Relazione del Progetto di Sistemi Operativi

Vannacci Serena 7030223- Ligabue Martin 7008391

Introduzione

Il progetto è un'applicazione software sviluppata con un'architettura modulare e una struttura definita. Si tratta di un sistema client-server progettato per gestire una rubrica elettronica, costituita da record di studenti frequentanti un corso di Laurea, che offre la possibilità di autenticazione per gli utenti possessori delle credenziali.
Il server viene avviato e resta in attesa di connessioni e richieste da parte dei client, può essere interrotto con un SIGINT, eventualmente prodotto da Ctrl+C, che viene catturato, salva la rubrica, chiude la socket ed esce.
Il client si connette al server e propone all'utente tramite riga di comando una serie di opzioni, alcune bloccate dall'autenticazione, altre sempre accessibili. Nel caso sia scelta un'opzione che richiede ulteriori dati, verranno richiesti i dati necessari, come il record da inserire o la sottoistruzione da eseguire, come la ricerca per nome o per matricola.
La funzione di autenticazione è stata impostata per accettare le password: "1234" e "password". Queste sono ovviamente scelte a scopo esemplificativo, ma possono essere cambiate inserendo una password qualsiasi e nel log del server sarà presente la password hashata secondo un algoritmo che al momento prende il primo carattere e lo fa seguire dalla somma dei valori dei caratteri della password sommati. La funzione di hash è facilmente scambiabile con una qualsiasi altra funzione più sicura.
Il modello di sicurezza del progetto presuppone che solo l'eseguibile *client* sia condiviso con gli utenti, in quanto l'accesso al file delle password, o della rubrica, porterebbero ovviamente alla inutilità dei sistemi di autenticazione messi in atto. Particolare cura è stata messa nell'assicurarsi che il client faccia quanti meno passaggi possibile, anche se potrebbe essere ulteriormente semplificato, e non esegua alcuna azione di sicurezza, che è invece gestita dal server.
Il server salva su file nella cartella *log* sia i client autenticati, che le operazioni eseguite, per poter verificare eventuali operazioni sospette. Per comodità ogni operazione è segnalata nell'output di *server* anche lì per avere un maggiore controllo in tempo reale di quello che sta succedendo.

Funzionalità del Progetto

Il progetto offre le seguenti principali funzionalità:

1. **Gestione della Rubrica**
   - **Aggiunta di Contatti**: Gli utenti possono aggiungere nuovi contatti alla rubrica, controllandone la buona sintassi e verificando i dati inseriti.
   - **Modifica di Contatti**: È possibile modificare i dettagli dei contatti esistenti.
   - **Eliminazione di Contatti**: Gli utenti possono rimuovere contatti dalla rubrica, e nel caso il record non sia presente ricevono un errore.
   - **Ricerca di Contatti**: I contatti possono essere visualizzati in un formato leggibile, è possibile per i client cercare tramite nome, cognome o matricola, mentre per il server è presente anche l'opzione di elencare l'intera rubrica a scopo di salvataggio.
2. **Autenticazione degli Utenti**
   - **Verifica delle Credenziali**: Il sistema verifica le credenziali degli utenti utilizzando i dati memorizzati tramite (semplice) hash in `passwords.txt`, nel caso la password risulti

corretta il server salva il pid del client e ne consente l'accesso a istruzioni protette.

3. **Comunicazione Client-Server**
    - **Client**: Il codice nel file `client.c` gestisce le richieste degli utenti e invia i dati al server.
    - **Server**: Il codice nel file `server.c` gestisce le richieste in arrivo dal client e risponde adeguatamente, chiamando le funzioni necessarie a soddisfare la richiesta, ad esempio attivando il processo di autenticazione, controllando che un client sia autenticato, avviando la ricerca dei record in rubrica, etc.

Ambiente di sviluppo

Il progetto è stato creato e testato su Ubuntu, Arch, GitHub Actions, e WSL. Sono stati utilizzati alcuni IDE, fra cui Visual Studio Code, CLion, GitHub Codespaces e in alcuni casi anche semplicemente Nano per modifiche rapide.

Istruzioni

Per compilare il progetto è sufficiente l'esecuzione del comando *make*, che richiama i metodi *client* e *server*, oltre che un comando per creare alcune cartelle di servizio.

Per eseguire un reset completo del progetto è stato aggiunto il comando *make clean*.

Per avviare il server è presente il comando *make run*.

Sia il server che i client possono anche essere eseguiti direttamente chiamandoli da riga di comando.

Per impostare il socket su un indirizzo esterno è possibile impostarlo nel file *common.h* seguendo i commenti.

Struttura del Progetto

Il progetto è organizzato in diverse cartelle e file, ciascuno con un ruolo specifico:

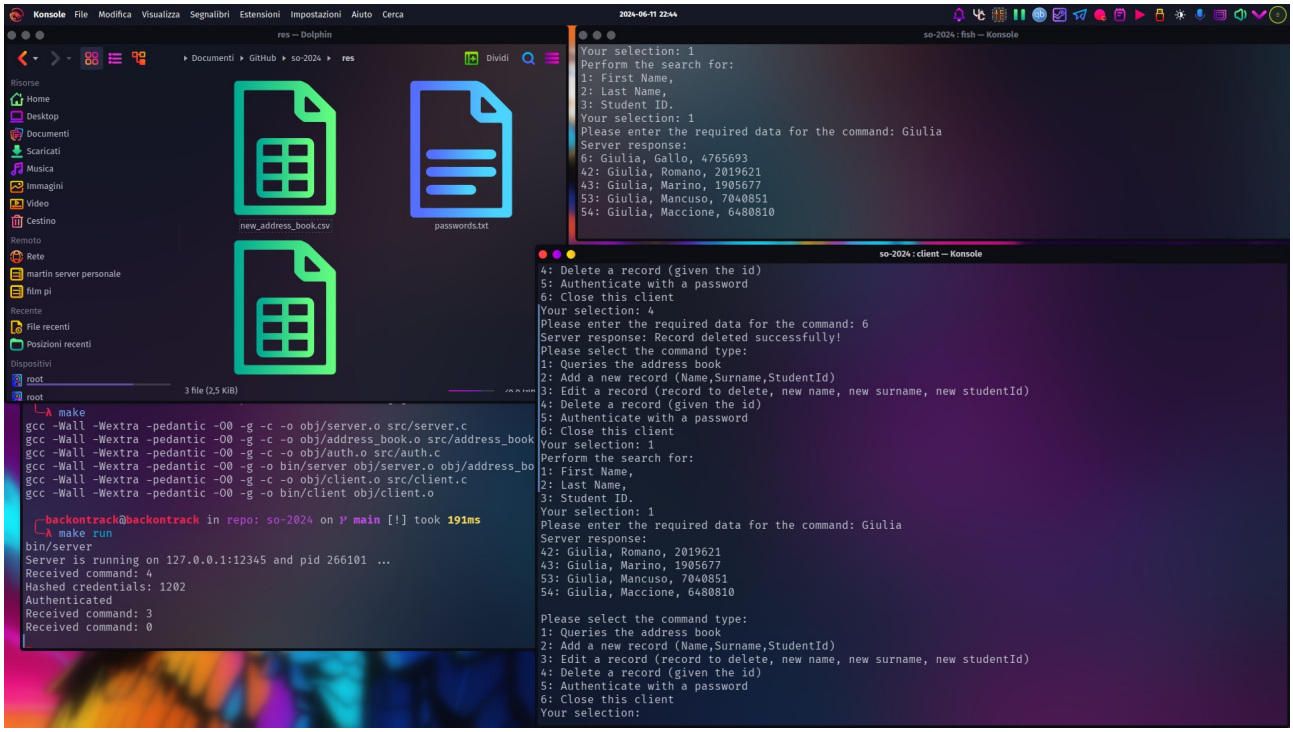
1. **File di Configurazione e Metadati**
   - `Makefile`: File per la compilazione automatizzata del progetto. Contiene tre metodi:
     - all: genera le cartelle che gli servono, compila server e client
     - clean: elimina le cartelle e i file dinamici, riporta il progetto allo stato originale
     - run: esegue il processo del server
2. **Cartella** `inc`
   - `address_book.h`: Header file per le funzioni della rubrica.
   - `auth.h`: Header file per le funzioni di autenticazione.
   - `common.h`: Header file per le funzionalità comuni del progetto.
3. **Cartella** `res`
   - `passwords.txt`: File di risorse contenente password per l'autenticazione.
   - `rubrica.csv`: File CSV contenente i dati della rubrica.
4. **Cartella** `log`
   - `allowedClients_log.txt:` Viene creato quando un client si autentica, e tiene traccia di tutti i client autenticati
   - change_log.txt: Viene creato quando vengono eseguite operazioni privilegiate, e ne tiene traccia.
5. **Cartella** `src`
   - `address_book.c`: Implementazione delle funzioni per la gestione della rubrica.
   - `auth.c`: Implementazione delle funzioni per la gestione dell'autenticazione.
   - `client.c`: Codice relativo alla parte client dell'applicazione.
   - `server.c`: Codice relativo alla parte server dell'applicazione.

Modulazione del Progetto

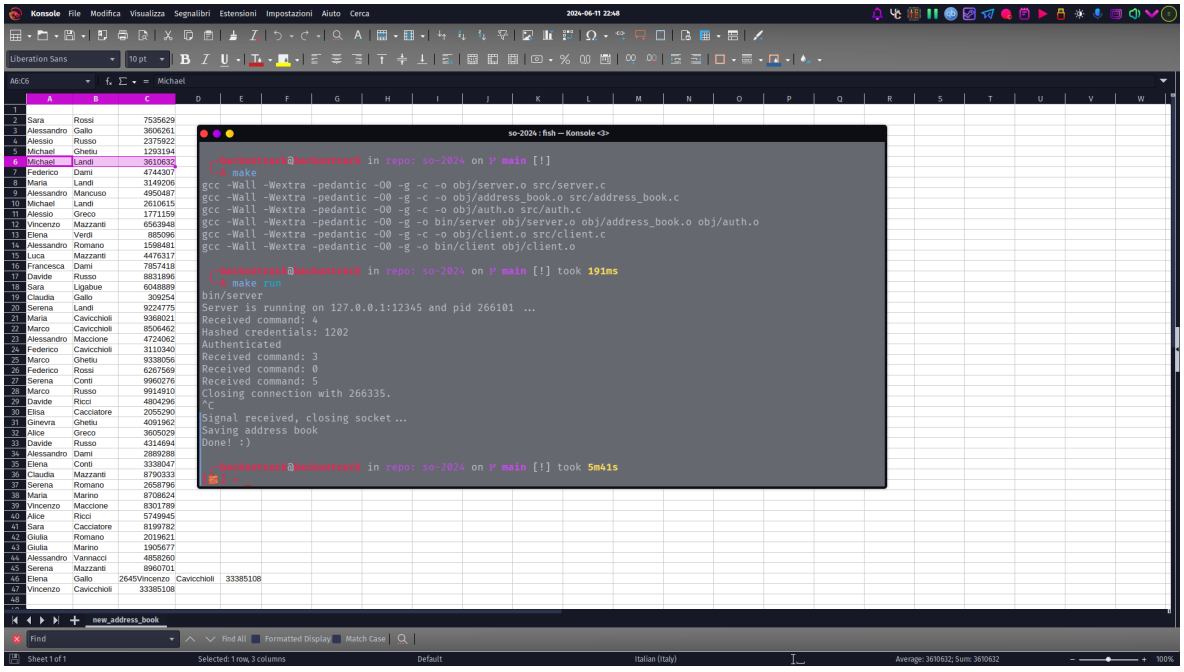
Il progetto è stato organizzato per facilitarne la manutenibilità e la scalabilità:

- **Header Files (`inc`):** Contengono le dichiarazioni delle funzioni e delle strutture dati, fornendo un'interfaccia chiara tra le varie parti del progetto.
- **Source Files (`src`):** Implementano le funzionalità dichiarate negli header files, separando la logica di implementazione dalle dichiarazioni.
- **Resource Files (`res`):** Contengono dati esterni come password e dati della rubrica, mantenendo il codice pulito e facile da gestire.

Esempio di esecuzione

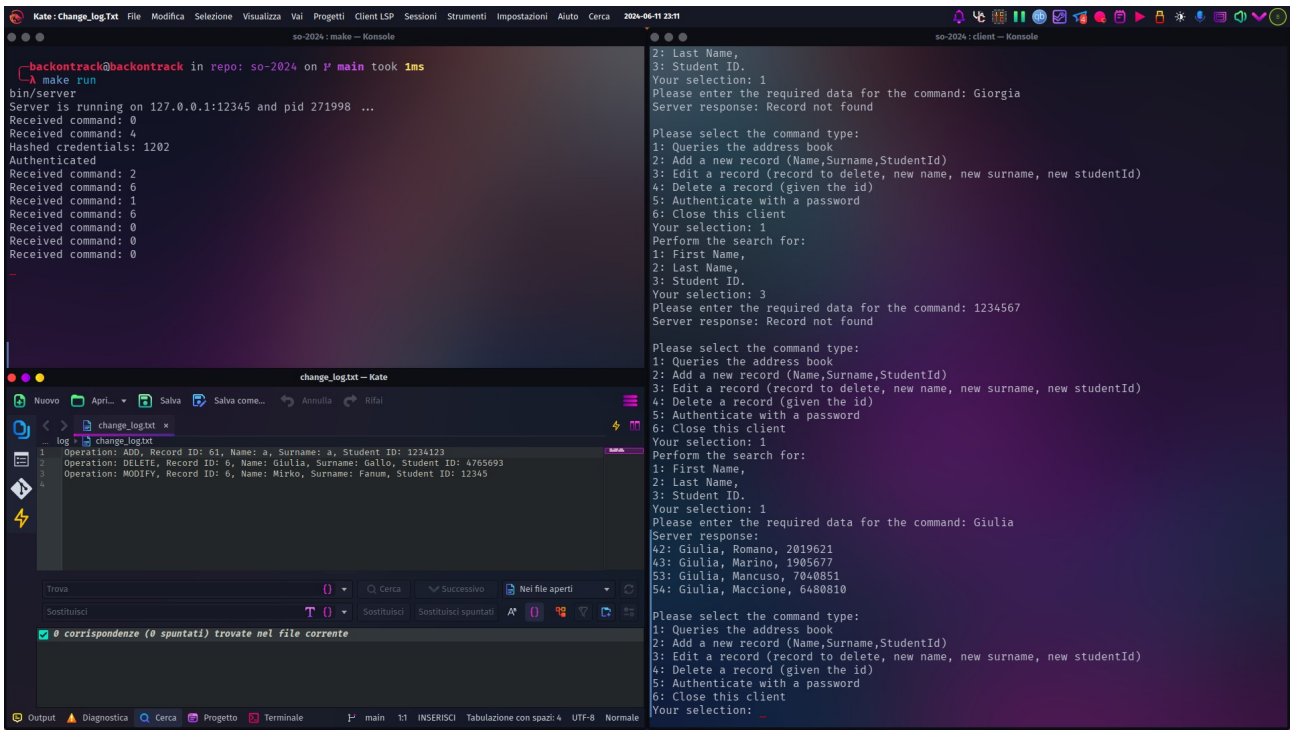


Nella schermata qui di esempio si vede a sinistra che è stato compilato il progetto ed è stato avviato il server, a destra sono stati avviati i client, il nome Giulia è stato cercato nella rubrica, poi si è eliminato il record numero 6, e per confermare si può verificare rifacendo una query del nome Giulia, che restituisce un campo in meno.



A conferma di ciò, chiudendo il server e aprendo la rubrica si nota che in posizione 6 Giulia non è più presente.

Un ulteriore esempio di esecuzione in cui vengono eseguite alcune operazioni di aggiunta, eliminazione e modifica dei record si può vedere nel seguente screenshot:



Conclusione

Il progetto rappresenta un esempio di progetto per un sistema di gestione di una rubrica, con una chiara separazione delle responsabilità e un'organizzazione modulare, che consente di adattarlo facilmente a varie esigenze.