

Prolećni semestar 2018/2019

TCP chat aplikacija

CS323

C/C++ Programski jezik

Projektna dokumentacija

Student:

Nikola Tasić 3698

Asistent:

Veljko Grković

Sadržaj

[1. Uvod 3](#_Toc491866984)

[2. Analiza 3](#_Toc491866985)

3[. Zaključak](#_Toc491866988) 4

4[. Reference](#_Toc491866989) 4

# Uvod

Projekat predstavlja prost linux sistem multithreaded TCP servera i vise mogucih klijenata na tom serveru koji medjusobno mogu da razmenjuju tekstualne poruke. Pri pokretanju servera otvara se novi socket na izbranom portu i server osluškuje taj port za nove klijente i njihove poruke. Pri pokretanju klijent aplikacije ona se povezuje na server na odabranoj IP adresi i portu i ostvaruje se komunikacija sa svim do tada povezanim klijentima.

# Analiza

* 1. **Cilj programa**

Cilj programa i je izrada aplikacije koja omogućava komunikaciju ljudi preko mreže u vidu tekstualnih poruka. Takođe sekundarni cilj je izrada pravilno napisanog multithreaded servera koji podrzava vise od jednog simlutanog klijenta jer dok sam istraživao po internetu vezano za ovaj projekat nisam našao ni jedan adekvatno napisan server iz kog bih mogao da izvučem neke zaključke.

* 1. **Struktura projekta**

Projekat se sastoji iz dva iz izvršiva (executable) fajla koji predtavljaju server i klijent programe.

Server se kreira na odabranom portu koji se bira preko prvog argumenta komandne linije ./cchat\_server 8080. Server inicijalizuje potrebne parametre iz konstanti i pomoću funkcije start\_server() otvara socket i pokreće listen funkciju. Na glavnom thread-u je pokrenuta while petlja koja služi za primanje komandi za sigurno gašenje servera. Zatim server pokrece thread koji osluškuje za nove konekcije i kada dobije novu konekciju otvara novi thread koji obrađuje komunikaciju sa tim klijentom. Po završetku sesije sa klijentom thread se gasi pomoću funkcije pthead\_detach(). Podaci o trenutno aktivnim socket-ima i njihovim odgovarajućim thread-ovima se skladište u dinamički alociranoj strukturi handler\_data\_t. Prilikom primljene poruke od klijenta server zna pomoću te strukture o kom se klijentu radi i proso prosledi tu poruku svim aktivnim klijentima osim tom klijentu koji je poslao poruku. Prilikom unosa karatera "q" izlazi se iz aplikacije.

Klijent aplikacija iz argumenata komandne linije prihvata adresu i port servera na koji će se povezati ./cchat 192.168.1.21 8080. Pri uspesnoj konekciji aplikacija će pitati korisnika za željeno korisničko ime i omogućiti mu da komunicija sa trenutno povezanim klijentima. Kada se novi korisnik poveže na server svi trenutno povezani klijenti će dobiti nasumično izabrano obaveštenje o novopovezanom klijentu. Klijent aplikacija takođe koristi više thread-ova. Jedan za prihvatanje poruka sa servera a drugi za prihvatanje korisničkog unosa sa terminala. Za izlaz iz aplikacije koristimo kljucnu reč "quit".

# Zaključak

Projekat predstavlja dobru osnovu i uvod u socket programiranje i takodje sigurno baratanje thread-ovima bez curenja memorije. Ceo projekat je testiran programom valgrind i nema curenja memorije. Takodje ovaj projekat popunjava prazninu u koja postoji na internetu kada su u pitanju primeri za učenje o bazičnim multithreaded klijent- server sistemima u programskom jeziku C.

# Reference

*[1]* StackOverflow, [*https://stackoverflow.com/*](https://stackoverflow.com/)

*[2]* Github, [*https://github.com/*](https://github.com/)

*[3]* Geeksforgeeks, [*https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/*](https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/)

*[4]* Codereview, [*https://codereview.stackexchange.com/questions/154969/c-multithreaded-chat-client-and-server*](https://codereview.stackexchange.com/questions/154969/c-multithreaded-chat-client-and-server)

*[5]* LAMS, [*http://lams.metropolitan.ac.rs:8080/lams/*](http://lams.metropolitan.ac.rs:8080/lams/)