

Prolećni semestar 2023/2024

PREDMET: CS130 C/C++ programski jezik

PROJEKTNI ZADATAK

Chat system

STUDENT

Aleksa Stanojević 5726

Student Softverskog inženjerstva

ASISTENT

Sofija Ilić

Niš, 09.06.2014.

Sadržaj

[1. Uvod 3](#_Toc7096)

[2. Analiza 4](#_Toc7097)

[2.1. Cilj programa 4](#_Toc7098)

[2.2. Elementi programa 4](#_Toc7099)

[2.3. Izrada programa 6](#_Toc7100)

[2.4. Funkcionisanje programa i prikazi ekrana 6](#_Toc7101)

[3. Zaključak 10](#_Toc7102)

[4. Reference 11](#_Toc7103)

# 1. Uvod

Ovaj projektni zadatak predstavlja čet konzolnu aplikaciju, u kojoj korisnik ima mogućnost da se poveže sa ostalim korisnicima preko servera u pythonu. Korisnik na početku ima mogućnost da unese ime sa kojim ga ostali korisnici prepoznaju na mreži.Nakon toga korisnik se povezuje na mrežu I ima mogućnost pisanja I čitanja poruka sa mreže u realnom vremenu.Poruke se unose čistim kucanjem u konzoli I pritiskom na taster “Enter” pri završetku ispisanja iste.Poruke koje se primaju sa mreže se pojavljuju u interaktivnom delu za prikaz poruka kako lokalnih(kucanih sa konzole) tako I prijemnih (prihvatanih sa mreže).Korisnik ima mogućnost izvšavanja odredjenih komandi sa prefiksom “#”.Nakon završenja program kada korisnik ugasi konzolu, korisnik obaveštava ostale korisnike na mreži da više nije u sesiji.

# 2. Analiza

## 2.1.Cilj programa

Cilj programa je da korisnik ima mogućnost da komunicira sa drugim korisnicima koji su fizički udaljeni od njega. Takođe ima mogućnost čuvanja poruka sesija kao I prikaza obrađenih poruka.

## 2.2.Elementi programa

Progam ima 4 header fajla,19 funkcija,3 strukture i “main.c” fajl.

* logger.h
  + open\_logger\_file
  + close\_logger\_file
  + logger\_log
  + read\_logger\_file
* message\_system.h
  + struct Message
  + messages\_clear
  + messages\_add
* networking.h
  + struct NetworkBuffer
  + connection\_read\_message
  + connection\_send\_message
  + process\_message\_receiving
  + process\_message\_sending
  + struct Connection
  + connection\_connect
  + connection\_cleanup
* smart\_console.h
  + console\_clear
  + console\_separator
  + console\_move\_cursor
  + console\_print\_centered
  + console\_print\_colored
  + print\_console

Header fajl logger sadrži funkcionalnosti za obradu čuvanja I isčitavanja podataka iz fajla.



*Slika 1. Header fajl\_logger.h*

LOGGER\_PATH – Konstanta koja oznacava putanju do lokalnog fajla za obradu podataka.

open\_logger\_file - Funkcija koja služi za otvaranje lokalnog fajla za upis/citanje podataka.

close\_logger\_file - Funkcija koja služi za zatvaranje lokalnog fajla za upis/citanje podataka.

logger\_log - Funkcija koja služi za upis u podataka kao novu liniju u fajlu.

read\_logger\_file - Funkcija koja služi za ispis celokupnih podataka koje se nalaze u fajlu.

Header fajl message system sadrži funkcionalnosti za obradu skladištenja,dodavanja I brisanja poruka.



*Slika 2. Header fajl message\_system.h*

MAX\_MESSAGES\_DISPLAYED – Konstanta koja oznacava maksimalni broj poruka prikazanih na ekranu.

BUFFER\_SIZE – Konstanta koja oznacava maksimalnu dužinu poruke

Message – Struktura koja čuva podatke o poruci (poruku,korisničko ime,vreme slanja)

messages\_clear – Funkcija koja služi za brisanje trenutnih poruka iz memorije

messages\_add – Funckija koja služi za dodavanje nove poruke u memoriju

Header fajl networking sadrži funkcionalnosti za povezivanje na sesiju I obradu slanja I primanja poruka sa iste.





*Slika 3. Header fajl networking.h*

PORT – Konstanta koja označava port na kome se server nalazi

HOST – Konstanta koja označava adresu na kojoj se server nalazi

NetworkBuffer – Struktura koja čuva podatke o poslatom I prijemnom bufferu podataka preko mreže

status\_t – Predefinicija imena unsigned long

SUCCESS – Konstanta koja označava potvrdnu povratnu poruku funkcije

SERVER\_ERROR – Konstanta koja označava grešku na server kao povratnu poruku funkcije

INVALID\_ARGUMENT\_ERROR – Konstanta koja označava grešku pogrešnog argumenta funkcije

UNDEFINED\_ERROR – Konstanta koja označava nedefinisanu grešku u kodu funkcije

connection\_read\_message – Funkcija koja služi za primanje poruka od serverske strane

connection\_send\_message – Funkcija koja služi za slanje poruka na serversku stranu

process\_message\_receiving – Funkcija koja služi za kompletnu obradu svih procesa primanja podataka sa serverske strane

process\_message\_sending – Funkcija koja služi za kompletnu obradu svih procesa slanja podataka na serversku stranu

Connection – Struktura koja čuva podatke o ostvarenoj konekciji sa serverskom stranom

connection\_connect – Funkcija koja služi za kompletan proces povezivanja klijenta na serversku stranu

connection\_cleanup – Funkcija koja služi za kompletan proces odjavljivanja sa serverske strane I otklanjanja podataka iz memorije

Header fajl smart console sadrži funkcionalnosti za obradu konzole,ispis u konzoli, prikaz,formatiranje I slično.



*Slika 4. Header fajl smart\_console.h*

console\_clear – Funkcija koja služi za čišćenje prikaza konzole

console\_separator – Funkcija koja služi za prikaz separator linije u konzoli

console\_move\_cursor – Funkcija koja služi za pomeranje kursora u konzoli radi ispisa teksta

console\_print\_centered – Funkcija koja služi za štampanje teksta u konzoli koji je centriran

CONSOLE\_COLOR\_RED – Konstanta koja označava crvenu boju teksta u konzoli

CONSOLE\_COLOR\_GREEN – Konstanta koja označava zelenu boju teksta u konzoli

CONSOLE\_COLOR\_YELLOW – Konstanta koja označava žutu boju teksta u konzoli

CONSOLE\_COLOR\_BLUE – Konstanta koja označava plavu boju teksta u konzoli

CONSOLE\_COLOR\_MAGENTA – Konstanta koja označava magenta boju teksta u konzoli

CONSOLE\_COLOR\_CYAN – Konstanta koja označava cyan boju teksta u konzoli

CONSOLE\_COLOR\_WHITE – Konstanta koja označava belu boju teksta u konzoli

console\_print\_colored – Funkcija koja služi za štampanje teksta u konzoli koji je obojen

print\_console – Funkcija koja služi za štampanje grafičkog korisničkog interfejsa programa u konzoli

Fajl main.c koristimo za pozivanje glavnih funkcija izvršenja programa.

*Slika 5. Entry point main.c*

Python kod za pokretanje servera ovog projekta

*Slika 6. python\_server.py*

## 2.3.Izrada programa

Najpre je izradjen header fajl logger.h sa funkcionalnostima čuvanja I čitanja logova sa fajlom kako bismo beležili svako moguće dešavanje od početka kreiranja programa do samog završetka (I nakon završetka) istog.

Nakon toga smo krenuli na kreiranje sistema za konektovanje na mrežu (header networking.h),gde smo pre svega kreirali funkcije za izradu direktne konekcije na server. Nakon testiranja uspešne konekcije smo izradili funkcije za uspešan (takođe uz testiranje) prenos podataka sa klijentske na serversku stranu i vice versa,nakon čega su odrađeni procesi za neprekidnu obradu slanja I primanja podataka sa servera.

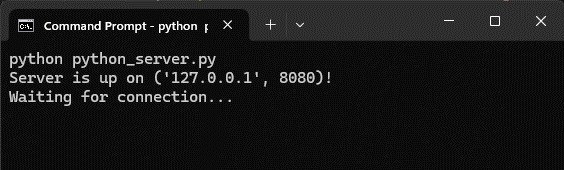
Nakon čega je usledila obrada message system headera kako bismo lakše skladištili obrađene poruke (poslate I primljene) sa mreže.

I na kraju je obrađen header fajl smart console koji nam služi za ukrašavanje grafičkog korisničkog interfejsa (konzole u ovom slučaju).

Nakon obrade svih header fajlova, sve funkcionalnosti su spojene u main.c fajlu koji koristimo kao entry point izvršnog programa.

## 2.4. Funkcionisanje programa i prikazi ekrana

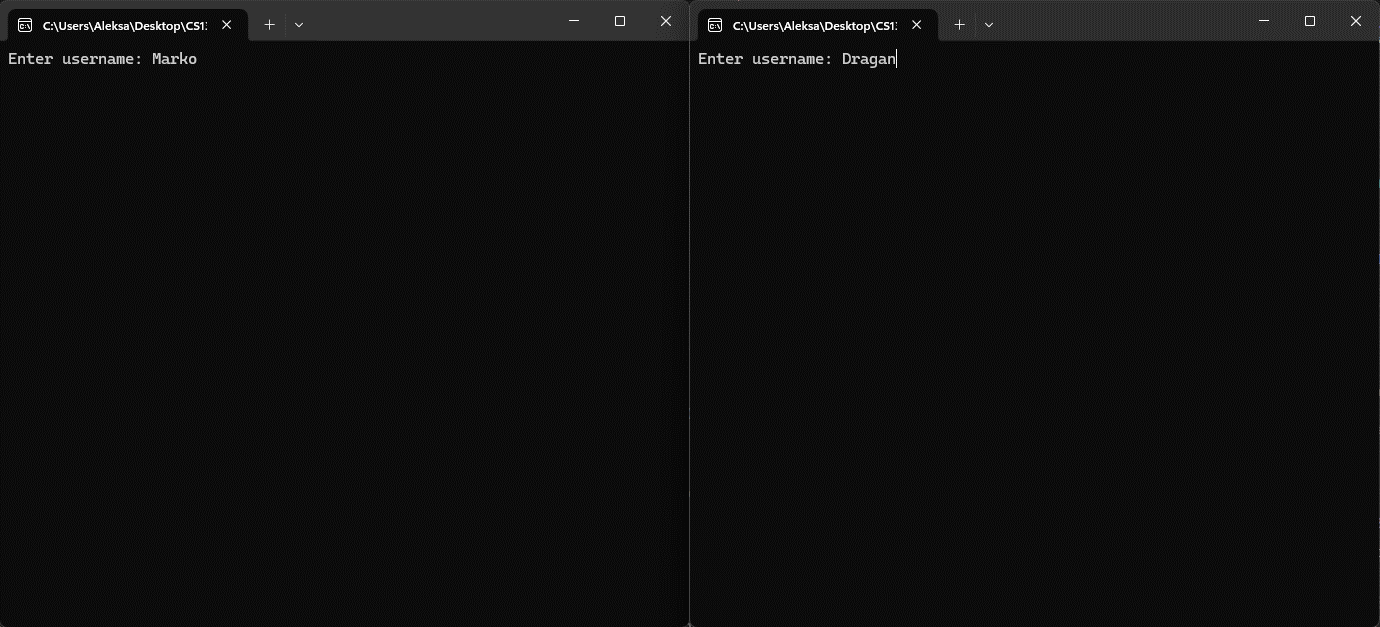
Za početak pokrećemo python server (ukoliko nije pokrenut) na jednoj mašini koja će izvršavati ulogu server.



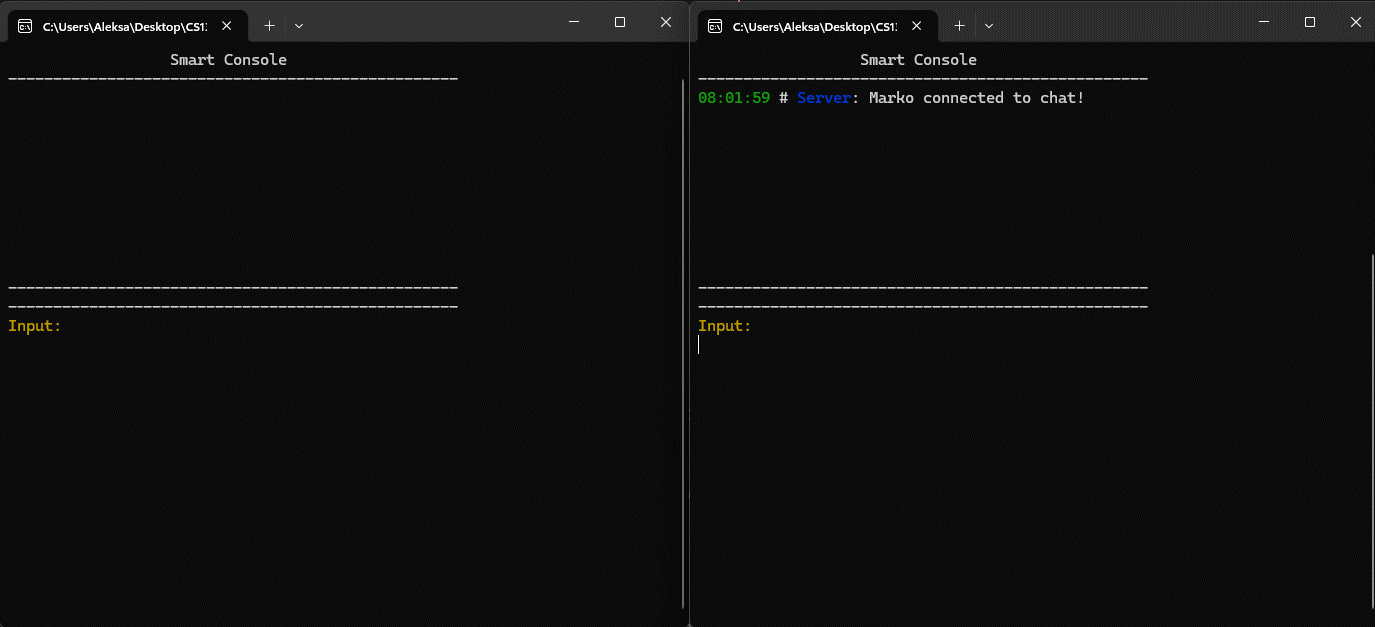
*Slika 7. Pokretanje python servera*

*Napomena: Prikaz LEVO na slikama koje slede je simulacija prikaza klijenta na fizički udaljenoj mašini.*

Zatim pokrećemo klijentsku stranu (naš program u C jeziku), koji nam odmah (ukoliko nije bilo problema sa otvaranjem logger fajlova) zahteva korisničko ime.

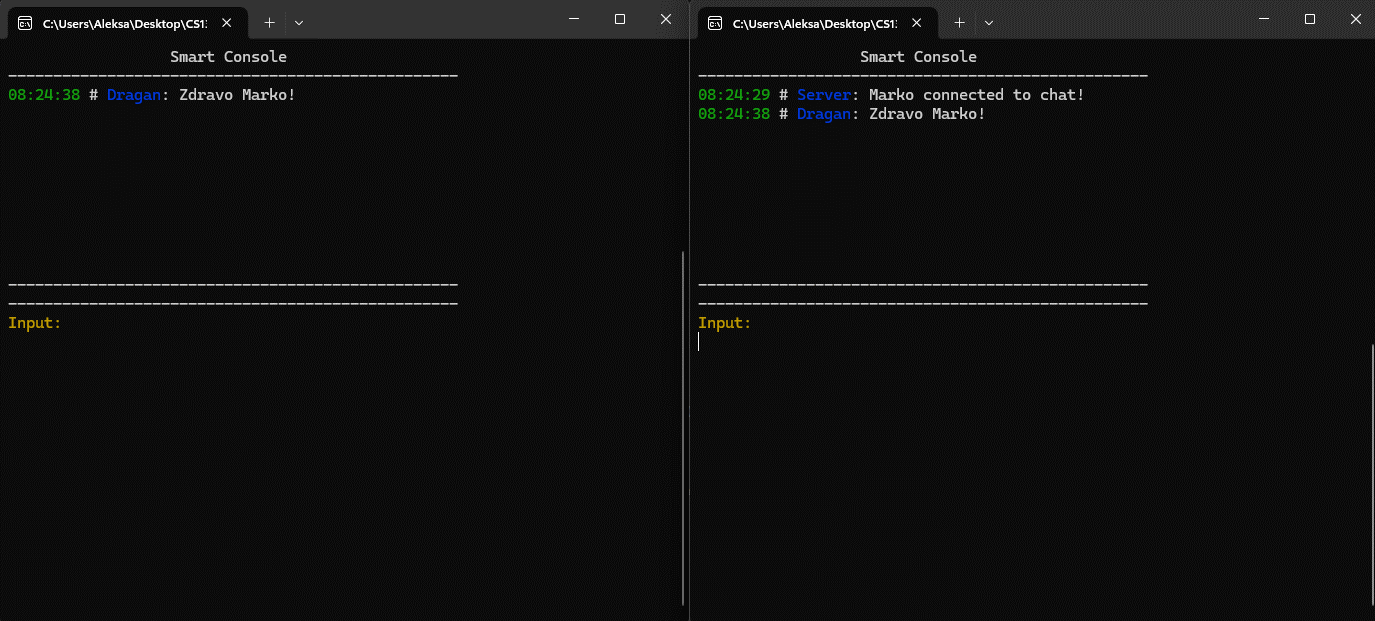
 *Slika 8. Program traži ime korisnika*

Nakon unošenja korisničkog imena, prikazuje nam se korisnički interfejs pametne konzole.Ukoliko se neko od korisnika priključio konekciji izaći će pozdravna poruka (kao na slici ispod)

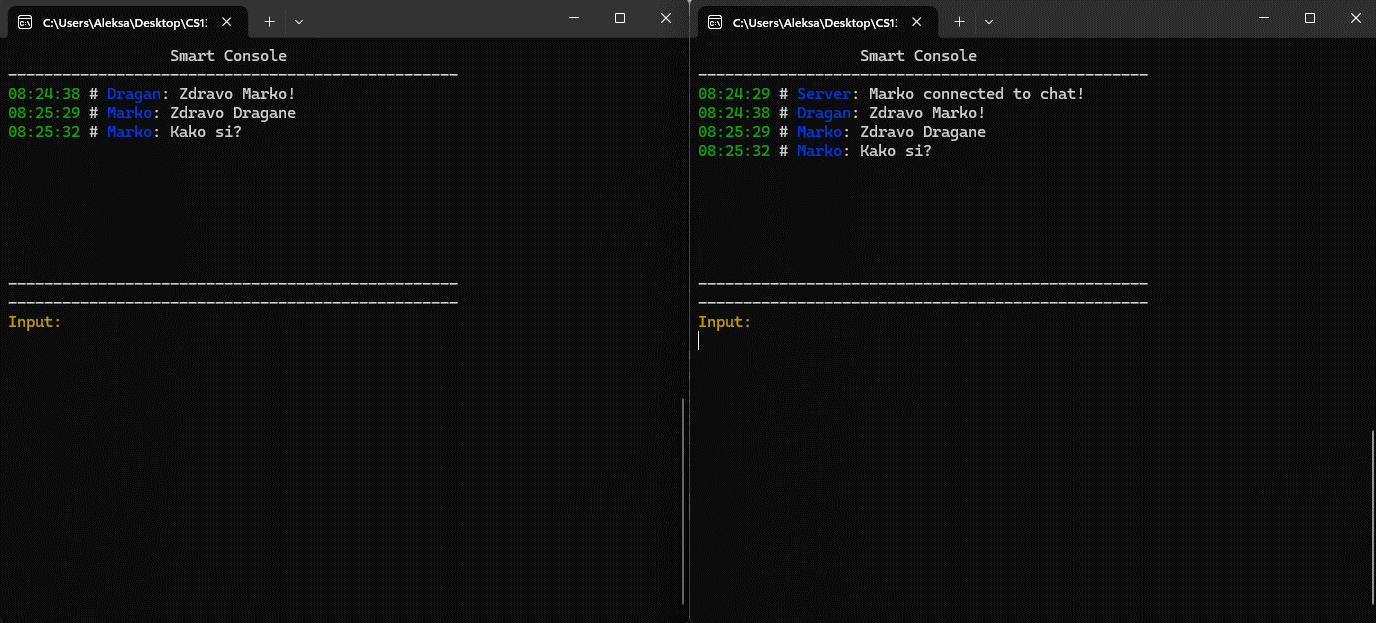


*Slika 9. Korisnički interfejs*

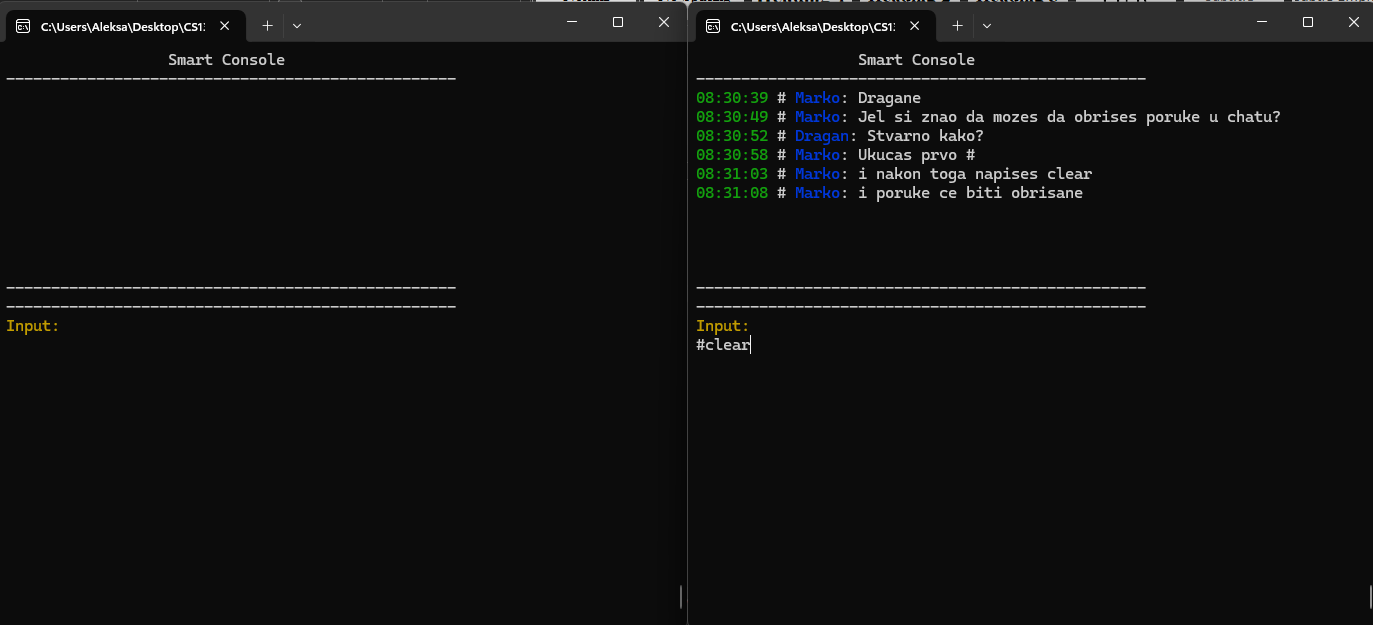
Kroz korisnički interfejs,unosom reči sa tastature omogućavamo korisniku da šalje poruke kao I da ih prima u realnom vremenu.

 *Slika 10. Proces poruke*

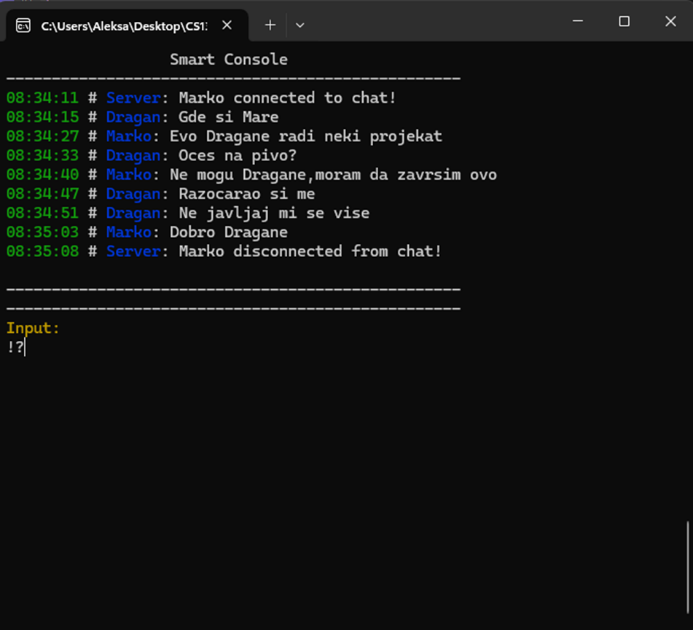
Korisnik šalje I prima poruke u realnom vremenu, tj. Omogućavamo mu da šalje više poruka kao I da ih prima bez čekanja na odgovor od strane drugih korisnika.

 *Slika 9. Tok razgovora*

Ukoliko korisnik želi da obriše trenutne poruke (iz konzole I memorije) potrebno je da korisnik unese komandu clear koja sledi uz prefix #. Kompletan kod komande je “#clear”,nakon koje će sve prethodne poruke biti izbačene iz memorije I konzole (vidi sliku levo).

 *Slika 10. Brisanje poruka*

Kadakorisnik isključi aplikaciju, ostalim korisnicima se javlja poruka sa servera da se taj korisnik isključio sa mreže.



*Slika 10. Kraj programa*

# 3. Zaključak

Softver je namenjen korisnicima koji žele da komuniciraju sa ostalim korisnicima koji su fizički udaljeni od korisnika.Tokom rada na projektu naučio sam osnove C jezika kao što su rad sa nizovima,pokazivačima,rad sa fajlovima,kao i rad sa socket mrežnom konekcijom.

# 4. Reference

1. Stack Overflow, [http://stackoverflow.com/questions/15388041/how-to-write-stdstring-tofile](http://stackoverflow.com/questions/15388041/how-to-write-stdstring-to-file)
2. Cplusplus, <http://www.cplusplus.com/forum/beginner/7553/>
3. Cplusplus, <http://www.cplusplus.com/doc/tutorial/files/>
4. Cplusplus, <http://www.cplusplus.com/reference/cstdlib/rand/>
5. LAMS, <http://lams.metropolitan.ac.rs:8080/lams/index.do>