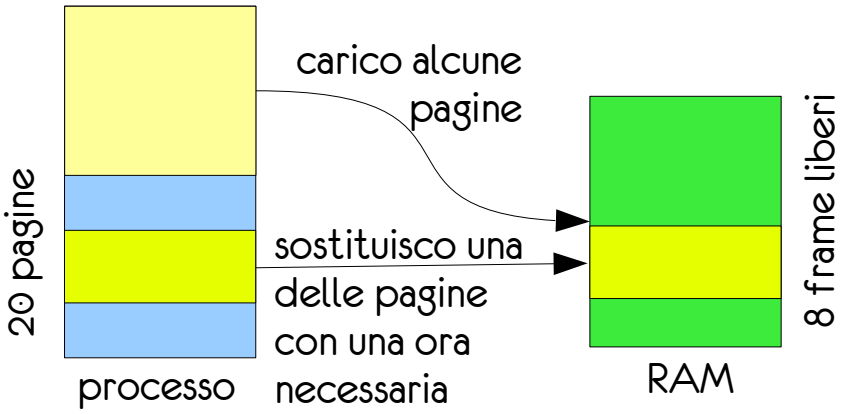
Commenti

- Paginazione:
  - memoria logica e memoria fisica completamente separate
- i processi hanno a disposizione uno spazio degli indirizzi più grande di quello fisico offerto dalla RAM
- realizzazione di un meccanismo dinamico tramite il quale caricare / sostituire pagine a seconda delle esigenze di esecuzione
- introduzione di meccanismi finalizzati ad aumentare l'efficienza, contrastando in parte gli appesantimenti imposti dalle moltiplicate letture/copiature di pagine
- incremento del livello di multiprogrammazione



Implementazione

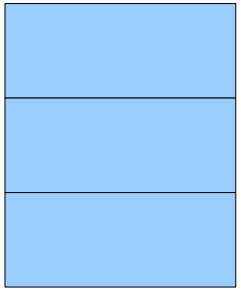
- Implementazione la paginazione su richiesta significa sviluppare due algoritmi:
  - algoritmo di allocazione dei frame:
    - spartisce M frame liberi fra N processi
  - algoritmo di sostituzione delle pagine:
    - seleziona le pagine da sostituire
    - occorre definire il criterio di selezione
    - occorre arricchire l'informazione con i dati necessari per applicare il criterio di selezione
  - è importante poter valutare i diversi algoritmi proposti
    - di norma si sceglie l'algoritmo che causa la frequenza di assenza delle pagine (page fault rate) minore

Algoritmi di sostituzione

- FIFO
- ottimale
- LRU (least-recently used)
- per approssimazione a LRU
  - seconda chance
  - algo. con bit supplementari di riferimento
- basato su conteggio:
  - least frequently used
  - most frequently used

Sostituzione FIFO

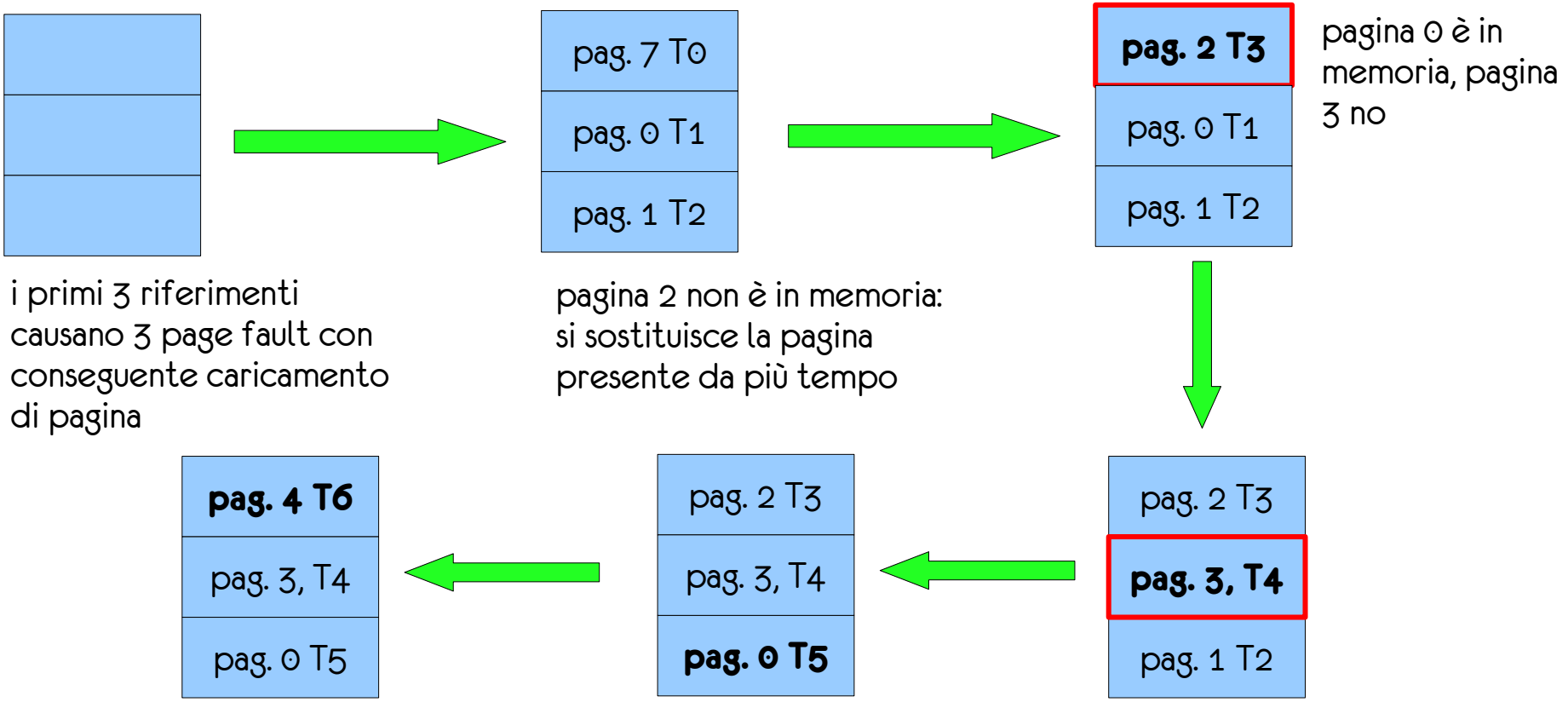
- a ogni pagina viene associato un marcatore temporale: l'istante in cui è stata caricata in memoria
- si sceglie di sostituire la pagina caricata meno di recente (quella in memoria da più tempo)
- basta organizzare le pagine in una coda FIFO
- si sostituisce sempre la pagina corrispondente al primo elemento della coda
- supponiamo di avere una RAM con 3 soli frame e di avere la seguente sequenza di riferimenti: 7, 0, 1, 2, 0, 3, 0, 4



i tre frame sono inzialmente vuoti

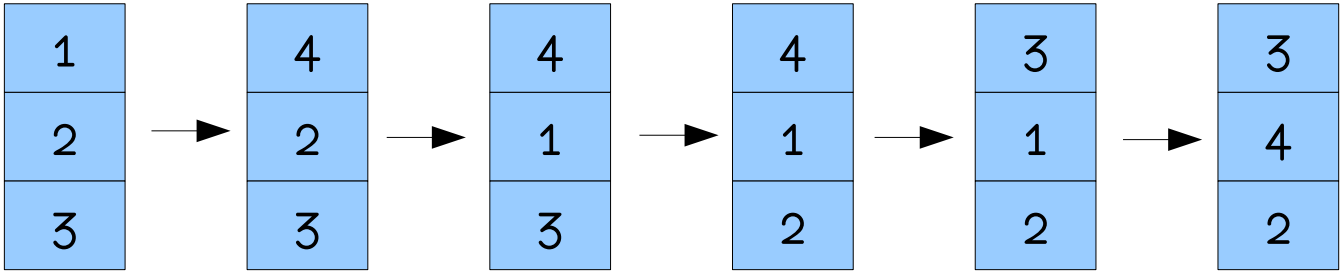
Sostituzione FIFO

- supponiamo di avere una RAM con 3 soli frame e di avere la seguente sequenza di riferimenti: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, ...



Commenti 1/2

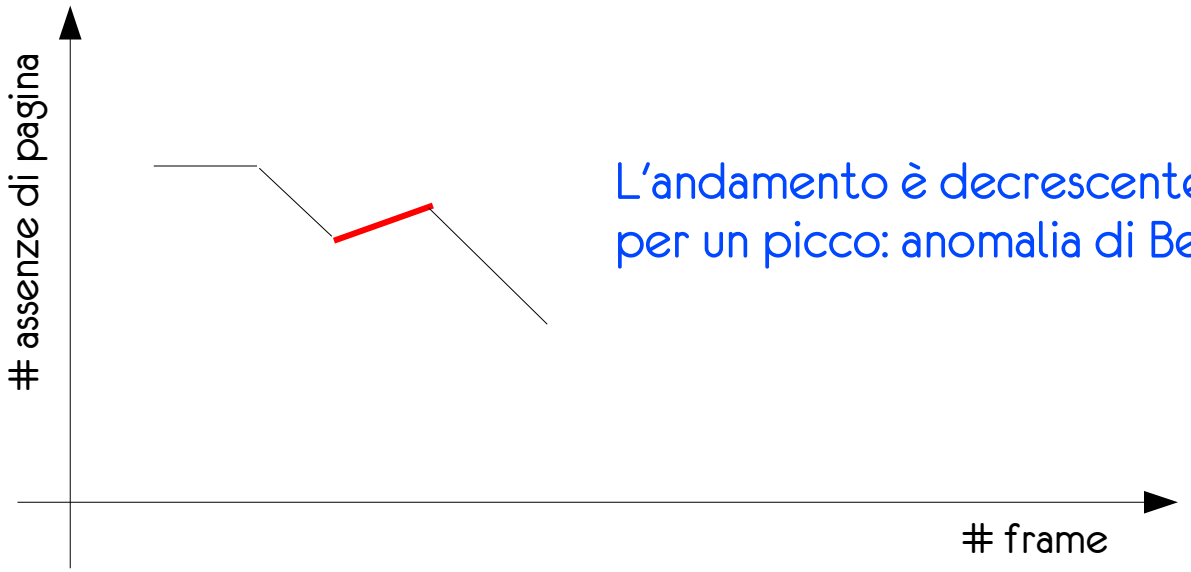
- FIFO è una tecnica semplice da realizzare ma non si comporta sempre bene
- il fatto che una pagina sia stata caricata tempo fa non significa che non sia più in uso, al contrario potrebbe essere una pagina di uso frequente: rimuoverla causerebbe sicuramente un successivo page fault
- consideriamo la sequenza 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4 sempre nel contesto di una RAM con tre frame:



ogni riferimento causa un page fault !!!

Commenti 2/2

- Il numero di page fault dipende dal numero di frame che compongono la RAM: in generale, maggiore il numero di frame, minore la frequenza di page fault
- Tuttavia l'algoritmo di sostituzione delle pagine FIFO presenta l'anomalia di Belady: la frequenza delle assenze può aumentare con l'aumentare del numero di frame in RAM
- Sul libro è riportato un esempio in cui si osserva questo andamento:



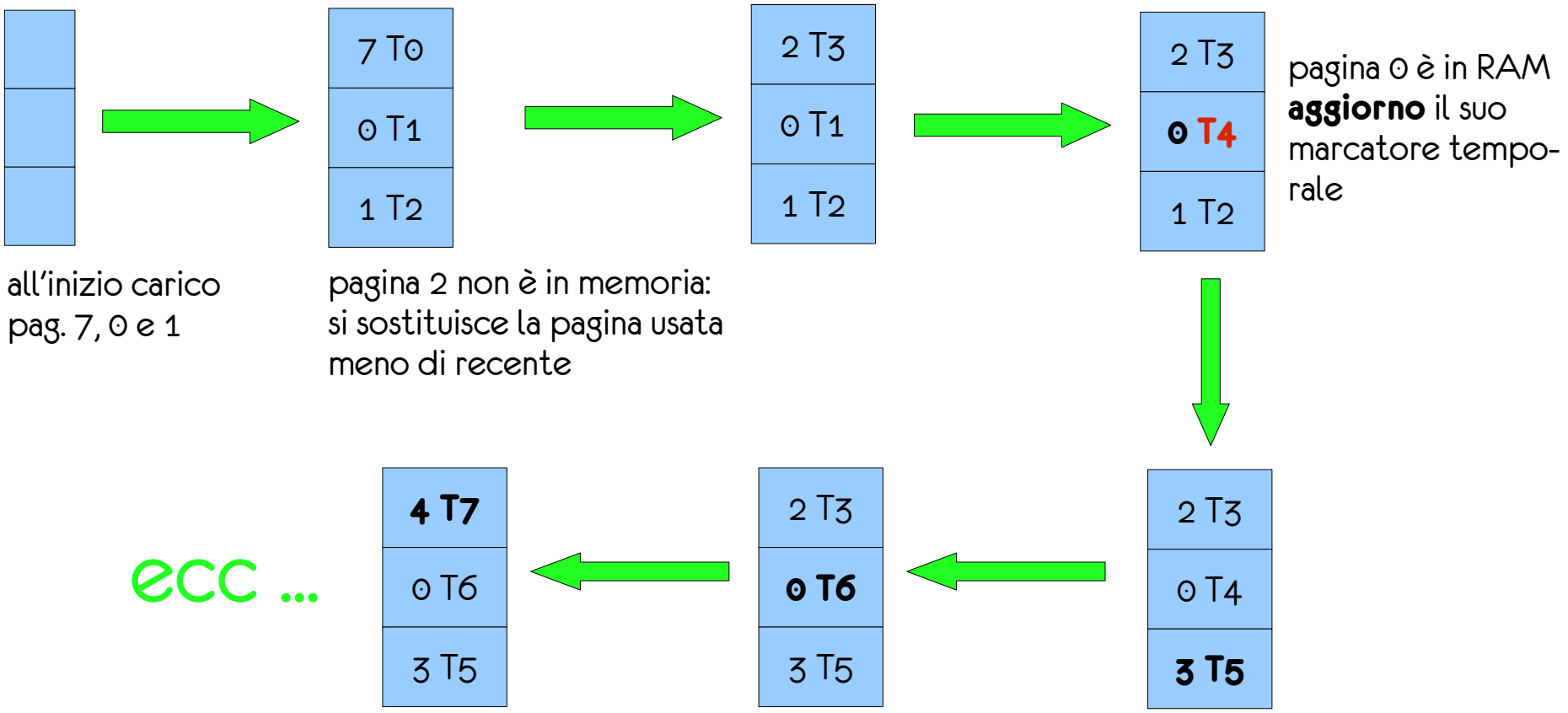
Algoritmo ottimale

- L'algoritmo ottimale per la sostituzione delle pagine è noto come OPT o MIN
- Non presenta anomalia di Belady
- presenta la frequenza di page fault più bassa
- criterio adottato: si sostituisce la pagina che non verrà usata per il periodo di tempo più lungo
- Sfortunatamente non è applicabile in quanto non è possibile sapere a priori quando una pagina verrà richiesta in futuro

Least-recently used

- Simile all'algoritmo FIFO, ma anziché basare la scelta sul tempo di caricamento la basa sul tempo di utilizzo
- LRU è un'approssimazione di OPT, in cui si basa la stima del momento in cui una pagina sarà usata sulla base di quanto accaduto finora
- A ogni pagina si associa un marcatore temporale che corrisponde all'istante di ultimo utilizzo
- criterio: si sceglie la pagina usata meno di recente
- Proviamo ad applicare l'algoritmo sull'esempio visto per FIFO: 3 frame in RAM con sequenza dei riferimenti 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, ...

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, ...



Commenti

- LRU è un algoritmo molto considerato anche se poche architetture forniscono l'HW per applicarlo
- L'unico problema (per risolvere il quale occorre un supporto HW) è determinare la pagina da sostituire, infatti gli aggiornamenti del marcatore temporale causano un continuo rimescolio delle pagine nella sequenza
- Soluzione 1:
  - si aggiunge alla CPU un contatore incrementato a ogni riferimento alla memoria
  - associo a ogni elemento della tabella delle pagine un campo x mantenere il marcatore temporale
  - ogni volta che si accede a una pagina si aggiorna il suo marcatore usando il valore del contatore
- Problemi:
  - overflow del contatore
  - la ricerca della vittima nella tabella delle pagine è sequenziale

Commenti

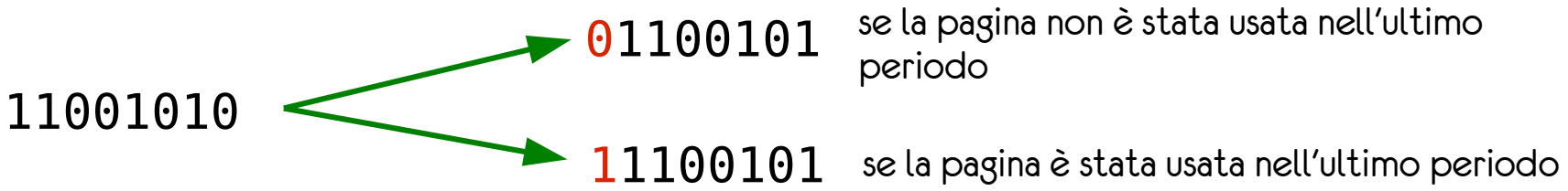
- Soluzione 2:
  - si mantiene uno stack di pagine
  - ogni volta che si fa riferimento a una pagina la si pone in cima allo stack (eventualmente togliendola da altra posizione nello stack)
  - la pagina in fondo allo stack è sempre quella usata meno di recente
- Altri commenti:
  - LRU non è soggetta ad anomalia di Belady

Approssimazione a LRU

- Poche architetture consentono di applicarel'LRU
- Alcune architetture consentono di applicarne un'approsimazione che si basa sull'uso del “bit di riferimento”
- anziché mantenere un contatore e un insieme di timestamp, si associa semplicemente a ogni elemento della tabella delle pagine un bit, che viene impostato a 1 ad ogni accesso (in lettura o in scrittura) alla pagina a cui fa riferimento
- all'inizio tutti i bit sono a 0
- dopo un po' alcuni bit saranno diventati 1
- manca l'informazione relativa a quando le varie pagine sono state usate
- esistono diversi metodi che si basano su questo supporto

Algo. con bit supplementare di riferimento

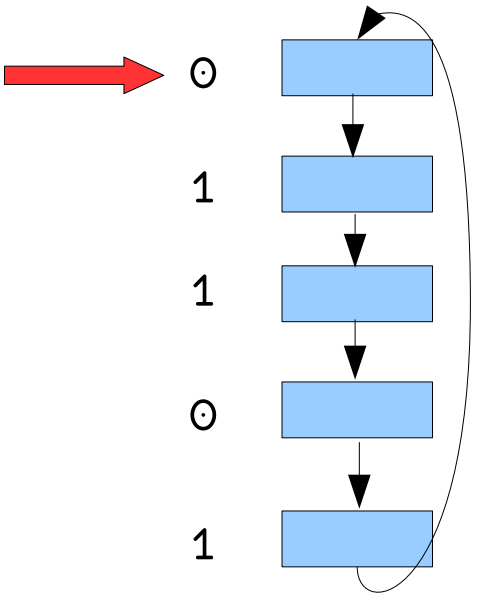
- È una tecnica di approssimazione della LRU in cui si associa a ogni elemento nella tabella delle pagine una serie di bit che realizzano dei registri a scorrimento
- A intervalli regolari un timer passa il controllo al SO, che sposta i bit nella sequenza traslandoli a destra di 1, copia il bit di riferimento nel bit più significativo della sequenza e scarta il bit meno significativo, es:



- quindi i bit di riferimento delle pagine vengono azzerati
- una pagina che ha il suo registro a 00000000 non è stata usata negli ultimi 8 periodi di tempo

Algo. di seconda chance

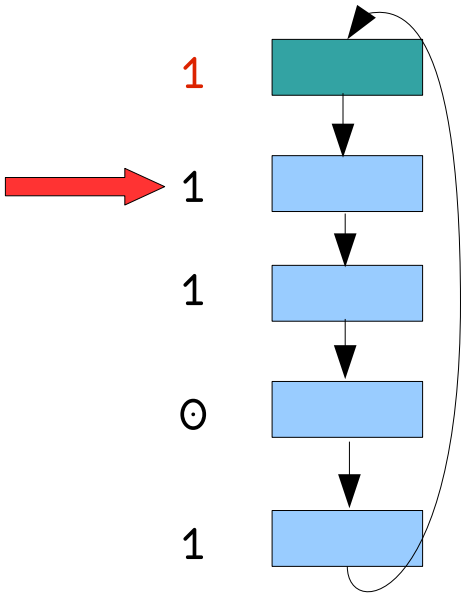
- è un algoritmo che unisce il metodo LRU all'approccio FIFO
- ogni pagina ha associato un bit di riferimento
- le pagine sono mantenute in una coda circolare FIFO



La freccia rossa indica il punto corrente di inizio della ricerca della vittima

La lista viene percorsa alla ricerca della prima pagina avente bit di riferimento a 0. Nell'esempio la pagina corrente diventa la vittima

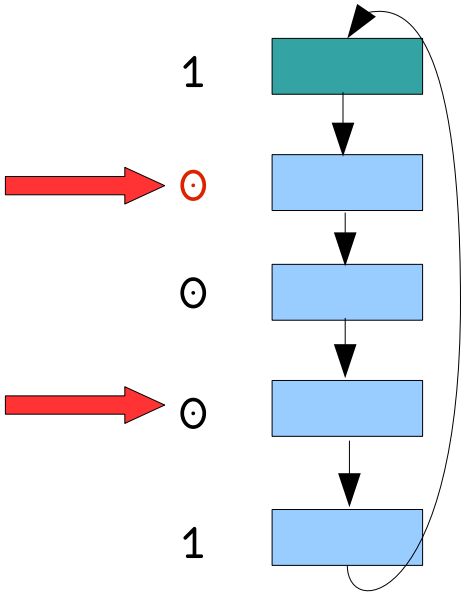
Algo. di seconda chance



Carico la pagina e le associo un bit di riferimento impostato ad 1

Quando diventerà necessario caricare una nuova pagina, la ricerca partirà dalla posizione successiva, in questo caso il bit di rif. ora vale 1

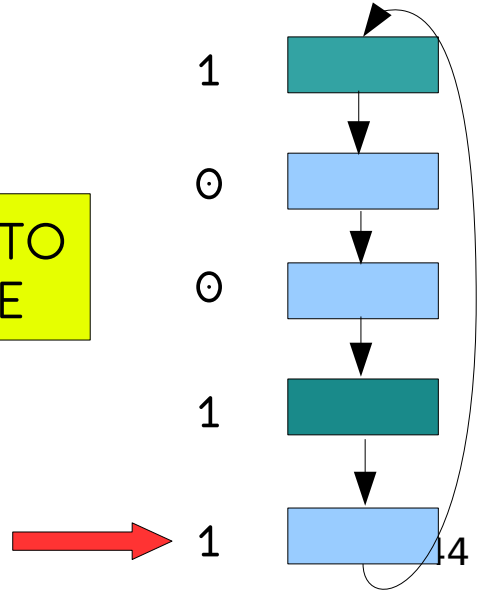
Algo. di seconda chance



Quando il bit di riferimento è impostato a 1: la pagina viene saltata ma il suo bit di riferimento viene azzerato.
Idem per la pagina successiva

A questo punto si incontra una pagina con bit di riferimento a 0 e la si sovrascrive mettendo il bit di riferimento a 1

RISULTATO FINALE



Cristina Baroglio

Osservazioni

- L'algoritmo di seconda chance è un'approssimazione dell'LRU in quanto mantiene l'informazione relativa alle pagine usate più di recente (bit di riferimento)
- la struttura circolare della coda è un modo per concedere un po' di tempo in RAM a ciascuna pagina: infatti una pagina con bit che passa da 1 a 0 non viene rimossa subito ma solo quando si tornerà alla sua locazione dopo aver percorso tutta la lista (avendo considerato e eventualmente scelto come vittima altre pagine)
- Se tutti i bit di riferimento sono impostati a 1, l'algoritmo di seconda chance si trasforma nell'algoritmo FIFO
- l'algoritmo può essere migliorato ...

Algo. di seconda chance migliorato

- Ogni pagina ha associata una coppia di bit <r, m>:
  - bit di riferimento: indica se la pagina è stata usata di recente
  - bit di modifica: indica se la pagina è stata modificata di recente
- Si individuano 4 classi di pagine:
  - <0, 0>: né usata di recente né modificata
  - <0, 1>: non usata di recente ma modificata
  - <1, 0>: usata di recente ma non modificata
  - <1, 1>: usata di recente e modificata
- Criterio di scelta: le classi sono ordinate sulla base del valore della coppia di bit, si sceglie una pagina appartenente alla classe in posizione inferiore fra quelle rappresentate in quel momento

00 < 01 < 10 < 11