

Relazione del Progetto Chatterbox 2017/2018

Di Gramaglia Gerlando

Matricola 530269

• Struttura del Codice:

il codice del progetto è diviso in più file i quali:

1. Chatty.c:

contiene il main e si occupa di avviare il Thread Pool e il Thread Listener (il codice di questi thread si trova nel medesimo file Chatty.c)

- a. Thread Listener accetta le connessioni avvalendosi dell'uso della select
- b. Thread Pool gestisce le richieste dei client.

2. Lista task.h:

contiene la struttura e le funzioni con cui "comunicano" i thread del Pool e il thread listener.

3. tabClient.h:

Contiene le strutture e le funzioni con cui gestire i clienti registrati e operazioni quali: registrazione e deregistrazione di un cliente, invio di messaggio/i e/o file, invio della history, funzioni di controllo etc. Esso si avvarrà del file icl hash.h.

4. fdclient.h:

Contiene le strutture e le funzioni necessari per la gestione dei clienti online.

5. setdesc.h:

Offre delle funzioni per lavorare in mutua esclusione con la maschera dei descrittori di connessione.

6. Connection.h: protocollo di comunicazione tra client e server

• Strutture dati:

La maggior parte delle strutture definite contengono al loro interno anche le variabili di mutua esclusione e variabili di condizione che sono necessarie per accedere in mutua esclusione ai campi delle strutture stesse. Questa scelta ha lo scopo di rendere più leggibile e di facile comprensione il codice sviluppato.

1. Lista_task.h -> Coda_task

a1. Coda fifo

a2. Variabili di mutua esclusione con cui accediamo alla coda sopra definita.

Coda_task è la struttura con cui "comunicano" i thread del pool e il listener secondo il seguente procedimento: il listener prenderà i descrittori di connessione attivi e li inserirà nella coda, i thread del pool estrarranno un descrittore alla volta e si occuperanno di gestire la richiesta del cliente oppure comunicargli che non può gestirlo chiudendo di conseguenza la connessione.

2. setdesc.h -> set_descriptor: contiene la maschera dei descrittori di connessione e le variabili per accedere in mutua esclusione. Tale struttura verrà utilizzata dal thread

listener e dai thread del Pool per stabilire quando ascoltare un certo descrittore oppure ignorarlo

3. **tabClient.h -> tab_clienti :**

Contiene:

3a. la tabella dei clienti registrati e la sua dimensione. Esso si avvarrà della tabella hash definita in icl hash.h. In tale tabella si andrà ad inserire come dato una struttura contenente le informazioni necessarie per la gestione della history del cliente, mentre per la chiave si userà il nome del cliente.

3b. Array di lock e la sua dimensione. La tabella hash viene infatti “divisa” in più sezioni, ogni lock ci permetterà di accedere in mutua esclusione a una sola sezione della tabella. La dimensione dell’array di lock è uguale al numero di thread del Pool, ciò significa che se ci sono N thread attivi la tabella sarà “divisa” in N sezioni. Il motivo di questa scelta sta nel fatto che il threadPool può lavorare al massimo con N lock diverse “contemporaneamente”.

4. **fdclient.h -> users_online:** struttura contenente la lista degli utenti connessi.

● **GESTIONE SEGNALI:**

la gestione dei segnali viene implementata nel file chatty.c ; il main si occupa di creare una maschera contenente i seguenti segnali che verranno bloccati:

1. SIGINT
2. SIGQUIT
3. SIGTERM
4. SIGUSR1

Tale maschera verrà passata ad un thread signal che farà la sigwait su tale maschera.

Quando arriverà uno di questi segnali verrà catturato e una volta riconosciuto verranno effettuate le seguenti operazioni:

1. SIGQUIT, SIGTERM, SIGINT: verrà settato a 1 un flag che “a cascata” farà terminare il thread listener, il threadPool e verranno deallocate tutte le principali strutture per infine terminare.
2. SIGUSR1: verranno stampate le statistiche

Il segnale SIGPIPE viene ignorato.

● **Concorrenza:**

Come detto sopra le variabili di mutua esclusione sono dichiarate all’interno delle strutture dati stesse. Per completezza elenchiamo le variabili di mutua esclusione e di condizione utilizzate:

1. Una lock per l’accesso alla lista dei clienti online
2. Una lock per l’accesso alla coda dei descrittori di connessione da gestire da parte del ThreadPool e una variabile di condizione per la wait (in caso di coda vuota) e la signal (per segnalare ai thread che la coda non è più vuota)
3. Una lock per la maschera dei descrittori di connessione

4. Un array di lock per la tabella dei clienti registrati
5. Due lock per le operazioni di write e read necessarie quando vengono effettuate operazioni di lettura o invio di un messaggio.

- **Riferimenti:**

icl_hash.h è il file presentato durante il corso, mentre per le operazioni di sendData e readData presenti nel file connection.h si è preso spunto dalle soluzioni degli esercizi presentati a lezione.

- **Macchine per esecuzione del test**

Il Codice è stato testato più volte sempre con successo sulla macchina virtuale fornita dai docenti e su Ubuntu 18.04.01 LTS.