Università degli Studi di Napoli Federico II Corso di Laurea in Ingegneria Informatica Esame di Sistemi Operativi Proff. Cinque, Cotroneo, Natella

Prova pratica del 13/04/2022 Durata della prova: 75 minuti

Lo studente completi il programma a corredo di questo documento, seguendo le seguenti indicazioni.

La prova sarà valutata come segue:

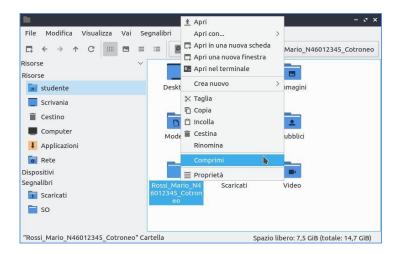
- A: Prova svolta correttamente.
- **B**: Il programma non esegue correttamente, con errori minori di programmazione o di concorrenza.
- C: Il programma non esegue correttamente, con errori significativi (voto max: 22).
- **INSUFFICIENTE**: Il programma non compila o non esegue, con errori gravi di sincronizzazione.

Istruzioni per la consegna dell'elaborato

L'elaborato dovrà essere svolto in una cartella dal nome: Cognome Nome Matricola Docente

Esempio:

In alternativa, è consentito creare il file compresso "tar" tramite l'interfaccia grafica.



All'interno della macchina virtuale, aprire il browser, accedere al team relativo all'esame, ed effettuare l'upload del file compresso, cliccando sul tasto relativo alla consegna. Attendere una notifica del docente prima di abbandonare l'aula.

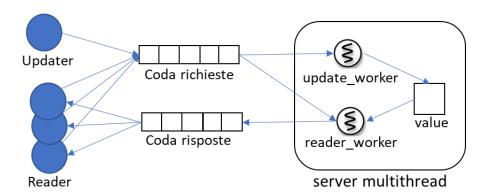
Testo della prova

Si realizzi in linguaggio C/C++ una applicazione **multiprocesso** che realizzi uno schema **client-server** con server **multithread**. Client e server comunicano attraverso due **code di messaggi UNIX**, una coda per le richieste e una coda per le risposte.

Il **client** genera 1 processo **updater** e 3 processi **reader**. Il processo updater effettua 10 richieste di aggiornamento, inviando al server un numero casuale attraverso la 'coda richieste'. I processi reader inviano ognuno 20 richieste di lettura al server sulla 'coda richieste', attendendo sulla 'coda risposte' un messaggio indirizzato a se e contenente il valore aggiornato.

Il server detiene una variabile condivisa value, e genera 1 thread update_worker e 1 thread reader_worker che lavorano sulla variabile condivisa. Il thread update_worker è in attesa delle richieste di aggiornamento sulla 'coda richieste'. Ogni richiesta viene gestita aggiungendo il valore ricevuto alla variabile condivisa value in mutua esclusione. Il thread reader_worker è in attesa delle richieste di lettura sulla medesima 'coda richieste'. Ogni richiesta viene gestita prelevando in mutua esclusione il valore della variabile condivisa value e rispondendo attraverso un messaggio indirizzato esattamente al reader che ha fatto richiesta, attraverso la 'coda risposte'.

I processi client e server vengono generati da un processo main attraverso la primitiva exec.



File da completare:

- header.h
- server.c
- client.c
- main.c