

Sistemi Operativi — A.A. 2005-2006, prova scritta del 21 settembre 2006

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

Sistemi Operativi — A.A. 2005-2006, prova scritta del 21 settembre 2006

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

Cognome: _____ Nome: _____ Matricola: _____

Sistemi Operativi — A.A. 2005-2006, prova scritta del 21 settembre 2006

- Libri e appunti chiusi. Vietato comunicare con chiunque. Vietato l'uso di cellulari, calcolatrici, palmari e affini. Tempo a disposizione: 60 minuti. Le domande sono etichettate con 1,2 o 3 asterischi:
 - * = domanda semplice, valutazione alta, rispondi a queste prima delle altre
 - ** = domanda di media difficoltà
 - *** = domanda difficile, valutazione bassa, rispondi dopo aver risposto alle altre

1. * Mostra lo schema che permette di tradurre indirizzi virtuali in reali per mezzo di inverted page tables. Descrivi i campi dell'inverted page table. Descrivi il problema delle collisioni e la sua soluzione.

schema	campi dell'inverted page table
--------	--------------------------------

Il problema delle collisioni

2. * Indica se le seguenti affermazioni sono vere o false con una crocetta nella rispettiva colonna.

Domanda	vero	falso
Una system call dà sempre luogo ad un mode switch.		
Un process switch avviene sempre contestualmente a 2 mode switch.		
Un interrupt viene gestito in modalità utente.		
Il process switch può avvenire sia in modalità kernel che in modalità utente.		
Il dispatcher viene sempre eseguito contestualmente ad un mode switch di tipo kernel → user.		
Se lo scheduling della CPU è preemptive, l'arrivo di dati genera sempre un process switch indipendentemente dalla priorità dei processi.		
Se un processo è in blocco da 10ms significa che 10ms fa ha eseguito una system call		
Ogni interrupt può essere associato ad un processo che ha richiesto una operazione di I/O.		
Molti page fault su un processo non modificano le prestazioni degli altri processi.		
Un processo per ottenere nuova memoria deve fare una system call.		
Un processo per lanciare un nuovo processo deve fare una system call.		
Una system call bloccante causa sempre un process switch.		

Cognome: _____ Nome: _____ Matricola: _____

Sistemi Operativi — A.A. 2005-2006, prova scritta del 21 settembre 2006

6. ** Considera i due sistemi raid4 e raid5 con 3+1 dischi. Supponi che ciascun disco impieghi un tempo w a scrivere un blocco. Assumi che il disk scheduler scriva sempre il blocco di parità contemporaneamente al blocco dati. Mostra i tempi che impiegano raid4 e raid5 nelle seguenti situazioni compilando la tabella.

<i>Descrizione della situazione</i>	<i>Tempo impiegato da raid4</i>	<i>Tempo impiegato da raid5</i>
Scrittura di un singolo blocco	w	
Scrittura di n blocchi distribuiti uniformemente. Supponi che tutte le richieste siano note allo scheduler del disco all'istante iniziale e può quindi ottimizzare.		
Commento		

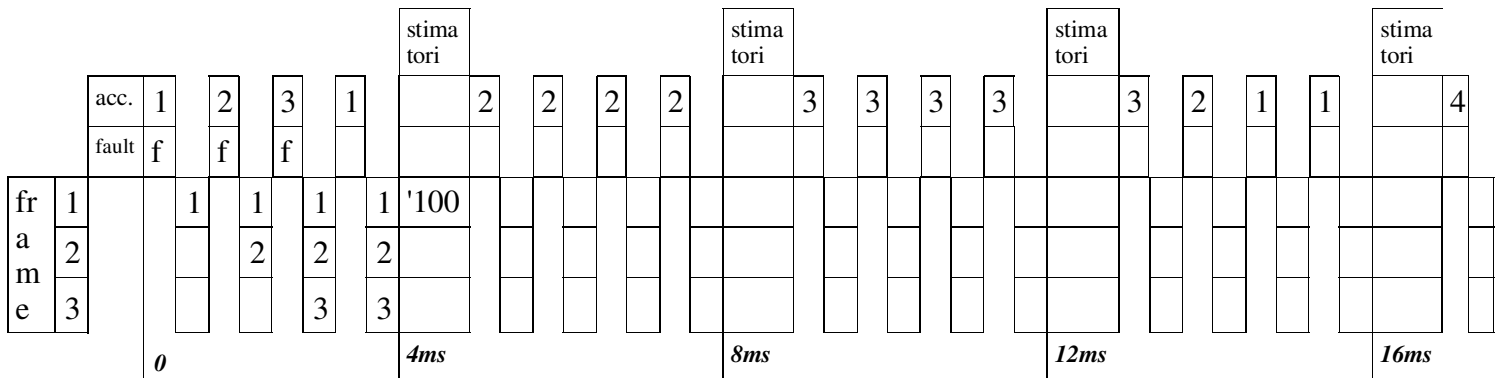
Considera ora raid5 con 3+1 dischi. Supponi che all'istante iniziale ci siano da schedulare 12 richieste per la scrittura dei blocchi 1,2,3,4,5,6,7,8,9,10,11,12. Mostra una possibile sequenza di scritture ottimizzata. Per chiarezza mostra anche l'organizzazione e la numerazione dei blocchi di raid5 con i blocchi di parità che stai considerando.

Numerazione blocchi raid 5

<i>disk1</i>	<i>disk2</i>	<i>disk3</i>	<i>disk4</i>
....			

Descrivi una sequenza di scritture ottimizzata dallo scheduler del disco.

7. ** Considera l'algoritmo di page replacement "aging" con 3 frame a disposizione, stimatore di anzianità a 3 bit con scorrimento a destra. Sweep ogni 4ms. Supponi che venga fatto un accesso a memoria ogni 1ms e la sequenza di accessi sia 1 2 3 1 2 2 2 3 3 3 3 2 1 1 4. Completa la seguente tabella.



Sistemi Operativi — A.A. 2005-2006, prova scritta del 21 settembre 2006

Verifica se LRU sostituirebbe le stesse pagine ed eventualmente spiega il perché delle differenze.

8. ** Considera un sistema con scheduling round robin. Nel sistema sono presenti n processi I/O bound con cpu burst trascurabile. Quale frazione di tempo mediamente ciascun processo aspetterebbe in coda ready? (ignora il tempo di esecuzione del process switch).

Se i processi fossero tutti cpu bound, quale frazione di tempo ciascun processo aspetta in coda ready?

Supponi ora di avere n processi tutti con I/O burst di durata b e cpu burst di durata c minore del quanto di tempo. Esprimi in formule la funzione $f(n,b,c)$ che dà la frazione di tempo di cpu utilizzato (cioè con almeno un processo running). Giustifica la risposta.

9. *** Dai una classe di stringhe di riferimenti a memoria su cui l'algoritmo di page replacement FIFO dà gli stessi page fault di CLOCK ma che non dia un fault ad ogni accesso. Considera 4 frame e 5 pagine.