
Esercitazione E2

Memoria virtuale

Memoria virtuale 1

Si consideri un sistema a memoria virtuale con indirizzi virtuali a 32 bit, indirizzi fisici a 24 bit e pagine di 4K byte, e descrittori di pagina di 8 byte:

- a) di quante pagine sono costituiti rispettivamente lo spazio di indirizzamento virtuale e quello fisico?
- b) di quanti bit è costituito l'offset?
- c) di quante righe è costituita e quanti byte occupa la tavola delle pagine?
- d) Come è organizzata una tavola delle pagine a due livelli corrispondenti rispettivamente a gruppi di 12 e 8 bit?
- e) quante pagine occupa una procedura il cui codice occupa 9734 byte.

Memoria virtuale 1

a)

a) di quante pagine sono costituiti rispettivamente lo spazio di indirizzamento virtuale e quello fisico?

- Essendo gli indirizzi virtuali a 32 bit lo spazio virtuale è costituito da 2^{32} byte
- Le pagine sono da 4 K byte = 2^{12} byte, pertanto il numero di pagine virtuali è di:

$$2^{32}/2^{12} = 2^{20} = 1 \text{ M pagine}$$

- Essendo gli indirizzi fisici a 24 bit lo spazio fisico è costituito da 2^{24} byte
- Le pagine sono da 4 K byte = 2^{12} byte, pertanto il numero di pagine fisico è di:

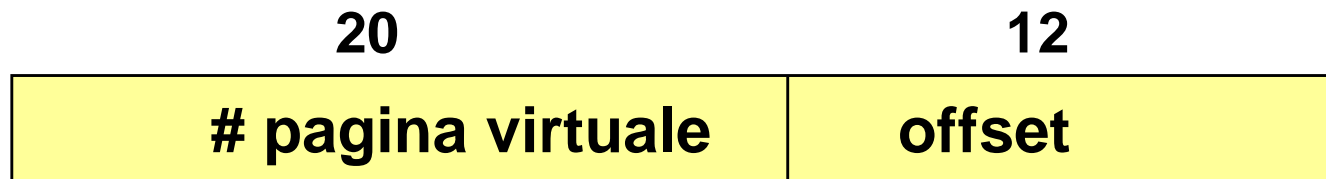
$$2^{24}/2^{12} = 2^{12} = 4 \text{ K pagine}$$

Memoria virtuale 1

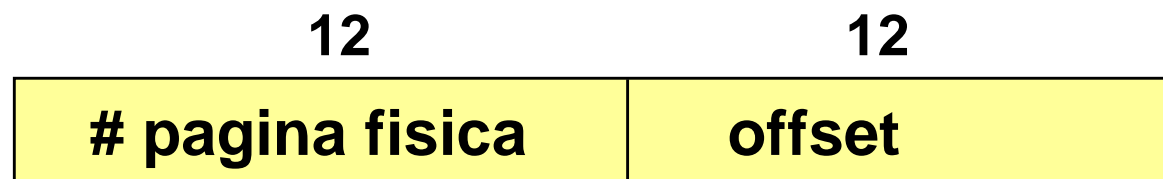
b)

b) di quanti bit è costituito l'offset?

- L'offset dipende solo dalla dimensione delle pagine
- Dato che le pagine sono da 4 K byte = 2^{12} byte, l'offset è di 12 bit
- Struttura dell'indirizzo virtuale:



- Struttura dell'indirizzo fisico



Memoria virtuale 1

c)

c) di quante righe è costituita e quanti byte occupa la tavola delle pagine?

- La tavola delle pagine ha un elemento per ogni pagina dello spazio di indirizzamento virtuale. Le pagine virtuali sono 2^{20}
- Ogni elemento della tavola è costituito da un descrittore di pagina.
- Nei dati del problema i descrittori di pagina sono di $8=2^3$ byte
- Pertanto la tavola delle pagine occupa complessivamente:

$$2^{20} \cdot 2^3 \text{ byte} = 2^{23} \text{ byte} = 8 \text{ Mbyte}$$

Memoria virtuale 1

d)

d) Come è organizzata una tavola delle pagine a due livelli corrispondenti rispettivamente a gruppi di 12 e 8 bit?

- Una tavola delle pagine a due livelli è costituita da:
 - Una *tavola di primo livello* in cui viene usato come indice il primo gruppo di bit (12 bit). Pertanto la tavola ha 2^{12} elementi.
 - Una *tavola di secondo livello* in corrispondenza a ciascun elemento della tavola di primo livello (nell'esempio 2^{12} tavole), in ciascuna delle quali viene usato come indice il secondo gruppo di bit (8 bit).
- Ciascun elemento della tavola di primo livello è un puntatore alla corrispondente tavola di secondo livello, ed ha le dimensioni di un indirizzo virtuale; nell'esempio 32 bit cioè 4 byte = 2^2 byte.
- Ciascun elemento della tavola di secondo livello è un descrittore, nell'esempio 8 byte = 2^3 byte.

d) Come è organizzata una tavola delle pagine a due livelli corrispondenti rispettivamente a gruppi di 12 e 8 bit?

— La *tavola di primo livello* ha 2^{12} elementi di 2^2 byte ciascuno, quindi occupa complessivamente:

$$2^{12} \cdot 2^2 \text{ byte} = 2^{14} \text{ byte} = 16 \text{ Kbyte}$$

— Ciascuna delle *tavole di secondo livello* ha 2^8 elementi di 2^3 byte ciascuno, quindi occupa complessivamente:

$$2^8 \cdot 2^3 \text{ byte} = 2^{11} \text{ byte} = 2 \text{ Kbyte}$$

— Dato che le tavole di secondo livello sono 2^{12} , esse occupano complessivamente:

$$2^{12} \cdot 2^{11} \text{ byte} = 2^{23} \text{ byte} = 8 \text{ Mbyte}$$

Memoria virtuale 1

e)

e) quante pagine occupa una procedura il cui codice occupa 9734 byte.

- La procedura viene allocata in pagine da 4 Kbyte = 4096 byte.
- Il numero di pagine occupate dalla procedura è dato da:

$$P = \lceil 9734/4096 \rceil = 3$$

- L'ultima pagina è solo parzialmente occupata. C'è uno sfrido pari a:

$$S = 3 \cdot 4096 - 9734 = 2554 \quad (\text{pari al } 2554/4096 \cong 62\%)$$

- Questo spreco corrisponde al fenomeno della *frammentazione interna*

Memoria virtuale 2

In un sistema a memoria virtuale si stima che la dimensione media dei processi sia di $s = 512$ KB. Assumendo che la dimensione dei descrittori di pagina è $e=16$ byte, confrontare l'occupazione di memoria dovuta alla tavola delle pagine ed alla frammentazione interna, nei seguenti casi:

- a) Pagine da $p=512$ byte
- b) Pagine da $p=4$ KB
- c) Pagine da $p=16$ KB
- d) Quale è la dimensione ottima (potenza di due!)?

- La dimensione ottima delle pagine costituisce un compromesso tra:
 - Dimensione della tavola delle pagine : cresce col numero di pagine utilizzate, e quindi col diminuire della dimensione delle pagine.
 - Spreco dovuto a frammentazione interna : cresce con la dimensione delle pagine; è pari a mezza pagina per ogni oggetto allocato.
- Per valutare l'overhead complessivo sia:
 - **s** : dimensione media del processo in byte
 - **p** : dimensione media della pagina in byte
 - **e** : dimensione del descrittore in byte

$$\text{overhead} = s/p \cdot e + p/2$$

- Il primo termine rappresenta l'occupazione nella tavola delle pagine relativa ad un processo
- Il secondo termine è lo sfrido attribuibile al processo: metà dell'ultima pagina allocata al processo

Memoria virtuale 2

ter

- La dimensione ottima delle pagine si ottiene, per dati valori dei parametri **s** ed **e**, trovando il valore di **p** che minimizza l'overhead:

$$p_{opt} = \sqrt{2 s e}$$

- Nel caso **s** = 512 Kbyte ed **e** = 16 byte, per le varie dimensioni di **p** le due componenti dell'overhead sono:
 - **p** = 512 byte: tavola=16 Kbyte, sfrido=256 byte, totale=16.25 Kbyte
 - **p** = 4 Kbyte: tavola=2 Kbyte, sfrido=2 Kbyte, totale=4 Kbyte
 - **p** = 16 Kbyte: tavola=512 byte, sfrido=8 Kbyte, totale=8.5 Kbyte
- Il valore ottimo (teorico) per la dimensione di pagina è:

$$p_{opt} = \sqrt{2 s e} = \sqrt{2 \cdot 2^{19} \cdot 2^4} = 2^{12} = 4 \text{ Kbyte}$$

Memoria virtuale 3

In un sistema a memoria virtuale occorre scegliere tra le seguenti pagine quale fare uscire dalla memoria centrale:

Pagina	Entrata	Riferita	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

Quale pagina sceglie ciascuno dei seguenti algoritmi?

- FIFO
- LRU
- NRU
- Second chance

Memoria virtuale 3: FIFO

Pagina	Entrata	Riferita	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

- Conta solo l'ordine di entrata
- Le pagine vengono scelte nell'ordine:
 - 3: entrata all'istante 110
 - 0: entrata all'istante 126
 - 2: entrata all'istante 140
 - 1: entrata all'istante 230

Memoria virtuale 3: LRU

Pagina	Entrata	Riferita	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

- Vale la vecchiaia dell'ultimo riferimento
- Le pagine vengono scelte nell'ordine:
 - 1: riferita all'istante 265
 - 2: riferita all'istante 270
 - 0: riferita all'istante 280
 - 3: riferita all'istante 285

Algoritmo NRU

- NRU (Not Recently Used)
- Ogni pagina ha un *Reference bit* e un *Modified bit*
- Le pagine vengono classificate
 1. Non referenziate e non modificate
 2. Non referenziate e modificate
 3. Referenziate e non modificate
 4. Referenziate e modificate
- La pagina da buttare viene selezionata random dalla classe non vuota più bassa

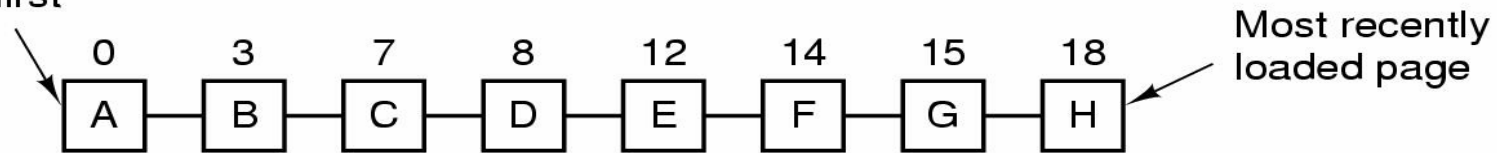
Memoria virtuale 3: NRU

Pagina	Entrata	Riferita	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

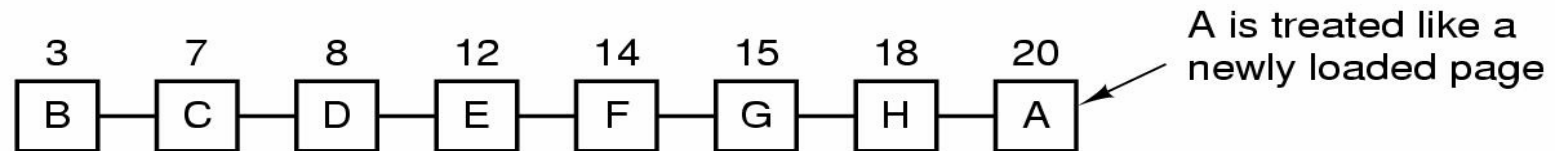
- Le pagine vengono scelte nell'ordine:
 - 2: non referenziata e non modificata
 - 1: non referenziata ma modificata
 - 0: referenziata e non modificata
 - 3: referenziata e non modificata

Algoritmo 'second chance'

Page loaded first



(a)



(b)

- Lista gestita FIFO: in base all'ordine di entrata
- Se la pagina in testa ha $R=0$ viene buttata fuori
- Se la pagina in testa ha $R=1$ viene messa in fondo alla coda con $R=0$
- Si passa alla successiva finché non se ne trova una con $R=0$

Memoria virtuale 3: second chance

Pagina	Entrata	Riferita	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

- Le pagine vengono esaminate in ordine FIFO:
 - 3: ha $R=1$ viene messa in coda con tempo 285 e $R=0$
 - 0: ha $R=1$ viene messa in coda con tempo 286 e $R=0$
 - 2: ha $R=0$ è viene espulsa
- Alle pagine 3 e 0, in quanto 'virtuose' è stata data una *second chance*
- 2 non era 'virtuosa' ed è stata punita subito.

Memoria virtuale 4

Un processo genera la seguente sequenza di riferimenti a pagine virtuali:

1, 2, 2, 4, 4, 1, 5, 8, 2, 5, 3, 4, 5, 3, 4, 1, 2, 3, 4, 5

Calcolare il *numero* e la *sequenza* di page fault che si verificano usando i seguenti algoritmi di rimpiazzamento:

- a) Ottimo, con 4 page frame
- b) FIFO, con 4 page frame
- c) LRU, con 4 page frame
- d) Working set con finestra k pari a 6

Memoria virtuale 4

a)

Algoritmo ottimo															7 page faults				
1	2	2	4	4	1	5	8	2	5	3	1	4	5	3	4	2	3	4	5
1	2	2	4	4	4	5	8	8	8	3	3	4	4	4	4	4	4	4	4
	1	1	2	2	2	4	5	5	5	5	5	3	3	3	3	3	3	3	3
			1	1	1	2	2	2	2	2	2	5	5	5	5	5	5	5	5
						1	1	1	1	1	1	2	2	2	2	2	2	2	2
P	P		P			P	P			P		P							

Memoria virtuale 4

b)

Algoritmo FIFO															11 page faults				
1	2	2	4	4	1	5	8	2	5	3	1	4	5	3	4	2	3	4	5
1	2	2	4	4	4	5	8	8	8	3	1	4	5	5	5	2	3	3	3
	1	1	2	2	2	4	5	5	5	8	3	1	4	4	4	5	2	2	2
			1	1	1	2	4	4	4	5	8	3	1	1	1	4	5	5	5
						1	2	2	2	4	5	8	3	3	3	1	4	4	4
P	P		P			P	P			P	P	P	P			P	P		

Memoria virtuale 4

c)

Algoritmo LRU														10 page faults					
1	2	2	4	4	1	5	8	2	5	3	1	4	5	3	4	2	3	4	5
1	2	2	4	4	1	5	8	2	5	3	1	4	5	3	4	2	3	4	5
	1	1	2	2	4	1	5	8	2	5	3	1	4	5	3	4	2	3	4
			1	1	2	4	1	5	8	2	5	3	1	4	5	3	5	2	3
						2	4	1	1	8	2	5	3	1	1	5	4	5	2
P	P		P			P	P	P		P	P	P				P			

Memoria virtuale 4

d)

Working set K=6													8 page faults						
1	2	2	4	4	1	5	8	2	5	3	1	4	5	3	4	2	3	4	5
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	2	3	3	3
			4	4	4	4	4	4	4	3	3	3	3	4	4	3	4	4	4
						5	5	5	5	5	5	4	4	5	5	4	5	5	5
							8	8	8	8	8	5	5			5			
												8							
P	P		P			P	P			P		P				P			