Prova Pratica 056 - 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 **157 - bisce (1/2)**

In una casa di campagna pericolante vive una colonia di bisce. Quando le bisce diventano adulte si raggruppano assieme formando un groviglio di esattamente **3** bisce.
Le bisce che eccedono le **3** formano un groviglio successivo. Ciascun groviglio è individuato univocamente da un numero intero crescente. Possono essere presenti contemporaneamente più grovigli di bisce, ma solo uno per volta può contenere meno di 3 bisce, le quali sono in attesa che il groviglio si completi. Man mano che le bisce diventano adulte si aggiungono al groviglio non completo o ne creano uno nuovi se sono già tutti completi.
Dopo che un singolo groviglio si è completato, ciascuna biscia del groviglio copula a caso nel groviglio per 4 secondi e poi muore. Prima di morire, una sola delle bisce di ciascun groviglio produce 3 bisce figlie. La biscia che produce le figlie è l'ultima che è entrata nel suo groviglio. Le bisce neonate diventano adulte in un tempo variabile tra 3 e 5 secondi e poi si aggrovigliano.

All'inizio ci sono **5 bisce appena nate** e nessun groviglio già formato. Quindi al massimo potranno esistere contemporaneamente 2 grovigli, completi o no. Metterò le info su questi grovigli in un vettore di 10 posizioni basandomi sull'identificatore del groviglio.

Modellare ed implementare il sistema descritto, utilizzando dei thread POSIX per ciascuna figura (**biscia**) ed avvalendosi delle opportune strutture dati per la sincronizzazione.
Utilizzare come base dell'esercizio i seguenti files, aggiungendo il codice necessario:

biscie.c DBGpthread.c DBGpthread.c printerror.h

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.

VEDERE DETTAGLI SULL'IMPLEMENTAZIONE DELLA BASE NELLA SUCCESSIVA PAGINA

Esercizio Esame Pratica Online **157 - bisce (2/2)**

CONTINUAZIONE

La variabile globale intera **IndiceUltimoGroviglio** indica l'ultimo groviglio che si sta formando.

La var globale intera **BisceUltimoGroviglio** indica quante sono le bisce che attualmente compongono il groviglio il cui indice è **IndiceUltimoGroviglio**.

Il vettore di **10** interi **StatoGroviglio [10]** indica se un groviglio di indice X si è completato.

Ad esempio, se una biscia sa di appartenere al groviglio di indice 57 e vuole sapere se il suo groviglio si è completato e quindi lei può iniziare l'accoppiamento, allora la biscia guarda nel vettore nella posizione 57**%10**; se ci trova un 1 allora può accoppiarsi per un secondo e poi morire, se trova uno 0 allora deve aspettare ancora. Esistono le seguenti funzioni:

void GroviglioNonCompleto(intIndiceGroviglio) { *StatoGroviglio[IndiceGroviglio%10]=0;* **}**

configura il groviglio IndiceGroviglio come non ancora completato.

void GroviglioCompleto(int IndiceGroviglio) { *StatoGroviglio*[IndiceGroviglio%10]=1; **}**

configura il groviglio IndiceGroviglio come completato.

int SituazioneGroviglio(int IndiceGroviglio);

restituisce 1 se il groviglio è completo, 0 altrimenti.

Ovviamente anche il vettore StatoGroviglio e le 3 funzioni sopra descritte DEVONO ESSERE ACCEDUTE in mutua esclusione.

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 **bisce.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.

Ricordate di inizializzare nel main le vostre variabili globali.

Esercizio Esame Pratica **- 158 - asterischi**

Realizzare uno script asterischi.sh che conta quante sono le righe che contengono almeno un asterisco * in tutti i file il cui nome termina con .h nella parte di filesystem a partire dalla directory /usr/include/

Lo script deve stampare sul proprio standard error il numero delle righe contate.