

Allocazione dei frame per i processi

All'atto della creazione del processo si pone il problema di quanti frame assegnare al processo e di quali pagine mettere dentro ai frame. La scelta più comune in questo caso è data dalla **pure demand paging**, ovvero si riempiono i frame fisici puramente su richiesta del processo. Questa scelta comporta, almeno in fase iniziale, un grosso numero di page fault dato dal fatto che i frame di memoria non sono inizializzati.

Supponendo di assegnare una certa quantità di frame ad un processo si sta scegliendo di dedicargli una certa quantità di RAM. Il massimo di memoria che si può concedere equivale all'intera RAM, ma si necessita di una limitazione della quantità di memoria assegnata al processo dettata dal grado di multiprogrammazione. Il minimo strutturale invece è dipendente dall'architettura del processore, ed equivale ad un certo numero di frame al di sotto del quale non è possibile eseguire un'istruzione. In generale vale che più memoria si assegna al processo e meno eventi di page fault genera la sua esecuzione.

Strategie

- **Equità** : se si pensa di assegnare una quantità di memoria prestabilita e divisa in maniera uguale a tutti i processi è una scelta che non tiene conto delle esigenze in termini di dimensione dei processi. Oltretutto le esigenze di memoria del processo sono cangianti nel tempo.
- **Proporzionale** : si può pensare di assegnare i frame di memoria in modo proporzionale alla dimensione (codice e dati) del processo. Si assegna al processo memoria ripartendo la quantità di memoria adattandola dinamicamente alle esigenze che cambiano in fase di esecuzione stessa. Al crescere delle esigenze, cresce la "taglia" del processo e di conseguenza il numero di frame che il processo si vede assegnati.

→ assegniamo $(a_i = s_i / S \times m)$ frame;

→ dove, $S = \sum s_i$;

- **Priorità** : un'altra scelta potrebbe basarsi sulla priorità del processo, infatti questa può influenzare il tipo e la qualità del servizio offerto.

Quando un algoritmo di sostituzione delle pagine effettua una scelta in senso **globale o locale**: ovvero il set di pagine su cui effettua la scelta può essere o tutte le pagine in memoria oppure un set ristretto alle sole pagine dedicate al processo. Un algoritmo che effettua una scelta in senso globale può portare alla penalizzazione di altri processi che si vedono sottratte delle pagine.

Trashing

Quando l'assegnazione di memoria ad un processo è al di sotto del **minimo strutturale**, esso viene sospeso perchè impossibilitato nell'esecuzione e si fa swapping su disco (scheduling a medio termine). Un po' sopra questa soglia c'è una zona rossa, detta zona di **trashing** nella quale l'esecuzione del processo è sofferente e vede un alto numero di page fault che ne rallenta fortemente l'esecuzione. Quando il trashing riguarda tutti i processi si parla di **sovraccarico del sistema** dovuto all'alto grado di multiprogrammazione, infatti in presenza di troppi processi si è costretti ad assegnare poca memoria ad ognuno di loro, rendendo la loro esecuzione "sofferente" dato l'alto numero di page fault.

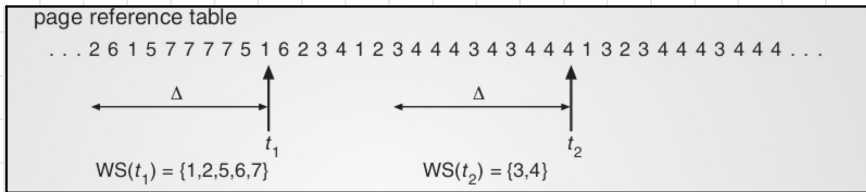
Si commisura la quantità di frame assegnati ad un processo in base alla "necessità del processo", e non alla loro taglia in termini di grandezza del codice o di dati. Quest'assegnazione è fortemente legata al concetto di località :

Concetto di **località** : è la porzione di codice e di dati necessari durante una determinata fase di vita del processo.

Il collegamento alla località nasce dal fatto che se un processo effettua accessi localizzati ad un certo numero pagine, sarà sufficiente tenere un numero di pagine fisso in memoria, pari alle esigenze di località del processo. In quel caso il processo genera difficilmente eventi di page fault. Bisognerebbe conoscere l'esigenza di **località** del processo in termini di pagine. Idealmente si dovrebbe allocare in RAM pari alle esigenze di località del processo.

Working Set

Per ogni processo si mantiene un working set, che vuole essere un'approssimazione delle esigenze di località del processo. Il working set si può definire formalmente a partire da un parametro **delta**, in particolare il working set rappresenta le pagine referenziate dal processo negli ultimi delta istanti di tempo. Il working set è un **insieme dinamico** cangiante in diversi delta (intervalli di tempo), sia in termini di cardinalità che di quali pagine contiene. Esso vuole essere un tentativo di **discretizzare la località del processo**. Durante l'impostazione del delta si deve tener conto che un delta piccolo non ricoprirebbe le esigenze di località mentre un delta troppo grande potrebbe comportare l'inclusione di pagine non più utilizzate e quindi portare ad una frammentazione della memoria.



Prevenzione

Si vuole tracciare il comportamento del processo per avere un feedback in termini di esigenze di pagine. Il working set, o meglio la sua cardinalità, dà informazioni su quante pagine allocare per il processo stesso. Avere numeri concreti sulle esigenze del processo, permette di poter prendere decisioni sull'assegnazione di memoria e di diagnosticare problemi :

- Se la cardinalità del working set inizia a crescere nei confronti dei frame assegnati (che risultano meno di quelli necessari) rappresenta un allarme utile per la prevenzione del **trashing**
- Si può anche diagnosticare in maniera preventiva il trashing totale di tutti i processi, ovvero il **sovraccarico di sistema**, valutando se la richiesta globale di pagine di tutti i processi, supera le capacità della RAM disponibile

Calcolo del working set

Il sistema operativo può solo approssimare il calcolo del working set usando un approccio simile all'aging (utilizzato per l'implementazione dell'NFU). Si hanno delle **IRQ periodiche** al termine delle quali, si effettua uno shift logico a destra di una posizione della rappresentazione binaria di un contatore che viene incrementato ogni volta che si fa riferimento ad una pagina. Questo significa che i bit di referenziamento più vecchi vengono spostati verso destra, e il **bit di referenziamento** più recente viene inserito nella posizione più significativa, così da pesare maggiormente lo stato nei cicli di clock più recenti. Si prevede pure un file di **log** che conserva lo storico del bit R in base al parametro delta, così da poter tenere traccia dei cambiamenti nel bit di referenziamento delle pagine nel corso del tempo.

Nel caso in cui l'algoritmo di sostituzione delle pagine sia l'aging si può sfruttare il contatore che presentano le varie pagine al posto di ricreare un meccanismo del tutto simile.

Page fault frequency

Esiste un modello differente per approssimare il trashing dei processi. La frequenza dei page fault (PFF) sarà bassa o nulla quando il processo è lontano dalla trashing, ovvero non soffre se ha sufficienti frame rispetto alla sua località. Se si monitora la PFF di ogni processo, si possono prendere decisioni sulla quantità di RAM da allocare per il processo. Questa strategia prevede la rilevazione di due **soglie di normalità** (inferiore e superiore) e se il PFF supera la soglia superiore significa che il processo sta per andare in sofferenza, ovvero le sue necessità sono cambiate e si necessita di assegnare dei frame liberi a quel processo. Si potrebbero anche assegnare dei frame occupati da altri processi nel caso in cui non si hanno dei frame liberi, ovviamente la scelta sul processo al quale si tolgono dei frame ricade su un processo la cui PFF è lontana dalla soglia di normalità.

I modelli di Working Set e PFF sono in relazione tra loro ed osservano fenomeni simili.