

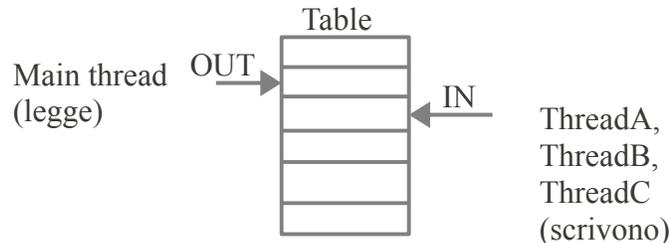
Esercizi da portare all'esame di Sistemi Operativi 15/16

- 1) Realizzare le funzioni PUnlock() e PLock() in C usando l'algoritmo di Peterson. Provare le funzioni nel seguente problema:
“Tre thread concorrenti T1, T2, T3 usano 3 variabili condivise, var0, var1, var2, in un ciclo infinito in cui ciascun uso corrisponde ad un ritardo random. T1 usa var0 e var1, T2 usa var1 e var2, T3 usa var2 e var0. Quando una variabile viene usata, il main scrive una tabella:

```
|T1 usa/non usa var0; usa/non usa var1|  
|T2 usa/non usa var1; usa/non usa var2|  
|T3 usa/non usa var2; usa/non usa var1|
```

- 2) Realizzare le funzioni LUnlock() e LLock() in C usando l'algoritmo di Lamport. Provare le funzioni nello stesso problema di mutua esclusione del punto precedente.

- 3) Realizzare il seguente programma concorrente in Java (chiamato Spooling.java).
Si tratta di un thread che legge continuamente un vettore di stringhe Table usando l'indice OUT. Il thread deve accorgersi se la stringa all'indice OUT è già stata letta o no. Se la stringa non è già stata letta, viene letta e scritta sulla console. L'indice OUT, inizializzato a 0, viene incrementato dopo ogni lettura. Dopo ogni lettura, il thread aspetta un tempo random. Naturalmente OUT non deve superare IN. Ci sono poi tre thread che scrivono in istanti random (scrittura asincrona) nella tabella all'indice condiviso IN la stringa: “Sono il Thread xx”, dove xx è il numero del thread. Dopo ogni scrittura l'indice IN viene incrementato.



Si tratta in sintesi di un produttore/consumatore con tre consumatori.
Il programma risolve i problemi di mutua esclusione nell'incremento dell'indice.

- 4) Realizzare il seguente programma concorrente in Java (chiamato MaxParallelo.java). Sia dato un array A di 100 interi.

Si tratta di trovare il numero più alto e metterlo all'ultimo posto, 99. Sequenzialmente questa operazione verrebbe fatta con la funzione

```
void f(A){for(i=0; i<99;i++){
    if(A[i] > A[i+1]) {
        t=A[i];
        A[i]=A[i+1];
        A[i+1]=t;}
}
```

Creare 20 thread, ciascuno dei quali applica la funzione f su una porzione di 5 elementi dell'array. Precisamente, il thread i-esimo applica la funzione f su A[5i]...A[5i+4], cioè su sottoarray di 5 elementi.

Abbiamo così 20 numeri, che sono i maggiori dei propri sotto array di 5 elementi, e che si trovano agli indici 4, 9, 14, ..., 5i+4, ..., 99. In altri termini, questi 20 indici si possono anche esprimere così: 5*0+4, 5*1+4, 5*2+4, ..., 5*i+4, ..., 5*19+4. Gli elementi di questa sequenza di 20 numeri vengono gestiti da quattro thread che applicano la funzione f sugli elementi dei 4 sotto array A[5*i+4], A[5*(i+1)+4], A[5*(i+2)+4], A[5*(i+3)+4], A[5*(i+4)+4], dove ogni sotto array ha 5 elementi. Vengono così generati 4 numeri che vengono gestiti da un thread che applica la funzione f su di essi per trovarne il massimo. Il main thread aspetta la terminazione di tutti i thread e stampa infine il valore più alto.

- 5) Realizzare un programma in C che in una fase iniziale carica in `char**pt` un testo ASCII formato da molte righe di un'ottantina di caratteri ciascuna. La matrice di stringhe e l'indice di riga sono definite nella seguente struttura: `struct{char ** pt; int index;}testo;`

Il testo viene visualizzato da tre pthread che fanno continuamente (fino alla conclusione del testo) le seguenti cose:

- ricevono l'indirizzo della struttura come argomento,
- incrementano index in mutua esclusione e
- visualizzano la riga index del testo. Scrivono cioè la stringa `pt[index]`.

Caricare inizialmente nella matrice di stringhe un documento con qualche decina di righe.

- 6) Una linea aerea si appoggia su una agenzia di viaggio dove ci sono alcuni operatori che dialogano con il pubblico. L'agenzia usa un database dove annota le prenotazioni. Gli operatori eseguono la prenotazione chiesta dal cliente seguendo le seguenti operazioni:
- a) leggono il database per vedere se il posto è disponibile
 - b) comunicano al cliente la disponibilità del posto
 - c) il cliente accetta la prenotazione
 - d) effettuano la prenotazione modificando il database
 - e) l'operatore conferma al cliente la prenotazione

Realizzare il programma descrivendo gli operatori con i posix thread in linguaggio C.

- 7) Scrivere due programmi nel linguaggio C che funzionano nel seguente modo:
Il primo genera una sequenza di 10000 numeri random interi da 0 a 10000 nel buffer BUF e li analizza uno alla volta. Se il numero è pari, invia il segnale SIGHUP (1) all'altro processo. Se il numero è dispari, invia il segnale SIGINT (2) all'altro processo.
Il secondo processo conta il numero di segnali SIGHUP e SIGINT che riceve e stampa il numero sul terminale. Provare il programma con diversi generatori di variabili casuali (uniforme, gaussiana, esponenziale) corrispondenti a diverse distribuzioni (da trovare in rete).
Se ci sono problemi con questi generatori, mandate una richiesta specifica al docente del corso di SO.
- 8) Modificare lo script alla pagina 30 di SystemProgram.odp inserendo la possibilità di attivare in pipe due comandi dati al prompt, tipo `%ls -l|wc -l`
- 9) Scrivere uno script in Bash shell per visualizzare un istogramma che mostra il tempo di esecuzione dei programmi che stanno eseguendo nel sistema e che appartengono all'utente dato come argomento allo script.