Nome/cognome ____________________________ N. di matricola (10 cifre) _____________________ Posizione: Riga ____ Col _____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2005/2006
CONCORRENZA - 19 Giugno 2006

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

Esercizio 1:

Sia dato un insieme di produttori, un insieme di consumatori e un processo censore.

Ogni produttore dopo aver prodotto un elemento chiama la funzione bbk.enqueue(id, el)

dove id e' il proprio identificatore, el l'elemento prodotto.

Ogni consumatore per ricevere un elemento da consumare chiama el=bbk.dequeue(id)

dove id e' il proprio identificatore.

Il processo censore puo' "isolare" produttori o consumatori chiamando bbk.censure(id)

Il buffer dovra' ignorare ogni ulteriore richiesta dal processo indicato. Gli elementi nel buffer prodotti dal processo indicato dovranno essere cancellati (al ricevimento della richiesta, non successivamente).

Esercizio 2:

In un sistema distribuito ci sono n macchine server con 1Gb di memoria, m macchine client che desiderano allocare memoria sui server e 1 arbitro centralizzato. I server implementano il meccanismo di allocazione della memoria rispondendo a messaggi inviati dall'arbitro. L'arbitro implementa la politica di allocazione della memoria. I client chiedono all'arbitro di allocare/deallocare memoria e comunicano direttamente con i server per le operazioni di lettura/scrittura. I messaggi scambiati dai tre attori sono:

Da:	A:	Messaggio:	Semantica:
client	arbitro	<alloc,M>	richiede di allocare M byte di memoria
arbitro	server	<alloc,x,y>	alloca l'area contigua di memoria [x,y]
arbitro	client	<done,sid,x>	memoria allocata sul server sid a partire dall'indirizzo x
client	arbitro	<dealloc,sid,x>	richiede di deallocare la memoria allocata sul server sid a partire da x
arbitro	server	<dealloc,x>	dealloca l'area di memoria [x,y] allocata in precedenza
client	server	<write,x,y,b,n>	copia sul server a partire dall'indirizzo x+y (x base, y offset) il contenuto dei primi n byte del buffer b
client	server	<read,x,y,n>	legge n byte dal server a partire dall'indirizzo x+y (x base, y offset)
server	client	<done,b>	invia al client il buffer con i byte richiesti dall'ultimo messaggio read

Implementare il protocollo sul lato client, server e arbitro utilizzando primitive di message passing asincrono.

L'arbitro deve implementare la politica di allocazione di memoria denominata best fit.

Esercizio 3:

Siano inizialmente S1 inizializzato a 0, S2 inizializzato a 0 e S3 inizializzato a 3.

Si considerino i seguenti tre processi.

P1: while(true) {S1.P();S2.V();S3.P();}

P2: while(true) {S1.V();S1.P();S3.P();}

P3: while(true) {S1.V();S2.P();S3.V();}

Sono possibili casi di deadlock? Se no, spiegare perche', altrimenti descrivere una possibile alternanza di processi che porti al deadlock. Se si e' risposto si' alla precedente domanda, e' certo che il sistema entri in deadlock? Se si, spiegare perche'; altrimenti mostrare un'esecuzione infinita e discutere informalmente la probabilita' che il deadlock si verifichi.

Nome/cognome __________________________ N. di matricola (10 cifre) ____________________ Posizione: Riga ____ Col _____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2005/2006
PARTE GENERALE - 19 Giugno 2006

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

Esercizio 1:

Si scriva l'algoritmo dell'ascensore come monitor.

I processi quando hanno necessita' di un blocco di disco chiamano

elevator.readwrite(op,bk,cyl,head,sector) (op==RD per read, op==WR per write)

Le richieste vanno accodate.

Il disco riceve le seguenti richieste:

disk.seek(cyl)

disk.access(op,bk,head,sector) (op==RD per read, op==WR per write)

Entrambe le chiamate al disco **non** sono bloccanti, il disco opera in maniera asincrona e chiama elevator.interrupt() quando ha terminato.

Scrivere il monitor elevator (che ha DUE procedure entry readwrite e interrupt) e che fa uso dell'oggetto disk predefinito.

Esercizio 2:

Si consideri l'algoritmo di allocazione di memoria basato su "buddy list" utilizzato in Linux. Supponiamo di avere a disposizione 128 pagine di memoria fisica. Mostrare la sequenza di allocazione delle pagine corrispondente alla sequenza di richieste di allocazioni seguente:

32 pagine - 8 pagine - 4 pagine - 64 pagine - 16 pagine

Esercizio 3:

Rispondete alla domanda (x*10+y)%7 dove y e' la terzultima e x la penultima cifra del vostro numero di matricola.

0. Spiegare il concetto di anomalia di Belady.
1. Spiegare il concetto di thrashing.
2. Spiegare il concetto di thread.
3. Spiegare il meccanismo di calcolo approssimato della lunghezza del CPU burst in SJF.
4. Spiegare il meccanismo dell'allocazione indicizzata in UNIX.
5. Spiegare il funzionamento del meccanismo di DMA.
6. Spiegare le condizioni necessarie affinchè si verifichi un deadlock.