

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

Nome: _____ Cognome: _____ Matricola: _____ fila: ___ posto: ___

Esercizio 1 (5 punti)

Si consideri un processore che dispone dei seguenti registri:

- i registri speciali PC (program counter) e PS (program status)
- un banco di registri utilizzato in stato utente che comprende i registri generali R1, R2 e lo stack pointer SP
- un banco di registri utilizzato in stato supervisore che comprende i registri generali R1', R2' e lo stack pointer SP'

L'hardware provvede al salvataggio dei registri PC e PS nello stack del nucleo quando riconosce un'interruzione e al loro ripristino dallo stack del nucleo quando esegue un'istruzione IRET.

Tutti i processi hanno uguale priorità e il sistema gestisce il processore con politica Round Robin. Al tempo t , quando sono presenti nel sistema (tra gli altri) il processo A, in stato di esecuzione, e il processo B che è bloccato in attesa della lettura di un blocco di dati dal disco, arriva un'interruzione dal disco che segnala il completamento dell'operazione di lettura.

L'interruzione è riconosciuta dal processore che individua il vettore di interruzione corrispondente, contenente i valori PC=0F01 e PS=A3AF.

Immediatamente dopo il tempo t , quando l'interruzione generata dal disco viene riconosciuta, i registri del processore, i descrittori di A e B e lo stack del nucleo hanno i contenuti mostrati in figura.

L'interruzione determina l'intervento del nucleo, che gestisce l'interruzione. Supponendo che l'interruzione da timer abbia minor priorità di quella da disco, si chiede:

- a) il contenuto dei descrittori, dei registri generali e speciali e dello stack del nucleo durante la fase di estrazione dell'istruzione eseguita subito dopo il riconoscimento dell'interruzione;
- b) il contenuto dei descrittori, dei registri generali e speciali e dello stack del nucleo durante la fase di estrazione dell'istruzione IRET con la quale termina la gestione dell'interruzione;
- c) il contenuto dei descrittori, dei registri generali e speciali e dello stack del nucleo durante la fase di estrazione dell'istruzione eseguita subito dopo la IRET.

DESCRITTORE DI A		DESCRITTORE DI B		STACK DEL NUCLEO		REG. GENERALI	
Stato	Esec	Stato	Bloccato	SP	3EAA
PC	23FA	PC	409A	3F0A		R1	07AA
PS	16F2	PS	16F2	3F09		R2	08AA
SP	3EFA	SP	500B	3F08			
R1	AAA1	R1	BBB6	3F07			
R2	AAA2	R2	BBB5	3F06		REG. NUCLEO	
				3F05		SP'	3F0A
				3F04		R1'	0000
PROCESSORE: Registri speciali						R2'	0001
PC	240A	PS	16F2				

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

Soluzione

- a) contenuto dei descrittori, dei registri generali e speciali e dello stack del nucleo durante la fase di estrazione dell'istruzione eseguita subito dopo il riconoscimento dell'interruzione:

DESCRITTORE DI A		DESCRITTORE DI B		STACK DEL NUCLEO		REG. GENERALI	
Stato		Stato		SP	
PC		PC		3F0A		R1	
PS		PS		3F09		R2	
SP		SP		3F08			
R1		R1		3F07			
R2		R2		3F06		REG. NUCLEO	
				3F05		SP'	
				3F04		R1'	
PROCESSORE: Registri speciali							R2'
PC		PS					

- b) contenuto dei descrittori, dei registri generali e speciali e dello stack del nucleo durante la fase di estrazione dell'istruzione IRET con la quale termina la gestione dell'interruzione:

DESCRITTORE DI A		DESCRITTORE DI B		STACK DEL NUCLEO		REG. GENERALI	
Stato		Stato		SP	
PC		PC		3F0A		R1	
PS		PS		3F09		R2	
SP		SP		3F08			
R1		R1		3F07			
R2		R2		3F06		REG. NUCLEO	
				3F05		SP'	
				3F04		R1'	
PROCESSORE: Registri speciali							R2'
PC		PS					

- c) contenuto dei descrittori, dei registri generali e speciali e dello stack del nucleo durante la fase di estrazione dell'istruzione eseguita subito dopo la IRET:

DESCRITTORE DI A		DESCRITTORE DI B		STACK DEL NUCLEO		REG. GENERALI	
Stato		Stato		SP	
PC		PC		3F0A		R1	
PS		PS		3F09		R2	
SP		SP		3F08			
R1		R1		3F07			
R2		R2		3F06		REG. NUCLEO	
				3F05		SP'	
				3F04		R1'	
PROCESSORE: Registri speciali							R2'
PC		PS					

Soluzione

- a) contenuto dei descrittori, dei registri generali e speciali e dello stack del nucleo durante la fase di estrazione dell'istruzione eseguita subito dopo il riconoscimento dell'interruzione:

DESCRITTORE DI A		DESCRITTORE DI B		STACK DEL NUCLEO		REG. GENERALI	
Stato	Esec	Stato	Bloccato	SP	invariato
PC	invariato	PC	invariato	3F0A	240A	R1	invariato
PS	invariato	PS	invariato	3F09	16F2	R2	invariato
SP	invariato	SP	invariato	3F08			
R1	invariato	R1	invariato	3F07			
R2	invariato	R2	invariato	3F06		REG. NUCLEO	
				3F05		SP'	3F08
				3F04		R1'	invariato
PROCESSORE: Registri speciali							R2'
PC	0F01	PS	A3AF				invariato

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

- b) contenuto dei descrittori, dei registri generali e speciali e dello stack del nucleo durante la fase di estrazione dell'istruzione IRET con la quale termina la gestione dell'interruzione:

DESCRITTORE DI A		DESCRITTORE DI B		STACK DEL NUCLEO		REG. GENERALI	
Stato	Esec	Stato	Pronto	SP	invariato
PC	invariato	PC	invariato	3F0A	invariato	R1	invariato
PS	invariato	PS	invariato	3F09	invariato	R2	invariato
SP	invariato	SP	invariato	3F08			
R1	invariato	R1	invariato	3F07			
R2	invariato	R2	invariato	3F06		REG. NUCLEO	
				3F05		SP'	invariato
				3F04		R1'	??
PROCESSORE: Registri speciali						R2'	??
PC	0F01+ ??	PS	invariato				

- c) contenuto dei descrittori, dei registri generali e speciali e dello stack del nucleo durante la fase di estrazione dell'istruzione eseguita subito dopo la IRET:

DESCRITTORE DI A		DESCRITTORE DI B		STACK DEL NUCLEO		REG. GENERALI	
Stato	Esec	Stato	Pronto	SP	invariato
PC	invariato	PC	invariato	3F0A		R1	invariato
PS	invariato	PS	invariato	3F09		R2	invariato
SP	invariato	SP	invariato	3F08			
R1	invariato	R1	invariato	3F07			
R2	invariato	R2	invariato	3F06		REG. NUCLEO	
				3F05		SP'	3F0A
				3F04		R1'	??
PROCESSORE: Registri speciali						R2'	??
PC	240A	PS	16F2				

Esercizio 2 (5 punti)

Si consideri un sistema che gestisce la memoria utilizzando un algoritmo di sostituzione LRU locale e che attribuisce a ogni processo un numero MaxBlocchi di blocchi a disposizione del working set. Questo numero è definito dinamicamente e viene incrementato o decrementato dal processo di sistema WorkingSetManager, che interviene periodicamente e applica la seguente politica:

- appartengono al working set di un processo le pagine il cui *tempo virtuale di ultimo riferimento* è maggiore o uguale di $t - 10$ dove t è il valore attuale del *tempo virtuale* del processo
- se il numero PagineInEccesso delle pagine non appartenenti al working set che sono caricate in memoria è maggiore di 1, si scaricano PagineInEccesso - 1 pagine applicando l'algoritmo LRU e si decreta di 1 il valore di MaxBlocchi per ogni pagina scaricata;
- se tutte le pagine caricate in memoria appartengono al working set, si incrementa di 1 il valore di MaxBlocchi;

Al verificarsi di un errore di pagina si ha il seguente comportamento:

- se il numero di pagine attualmente caricate è minore di MaxBlocchi, la pagina riferita viene caricata assegnando il primo dei blocchi dalla lista BlocchiDisponibili.
- se il numero di pagine attualmente caricate è uguale a MaxBlocchi, viene scaricata la pagina prescelta dall'algoritmo di sostituzione e la pagina riferita viene caricata al suo posto (usando i blocchi già assegnati a quel processo).

A un certo istante, quando il tempo virtuale del processo A è $t=100$, il valore di MaxBlocchi assegnato al processo A è MaxBlocchi=6 la tabella delle pagine del processo A è la seguente:

Pagina	Blocco	Tempo virtuale di ultimo riferimento
0	12	93
1	13	88
2	-	
3	14	81
4	15	78
5	-	
6	16	63
7	-	
8	17	99

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

9	-	
Processo A		

A partire da questo istante interviene il processo WorkingSetManager, che applica la politica sopra definita. Durante l'esecuzione di WorkingSetManager il tempo virtuale dei processi rimane invariato.

Successivamente il processo A avanza ed esegue i seguenti riferimenti alla memoria:

- al tempo 101 riferisce la pagina 3
- al tempo 102 riferisce la pagina 0
- al tempo 103 riferisce la pagina 1

Si chiede:

- a) Le pagine eventualmente scaricate da WorkingSetManager; il valore di MaxBlocchi assegnato al processo A quando termina l'esecuzione di WorkingSetManager; la configurazione della tabella delle pagine del processo A quando termina l'esecuzione di WorkingSetManager.
- b) i caricamenti e gli eventuali scaricamenti di pagine eseguiti nei tempi 101,102 e 103; il valore di MaxBlocchi assegnato al processo A al tempo 103; la configurazione della tabella delle pagine del processo A al tempo 103

Soluzione

a-1) Pagine scaricate dal WorkingSetManager : _____

a-2) Valore di MaxBlocchi assegnato al processo A quando termina l'esecuzione di WorkingSetManager: _____

b-1) Caricamenti e gli eventuali scaricamenti di pagine eseguiti nei tempi 101,102 e 103

- tempo 101: rimosse le pagine _____; caricate le pagine _____ nei blocchi _____, rispettivamente
- tempo 102: rimosse le pagine _____; caricate le pagine _____ nei blocchi _____, rispettivamente
- tempo 103: rimosse le pagine _____; caricate le pagine _____ nei blocchi _____, rispettivamente

b-2) Valore di MaxBlocchi assegnato al processo A al tempo 103: _____

a-3) Configurazione della tabella delle pagine del processo A quando termina l'esecuzione di WorkingSetManager:

Pagina	Blocco	<i>Tempo virtuale di ultimo riferimento</i>
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
Processo A		

b-3) Configurazione della tabella delle pagine del processo A al tempo 103 :

Pagina	Blocco	<i>Tempo virtuale di ultimo riferimento</i>
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
Processo A		

Soluzione

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

a-1) Pagine scaricate dal WorkingSetManager : 6, 4, 3

a-2) Valore di MaxBlocchi assegnato al processo A quando termina l'esecuzione di WorkingSetManager: 3

b-1) Caricamenti e gli eventuali scaricamenti di pagine eseguiti nei tempi 101,102 e 103

- tempo 101: rimossa la pagina 1 e caricata la pagina 3 nel blocco 13
- tempo 102: aggiornato l'istante di ultimo riferimento della pagina 0
- tempo 103: rimossa la pagina 8 e caricata la pagina 1 nel blocco 17

b-2) Valore di MaxBlocchi assegnato al processo A al tempo 103: MaxBlocchi= 3

a-3) Configurazione della tabella delle pagine del processo A quando termina l'esecuzione di WorkingSetManager:

Pagina	Blocco	Tempo virtuale di ultimo riferimento
0	12	93
1	13	88
2	-	
3	-	
4	-	
5	-	
6	-	
7	-	
8	17	99
9	-	
Processo A		

b-3) Configurazione della tabella delle pagine del processo A al tempo 103 :

Pagina	Blocco	Tempo virtuale di ultimo riferimento
0	12	102
1	17	103
2	-	
3	13	101
4	-	
5	-	
6	-	
7	-	
8	-	
9	-	
Processo A		

Esercizio 3 (5 punti)

Un sistema operativo che gestisce la memoria con paginazione a domanda, utilizza l'algoritmo di sostituzione *Second Chance* in una versione che tiene conto, oltre che del bit di riferimento, anche del bit di modifica delle pagine caricate in memoria. Gli elementi della *CoreMap* hanno i seguenti campi:

- *Proc*: processo a cui è assegnato il blocco; il campo è vuoto se il blocco è libero;
- *Pag* indice della pagina caricata nel blocco;
- *Rif*: bit di pagina riferita: valore 1 assegnato dall'hardware quando la pagina viene riferita in lettura o scrittura, valore 0 assegnato da *Second Chance*;
- *Mod*: bit di pagina modificata: valore 1 assegnato dall'hardware quando la pagina viene riferita in scrittura, valore 0 assegnato dal sistema operativo quando completa il salvataggio della pagina in memoria secondaria, a seguito di un comando generato da *Second Chance*.

Al tempo t la *Core Map* ha la configurazione mostrata in figura, con il puntatore dell'algoritmo di *SecondChance* posizionato sul blocco 11. Sono ignorati i blocchi di memoria occupati dal sistema operativo.

												↓													
Proc	D	D	A	A	A	B	C	A	B	A	B	B	C	D	D	C	D	B	A	B	C	C	D	C	
Pag	0	8	5	10	12	7	0	1	0	2	9	6	3	1	2	10	6	2	7	3	12	7	11	2	
Rif	1	0	0	0	1	1	0	1	0	0	1	1	1	1	1	1	1	1	0	1	1	1	0		

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

Mod	1	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	1	1	0	0	0	1	0	0
Blocco	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Core Map al tempo t

L'algoritmo *Second Chance*, che viene eseguito se si verifica un page fault e non ci sono pagine libere, percorre gli elementi della *CoreMap* a partire da quello sul quale è inizialmente posizionato il puntatore; esamina i bit *Rif* e *Mod* dell'elemento corrente, e:

- se $Rif = 1$ assegna $Rif = 0$; se $Mod = 1$ comanda il salvataggio in memoria secondaria della pagina caricata nel blocco corrente; quindi posiziona il puntatore sull'elemento successivo della *CoreMap* e continua la scansione;
- se $Rif = 0$ e $Mod = 0$ sceglie la pagina corrente come *vittima*; quindi posiziona il puntatore sull'elemento successivo della *CoreMap* e termina.

E' da notare che le pagine per le quali è stato richiesto il salvataggio in memoria secondaria restano marcate come modificate finché l'operazione di salvataggio non è stata completata.

A partire dal tempo t si verifica la seguente sequenza di eventi:

Tempo	Evento	Page Fault	Tempo	Evento	Page Fault
$t+1$	Proc C esegue lettura di pag 2		$t+5$	Proc A esegue lettura di pag 2	
$t+2$	Poc C esegue scrittura di pag 0		$t+6$	Termina salvataggio di D-1, D-2, D-8	
$t+3$	Proc C esegue scrittura di pag 3		$t+7$	Proc A esegue scrittura di pag 10	
$t+4$	Proc C esegue lettura di pag 5	SI	$t+8$	Proc A esegue scrittura di pag 4	SI

Si chiede la configurazione della *CoreMap* subito dopo la gestione degli eventi che si verificano ai tempi $t+3$, $t+4$; $t+7$ e $t+8$.

Soluzione

Proc																								
Pag																								
Rif																								
Mod																								
Blocco	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Core Map dopo la gestione dell'evento che si verifica al tempo $t+3$

Proc																								
Pag																								
Rif																								
Mod																								
Blocco	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Core Map dopo la gestione dell'evento che si verifica al tempo $t+4$

Proc																								
Pag																								
Rif																								
Mod																								
Blocco	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Core Map dopo la gestione dell'evento che si verifica al tempo $t+7$

Proc																								
Pag																								
Rif																								
Mod																								
Blocco	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Core Map dopo la gestione dell'evento che si verifica al tempo $t+8$

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

Soluzione

												↓												
Proc	D	D	A	A	A	B	C	A	B	A	B	B	C	D	D	C	D	B	A	B	C	C	D	C
Pag	0	8	5	10	12	7	0	1	0	2	9	6	3	1	2	10	6	2	7	3	12	7	11	2
Rif	1	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1
Mod	1	1	0	0	1	0	1	1	0	0	0	1	1	1	1	0	1	1	0	0	0	1	0	0
Blocco	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Core Map dopo la gestione dell'evento che si verifica al tempo t+3

																								↓
Proc	D	D	A	A	A	B	C	A	B	A	B	B	C	D	D	C	D	B	A	C	C	C	D	C
Pag	0	8	5	10	12	7	0	1	0	2	9	6	3	1	2	10	6	2	7	5	12	7	11	2
Rif	1	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1
Mod	1	1	0	0	1	0	1	1	0	0	0	1	1	1	1	0	1	1	0	0	0	1	0	0
Blocco	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Core Map dopo la gestione dell'evento che si verifica al tempo t+4

																								↓
Proc	D	D	A	A	A	B	C	A	B	A	B	B	C	D	D	C	D	B	A	C	C	C	D	C
Pag	0	8	5	10	12	7	0	1	0	2	9	6	3	1	2	10	6	2	7	5	12	7	11	2
Rif	1	0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1
Mod	1	0	0	1	1	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	0	0
Blocco	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Core Map dopo la gestione dell'evento che si verifica al tempo t+7

			↓																					
Proc	D	A	A	A	A	B	C	A	B	A	B	B	C	D	D	C	D	B	A	C	C	C	D	C
Pag	0	4	5	10	12	7	0	1	0	2	9	6	3	1	2	10	6	2	7	5	12	7	11	2
Rif	0	1	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
Mod	1	1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	0	0
Blocco	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Core Map dopo la gestione dell'evento che si verifica al tempo t+8

Esercizio 4 (3 punti)

In un sistema vengono generati i processi A,B,C,D,E con i tempi di arrivo e le durate (tempo di utilizzo del processore necessario per arrivare alla terminazione) sotto specificate:

Processo	Durata	Tempo di generazione
A	25	0
B	40	1
C	5	2
D	15	3
E	10	4

Tutti i tempi sono espressi in millisecondi

La politica di gestione del processore è la Round-Robin, con quanto di tempo pari a 10 msec e con tempo di commutazione di contesto trascurabile. Se un processo si sospende, alla riattivazione ottiene un intero quanto di tempo, indipendentemente dalla frazione consumata prima della sospensione.

E' inoltre definito il semaforo *sem*, con valore iniziale 0 e coda inizialmente vuota, sul quale vengono eseguite le seguenti operazioni:

- al tempo 12 il processo in esecuzione esegue $P(sem)$ (per gli studenti degli A.A. antecedenti al 2013/14: *wait(sem)*)
- al tempo 28 il processo in esecuzione esegue $P(sem)$ (per gli studenti degli A.A. antecedenti al 2013/14: *wait(sem)*)
- al tempo 36 il processo in esecuzione esegue $V(sem)$ (per gli studenti degli A.A. antecedenti al 2013/14: *signal(sem)*)
- al tempo 40 il processo in esecuzione esegue $V(sem)$ (per gli studenti degli A.A. antecedenti al 2013/14: *signal(sem)*)
- al tempo 50 il processo in esecuzione esegue $V(sem)$ (per gli studenti degli A.A. antecedenti al 2013/14: *signal(sem)*)

Si chiede di compilare la tabella riportata nello schema di soluzione fino al tempo 50, riportando in ogni riga lo stato raggiunto dal sistema (processo in esecuzione, contenuto della Coda Pronti, valore di *sem*, contenuto della coda di *sem*) subito dopo aver gestito l'evento che si verifica al tempo indicato nella prima colonna.

Soluzione

(indicare tra parentesi tonde il tempo residuo di esecuzione del processo)

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

Esercizio 5 (3 punti)

Un sistema con 5 processi A, B, C, D, E e risorse dei tipi R1, R2, R3 rispettivamente di molteplicità [5, 4, 7] adotta nei confronti dello stallo la politica di riconoscimento ed eliminazione. Al tempo t il sistema raggiunge lo stato mostrato nella seguente tabella.

Assegnazione attuale			
	R1	R2	R3
A	2	1	
B			2
C		1	3
D	1		1
E	2	1	

Esigenza Massima (assegnazione attuale + esigenza residua)			
	R1	R2	R3
A	2	3	6
B	1	2	3
C		4	4
D	3	2	4
E	2	1	1

Esigenza residua (tiene conto dell' assegnazione attuale)			
	R1	R2	R3
A	0	2	6
B	1	2	1
C	0	3	1
D	2	2	3
E	0	0	1

Molteplicità		
R1	R2	R3
5	4	7

Disponibilità		
0	1	1

Dire se il sistema è in stallo. Inoltre:

- In caso affermativo dire quali processi sono in stallo e quale processo è possibile far terminare per rimuovere lo stallo. Motivare la risposta.
- In caso negativo dire se l'assegnazione di una risorsa di tipo R3 al processo E porta il sistema in stallo

Soluzione

Verifica dello stallo:

- Il processo _____ può terminare
La disponibilità di {R1, R2, R3} diviene { _____ }
- Il processo _____ può terminare
La disponibilità di {R1, R2, R3} diviene { _____ }
- Il processo _____ può terminare
La disponibilità di {R1, R2, R3} diviene { _____ }
- Il processo _____ può terminare
La disponibilità di {R1, R2, R3} diviene { _____ }
- Il processo _____ può terminare
La disponibilità di {R1, R2, R3} diviene { _____ }

Il sistema è in stallo [SI/NO]? _____

Se il sistema è in stallo:

quali processi sono in stallo? _____

indicare quali processi si possono sopprimere per rimuovere lo stallo, spiegare perchè:

Se il sistema non è in stallo:

supponendo di assegnare una risorsa di tipo R3 al processo E, si analizza lo stato risultante.

- Il processo _____ può terminare
La disponibilità di {R1, R2, R3} diviene { _____ }
- Il processo _____ può terminare
La disponibilità di {R1, R2, R3} diviene { _____ }
- Il processo _____ può terminare
La disponibilità di {R1, R2, R3} diviene { _____ }
- Il processo _____ può terminare
La disponibilità di {R1, R2, R3} diviene { _____ }
- Il processo _____ può terminare
La disponibilità di {R1, R2, R3} diviene { _____ }

Il sistema è in stallo [SI/NO]? _____

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

Soluzione

Verifica dello stallo:

- 1) Il processo E può terminare
La disponibilità di {R1, R2, R3} diviene { 2,2,1 }
- 2) Il processo B può terminare
La disponibilità di {R1, R2, R3} diviene { 2,2,3 }
- 3) Il processo D può terminare
La disponibilità di {R1, R2, R3} diviene { 3,2,4 }

I processi A e C non possono terminare, quindi il sistema è in stallo.

I processi in stallo sono A e C.

Per rimuovere lo stallo è possibile sopprimere uno qualsiasi dei processi A e C. Infatti se si sopprime A i processi E, B e D possono terminare (come si è verificato in precedenza) e a questo punto C ha a disposizione tutte le risorse del sistema per terminare. In modo simile, se si sopprime C, oltre ai processi E, B e D anche A può terminare.

Esercizio 6 (3 punti)

Un disco con 2 facce, 100 settori per traccia e 200 cilindri ha un tempo di seek proporzionale al numero di cilindri attraversati e pari a 0,1 ms per ogni cilindro. Il periodo di rotazione è di 10 msec: conseguentemente, il tempo impiegato per percorrere un settore è di 0,1 msec.

A un certo tempo (convenzionalmente indicato come $t=0$) termina l'esecuzione dei comandi sul cilindro 40 e sono pervenute, nell'ordine, le seguenti richieste di lettura o scrittura:

- cilindro 48, faccia 0, settore 8
- cilindro 36, faccia 1, settore 9
- cilindro 180, faccia 1, settore 5

Successivamente arrivano tre ulteriori comandi di lettura:

- cilindro 77, faccia 0, settore 60 al tempo 21,6
- cilindro 77, faccia 1, settore 61 al tempo 21,7
- cilindro 35, faccia 1, settore 62 al tempo 35

Calcolare il tempo necessario per eseguire tutte queste operazioni supponendo che si adotti la politica di scheduling Shortest Time Seek First.

Per il ritardo rotazionale dopo un'operazione di seek si assume sempre il valore di caso peggiore, pari a un intero periodo di rotazione.

Si assuma inoltre che il controllore sia dotato di sufficiente capacità di buffering e che sia sempre in grado di accettare senza ritardo i dati letti dal disco o quelli da scrivere sul disco. Inoltre, si assuma che i comandi sullo stesso cilindro vengano eseguiti in ordine FIFO.

Soluzione

op. su cilindro:	settore:			
inizio:	seek:	rotazione:	percorrenza:	fine:
op. su cilindro:	settore:			
inizio:	seek:	rotazione:	percorrenza:	fine:
op. su cilindro:	settore:			
inizio:	seek:	rotazione:	percorrenza:	fine:
op. su cilindro:	settore:			
inizio:	seek:	rotazione:	percorrenza:	fine:
op. su cilindro:	settore:			
inizio:	seek:	rotazione:	percorrenza:	fine:

Soluzione

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

op. su cilindro:	36	settore:	9				
inizio:	0	seek:	0,4	rotazione:	10	percorrenza:	0,1 fine: 10,5
op. su cilindro:	48	settore:	8				
inizio:	10,5	seek:	1,2	rotazione:	10	percorrenza:	0,1 fine: 21,8
op. su cilindro:	77	settore:	60				
inizio:	21,8	seek:	2,9	rotazione:	10	percorrenza:	0,1 fine: 34,8
op. su cilindro:	77	settore:	61				
inizio:	34,8	seek:	0	rotazione:	0	percorrenza:	0,1 fine: 34,9
op. su cilindro:	180	settore:	5				
inizio:	34,9	seek:	10,3	rotazione:	10	percorrenza:	0,1 fine: 55,3
op. su cilindro:	35	settore:	62				
inizio:	55,3	seek:	14,5	rotazione:	10	percorrenza:	0,1 fine: 79,9

Esercizio 7 (2 punti)

Un disco di 200 GByte ospita un file system basato su i-node che utilizza blocchi di 4 Kbyte e puntatori di 32 bit, e riserva 10.000 blocchi per la memorizzazione dell'i-list. Ogni i-node occupa 128 bytes e include 10 puntatori diretti e tre puntatori indiretti: uno indiretto singolo, uno indiretto doppio e uno indiretto triplo

Si chiede:

1. Il massimo numero di blocchi indirizzabili tramite un puntatore indiretto singolo
2. Il massimo numero di blocchi indirizzabili tramite un puntatore indiretto doppio
3. Il massimo numero di blocchi indirizzabili tramite un puntatore indiretto triplo
4. Il massimo numero di blocchi indirizzabili da un i-node
5. la massima dimensione di un file
6. il massimo numero di files memorizzabili nel disco

Soluzione

1. Massimo numero di blocchi indirizzabili tramite un puntatore indiretto singolo: _____
2. Massimo numero di blocchi indirizzabili tramite un puntatore indiretto doppio: _____
3. Massimo numero di blocchi indirizzabili tramite un puntatore indiretto triplo: _____
4. Massimo numero di blocchi indirizzabili da un i-node: _____
5. Massima dimensione di un file: _____
6. Massimo numero di file memorizzabili nel disco: _____

Soluzione

1. Massimo numero di blocchi indirizzabili tramite un puntatore indiretto singolo: $4\text{Kbyte} / 4\text{ bytes} = 1024$
2. Massimo numero di blocchi indirizzabili tramite un puntatore indiretto doppio: 1024^2
3. Massimo numero di blocchi indirizzabili tramite un puntatore indiretto triplo: 1024^3
4. Massimo numero di blocchi indirizzabili da un i-node: $10 + 1024 + 1024^2 + 1024^3 = 10 + 2^{10} + 2^{20} + 2^{30}$
5. Massima dimensione di un file: 200 GB, pari alla dimensione del disco.
infatti la capacità di indirizzamento di un i-node è superiore a 2^{30} blocchi, pari $4 * 2^{40}$ byte, ed eccede quindi la capacità del disco.
6. Massimo numero di file memorizzabili nel disco: $10.000 * 4\text{KBytes}/128\text{ bytes} = 320.000$

Esercizio 8 (2 punti)

Considerate le due primitive wait e signal che operano sulle variabili di condizione. Per ognuna delle due dire se può causare la sospensione o la terminazione del thread invocante e se può comportare un prerilascio del processore.

Cond.wait() (per gli student degli A.A. antecedenti il 2013/14: pthread_cond_wait(cond))

Può far terminare il processo in esecuzione [SI/NO]? _____

motivazione: _____

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

Può provocare la sospensione del processo in esecuzione [SI/NO]? _____

motivazione: _____

Può comportare l'invocazione dello schedatore [SI/NO]? _____

motivazione: _____

Cond.signal() (per gli student degli A.A. antecedenti il 2013/14: pthread_cond_signal(cond))

Può far terminare il processo in esecuzione [SI/NO]? _____

motivazione: _____

Può provocare la sospensione del processo in esecuzione [SI/NO]? _____

motivazione: _____

Può comportare l'invocazione dello schedatore [SI/NO]? _____

motivazione: _____

Soluzione

Cond.wait() (per gli student degli A.A. antecedenti il 2013/14: pthread_cond_wait(cond))

Può far terminare il processo in esecuzione [SI/NO]? NO

motivazione: comporta solo l'attesa del thread invocante

Può provocare la sospensione del processo in esecuzione [SI/NO]? SI

motivazione: comporta solo l'attesa del thread invocante, che viene quindi sospeso fino alla riattivazione tramite signal

Può comportare l'invocazione dello schedatore [SI/NO]? SI

motivazione: necessario, dato che il thread in esecuzione si sospende

Cond.signal() (per gli student degli A.A. antecedenti il 2013/14: pthread_cond_signal(cond))

Può far terminare il processo in esecuzione [SI/NO]? NO

motivazione: comporta la riattivazione di un eventuale thread sospeso sulla variabile di condizione.

Può provocare la sospensione del processo in esecuzione [SI/NO]? NO

motivazione: comporta la riattivazione di un eventuale thread sospeso sulla variabile di condizione, e non ha effetto sul thread in esecuzione

Può comportare l'invocazione dello schedatore [SI/NO]? SI

motivazione: nel caso di semantica Hoare. Invece, nel caso di semantica MESA, il thread in esecuzione mantiene il processore.

Esercizio 9 (2 punti)

In un sistema UNIX, un processo genera un figlio eseguendo il seguente frammento di codice:

```
printf("UNO");
```

Sistemi Operativi e Laboratorio, Prova del 16/2/2015

```
pid=fork();
if(pid==0)
{
    execl("xxx",NULL);
    printf("DUE");
    exit(1);
}
else if (pid>0)
    {
        pid=wait(&status);
        printf("%n",status);
        exit(2);
    }
printf("TRE");
```

Che cosa stampano per effetto di questo codice il processo padre e il processo figlio se la *fork* ha successo e la *exec* fallisce?

Soluzione

Il padre stampa: _____

Il figlio stampa: _____

Soluzione

Il padre stampa: "UNO1"

Il figlio stampa: "DUE"