

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

Prova pratica del 19/04/2021
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.

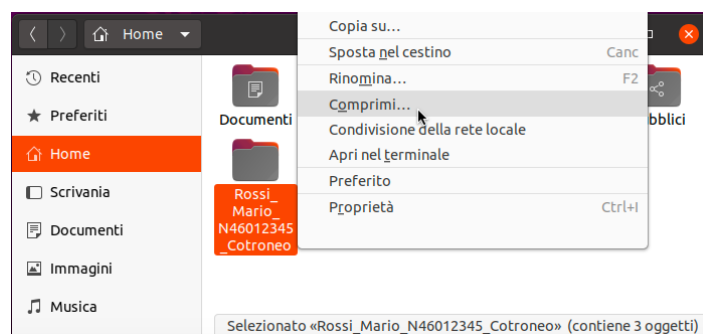
Istruzioni per la consegna dell'elaborato

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

Esempio:

```
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
```

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



All'interno della macchina virtuale, aprire il browser all'indirizzo:

Nel form, cliccare su "**aggiungi file**", selezionare il file compresso contenente il proprio svolgimento, e indicare il proprio nome ed email @studenti.unina.it.

Attendere una notifica del docente, e quindi scollegarsi dalla piattaforma di VirtualClassroom.

Testo della prova

Si realizzi in linguaggio C/C++ un programma client-server **multiprocesso** basato su **code di messaggi UNIX**. Il processo server gestisce una risorsa rappresentata da una variabile di tipo intero. Il server 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 server 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 server. Sia i processi client e sia il server 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 server.

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 server. Prima di servire la richiesta, il server 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 server 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 server elabora la richiesta del client come indicato sopra. Se la coppia non è stata trovata, il server 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 server. Utilizzare una unica struttura dati per i messaggi sia per i client sia per il server, riempiendo solo i campi necessari, e usando il campo tipo per distinguere tra i messaggi dei client e quelli del server.

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

