

SISTEMI OPERATIVI
prova di esame del 7 aprile 2003

- 1) Un video server gestisce stream a 30 frame/s. Ciascun frame richiede 4ms di elaborazione su una CPU a 500MHz. I frame sono registrati su disco in blocchi di 256kB che contengono 15 frame. Nella configurazione attuale il server dispone di 6 CPU a 1GHz, e di 10 dischi con tempo di accesso medio pari a 15ms ed una velocità di trasferimento di 100MB/s.
- determinare se nella configurazione data l'applicazione è CPU-bound oppure I/O-bound, calcolando anche il massimo numero di stream S_{max} che il server è in grado di gestire, supponendo che le utilizzazioni della CPU e dei dischi non possano mai superare rispettivamente l'80% ed il 40%;
 - determinare se e come varia S_{max} con una organizzazione dei dischi in blocchi da 60 frame e 1MB;
 - determinare la dimensione di memoria centrale necessaria per allocare doppi buffer per tutte S_{max} le stream? nel caso b);
 - determinare il volume di traffico tra dischi e memoria nei casi a) e b)

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

1, 2, 3, 1,4, 1, 2, 5, 4, 3, 6, 8, 3, 8, 6, 2, 4, 1, 2, 6, 5, 8

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

- ottimo con 3 page frame;
- FIFO con 3 page frame;
- LRU con 3 page frame;
- Working Set con finestra k pari a 3.

- 3) Con riferimento alla **gestione della memoria virtuale**, indicare quali tra le seguenti affermazioni sono corrette:

- l'algoritmo di rimpiazzamento ottimo si basa sulla conoscenza dettagliata dei soli riferimenti passati;
- località spaziale significa forte probabilità di fare riferimento a indirizzi contigui in tempi vicini;
- una tavola delle pagine a due livelli ha una dimensione complessiva inferiore ad una ad un livello solo;
- la segmentazione presenta il problema della frammentazione esterna;
- la dimensione dello spazio di indirizzamento virtuale dipende dalla memoria installata;
- con l'algoritmo del working set il numero di page frames assegnate ad un processo diminuisce all'aumentare della località;
- le istruzioni macchina fanno riferimento ad indirizzi virtuali;

- 4) Con riferimento alla **gestione dei processi e dei thread**, indicare quali delle seguenti affermazioni sono corrette:

- la creazione di un thread è meno costosa della creazione di un processo;
- i processi I/O bound tendono a consumare tutto il quantum di CPU a loro concesso;
- un processo che esegue una UP(S) quando $S=0$ si entra in uno stato di blocco;
- nello scheduling RR con n processi, ciascun processo riceve una frazione di CPU pressappoco pari a $1/n$;
- condizione sufficiente al verificarsi del deadlock è che sia possibile assegnare risorse in uso esclusivo;
- lo scheduling SSTF minimizza la varianza del tempo medio di risposta;
- thread diversi dello stesso processo non possono avere risorse comuni;

- 5) Con riferimento alla **gestione della sicurezza** dire quali tra le seguenti affermazioni sono corrette:

- nella cifratura con RSA e DES, DES è usato solo per lo scambio delle chiavi;
- i file non eseguibili (es. file MS word) non possono mai contenere virus;
- l'amministratore di un sistema UNIX non può conoscere le password degli utenti;
- occorre la chiave privata del firmatario per verificare l'autenticità della firma digitale;
- nella classificazione dell'orange book B è la classe di sicurezza più alta;
- nella firma digitale il firmatario firma il digest con la sua chiave pubblica;
- i virus non sempre modificano la dimensione dei file infettati;

- 6) Con riferimento all'organizzazione del **file system**, dire quali tra le seguenti affermazioni sono corrette:

- è possibile avere più file system sullo stesso disco;
- nel file system di Unix tutti gli utenti del gruppo dell'owner hanno gli stessi diritti dell'owner del file;
- per deframmentare un disco occorre prima riformattarlo;
- nella gestione della disk cache la politica *write through* garantisce un migliore allineamento tra cache e disco rispetto alla politica *write back*;
- l'algoritmo dell'ascensore è migliore di quello SSF in termini di valor medio del tempo di accesso;
- un *i-node* occupa diversi KB;
- lo scheduling dei dischi SSF sfrutta la conoscenza della coda di richieste di accesso;

ESERCIZIO 1 – Video Server

Il numero massimo di stream gestibili dalle CPU è:

$$S_{\max} = \frac{U_{CPU}}{t_u} \times N^{\circ}cpu$$

Considerando il fatto che la frequenza do CPU è raddoppiata, il tempo di utilizzazione di ciascuno stream è dato da:

$$t_u = \frac{t_{el} \times n^{\circ} frame_per_stream}{2} = \frac{30 \times 4ms}{2} = 60ms$$

$$S_{CPU} = \frac{800ms}{60ms} \times 6 = 80stream$$

per quanto riguarda i dischi, sapendo che il tempo di trasferimento di un blocco è $\frac{256kB}{100MB/s} \approx 2.5ms$ e che per ogni stream occorrono 2 blocchi, il tempo di utilizzazione su dischi di ogni stream è:

$$t_u = 2 \times (t_{tr} + t_A) \cong 35ms$$

ed il numero di stream gestibili dai dischi è:

$$S_{disk} = \frac{U_{DISK}}{t_u} \times N^{\circ}disk = \frac{400ms \times 10}{35ms} = 114stream$$

Il numero massimo di stream gestibili è quello delle CPU ($S_{\max}=80$), che essendo saturate, determinano un processo CPU-bound.

È possibile evitare di svolgere nuovamente tutti i calcoli, facendo attenzione al fatto che potenziando i dischi non ci sono cambiamenti per quanto riguarda le stream, poiché sarebbero dovute essere potenziate le CPU, quindi **Smax resta pari ad 80**.

Per allocare i dati in doppi buffer occorre una quantità di memoria pari a:

$$D_{memoria} = 2buffer \times 1MB \times 80stream = 160MB$$

Nel caso a) il volume di traffico è:

$$V = 2blocchi \times \frac{1}{4}MB \times 80stream = 40MB/s$$

anche nel caso b) il volume di traffico è:

$$V = \frac{1}{2}blocchi \times 1MB \times 80stream = 40MB/s$$

ESERCIZIO 2 – algoritmi di rimpiazzamento

ottimo: viene sostituita la pagina che verrà richiamata più tardi, rispetto alle altre pagine già presenti in memoria. in caso di più possibilità verrà eliminata l'ultima pagina della pila.

1	2	3	3	4	4	4	5	5	3	6	8	8	8	8	2	4	1	1	1	5	8	
x	1	2	2	2	2	2	4	4	4	3	6	6	6	6	8	2	2	2	2	2	1	5
	x	1	1	1	1	1	2	2	2	2	3	3	3	3	6	6	6	6	6	6	2	1
		x		x			x		x	x	x				x	x	x				x	x

13 page fault

FIFO: le pagine sono in ordine di chiamata e viene sostituita l'ultima della pila.

1	2	3	3	4	1	2	5	4	3	6	8	8	8	8	2	4	1	1	6	5	8	
x	1	2	2	3	4	1	2	5	4	3	6	6	6	6	8	2	4	4	4	1	6	5
	x	1	1	2	3	4	1	2	5	4	3	3	3	3	6	8	2	2	4	1	6	
		x		x	x	x	x	x	x	x	x				x	x	x			x	x	x

17 page fault

LRU: le pagine vengono di volta in volta riorganizzate, mettendo in cima l'ultima pagina chiamata, nel caso di una pagina già presente, questa torna in cima alla pila; viene sostituita la pagina alla fine della pila.

1	2	3	1	4	1	2	5	4	3	6	8	3	8	6	2	4	1	2	6	5	8	
x	1	2	3	1	4	1	2	5	4	3	6	8	3	8	6	2	4	1	2	6	5	
	x	1	2	3	1	4	1	2	5	4	3	6	6	3	8	6	2	4	1	2	6	
		x		x		x	x	x	x	x	x				x	x	x			x	x	x

16 page fault

Working Set (k=3): una finestra costituita delle prime k pagine chiamate, viene "shiftata" di un passo alla volta, ogni volta entrerà un nuovo elemento in sostituzione di quello posto all'inizio della finestra; se la pagina che entra non è già presente nella finestra si genera una page fault.

			→																			
1	2	3	1	4	1	2	5	4	3	6	8	3	8	6	2	4	1	2	6	5	8	
x	x	x	x	x																		

20 page fault