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

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

Cognome:	Nome:	Matricola:		
Sistemi Operativi — A.A.	2005-2006, prova scritt	ta del 21 settembre 2006		
Tempo a disposizione: 60 mi * = domanda semplice, va ** = domanda di media dif	inuti. Le domande sono etiche llutazione alta, rispondi a que	este prima delle altre	lmari e af	fini.
1. * Mostra lo schema che peri campi dell'inverted page tabl		tuali in reali per mezzo di inverted page collisioni e la sua soluzione.	tables. De	scrivi i
schema		campi dell'inverted page table		
Il problema delle collisioni				
2. * Indica se le seguenti affern Domanda	nazioni sono vere o false con	una crocetta nella rispettiva colonna.	vero	falso
Una system call dà sempre luos	go ad un mode switch.			
Un process switch avviene sem	pre contestualmente a 2 mod	le switch.		
Un interrupt viene gestito in me	odalità utente.			
Il process switch può avvenire	sia in modalità kernel che in	modalità utente.		
Il dispatcher viene sempre eseg	guito contestualmente ad un n	node switch di tipo kernel→ user.		
Se lo scheduling della CPU è p indipendentemente dalla priori		nera sempre un process switch		
Se un processo è in blocco da 1	Oms significa che 10ms fa ha	a eseguito una system call		
Ogni interrupt può essere assoc	ciato ad un processo che ha ri	chiesto una operazione di I/O.		
Molti page fault su un processo	non modificano le prestazio	ni degli altri processi.		
Un processo per ottenere nuova	a memoria deve fare una syst	em call.		
Un processo per lanciare un nu	ovo processo deve fare una s	ystem call.		
Una system call bloccante caus	a sempre un process switch.			

3. ** 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 3 istruzioni che vengono eseguite consecutivamente. Calcola quanti page fault può generare al più ciascuna istruzione durante l'esecuzione del frammento in questione. 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	a parti di page	Page faults dovuti a codice o dati non residenti
0x003ffff6	carica nel registro A 4 byte a partire da 0x803ffffe		
0x003ffffB	carica nel registro B 4 byte a partire da 0x80400ffe		
0x00400000	aggiungi ad A il valore di B		

4. ** 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 1.5q e qullo di B dura 1.7q. Il cpu burst di A, il cpu burst 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 handler. Mostra schematicamente, nella seguente tabella, l'ordine con cui tali attività vengono eseguite (una sola croce per ciascuna colonna).

		npo _	-	-											
user mode	A														
	В														
	С														
mod	e switch														
	dispatching														
kernel mode	_														
	interrup handler														1

5 * Descrivi la tecnica di scheduling denominata feedback

	Descrivi la tecimea di semedaning	5 ** * *
1		
1		
1		
1		
1		
1		
1		
1		
1		
1		
1		
1		
1		
1		
1		
1		

	a di <i>n</i> blo ote allo s																		
Comme	ento																		
		. 15	2.1				1111					,			10				
scrittu	idera ora ura dei bl	locchi 1,	2,3,4,5	,6,7,8,	9,10,1	1,12.	Most	ra una	pos	sibile	sequ	enza	di scr	ittur	e otti	mizz	ata. P	er	
	ezza mos derando.	tra anch	e l'org	anizza	zione e	e la nu	mera	zione	dei l	oloccl	ni di 1	raid5	con i	bloc	chi d	i pari	tà ch	e stai	
	erazione	blocchi i	raid 5																
	T		1		Desc	rivi uı	na sec	quenza	a di s	scrittu	ire ot	timiz	zata d	allo	sched	luler	del d	isco.	
disk1	disk2	disk3	disk4																
•••																			
																			J
. ** Co	onsidera	l'algoritr	no di p	age re	placem	nent "a	iging	con í	3 fra	me a	dispo	sizio	ne, sti	mate	ore di	anzi	anità	a 3 bit	
di acc	corrimen cessi sia	to a dest	ra. Swo 2223	eep og 3 3 3 3	ni 4ms 3 2 1	s. Supp 1 4. C	ompl	eta la	nga i segu	ente 1	un ac tabell	cesso a.	a me	mori	a ogi	11 1 m	is e 1a	seque	ızε
			S	tima				stima]				stima]				stima	
ac	ec. 1	2 3	1	ori 2	2	2	2	tori	3	3	3	3	tori	3	2	1	1	tori	4
		f f														1			-
r 1	1	1 1	1 '	100															Ţ
a 2		2 2	1 ++																
e 3		3		lms				8ms					12ms					16ms	L
	0		4	riits				oms					121118					Toms	

__Nome: _____

scrivere un blocco. Assumi che il disk scheduler scriva sempre il blocco di parità contemporaneamente al blocco

Tempo impiegato

da raid4

w

Tempo

impiegato da raid5

6. ** Considera i due sistemi raid4 e raid5 con 3+1 dischi. Supponi che ciascun disco impieghi un tempo w a

dati. Mostra i tempi che impiegano raid4 e raid5 nelle seguenti situazioni compilando la tabella.

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

Descrizione della situazione

Cognome:____

Scrittura di un singolo blocco

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 <i>n</i> processi I/O bound con cpu burs 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.
Tummig). Grastifica la risposta.
9. *** Dai una classe di stringhe di riferimenti a memoria su cui l'algoritmo di page replacement FIFO da gli stessi page fault di CLOCK ma che non dia un fault ad ogni accesso. Considera 4 frame e 5 pagine.