

Corso di Programmazione di Reti
ELABORATO

Traccia 1: Sistema di Chat Client-Server
“Chat in Compagnia”

Zaccaroni Cristina

Matricola: 990028

cristina.zaccaroni@studio.unibo.it

Indice

1 Obiettivo del progetto	3
2 Server	4
3 Client	5
4 Impostazioni e funzionamento	6

1. Obiettivo del progetto

Implementare un sistema di chat client-server in Python, dove il server è in grado di gestire più client contemporaneamente e consente agli utenti di inviare e ricevere messaggi in una chatroom condivisa.

Il client invece deve consentire agli utenti di connettersi al server, di inviare messaggi alla chatroom e ricevere messaggi dagli altri utenti.

La connessione del Client al Server e l'ingresso nella chat devono avvenire tramite un'interfaccia grafica realizzata con la libreria tkinter.

2. Server

Per il Server sono stati utilizzati due dizionari: clients e indirizzi per registrare nome associato e indirizzo dei client in ingresso.

Per la connessione è stato inoltre necessario importare alcuni elementi della libreria socket e la classe Thread dalla libreria threading per attivare e gestire un thread per ogni client.

Per la connessione è stata utilizzato un socket TCP e quindi importando dalla libreria socket la costante SOCK_STREAM, oltre alla costante AF_INET per indicare il protocollo usato per il formato degli indirizzi IP (in questo caso IPv4).

Le funzioni necessarie al Server sono:

- **accetta_conessioni_in_entrata**

Viene attivata all'avvio dell'applicazione Server.

Il Server si mette in ascolto sul socket e appena arriva una richiesta di connessione da un client, se la connessione va a buon fine, aggiorna il dizionario indirizzi aggiungendo host e porta del nuovo client.

- **gestisce_client**

Si occupa di gestire un singolo client che entra nella Chat.

Il Server riceve il messaggio inviato dal Client appena avviene la connessione e contenente il nome, salva il nome nel dizionario clients e gli restituisce un messaggio di benvenuto personalizzato col nome del client, accompagnato dall'indicazione di come uscire dalla chat.

Poi il Server si mette in ascolto del thread del singolo client e ne gestisce l'invio dei messaggi o l'uscita dalla Chat.

Ogni messaggio ricevuto viene inviato a tutti i partecipanti alla chat preceduto dal nome del client che ha inviato il messaggio (funzione broadcast).

Se il Server riceve il messaggio di uscita {quit} dal client, chiude il client, elimina il client dal dizionario e se vi sono altri client ancora in chat invia a tutti un messaggio comunicando che quel client ha abbandonato la chat. Nel caso in cui il client in uscita sia l'ultimo rimasto in chat non viene inviato alcun messaggio.

- **broadcast**

La funzione invia un messaggio in broadcast a tutti i client.

3. Client

Per il Client oltre alle librerie usate per il Server è stata utilizzata anche la libreria tkinter, con la quale è stata realizzata una piccola interfaccia grafica formata da due finestre, la prima per collegarsi al Server e inviare il nome e la seconda, che si apre subito dopo la connessione, per inviare e ricevere i messaggi della chat oppure uscire.

Il Client è stato creato come Classe, perché mi è risultato più semplice e schematico soprattutto per la creazione della parte grafica.

Le funzioni principali del Client (escluse quelle della parte grafica) sono:

- **request_connection**

Crea e inizializza il socket del Client per la connessione e cerca di stabilire la connessione utilizzando l'host e la porta inseriti dall'utente nella finestra di login.

Questa funzione viene richiamata all'interno della funzione `connetti_accedi`.

- **connetti_accedi**

Verifica che siano stati inseriti tutti i dati necessari per la connessione (host e porta) e se la connessione va a buon fine verifica che sia stato inserito il nome del Client, invia tramite socket al Server un messaggio col nome e viene creato il Thread per ricevere i messaggi attraverso la funzione `receive`.

Chiude in automatico la finestra di login e apre quella della chat.

- **send**

La funzione invia i messaggi al Server tramite il socket. I messaggi vengono inviati solo se non sono vuoti.

Se il messaggio inviato è {quit} (cioè è stato premuto il pulsante "Esci") viene chiuso il socket e anche la finestra chat.

Ogni volta che viene inviato un messaggio viene ripulita la casella di inserimento.

- **receive**

La funzione riceve i messaggi utilizzando il Thread creato dalla funzione `connetti_accedi` e la funzione `recv` del socket.

- **close**

E' la funzione che chiude il socket e l'interfaccia grafica nel caso in cui la finestra chat venga chiusa cliccando la X.

Le altre funzioni presenti nel codice servono per la creazione e gestione della GUI e sono debitamente commentate.

3. Impostazioni e funzionamento

Per il funzionamento dei due script è necessario avere installato Python.

Il Server è impostato per accettare 4 connessioni.

In fase di connessione, se non viene indicata una porta specifica, viene utilizzata la porta 53000, mentre è obbligatorio inserire il Server host o localhost.

Per iniziare occorre eseguire lo script Server.py che apre la porta sul Server e attende le connessioni dei Client.

Poi è necessario eseguire lo script Client.py , uno per ogni Client e su terminali diversi per simulare i diversi utenti connessi alla stessa chatroom.

Appena lanciato questo script si apre la finestra di login, dove l'utente deve inserire l'host, la porta (non obbligatoria) e il nome. In caso di mancata compilazione di campi obbligatori si aprono messaggi di errore.

Una volta inseriti i dati richiesti, per connettersi ed entrare nella Chatroom si clicca il pulsante "Accedi" oppure si può premere il tasto "return".

A questo punto se la connessione va a buon fine si chiude la finestra del login e si apre la finestra della Chat che riporta in alto il nome del Client connesso e nella parte che contiene i messaggi riporta invece il messaggio di benvenuto inviato dal Server subito dopo la connessione.

Nella parte bassa della finestra della Chat c'è l'indicazione per inviare i messaggi, la casella di inserimento messaggi e a lato ci sono il pulsante "Invia" ed "Esci".

Un messaggio può essere inviato sia col pulsante "Invia" che premendo "return" da tastiera.

Si esce dalla chat premendo "Esci" oppure cliccando la X della finestra.