

Esercitazioni di Sistemi Operativi

Le esercitazioni vanno realizzate in ambiente UNIX (oppure LINUX).

Problemi di mutua esclusione nel modello a memoria comune

1. Scrivere una applicazione concorrente che generi due processi figlio che competano per l'uso di un buffer di memoria (un processo figlio si comporti da scrittore e l'altro da lettore).

Problemi di cooperazione nel modello a memoria comune.

2. Scrivere una applicazione concorrente che implementi il problema del Produttore/ Consumatore.
1a soluzione: il programma crei due processi che agiscono, rispettivamente, da produttore e consumatore, comunicando attraverso un unico buffer di memoria.

2a soluzione: i processi produttore e consumatore comunicano attraverso un pool di buffer di memoria gestito come vettore circolare.
3. Scrivere una applicazione concorrente che implementi il problema del Produttore/ Consumatore nella variante che prevede più produttori e più consumatori e l'impiego di un pool di buffer di memoria.
4. Scrivere una applicazione concorrente che implementi il problema dei Lettori/ Scrittori .
1a soluzione: lettori e scrittori accedano ad un pool di buffer di memoria condivisa, con attesa indefinita dei processi scrittori.
2a soluzione: lettori e scrittori accedano ad un pool di buffer di memoria condivisa, con attesa indefinita sia dei processi scrittori che dei lettori.

Problemi di cooperazione nel modello a scambio di messaggi.

5. Scrivere una applicazione concorrente che implementi il problema del Produttore/ Consumatore utilizzando le primitive send e receive per la comunicazione mediante code di messaggi. (Costruzione di un protocollo sincrono mediante le primitive di scambio messaggi asincrone).
6. Scrivere una applicazione concorrente che implementi uno schedulatore di processi che gestisce tre livelli di priorità usando le code di messaggi.
7. Si realizzi un processo che riceve messaggi da altri processi mediante una coda di messaggi e risponde con un intero che dice quanti messaggi sono stati ricevuti fino a quel momento. Per la comunicazione delle risposte si deve usare la stessa coda di messaggi delle richieste. La coda di messaggi deve essere creata dal primo processo che ne fa uso, non necessariamente il servente.

Suggerimenti: Si usi un tipo particolare per i messaggi spediti verso il processo contatore, e il pid del processo mittente per i messaggi di risposta.

Impiego del costruito monitor

8. Si implementi il problema dei lettori/ scrittori utilizzando il costruito monitor.
9. Si implementi il problema produttori/ consumatori utilizzando il costruito monitor.
10. Scrivere una applicazione concorrente che implementi mediante un monitor il problema della gestione di un'unità a disco a teste mobili (secondo l'algoritmo dell'ascensore).

Impiego di file

11. Si realizzino due processi che interagiscono tramite memoria comune al fine di leggere dati da un disco. Il primo processo legge da disco blocchi di dati (di un file il cui nome viene letto da programma) e li deposita in un insieme di buffer allocati in memoria comune. Il secondo processo stampa a video ciò che trova in tali buffer. Lo studente è libero di scegliere il tipo di sincronizzazione fra i processi, purché corretta, motivando opportunamente la scelta.

12. Si realizzino due processi che interagiscono tramite memoria comune al fine di duplicare dati da un disco. Il primo processo legge da disco blocchi di dati (di un file il cui nome viene letto da programma) e li deposita in un insieme di buffer allocati in memoria comune. Il secondo processo crea un secondo file (il cui nome viene letto da programma) e scrive sul file ciò che trova in tali buffer. Lo studente è libero di scegliere il tipo di sincronizzazione fra i processi, purché corretta, motivando opportunamente la scelta.