

Relazione progetto Reti 1

Strumenti di sviluppo:

- VM WSL Ubuntu 20.04 su OS Windows 10.
- GCC Version: gcc (Ubuntu 9.3.0-10ubuntu2) 9.3.0
- Editor: Notepad ++

Comandi di compilazione:

SERVER: gcc server-project.c -o server-project -lm

CLIENT: gcc client-project.c -o client-project

Comandi di esecuzione (localhost)

SERVER: ./server-project 11000

CLIENT: ./client-project 127.0.0.1 11000

Workflow Client

L'avvio del client richiede i argomenti IP e PORT, se tali argomenti non vengono forniti verrà mostrato un errore a video

Gli argomenti vengono forniti alla funzione connect() che mi permette di aprire una connessione al server. La funzione restituisce un intero qual ora fosse inferiore a 0 significa che la connessione non ha avuto buon esito ed avviso l'utente.

Avvenuta la connessione richiamo la funzione parse_response fornendo il buffer di risposta della read sul socket e struct di output. La funzione effettua un controllo di esistenza del carattere '\n' requisito che il messaggio di testo deve avere.

Attraverso la funzione strtok identifico i token separati dal carattere spazio.

Essendo a conoscenza della sequenza dei possibili token, vado ad estrarre tutte le informazioni valorizzando la struct, ottenendo in questo modo i seguenti valori:

outcome: identifica lo stato della risposta OK oppure ERR

typeRes: identifica il tipo di risposta ES. START per il messaggio iniziale, STATS per le statistiche ottenute.

message: che conterrà l'eventuale messaggio di risposta (nel caso di typeRes == START)

Seguono poi i valori di ritorno delle statistiche nElementElab, avg, var;

Se la sequenza attesa non rispetta quella che il messaggio dovrebbe avere, la funzione restituirà ERROR e verrà stampato un messaggio di avviso.

Dopo che il programma ha stampato il messaggio di benvenuto entrerà in un ciclo do while chiedendo quanti valori vorrà spedire al server nel primo messaggio.

In un primo controllo il programma effettuerà la verifica che sia diverso da 0, in caso contrario l'utente non intende inserire altri valori e di conseguenza attenderà la risposta dal server contenente le statistiche dei valori contenuti in tutti i messaggi inviati.

La funzione checkint controllerà che tale sia valido. Se non lo fosse tale funzione restituirà -1 a cui seguirà una stampa a video di errore.

Verranno poi richiesti sequenzialmente i valori, fino all'invio del messaggio al server contenente il numero dei valori inseriti e la corrispondente lista.

Il programma attenderà un messaggio di conferma da parte del server contenente il numero dei valori effettivamente letti e confronterà questo valore con il numero dei valori inviati. Se il controllo restituirà un esito negativo verrà stampato un messaggio a video.

Ad ogni nuovo utilizzo del buffer verrà preventivamente richiamata la funzione ClearBuffer così da avere la garanzia la lettura corrente non sarà inquinata dalla precedente. Inoltre verrà effettuato un richiamo alla Delf che mi cancellerà il ritorno a capo in modo tale che all'utilizzo dell'eventuale messaggio da stampare in output non ottenga due ritorni a capo.

Workflow Server

L'avvio del server richiede come parametro il numero di porta su cui resterà in ascolto.

Tale argomento verrà assegnato alla struct indirizzo e mi permetterà di assegnare l'indirizzo locale alla socket. Richiamando la funzione listen, il server si mette in ascolto pronto a ricevere connessioni.

Il programma entra in un ciclo infinito, e attraverso la funzione accept() attenderò connessioni in ingresso.

All'arrivo della connessione, verrà inviato al client un messaggio di benvenuto, ed entrerà in un ciclo do while in attesa di un messaggio nel seguente formato: <Numero_dati> <dato1> <dato2> <datoN>, fino a che il numero degli elementi inviati non sarà pari a 0. Tale valore scatenerà il calcolo della media e della varianza.

Ricevuto il messaggio, attraverso la funzione strtok ciclo tutti i token che il messaggio contiene, considerando che il primo valore indicherà il numero degli elementi da elaborare e i successivi invece sono i valori della mia popolazione.

Ogni valore verrà controllato con la funzione CheckInt, e alla fine dei token verrà verificato che il valore degli elementi elaborati all'interno del messaggio sia coerente con il primo valore del mio messaggio.

Eventuali errori catturati nella fase di acquisizione dati scateneranno l'invio di messaggio sul cliente nel formato "ERR DATA Messaggio\n".

Ad ogni nuovo utilizzo dei buffer verrà preventivamente richiamata la funzione ClearBuffer così da avere la garanzia la lettura corrente non sarà inquinata dalla precedente.