Laboratorio su Programmazione Concorrente in C

Problemi classici e derivati

Nona lezione di laboratorio …

Lezione 9 in laboratorio: concorrenza.

Il menù di oggi:

- Sincronizzazione Circolare con passaggio di testimone,
- Estensione del barbiere
- Estensione di produttori e consumatori
- Riscrittura di lettori e scrittori

... e ... gran finale .....

- Masochisti a San Zaccaria (è un esercizio un po' da masochisti)

Esercizio 31: Staffetta 4x400 con passaggio sicuro del testimone

Una squadra di 4 atleti si esercita in pista correndo la staffetta 4x100.

Uno alla volta ciascun atleta fa un giro di pista tenendo in mano il testimone.

Alla fine del giro l'atleta j-esimo poggia il testimone in mano al successivo atleta (j+1)-esimo ma non molla la presa sul testimone.

L'atleta successivo (j+1)-esimo sentito il testimone in mano, grida per far sapere al precedente atleta j-esimo che ha preso il testimone.

Sentito l'urlo, l'atleta precedente j-esimo molla la presa e grida al successivo (j+1)-esimo di partire.

Il successivo (j+1)-esimo sente l'urlo e parte a tutta birra per il suo giro di pista.

Gli atleti corrono nell'ordine stabilito dal loro indice, impiegano 1 secondo a fare il giro di pista, poi si rimettono in fila per fare un altro giro quando sarà il loro turno.

Modellare ed implementare il sistema descritto, utilizzando il più bel linguaggio del mondo (ANSI C, ovviamente) e dei thread POSIX per ciascuna figura (ciascun atleta) ed avvalendosi delle opportune strutture dati per la sincronizzazione.
Scrivere il Makefile e inserire il controllo d'errore delle funzioni.

Soluzione es31: staffetta4x400

http://www.cs.unibo.it/~ghini/didattica/sistemioperativi/ESERCIZI/es31_staffetta4x400.tgz

Esercizio 32: Barbiere prudente. (1)

L'esercizio è una estensione del "barbiere che saluta" presentato a lezione.
Rispetto a quell'esercizio c'è una sola differenza che nella descrizione è in rosso.

- Il barbiere serve il cliente di barba e capelli, poi dice al cliente di andarsene **e aspetta che il cliente se ne vada.**
- **Il cliente si alza e dice al barbiere che la poltrona è libera**, e se ne va, tornerà nel negozio dopo 3 secondi**.**

- In sostanza, il barbiere prima di chiamare un altro cliente, aspetta che il cliente che ha appena finito di servire si alzi dalla poltrona.

Nella slide successiva è proposta la descrizione completa del problema.

Implementare il sistema, con 5 sedie, usando 5 thread cliente e 1 thread barbiere.
Scrivere il Makefile e scrivere i necessari controlli di errore.

SUGGERIMENTO: riutilizzare il più possibile l'esempio barbiere che saluta.
presentato a lezione e disponibile qui:
http://www.cs.unibo.it/~ghini/didattica/sistemioperativi/PTHREAD/BARBIERE/barbiereconsaluto.c

Soluzione es32: barbiere prudente

http://www.cs.unibo.it/~ghini/didattica/sistemioperativi/ESERCIZI/es32_barbiereprudente.tgz

Esercizio 32: Barbiere prudente. (2)

Descrizione completa

- Un barbiere è un maniaco del controllo e vuole perciò organizzare tutti i movimenti dei suoi clienti. Nel negozio del barbiere ci sono **N sedie** in cui stanno i clienti in attesa del taglio ed **1 poltrona** su cui sta il cliente mentre viene servito e dove pisola il barbiere mentre aspetta che arrivi un qualche cliente.
- Se non ci sono cliente il barbiere pisola.
- Quando un cliente arriva, se c'e' qualche sedia libera allora lui entra e si siede aspettando di essere servito.
- Se tutte le sedie sono occupate invece il cliente se ne va subito.
- Se le sedie sono tutte vuote e il barbiere sta pisolando, il cliente si siede e poi chiama il barbiere per svegliarlo.
- Allora il barbiere si sveglia, dice a un cliente di sedersi nella poltrona e attende che questo si sieda.
- Il cliente chiamato dal barbiere si siede e dice al barbiere di essere seduto e pronto ad essere servito, poi si mette in attesa che il barbiere gli dica di andarsene.
- Il barbiere serve il cliente di barba e capelli, poi dice al cliente di andarsene **e aspetta che il cliente se ne vada.**
- **Il cliente si alza e dice al barbiere che la poltrona è libera**, e se ne va, tornerà nel negozio dopo 3 secondi**.**
- Il barbiere chiama un altro cliente perché sa che la poltrona è libera.

Le parti in rosso sono le sole differenze rispetto al caso del "barbiere che saluta" descritto a lezione.

Esercizio 33: N Produttori, M Consumatori suddivisi in due categorie.

Abbiamo un buffer di scambio, di tipo long int, utilizzato da Produttori e Consumatori per scambiarsi dati. I consumatori sono divisi in due categorie A e B.

I consumatori devono accedere a turni di categoria, prima un consumatore di tipo A, poi un consumatore di tipo B, etc etc. All'interno di una stessa categoria di consumatori, non occorre un turno ferreo ma è sufficiente una generica equità di accesso al buffer.

Ciascun Produttore impiega un secondo a produrre il dato e impiega un secondo a depositare nel buffer, in mutua esclusione, il dato prodotto, una volta che ha avuto il permesso di depositarlo.

Analogamente, ciascun Consumatore impiega un secondo a prelevare, in mutua esclusione, il dato dal buffer condiviso, e impiega un secondo a consumare il dato dopo averlo prelevato dal buffer.

Ci sono 10 Produttori, 3 Consumatori A e 4 Consumatori B.

Implementare il sistema descritto, facendo uso di POSIX thread per ciascuna figura (produttori e consumatori).
Scrivere il Makefile e assicurarsi di avere inserito tutti i necessari controlli di errore.

Soluzione es33: 2 categorie cons

http://www.cs.unibo.it/~ghini/didattica/sistemioperativi/ESERCIZI/es33_2categorieCons.tgz

Esercizio 34: Lettori e Scrittori senza mutex annidate.

Considerate il problema degli N Lettori ed M Scrittori su un buffer condiviso.

Ricorderete che la soluzione proposta per modellare il sistema

http://www.cs.unibo.it/~ghini/didattica/sistemioperativi/PTHREAD/NLett_MScritt/NLettoriMScrittori.c

http://www.cs.unibo.it/~ghini/didattica/sistemioperativi/PTHREAD/NLett_MScritt/Makefile

utilizzava due variabili di tipo pthread_mutex_t ed effettuava due chiamate annidate alla pthread_mutex_lock per proteggere il buffer.

Realizzate una variante della soluzione che sfrutti qualche variabile di tipo pthread_cond_t per evitare di fare le due chiamate annidate alla lock.
Scrivere il Makefile e assicurarsi di avere inserito tutti i necessari controlli di errore.

Ipotizziamo che ci siano 10 Lettori e 5 Scrittori,
che ogni lettore impieghi 1/10 di secondo a leggere il dato dal buffer e poi 1/10 di secondo a usare il dato.
che ogni scrittore impieghi 1/10 di secondo a produrre il dato e poi 1/10 di secondo a scrivere il dato nel buffer.

Soluzione es34: LettoriScrittori senza mutex annidate

http://www.cs.unibo.it/~ghini/didattica/sistemioperativi/ESERCIZI/es34_LettoriScrittori_SenzaMutexAnnidate.tgz

Esercizio 35: Masochisti a San Zaccaria

Nella periferia di San Zaccaria (RA) c'è una lunga strada ad anello che attraversa uno stretto ponte sul torrente Bevano. Indichiamo con gli indici 0 e 1 i due estremi del ponte. Frotte di masochisti girano in tondo in auto su questa strada attraversando più e più volte il ponte. Le auto di tipo **A** seguono questo percorso: arrivano al ponte dal lato 0, attraversano e vanno nel lato 1, fanno tutto l'anello e tornano a 0. Le auto di tipo **B** fanno il percorso nel senso opposto.
Le auto impiegano 2 secondi a fare il giro, fino a tornare al ponte all'altro lato. Allo scadere dei propri 2 secondi ogni auto cerca di collocarsi in prima posizione a quel lato del ponte.
Se due o più auto arrivano al ponte dallo stesso lato del ponte, solo la prima che arriva può andare in prima posizione e tentare di attraversare il ponte. Le altre auto aspettano che la prima attraversi il ponte e poi cercano a loro volta di collocarsi in prima posizione da quel lato del ponte per poi attraversare.
Arrivati in prima posizione i masochisti si fermano e controllano se all'altro lato c'è un auto in prima posizione. **Se non c'é nessuno, il masochista si ferma** e aspetta che arrivi qualcuno all'altro lato del ponte, **con la speranza masochistica** di essere coinvolto in un incidente quando attraverserà il ponte. Quando c'è un'auto in prima posizione ad entrambi i lati del ponte, le due auto devono organizzarsi tra loro in qualche modo, sfruttando variabili globali e sincronizzazioni, per attraversare entrambe il ponte, una alla volta, in un qualche ordine, prima di ogni altra auto. Ogni masochista attraversa gridando "aaaaah!". Ci sono 3 auto A e 4 auto B.
Modellare ed implementare il sistema descritto, **utilizzando dei thread** per ciascuna auto, avvalendosi delle opportune strutture dati per la sincronizzazione. Scrivere il Makefile per compilare e linkare i sorgenti. Inserire il necessario controllo di errore. In caso di errore grave, terminare il programma producendo un avviso a video. **Soluzioni es35** in
http://www.cs.unibo.it/~ghini/didattica/sistemioperativi/ESERCIZI/es35_masochistiasanzaccaria.tgz