

Esercizi Programmazione Concorrente

Esercizio 1- Prelievo Moduli

Nella segreteria dell'Università N studenti devono compilare il modulo per la Domanda di Laurea.

All'apertura della segreteria una impiegata ha disposto M copie del modulo sulla scrivania ($N > M$). Ogni studente prende una copia del modulo alla volta. Quando le copie del modulo sono terminate, gli studenti si bloccano in attesa che l'impiegata rifornisca la scrivania di moduli.

Lo studente che preleva l'ultima copia del modulo dalla scrivania della segreteria, chiama l'impiegata in attesa affinché vada a prelevare altre M copie del modulo in archivio.

Quando viene avvisata che sono terminate le M copie del modulo sulla scrivania, l'impiegata: a) preleva altre M copie dall'archivio, b) le dispone sulla scrivania, c) sveglia gli studenti in attesa e d) si sospende, mettendosi a sbrigare le pratiche amministrative inavase.

Inizialmente ci sono M copie del modulo a disposizione degli studenti, sulla scrivania della segreteria.

Modellare ed implementare il sistema descritto, utilizzando dei thread POSIX per ciascuna figura (segretaria e studenti) ed avvalendosi delle opportune strutture dati per la sincronizzazione.

Esercizi Programmazione Concorrente

Esercizio 2- Ponte a singola corsia

Nella nota località di Gattolino, c'è un traballante ponte che supera il fosso che separa la chiesa dall'osteria. Il ponte ha una sola corsia e permette il transito ad una sola fila di auto.

Il ponte è talmente corto che chi sta per entrare vede se qualcuno, dall'altra parte, sta transitando sul ponte in direzione opposta.

Quando alcune auto stanno percorrendo il ponte, le auto che procedono in direzione opposta devono aspettare fino a che tutti i veicoli hanno liberato il ponte, poi possono incolonnarsi a loro volta ed impegnare il ponte. Ciascuna auto impiega 1 secondo per attraversare il ponte. Poiché il ponte è corto, sul ponte possono stare fino a 5 auto contemporaneamente.

Modellare ed implementare il sistema descritto, utilizzando dei thread POSIX per ciascuna figura (macchine nelle due direzioni) ed avvalendosi delle opportune strutture dati per la sincronizzazione. Si ipotizzi che le macchine arrivino sul ponte su ciascun lato circa a gruppi di 5 auto ogni secondo.

Esercizio 3-Variante ubriaca del Ponte a singola corsia

Gli avventori dell'osteria escono ubriachi. Circa un'auto ogni 10 che entra nel ponte dal lato osteria, CADE nel fosso durante l'attraversamento e non viene recuperata. Le altre auto fanno finta di niente e proseguono gli attraversamenti.

Modellare il sistema mediante thread POSIX, estendendo il modello precedente.

Esercizi Programmazione Concorrente

Esercizio 4- Pizzeria Dal Zozzo

Nella pizzeria da asporto, il pizzaiolo Pino Dal Zozzo prepara le pizze fintanto che il bancone non è pieno. Al massimo sul bancone possono essere posizionate N pizze. Una volta che il bancone è pieno, il pizzaiolo si sospende, mettendosi a leggere il giornale. Man mano che arrivano i clienti, questi si prendono una pizza a testa e se la mangiano ed escono. Se il bancone è vuoto i clienti si sospendono in attesa che, sul bancone, sia disponibile almeno una pizza. Il cliente che prendere l'ultima pizza disponibile sul bancone risveglia il pizzaiolo, che si rimette a preparare le pizze.

Inizialmente il bancone è pieno di pizze (N), il pizzaiolo sta leggendo il giornale e non ci sono clienti in attesa.

Modellare ed implementare il sistema descritto, utilizzando dei thread POSIX per ciascuna figura (pizzaiolo e clienti) ed avvalendosi delle opportune strutture dati per la sincronizzazione.

Esercizi Programmazione Concorrente

Esercizio 5 - Legione Straniera

Nella caserma della legione straniera, alla mattina ci si mette in coda per radersi.

N legionari attendono, in due diverse file, che vengano loro assegnati, in una fila un rasoio e nell'altra fila una scodella con la schiuma da barba.

Il numero di rasoi è $M < N$. Il numero delle scodelle è $K < M$.

Dopo essersi rasati, i legionari restituiscono i rasoi e le scodelle ai due sottoufficiali che gestiscono una fila ciascuno.

Modellare ed implementare il sistema descritto, utilizzando dei thread POSIX per ciascuna figura (i due sottoufficiali che distribuiscono e i legionari) ed avvalendosi delle opportune strutture dati per la sincronizzazione.

Esercizi Programmazione Concorrente

Esercizio 6 - Fornaio FIFO (1/2)

Un fornaio ha deciso di installare una macchinetta che distribuisce dei biglietti con un numero crescente per dirimere le liti tra i clienti in coda per essere serviti. Ciascun cliente che entra accede al distributore e prende un biglietto, poi aspetta di essere chiamato.

Ogni volta che il fornaio finisce di servire un cliente, guarda se c'è qualcuno in attesa.

Se nel negozio non c'è nessuno, il fornaio aspetta.

Appena entra un cliente, il fornaio aspetta che il cliente abbia preso il biglietto, poi il fornaio guarda il biglietto consegnatogli dall'ultimo cliente che ha servito e chiama il numero del biglietto successivo. Il cliente chiamato consegna al fornaio il proprio biglietto e viene servito.

Se invece nel negozio c'è qualcuno in coda, il fornaio guarda il biglietto consegnatogli dal cliente appena servito e chiama il numero del biglietto successivo.

Il cliente chiamato si avvicina al bancone, consegna al fornaio il proprio biglietto e viene servito.

Ipotizziamo che:

- ad apertura negozio il fornaio prende lui stesso il biglietto col numero 0.
- il primo cliente prende il biglietto col numero 1.
- nessun cliente va via dopo avere preso il biglietto; tutti restano in coda fino ad essere serviti.

Modellare ed implementare il sistema descritto, utilizzando dei thread POSIX per ciascuna figura (il fornaio, i clienti) ed avvalendosi delle opportune strutture dati per la sincronizzazione.

Occorre GARANTIRE che vengano serviti i clienti nell'ordine con cui hanno preso i biglietti.

Il distributore di biglietti deve essere acceduto da un cliente alla volta.

Usare una mutex aggiuntiva per proteggere, separatamente da altre variabili, la sola variabile **contatore**, inizializzata a 0, che viene usata dal distributore di biglietti.

Esercizi Programmazione Concorrente

Continuazione Esercizio 6 - Fornaio FIFO (2/2)

Si consiglia di utilizzare una coda di attesa in cui viene mantenuta una struttura dati di tipo `{ unsigned int valore; pthread_t tid; }`

continuare