

Gaspare Ferraro matricola 520549 corso B

chatterbox

Progetto per il modulo
di laboratorio SOL 2017

Relazione del progetto

Strutture dati

Iniziamo descrivendo le strutture dati utilizzate nel progetto, dal punto di vista delle funzionalità offerte, per poi analizzarne il loro utilizzo all'interno del progetto.

HashMap, leggera implementazione di una mappa, fornita a lezione, che fornisce operazioni di creazione e distruzione di mappe, inserimento e rimozioni di coppie chiave-valori in base alle chiavi e ricerca di valori per chiave.

Queue, coda implementata mediante array circolare di dimensione finita, decisa alla creazione, che fornisce operazioni mutuamente esclusive di pop e push di interi oltre al costruttore e distruttore della struttura dati in memoria.

LinkedList, lista implementata mediante nodi collegati a catena che fornisce la possibilità di gestire informazioni generiche e di poter decidere la dimensione massima, eventualmente infinita, del numero di elementi. Anche in questo caso vengono fornite le operazioni di pop e push e del costruttore e distruttore della struttura dati in memoria.

Op, le operazioni fornite dal server vengono codificate con il client mediante codici interi predefiniti associati.

Message, un messaggio scambiato tra client e server viene codificato come una coppia header-data, la parte header specifica il richiedente e l'operazione da svolgere, la parte data specifica il destinatario dell'operazione e il buffer di dati associato.

ServerConfiguration, struttura che contiene le configurazioni necessarie del server, lette dal file specificato all'avvio.

Statistics, struttura che contiene le statistiche raccolte dal server riguardo le operazioni effettuate, in termini di messaggi, file, errori e utenti.

User, identifica un utente registrato sul server del quale vengono memorizzati il nickname che lo identifica, la cronologia dei messaggi pendenti ancora da inviare e un descrittore della connessione se l'utente risulta collegato.

Group, un gruppo viene identificato sul server dal suo nome e dalla lista dei membri che sono iscritti a tale gruppo.

Le ultime due strutture, più complesse e basate sulle precedenti, fornisco metodi per gestire le due entità principali del server: utenti e gruppi.

UserManager, gestisce gli utenti offrendo operazioni di registrazione/cancellazione, connessione/disconnessione, invio di messaggi pendenti, elencazione degli utenti collegati oltre che a costruttore e distruttore delle strutture.

GroupManager, gestisce i gruppi offrendo operazioni di creazione/cancellazione di nuovi gruppi, registrazione/deregistrazione di utenti a gruppi e operazioni di ricerca e controllo su utenti e gruppi.

Gestione memoria

Particolare attenzione è stata rivolta alla gestione della memoria dinamica: dopo ogni operazione di allocazione è stata eseguita un'operazione di pulizia, al fine di accidentalmente operazioni su memoria *sporca*, ricordandosi inoltre di de-allocare la memoria richiesta quando questa non è più utilizzata.

Inoltre il server subito prima di terminare effettua operazioni le ultime operazioni di pulizia, usando tra l'altro le operazioni di distruzione delle strutture dati precedentemente descritte.

Accesso ai files

Il server nel corso della sua esecuzione interagisce con diversi files, in particolare all'avvio legge le configurazioni dal file specificato e alla ricezione del segnale SIGUSR1 appende su un file le statistiche raccolte.

Per ogni file scambiato tra gli utenti ne viene salvata una copia nella cartella specificata nella configurazione, omettendo eventuali percorsi nel nome del file ricevuto.

Tutte operazioni sui file vengono svolte usando le operazioni di lettura e scrittura bufferizzate.

Divisione in files

Per migliorare la gestione dei singoli moduli, il progetto è stato suddiviso su più files, ognuno dei quali dichiara e implementa funzionalità vicine tra loro.

In particolare sono stati scritti file per:

- gestione delle configurazioni
- protocollo di comunicazione
- gestione degli utenti
- gestione dei gruppi
- hashmap
- lista linkata
- coda circolare
- operazioni e messaggi

Il file sorgente *chatty.c* si preoccupa di raccogliere tutti i moduli per fornire le funzionalità richieste.

Architettura del server

Iniziamo descrivendo le strutture dati utilizzate nel progetto, dal punto