

Resident set

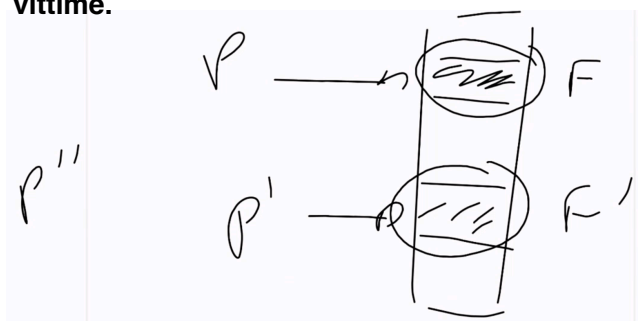
Il resident set è il numero dei frame che noi diamo in uso ad uno specifico processo e quindi che diamo in uso ad uno specifico address space.

Quando consideriamo un address space sappiamo che logicamente è composto da una zona user con pagine user e una zona kernel con pagine kernel, e tipicamente quando noi parliamo di resident set esso è la rappresentazione della quantità di memoria fisica che noi diamo in uso a questo processo associato a questo address space per ospitare le pagine logiche USER. Perché alla fine abbiamo detto che le pagine kernel sono comunque sempre in RAM.

- minore è la taglia del resident set, maggiore è la quantità di processi mantenuti in memoria (e quindi il grado di multiprogrammazione), inoltre il tempo di swapping viene ridotto
- se il resident set è troppo piccolo, la frequenza di page fault potrebbe essere inaccettabilmente alta

Ci sono schemi differenti:

Uno è l'allocazione fissa: tipicamente quando il processo nasce assegniamo un certo numero di frame in RAM che è fisso. La sostituzione delle pagine avviene nell'insieme dei frame scelto. Quindi applichiamo gli algoritmi di sostituzione solo all'interno dell'insieme dei frame. Quando ci serve P'' invece di andare ad utilizzare un altro frame all'interno della memoria, dobbiamo selezionare la vittima nell'insieme che avevamo dei frame, quindi selezioniamo o P o P' come vittime.



Questa è l'allocazione fissa.

L'allocazione variabile: il numero dei frames per un dato processo può variare durante l'esecuzione di ogni thread di quel processo.

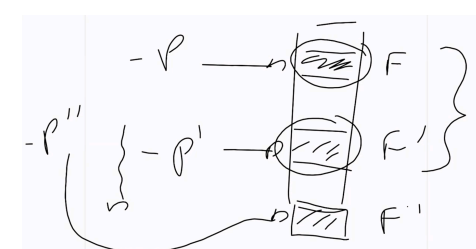
Allocazione fissa

- si assegnano al processo un numero di frames fisso, deciso alla sua attivazione (criticità della scelta della taglia)
- la sostituzione di pagina coinvolge solo frames del processo

Allocazione variabile

- il numero di frames per un dato processo può variare durante la sua esecuzione (migliore utilizzo della memoria, ma complessità di gestione superiore)
- la sostituzione di pagina può coinvolge frames di altri processi

A run-time riserviamo un altro frame per P'' per portare quest'ultimo all'interno della RAM.



In questo schema abbiamo ampliato il residence set dell'applicazione.

Allocazione mista

Periodicamente si rivaluta la taglia del resident set (e quindi ampliarla) nel momento in cui abbiamo necessità di diminuire il numero dei page fault nel caso in cui questo sia eccessivo.

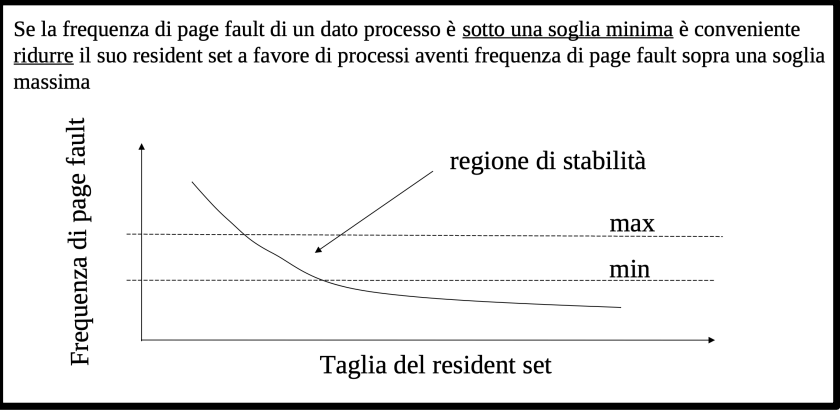
Andiamo a sostituire soltanto all'interno dei frames che avevamo allocato, ma periodicamente rivalutiamo la taglia del resident set, quindi il numero effettivo delle pagine fisiche. In modo tale che, se ci rendiamo conto che sono avvenuti troppi page fault, possiamo diminuire questo numero di page fault per lo specifico processo.

- quando un processo è attivato, si alloca un dato numero di frames, caricati tramite paginazione su richiesta o prepaginazione
- quando avviene un page fault, la pagina da sostituire viene selezionata tra quelle del resident set del processo
- periodicamente si rivaluta la taglia del resident set dei processi attivi, per favorire l'aumento di quella dei processi con meno località

Strategie di rivalutazione della taglia

- Working Set
- Frequenza di page fault

Questo viene ad essere attuato nel momento in cui cerchiamo di rivalutare la taglia utilizzando esattamente la frequenza dei page fault come indicatore se questa taglia debba essere migliorata oppure NO. Se per uno specifico processo l'obbiettivo è mantenere la sua frequenza di page fault sotto ad un minimo e non al di sopra di un massimo, e se questa cosa avviene, chiaramente dobbiamo cambiare il numero dei frame che riserviamo per quello specifico processo. Supponendo di avere all'interno del nostro S.O un algoritmo per la sostituzione delle pagine che sia efficace, aumentare il numero dei frame per quel processo ci tende a far diminuire dei page fault che quel processo ospiterà.



Quindi abbiamo una regione di stabilità dove quello che cerchiamo di fare è evitare di avere un numero di page fault che siano sotto un minimo ma non al di sopra di un massimo.

Working Set

Si definisce Working Set di un processo ad un certo istante di tempo T, un insieme $W(T, \Delta)$ che è costituito dagli ultimi Δ riferimenti di pagina per quel processo.

Quindi se guardo una timeline e mi metto al tempo T e guardo cosa sta succedendo per un certo processo P, e guardo indietro, riesco ad identificare Δ interazioni con le pagine a partire da questo tempo in cui io comincio ad osservare all'indietro le pagine. Questi Δ accessi alle pagine potrebbero essere caduti tutti all'interno della stessa pagina P oppure potrebbero essere sparsi su più pagine. Questa informazione è importante per capire se la mia applicazione ha bisogno di più pagine o di meno pagine, quindi ha bisogno di più frame o meno frame che mantengano queste pagine, per lavorare in maniera adeguata all'interno della memoria.

Se una pagina è stata usata di recente, probabilmente appartiene al Working Set, perché sarà stata riferita all'interno di questi Δ riferimenti che stiamo considerando.

Assumendo che il comportamento nel recente passato - quindi quello espresso tramite il working set - rappresenta ciò con cui lavorerò nel futuro, il working set approssima la località, e io dovrei ovviamente scegliere una quantità di frame da riservare (per una specifica applicazione) per mantenere il working set, in funzione del parametro Δ .

Se Δ è troppo grande, con il working set catturiamo anche la variazione di località: se io considero un Δ che è superiore alle due ore, tipicamente due ore fa stavo utilizzando cose diverse a ciò che sto utilizzando ora. Se Δ è troppo piccolo chiaramente la località può non venir catturata e tipicamente noi - a livello di sistema operativo - dovremmo ingrandire o ridurre il resident set in modo tale da ottenere una taglia pari al working set. Il problema è che dobbiamo catturare tutti e Δ i riferimenti a pagine di un dato processo, che non è semplice.

Se i Δ riferimenti sono mille e questi mille avvengono all'interno della stessa pagina, per catturare tutti e mille i riferimenti a questa pagina dovrei avere un sottosistema della CPU che mi permetta di eseguire un'attività in cui io incremento un contatore per esattamente ognuno di quegli accessi, e poi vado ad incrementare il contatore per quella specifica pagina, quindi per uno specifico frame su cui questa pagina cade. Non è semplice.

Si definisce Working set di un processo al tempo t , l'insieme $W(t, \Delta)$ costituito dagli ultimi Δ riferimenti di pagina per quel processo

- se una pagina è stata usata di recente, allora probabilmente appartiene al Working set
- assumendo il comportamento nel recente passato come rappresentativo dell'immediato futuro, il Working set approssima la località del processo
- la precisione del Working set dipende dal parametro Δ
 1. Se troppo grande, può venir catturata anche la variazione di località
 2. Se troppo piccolo, la località può non venir catturata completamente

Uso

- il sistema operativo dovrebbe ingrandire o ridurre il resident set in modo da ottenere una taglia pari al Working set

Problemi

- difficoltà pratica di scegliere il valore di Δ e di valutare dove cadano gli ultimi Δ riferimenti a pagine di un dato processo

