

Prova Pratica 070 - 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 vi dovete già essere scaricati prima

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 - 185 - catapulte

Siamo nel 1460, si sta combattendo una battaglia tra i Riminesi e San Marino. I contendenti sono schierati su due opposte trincee ciascuno con 4 catapulte indicizzate da 0 a 3 su un lato e da 4 a 7 sull'altro lato. Ogni 5 secondi un **trombone** suona ed allora ciascuna **catapulta ancora funzionante** dei due schieramenti **crea** un nuovo **proiettile** e lo lancia verso gli avversari.

Ciascuna catapulta ancora in funzione, quindi, lancia un **proiettile** scegliendo ogni volta casualmente, come bersaglio, una delle catapulte avversarie, non distinguendo tra catapulte in funzione o catapulte distrutte.

Il volo del proiettile, a cui è stato passato l'indice della catapulta da colpire, dura 3 secondi, poi il proiettile si schianta sulla catapulta prescelta e, se questa è ancora in funzione, la distrugge. Anche il proiettile viene distrutto.

La catapulta colpita viene distrutta e, se non ci sono più catapulte alleate in funzione, la catapulta colpita per ultima termina la battaglia ed il programma.

Modellare ed implementare il sistema descritto, utilizzando dei thread POSIX per ciascuna figura (**catapulta**, **trombone** e **proiettile**) ed avvalendosi delle opportune strutture dati per la sincronizzazione. Il **main** stesso svolge il ruolo di thread **trombone**.

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

catapulte.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.

La funzione `attendi(int min, int max)` implementata nel file **catapulte.c** genera un numero casuale X compreso tra min e max ed attende X secondi prima di restituire il controllo al chiamante. Non è obbligatorio utilizzare, nel codice che aggiungerete voi, le funzioni `DBG*` contenute nei files `DBGpthread.*` ma è consigliato.

Esercizio Esame Pratica - 186 - **escludi.sh**

Implementare uno script `escludi.sh` che legge il file `/usr/include/stdio.h` e mette in output, nell'ordine in cui le ha lette, le righe di questo file che posseggono TUTTE le seguenti caratteristiche:

- 1) non contengono la stringa `"#"`
- 2) non contengono la stringa `".h"`
- 3) contengono la stringa `"int"`
- 4) hanno almeno 5 caratteri (decidete voi se considerare nel conto l'andata a capo finale di ciascuna riga oppure non considerarla).