

Introducción a los Sistemas Operativos

Administración de Memoria - II



- ✓ Versión: Agosto 2013
- ✓ Palabras Claves: Procesos, Espacio de Direcciones, Memoria, Seguridad, Paginación, Segmentación

Algunas diapositivas han sido extraídas de las ofrecidas para docentes desde el libro de Stallings (Sistemas Operativos) y el de Silberschatz (Operating Systems Concepts). También se incluyen diapositivas cedidas por Microsoft S.A.

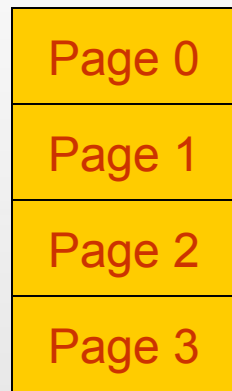


Paginación

- ✓ La memoria es dividida lógicamente en pequeños trozos de igual tamaño → Marcos
- ✓ El espacio de direcciones de cada proceso es dividido en trozos de igual tamaño que los marcos → Paginas
- ✓ El SO mantiene una tabla de paginas por cada proceso.
 - ✓ Contiene el marco en la que esta situada cada pagina.
 - ✓ La dirección lógica consiste en un numero de pagina y un desplazamiento dentro de la misma.



Paginación - Ejemplo

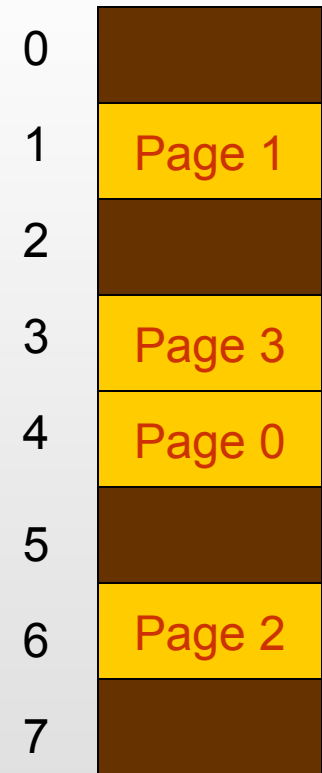


logical
memory

0	4
1	1
2	6
3	3

page
table

frame
number



physical
memory



Paginación – Ejemplo II

Frame number	Main memory
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

(a) Fifteen Available Frames

Frame number	Main memory
0	A.0
1	A.1
2	A.2
3	A.3
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

(b) Load Process A

Frame number	Main memory
0	A.0
1	A.1
2	A.2
3	A.3
4	B.0
5	B.1
6	B.2
7	
8	
9	
10	
11	
12	
13	
14	

(c) Load Process B



Paginación – Ejemplo II (cont.)

Main memory	
0	A.0
1	A.1
2	A.2
3	A.3
4	B.0
5	B.1
6	B.2
7	C.0
8	C.1
9	C.2
10	C.3
11	
12	
13	
14	

(d) Load Process C

Main memory	
0	A.0
1	A.1
2	A.2
3	A.3
4	
5	
6	
7	C.0
8	C.1
9	C.2
10	C.3
11	
12	
13	
14	

(e) Swap out B

Main memory	
0	A.0
1	A.1
2	A.2
3	A.3
4	D.0
5	D.1
6	D.2
7	C.0
8	C.1
9	C.2
10	C.3
11	D.3
12	D.4
13	
14	

(f) Load Process D



Paginación – Ejemplo II (cont.)

0	0
1	1
2	2
3	3

Process A
page table

0	—
1	—
2	—

Process B
page table

0	7
1	8
2	9
3	10

Process C
page table

0	4
1	5
2	6
3	11
4	12

Process D
page table

13
14

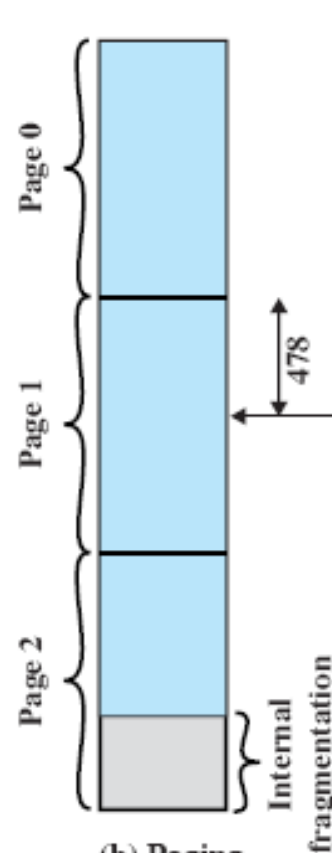
Free frame
list



Paginación – Direcciones Lógicas

Logical address =
Page# = 1, Offset = 478

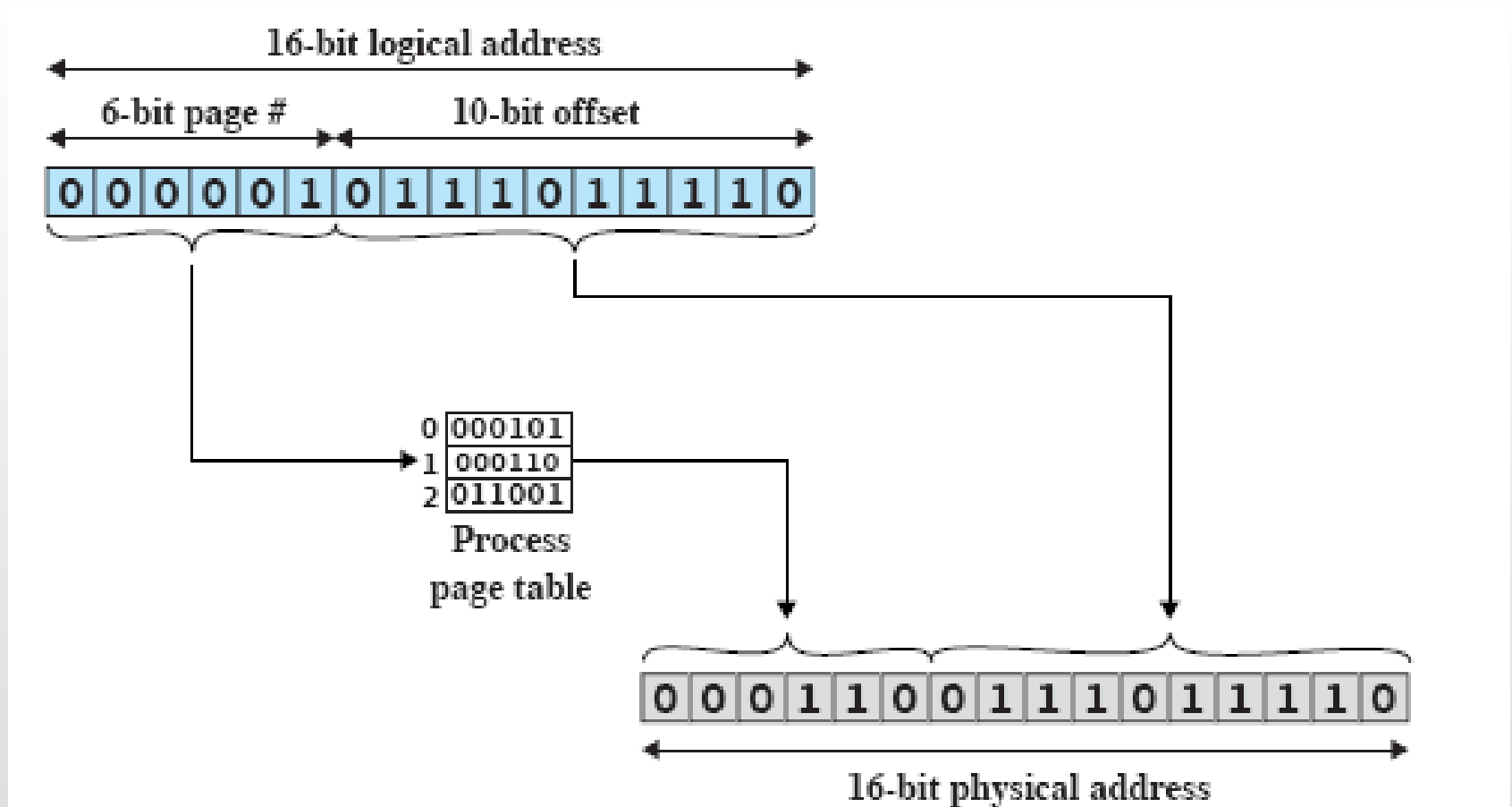
0000010111011110



(b) Paging
(page size = 1K)



Traducción de direcciones

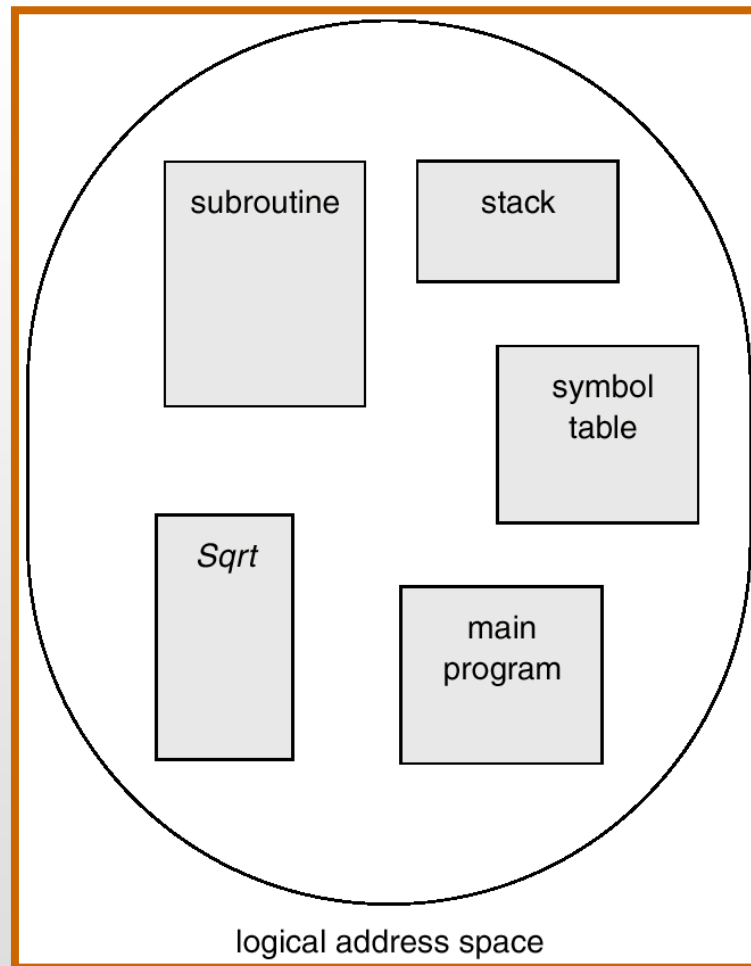


Segmentación

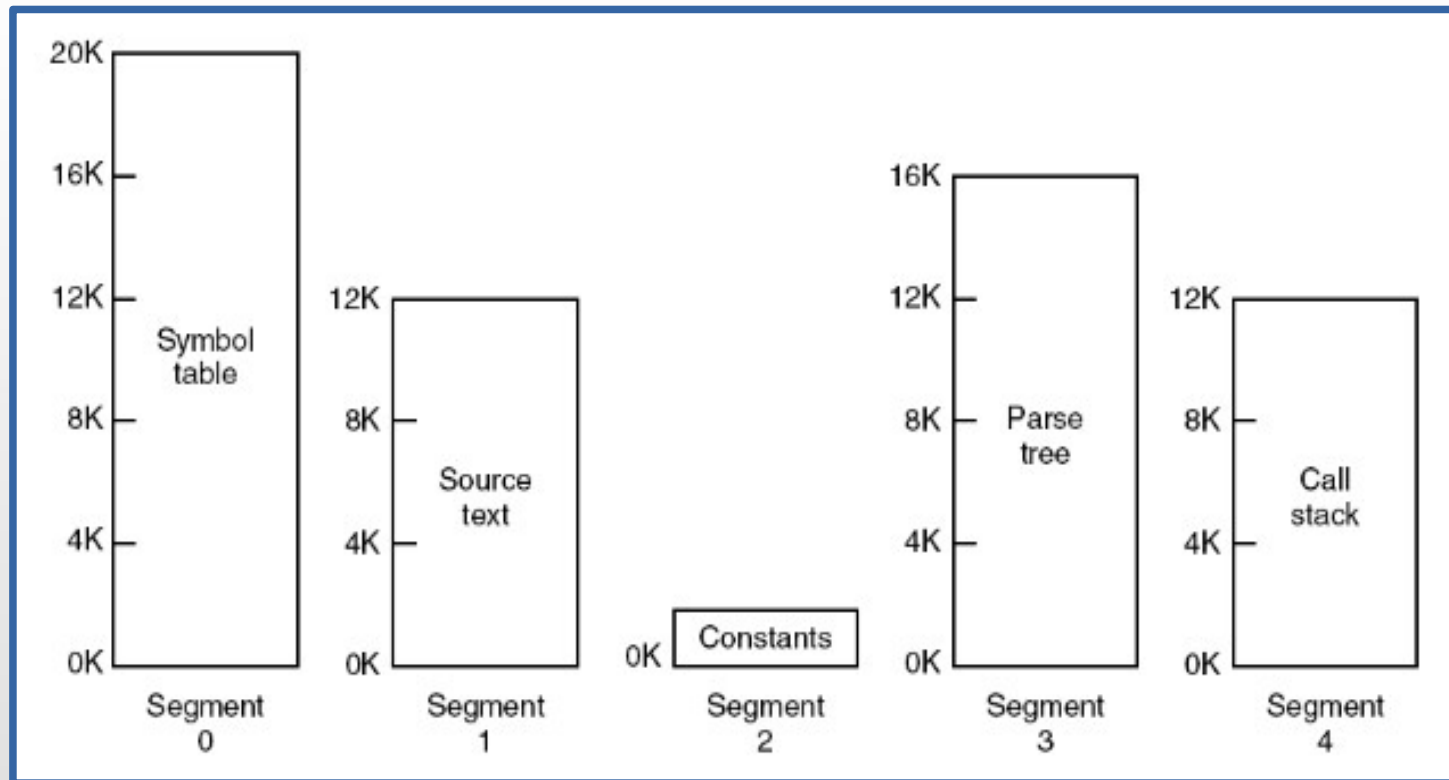
- ✓ Esquema que soporta el “punto de vista de un usuario”
- ✓ Un programa es una colección de segmentos. Un segmento es una unidad lógica como:
 - ✓ Programa Principal, Procedimientos y Funciones, variables locales y globales, stack, etc.
- ✓ Similar a particiones dinámicas.



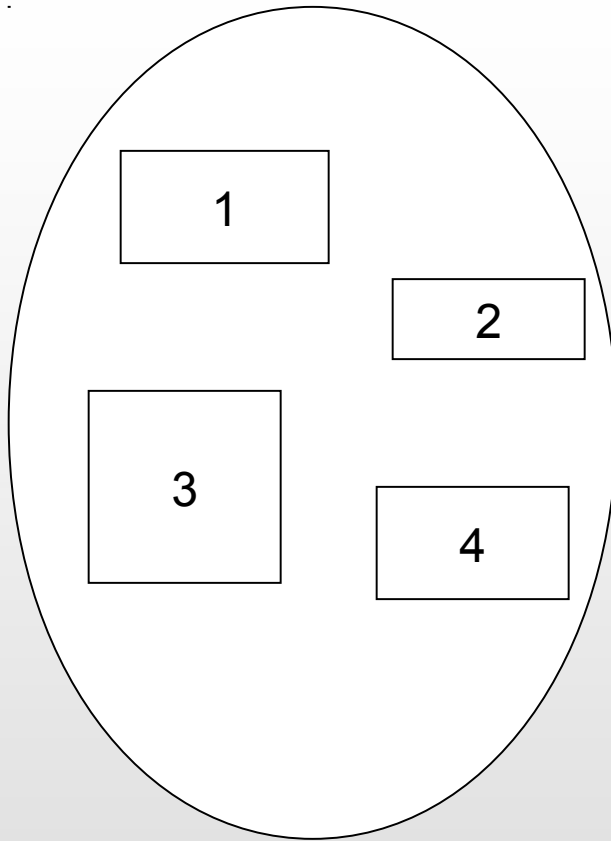
Programa desde la visión del usuario



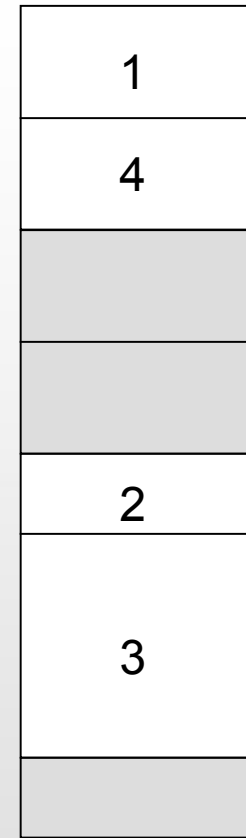
Espacios de direcciones de cada segmento



Segmentación - Vista Lógica



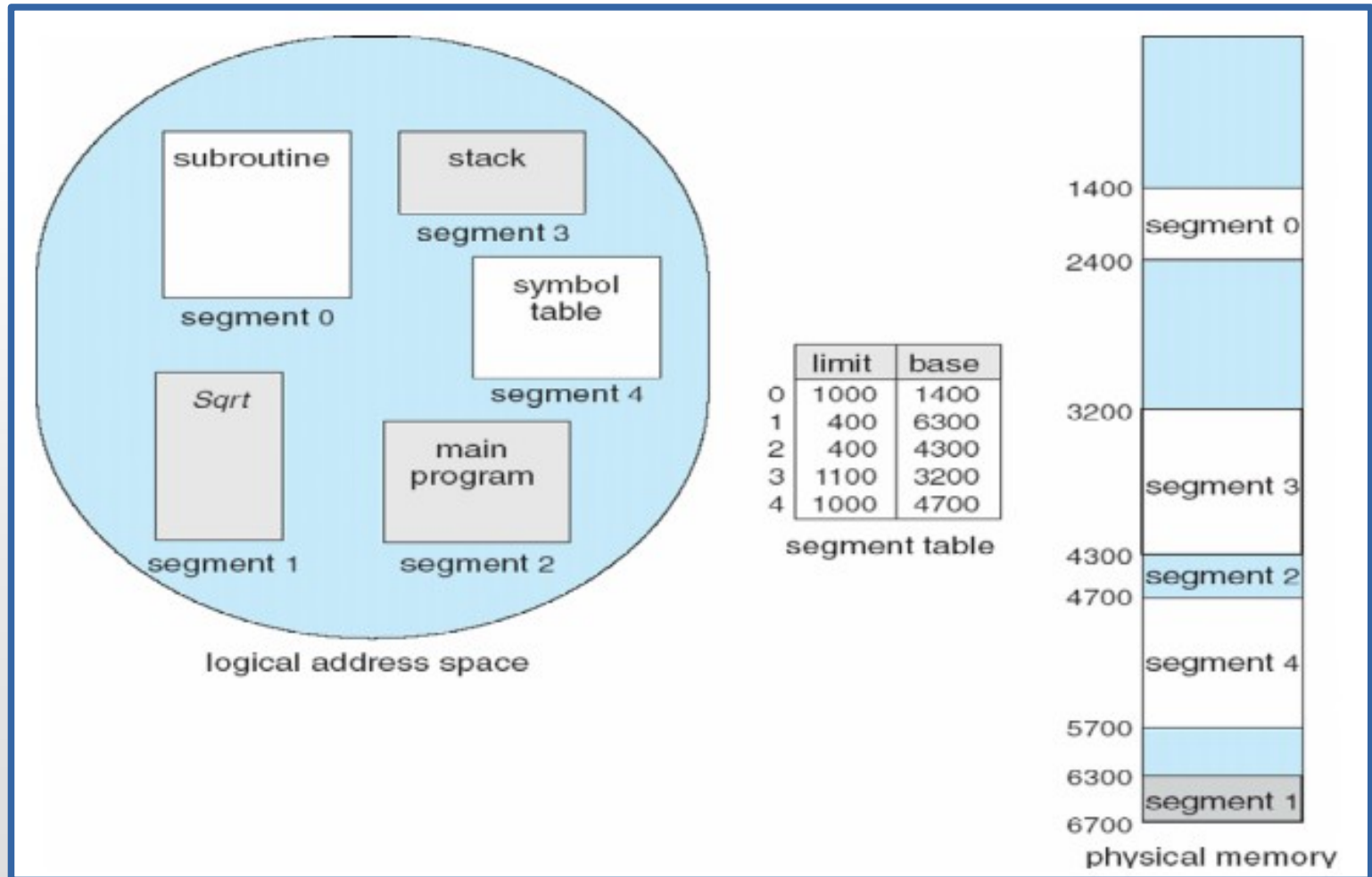
user space



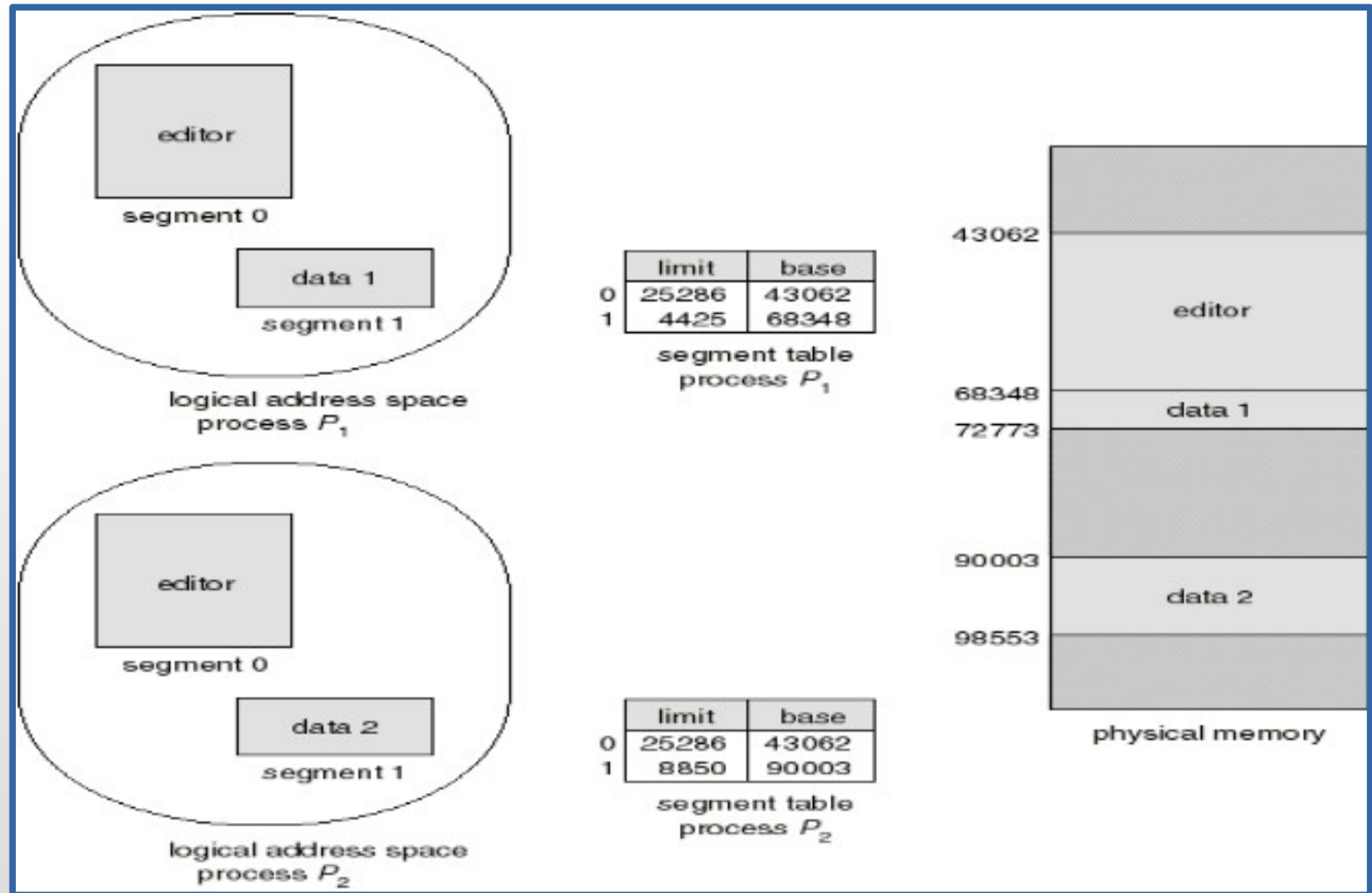
physical memory space



Ejemplo de Segmentación



Compartición de Segmentos



Segmentación (cont.)

- ☑ Todos los segmentos de un programa pueden no tener el mismo tamaño (código, datos, rutinas).
- ☑ Las direcciones Lógicas consisten en 2 partes:
 - ✓ Selector de Segmento
 - ✓ Desplazamiento dentro del segmento



Segmentación (cont.) - Arquitectura

✓ *Tabla de Segmentos*

✓ *Permite mapear la dirección lógica en física.
Cada entrada contiene:*

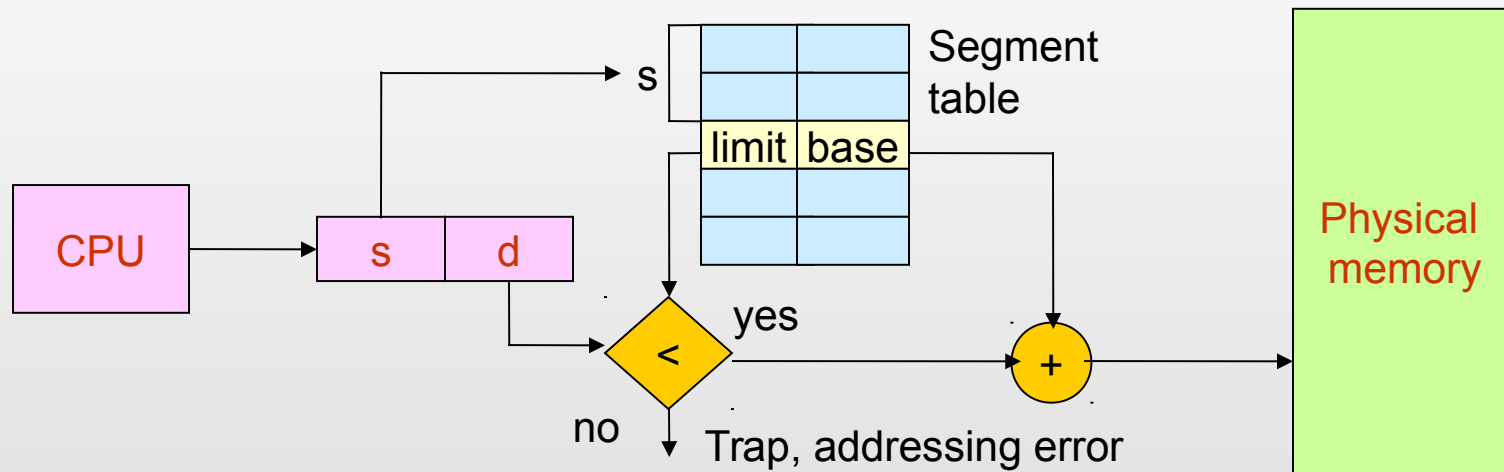
- ♦ *Base: Dirección física de comienzo del segmento*
- ♦ *Limit: Longitud del Segmento*

✓ *Segment-table base register (STBR):* apunta a la ubicación de la tabla de segmentos.

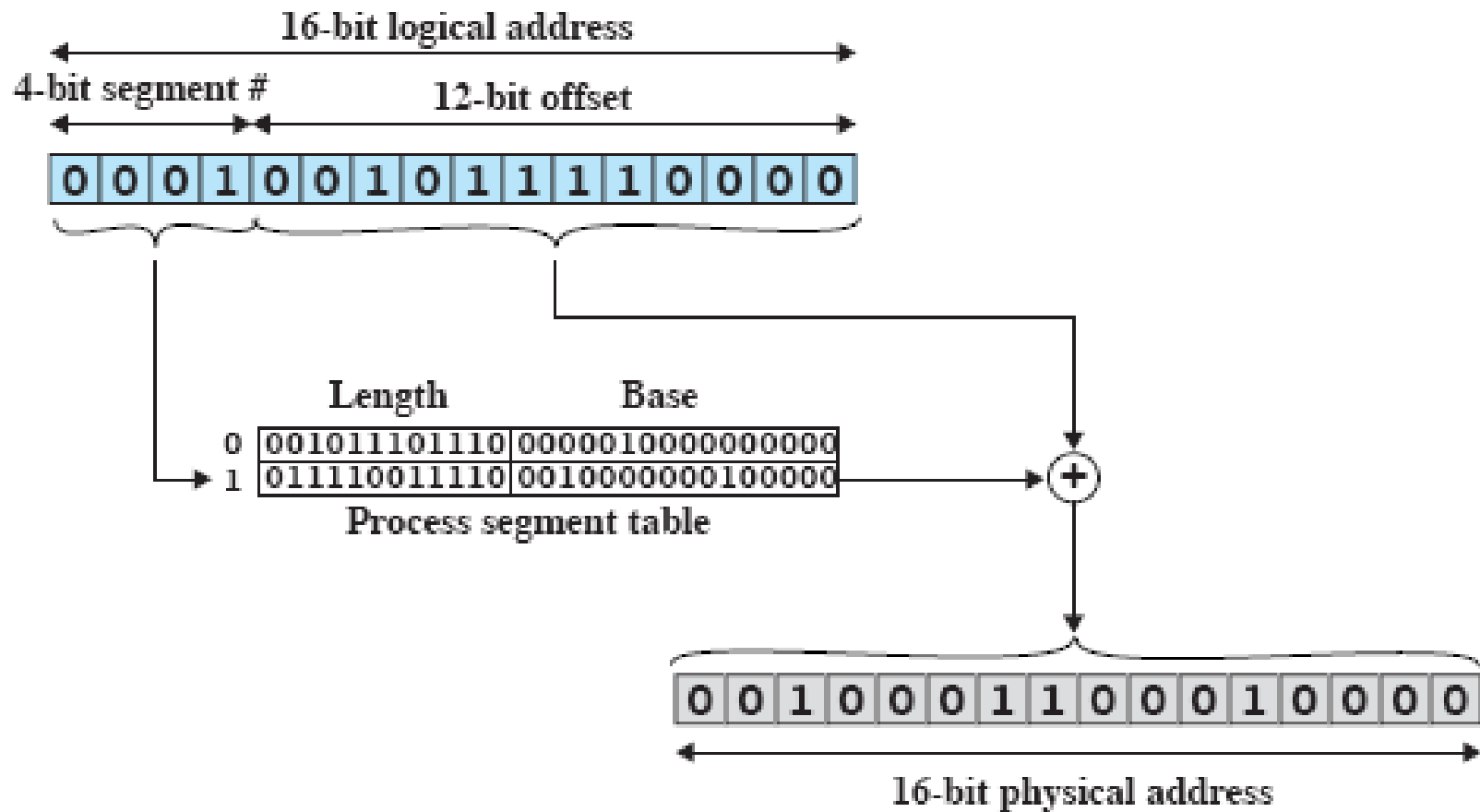
✓ *Segment-table length register (STLR) :* cantidad de segmentos de un programa



Segmentación (cont.)



Segmentación - Direcciones (cont.)



(b) Segmentation

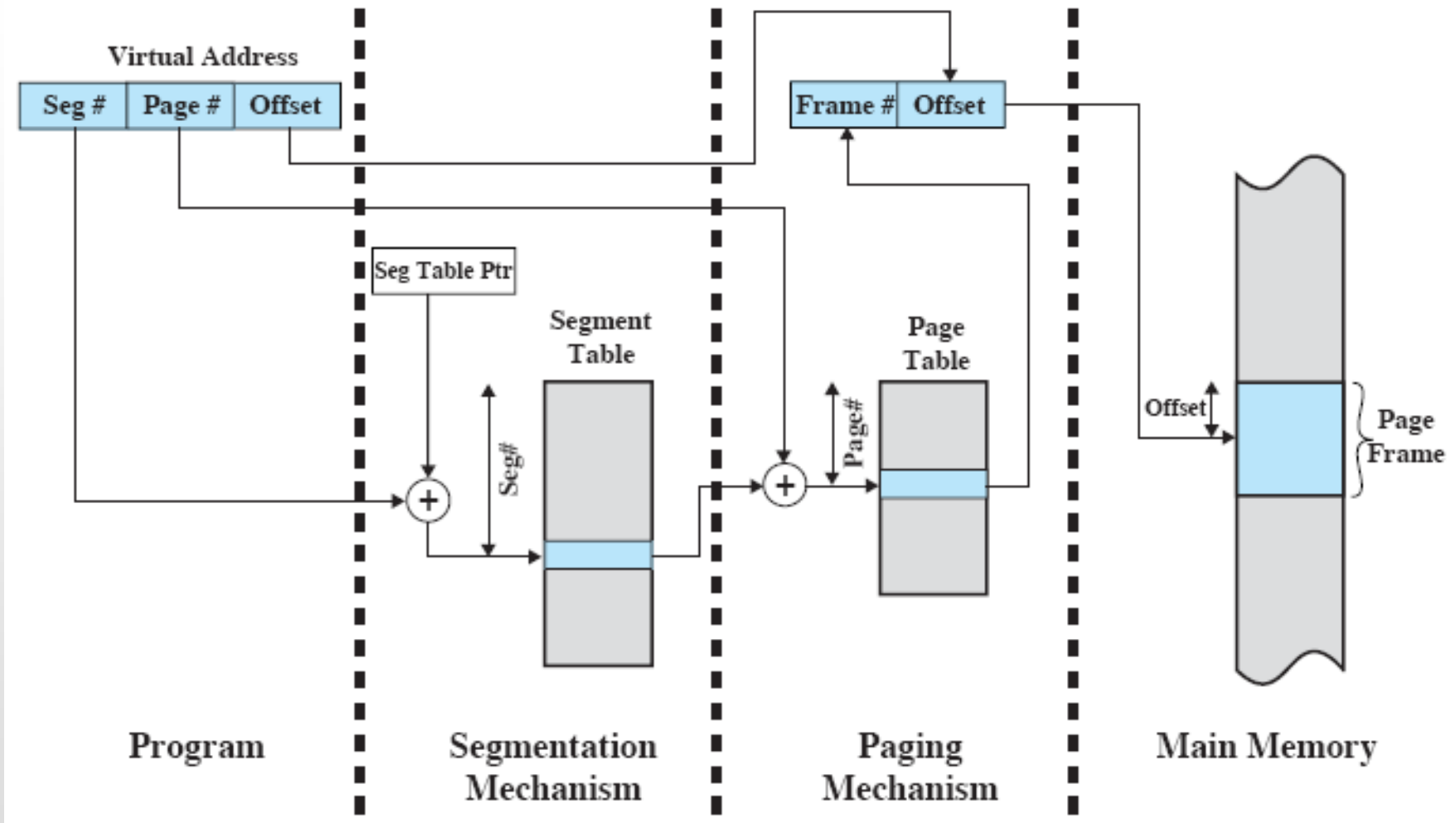


Segmentación Paginada

- ☑ La paginación
 - ✓ Transparente al programador
 - ✓ Elimina Fragmentación externa.
- ☑ Segmentación
 - ✓ Es visible al programador
 - ✓ Facilita modularidad, estructuras de datos grandes y da mejor soporte a la compartición y protección
- ☑ Cada segmento es dividido en paginas de tamaño fijo.



Segmentación Paginada (cont.)



Intel 386

The Intel 386 uses segmentation with paging for memory management with a two-level paging scheme.

