Algoritmi a stack
venerdì 19 maggio 2023 16:40

Algoritmi a stack

Metodo usato dai sistemi operativi attuali per la selezione della vittima.

Gli algoritmi a stack sono tutti gli algoritmi che non soffrono dell'anomalia di Belady. Noi siamo sicuri che, se noi implementiamo un algoritmo a stack, allora siamo tranquilli che scalando in alto il numero dei frame che abbiamo all'interno della nostra architettura abbiamo comunque un vantaggio effettivo. Quindi non è vero che paghiamo più Page Fault *come accadeva nell'algoritmo FIFO*.

Qual è la proprietà che ci dice se un algoritmo è a stack?
Quali sono le caratteristiche che ci dicono se un algoritmo è a stack oppure no?
Per ogni sequenza di accessi alla memoria logica r, dove r rappresenta l'insieme della pagine logiche che noi stiamo toccando, per un algoritmo a stack abbiamo esattamente questa situazione qua:

$$M(n+1,r) \supseteq M(n,r)$$

- per qualsiasi sequenza di riferimenti l'insieme delle pagine in memoria per *n* frames è sempre un sottoinsieme dell'insieme di pagine in memoria per *n+1* frames

L'insieme di queste pagine logiche che manteniamo all'interno della memoria se abbiamo n+1 frames è sicuramente un sovrainsieme dell'insieme delle pagine logiche M avendo n frames.
Quindi avere un frame in più ci aiuta a mantenere qualcosa in più, quindi esattamente ciò che avevamo prima più qualche altra cosa all'interno della RAM.
Se un algoritmo ha questa caratteristica, ha questa proprietà:

Proprietà
- non mostrano anomalia di Belady (la frequenza di page fault decresce all'aumentare del numero di frames)

Quindi passando da n ad n+1 frames, stiamo diminuendo l'insieme delle pagine dove un accesso potrebbe generare un page fault.
Stiamo favorendo il fatto che la frequenza di page fault vada effettivamente in qualche modo a decadere.

LRU è un algoritmo a stack

Ma di difficile implementazione.
Però andiamo a vedere come possiamo approssimare il comportamento di LRU con l'algoritmo dell'orologio.