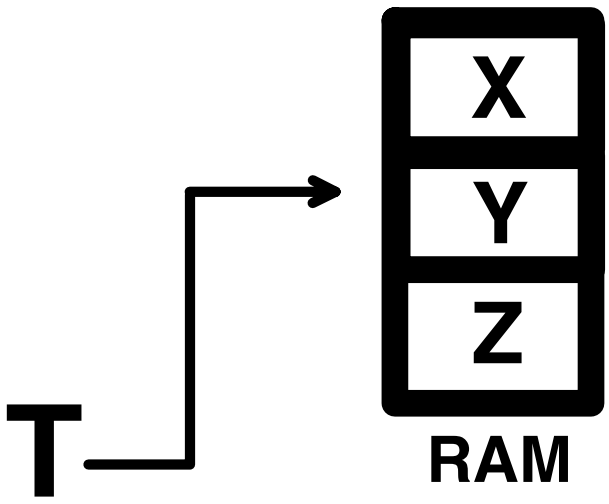
Algoritmo Ottimo

giovedì 18 maggio 2023 14:27

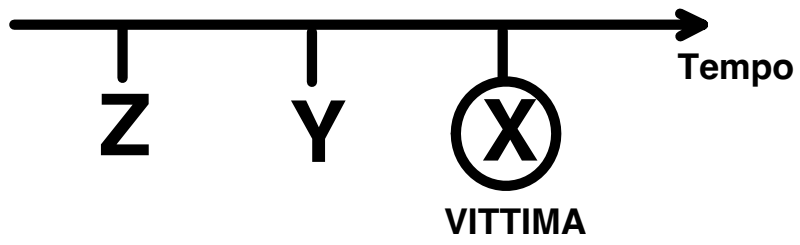
Algoritmo ottimo

Un algoritmo ideale molto interessante per la scelta delle vittime all'interno di un sistema di memoria virtuale basato su paginazione, è l'algoritmo ottimo.
La scelta di una vittima si ha quando non abbiamo più spazio all'interno della RAM, ciò vuol dire che tutti i frame sono occupati con delle pagine logiche, però dobbiamo materializzare all'interno della RAM qualcos'altro.
Perciò dobbiamo selezionare una vittima tra le pagine già materializzate che deve essere chiaramente portata fuori dalla RAM, ossia swapped out.
In questo algoritmo l'idea è che si seleziona la pagina alla quale ci si riferirà dopo il più lungo tempo, ovviamente rispetto alle altre pagine che sono già presenti all'interno della RAM.

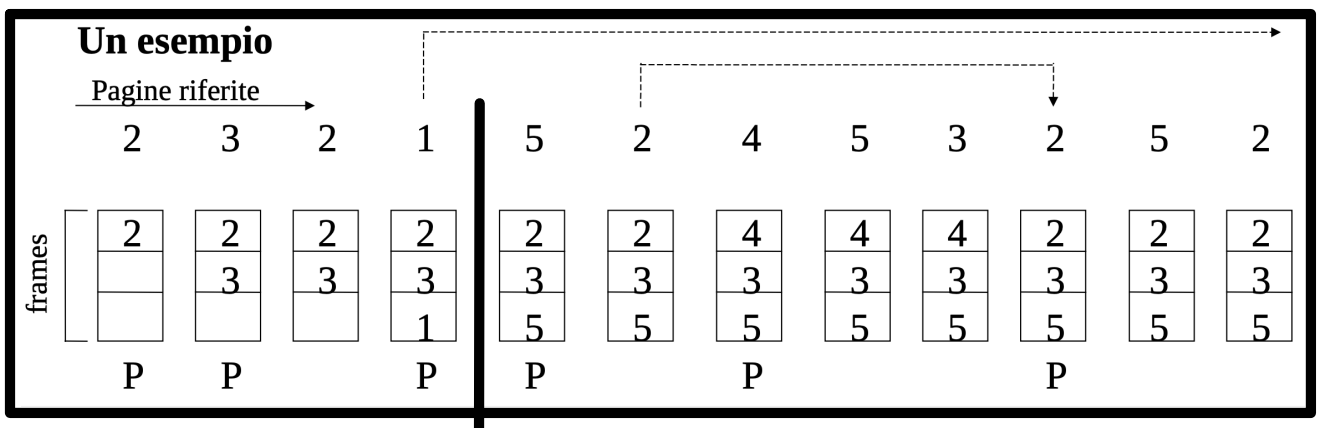
Se per esempio ho una RAM, dove all'interno ho rappresentato un frame occupato con una pagina X, un frame con una pagina Y e un frame con una Z, e devo portare all'interno di questa RAM piena un'altra pagina T, devo eliminare X O Y O Z: quella che eliminiamo dalla RAM è la pagina che non viene utilizzata per più lungo tempo.



Se considero una TimeLine e la pagina X viene utilizzata infondo, la pagina Y viene utilizzata in mezzo e la pagina Z viene utilizzata prima, chiaramente la vittima da selezionare è X.



Questa selezione avviene attraverso la conoscenza del futuro: bisogna conoscere il futuro circa di *quando avverranno gli accessi alle pagine*. **Quindi soltanto sapendo questi accessi alle pagine che verranno in futuro, potrò determinare, quale di queste pagine, verrà toccata dopo più lungo tempo.**
Questo ci porta a dire che questo è un algoritmo non applicabile nella realtà, ma è molto interessante perché utilizzabile come termine di paragone per valutare altri algoritmi: a valle del fatto che si è osservato quali sono gli accessi effettivi alle pagine, noi possiamo - utilizzando questa sequenza nota che è già accaduta - valutare cosa "SAREBBE" successo con questo algoritmo di sostituzione, e comparare cosa è "REALMENTE" successo avendo un altro algoritmo di sostituzione pragmatico che non ha necessità di conoscenza degli eventi futuri.
Esempio.
Supponiamo di avere 3 frames e supponiamo di considerare i riferimenti alle pagine indicando la pagina, a cui questi riferimenti, afferiscono.
Quando accediamo alla prima pagina 2, e questa non è mai stata materializzata all'interno della RAM, abbiamo sicuramente un PAGE FAULT e questo va a portarci la risoluzione di questa pagina (in termini di presenza) all'interno di un frame in RAM:



Poi dopo abbiamo l'accesso alla pagina 3, ancora un page fault e quindi il caricamento di questa pagina all'interno di uno dei frames. Poi abbiamo DI NUOVO la pagina 2 e non abbiamo nessun PAGE FAULT perché essa (la pagina) è già presente all'interno della RAM, poi accediamo alla pagina 1 abbiamo un page fault e la carichiamo nell'ultimo frame libero.

Fino al punto precedente, la sostituzione ancora non è avvenuta. Questi fault sono avvenuti semplicemente perché stavamo accedendo a delle pagine mai materializzate all'interno della RAM ed è stato necessario effettuare la materializzazione. Se noi poi vogliamo caricare la pagina 5 e quindi vogliamo considerarne il suo accesso in RAM - perché noi necessitiamo di averla in RAM per far si che la CPU possa eseguire le istruzioni per accedere a quelle informazioni - chiaramente dobbiamo selezionare una vittima tra la pagina 2, la pagina 3 e la pagina 1, ossia le pagine presenti nella RAM (nei corrispettivi frames) all'istante precedente.
Guardando la sequenza degli accessi ci rendiamo conto che la pagina che verrà acceduta più tardi di tutte le altre è la pagina 1, in particolare non viene proprio più acceduta, perciò eliminiamo la pagina 1 per far spazio alla pagina 5. Successivamente accediamo alla pagina 2 che è già presente in RAM e quindi non ci sono problemi.
Poi accediamo alla pagina 4 che non è in RAM e dobbiamo scegliere una vittima tra la pagina 2, la pagina 3 e la pagina 5: la vittima è la pagina 2, perché guardando la sequenza futura la 2 è quella che viene acceduta più tardi.

I primi page fault vengono detti "Sistemici", e sono page fault obbligatori nel momento in cui dobbiamo portare la RAM a lavorare per far sì che tutte le sue zone siano realmente utilizzate.