Python Chat

Andrea Baldazzi : 0001071149

May 20, 2024

Indice

1 Introduzione

Questo progetto fa riferimento alla traccia 1 proposta, ovvero l'implemetazione di un sistema di chat Client-Server realizzato in Python. Il sistema é in grado di poter accettare piú connessioni in contemporanea da parte di svariati client.

Per la realizzazione del progetto si é scelto di utilizzare il protocollo TCP, per garantire la corretta ricezione di messaggi scambiati tra il server e i client.

2 Client

Il client é la parte del programma utilizzata dall'utente che vuole far parte della chatroom e comunicare con gli altri nella stanza. Fa uso di un approccio orientato alla connessione, in quanto viene creato un socket python di tipo SOCK_STREAM, utilizzando cosí il protocollo TCP.

Inizialmente il client chiede all'utente di inserire i dati necessari per la connessione, ovvero l'indirizzo ip e la porta del server oltre ad un username con il quale sará identificato all'interno della chat. In caso un valore venisse omesso si considerano i valori di default: 127.0.0.1 per l'ip, 25565 per la porta e "User" seguito da un numero casuale come nome utente.

In caso la connessione non vada a buon fine non succede nulla, lasciando all'utente la possibilitá di modificare i parametri inseriti.

Una volta stabilita la connessione viene creato un nuovo thread per la gestione della ricezione dei messaggi. Infatti, mentre il main resta in ascolto di possibili messaggi da inviare al server, il nuovo thread é continuamente in attesa di ricevere messaggi dal server in modo da poterli mostrare all'utente.

Questo approccio consente di mantenere l'interfaccia fluida scorporando invio e ricezione in due flussi distinti che eseguono in contemporanea.

Sia la ricezione, che l'invio sono rispettivamente decodificando o codificando i messaggio da e verso il server con la codifica UTF-8.

Inoltre il client si occupa anche della gestione della terminazione della chat:

- Se la finestra della chat viene chiusa o si digita il comando di uscita comunica al server l'uscita.
- Se il server si interrompe improvvisamente senza avvisare arresta l'applicazione.
- Se il server comunica la chiusura della chat si arresta l'applicazione

3 Server

Il server della chat permette ai vari utenti di collegarsi per comunicare tra di loro. Come avviene per i client utilizza, tramite socket Python, il protocollo TCP per lo scambio di messaggi connection-oriented.

Appena avviato il programma il server crea un socket di tipo SOCK_STREAM e si mette in ascolto all'indirizzo 127.0.0.1 (localhost) alla porta 25565. Un thread principale di occupa di stare in ascolto di possibili client che vogliono connettersi.

Per ogni client che si connette con successo viene creato un nuovo thread, facendo il "forking" di quello principale; sará poi lui ad occuparsi della comunicazione con il client.

Questo approccio permette di non sovraccaricare il thread principale e di mantenere cosí l'applicazione fluida.

Per la comunicazione con i client il server mantiene in memoria due dizionari, associando ai socket comunicanti con i client relativi indirizzi ip e username. Appena si riceve un messaggio da un client, quest'ultimo viene subito trasmesso in broadcast a tutti gli altri, tuttavia non viene memorizzato per cui i client che si sono collegati in un secondo momento non potranno vedere lo storico della chat.

Infine, come fanno i client, il server gestisce le chiusure e i relativi errori:

- Se la console del server viene chiusa, comunica la chiusura a tutti i client connessi
- Se un client si interrompe improvvisamente viene disconnesso dal server
- Se un client comunica la chiusura viene disconnesso dal server

4 Utilizzo

L'apertura dei due script 'client.py' e 'server.py' puó essere effettuata tramite riga di comando, previa installazione di Python sulla propria macchina. Se si dispone di una macchina con Windows, é anche possibile eseguire i due script cmd 'launch_client.bat' e 'launch_server.bat' per eseguire l'applicazione senza aprire un terminale, lasciando quindi la sola GUI.

Client

Quando si apre il client viene chiesto all'utente di inserire l'indirizzo ip e la porta del server al quale si vuole connettere, oltre all'username con il quale vuole essere riconosciuto in chat. Se uno di questi parametri viene omesso, vengono utilizzati i valori di default (vedi Capitolo 2).

Una volta entrato in chat, viene presentata un'interfaccia minimale in cui é visualizzata la chat history oltre a un campo dove poter inserire la frase da inviare. Ogni messaggio in chat sará preceduto dal nome dell'utente che l'ha scritta.

L'arresto dell'applicazione puó essere effettuata chiudendo la finestra o digitando {quit} in chat.

Server

Quando viene avviato il server, viene presentata una console GUI minimale sulla quale vengono mostrati tutti gli eventi che si succedono all'interno della chat.

L'arresto del server avviene tramite la chiusura della finestra.

Considerazioni finali

Il corretto funzionamento del progetto é stato testato su Windows, utilizzando un interprete Python 3.11.8, sia da riga di comando sia utilizzando i file '.bat' provvisti. Si é verificata l'avvenuta connessione, la mancata connessione, la comunicazione tra il server e i vari client collegati e la disconessione forzata da parte di entrambi.