



Mreže računala

Vježbe 05 – složeniji primjer

Matko Botinčan
Zvonimir Bujanović
Igor Jelaska
Maja Karaga

Dizajn protokola

- dosadašnji klijenti i serveri bilo su prilično jednostavni – ili su poslali samo 1 poruku, ili su naizmjenično primali i slali istovrsne poruke
- u ovim vježbama ćemo napraviti malo složeniji par (klijent, server) koji će trebati razmjenjivati različite poruke različitih duljina
- bit će potrebno *dizajnirati protokol* za komunikaciju između klijenta i servera

Zadatak 1

- potrebno je napraviti dizajn protokola za mrežnu aplikaciju za razmjenu poruka
- sve poruke se čuvaju na serveru (višenitnom)
- svaka poruka ima pošiljatelja i primatelja
- klijent od servera može tražiti:
 - popis svih do sada serveru poznatih klijenata
 - popis svojih nepročitanih poruka (pročitane se brišu)
 - slanje nove poruke nekom drugom korisniku
 - prilikom spajanja na server, klijent treba navesti svoje ime (radi jednostavnosti nećemo koristiti password)

Rješenje Zadatka 1

Jedan mogući dizajn sustava komunikacije:

- imamo nekoliko vrsta poruka između servera i klijenta:
 - *LOGIN ime* – klijent prilikom logiranja serveru šalje poruku sa svojim imenom; to mora biti prva poruka koju server prima od danog klijenta
 - *USERS* – klijent šalje serveru zahtjev za popisom korisnika
 - *USERLIST ime1 ime2 ... imeN* – server šalje klijentu odgovor na zahtjev sa popisom korisnika
 - *BYE* – klijent šalje serveru znak za prekid komunikacije. Nakon toga server prekida komunikaciju.

Rješenje Zadatka 1

Jedan mogući dizajn sustava komunikacije:

- imamo nekoliko vrsta poruka između servera i klijenta:
 - *NEWMESSAGE ime poruka* – klijent šalje serveru novu poruku *poruka* za korisnika *ime*
 - *CHECKMESSAGES* – klijent šalje serveru poruku da želi provjeriti ima li novih poruka
 - *MESSAGE N ime poruka* – server šalje klijentu informaciju da ima još *N* nepročitanih poruka; šalje mu jednu od njih: poslao ju je korisnik sa imenom *ime*. Nakon ovog klijent ima još *N-1* nepročitanu poruku; pristigla se označava kao pročitana. Ako je *N=0*, informacija ne sadrži ime ni poruku.

Rješenje Zadatka 1

Protokol komunikacije:

- svaka poruka sastojat će se od 2 dijela: header-a i tijela poruke
- header će se sastojati od 2 integera
 - svaki integer šaljemo binarno (4 byte-a)
 - prvi integer predstavlja duljinu poruke, ne uključujući header
 - drugi integer predstavlja kod kojim se jednoznačno određuje jednu od 7 vrsta poruke (LOGIN, USERS, USERLIST, NEWMESSAGE, CHECKMESSAGES, MESSAGE, BYE). Vrijedosti oznaka će biti definirane u datoteci messageProtocol.h koju trebaju uključiti i klijent i server.

Rješenje Zadatka 1

Na primjer:

```
// messageProtocol.h
```

```
#define LOGIN      1
#define USERS      2
#define USERLIST   3
#define BYE        4
#define NEWMESSAGE 5
#define CHECKMESSAGES 6
#define MESSAGE    7
```

Primjer poruke

5	1	Mirko
---	---	-------

header

tijelo poruke

Primjer komunikacije

(headere pišemo u [], dekadski radi lakšeg čitanja)

```
client: [5,LOGIN] Mirko
```

```
client: [0,USERS]
```

```
server: [17,USERLIST] Mirko Slavko Ana
```

```
client: [21,NEWMESSAGE] Slavko Bok! Kako ide?
```

```
client: [0,CHECKMESSAGES]
```

```
server: [21,MESSAGE] 2 Ana Pa di si Mirko!
```

```
client: [0,BYE]
```

Rješenje Zadatka 1

Klijent:

- prima ime klijenta npr. putem komandne linije
- "iscrta" meni klijentu; u meniju nudi:
 - ispis svih korisnika koji su poznati serveru
 - pregledavanje novih poruka pristiglih za tog korisnika
 - pisanje nove poruke za nekog drugog korisnika
 - izlaz iz programa (prekid komunikacije)

Rješenje Zadatka 1

Server:

- ima popis svih korisnika u datoteci `message.userlist.txt`; kada se neki korisnik ulogira, pogleda se u datoteku postoji li njegovo ime; ako ne, doda se na kraj datoteke
- za svakog korisnika ima popis njegovih novih poruka u datoteci `message.ime.txt`
- radi jednostavnosti: 1 poruka = 1 linija teksta
- kada korisnik pošalje novu poruku za nekog sa imenom *ime*, ona se doda na kraj datoteke `message.ime.txt`
- kada korisnik provjerava svoje poruke, prebroji se koliko redaka ima datoteka `message.korisnik.txt`; korisniku se pošalje prvi redak te datoteke. Taj prvi redak se onda obriše iz datoteke.

Rješenje Zadatka 1

`message.userlist.txt:`

```
Mirko
Slavko
Ana
```

`message.Mirko.txt:`

```
Ana Pa di si Mirko!
Slavko Evo tako malo gledam sta ima...
```

Zadatak 2

- Napravite implementaciju servera i klijenta prema opisanom protokolu.