

Algoritmos de sustitución

- Podemos tener las siguientes combinaciones
 - » asignación fija y sustitución local
 - » asignación variable y sustitución local
 - » asignación variable y sustitución global
- Veremos distintos algoritmos de sustitución y nos basaremos (por simplicidad) en que se utiliza una política de asignación fija y sustitución local
- *Cadena de referencia*, $\omega = r_1, r_2, r_3, \dots, r_i, \dots$: secuencia de números de páginas referenciadas por un proceso durante su ejecución

Algoritmo Optimo

- Se sustituye la página que no será objeto de ninguna referencia posterior o que se reference más tarde
 - » 4 marcos de página
 - » **Problema**: debemos tener un “conocimiento perfecto” de la cadena de referencia
 - » Se utiliza para medir cómo de bien se comportan otros algoritmos
 - » Faltas de páginas: 6

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

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| * | * | * | * | | | * | | | | * | |

Algoritmo FIFO

- Se sustituye la página por orden cronológico de llegada a MP (la página más antigua)

» 4 marcos de página
» Faltas de página: 10
» Sufre de la *Anomalía de Belady*: “más marcos no implican menos faltas de páginas”

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

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 4 | 4 |
| 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 5 | |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | | |

* * * * * * * * * *

Algoritmo LRU

- Se sustituye la página que fue objeto de la referencia más antigua (*Least Recently Used*)
 - » 4 marcos de página
 - » Faltas de página: 8
 - » Implementación del algoritmo:
 - con contadores
 - con pila
 - » Mayor coste

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

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 5 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 4 | 4 | | |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | | |
| * | * | * | * | | | * | | * | * | * | |

Implementaciones de LRU

- LRU con contador

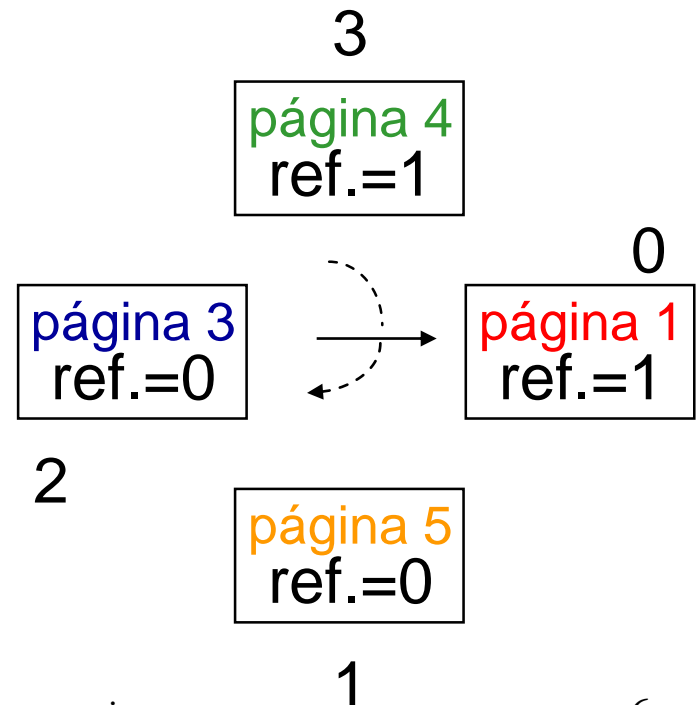
- » Cada entrada de la tabla de páginas tiene un contador. Cada vez que se referencia la página, se copia el tiempo del reloj en el contador
- » Cuando necesitamos cambiar una página, se miran todos los contadores y se elige la que tiene el menor tiempo

- LRU con pila

- » Los números de páginas se mantienen en una pila (lista doblemente enlazada). Cuando se referencia una página se mueve a la cima de la pila (cambio de seis punteros como máximo)
- » No hay que hacer búsqueda para la sustitución de una página, se sustituye la del fondo de la pila

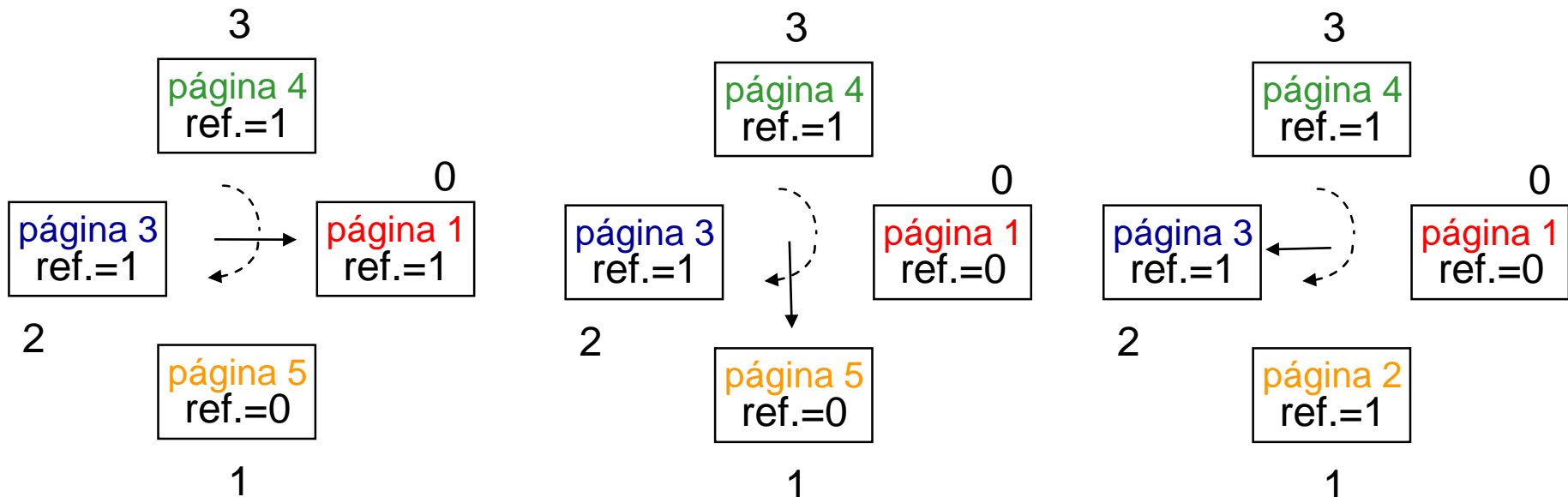
Algoritmo del reloj

- Cada página tiene asociado un bit de referencia R (lo pone a 1 el hardware)
- Los marcos de página se representan por una lista circular y un puntero a la página visitada hace más tiempo
- Selección de una página:
 1. Consultar marco actual
 2. ¿Es R=0?
 - No: R=0; ir al siguiente marco y volver al paso 1
 - Si: seleccionar para sustituir e incrementar posición

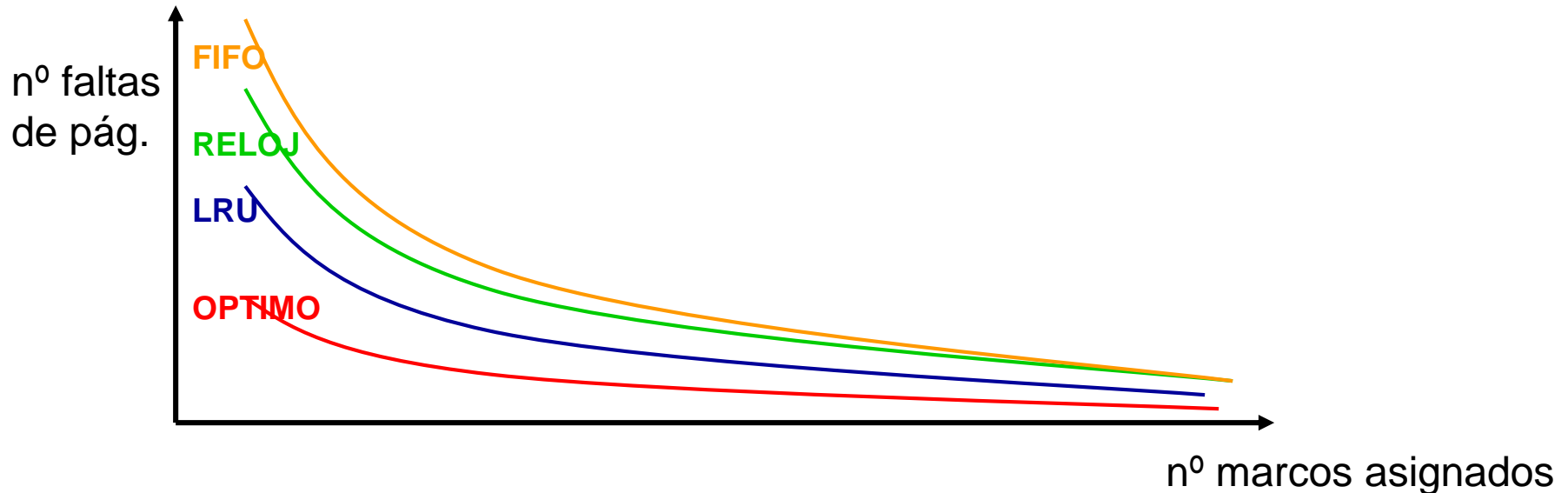


Ejemplo algoritmo del reloj

El proceso intenta acceder a la **página 2** de su espacio de direcciones. Solo hay 4 marcos de página.



Comparación



- Conclusión:
 - » Influye más la cantidad de MP disponible que el algoritmo de sustitución usado