Progetto Reti di Elaboratori

DUBKOV ANDREI

May 2022

1 Funzioni sviluppate

Ci sono 2 funzioni principali: **main** e **write_to_socket**. La prima è il punto di entrata del programma, seconda permette di inviare i dati via socket. All'inizio il nostro client legge subbito tutti i messaggi(che vengono trovati nel suo file corrispondente client$i$_in.txt) e mette dentro l'array *txt_messages* dopodiché si connette al server, in caso se non riesce viene stampato il messaggio su stdout,dopo viene aperto un nuovo file client$i$_out.txt, dove il client salverà tutti i messaggi. In seguito viene subito controllata la condizione "EXIT" per interrompere la connessione e uscire dal ciclo while. Poi il client aspetta il messaggio che viene trasmesso dal server, potrebbero essere mandati due tipi di messaggi: uno che ha lunghezza di stringa uguale a 1(ovviamente questo per il primo messaggio inviato dal server per inziare la comunicazione) e un'altro con lunghezza maggiore di 1(messaggi dopo il primo). Client inizia la comunicazione(legge dal suo file di input e manda il messaggio in rete) in caso se il messaggio trasmesso contiene il suo id oppure se il messaggio trasmesso ha lunghezza maggiore di 1, si prende l'ultimo carattere di stringa trasmessa e viene controllato se corrisponde o no all'ID precedente a quello del client. Se è cosi dovrebbe leggere dal file di input(la funzione **write_to_socket**), altrimenti scrivere sul suo file out.

2 Problemi incontrati

Durante l'allocazione delle stringhe dentro array l'ultimo carattere di tutte le stringhe tranne l'ultima è linefeed, questo si è stato risolto con un semplice controllo se la linea non è prima nel file(per evitare index out of boud), allora dovremmo sostituire l'ultimo carattere della linea precedente(cosi evitiamo di fare la stessa operazione nell'ultima riga) con il NULL. Lo stesso problema abbiamo anche quando riceviamo i messaggi inviati dagli altri client, però sostituiamo il carattere prima del penultimo. Poi è stata implementata la funzionalità per controllare la lunghezza del messaggio per fare in modo tale che il primo messaggio potrebbe contenere anche un id diverso da 1. Ovviamente in macros abbiamo definito variabili **ID_CLIENT** e **ID_PREVIOUS** per controllare se viene passato l'id precedente e dobbiamo leggere dal file input.