

Traccia 1:

Sistema di Chat Client-Server

Progetto di Reti di Marco Zignani 0001031354

Server

Per realizzare il server ho importato la libreria socket, la quale contiene le costanti AF_INET per indicare l'utilizzo del protocollo IPv4 e della costante SOCK_STREAM che definisce il protocollo di connessione TCP. Ho importato anche la libreria threading in quanto il server si compone di 2 thread che eseguono 2 funzioni:

- *connessionInEntrata* che effettua la funzione di ascolto e istanzia il thread per la gestione del client all'arrivo di una richiesta di connessione gestita con un flag SYN.
- *gestisceClient* usa la funzione broadcast per inviare a tutti i client attualmente connessi i messaggi provenienti da un singolo client. Se viene ricevuto il messaggio di terminazione della connessione la funzione chiude il client e notifica a tutti gli altri client che il client si è disconnesso.

Il thread main istanzia il nuovo thread che grazie alla funzione join accetta le connessioni e aspetta la terminazione.

Client

Ho utilizzato la libreria tkinter per avere l'interfaccia grafica su cui inviare messaggi grazie alla funzione invio e poter leggere i messaggi in arrivo.

Ogni client ha un socket per connettersi al server tramite la funzione connect che invia un messaggio con flag SYN.

Siccome la funzione recv è bloccante viene creato un nuovo thread per la ricezione dei messaggi

Utilizzo del progetto

Utilizzare un interprete python per aprire i 2 file. Eseguire prima il file server.py così da metterlo in ascolto e successivamente eseguire uno o più istanze del file client.py su console diverse.

Inserire in ogni client l'indirizzo IP 127.0.0.1 e porta 49000.

Scrivere il nome del client nelle finestre create e inviare uno o più messaggi dai vari client.

Per chiudere la connessione digitare {quit} compreso di { }.