

Prova pratica del 19/04/2021
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 -czvf ./Rossi Mario N46012345 Cotroneo.tar.gz ./Rossi Mario N46012345 Cotroneo
```

The screenshot shows a Windows File Explorer window. The left sidebar has a navigation pane with 'Home' selected. The main area displays a folder named 'Rossi_Mario_N46012345_Cotroneo'. A right-click context menu is open over the folder, listing actions such as 'Copia su...', 'Sposta nel cestino', 'Rinomina...', 'Comprimi...', 'Condivisione della rete locale', 'Apri nel terminale', 'Preferito', and 'Proprietà'. The 'Comprimi...' option is highlighted. At the bottom of the window, a status bar indicates 'Selezionato «Rossi Mario N46012345 Cotroneo» (contiene 3 oggetti)'.

[illegible]

Testo della prova

Si realizzi in linguaggio C/C++ un programma client-server **multiprocesso** basato su **code di messaggi UNIX**. Il processo servente gestisce una risorsa rappresentata da una variabile di tipo intero. Il servente si pone in ascolto di **messaggi di richiesta di scrittura della risorsa**, che contengono un campo "tipo", il PID del richiedente, un valore intero da scrivere (tra 0 e 10), e un valore intero detto "**token**" (vedi più avanti). Il servente sovrascrive la risorsa con il valore della richiesta, e risponde al richiedente inviando un messaggio di risposta positiva (utilizzando il campo "tipo" e il PID per indirizzarlo al richiedente, e un secondo campo "esito" impostato ad "1").

Solo i client autorizzati possono inviare richieste al servente. Sia i processi client e sia il servente comunicano con un **processo autenticatore** per gestire la autenticazione dei client. Il processo autenticatore può ricevere due tipologie di messaggi:

1. Richieste di autorizzazione, provenienti dai client;
2. Conferme di autorizzazione, provenienti dal servente.

Inizialmente, i client inviano un **messaggio di richiesta autorizzazione** al processo autenticatore (indicando il proprio PID nella richiesta). Il processo autenticatore genera (casualmente) un numero tra 0 e 10000 (il "**token**"), lo memorizza in un array, e lo invia tramite un messaggio al client richiedente.

Successivamente, il client invia il messaggio di richiesta al servente. Prima di servire la richiesta, il servente invia al processo autenticatore un **messaggio di conferma autorizzazione**, contenente il PID e il token del processo client da autorizzare, e il PID del server. Il processo autenticatore cerca la coppia <PID-client, token> all'interno del suo array, e risponde al processo servente con un messaggio di risposta, contenente il PID del server nel campo "tipo", e con un campo "esito" pari ad "1" se la coppia è stata trovata, o "0" altrimenti. Se la coppia è stata trovata, il servente elabora la richiesta del client come indicato sopra. Se la coppia non è stata trovata, il servente invia un messaggio di risposta negativa al client (come sopra, usando il PID del client nel campo "tipo", e ponendo il campo "esito" a 0).

Il processo autenticatore userà una coppia di code sia per i messaggi da/verso i client, sia per i messaggi da/verso il servente. Utilizzare una unica struttura dati per i messaggi sia per i client sia per il servente, riempiendo solo i campi necessari, e usando il campo tipo per distinguere tra i messaggi dei client e quelli del servente.

Per verificare il funzionamento del programma, si creino 2 processi client, di cui uno invia 5 richieste al servente dopo aver ricevuto il token di autenticazione; e un altro invia 5 richieste senza aver fatto l'autenticazione.

