

# Relazione Progetto Reti

## 1 Introduzione

Il progetto consiste di due programmi, `server.c` e `client.c`, in grado di comunicare tra loro tramite socket. I programmi si scambiano messaggi seguendo un protocollo prestabilito, trasparente all'utente ed in grado di interoperare con programmi che implementano il medesimo protocollo.

## 2 Funzionalità

### 2.1 Client

Dopo aver eseguito la connessione al server, il programma mostra un menù testuale dal quale si possono effettuare le seguenti operazioni: inviare dati al server, calcolare media e varianza, mostrare istruzioni di utilizzo e disconnessione.

I dati inseriti dall'utente vengono parsificati e formattati per aderire ai requisiti protocollari prima di essere inviati.

### 2.2 Server

Dopo aver inizializzato socket e porta, il programma accetta eventuali connessioni da parte dei client che si vogliono connettere, ed attende la ricezione di messaggi opportunamente formati: qualora vengano ricevuti messaggi contenenti errori sintattici (che violano il protocollo), il programma informa il client in maniera opportuna e chiude la connessione. Se i messaggi ricevuti sono sintatticamente corretti, procede con la gestione delle richieste.

## 3 Utilizzo

### 3.1 Client

#### 3.1.1 Avviare il programma

Il programma prende come argomenti l'indirizzo IPv4 del server ed il numero di porta su cui il server ascolta. Se gli argomenti non sono presenti, il programma mostra un messaggio con istruzioni di utilizzo. Se gli argomenti inseriti sono corretti, e la connessione avviene correttamente, il programma mostra un menù contenente varie opzioni.

#### 3.1.2 Opzione 1: Inviare dati al server

Selezionando l'opzione 1 (**1. Send data to server**), il programma sollecita l'utente ad inserire i campioni (numeri interi), uno per riga, terminati dal carattere 'q': l'inserimento di caratteri non numerici è impedito ed opportunamente segnalato all'utente.

#### 3.1.3 Opzione 2: Calcolare media e varianza

Selezionando l'opzione 2 (**1. Calculate mean and variance**), il programma invia al server il messaggio di fine dati ed attende una risposta dal server. Se l'operazione avviene con successo, vengono mostrati i risultati, altrimenti viene mostrato un messaggio d'errore. In entrambi i casi, la connessione viene quindi chiusa.

### 3.1.4 Opzione 3: Mostrare istruzioni di utilizzo

Selezionando l'opzione 3 (`1. Show usage instructions`), il programma mostra all'utente come utilizzare le varie funzioni messe a disposizione.

### 3.1.5 Opzione 4: Disconnessione

Selezionando l'opzione 4 (`1. Close connection`), il programma si disconnette dal server e termina l'esecuzione.

È consigliato terminare il programma selezionando quest'opzione.

## 3.2 Server

### 3.2.1 Avviare il programma

Il programma prende come argomento il numero di porta sul quale si deve mettere in ascolto. Se gli argomenti non sono presenti, il programma mostra un messaggio con istruzioni di utilizzo.

Se gli argomenti inseriti sono corretti, e la connessione avviene correttamente, il programma si mette in ascolto per eventuali connessioni.

### 3.3 Terminare il programma

Il programma può essere terminato in ogni momento inviando il segnale `SIGINT` (Ctrl-C).

## 4 Scelte di programmazione

Entrambi i programmi sono stati sviluppati su piattaforma Linux (Arch Linux), compilati con `gcc` e debuggati/testati con `GDB` e `Valgrind`. Il programma `server.c` utilizza la libreria `math.h` per calcolare i quadrati, e richiede linking a tempo di compilazione (tramite flag `-lm`).

I messaggi mostrati dai programmi sono evidenziati con colori diversi: il colore blu indica messaggi informativi sullo stato dei processi, il colore rosso indica messaggi d'errore ed il colore verde indica operazioni svolte con successo.

Entrambi i programmi controllano la correttezza sintattica dei messaggi per fornire maggiore interoperabilità.