

Prova pratica del 08/07/2022
Durata della prova: 75 minuti

- **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.

```
mkdir Rossi_Mario_N46012345_Cotroneo
cd Rossi_Mario_N46012345_Cotroneo
.... copiare nella cartella i file forniti per l'esame ....
.... sviluppare il programma ....
.... per la consegna, dalla shell (assumendo di essere ancora nella cartella di lavoro),
    creare un file compresso ("tar") con i seguenti comandi:
cd ..
tar -cvfz ./Rossi_Mario_N46012345_Cotroneo.tar.gz ./Rossi_Mario_N46012345_Cotroneo
```

The screenshot shows a Windows File Explorer window. The top menu bar includes 'File', 'Modifica', 'Visualizza', 'Vai', and 'Segnalibri'. The left sidebar, under the 'Risorse' (Resources) section, lists 'studente' (selected), 'Scrivania' (Desktop), 'Cestino' (Recycle Bin), 'Computer', 'Applicazioni', 'Rete', 'Dispositivi', and 'Segnalibri'. The main pane shows a folder named 'Rossi_Mario_N46012345_Cotroneo' selected. A context menu is open over this folder, with the 'Comprimi' (Compress) option highlighted. Other menu options include 'Apri', 'Apri con...', 'Apri in una nuova scheda', 'Apri una nuova finestra', 'Apri nel terminale', 'Crea nuovo', 'Taglia', 'Copia', 'Incolla', 'Cestina', 'Rinomina', 'Proprietà', 'Scaricati', and 'Video'. The status bar at the bottom indicates the file name 'Rossi Mario N46012345 Cotroneo' and the available space 'Spazio libero: 7.5 GiB (totale: 14.7 GiB)'.

[illegible]

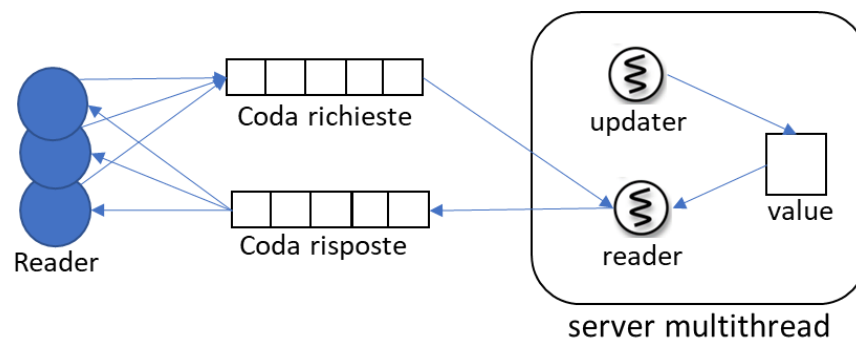
Testo della prova

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 processo client (file reader.c) genera 3 processi **reader**. Ogni reader invia 5 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 **updater** e tanti thread **reader** quante le richieste ricevute dai reader lato client attraverso la 'coda_richieste'. Tutti i thread lavorano sulla variabile condivisa accedendovi come **lettori-scrittori con starvation di entrambi**. L'updater è lo scrittore ed effettua 4 scritture del valore condiviso ad un valore casuale tra 1 e 10, aspettando un tempo casuale tra 1 e 2 secondi tra due richieste. Ogni reader è un lettore ed effettua una singola lettura del valore condiviso inviandolo al corrispondente reader lato client attraverso la 'coda_risposte'.

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



File da completare:

- server.c
- reader.c
- main.c