Idee Progetto SO

Struttura rubrica:

* lista di struct (allocata dinamicamente)
* vettore di struct → statico, non si possono aggiungere elementi X
* file di testo → complesso da gestire X

Ordinamento Rubrica:

* Numero di telefono

Campi Della Rubrica:

* Nome
* Cognome
* Numero Telefono
* Indirizzo (via, civico random, città, provincia)

Modalità Autenticazione Client:

* password uguale per tutti (Definizioni Comuni)

Operazioni Richieste da Client:

* Modifica Numero Telefono
* Modifica Indirizzo
* Ricerca Utente Con Coppia Nome Cognome
* Ricerca Utenti Con Cognome X
* Aggiungi Record
* Elimina Record dato N. Telefono
* Visualizza tutti i Record

Modalità Visualizzazione Record per Client

* Telefono\_Cresc 1
* Telefono\_Dec 2
* Cognome\_Cresc 3
* Cognome\_Dec 4

Gestione Conflitti Client (*la coda delle richieste è gestita automaticamente da accept()* ):

* 1 richiesta Client per volta viene soddisfatta dal server
* Le richieste vengono soddisfatte dal server nell'ordine di arrivo
* Quindi si organizzano le richieste in una coda

Divisione Compiti

Bolle

* Generazione Rubrica (fatto)
* Scheletro Client (es. Operazioni Richieste dal Client al Server)

Nicco

* Scheletro Server (creazione socket e bind, ascolto, accept, login utente e fork) ✓

(ho lasciato commentato tutta la parte di ChatGPT che ancora non ho fatto/controllato e che deve ancora essere implementato)

Da fare:

* Gestire il login utente ricevendo un segnale particolare (in parallelo col lato client)
* Gestire l'invio di messaggi tra server e client tramite segnali
* Gestire la richiesta in base al tipo
* Gestione errori nelle varie fasi

DOPO:

* refactoring spostando in altri file alcune procedure (utilities) usate dal server
* controllare se la configurazione della socket è corretta (specialmente la porta)

Idee:

* Il client non ha accesso alla rubrica, solo il server può accedervi

PROTOCOLLO DI COMUNICAZIONE CLIENT-SERVER

Inizio comunicazione = richiesta del client:

* Invio di 4 byte (un int) per passare il tipo di richiesta
* Se la richiesta è di scrittura si deve inviare la password che è una stringa (autenticazione)

Gestione richiesta da parte del server:

* Invio risultati (solo per lettura)
* Invio messaggio di fine (?)

Relazione Client-Server

1. Server mandato in exe, rimane in ascolto dei client e quindi è sospeso
2. Client effettuano la richiesta inviando 4 byte (un int), ad identificare il tipo di richiesta e il client si sospende
3. Il Server legge il tipo della richiesta e se di scrittura si sospende di nuovo attendendo l’invio della password
4. Il Client se necessario si riattiva, inviando la password per poi sospendersi nuovamente
5. Il Server controlla la password inviata, e se corretta effettua l’operazione richiesta. Altrimenti genera un errore.

Gestione Richiesta di Lettura:

1. Dopo che il server ha letto la richiesta di lettura, si sospende e attende i parametri necessari a soddisfarla
2. Il Client invoca il metodo corrispondente alla richiesta di Lettura e richiede all'utente un dato per la Lettura (controlla che il dato sia plausibile). Il parametro ottenuto viene passato al server. Il Client si sospende in attesa che il Server esegua la richiesta
3. Il Server invoca il metodo che, attraverso il riferimento al database, ritrova i record/le info richieste. Invia i record richiesti se l'operazione è andata a buon fine, altrimenti un segnale di errore al client

Gestione Richiesta di Scrittura (supponendo una previa autenticazione):

1. Il server si sospende in attesa dei parametri per la modifica della rubrica
2. Il client invoca un metodo per la richiesta all'utente di questi parametri e, se corretti, li passa al server. Altrimenti solleva un errore
3. Il server riceve i parametri per la modifica della rubrica e, attraverso il riferimento al file, invoca un metodo per effettuare il cambiamento. Alla fine il server invierà un messaggio di avvenuta modifica o meno, a seconda del successo dell'operazione.