Presentazione SO

Progetto: Bacheca Elettronica

Sviluppato da: Daniele De Turris

Organizzazione dei file

All’interno della cartella del progetto sono presenti:

- i file ".c" e ".h" contenenti il codice del client e del server
- un makefile per la compilazione del codice
- una cartella "sqlite" contenente la libreria sqlite (da compilare insieme al server nel caso in cui non sia già installata sul sistema)
- una cartella "Test" contenente un programma per eseguire stress test sul server (una volta avviato, dopo i parametri richiesti, genererà n client che eseguiranno m richieste al server generate in modo randomico)
- un file bacheca_test.db (presente nella cartella "Test")che, se rinominato in bacheca.db, fornisce al server un databse di esempio con più di 15.000 annunci, utile per verificare le prestazioni del server con molti annunci da gestire

Architettura del Server

Il server è stato organizzato secondo una struttura prethreaded, quindi è così composto:

- Main_thread: è il thread principale, incaricato del bootstrap del server (creazione della socket, creazione/preparazione del DB) e, una volta completato, si occupa di mantere in uno stato coerente un pool di thread.
- Thread pool: insieme di thread in attesa di una richiesta da gestire; una volta acquisita procedono con il gestirla, per poi ritornare in attesa.

Ciclo di vita di un thread del pool

Ogni thread esegue ciclicamente queste operazioni:

- controlla che il pool non sia già pieno (10 thread in attesa di una richiesta), quindi si mette in ascolto sulla socket o termina
- una volta ricevuta la richiesta procede a gestirla
- controlla che non abbia già gestito 10 richieste, altrimenti termina

Persistenza

La persistenza dei dati è gestita dal server tramite un database, utilizzando sqlite come DBMS.

Il database è contenuto in un file chiamato "bacheca.db" e viene generato, se non esistente, all’avvio del server.

All’interno del database sono presenti 2 tabelle: "users" contenente le informazioni sugli utenti registrati, e "announcements" contenente le informazioni degli annunci pubblicati.

Il database è acceduto parallelamente da tutti i thread impegnati a gestire richieste.

Architettura del Client

Il client, essendo orientato all'utente finale, non presenta nessun grado di parallelismo, ma inoltra sequenzialemente le richieste fornite dall'utente.

Le operazioni disponibili all'utente sono:

- help
- login/logout
- register
- list/mylist
- post
- delete
- exit