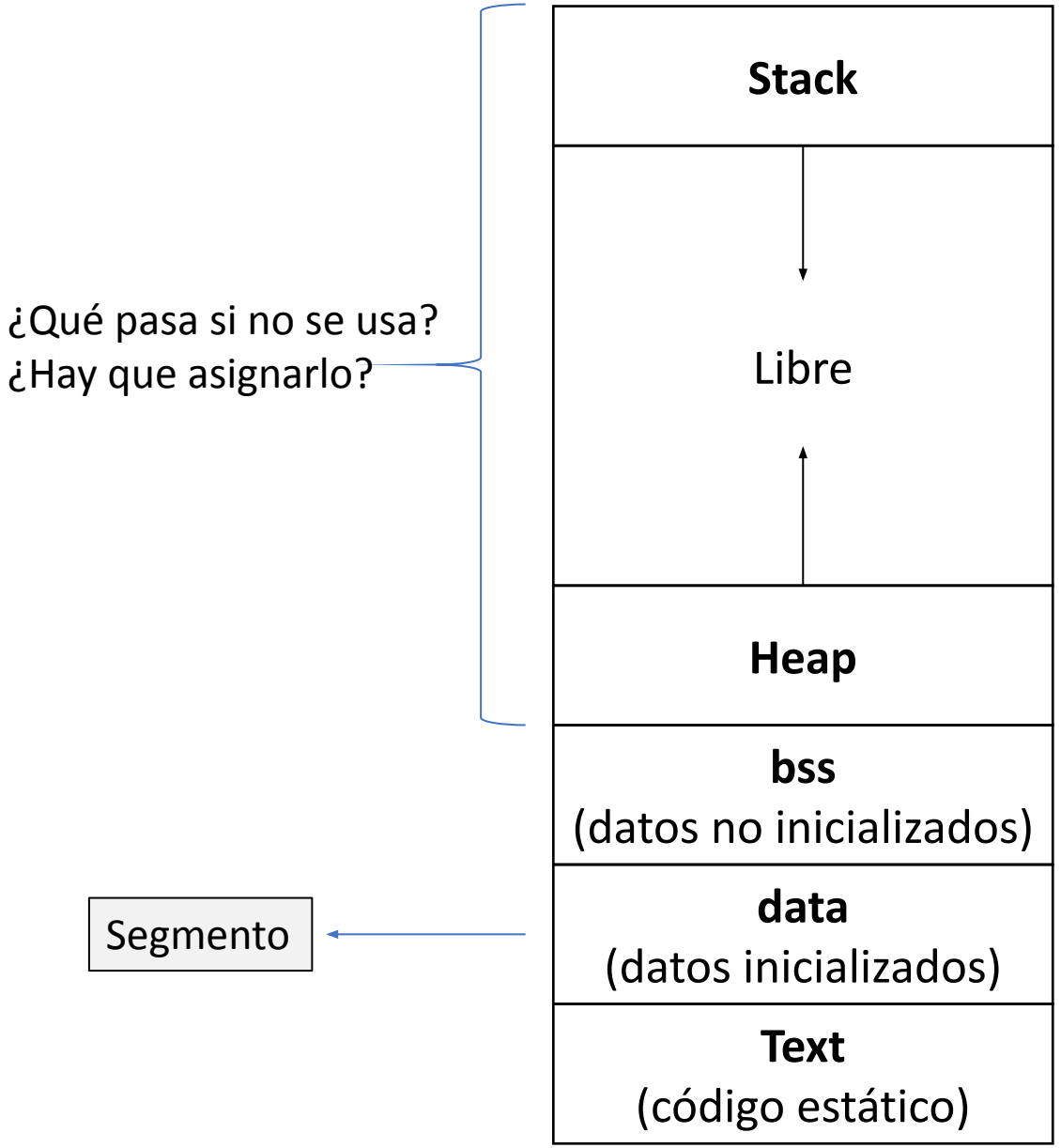
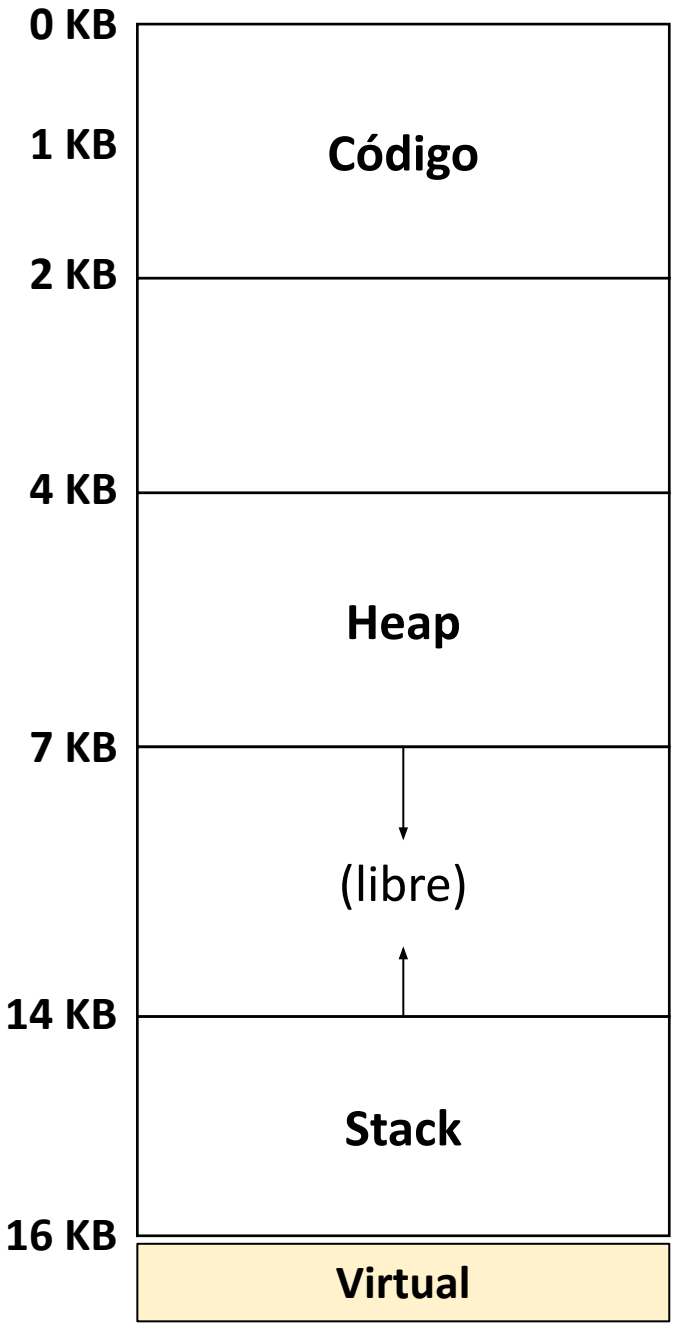
Segmentación

Adaptación (ver referencias al final)

Segmentación

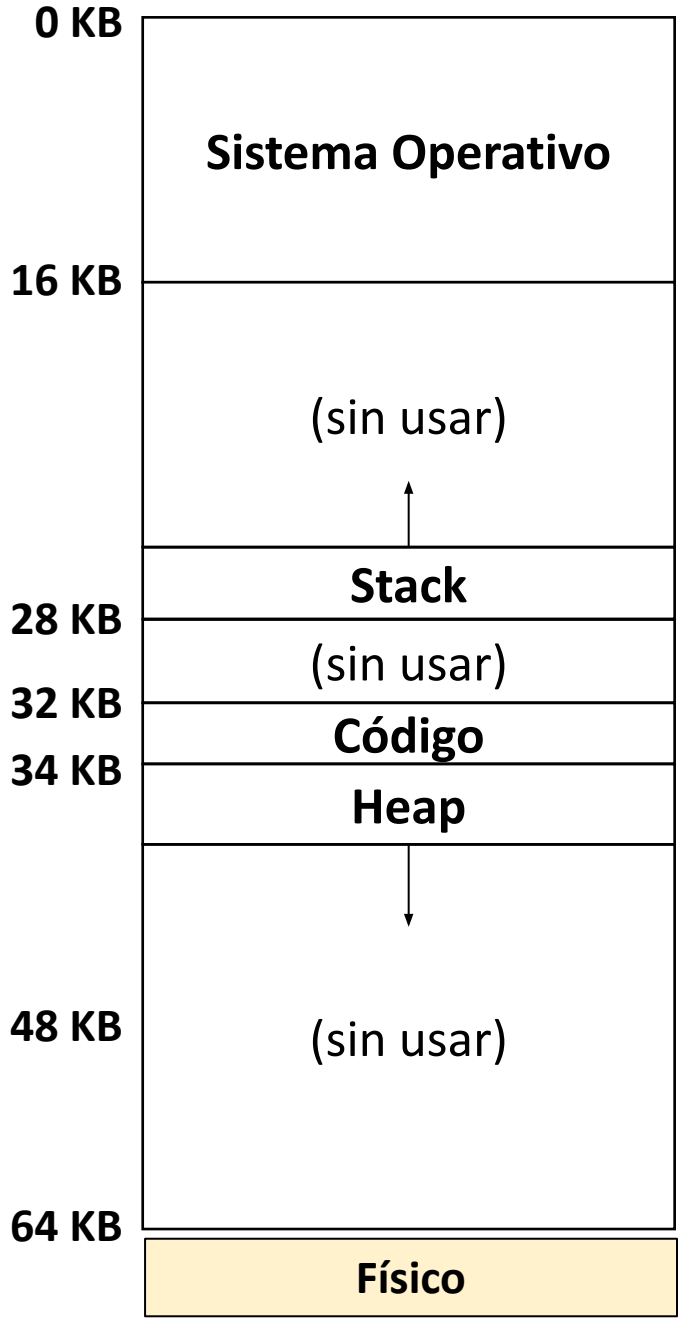
- Tener **más de un registro base y un registro límite** por segmento lógico del espacio de direccionamiento de un proceso.
- Ubicar cada segmento en diferentes partes de la memoria física
  - Evitar asignar espacio no usado. P. Ej.: Heap y Stack
- No se hace asignación contigua de todo la imagen del proceso.

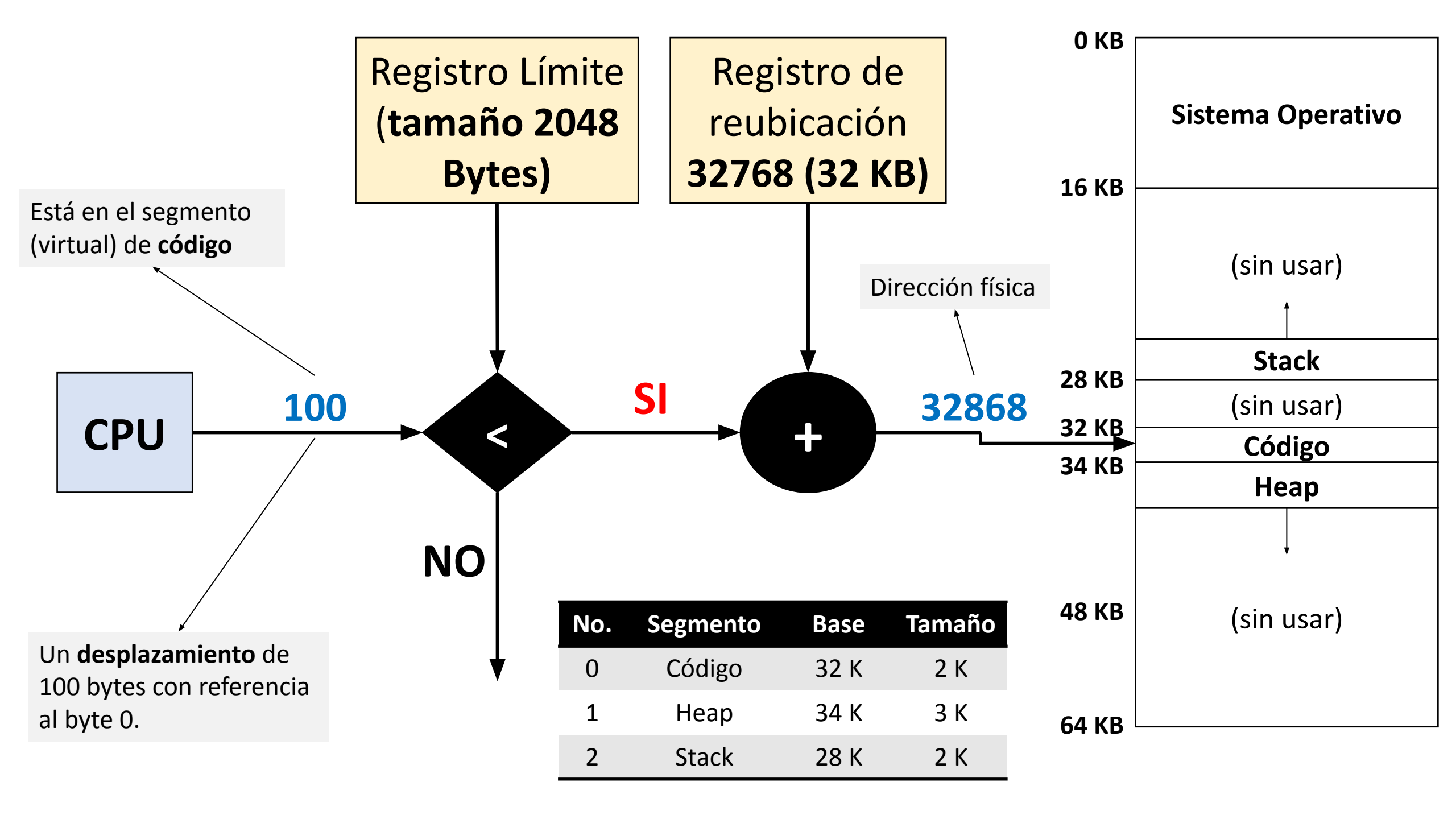




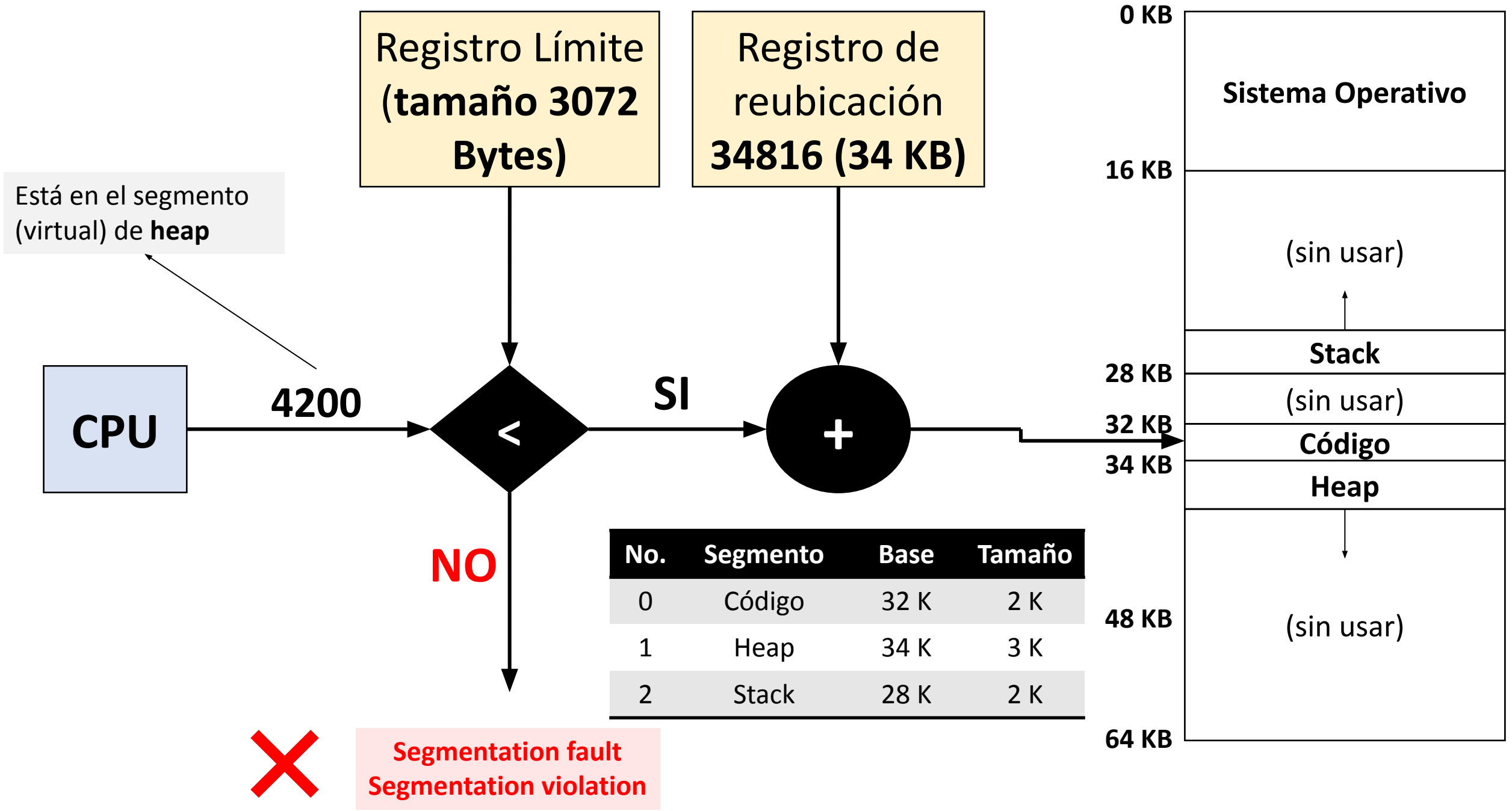
No.	Segmento	Base	Tamaño
0	Código	32 K	2 K
1	Heap	34 K	3 K
2	Stack	28 K	2 K

- Espacio lógico de direccionamiento (izquierda) asignado en memoria física (derecha)
- Cada segmento independientemente en memoria física





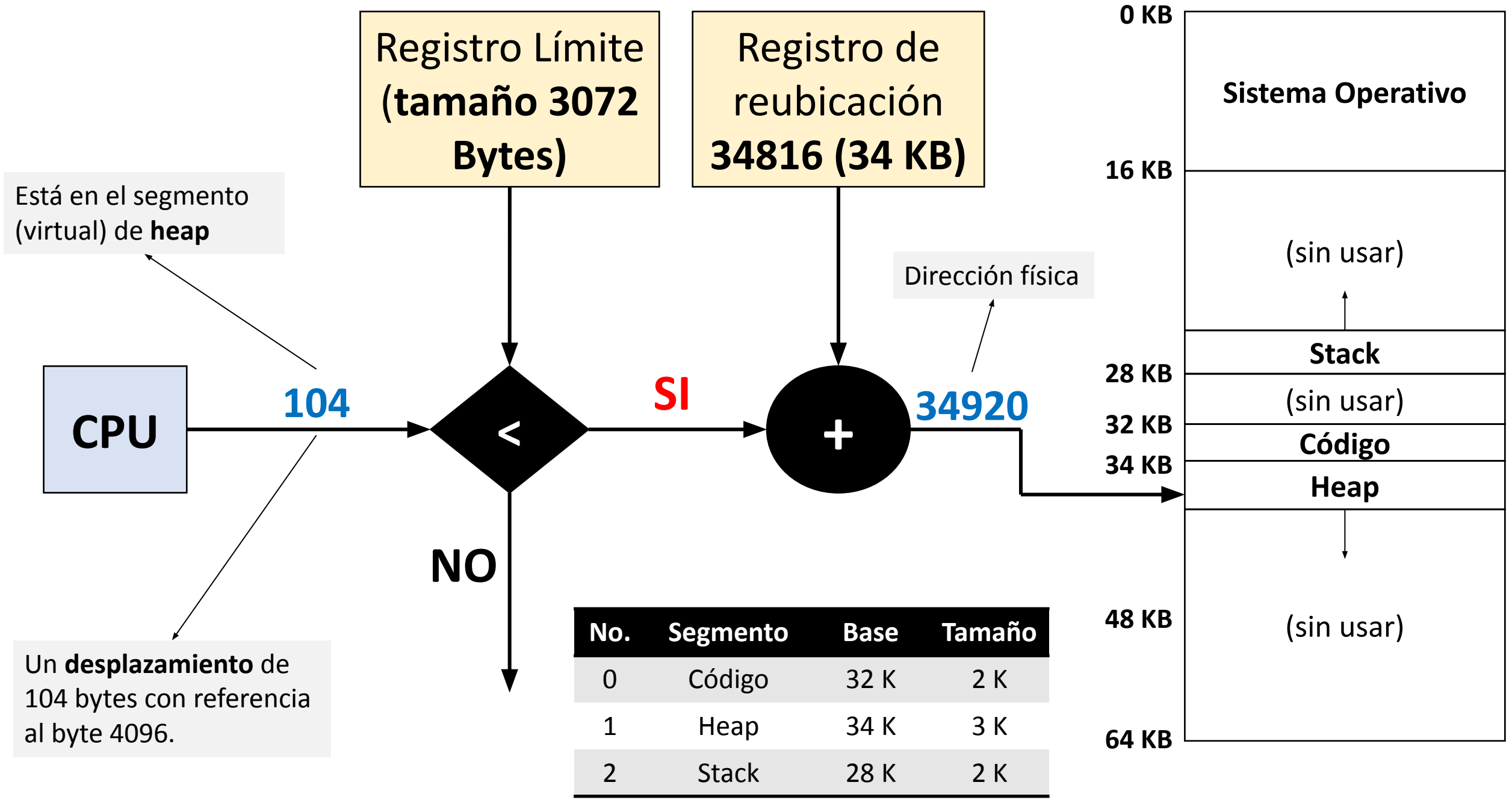
No.	Segmento	Base	Tamaño
0	Código	32 K	2 K
1	Heap	34 K	3 K
2	Stack	28 K	2 K



No.	Segmento	Base	Tamaño
0	Código	32 K	2 K
1	Heap	34 K	3 K
2	Stack	28 K	2 K

Traducción del direccionamiento

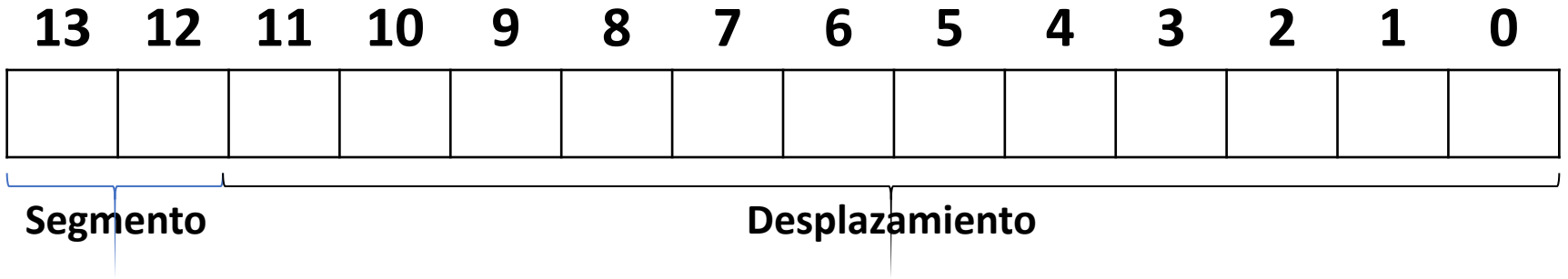
- En este último caso hay extraer primero el desplazamiento.
  - ¿A qué bytes se está haciendo referencia en este segmento?
- Cálculo del desplazamiento
  - 4200 (dirección virtual de referencia) – 4096 (inicio del segmento) $= \mathbf{104}$



No.	Segmento	Base	Tamaño
0	Código	32 K	2 K
1	Heap	34 K	3 K
2	Stack	28 K	2 K

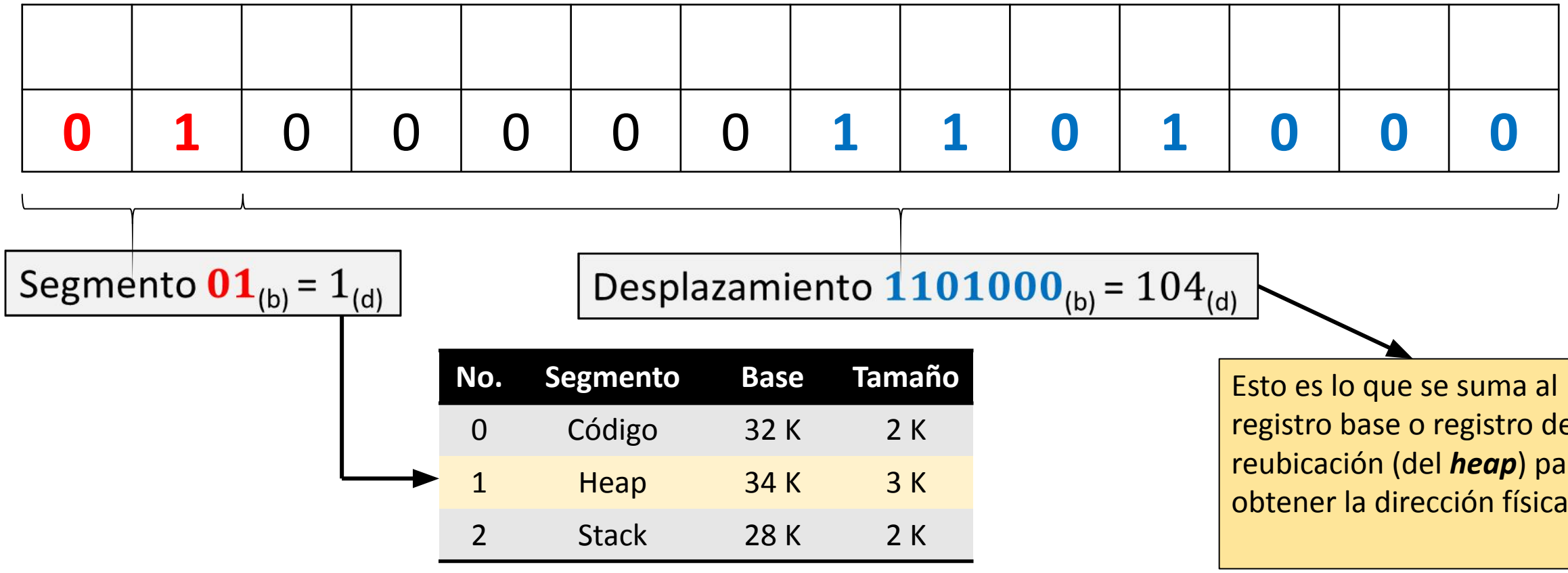
Traducción del direccionamiento

- Espacio virtual de direccionamiento (16 KB): 16384 Bytes
- Tamaño en bits de las direcciones: $2^{14} = 16384$
  - Se necesitan 14 bits para direccionar un espacio de 16KB.
- Se tienen tres segmentos: código, *heap* y *stack*
- De los 14 bits de la dirección se usarán 2 bits (los más significativos) para indicar el segmento.
  - $2^2 = 4$ Se necesitan al menos dos bits para direccionar tres segmentos.



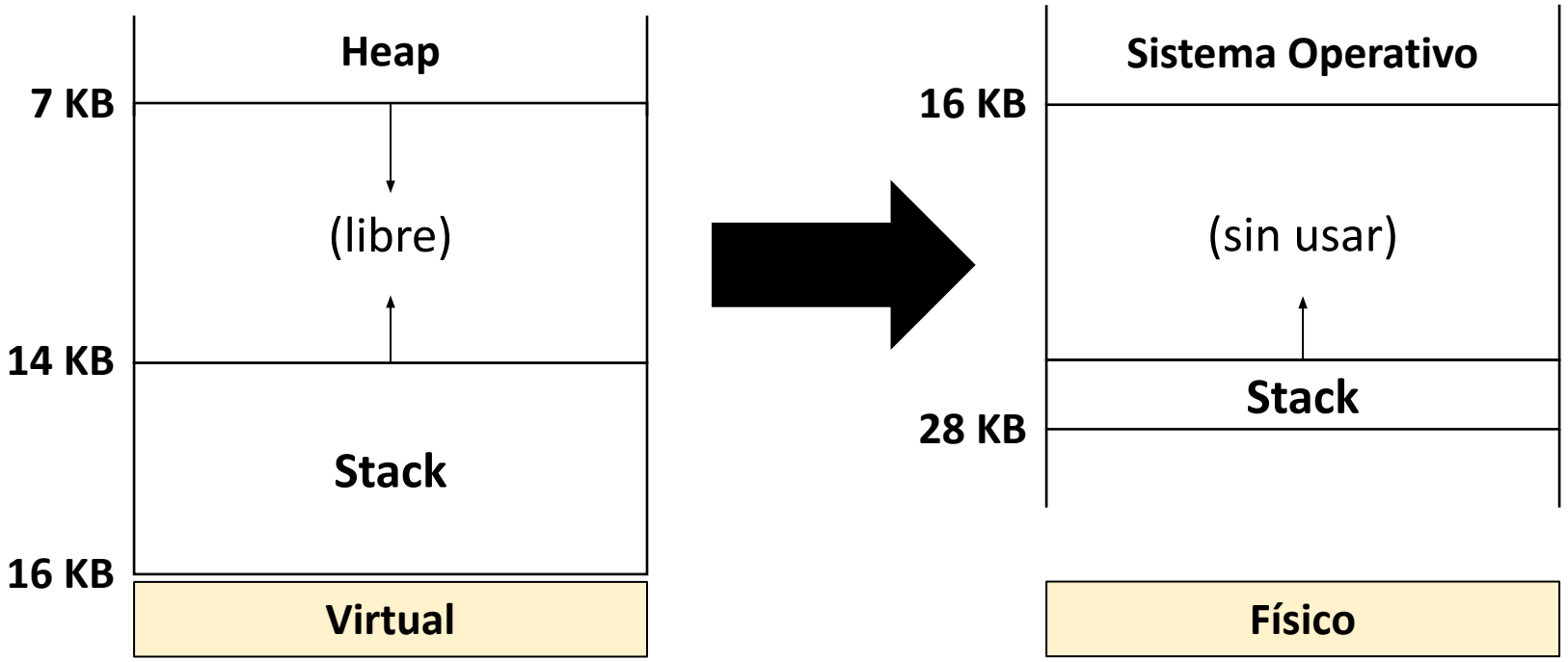
Traducción del direccionamiento

- Dirección virtual: 4200



No.	Segmento	Base	Tamaño
0	Código	32 K	2 K
1	Heap	34 K	3 K
2	Stack	28 K	2 K

¿Qué pasa con el segmento *stack*?



- El segmento *stack* crece en **hacia atrás**
- Crece desde direcciones altas hacia direcciones bajas
- Esto es así en la práctica

¿Qué pasa con el segmento *stack*?

- Se necesita soporte adicional del hardware para indicar esta situación

Segmento	Base	Tamaño (máx. 4K)	Crece positivamente
Código$_{00}$	32 K	2K	1
Heap$_{01}$	34 K	3K	1
Stack$_{11}$	28 K	2K	0

¿Qué pasa con el *stack*?

- Dirección virtual (15 KB) = 15360

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

Segmento $\mathbf{11}_{(b)} = 3_{(d)}$

Desplazamiento $\mathbf{11000000000}_{(b)} = 3072_{(d)} = 3$ KB

Segmento	Base	Tamaño (máx. 4K)	Crece positivamente
Código_{00}	32 K	2 K	1
Heap_{01}	34 K	3 K	1
Stack_{11}	28 K	2 K	0

¿Qué pasa con el *stack*?

- Para esta traducción es necesario restar al desplazamiento el valor del tamaño máximo de segmento.
  - $\mathbf{3\ KB - 4\ KB = -1\ KB}$ (desplazamiento negativo)
- El resultado anterior se le suma al registro base (registro de reubicación) del *stack*
  - $\mathbf{28\ KB + (-1\ KB) = 27\ KB}$
- El registro límite valida que el valor absoluto del desplazamiento sea menor o igual al tamaño del segmento
  - $\mathbf{|-1KB| \leq 2KB}$ (en ese caso **2KB** es el tamaño del stack en memoria física)

Segmentos compartidos

- Para compartir segmentos se necesita soporte adicional del hardware
  - P. Ej.: shared objects, DLL's

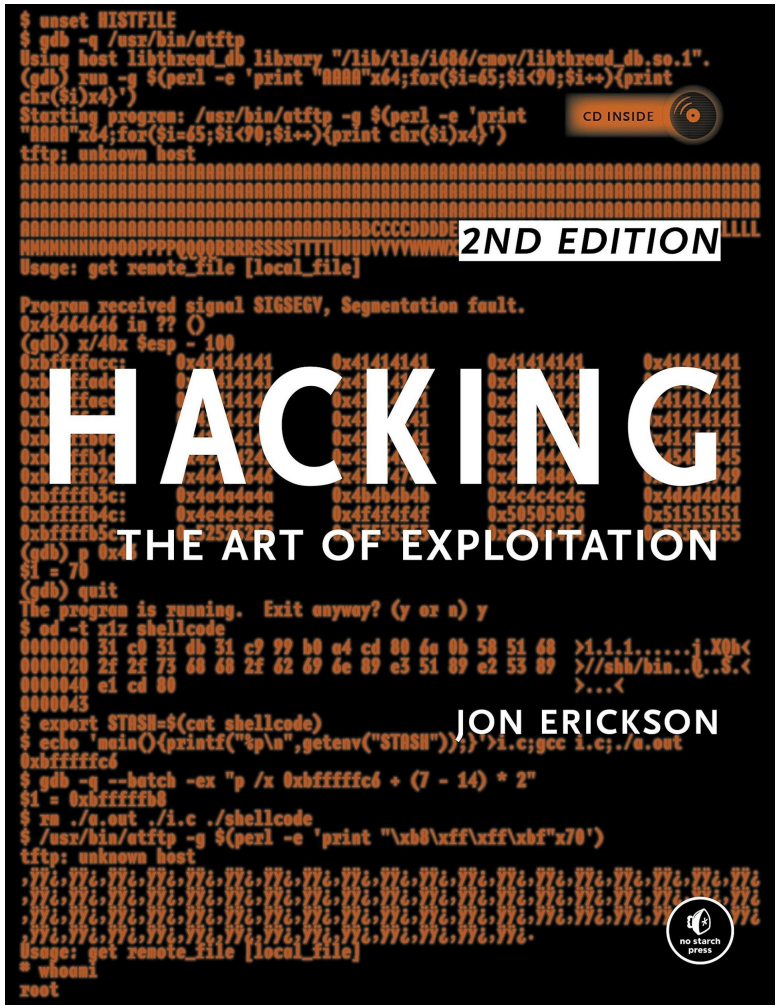
Segmento	Base	Tamaño (máx. 4K)	Crece positivamente	Protección
$Código_{00}$	32 K	2K	1	Read-Execute
$Heap_{01}$	34 K	3K	1	Read-Write
$Stack_{11}$	28 K	2K	0	Read-Write

Soporte desde el S.O

- ¿Qué hacer en el cambio de contexto?
  - Guardar registros por proceso
  - ¿Qué hacer frente a demanda dinámica de memoria? P. Ej.: `malloc()`
  - Encontrar espacio libre para asignar
  - Actualizar registros en (tabla de hardware)
  - Rechazar solicitudes cuando no hay suficiente memoria disponible
- Administración del espacio libre
  - Encontrar espacio libre para los segmentos de un nuevo proceso
  - Segmentos de diferente tamaño
  - Problemas de fragmentación externa

Lecturas recomendadas

- Smashing The Stack For Fun And Profit
  - Aleph One, Underground.org
  - Buffer Overflow, Stack Overflow
- File Infection Techniques
  - Bill Harrison
  - CS 4440/7440 Malware Analysis & Defense
- Hacking The Art Of Exploitation
  - Jon Erickson



Referencias

- Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. (2018). Segmentation. In *Operating Systems. Three Easy Pieces* (pp. 1–12). Arpaci-Dusseau Books.