

Sveučilište u Zagrebu
PMF – Matematički odsjek



Mreže računala

Vježbe 05

Zvonimir Bujanović
Slaven Kožić
Vinko Petričević

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 mrežnu aplikaciju koja će služiti za vođenje jednostavnog skladišta.
- Server vodi evidenciju o količini pojedinog artikla koji se čuva na skladištu.
- Klijent može:
 - dovoziti nove artikle na skladište
 - odvoziti postojeće artikle sa skladišta
 - provjeravati koliko pojedinih artikala ima na skladištu
- Server treba davati klijentu odgovor o eventualnom neuspješnom izvršavanju zahtjeva.

Rješenje **Zadatka 1**

Jedan mogući dizajn sustava komunikacije:

- imamo nekoliko vrsta poruka između servera i klijenta:
 - *STAVI artikl količina* – klijent šalje serveru informaciju o tome da želi na skladište dovesti izvjesnu količinu artikla
 - *UZMI artikl količina* – klijent šalje serveru informaciju o tome da želi sa skladišta odvesti izvjesnu količinu artikla
 - *KOLIKO artikl* – klijent šalje serveru poruku da želi doznati koliko se artikla nalazi na skladištu
 - *KOLIKO_R artikl količina* – server šalje klijentu poruku sa količinom artikla koja se nalazi u skladištu
 - *BOK* – klijent šalje serveru znak za prekid komunikacije. Nakon toga server prekida komunikaciju.

Rješenje **Zadatka 1**

Jedan mogući dizajn sustava komunikacije:

- server na svaki klijentov zahtjev odgovara je li zahtjev uspio ili nije
- odgovor je oblika
 - *ODGOVOR poruka* – poruka je string koji je jednak "OK" ako je zahtjev uspješno obrađen, inače sadrži opis greške

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 6 vrsta poruke (STAVI, UZMI, KOLIKO, KOLIKO_R, BOK, ODGOVOR). Vrijednosti oznaka će biti definirane u datoteci skladisteProtokol.h koju trebaju uključiti i klijent i server.

Rješenje Zadatka 1

Na primjer:

```
// skladisteProtokol.h
```

```
#define UZMI          1
```

```
#define STAVI         2
```

```
#define KOLIKO        3
```

```
#define KOLIKO_R      4
```

```
#define BOK           5
```

```
#define ODGOVOR       6
```

Primjer poruke



header

tijelo poruke

Primjer komunikacije

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

```
client: [7,STAVI] DASKE 3
```

```
server: [2,ODGOVOR] OK
```

```
client: [7,UZMI] DASKE 15
```

```
server: [27,ODGOVOR] Nema dovoljno artikla DASKE
```

```
client: [3,KOLIKO] DASKE
```

```
server: [2,ODGOVOR] OK
```

```
server: [7,KOLIKO_R] DASKE 3
```


Rješenje Zadatka 1

Klijent:

- Prima IP-adresu i port servera npr. putem komandne linije.
- "Iscrta" meni za korisnika; u meniju nudi:
 - ispis koliko ima nekog artikla na skladištu
 - stavljanje neke količine artikla u skladište
 - micanje neke količine artikla sa skladišta
 - izlaz iz programa (prekid komunikacije)

Rješenje Zadatka 1

Server:

- Ima polje u kojem je popis imena i količina pojedinog artikla. (Elegantnije je koristiti tip map iz STL-a.)
- Kada klijent pošalje zahtjev za stavljanje novog artikla na skladište, server prođe kroz polje i ako u njemu postoji taj artikl, poveća njegovu količinu, a ako nema tog artikla onda ga doda na kraj polja.
- Ako klijent traži artikl kojeg nema u skladištu ili ga nema dovoljno, server šalje poruku o grešci i ne provodi klijentovu naredbu.

Zadatak 2

- Napravite implementaciju servera i klijenta prema opisanom protokolu.

Zadatak 3

- Dodajte u prethodni program mogućnost da klijent dobije popis svih artikala koji se nalaze u skladištu.