

Progetto di Programmazione di Reti

Traccia 1: Chat Client-Server

Matteo Giorgini
Matricola: 0001136576

11 giugno 2024

Indice

| | | |
|----------|----------------------|----------|
| 1 | Consegna | 2 |
| 2 | Server | 3 |
| 3 | Client | 4 |
| 4 | Funzionamento | 5 |

Capitolo 1

Consegna

Implementare un sistema di chat client-server in Python utilizzando socket programming. Il server deve essere in grado di gestire più client contemporaneamente e deve consentire agli utenti di inviare e ricevere messaggi in una chatroom condivisa. Il client deve consentire agli utenti di connettersi al server, inviare messaggi alla chatroom e ricevere messaggi dagli altri utenti.

Capitolo 2

Server

Per realizzare il server ho dovuto importare dalla libreria socket la funzione `socket` per creare un socket e le costanti `AF_INET`, per indicare che il protocollo usato è di tipo IPv4, e `SOCK_STREAM`, in modo che il socket sia orientato alla connessione secondo il protocollo TCP. Le funzionalità richieste sono svolte da thread diversi, utilizzando la libreria importata `Threading`, i quali utilizzano 2 funzioni:

- `incoming_connections`: viene attivata all'avvio dell'applicazione e svolge il compito di ascoltare e accettare le richieste di connessione al server da parte di un client attraverso un flag `SYN`.
- `manage_client`: gestisce un singolo client e per ogni messaggio ricevuto da un client lo invia a tutti gli altri utilizzando la funzione `broadcast`, la quale manda con il nome del client mittente il messaggio a tutti i client memorizzati. Se riceve come messaggio quello di terminazione della connessione ("quit"), termina il client e notifica tutti gli altri che si è disconnesso.

Capitolo 3

Client

Per realizzare il client ho utilizzato oltre alle librerie utilizzate per il server anche la libreria `tkinter` per realizzare una piccola interfaccia grafica dove inviare e ricevere i messaggi della chat. Quando un client si vuole connettere viene creato un socket per la connessione al server con la funzione `connect` che manda il messaggio con il flag `SYN`. Dopo aver stabilito la connessione viene creato un thread per ricevere i messaggi attraverso la funzione `recv`. I messaggi vengono ricevuti tramite la funzione `receive_messages` e vengono mostrati nell'interfaccia grafica insieme a quelli inviati. Per inviare messaggi viene utilizzata la funzione `send_messages` che invia il messaggio tramite la socket creata sul server. Se si invia il messaggio per terminare la connessione viene chiuso il socket e l'interfaccia grafica e viene inviato un messaggio di disconnessione.

Capitolo 4

Funzionamento

Per utilizzare gli script è necessario installare Python se non è presente sulla macchina. Come prima cosa è necessario eseguire lo script `server.py` che apre la porta sul server e attende le richieste di connessione. Successivamente si può eseguire lo script `client.py` su più terminali in modo da simulare più utenti connessi alla chat. All'avvio del client verrà richiesto di inserire l'indirizzo del server (in questo caso è stato utilizzato `127.0.0.1`) e la porta (`42069`), se queste informazioni risultano errate viene chiesto di reinserirle. Dopo aver effettuato il collegamento al server si avvia l'interfaccia grafica e come prima cosa bisogna digitare e inviare il proprio nome utente, dopodichè sarà possibile partecipare alla chat inviando e ricevendo messaggi, ed eventualmente uscire inviando il rispettivo comando `quit`;