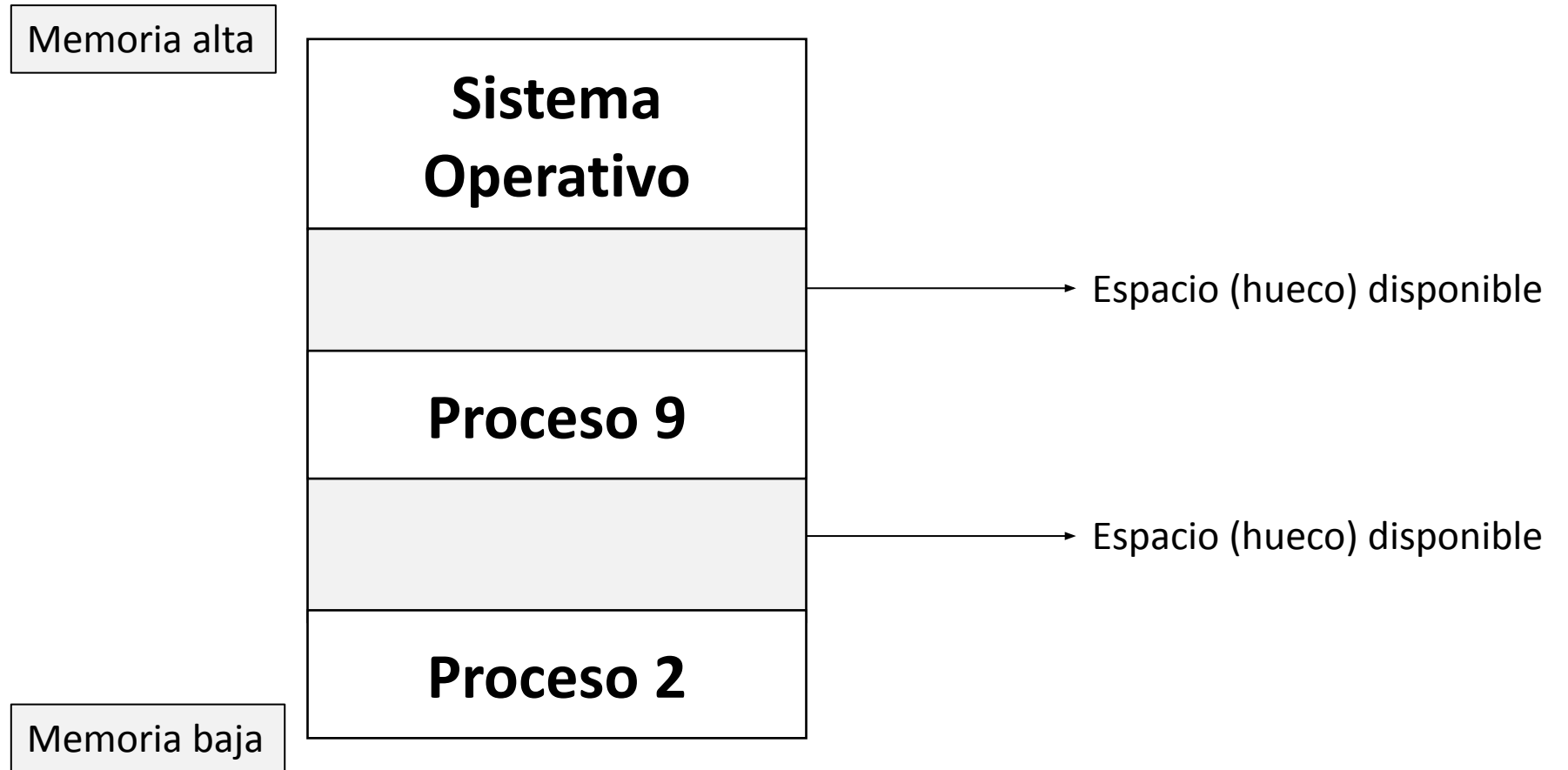
# Asignación contigua



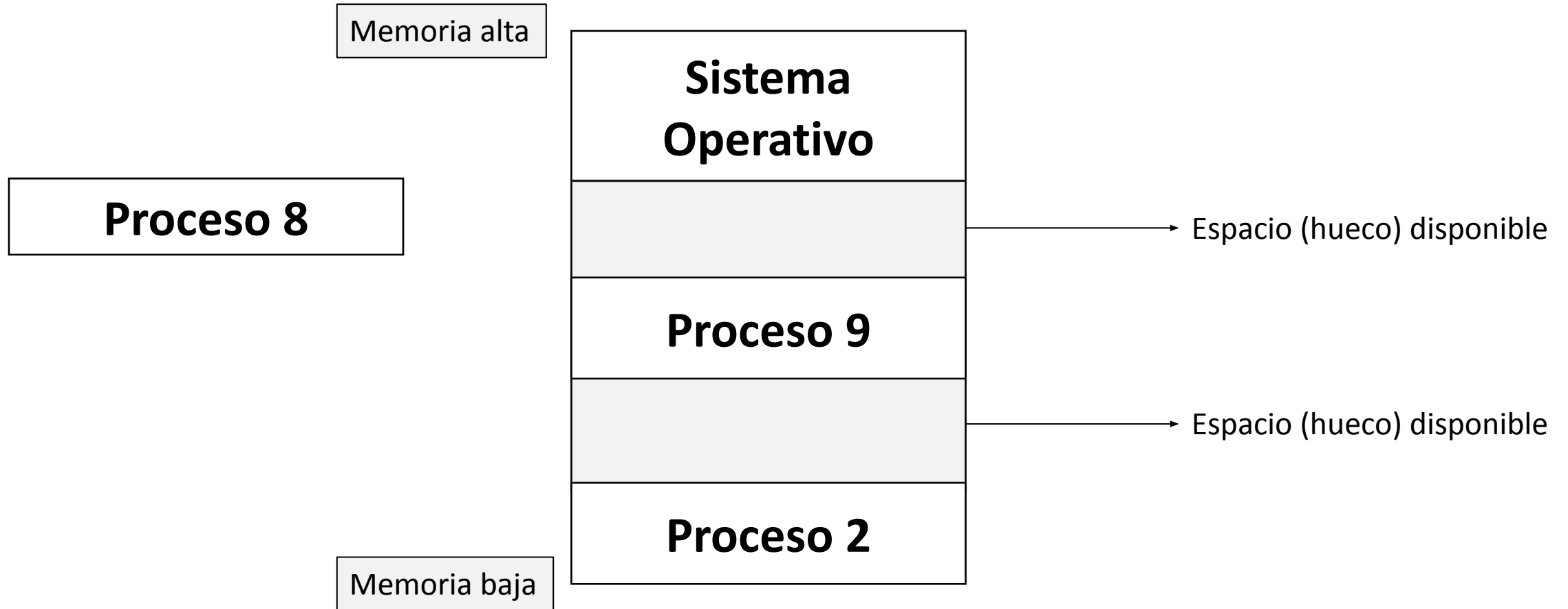
# Asignación contigua



# Asignación contigua

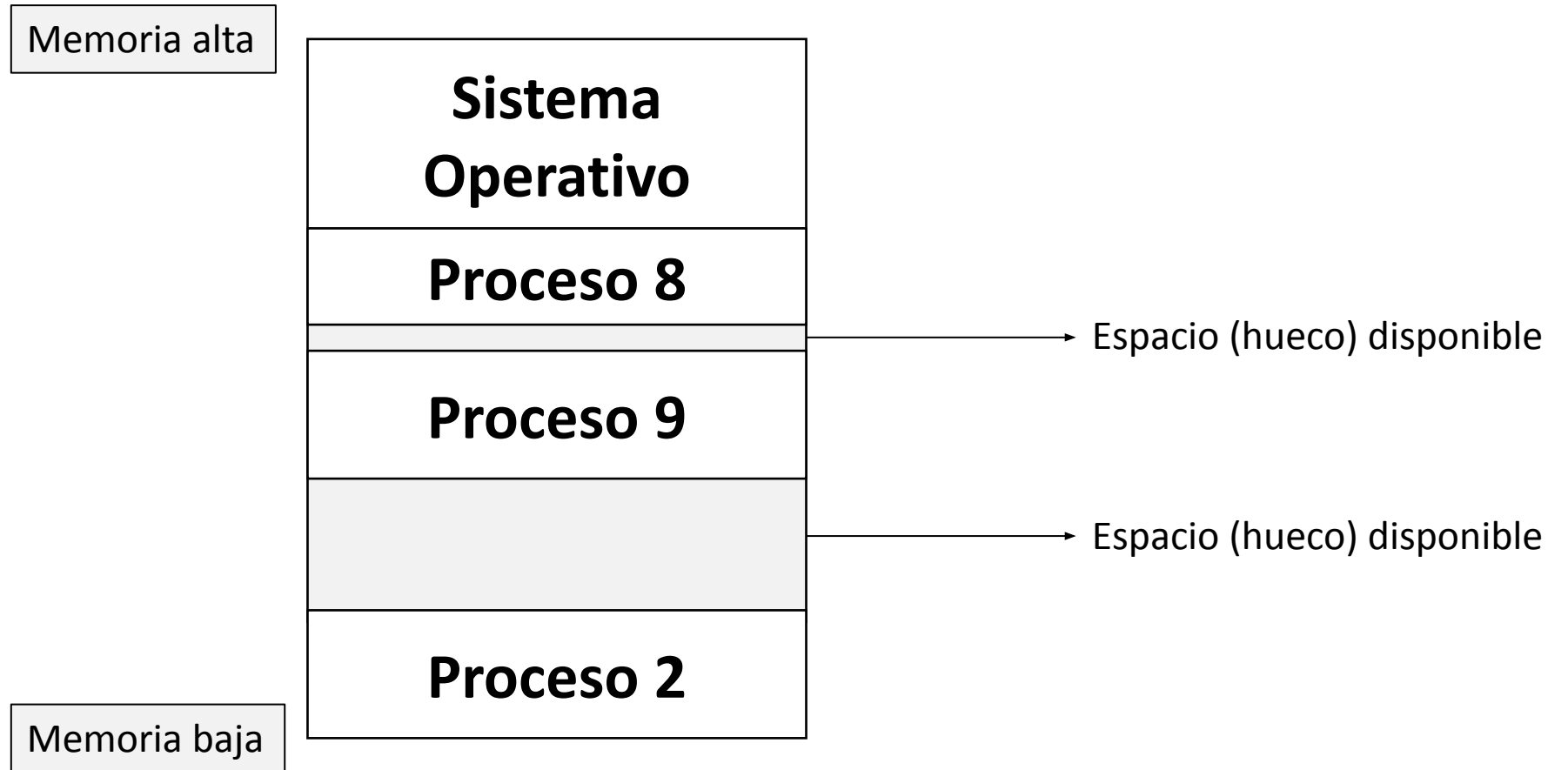


# Asignación contigua

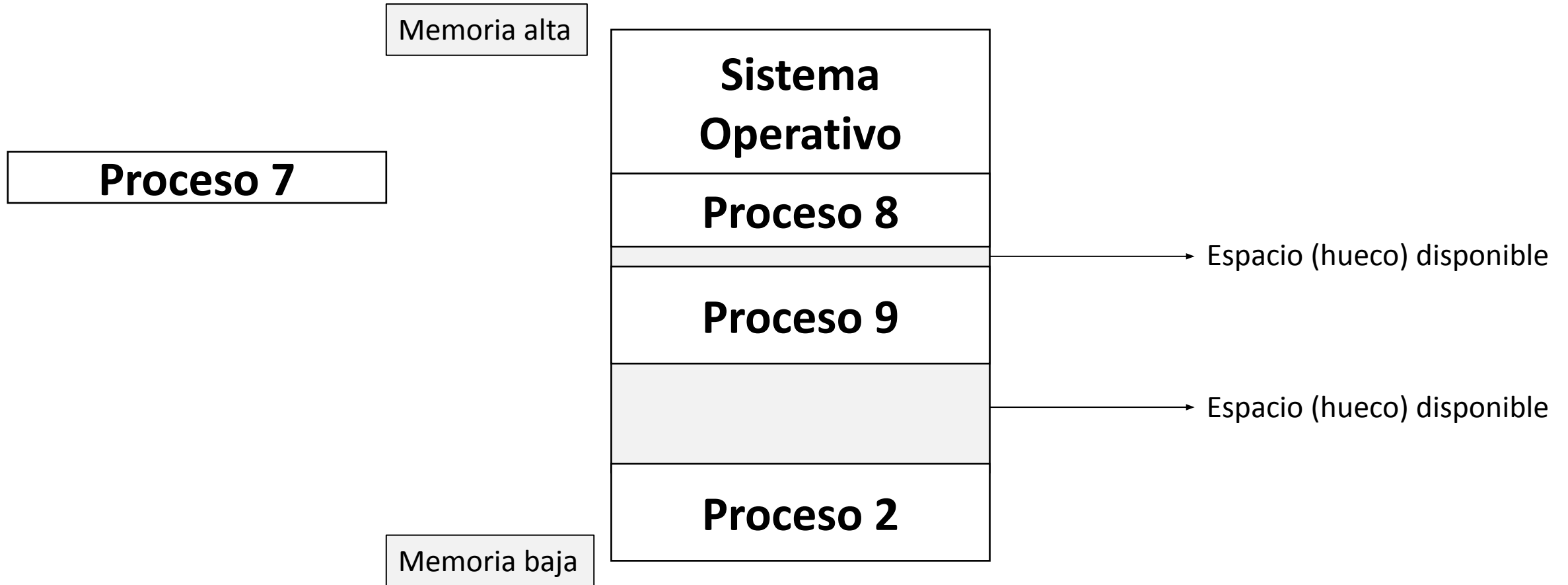




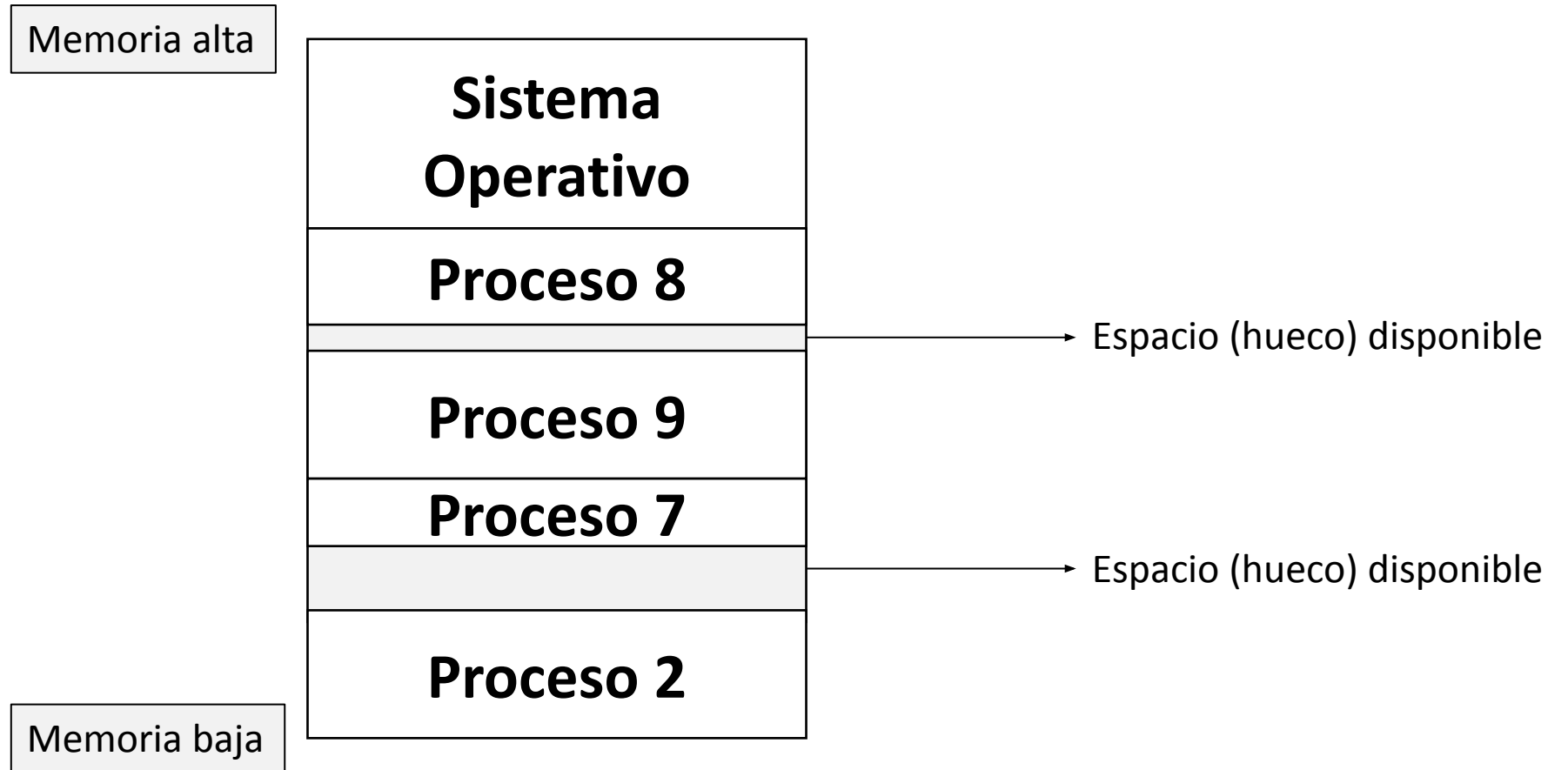
# Asignación contigua



# Asignación contigua



# Asignación contigua



# Asignación contigua

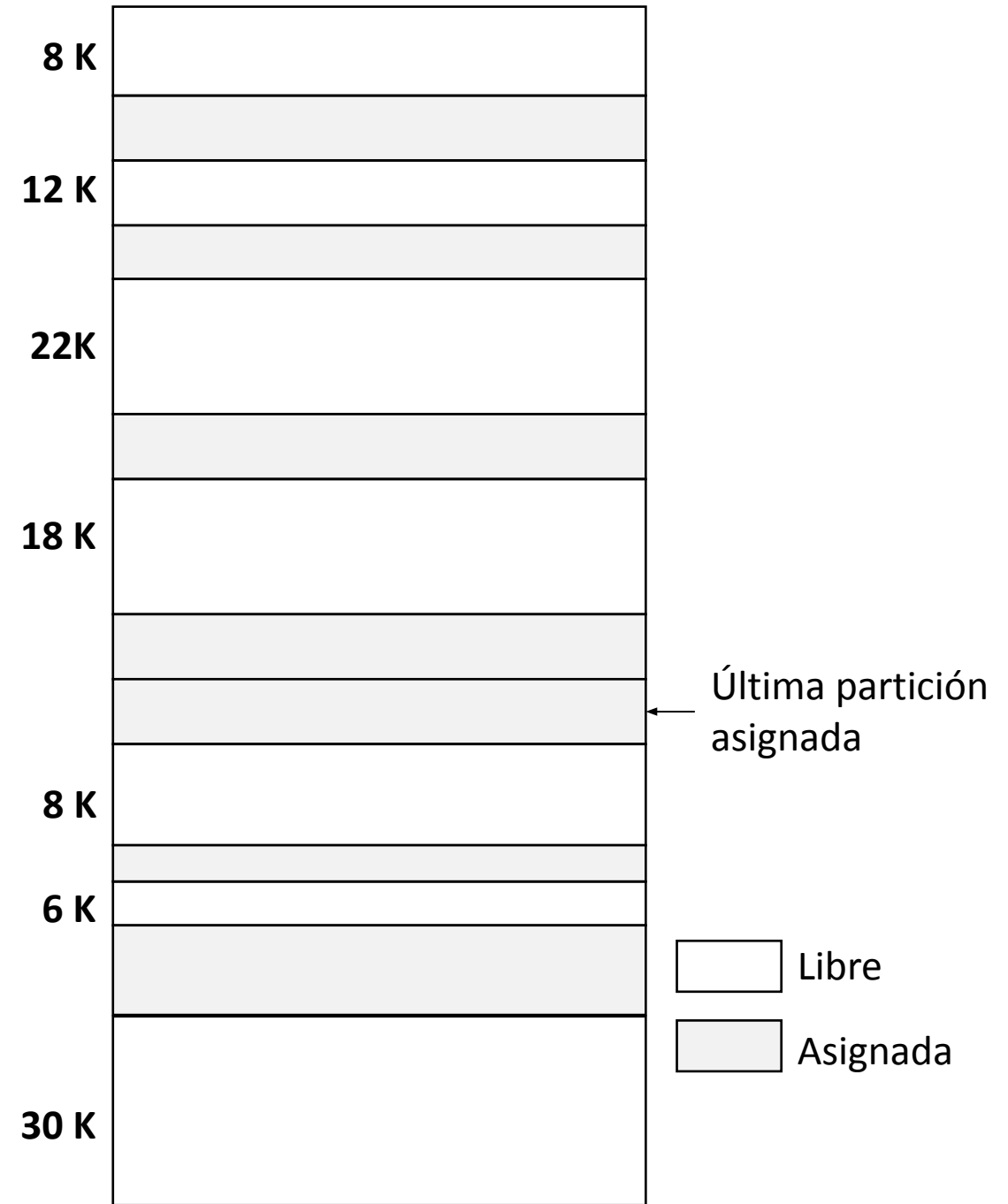
- ¿Qué sucede cuando no hay suficiente memoria para almacenar un nuevo proceso?
  - No aceptar el proceso
  - Ponerlo en cola de espera a que se libere el recurso
- Procesos que terminan devuelven el recurso de memoria ocupada
- Se debe llevar un registro de qué espacios (huecos) están disponibles
  - Espacios disponibles contiguos se unen
- En general se debe satisfacer la demanda de  $n$  bytes a partir del espacio disponible.

# Estrategias de asignación contigua

- Estrategias de asignación
  - First-fit (primer ajuste)
  - Best-fit (mejor ajuste)
  - Worst-fit (peor ajuste)
- Estrategias usadas para seleccionar un espacio libre para asignar a un proceso.

# Estrategias

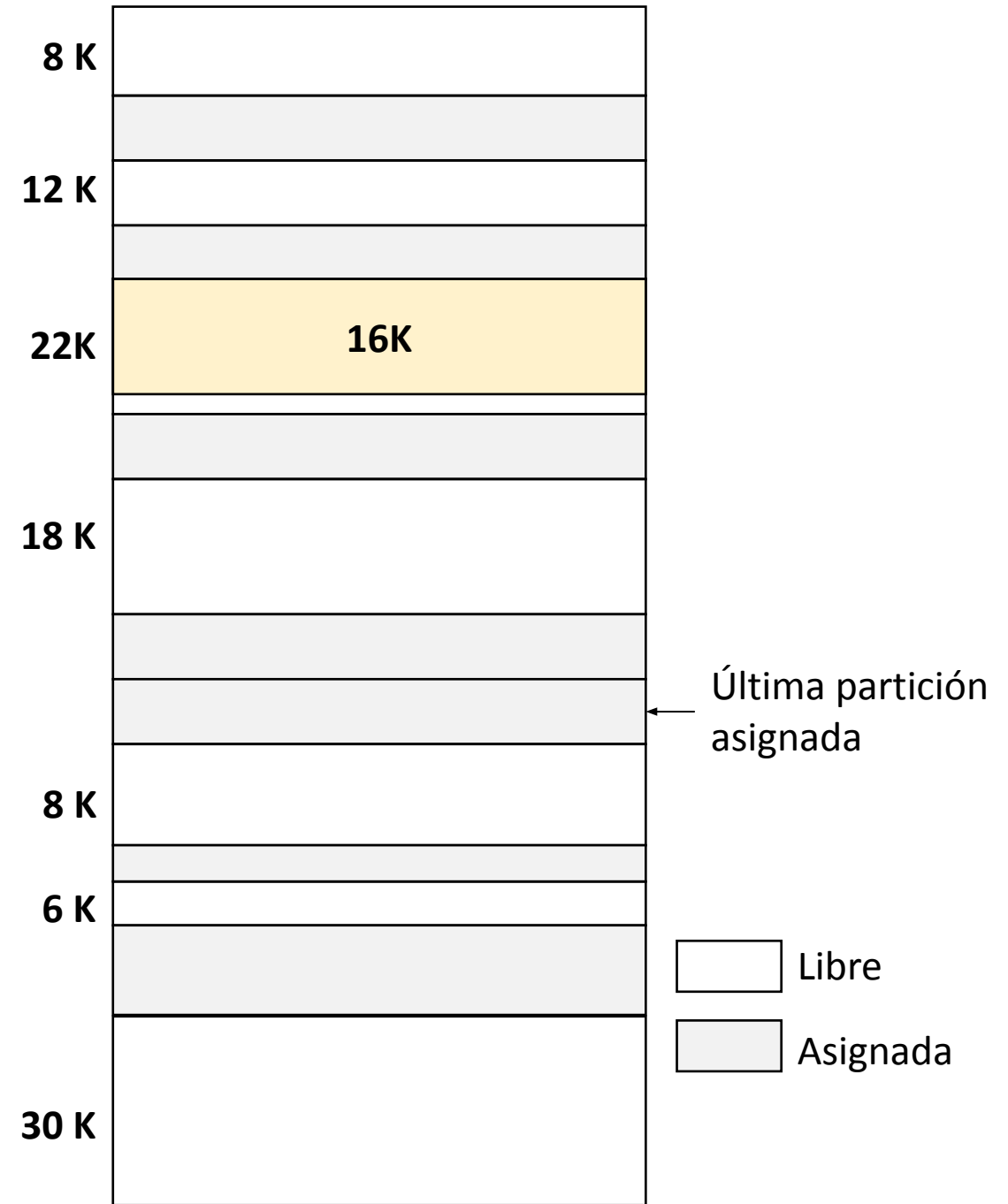
- Se necesita espacio para un proceso que mide 16K



# Estrategias

- **Primer ajuste**

- Primer hueco en donde quepa el proceso.
- Es la mejor política de ajuste



# Estrategias

- **Mejor ajuste**

- Buscar el hueco más pequeño en donde quepa el proceso
- Quedan huecos inutilizables
- Implica mantener una estructura (lista) ordenada por tamaño de los huecos
- Produce fragmentos muy pequeños

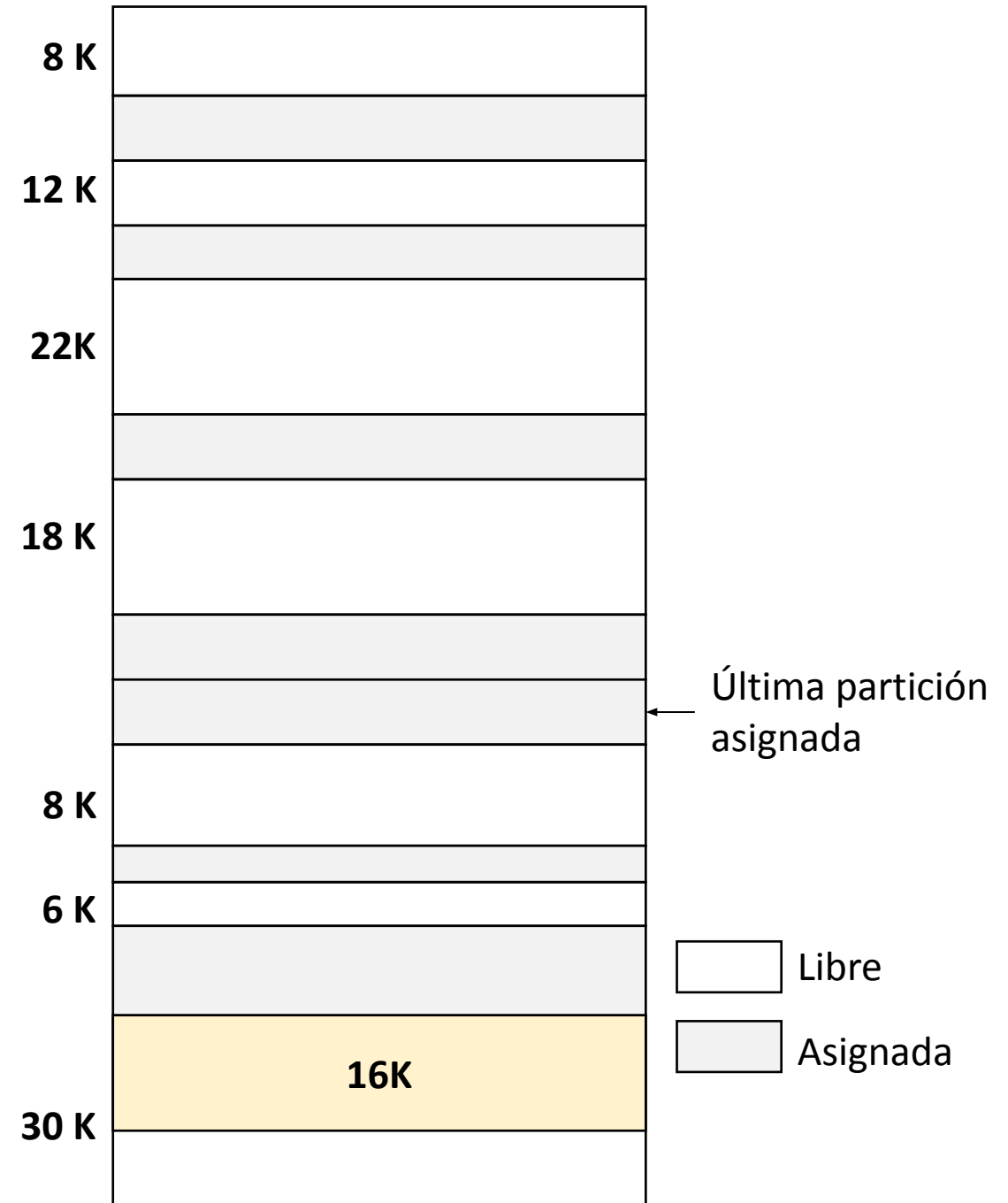




# Estrategias

- **Peor ajuste**

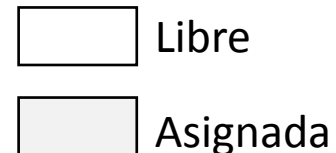
- Buscar el hueco más grande.
- Hay que mantener una lista de huecos ordenada por tamaño.
- Se busca dejar huecos no tan pequeños que sí sean utilizables
- Produce fragmentos muy grandes



# Fragmentación externa

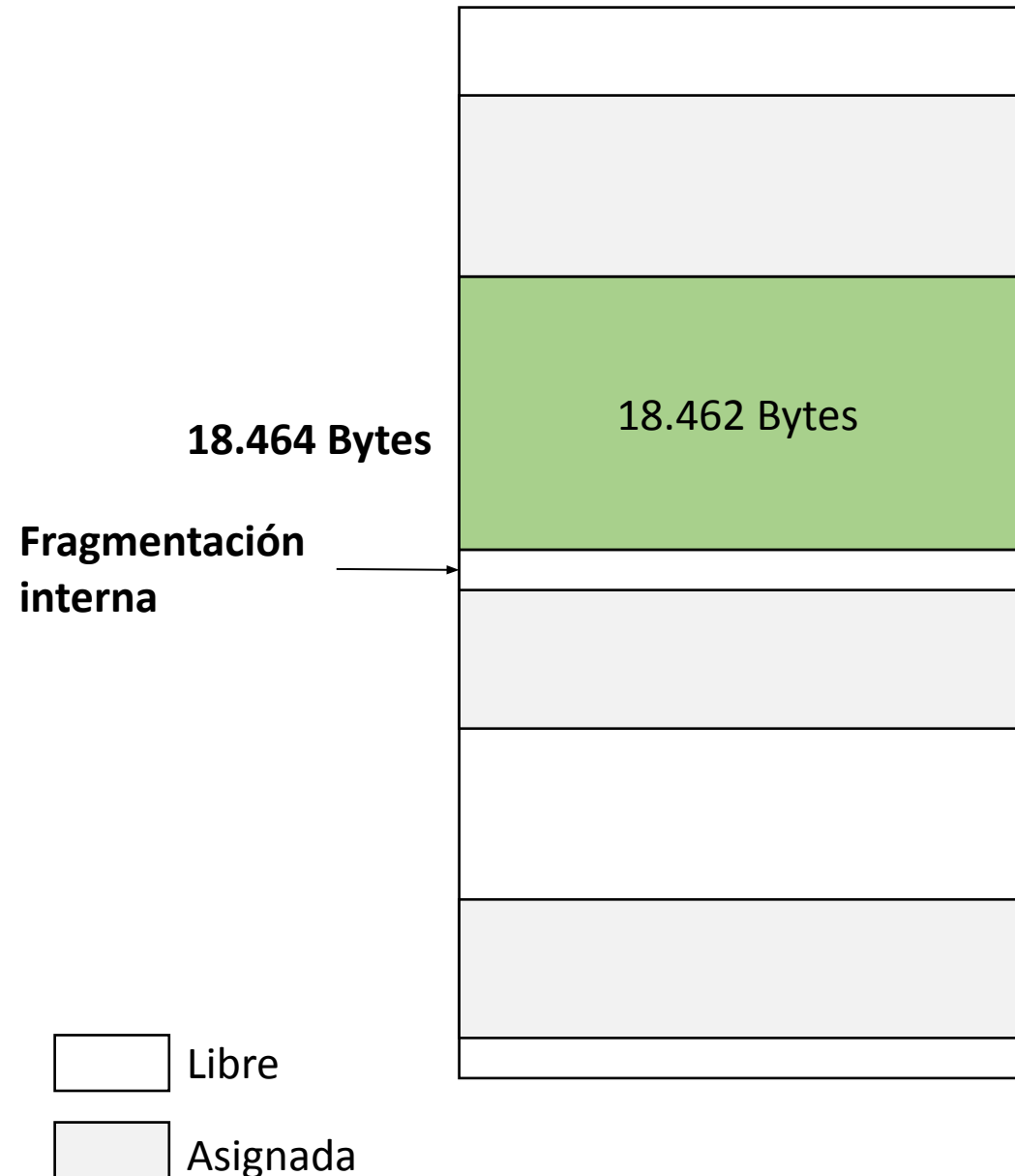
- Existe espacio disponible para satisfacer la demanda pero el espacio disponible es no contiguo.
- El espacio disponible está distribuido en huecos pequeños.
- Huecos entre las particiones asignadas,

Fragmentación externa



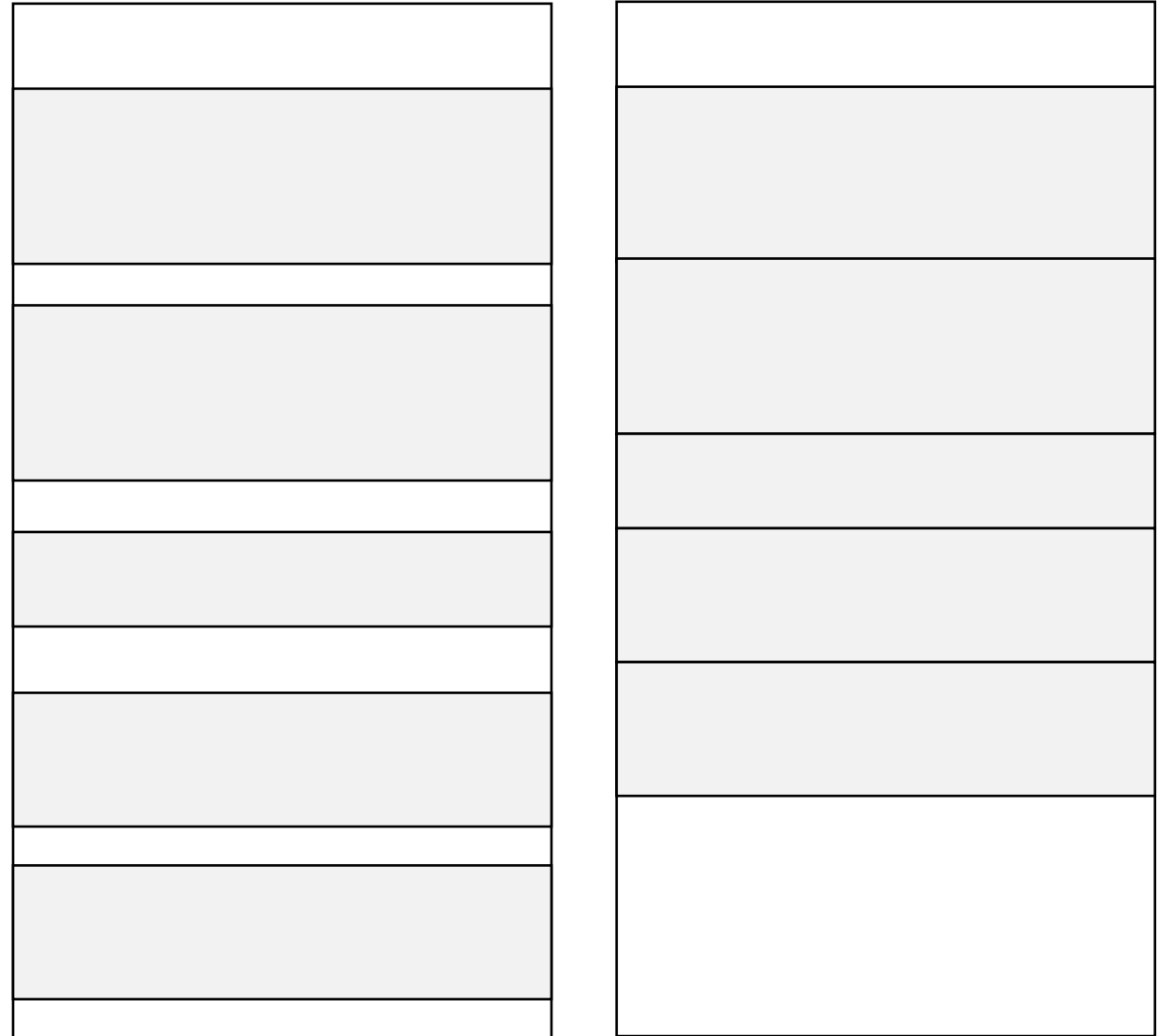
# Fragmentación interna

- Llega proceso solicitando 18.462 Bytes
- Se asigna todo el espacio disponible de los 18.464 Bytes
- Sobran 2 Bytes que es mejor tenerlos asignados a un proceso
  - Podría (o no) pedirlos (**malloc**, **new**)
- La memoria asignada es un poco más grande de la solicitada.



# Compactación

- Técnica para disminuir la fragmentación externa.
- Funciona siempre y cuando la **reubicación** de direcciones sea dinámica
  - Las direcciones se calculan en tiempo de ejecución
- Implica mover los procesos a nuevos espacios de memoria
  - ¿Cada cuanto?
  - ¿Costo?



# Referencias

- Carretero Pérez, J., García Carballeira, F., De Miguel Anasagasti, P., & Pérez Costoya, F. (2018). Contiguos Memory Allocation. In *Operating Systems Concepts* (10th ed., pp. 356–360). John Wiley & Sons, Inc.