

Prova Pratica 057 - online

La prova consiste di due esercizi, il primo è un esercizio di programmazione concorrente, il secondo è un esercizio bash.

Avrete a disposizione circa 50 minuti per risolvere i due esercizi e consegnarle la soluzione richiesta.

Potete usare il materiale contenuto in <http://www.cs.unibo.it/~ghini/didattica/TREE4OS2021.tgz> che dovete già avere scaricato prima di questo esame

Non potete navigare in internet cercando informazioni all'esterno.
Ovviamente, non potete comunicare con nessuno, in nessun modo.

Salvate i files spesso, per premunirsi in caso di crash.

Esercizio Esame Pratica Online - 159 - birra (1/2)

Un **distributore** di birra poco igienico, adatto ai negazionisti del covid, funziona così:

Quando un **cliente** deposita un bicchiere vuoto nel piatto del distributore, il distributore appena possibile riempie il bicchiere impiegando circa 2 secondi, in modo che qualcuno poi possa prendere il bicchiere pieno. C'è spazio per un solo bicchiere nel piatto del distributore.

Un cliente si mette in coda per depositare il bicchiere vuoto, appena può deposita un bicchiere vuoto nel piatto del distributore impiegando circa 2 secondi per appoggiare il bicchiere nel piatto. Dopo avere depositato il bicchiere, il cliente va in bagno a svuotare la vescica, impiegando un tempo casuale compreso tra 2 e 4 secondi, poi torna al distributore e si mette in coda per prendere un bicchiere pieno. Dopo avere preso il bicchiere pieno, il cliente impiega un tempo casuale variabile tra 5 e 8 secondi per bere e svuotare il bicchiere. Poi cerca di depositare il bicchiere vuoto, poi fa pipì e poi cerca di bere ancora e così via all'infinito.

Le due code, quella per depositare il bicchiere vuoto e quella per prelevare il bicchiere pieno, NON SONO delle code FIFO. L'unico vincolo che devono rispettare è questo: SE ARRIVA UN CLIENTE che deve svolgere l'operazione di "**deposito**" ["**prelievo**"] e quella operazione è possibile subito perché il distributore "ha il piatto vuoto" ["ha un bicchiere pieno nel piatto"] ALLORA il Cliente svolge subito l'operazione SOLO SE NON C' E' GIA' UN ALTRO CLIENTE IN CODA che deve svolgere la sua stessa operazione. Se invece c'è già qualcuno in coda, allora il cliente neo-arrivato si mette in coda.

All'inizio ci sono 4 clienti col bicchiere pieno e il piatto del distributore è vuoto.

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

Utilizzare come base dell'esercizio i seguenti files, aggiungendo il codice necessario:

birra.c DBGpthread.c DBGpthread.c printerror.h

SEGUE IN PAGINA SUCCESSIVA

Esercizio Esame Pratica Online - 159 - birra (2/2)

CONTINUAZIONE

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.

Non è obbligatorio utilizzare, nel codice che aggiungerete voi, le funzioni DBG* contenute nei files DBGpthread.* ma è fortemente consigliato.

Le variabili **NumClientiInCodaDeposito** e **NumClientiInCodaPrelievo** indicano il numero di clienti rispettivamente già in attesa di depositare il bicchiere vuoto e di prelevare un bicchiere pieno.

La variabile StatoDistributore vale 0 (PIATTOVUOTO) se il piatto del distributore è vuoto, vale 1 (APPOGGIANDO BICCHIERE) se un cliente sta depositando il bicchiere vuoto nel piatto, vale 2 (BICCHIEREVUOTO) se c'è un bicchiere vuoto nel piatto, vale 3 (BICCHIEREPIENO) se c'è un bicchiere pieno nel piatto.

Le condition variable **condAttesaPiattoVuoto** e **condAttesaBicchierePieno** servono a mantenere in coda rispettivamente i clienti in attesa di depositare un bicchiere vuoto nel piatto del distributore ed i clienti in attesa di prelevare un bicchiere pieno.

La condition variable **condDistributoreAttendeBicchiereVuoto** sveglia il distributore affinché riempia il bicchiere.

Per implementare le attese di un tempo variabile, usare la funzione **attendi(int min, int max)**. La funzione **attendi(int min, int max)** implementata nel file **birra.c** genera un numero casuale X compreso tra min e max ed attende X secondi prima di restituire il controllo al chiamante.

Le variabili globali già inserite sono inizializzate nel main.

Potete aggiungere variabili globali, se vi servono, e ricordate di inizializzarle.

Esercizio Esame Pratica - 160 - nAnB

Realizzare uno script `nAnB.sh` che prende come argomento una stringa di testo e verifica che la stringa sia formata esattamente da `N` caratteri `A` seguiti da `N` caratteri `B`, con un `N` qualunque maggiore di zero.

Se la stringa corrisponde ai requisiti lo script scrive sullo standard output il Numero `N`.

Altrimenti scrive 0.

Ad esempio, passando come argomento:

`AAABBB` vedo in output 3

`ADABBB` vedo in output 0 perché c'è carattere diverso da `A` e da `B`

`AAB` vedo in output 0

`AabB` vedo in output 0

Realizzare lo script richiesto completando lo script base allegato.

Suggerimento: vi ricordo l'esistenza dei seguenti operatori per le variabili bash

`${VAR:offset}` sottostringa che parte dal offset-esimo carattere del contenuto della variabile di nome `VAR`

`${VAR:offset:length}` sottostringa lunga `length` che parte dal offset-esimo carattere del contenuto della variabile di nome `VAR`