

Algoritmi di rimpiazzo delle pagine

È necessario mantenere un certo numero di pagine libere

Problema: quale pagina rimuovere?

Meglio scegliere una pagina poco usata

Se la pagina è modificata va salvata

1

Algoritmo ottimale

Rimpiazzare la pagina che sarà usata nel futuro più remoto

Ottimale, ma irrealizzabile

Occorre usare una *stima*; quale?

- usare la pagina che è inutilizzata da più tempo
- usare la pagina che è in memoria da più tempo
- usare una pagina dal processo che ne ha di più

2

Not Recently Used page replacement algorithm

Each page has Reference bit, Modified bit

bits are set when page is referenced, modified

Pages are classified:

referenced	modified
0	0
0	1
1	0
1	1

NRU removes page at random from lowest numbered non empty class

3

FIFO page replacement algorithm

Maintain a linked list of all pages

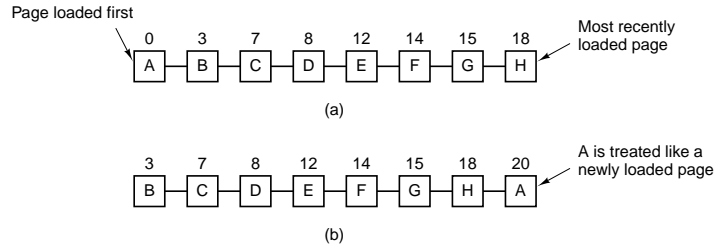
in order they came into memory

Page at beginning of list replaced

Disadvantage: page in memory the longest may be often used

4

Second chance page replacement algorithm

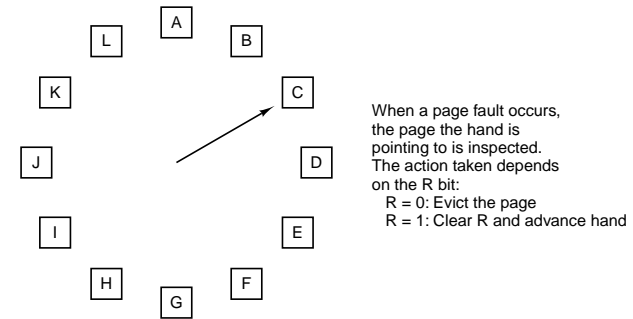


Le pagine sono in ordine di arrivo

Se la pagina in testa ha il bit *referenced*, viene rimessa in fondo alla coda

5

Clock page replacement algorithm



6

Least Recently Used (LRU)

Euristica: una pagina usata di recente verrà usata di nuovo
 ⇒ butta via la pagina non usata da più tempo

Conserva tutte le pagine in una lista

- aggiorna la lista ad ogni riferimento a memoria

Oppure: conserva un timestamp per ogni pagina

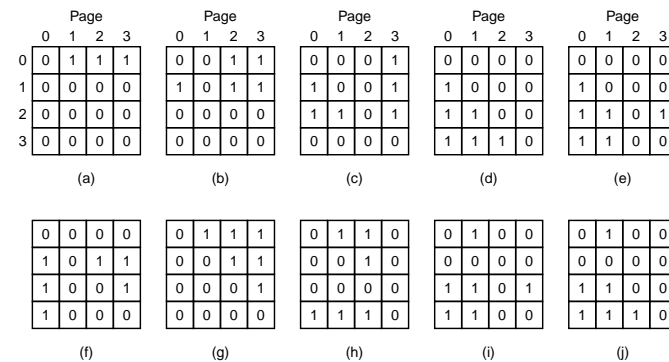
- aggiorna il timestamp ad ogni rif. a memoria

Impossibile da implementare senza supporto HW

In pratica questo algoritmo è poco usato

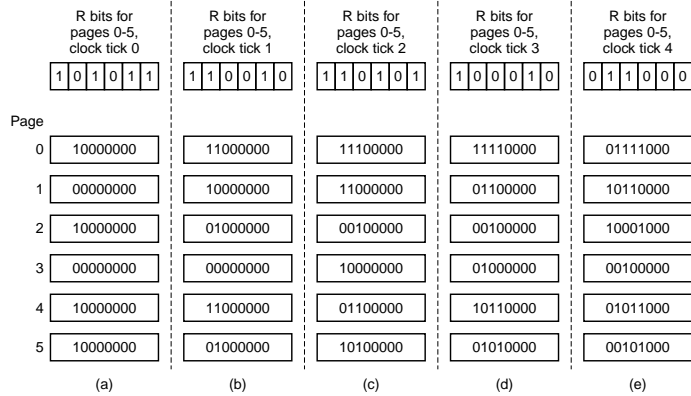
7

Altra implementazione di LRU



8

Approssimare LRU in software: Not Frequently Used (aging)



Per ogni pagina, un registro a scorrimento

- ad ogni clock tick, per ogni pagina:
 - shift right, e copia il bit R nel bit più significativo
- rimuovi la pagina con il valore più basso nel registro

9

Locality of Reference

The locality principle states that processes tend to reference memory in patterns, not randomly. Memory references tend to be clustered.

- If a page is referenced, it is likely that the same page will be referenced again in the near future (temporal locality)
- If a page is referenced, it is likely that nearby pages will also be referenced (spatial locality)

Si verifica *sperimentalmente* che molti programmi esibiscono questa località

Gli algoritmi che esibiscono località sono da preferire

10

Working Set theory (Peter Denning)

At any given time, each process has a working set of pages; that is, the pages that it is actually using.

The working set is usually a small portion of the entire address space of the process. The working set may change over time, triggering page faults, but these will not occur frequently.

The working set theory predicts that **if** the operating system can keep every process's working set in main memory, there will be few page faults and the system will perform well.

11

Thrashing

Se l'insieme dei working set dei processi in esecuzione è più grande della RAM disponibile

Vengono generati page fault ogni poche istruzioni

Il sistema passa il suo tempo nel memory manager

Non viene fatto lavoro utile

⇒ Thrashing (agitarsi in maniera scomposta)

12

Stima del working set

Working set: pagine accedute nell'intervallo di tempo $(t, t - \delta)$

wss_i : working set size for process i

$$D = \sum_i wss_i$$

obiettivo: tenere $D <$ numero di pagine fisiche

posso ridurre D swappando alcuni processi

13

Segmentazione + paginazione: MULTICS

Hardware: Honeywell 6000

Virtual address space: max 2^{18} segmenti, ciascuno di max 2^{16} parole

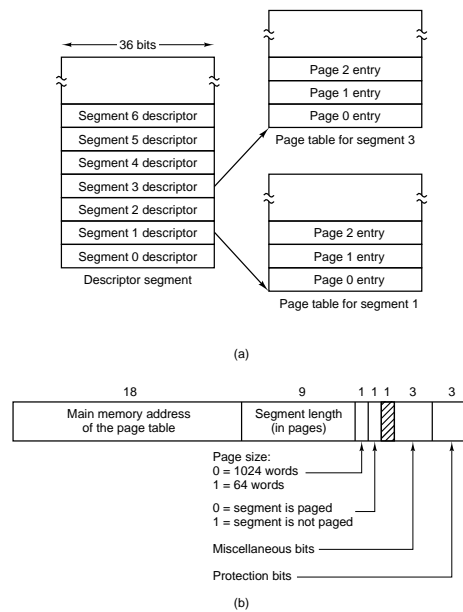
Ogni segmento è una memoria virtuale paginata in miniatura

Ogni processo ha una *segment table*, composta da *segment descriptors*

Ogni segmento può essere in memoria o no. Se in memoria:

- il segment descriptor punta alla *page table* del segmento

14

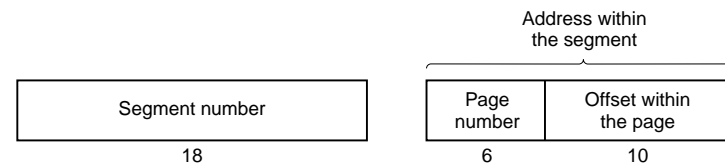


15

Indirizzi virtuali in Multics

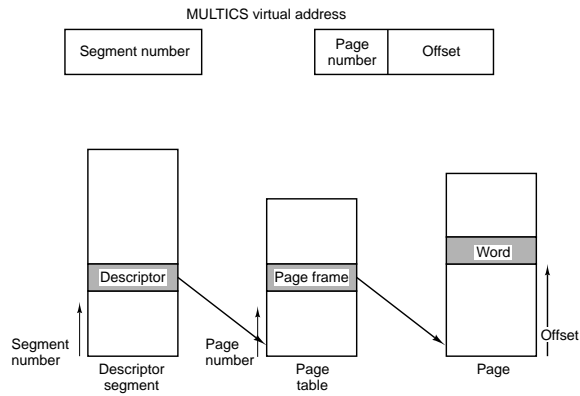
Un indirizzo è una coppia (segmento, indirizzo nel segmento)

L'indirizzo nel segmento è diviso in (page number, offset)



16

Traduzione di un indirizzo in Multics



17

La TLB in Multics

Comparison field		Page frame	Protection	Age	Is this entry used?
Segment number	Virtual page				
4	1	7	Read/write	13	1
6	0	2	Read only	10	1
12	3	1	Read/write	2	1
					0
2	1	0	Execute only	7	1
2	2	12	Execute only	9	1

Conserva le 16 traduzioni più recenti

18

Memoria segmentata e paginata: l'architettura Intel

Spazio di indirizzamento: max 2^{16} segmenti, ciascuno di 4GB

Local Descriptor Table (LDT) per processo

- testo, stack, dati, ...

Global Descriptor Table (GDT) unica

- segmenti di sistema

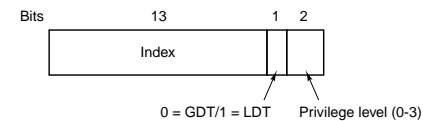
19

I registri del x86 che gestiscono i segmenti

CS: registro che contiene un *selector* per il segmento codice

DS: registro che contiene il selector per il segmento dati

Un *segment selector* è un registro di 16 bit



20

Il segment descriptor del x86

Il S.O. manipola CS e DS

