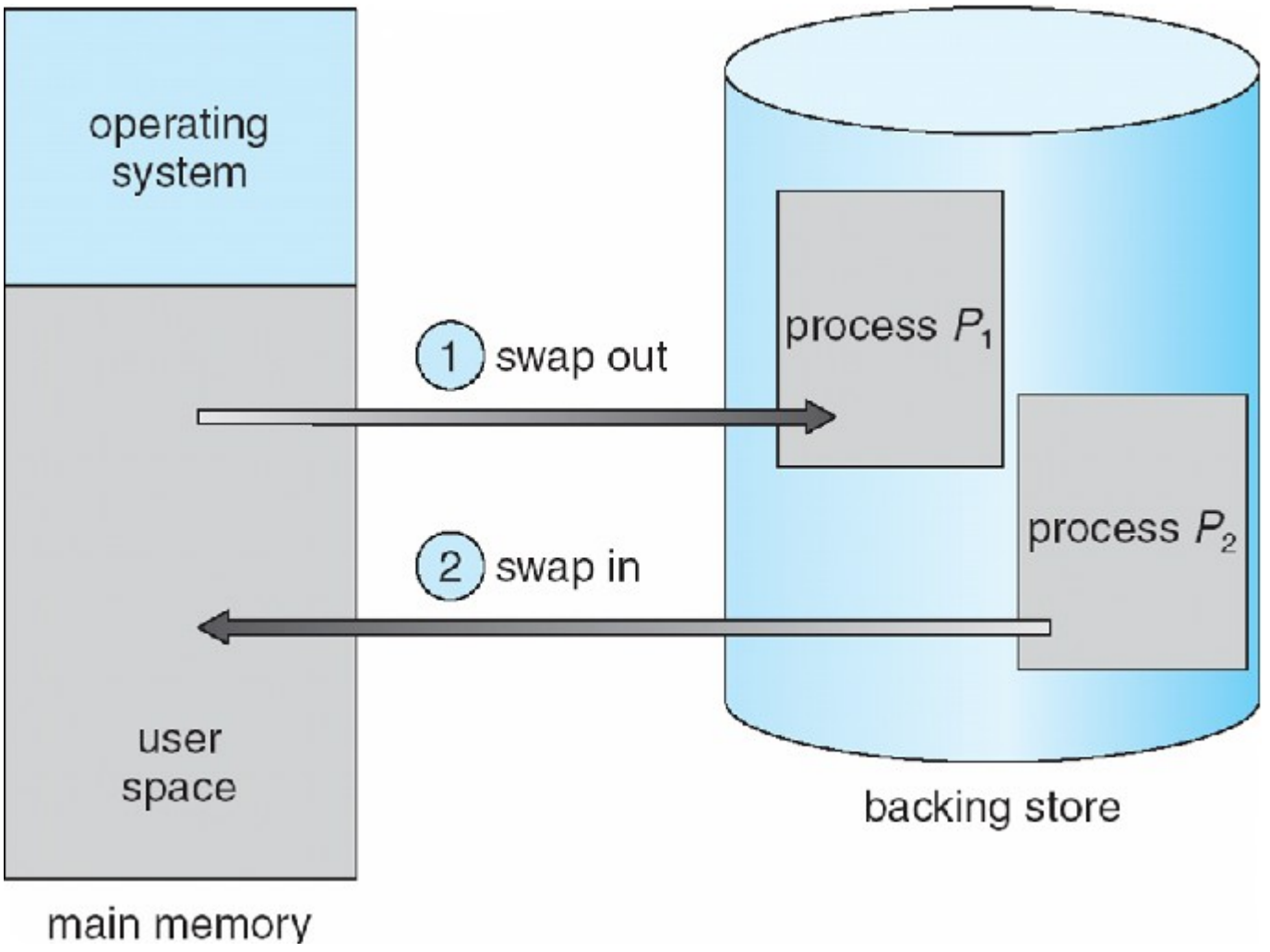
Sistemas Operativos 1

Memoria Virtual

Swapping

- Un proceso puede ser sacado temporalmente de memoria a almacenamiento secundario, y traído nuevamente a memoria para continuar su ejecución.
- Almacenamiento secundario – discos rápidos y suficientemente grandes para acomodar copias de la imagen de memoria; debe proveer acceso Directo
- Roll out, roll in – variante de swapping usado para planificadores por prioridad; los procesos de menor prioridad son desalojados de forma que los de mayor prioridad puedan ejecutarse
- La mayor parte del swap es tiempo de transferencia; proporcional al tamaño de memoria

Diagrama de Swapping



Memoria virtual

- Memoria Virtual – separación de la memoria lógica del usuario y la memoria física
- Sólo una parte del programa se necesita en memoria para su ejecución
- El espacio de direcciones lógico puede ser mucho más grande que el espacio de direcciones físicas
- Permite que el espacio de memoria sea compartido por varios procesos
- Permite la creación de procesos en forma más eficiente

Paginación Bajo Demanda

- Carga una página en memoria sólo cuando se necesita
  - Menos E/S
  - Menos memoria
  - Respuesta más rápida
  - Más usuarios
- Se necesita una página => se referencia
  - Referencia inválida => termina
  - No está en memoria => se trae a memoria

Valid-Invalid Bit

- Con cada página se asocia un bit valid-invalid (v in-memory, i not-in-memory)
- Inicialmente el bit valid-invalid es puesto en i para todas las entradas
- Ejemplo de tabla de página:
  - Durante la traducción, si el bit valid-invalid
  - en la entrada de la tabla es i page fault

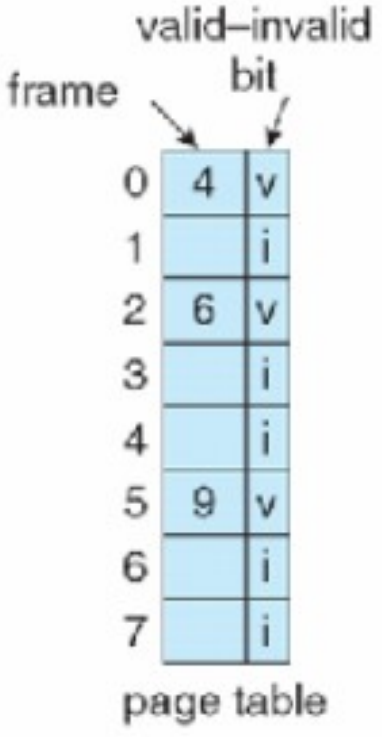
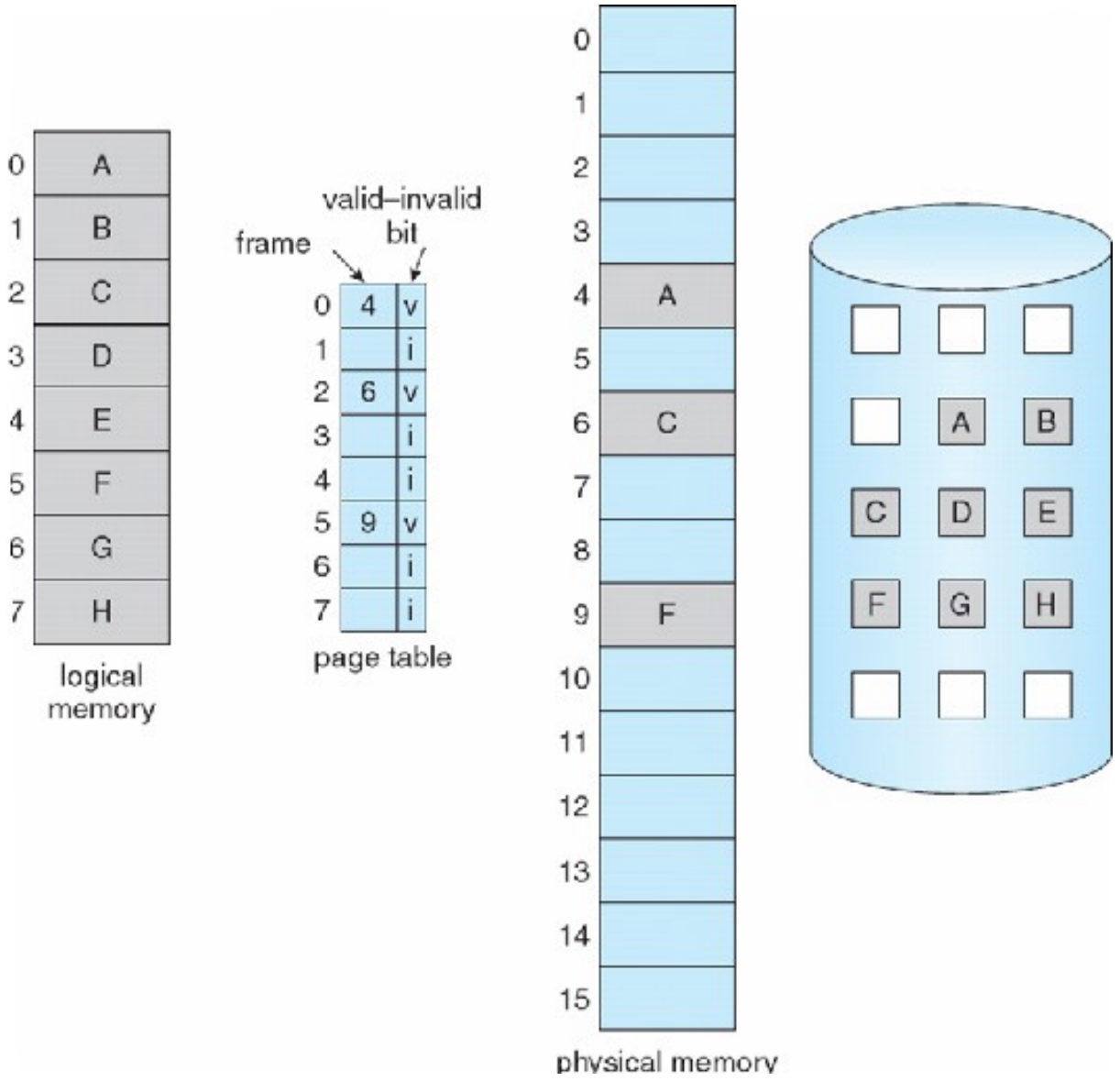


Tabla de Página Cuando Algunas Páginas no Están en Memoria

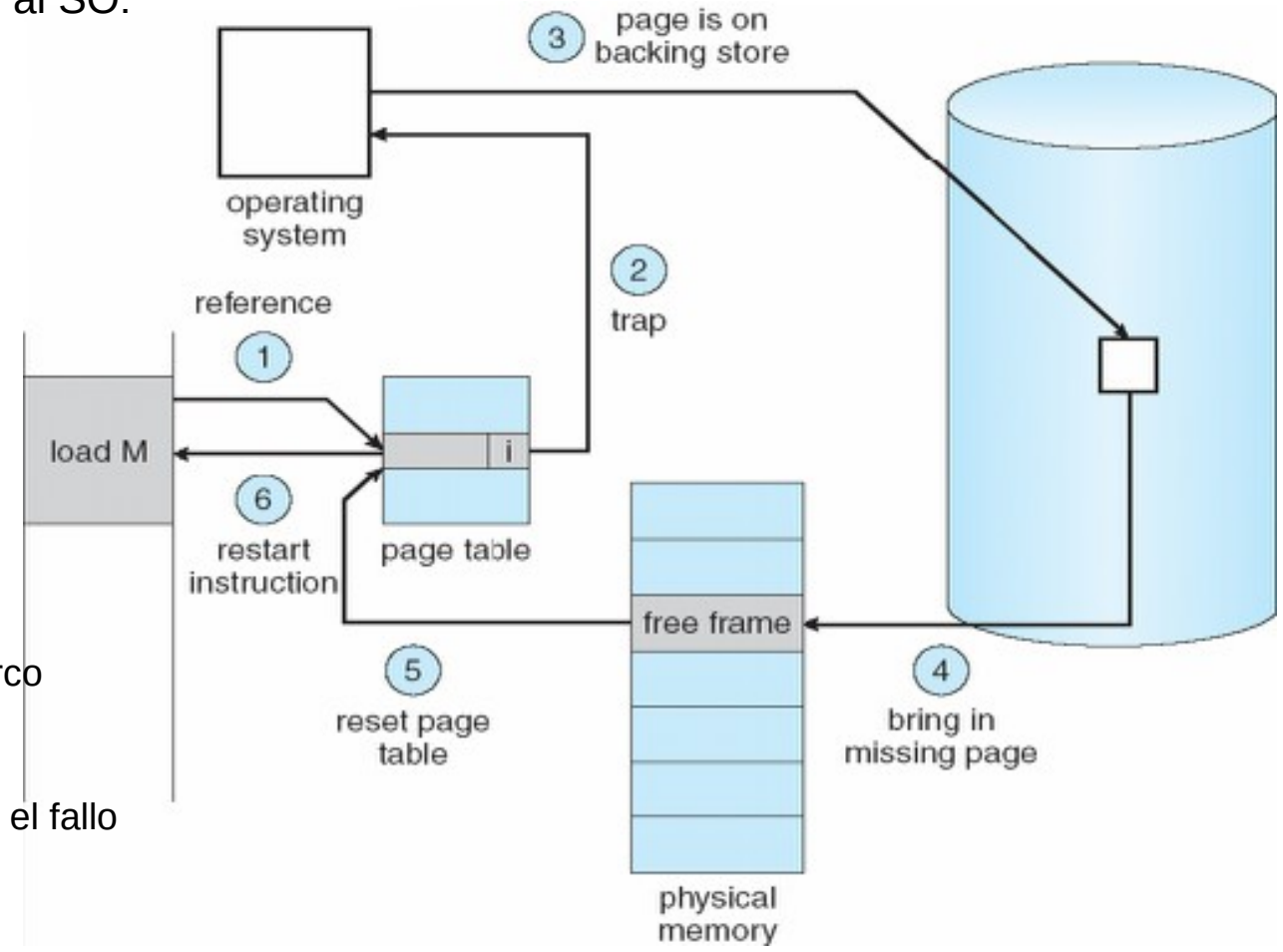


Fallo de Página

- Cuando hay una referencia a una página, la primer referencia genera un trap al SO: # Page Fault

El SO decide:

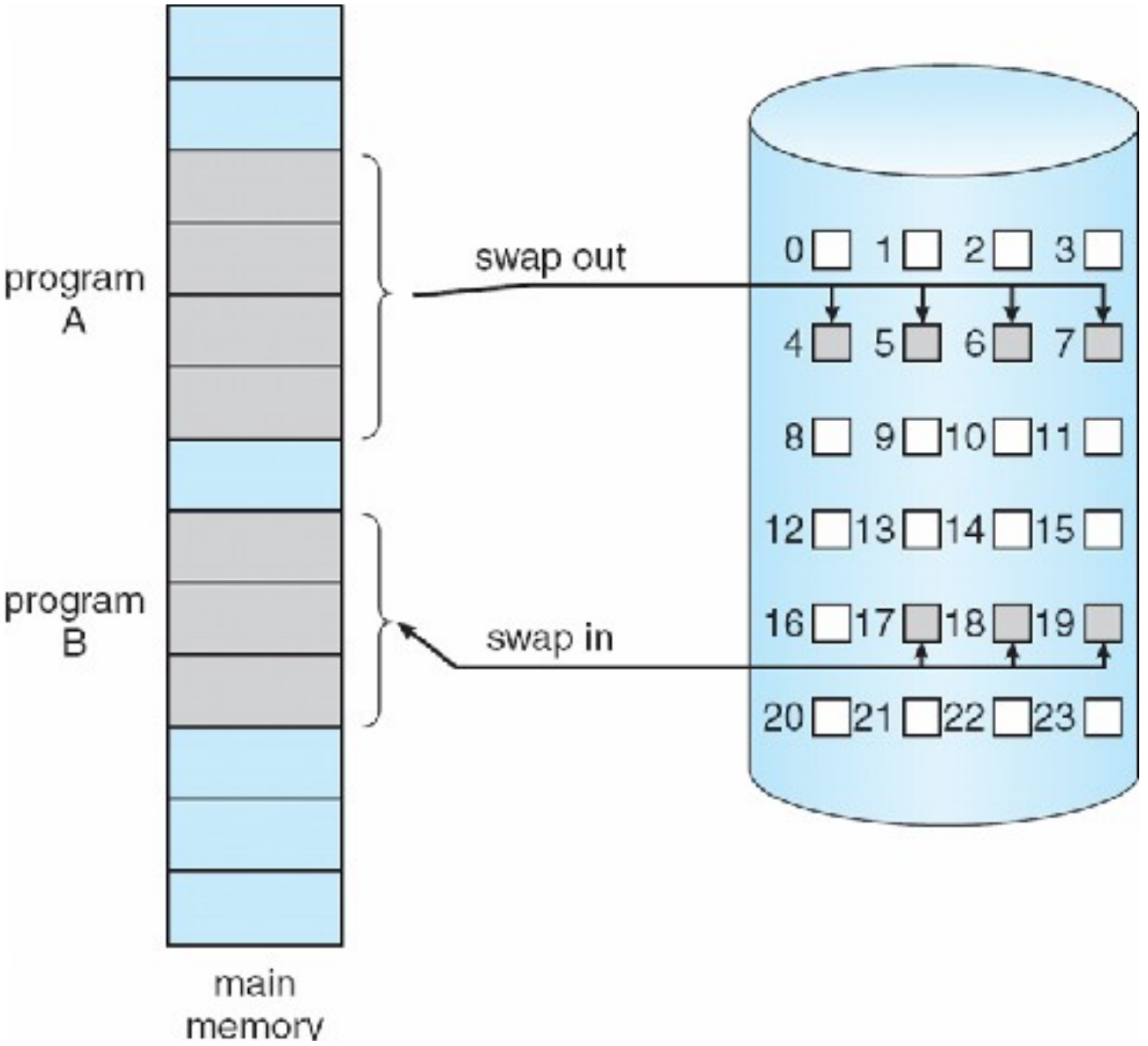
- referencia inválida => abort
- sólo no está en memoria
  - Obtiene un marco libre
  - Hace swap de la página en el marco
  - Pone el validation bit = v
  - Reinicia la instrucción que generó el fallo de página



Qué sucede si no Hay Marcos Libres?

- Reemplazo de página – busca una página en memoria pero que no este en uso y hace **swap out**
  - algoritmo
  - performance – queremos un algoritmo que resulte en el menor número de fallos de página
- La misma página puede ser traída a memoria muchas veces

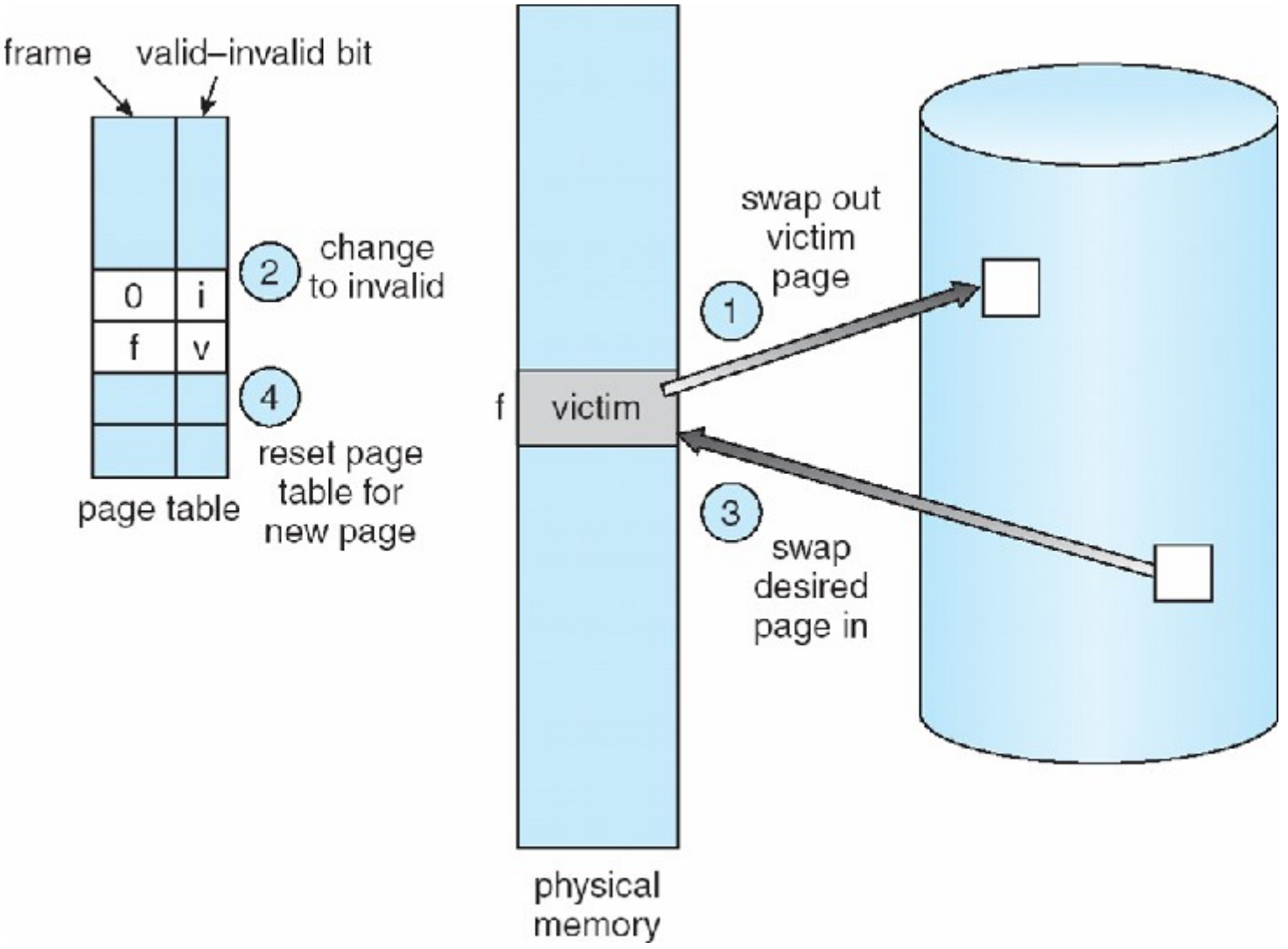
Transferencia de Páginas a Disco



Reemplazo Básico

- Encontrar la ubicación de la página en disco
- Encontrar un marco libre:
  - Si existe, usarlo
  - Si no, usar un algoritmo de reemplazo de página para encontrar una víctima
- Traer la página al marco libre; actualizar las tablas de marcos y páginas
- Reiniciar la ejecución del proceso

Reemplazo de Páginas

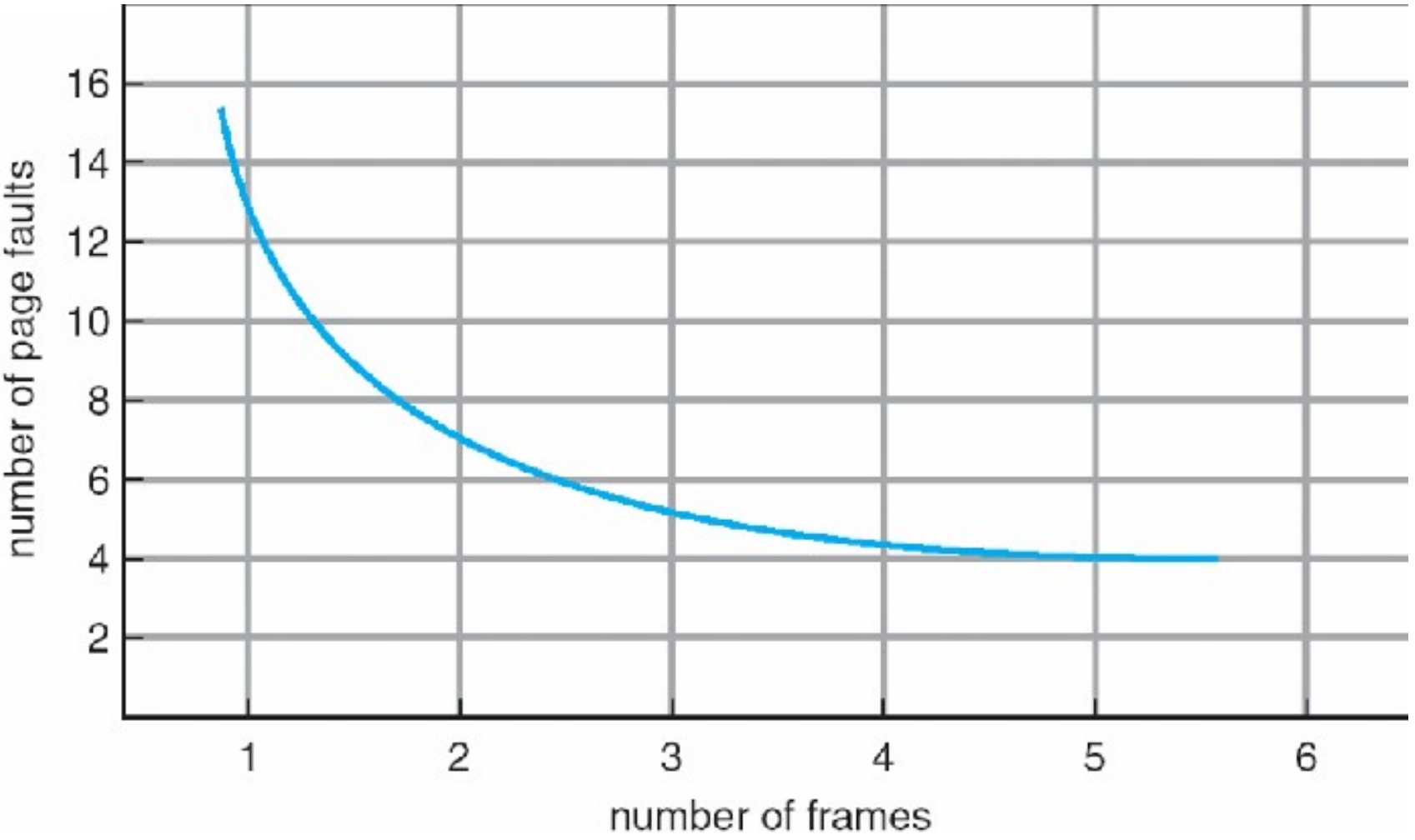


Algoritmos de Fallo

- Queremos la menor frecuencia de fallo
- Evaluamos el algoritmo corriendo con una secuencia
- particular de referencias a memoria (reference string) y calculamos el número de fallos de la secuencia
- En los ejemplos usamos la siguiente secuencia:

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

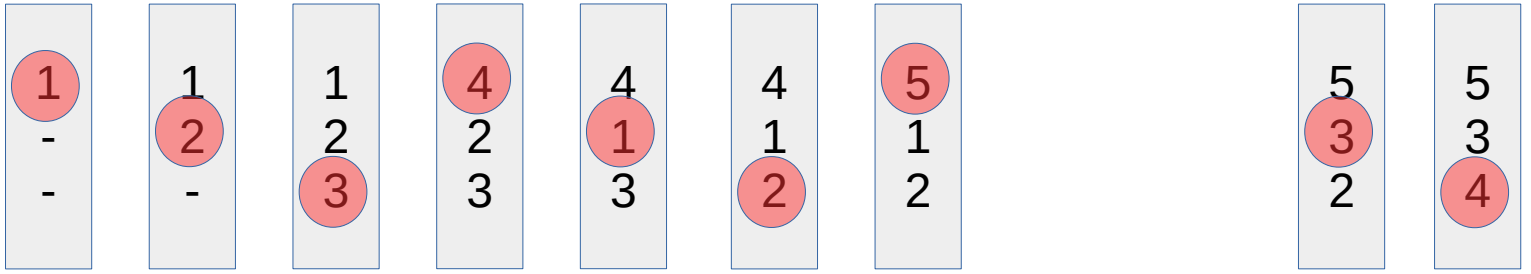
Gráfico de Fallos vs #Marcos



First-In Fist Out

- Ejemplo con 3 marcos
- Reference string:

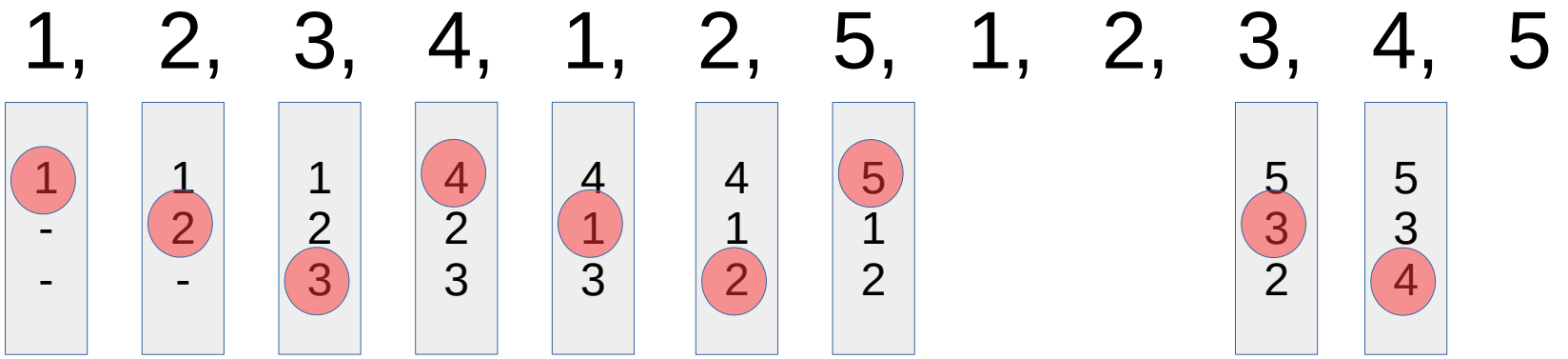
1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5



9 fallos de página

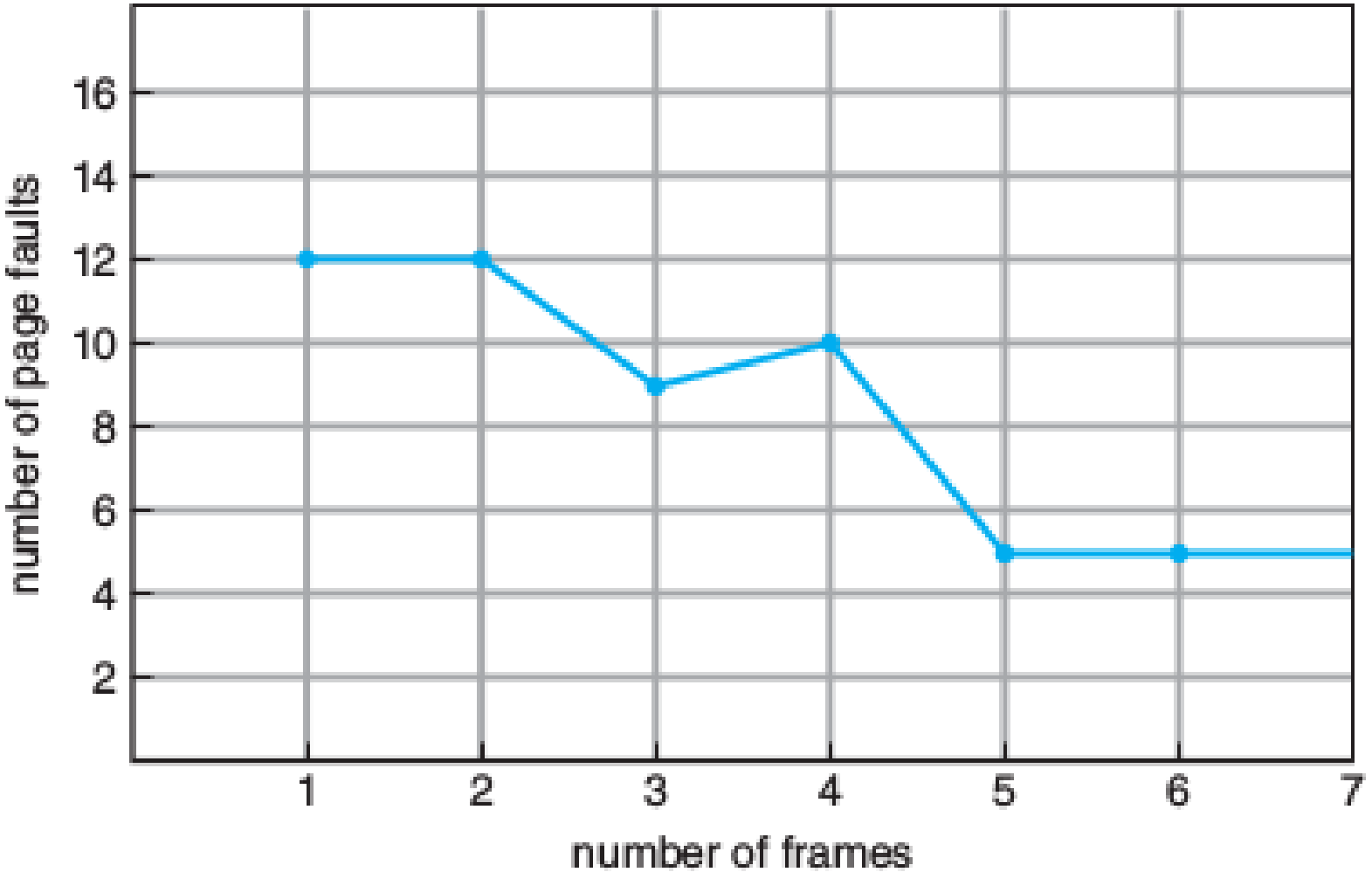
First-In Fist Out

- Reference string:



- Anomalía de Belady's: más marcos más fallos

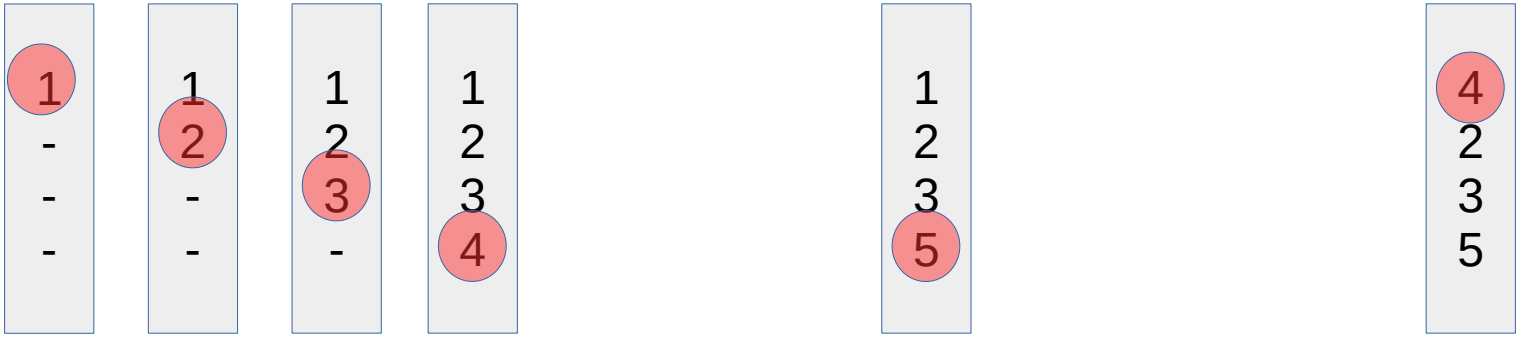
Anomalía de Belady con FIFO



Algoritmo Óptimo

- Reemplaza la página que no va a ser usada por el período más largo... **Cómo lo sabemos?**
- Usado para medir como performa un algoritmo
- ejemplo con 4 marcos

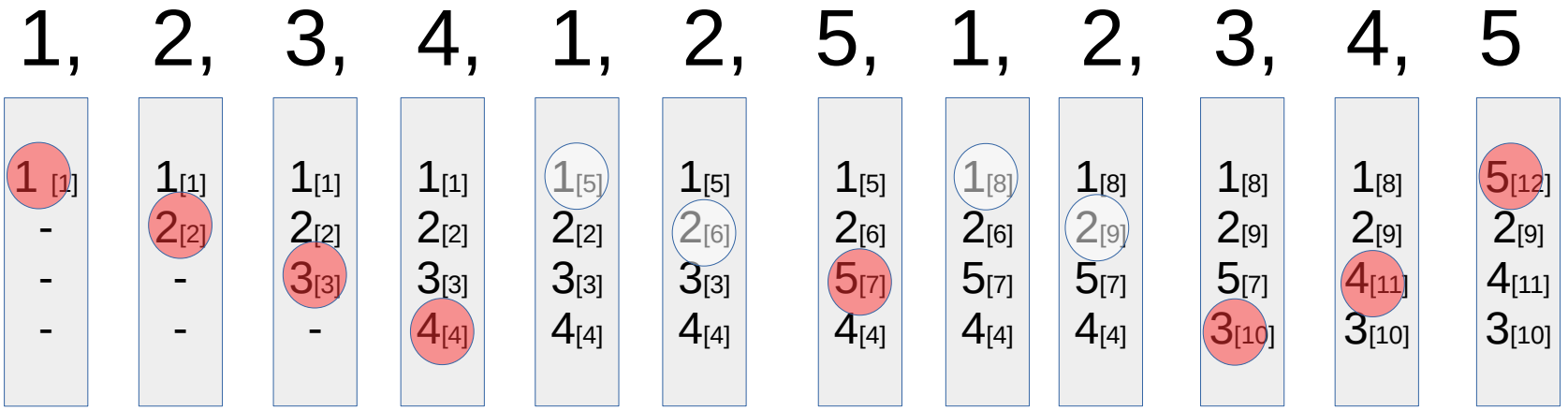
1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5



6 fallos de página

Least Recently Used (LRU)

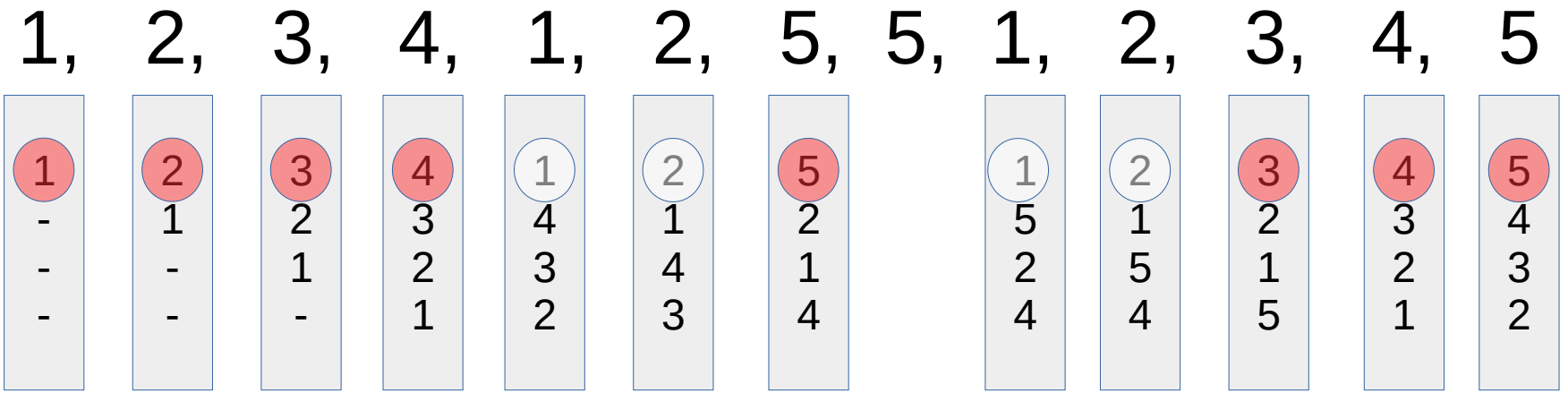
- Implementación del contador
- Cada página tiene un contador; cada vez que una página es referenciada se copia el clock en el contador
- Cuando se necesita reemplazar una página, se mira el contador para seleccionar el reemplazo. (**menor valor**)
- ejemplo con 4 marcos



8 fallos de página

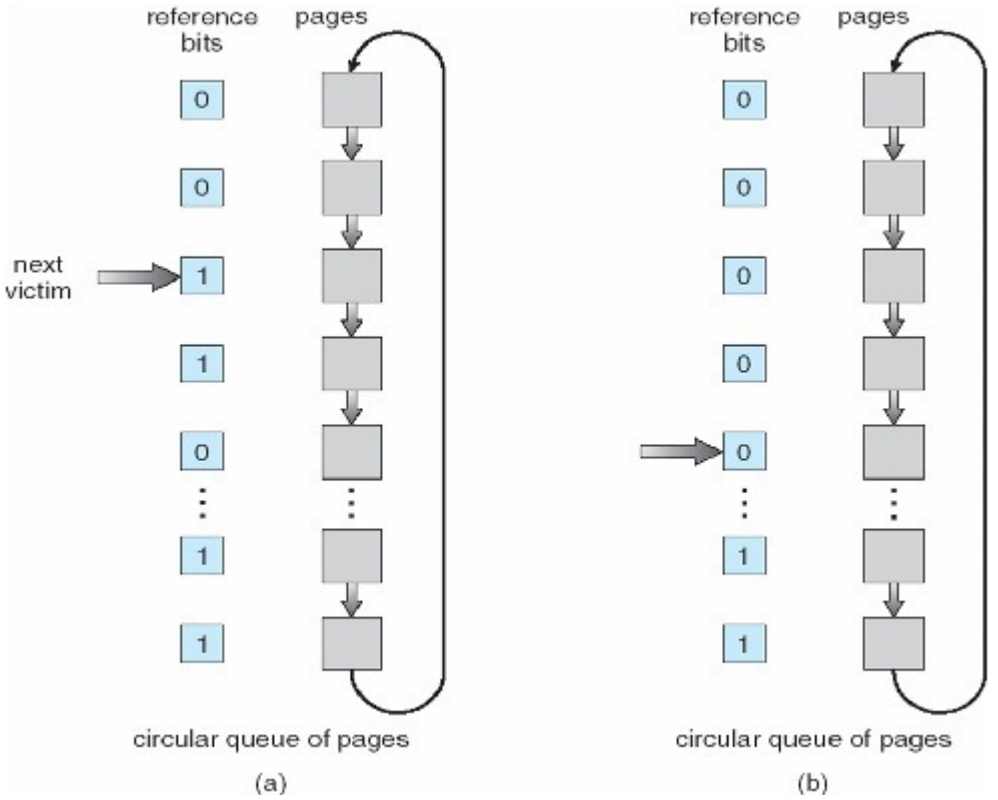
Reemplazo LRU: con stack

- Implementación con un Stack
  - mantiene el stack como una lista doblemente enlazada:
- Referencia a página:
  - la mueve al tope
  - requiere actualizar punteros (si referencia a página != a la anterior)
- No requiere búsqueda
- ejemplo con 4 marcos



Algoritmo de Segunda Oportunidad (Reloj)

- Necesita un bit de referencia
- Remplaza en el sentido de las agujas del reloj si el bit = 0
- Si la página a reemplazar tiene el bit de referencia = 1, entonces:
  - pone el bit en 0
  - deja la página en memoria
  - reemplaza la siguiente página con las mismas reglas



Algoritmos de Conteo

- Mantiene un contador con el número de referencias hechas a cada página
- **LFU:** reemplaza la página con el menor número de referencias
- **MFU:** reemplaza la página con el mayor número de referencias, se basa en el argumento de que la página con menos referencia recién se trajo a memoria

Asignación de Marcos

- Cada proceso necesita un número mínimo de páginas
-
- Existen 2 esquemas principales de asignación
  - asignación fija
  - asignación por prioridad

Asignación Fija

- Igual Asignación
  - por ejemplo, si hay 100 marcos y 5 procesos, cada uno recibe 20 marcos
- Asignación Proporcional
  - asigna de acuerdo al tamaño de los procesos

Asignación por Prioridad

- Usa un esquema de asignación proporcional usando prioridades en lugar del tamaño
- Si un proceso genera un fallo,
  - se selecciona para reemplazo uno de sus marcos
  - se selecciona para reemplazo un marco de un proceso con menor prioridad

Reemplazo Global vs Local

- Reemplazo Global
  - un proceso selecciona para reemplazo de toda la lista de marcos; un proceso puede tomar marcos de otro proceso
- Reemplazo Local
  - cada proceso selecciona solo de su propio conjunto de marcos

Thrashing

- Si un proceso no tiene “suficientes” páginas, la tasa de fallos es muy alta.
- Esto genera:
  - baja utilización de CPU
  - surge una condición de swapping inútil y repetitiva,
  - la tasa de fallos de página típicamente se eleva.
- Thrashing ≡ un proceso está más tiempo ocupado haciendo swapping

Trashing