



PROJEKAT IZ OSNOVA RAČUNARSKIH MREŽA

TCP CHAT

Studenti:

Mladen Bazina, RA194/2020

Jovan Crnčević, RA129/2021

Mentor:

Nemanja Trifunović

Novi Sad, januar 2024.

Kratak opis projekta

- Projekat predstavlja jednostavan sistem za razmenu tekstualnih poruka između više klijenata putem TCP/IP protokola. Ovaj sistem sastoji se od dva programa:
 - serverskog programa (server.c)
 - klijentskog programa (client.c)
- Serverski program se ponaša kao centralna tačka za povezivanje korisnika, a klijentski program korisnicima omogućava slanje i primanje poruka.

Server program

- Inicijalizacija klijenta
 - Na početku, kreira se niz struktura ClientInfo koje predstavljaju informacije o klijentima.
 - Postavlja se mutex (clientsMutex) za sinhronizaciju pristupa nizu klijenata.
- Podešavanje server socket- a
 - Kreira se socket.
 - Pravi se struktura server koja sadrži informacije o serveru (IP adresa, port).
 - Vrš se bind server socket- a na određenu IP adresu i port.

Server program

- Slušanje na serveru
 - Server je u beskonačnoj petlji i čeka na dolazne konekcije.
- Povezivanje novog klijenta
 - Prihvata se nova konekcija od klijenta i prima se njegovo korisničko ime.
- Obrada klijenta
 - Svaki klijent se smešta u niz clients zajedno sa svojim podacima (socket, adresa, korisničko ime).
 - Za svakog klijenta se kreira zaseban thread (HandleClient) koji obrađuje komunikaciju sa tim klijentom.

Server program

- Obrada klijentskih poruka
 - U funkciji `HandleClient`, server neprestano prima poruke od klijenta.
 - Poruke se ispisuju na konzoli sa informacijama o korisniku koji ih šalje.
 - Poruke se potom šalju svim ostalim klijentima koristeći funkciju `BroadcastMessage`.
- Broadcast poruka
 - Funkcija `BroadcastMessage` šalje primljenu poruku svim klijentima osim onom koji je poslao poruku, oslanjajući se na mutex.

Klijent program

- Validacija argumenata komandne linije
 - Proverava se broj argumenata komandne linije jer klijent mora uneti svoje korisničko ime.
- Inicijalizacija klijentskog socketa
 - Klijent kreira socket i pravi se struktura sa IP adresom i portom servera.
- Povezivanje sa serverom
 - Klijent se povezuje sa serverom koristeći funkciju connect.

Klijent program

- Slanje korisničkog imena
 - Klijent šalje svoje korisničko ime serveru pomoću funkcije send.
- Potvrda konekcije
 - Klijent ispisuje poruku o uspešnom povezivanju sa serverom.
- Primanje poruka
 - Kreira se poseban thread (ReceiveMessages) za primanje poruka sa servera.
 - Thread neprestano prima poruke i ispisuje ih na konzoli.

Klijent program

- Slanje poruka
 - U while petlji se neprestano prima unos sa tastature i šalje se serveru kako bi ih on prosledio ostalim klijentima.
- Prekid veze sa serverom
 - Nakon završetka komunikacije klijent zatvara socket.

Napomene

- Mutex clientsMutex koristi se za sinhronizaciju pristupa klijentima kako bi se izbegli problemi sa pristupom nizu klijenata.
- Svaki klijent ima svoj thread za primanje poruka kako bi se omogućila simultana komunikacija sa više klijenata.
- Server podržava do MAX_CLIENTS istovremenih klijenata. Novi klijenti neće biti prihvaćeni ukoliko je kapacitet popunjen.
- Kao kostur za izradu projekta se koristila postavka sa vežbi 4.

Uputstvo za pokretanje programa

- Kompajliranje programa se, zbog threadova, vrši komandama:
 - `gcc -lpthread -o server server.c`
 - `gcc -lpthread -o client client.c`
- Prvo je potrebno pokrenuti serverski program komandom:
 - `./server`
- Nakon toga se pokreću klijenti, maksimalno njih `MAX_CLIENTS`, i kao argument komandne linije se unosi korisničko ime od maksimalno 20 karaktera:
 - `./client KORISNIČKO_IME`