

NFU : not recently used

LRU è impraticabile, allora un'alternativa
via software: NRU

Terminiamo un counter in ogni voce della tabella delle
pagine, periodicamente il valore di R gira di
essere azzerato è sommato al counter.

Quando si deve fare uno swap prendiamo la pagina
col counter più basso.

L'informazione è ovviamente approssimativa

Problema: potrebbe privilegiare pagine usate molto
in passato ma poco recenti.
Aumenterebbe il page fault

AGING

È simile a NFU, ma risolve il problema.

- Il contatore al ciclo di clock ha shift a dx di 1 bit
il bit tutto a sx diventa R.

Il contatore diventa una storia: se è a 8 bit
abbiamo informazioni sugli ultimi 8 usi.

La scelta si basa sul valore del counter e
non su quanti 1 ci sono, ma 1 gira a sx
ha un peso maggiore.

L'idea è sempre rimuovere la pagina col counter
più basso.

L'informazione rimane comunque approssimativa perché
l'1 ci indica che la pagina è stata usata
ma non quanto recentemente.

Confronto Prestazioni

Aspettativa: più pagine ho meno fault ho

- metrica: numero di fault fissando un numero
di pagine in RAM

RAM: 3 pagine

Sequenza uso pagine:

7 0 1 2 0 3 0 4 2 3
0 3 2 1 2 0 1 7 0 1

OPT: algoritmo in cui sappiamo a priori di non
la pagina usata nel futuro più lontano

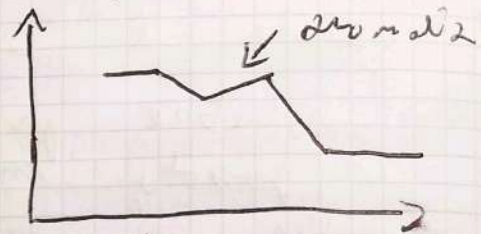
9 fault

FIFO: si sceglie la pagina, nella RAM, più vecchia

15 fault

Il FIFO presenta l'anomalia di Belady ovvero
aumentando il frame è possibile potrebbero aumentare
i fault

L'anomalia accade tutti i Fault
di algoritmi implementati
con il FIFO quindi anche
secondo chance e Clock



Anche NRU potrebbe andare in contro all'anomalia
se per caso tutte le pagine sono nella stessa
classe e la gestione interna della classe è FIFO

LRU: rimuovere le pagine meno usate di
recente (NRU e AGING sono approssimazioni)

12 fault

Non soffre dell'anomalia dato che gode di

proprietà di inclusione: l'insieme delle pagine
contenute avendo n frame è incluso nell'insieme
che si avrebbe con $n+1$ frame all'istante t

$$B_t(n) \subseteq B_t(n+1) \quad \forall t, n$$

NFU, AGING: godono della proprietà essendo simili
a LRU.

Nella pratica si usa il clock con un
aggiustamento che riguarda il bit di modifica
oppure l'aging

Allocazione dei frame

Per demand paging: paginazione per richiesta per
Un processo quando nasce è senza pagine dunque
all'inizio ci saranno non pochi fault, l'andamento
in pochi istanti comunque poi si stabilizza.
Quanti frame assegniamo a un processo?

- **Minimo strutturale**: si assegna il minimo di
pagine in cui sono contenute le strutture con
varie specifiche che servono al processo per la
esecuzione. Il minimo dipende dall'architettura.
Sotto questo minimo l'esecuzione è impossibile.

- **Massimo**: memoria libera

Strategie allocazione

- **Equa**: Ogni processo ha pari RAM dedicata
Poi dare problemi. Chi non ha bisogno
di quelle pagine ne ha troppe, chi invece ha
bisogno di più pagine ne ha poche.

- **Proporzionale**: danno RAM in base alla dimensione
del processo (valore che varia)
Supponiamo di avere il processo i di dimensione S_i
Dedichiamo $a_i = S_i / S \times m$ frame

- m : frame disponibili per i processi
- S : somma di tutti i frame usati $S = \sum S_i$
L'allocazione dunque è proporzionale alla taglia S_i
e dipende dal livello di multiprogrammazione
 S_i varia \rightarrow anche a_i
Il sistema tratta i processi senza bobare per
alle priorità.

Allocazione per priorità: dare più frame se il
processo ha priorità alta.

Allocazione locale: la vittima ha scegliere ^{una} pagina dello stesso processo che deve inserire una pagina

Allocazione globale: le pagine vengono viste tutte allo stesso modo

Quella che si usa nella pratica è la globale

Cosa succede se ci sono pochi frame per un processo?

- Sotto il minimo strutturale: l'esecuzione non è fattibile, si sospende il processo e si fa mapping in disco

- poco spa: thrashing

Il thrashing consiste nell'avere il minimo teorico (o poco spa) di frame per ~~non~~ troppo pochi, il numero di page fault sarebbe elevatissimo.

Tutti i processi in thrashing: intensa in memoria (eccessivo livello di multi-programmazione)

Di solito quando il sistema si rende conto che c'è un processo in thrashing lo svincola dagli altri frame.

Bisognerebbe trovare un modo per dare le frame necessarie per le necessità del processo

bediti: insieme di sezioni utili per un processo in un certo istante.

La bediti è legata alle necessità del processo in un certo istante

L'idea potrebbe essere tracciare in qz che modo la bediti e dare un numero di frame minimo per soddisfare la coerenza della bediti in un certo istante