

6 - Algoritmi sostituzione delle pagine

Algoritmi di Sostituzione delle Pagine

Quando si verifica un page fault, per far spazio alla pagina entrante il sistema operativo deve scegliere una pagina da rimuovere dalla memoria. Se la pagina da rimuovere è stata modificata mentre era in memoria, deve essere riscritta sulla memoria non volatile per aggiornare la copia su disco o SSD. Se invece la pagina non è stata modificata, la copia su disco o SSD è già aggiornata e non c'è bisogno di riscrivere. La pagina da leggere sovrascrive semplicemente la pagina sfrattata.

Sarebbe possibile prendere una pagina a caso da rimuovere a ogni page fault, ma le prestazioni del sistema migliorano molto se si sceglie una pagina non particolarmente utilizzata. Se si sfrattasse una pagina usata frequentemente, con ogni probabilità dovrebbe essere riportata in memoria a breve, con il risultato di un ulteriore sovraccarico.

Algoritmi di Sostituzione delle Pagine

1. Algoritmo Ottimale

Ciascuna pagina può essere etichettata con il numero di istruzioni da eseguire prima di ricevere un riferimento per la prima volta. L'algoritmo di sostituzione ottimale indica che deve essere rimossa la pagina con l'etichetta con il numero più alto. Tuttavia, questo algoritmo è irrealizzabile nel contesto di un sistema operativo che non può prevedere quando verrà fatto il prossimo riferimento a ciascuna pagina.

2. Not Recently Used (NRU)

Questo algoritmo usa due bit di stato, R (riferimento) e M (modifica), per classificare le pagine in quattro categorie. Alla base di NRU c'è la periodica pulizia del bit R per distinguere le pagine non recentemente referenziate da quelle referenziate recentemente.

Quando avviene un page fault, il sistema operativo ispeziona tutte le pagine suddividendole in 4 classi:

- Classe 0 ($R=0$ e $M=0$)
- Classe 1 ($R=0$ e $M=1$)
- Classe 2 ($R=1$ e $M=0$)
- Classe 3 ($R=1$ e $M=1$)

L'algoritmo NRU rimuove quindi una pagina a caso dalla classe non vuota col numero più basso.

3. First-In, First-Out (FIFO)

Questo algoritmo rimuove la pagina più vecchia in memoria quando si verifica un page fault. Tuttavia, FIFO può soffrire di ricorsione se una pagina vecchia è ancora frequentemente usata.

4. Second Chance

Una variante di FIFO che impedisce di scartare pagine frequentemente usate. Controlla il bit R delle pagine per decidere se rimuoverle. Se è 0 la pagina è inutilizzata e viene sostituita. Se è 1 il bit viene posto a zero e messo in fondo alla lista

5. Clock

Un miglioramento di Second Chance, Clock usa una lista circolare di frame di pagina. Quando si verifica un page fault, la pagina indicata dalla lancetta viene controllata; se $R=0$ si rimuove, altrimenti se $R=1$ viene azzerato e si passa alla successiva pagina. Si ripete finché non si trova una pagina con $R=0$.

6. Least Recently Used (LRU)

Questo algoritmo rimuove la pagina che non è stata referenziata per il periodo più lungo. L'implementazione richiede una lista ordinata delle pagine, con aggiornamenti frequenti per mantenere l'ordine corretto.

7. Simulazione del LRU via Software - NFU (Not Frequently Used)

NFU associa a ogni pagina un contatore che tiene traccia del numero di riferimenti. Alla base di NFU c'è il concetto di aging per aggiornare dinamicamente il contatore.

8. Working Set

Questo algoritmo si concentra sull'insieme di pagine che un processo sta attualmente utilizzando, mantenendo le pagine nel working set in memoria per evitare page faults.

9. WSClock

Una versione avanzata di Clock che integra le informazioni del working set per migliorare l'efficienza di sostituzione delle pagine.

Questi sono alcuni degli algoritmi di sostituzione delle pagine più utilizzati nei sistemi operativi moderni, ognuno con vantaggi e limitazioni basati sulle caratteristiche specifiche dell'applicazione e dell'hardware disponibile.