Sistemi Operativi — A.A. 2006-2007, prova scritta del 10 settembre 2007

Usa questa pagina per la brutta, staccala, non consegnarla.

Sistemi Operativi — A.A. 2006-2007, prova scritta del 10 settembre 2007

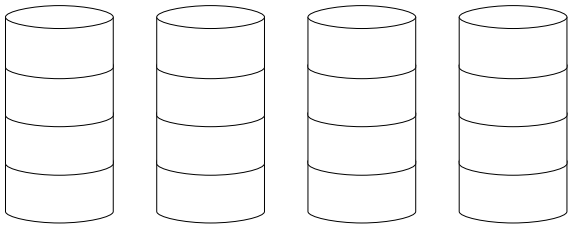
Usa questa pagina per la brutta, staccala, non consegnarla.

Cognome:_____ **Nome:** _____ **Matricola:**_____

Sistemi Operativi — A.A. 2006-2007, prova scritta del 10 settembre 2007

Libri e appunti chiusi. Vietato comunicare con chiunque. Vietato l'uso di cellulari, calcolatrici, palmari e affini. Tempo a disposizione: 60 minuti.

1. Considera 4 dischi in raid 5. Mostra come sono disposti i blocchi logici nei blocchi fisici compilando il seguente schema (indica con P(x,y,z) la parità per i blocchi logici x, y, z).



Supponi che a partire da una situazione di cache vuota si debbano leggere i blocchi logici 1, 5 e 8 e per leggere un blocco e' necessario un tempo R, entro quanto tempo l'insieme delle tre letture sara' ultimato? perche'?

Supponi che a partire da una situazione di cache vuota si debbano scrivere i blocchi logici 1, 5 e 8. Quante operazioni di lettura sono necessarie? Quante di scrittura? Perche'?

Operazioni di lettura: Operazioni di scrittura:

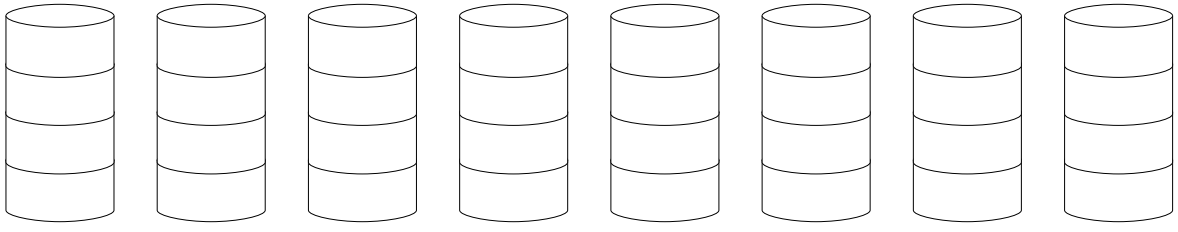
Spieagazione:

Come sopra ma i blocchi da scrivere sono 4, 5 e 6, supponi che l'I/O scheduler veda le richieste contemporaneamente e ottimizzi.

Operazioni di lettura: Operazioni di scrittura:

Spieagazione:

Considera 8 dischi in raid 15. Supponi i blocchi logici numerati in sequenza a partire da 1. Indica nel seguente schema come sono raggruppati i dischi, per ciascun blocco fisico quale è il numero del blocco logico contenuto e dove sono situati i blocchi di parità (es. indica con P(x,y,z) la parità per i blocchi logici x, y, z).



Sistemi Operativi — A.A. 2006-2007, prova scritta del 10 settembre 2007

2. Considera un sistema con architettura del kernel "execution within user process". In tale sistema sono presenti tre processi: A e B sono I/O bound e C è puramente cpu-bound. Lo scheduler è round robin con quanto q. A è inizialmente in testa alla coda ready seguito da B e C. L'I/O burst di A dura $1.2q$ e qullo di B dura $1.3q$. Il cpu burst di A, il cpu busrt di B, i tempi di dispatching e di esecuzione di system call e dell'interupt handler sono tutti molto piccoli e trascurabili rispetto a q.

Il processore esegue di volta in volta A, B, C, mode switching, dispatcing, system call e interrupt handlers. Mostra schematicamente, nella seguente tabella, l'ordine con cui tali attività vengono eseguite (una sola croce per ciascuna colonna). Indica anche quali processi sono running, quali ready e quali bloccati in ciascun istante come indicato nell'esempio.

tempo →

proc. in user mode	A	x																												
	B																													
	C																													
mode switch			x																											
kernel mode	dispatching				x																									
	system call i/o			x																										
	interrup handler per il timer (q scaduto)																													
	interrup handler per i/o																													
stati processi	running	A	A	A																										
	ready	B C	B C	B C																										
	in blocco				A																									

3. Considera un processo di 5 pagine a cui sono stati assegnati 4 frame. Descrivi come deve essere fatta la stringa di rifiermimenti a memoria del processo in modo che

- sia l'algoritmo FIFO che l'algoritmo CLOCK diano page fault negli stessi istanti sostituendo le stessa pagine
- la stringa sia infinita (periodica)
- la stringa preveda accessi a tutte le pagine un infinito numero di volte
- non vi sia un fault ad ogni accesso.

Cognome:_____ **Nome:** _____ **Matricola:**_____

Sistemi Operativi — A.A. 2006-2007, prova scritta del 10 settembre 2007

4. Mostra lo schema di segmentazione paginata.

Considera una architettura stile Pentium: pagina di 4 KB, paginazione a due livelli, pte 4 byte, root page table sempre in memoria. Il frammento di codice assembly mostrato è composto da 2 istruzioni di 5 byte l'una che vengono eseguite consecutivamente. Calcola quanti page fault può generare al più ciascuna istruzione durante l'esecuzione del frammento in questione nelle fasi di fetch e di esecuzione. Considera le istruzioni eseguite di seguito e supponi che le pagine caricate dalle istruzioni precedenti permangano residenti durante l'esecuzione delle istruzioni successive.

Indirizzo	istruzione	Page faults dovuti a parti di page table non residenti		Page faults dovuti a codice o dati non residenti	
		fetch	execute	fetch	execute
`0x007ffff6`	carica nel registro A 4 byte a partire da `0x007feffe`				
`0x007ffffB`	scrivi il contenuto di A in 4 byte a partire da `0x007fdffe`				

Sistemi Operativi — A.A. 2006-2007, prova scritta del 10 settembre 2007

5. Illustra il ruolo degli interrupt nell'ambito delle varie fasi della gestione di un page fault.