Sistemas Operativos Avanzados

Tema 4. **Administración de la Memoria Virtual**

Profesor:

Dr. José Octavio Gutiérrez García

octavio.gutierrez@itam.mx

Memoria Stack - Pila

- Región especial de la memoria de un proceso que almacena las variables temporales (y pequeñas).

- Administrada y optimizada por el CPU

- Muy rápida
- Tamaño limitado por el SO
- Las variables no se pueden redimensionar
- Sólo variables locales

Memoria Stack - Pila

```c
#include <stdio.h>

double multiplyByTwo (double input) {
  double twice = input * 2.0;
  return twice;
}

int main (int argc, char *argv[])
{
  int age = 30;
  double salary = 12345.67;
  double myList[3] = {1.2, 2.3, 3.4};

  printf("double your salary is %.3f\n", multiplyByTwo(salary));

  return 0;
}
```

```
double your salary is 24691.340
```

Memoria Heap

- Región de la memoria (mayoritariamente) gestionada por el programador.

- Más libre y grande que la pila.

- Las variables pueden ser globales
- No tan rápida como la pila
- Tamaño “ilimitado”
- Las variables se pueden redimensionar

Memoria Heap

```c
#include <stdio.h>
#include <stdlib.h>

double *multiplyByTwo (double *input) {
  double *twice = malloc(sizeof(double));
  *twice = *input * 2.0;
  return twice;
}

int main (int argc, char *argv[])
{
  int *age = malloc(sizeof(int));
  *age = 30;
  double *salary = malloc(sizeof(double));
  *salary = 12345.67;
  double *myList = malloc(3 * sizeof(double));
  myList[0] = 1.2;
  myList[1] = 2.3;
  myList[2] = 3.4;

  double *twiceSalary = multiplyByTwo(salary);

  printf("double your salary is %.3f\n", *twiceSalary);

  free(age);
  free(salary);
  free(myList);
  free(twiceSalary);

  return 0;
}
```

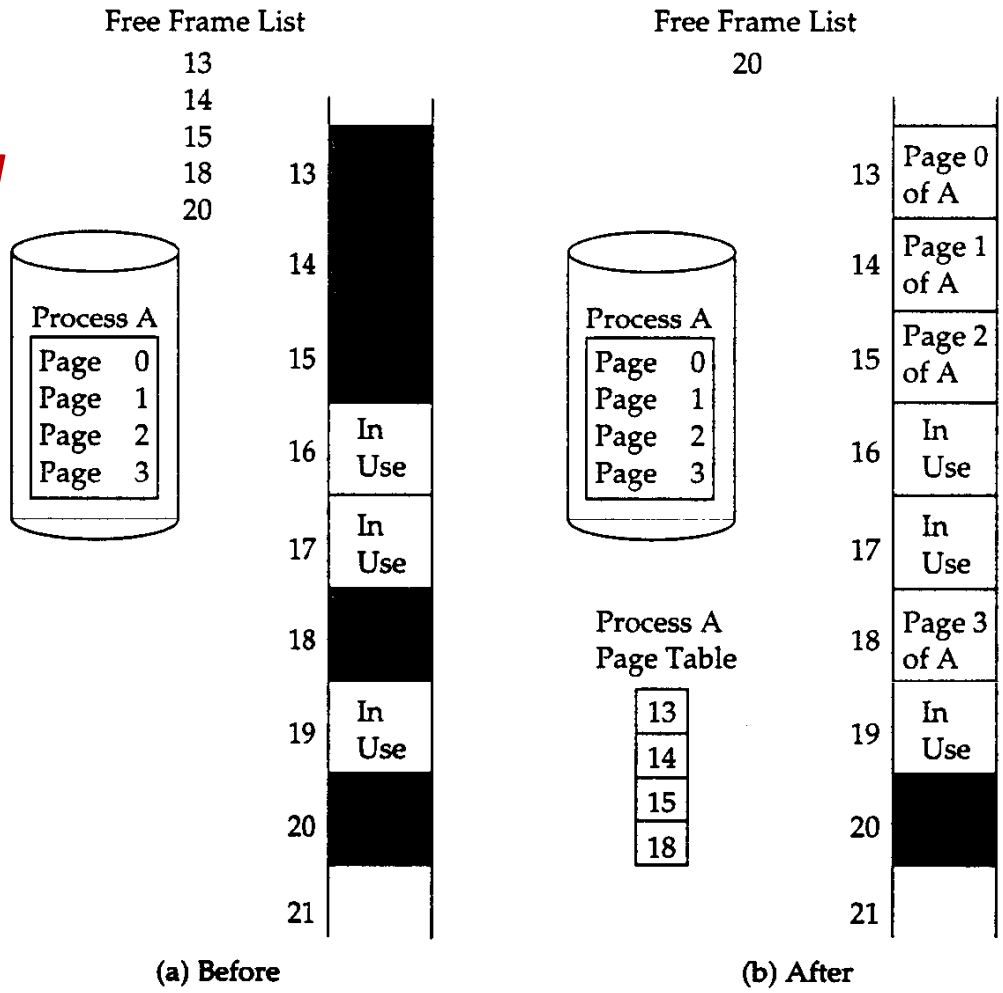
Memoria Virtual

La memoria virtual es un sistema de gestión de memoria que da a una computadora la apariencia de tener más memoria principal de lo que realmente tiene.

Paginación

División de memoria en pequeñas partes llamadas marcos o ***marcos de página***

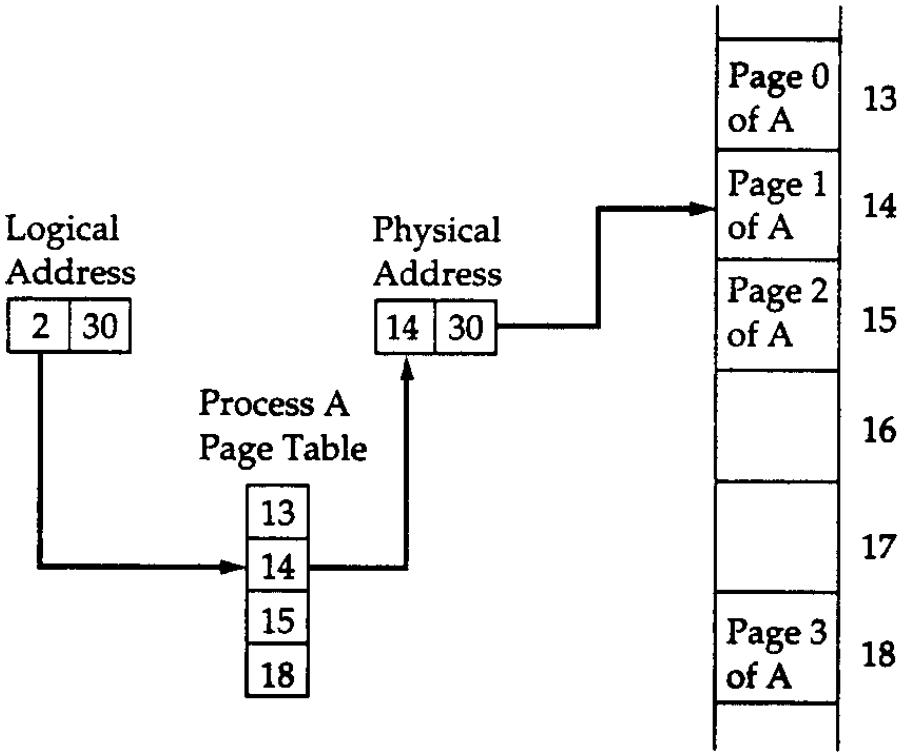
- Cargar un programa en memoria, implica cargar sus páginas.
- Limita el desperdicio de memoria a una fracción de la última página
- Las páginas no necesitan ser contiguas



Paginación

División de memoria en pequeñas partes llamadas marcos o ***marcos de página***

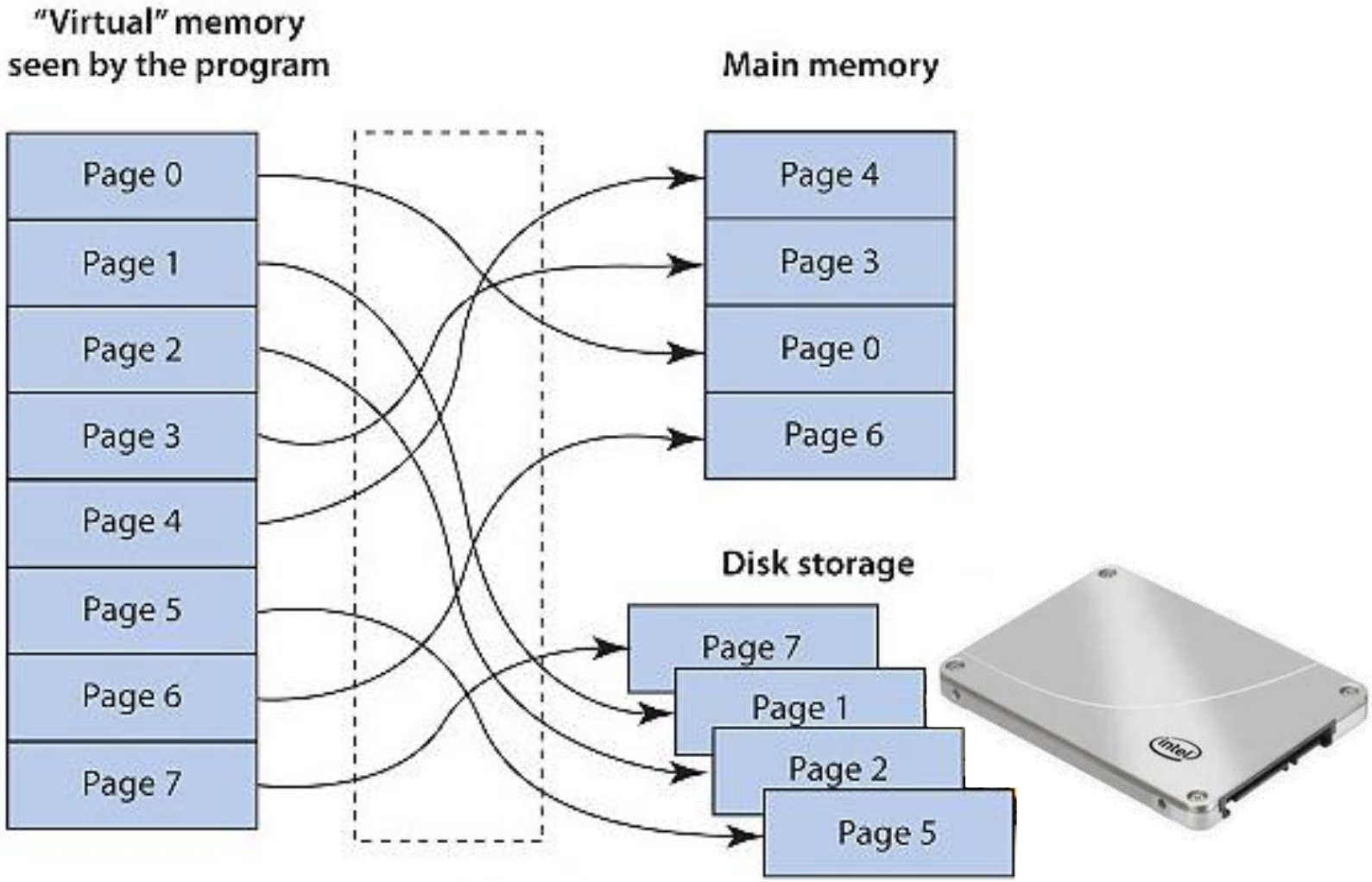
- Cargar un programa en memoria, implica cargar sus páginas.
- Limita el desperdicio de memoria a una fracción de la última página
- Las páginas no necesitan ser contiguas



“Working set”

Colección ***mínima*** de ***páginas*** de un proceso que se deben cargar en la memoria principal para operar de manera eficiente y sin fallos de página innecesarios .

Memoria Virtual



Memory Management Unit (MMU)

Asocia direcciones virtuales a las direcciones físicas

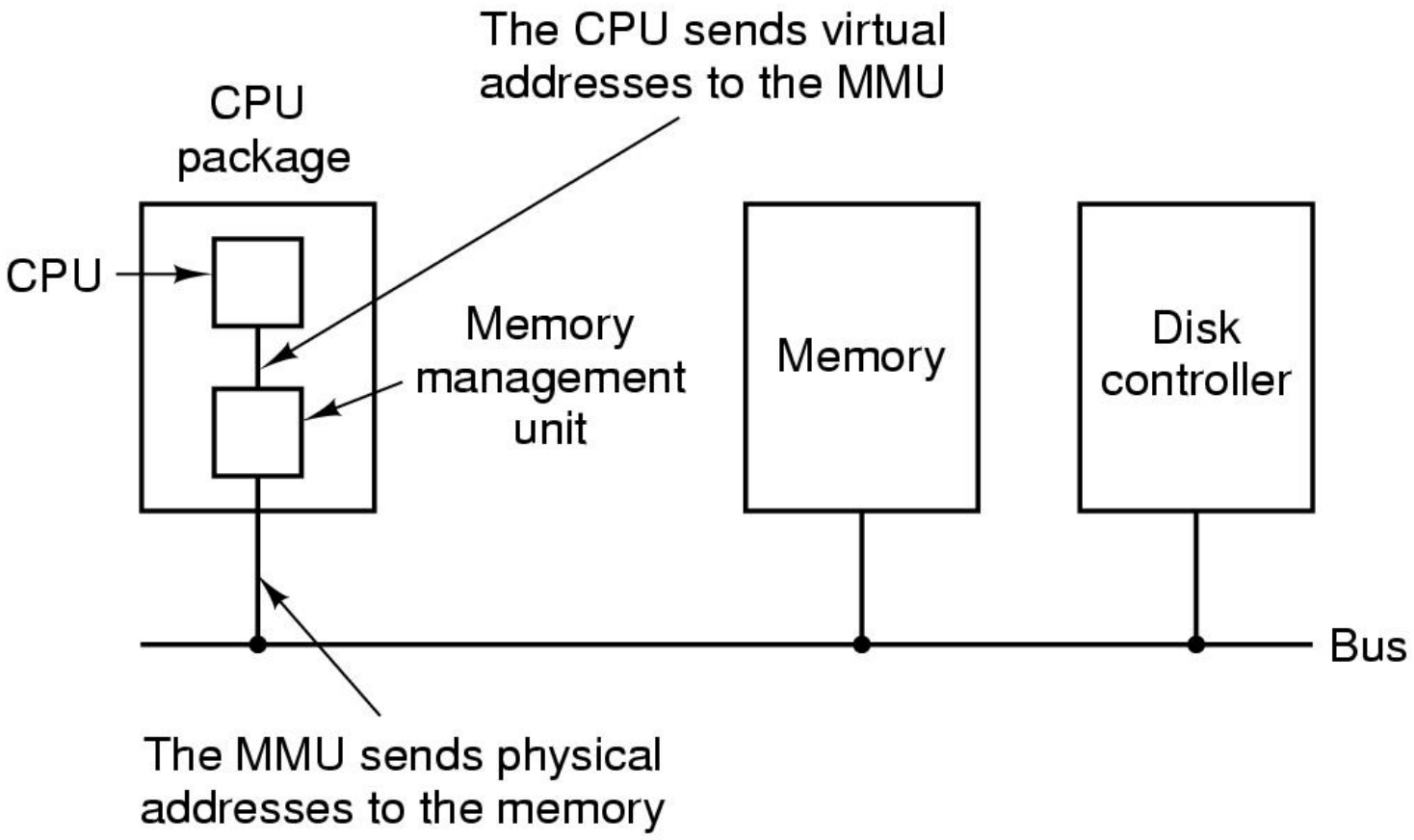
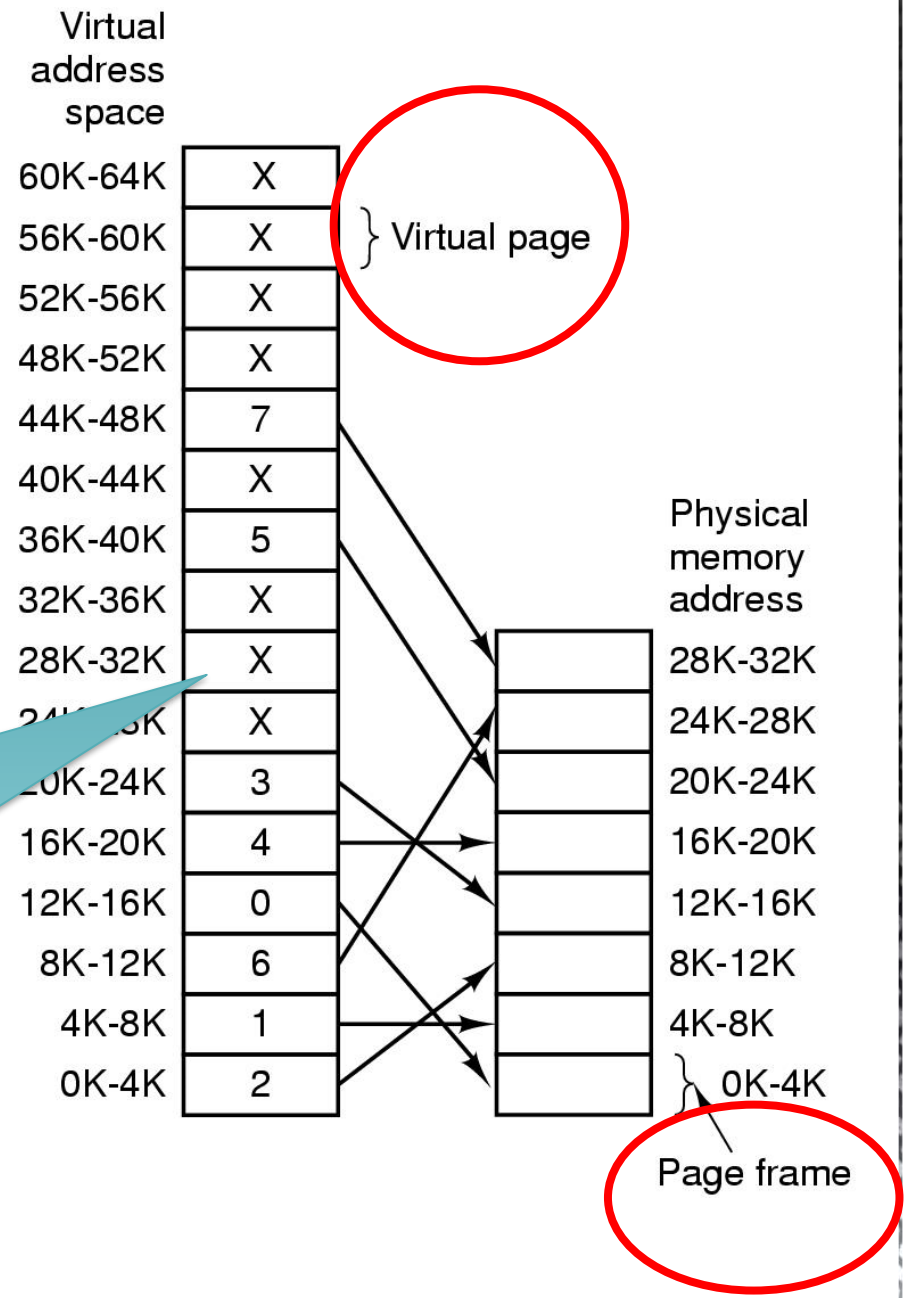


Tabla de páginas

Relación entre las direcciones virtuales y las direcciones de memoria física

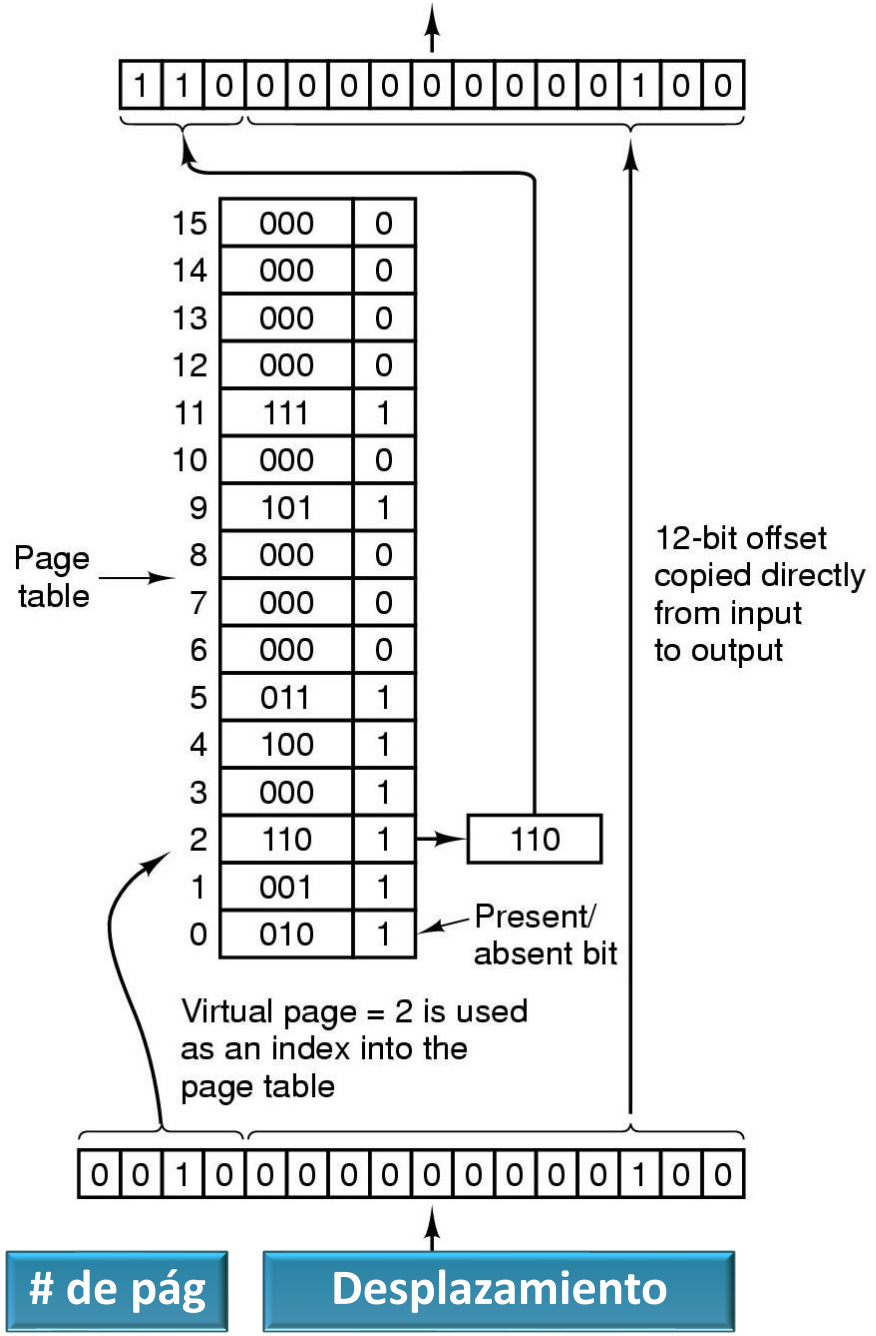


¿Qué pasa si el programa hace referencia a direcciones no asociadas?

Fallo de Página

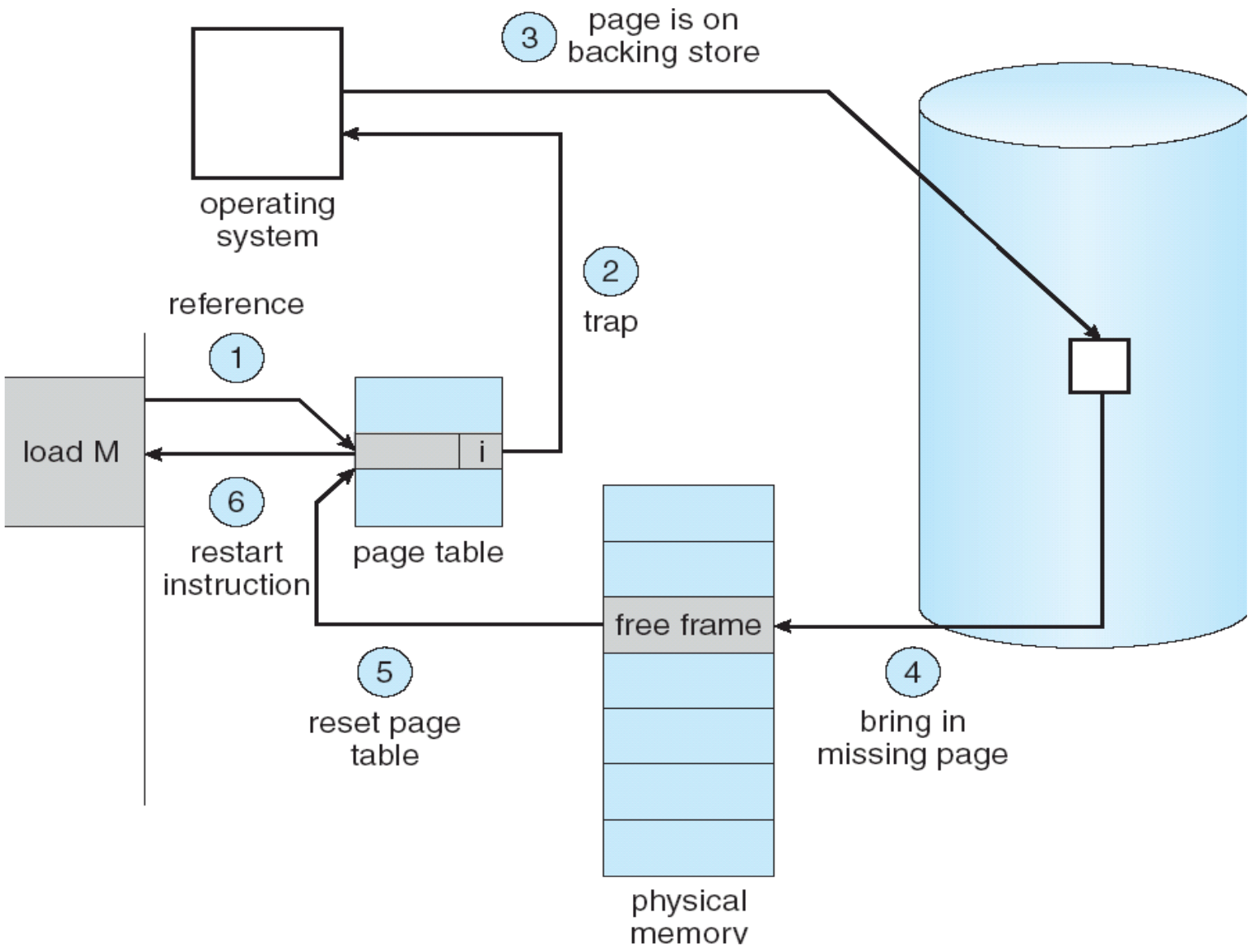
Salida Física

Operación interna de la MMU

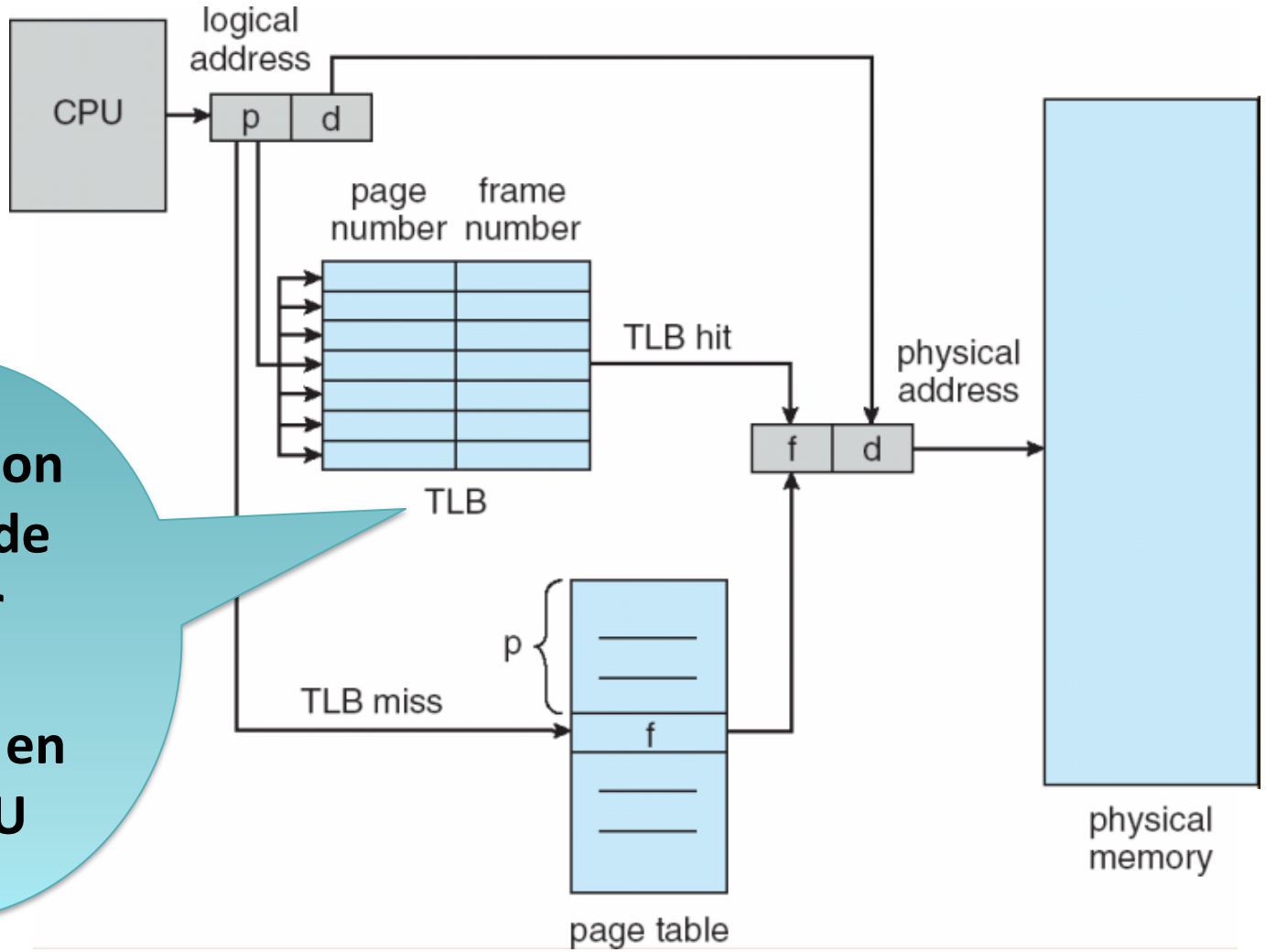


Entrada Virtual

Pasos del manejo de un fallo de página



Memoria Virtual – "Versión 2.0"

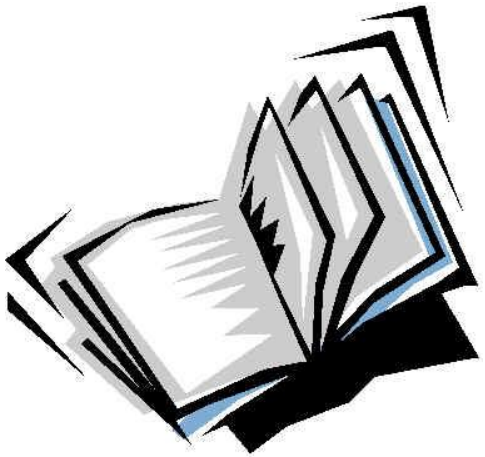


Translation Lookaside Buffer

Se ubica en la MMU

Algoritmos de reemplazo de páginas

1. Óptimo
2. Primera en entrar, primera en salir
3. No usadas recientemente
4. Segunda oportunidad
5. Reloj
6. Menos usadas recientemente



Algoritmo de reemplazo de páginas ÓPTIMO

- ***Idea***

  Seleccionar la página que no va a ser requerida por el mayor tiempo

- ***Problema***

  Irrealizable

  No se puede saber el futuro de un programa

- ***Sin embargo***

  Se utiliza como punto de referencia:

  Se puede usar el *log* para ver cuales fueron las páginas que no fueron requeridas por el mayor tiempo.

Algoritmo de reemplazo de páginas "Primera en entrar, primera en salir"

- ***Idea***

Reemplazar la página más antigua.

- ***Desventajas***
  - La página más antigua podría ser necesaria nuevamente
  - Algunas páginas pueden ser importantes

Algoritmo de reemplazo de páginas “no usadas recientemente”

- Bit de **referencia** (lectura o escritura)
  - 1 cuando la página ha sido referenciada
  - Se ajusta a 0 periódicamente
- Bit de **modificación**
  - 1 cuando la página ha sido modificada

Algoritmo de reemplazo de páginas “no usadas recientemente”

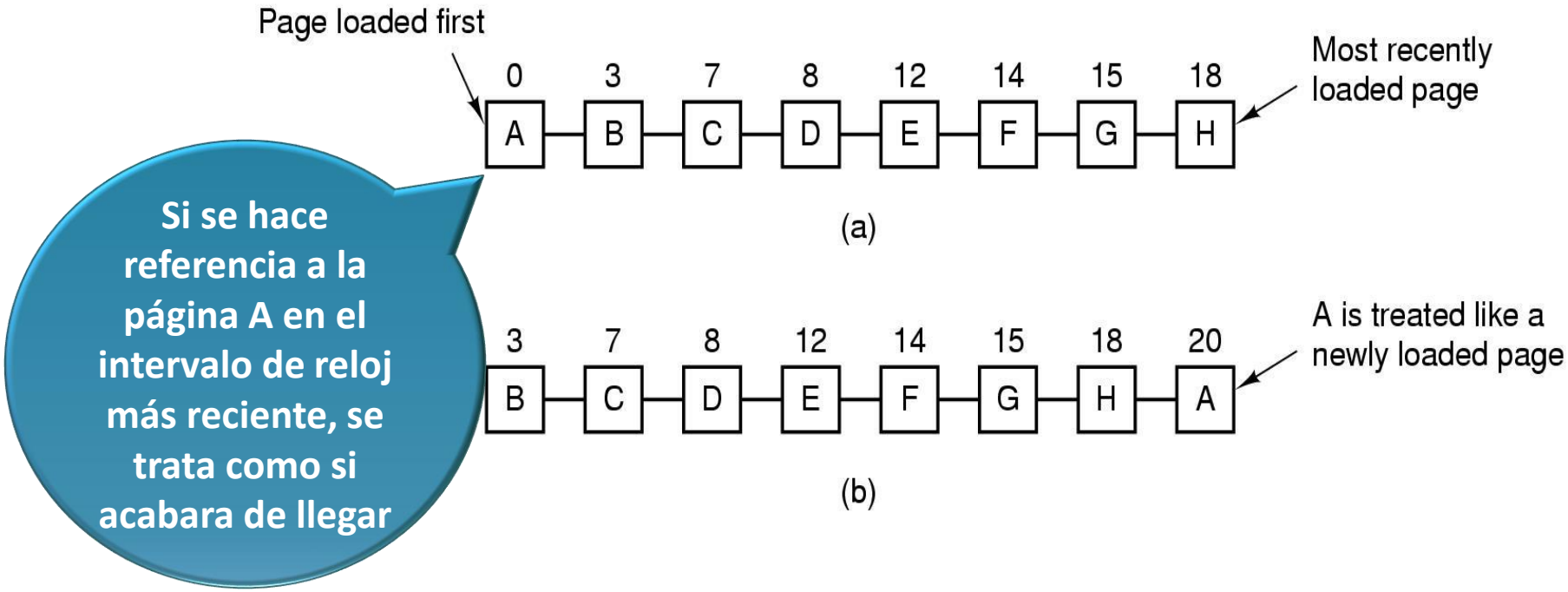
Primero reemplaza de la Clase 0 aleatoriamente

- Clase 0: **no** ha sido referenciada, **no** ha sido modificada
- Clase 1: **no** ha sido referenciada, ha sido modificada
- Clase 2: ha sido referenciada, **no** ha sido modificada
- Clase 3: ha sido referenciada, ha sido modificada

¿Es imposible que una página sea de clase 1?

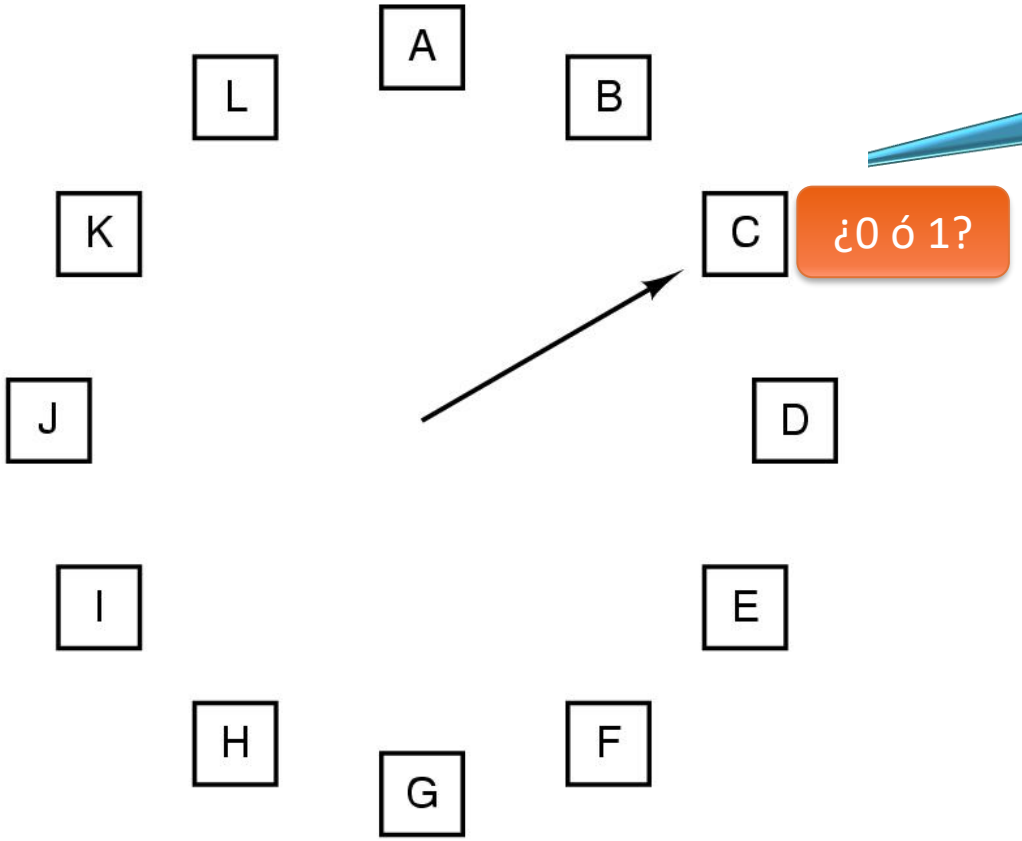
Algoritmo de reemplazo de páginas "segunda oportunidad"

- Modificación del algoritmo FIFO utilizando el Bit de **referencia**



Algoritmo de reemplazo de páginas “reloj”

- Modificación del algoritmo “segunda oportunidad”



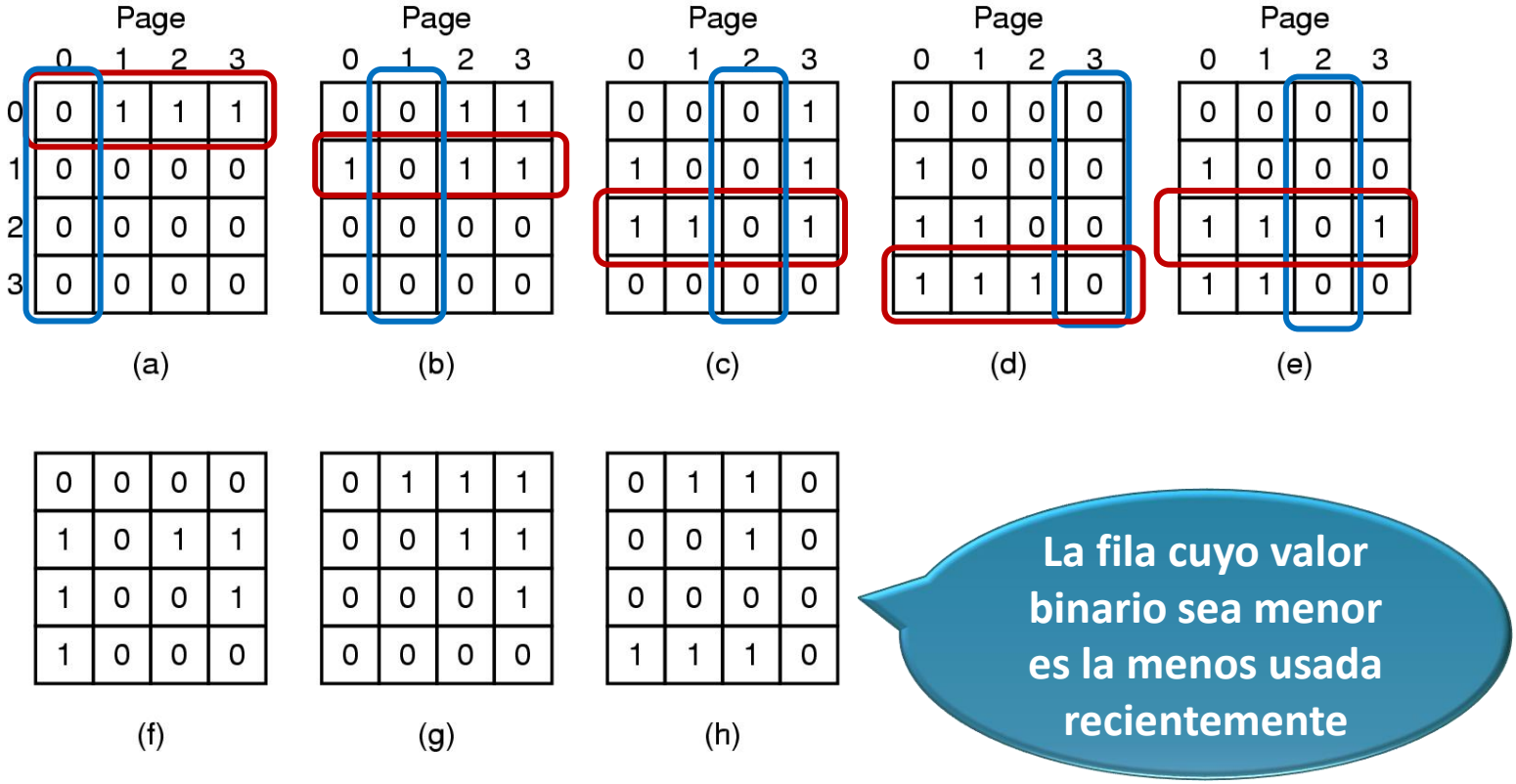
La manecilla apunta a la página más antigua

Cuando ocurre un fallo de página. Si el bit de referencia es 0, la página se desaloja, sino, el bit de referencia se limpia y la manecilla avanza

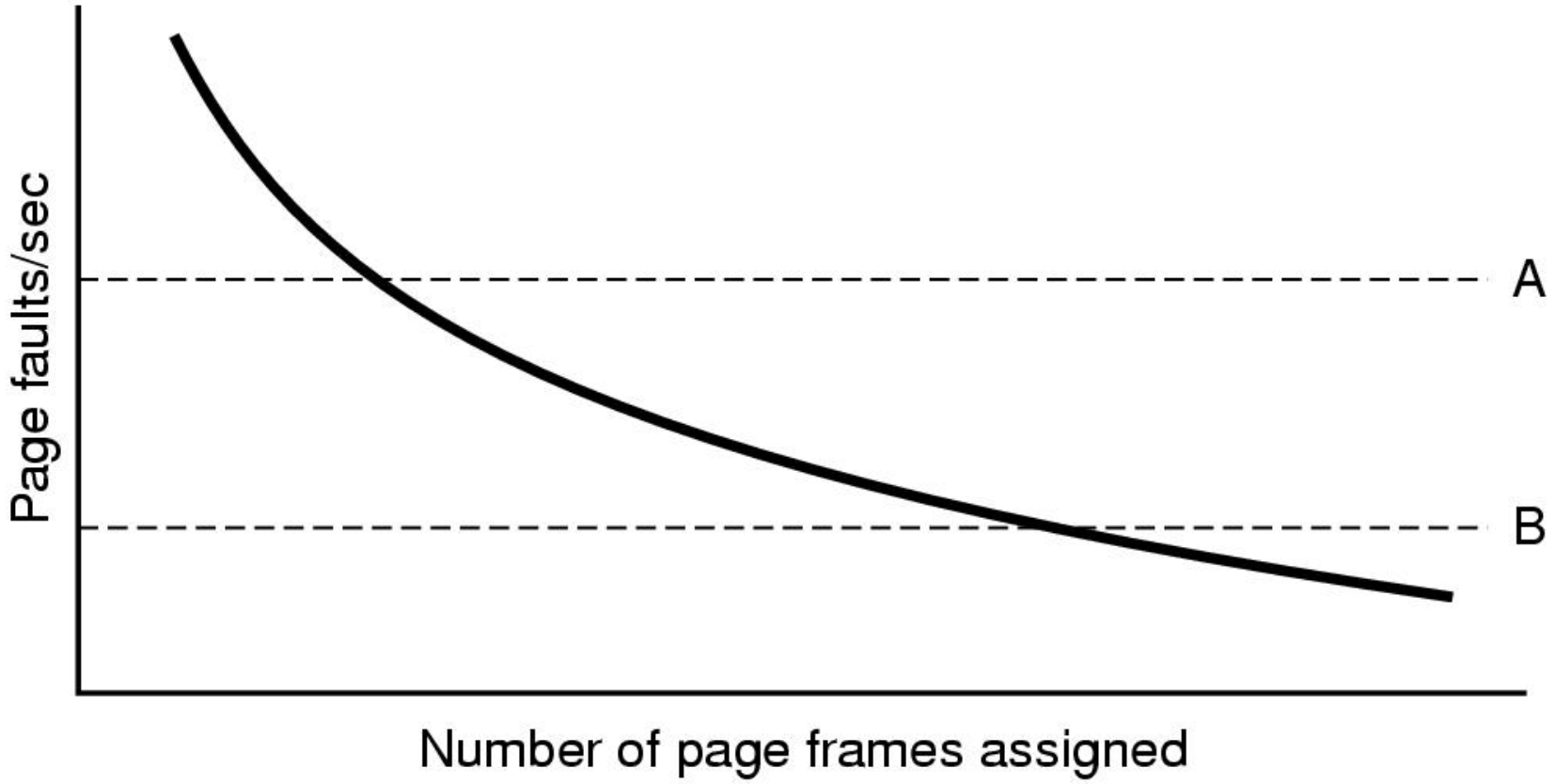
Algoritmo de reemplazo de páginas "menos usadas recientemente"

- Asume que las páginas usadas recientemente se utilizarán de nuevo pronto

Referencia a las páginas en el orden 0,1,2,3,2,1,0,3

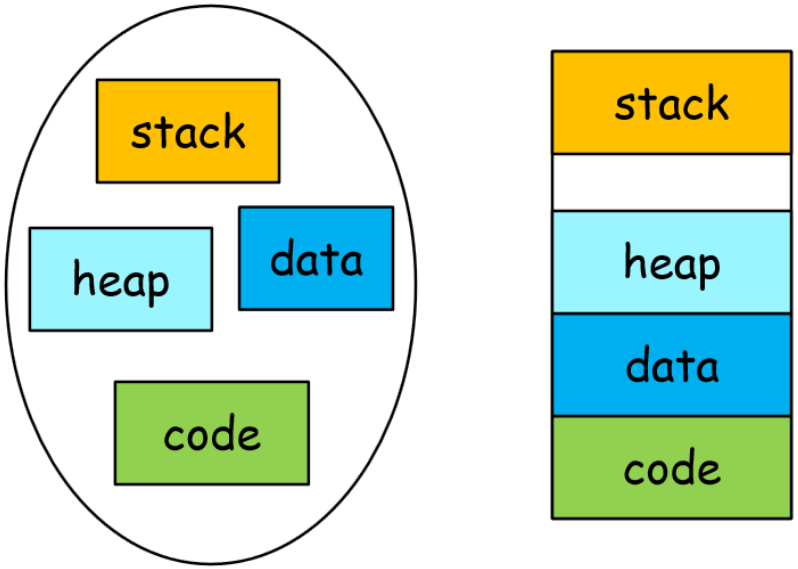


Fallos de páginas VS # marcos de página asignadas



Segmentación

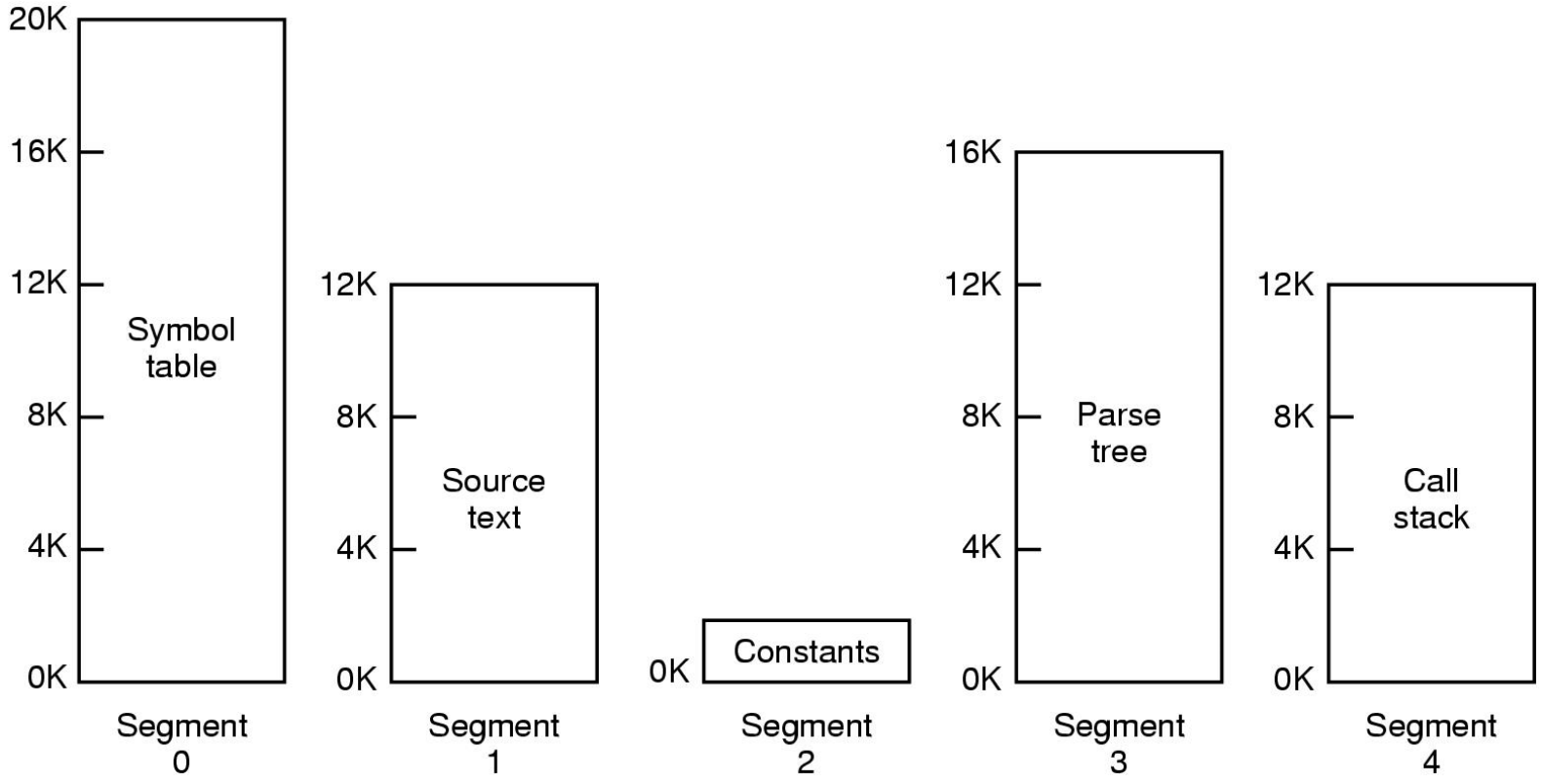
- Es una técnica de gestión de memoria que asigna la memoria virtual por segmentos



- **Cada proceso y sus datos se dividen en segmentos de longitud variable.**
- **Un proceso carga sus segmentos en particiones dinámicas no necesariamente contiguas.**
- **Todos los segmentos de un proceso se deben de cargar en memoria.**

Segmentación

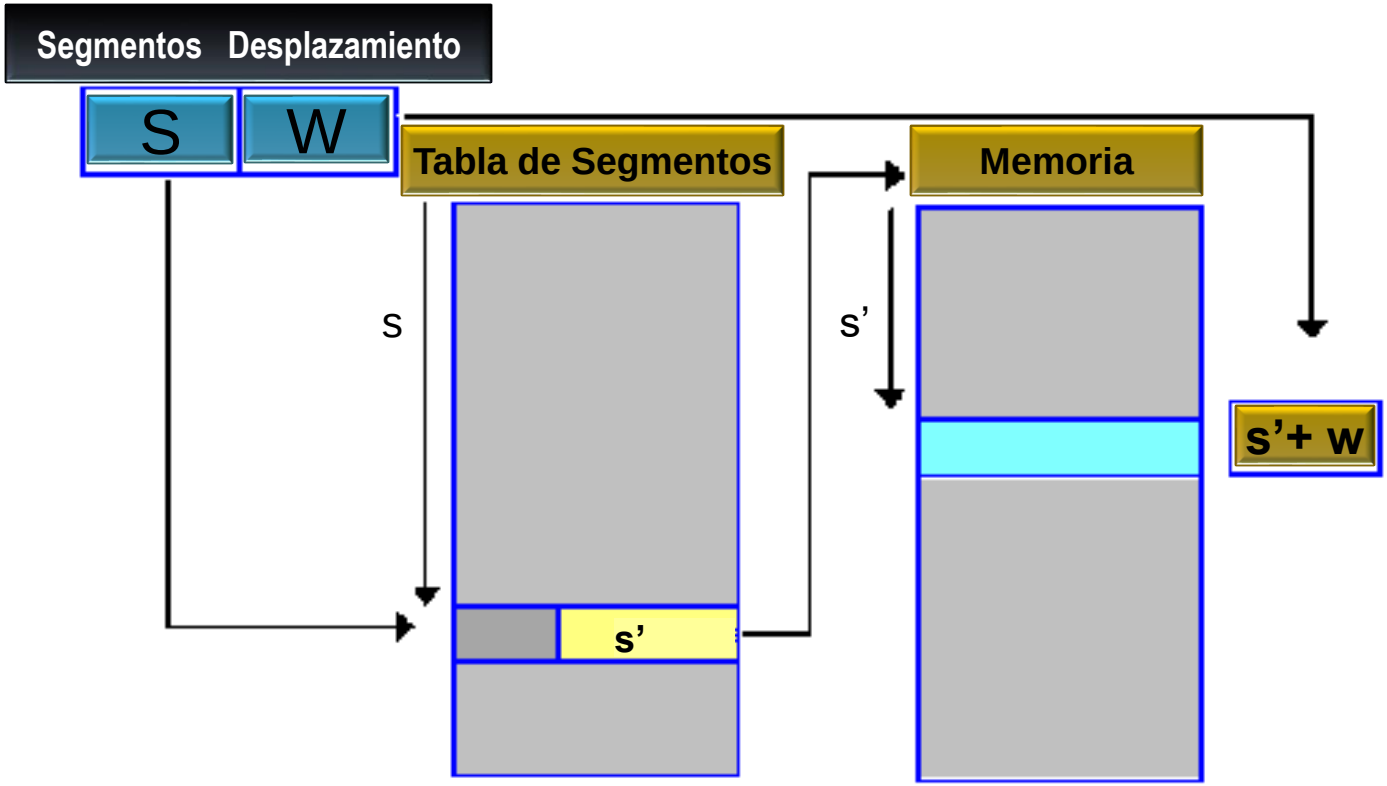
- Es una técnica de gestión de memoria que asigna la memoria virtual por segmentos



Segmentación

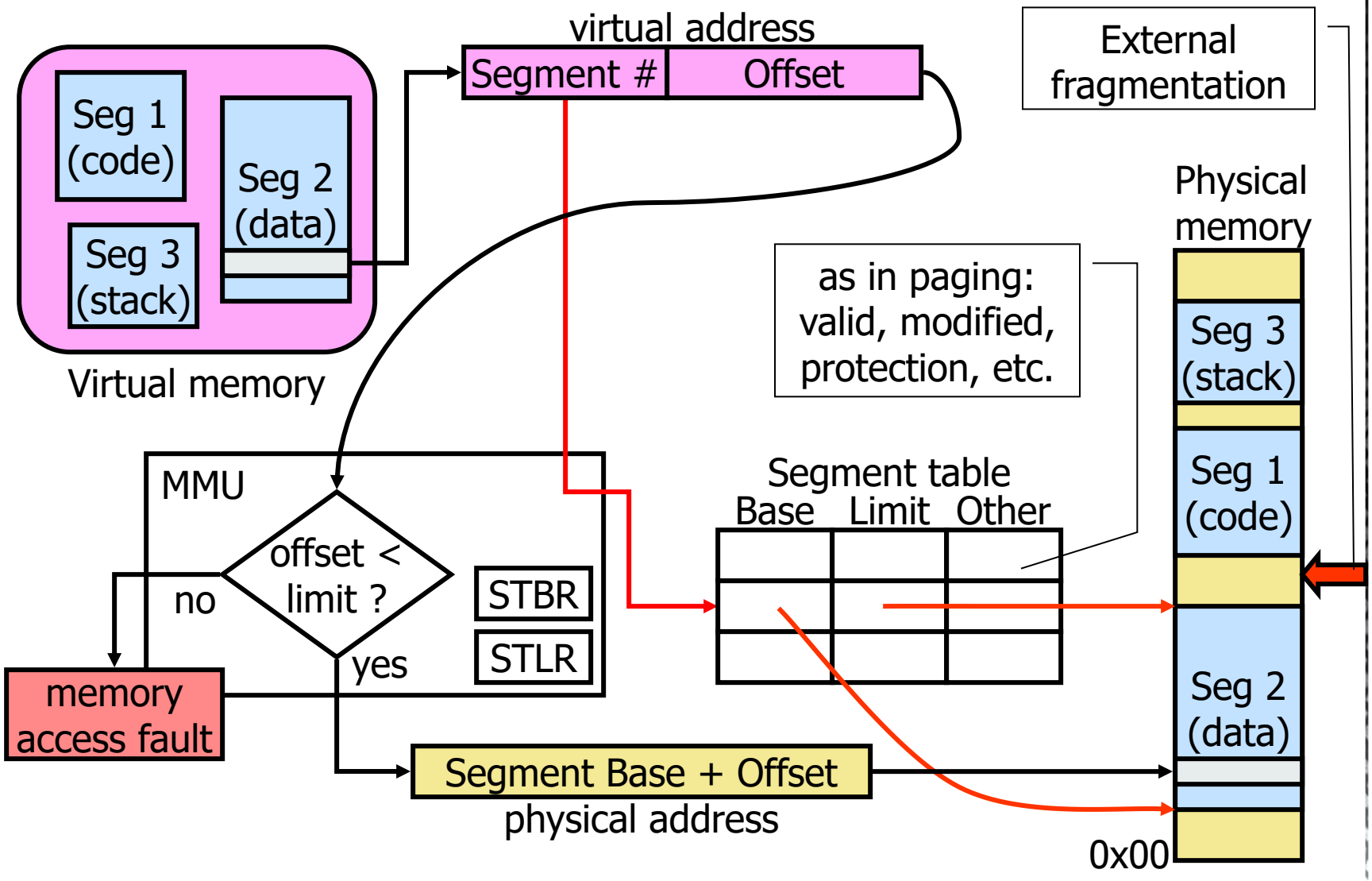
- El SO mantiene una tabla de segmentos para cada proceso.

- Una dirección de memoria es un número de segmento (S) y un desplazamiento dentro de segmento (W).

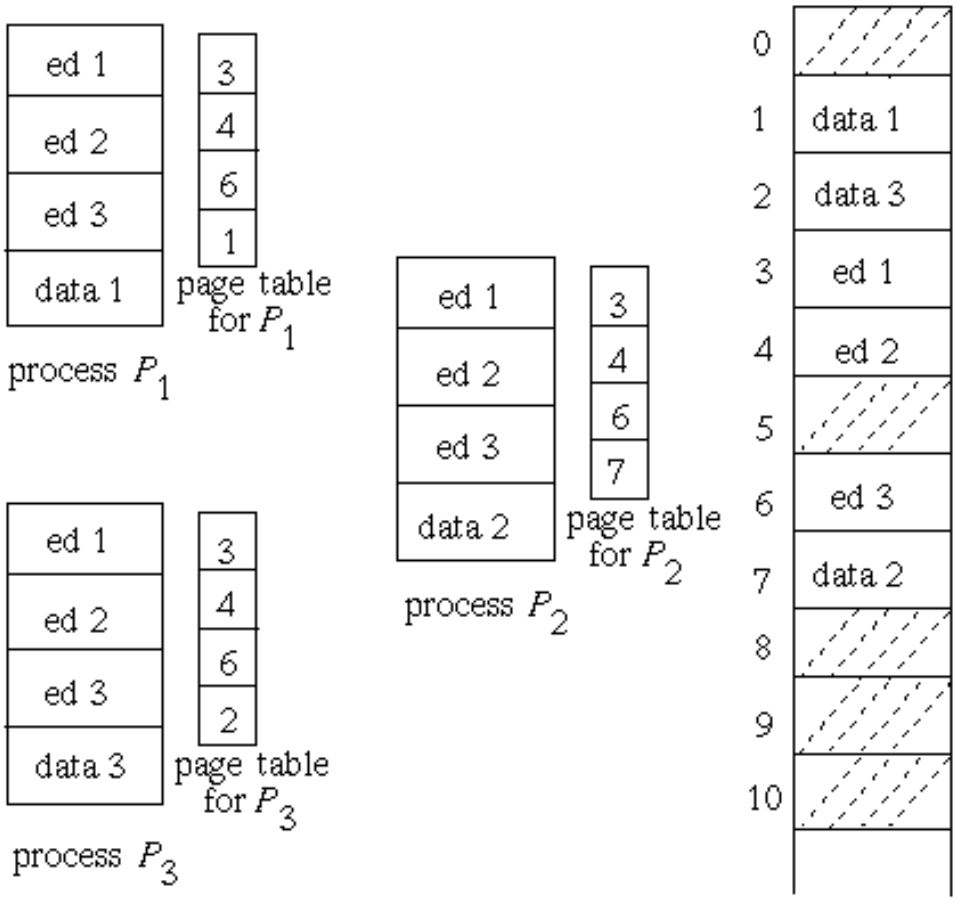


Segmentación

STBR: Segment-table base register
STLR: Segment-table length register



Páginas compartidas



- Una sola copia de páginas (de código compartido) en común entre procesos.
- Datos privados en páginas separadas