

# Progetto di reti

## Traccia 2

Daniel Capannini, matricola: 0000971194, mail: daniel.capannini@studio.unibo.it

la traccia 2 chiedeva di costruire un client-server UDP con vari comandi:

LIST: questo comando richiesto al server, fornisce la lista di file che il server contiene e quindi possono essere scaricati. I file vengono salvati in `file_name` che viene aggiornato ad ogni passaggio dentro il while con il comando `file_name = os.listdir('./file_server/')`, la `'./file_server/'` è la directory del server.

GET: questo comando seguito dal nome di un file permette di ricevere una copia del file se questo file è presente nella directory del server e non è presente nella directory del client, anche se il file è diverso o aggiornato ed è presente della directory del client non è possibile scaricarlo, i file devono avere obbligatoriamente nomi diversi. I file di dimensioni maggiori di `BUFFER_SIZE=4096` vengono inviati suddivisi in più pacchetti e ricomposti dal client, grazie a questo metodo è possibile il download di file di qualsiasi dimensione. Quando si è trasmesso tutto il file viene inviato un messaggio di avviso di fine `'fine_fine'` per avvisare il client della fine della trasmissione del file.

PUT: questo comando seguito dal nome di un file se questo è presente sul client e non sul server permette di farne l'upload, come per GET i file devono avere nomi diversi e non semplicemente contenuto diverso. Il funzionamento e l'implementazione è uguale a quella della GET con parti invertite. Il server prima che la transizione abbia inizio invia un messaggio di conferma `'GET'`.

EXIT: comando aggiunto per permettere la chiusura del socket e dell'eseguibile del client, non tocca il server il quale rimane in attesa di comandi.

Anche il client si tiene traccia dei file che ha nella propria directory ad ogni ciclo del while tramite `file_name = os.listdir('./file_client/')`.

Sia nel server che nel client all'interno dei cicli while è presente un try per intercettare gli errori, i quali forniscono in messaggio informativo su esse e poi si ritorna all'inizio del ciclo while.