



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У  
НОВОМ САДУ



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА  
НОВИ САД  
Департман за рачунарство и аутоматику  
Одсек за рачунарску технику и рачунарске комуникације

## ИСПИТНИ РАД

Кандидат: Милица Војновић  
Број индекса: РА 59-2020

Предмет: Међурачунарске комуникације и рачунарске мреже 1  
Тема рада: Наменски систем електронске поште

Ментор рада: Проф. Илија Башичевић

Нови Сад, Јануар 2024.

## 1. Zadatak

Korišćenjem QT razvojnog okruženja realizovati primer namenskog sistema elektronske pošte. Primer se sastoji iz serverske i klijentske strane. Klijent preko tastature može da zadaje sledeće komande:

Login – Prijava korisnika.

Logout – Odjava korisnika.

Send – Slanje poruke.

Check – Provera da li postoji poruka za prijavljenog korisnika.

Stat – Provera koja vraća statistiku poput broja i tipova poruka za prijavljenog korisnika.

Delete – Brisanje poruke za prijavljenog korisnika.

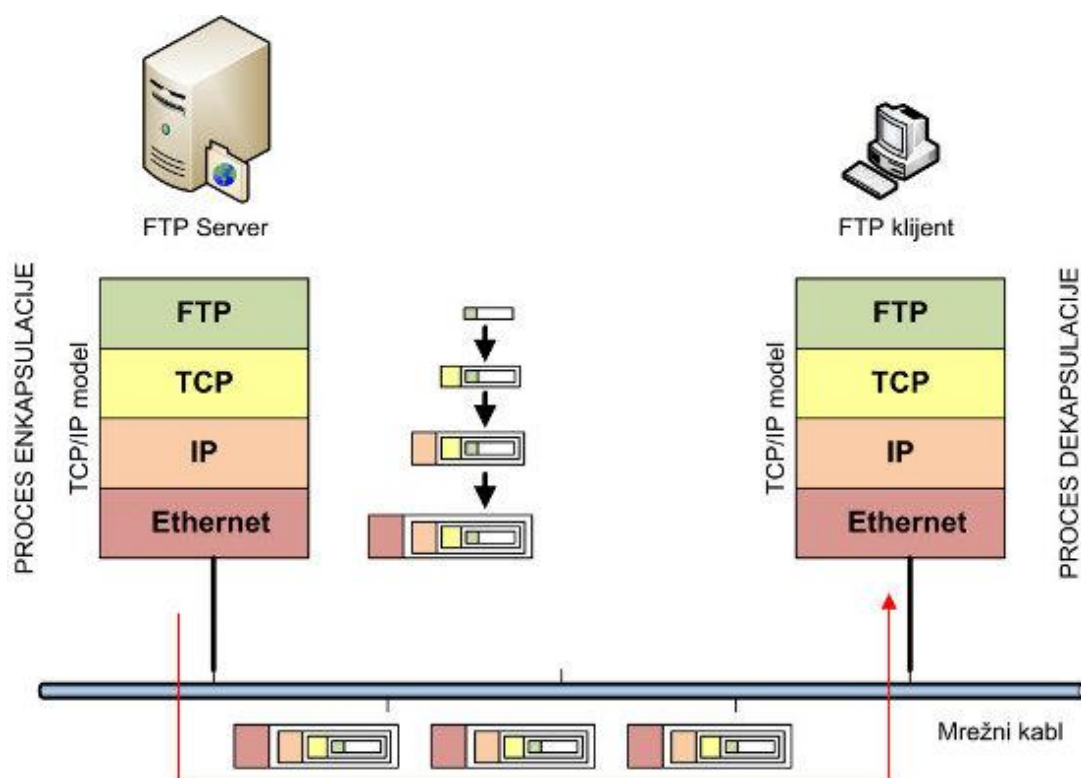
Clean – Brisanje svih poruka za prijavljenog korisnika.

Receive – Prijem poruke.

Napisati odgovarajuću dokumentaciju po ugledu na priloženi šablon.

## 2. Koncept rešenja

Prvenstveno, radi lakšeg razumevanja i planiranja protokola, formirani su MSC (Message Sequence Chart) i SDL (Specification and Description Language) dijagrami. Oni će biti priloženi zajedno sa rešenjem u PDF formi. Rešenje je realizovano pomoću QT Creatora, klijentska strana je formirana kao QT Widgets Application sa svojim UI radi preglednosti i lakšeg korišćenja, a serverska strana kao QT Console Application. Odabran je TCP (Transmission Control Protocol) kao glavni protokol komunikacije, jer je on najpogodniji za dati primer razgovora klijenta i servera. Svi razgovori klijenta i servera su modelovani po MSC i SDL dijagramima.



Slika 1. TCP protokol

### 3. Opis rešenja

Okruženje u kojem je projekat urađen nudi rešenje za rad sa mrežnom komunikacijom u vidu biblioteka kao što su <QTcpServer> (klasa koja obezbeđuje server baziran na TCP-u) i <QTcpSocket> (klasa koja obezbeđuje TCP sockete). Sama logika obrade odgovora sa klijentske i serverske strane je urađena u funkcijama koje su namenjene za određene zadatke koji postoje.

Primeri korišćenja biblioteka sa serverske strane:

```
QTcpServer server;                //otvaranje servera za slušanje korisnika koji žele da se povežu
server.listen( "127.0.0.1", 12345);    //proizvoljne fiksne vrednosti, adresa i port servera
QTcpSocket incomingSocket = server.nextPendingConnection();    //formiranje socketa
QString receivedMessage = incomingSocket->readAll().data();    //prijem poruka
incomingSocket->write("LOGIN_CONFORMATION");    //slanje TCP paketa
```

Primeri korišćenja biblioteka sa klijentske strane:

```
QTcpSocket clientSocket;    //pronalaženje servera i formiranje socketa na klijentskoj strani
clientSocket.connectToHost("127.0.0.1", 12345, QIODevice::ReadWrite);
clientSocket.write("LOGIN_REQUEST");    //slanje poruke
QString receivedMessage = clientSocket.read(MAX_SIZE_OF_MESSAGE);    //prijem poruke
clientSocket.disconnectFromHost();    //odvezivanje od servera
```



## 4. Testiranje

Tokom testiranja pokrili smo sve slučajeve i stanja u kojima klijent može da bude. Funkcije kao što su Login, Send i Delete zavise od parametara koji su poslani zajedno sa signalom. U slučaju da parametri nisu odgovarajući (pr. klijent nije uneo dobru šifru) klijent od servera dobije odbijajući signal, dok za tačan unos dobija potvrdu. Funkcije kao što su Check, Stat i Receive zajedno sa odgovorom od servera dobijaju i informacije koje su tražile, tj. server pored signala šalje i neke vrednosti. Logout i Clean su funkcije koje šalju samo signal, i isto tako samo ga i primaju.

Naravno, sve ove funkcije zavise od toga da li je konekcija uopšte uspostavljena, da li je server i dalje aktivan i da li su poruke sa jedne i sa druge strane stigle, ili su se zagubile u mreži. U ovom projektu radimo isključivo sa “127.0.0.1” adresom, koja je IPv4 specijalna adresa (loopback address) rezervisana samo za rad na lokalnom računaru. Ova adresa na sebi nema puno saobraćaja, zbog čega je gubitak paketa gotovo nepostojeći.

Kao sredstvo za testiranje koristili smo Wireshark, alat za analizu paketa na mreži.

\*Adapter for loopback traffic capture

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

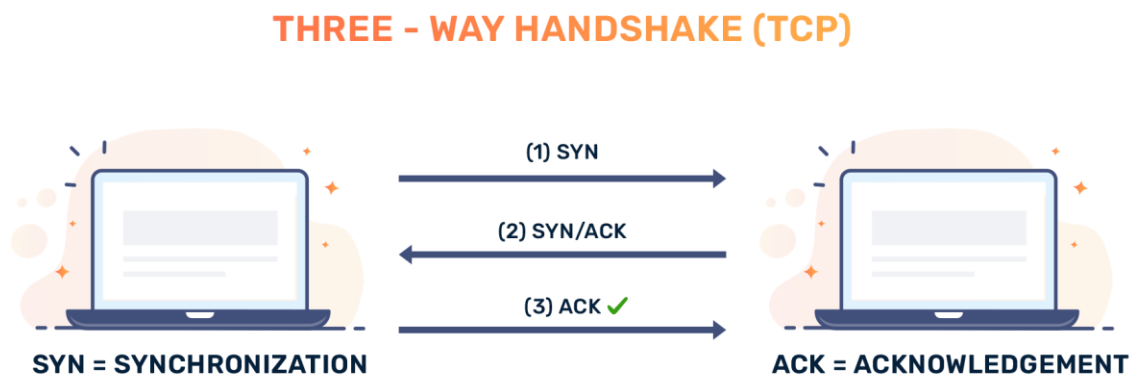
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	52533 → 12345 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 S
2	0.000087	127.0.0.1	127.0.0.1	TCP	56	12345 → 52533 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=654
3	0.000126	127.0.0.1	127.0.0.1	TCP	44	52533 → 12345 [ACK] Seq=1 Ack=1 Win=327424 Len=0
4	8.611012	127.0.0.1	127.0.0.1	TCP	64	52533 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=20
5	8.611069	127.0.0.1	127.0.0.1	TCP	44	12345 → 52533 [ACK] Seq=1 Ack=21 Win=2161152 Len=0
6	8.611465	127.0.0.1	127.0.0.1	TCP	56	12345 → 52533 [PSH, ACK] Seq=1 Ack=21 Win=2161152 Len=12
7	8.611498	127.0.0.1	127.0.0.1	TCP	44	52533 → 12345 [ACK] Seq=21 Ack=13 Win=327424 Len=0

Slika 4. Primer gledanja paketa na adresi u Wireshark-u

Na slici možemo videti primer povezivanja sa serverom, slanja jednog signala i dobijanja odgovora. Prva tri paketa predstavljaju slanje želje za povezivanjem od klijentske strane [SYN], potvrda povezivanja od serverske strane [SYN, ACK] i potvrda o pristiglom paketu opet od klijenta [ACK]. Sledeća dva paketa su signal od klijenta [PSH, ACK] i potvrda od servera da je paket stigao [ACK]. Poslednja dva paketa su odgovor od servera [PSH, ACK] i opet potvrda od klijenta da je paket stigao [ACK].

Ovaj primer povezivanja je poznat kao Three-Way Handshake, koji je proces korišćen u TCP/IP kako bi se ostvarilo povezivanje između klijenta i servera.



Slika 5. Izgled Three-Way Handshake-a

## **5. Zaključak**

Projekat je realizovan u QT Creator 12.0.1 okruženju, u C++ programskom jeziku, napravljena su dva projekta, jedan je Server i drugi Client..

MSC i SDL se nalaze u odvojenom PDF-u koji će biti predan zajedno sa ovim PDF-om, a za njegovu izradu korišćen je alat draw.io. Projekat je napravljen za lokalnu mrežu i koristi TCP protokole. Napravljen je namenski sistem elektronske pošte sa funkcionalnim elementima kao što je bilo i postavljeno u zadatku (Login, Logout, Send, Delete, Check, Receive, Stat, Clean).



## 6. Literatura

- [1] *Priručnik radnog okruženja za pisanje protokola, Verzija 0.2*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2007
- [2] <https://www.rt-rk.uns.ac.rs/predmeti/e2/mrkirm-1-me%C4%91ura%C4%8Dunarske-komunikacije-i-ra%C4%8Dunarske-mre%C5%BEE-1>
- [3] <https://www.youtube.com/watch?v=EkjaiDsiM-Q&list=PLS1QulWo1RIZiBcTr5urECberTITj7gjA&pp=iAQB>
- [4] <https://github.com/ktulgar/Learning-QT/tree/main/Simple%20Server%20-%20Cient%20Application>