



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA



INDUSTRIJSKI KOMUNIKACIONI PROTOKOLI U EES PROJEKAT

Asistent:
Saša Tošić

Studentkinje:
Milica Blaženović, PR 136/2018
Dajana Mišević, PR 3/2019

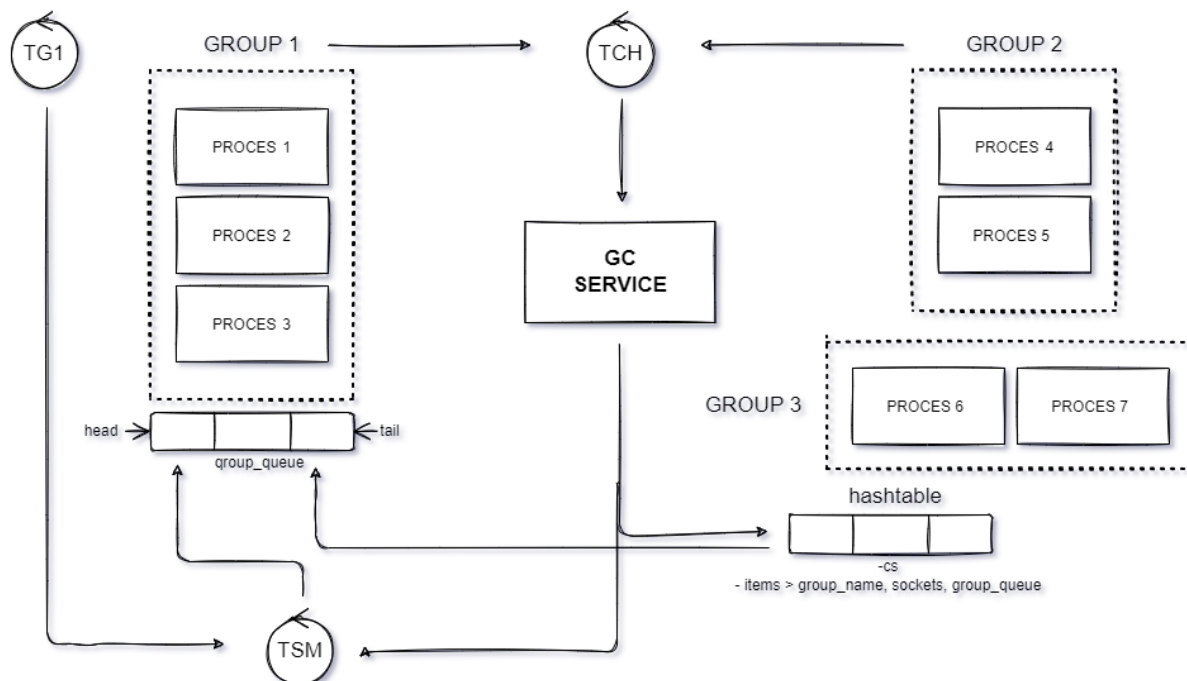
Novi Sad, 2022.

Sadržaj

Sadržaj	1
Opis dizajna	2
Strukture podataka	3
Testiranje rešenja	4

Opis dizajna

Razviti servis za razmenu podataka preko redova za poruke (Group Communication Service). Servis treba da omogući razmenu poruka izmedju više procesa koji se nalaze na istom servisu, preko odvojenih redova poruka za svaku grupu procesa. Jedan proces može pripadati jednoj grupi, a grupa može imati i više članova ali samo jedan red poruka.



Glavnu komponentu projekta predstavlja **Server** sa kojim komunicira 1 ili više klijenata. **Klijent** implementira interfejs sa metodama za konektovanje – klijent se kači na određeni red na servisu (ako grupa ne postoji kreira je, ako postoji samo dodaje socket konektovanog klijenta na datu grupu u hash tabeli), slanje poruke - klijent šalje poruku svim procesima iz iste grupe uz ispisivanje poruke na ekranu, diskonektovanje - klijent šalje zahtev za diskonekcijom, server ga briše iz grupe i zatvara njegov socket. Pri pokretanju klijenta, otvara mu se mogućnost za povezivanje na grupu i za izlazak iz sistema. Klikom na prvu opciju klijent šalje komandu za konektovanjem, server je obradi, doda u hash tabelu i vrati klijentu povratnu informaciju o uspešnom konektovanju. Klijentu se nakon toga otvara meni za slanje poruka, a u svakom trenutku prikazana mu je opcija za diskonektovanje.

```
D:\VII SEMESTAR\industrijski komunikacioni protokoli u ees\proj...
----- DOBRODOSLI U SERVIS ZA GRUPNI CET :D-----
1. PRIKLJUCI SE GRUPI
2. IZADJI

Klijent je spreman za recv!
Unesite ime grupe u kojoj zelite slati poruke:
ikp

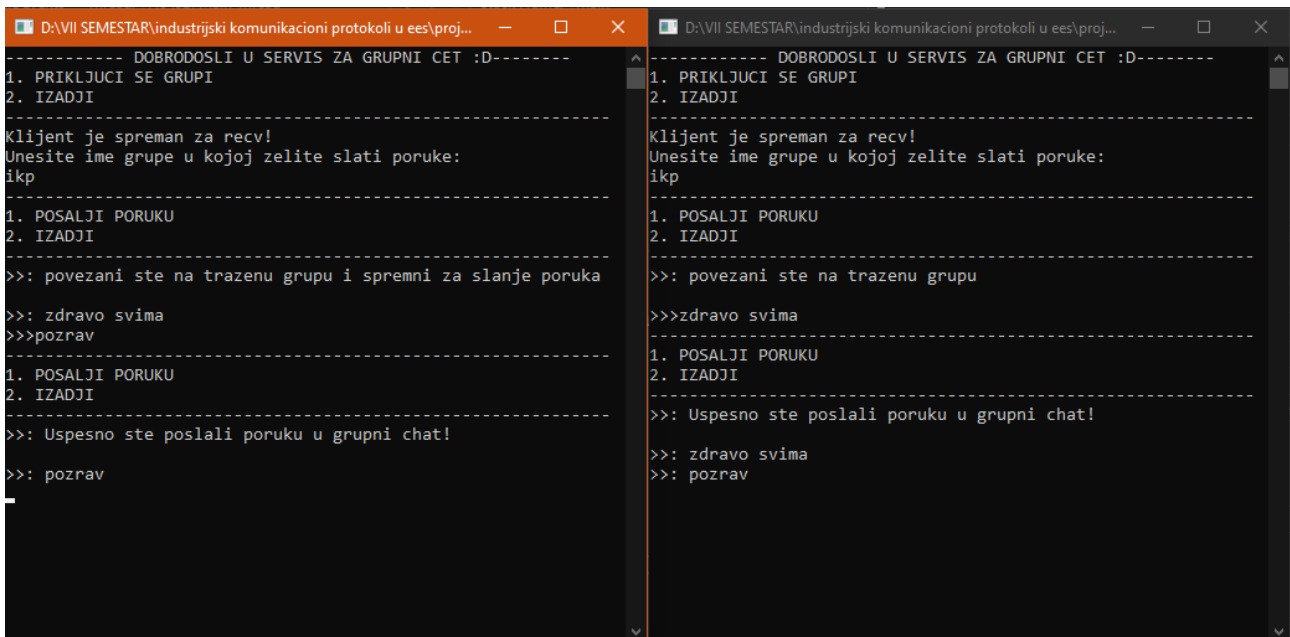
1. POSALJI PORUKU
2. IZADJI

>>: povezani ste na trazenu grupu i spremni za slanje poruka

D:\VII SEMESTAR\industrijski komunik...
Server socket is set to listening mode. Waiti
ng for new connection requests.
New client request accepted. Client address:
127.0.0.1 : 53040
Nova konekcija prihvacena.Soket: 220
New client request accepted. Client address:
127.0.0.1 : 53041
Nova konekcija prihvacena.Soket: 232
Poruka od klijenta (grupa na koju se zeli kon
ektovati): ikp.

Uspesno kreirana grupa!
Uspesno dodat soket!
Duzina liste soketa: 1
```

Svaki put kada klijent pošalje poruku u grupu ona se pošalje svim procesima u toj grupi i ispiše na ekranu.



```
----- DOBRODOSLI U SERVIS ZA GRUPNI CET :D-----
1. PRIKLJUCI SE GRUPI
2. IZADJI

Klijent je spreman za recv!
Unesite ime grupe u kojoj zelite slati poruke:
ikp

-----
1. POSALJI PORUKU
2. IZADJI

>>: povezani ste na trazenu grupu i spremni za slanje poruka

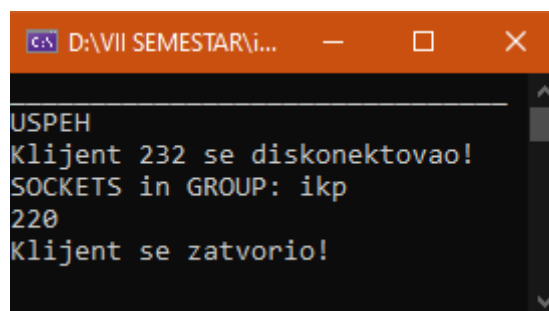
>>: zdravo svima
>>>pozrav

-----
1. POSALJI PORUKU
2. IZADJI

>>: Uspesno ste poslali poruku u grupni chat!

>>: pozrav
```

Klikom na opciju izađi, klijent se briše iz sistema dok ostali procesi te grupe neometano nastavljaju sa razmenom poruka.



```
USPEH
Klijent 232 se diskonektovao!
SOCKETS in GROUP: ikp
220
Klijent se zatvorio!
```

Strukture podataka

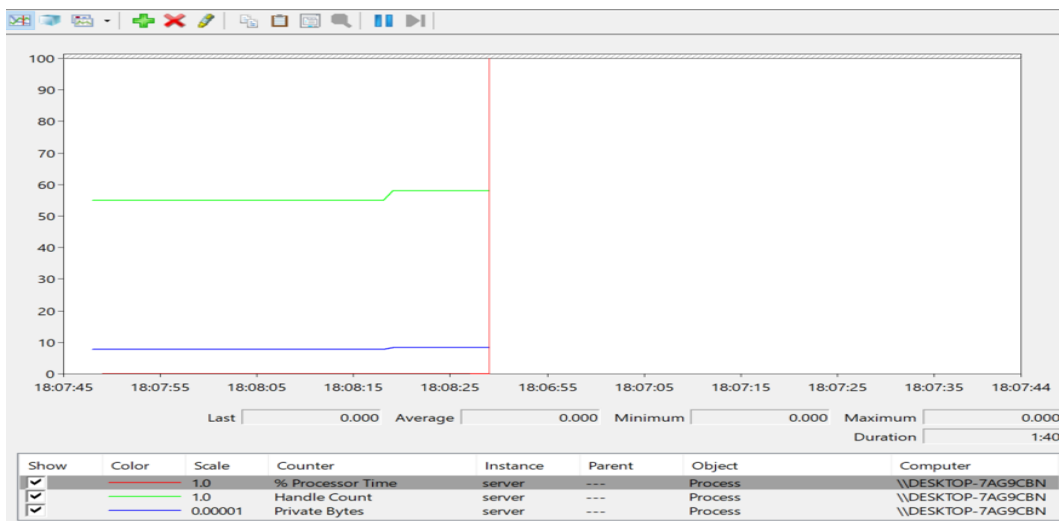
HASH TABELA – sadrži informacije o grupama i njihovim članovima. Pomoću hash funkcije kojoj se prosledi ime grupe, računa index u tabeli na koji će biti smeštena. Na dati index pored imena grupe čuva se i lista socketa klijenata povezanih na tu grupu, kao i red te grupe u koji se smeštaju poruke.

QUEUE – red koji služi za čuvanje poruka. Svaki put kada se doda nova grupa u hash tabelu ujedno se inicijalizuje njen red. Pomoću metode hashtable_getqueue(hashtable* ht, char* group_name) dobavlja se red prosledene grupe nad kojim se mogu vršiti operacije enqueue i dequeue koje služe za smeštanje pristiglih poruka u red i skidanje sa reda kako bi se poslale svim procesima.

LISTE – implementirane su liste koje služe za skladištenje socketa jedne grupe. Sadrže metodu za dodavanje soketa prilikom konekcije, kao i za brisanje soketa iz liste prilikom diskonekcije klijenta. MESSAGE – pomoćna struktura koja služi za skladištenje informacija o komandi koja se šalje serveru i sadržaju unete poruke od strane klijenta.

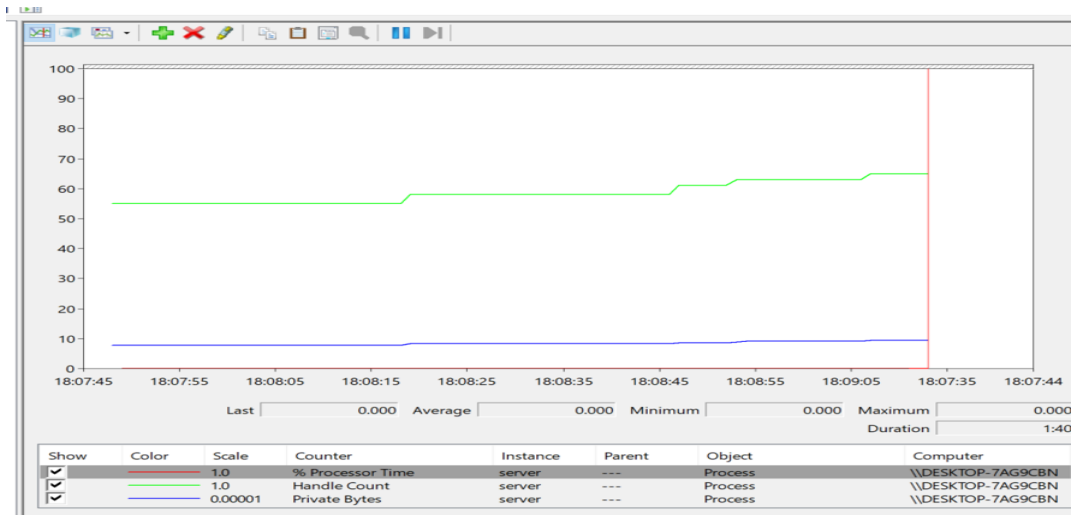
POM – pomoćna struktura u kojoj se čuvaju informacije potrebne za prosleđivanje nitima kao i brojač aktivnih niti.

Testiranje rešenja



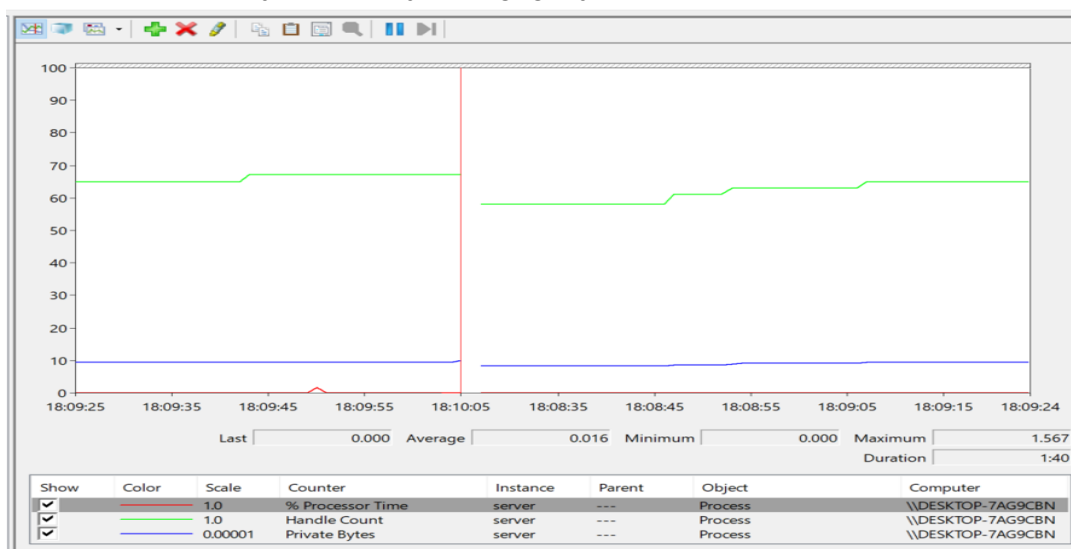
Slika 1

Na slici 1 prikazana je inicijalizacija server komponente i pokretanje jednog klijenta.



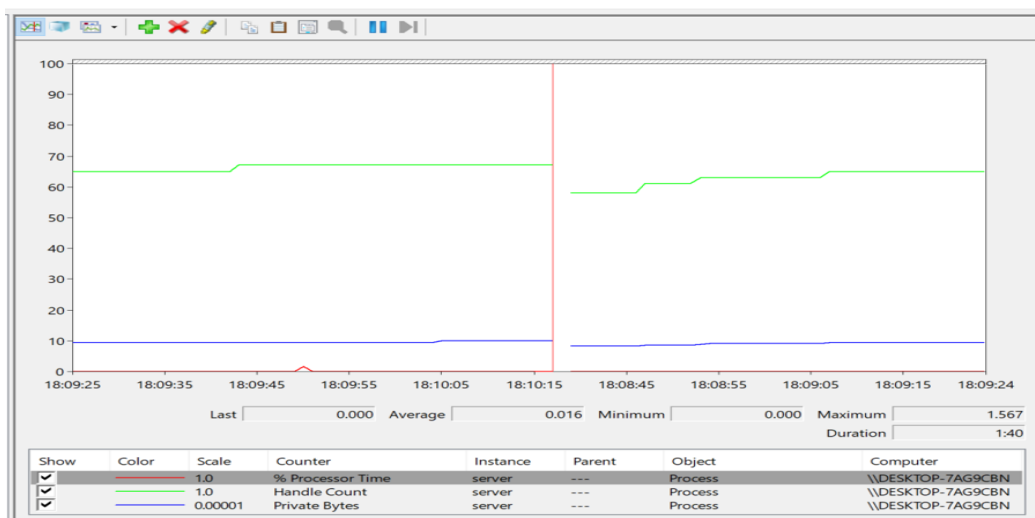
Slika 2

Na slici 2 prikazano je pokretanje i drugog klijenta.



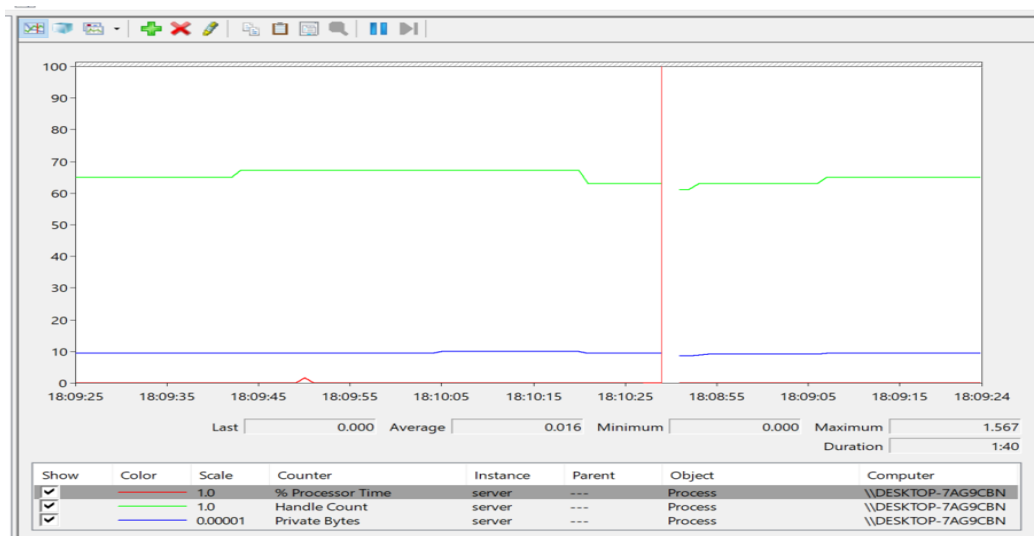
Slika 3

Na slici 3 prikazano je prijavljivanje oba klijenta na istu grupu.



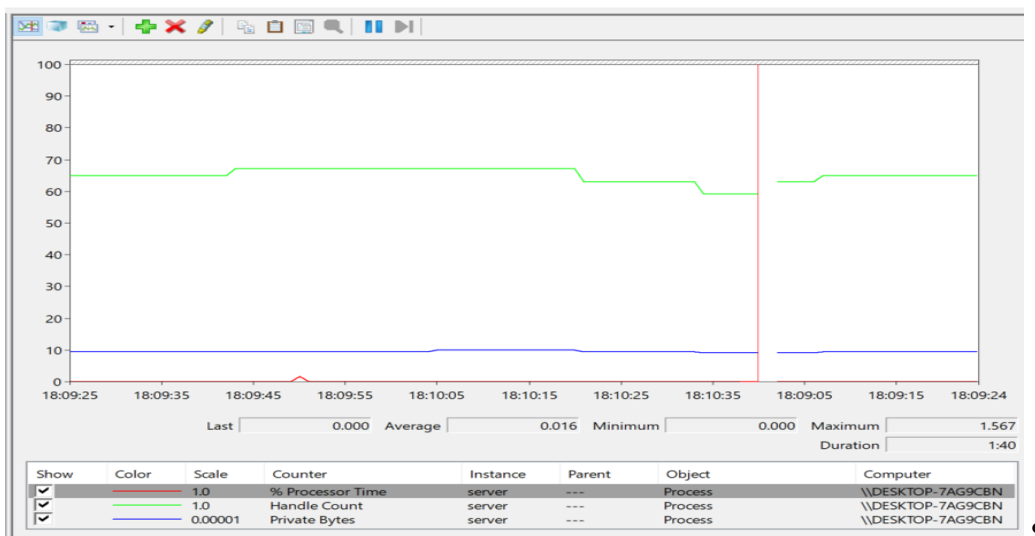
Slika 4

Na slici 4 prikazano je razmenjivanje poruka između 2 klijenta iz iste grupe.



Slika 5

Na slici 5 prikazano je diskonektovanje jednog klijenta.



Slika 6

Na slici 6 prikazano je gašenje i drugog, poslednjeg klijenta.

ID	Time	Allocations (Diff)	Heap Size (Diff)	
1	5.67s	234 (n/a)	82.54 KB (n/a)	
2	26.70s	243 (+9 ↑)	85.80 KB (+3.27 KB ↑)	
3	35.71s	252 (+9 ↑)	89.07 KB (+3.27 KB ↑)	
4	47.65s	260 (+8 ↑)	91.98 KB (+2.91 KB ↑)	
5	57.13s	267 (+7 ↑)	94.82 KB (+2.84 KB ↑)	
6	68.87s	270 (+3 ↑)	95.36 KB (+0.54 KB ↑)	
7	80.78s	273 (+3 ↑)	95.89 KB (+0.54 KB ↑)	
8	88.96s	264 (-9 ↓)	92.91 KB (-2.98 KB ↓)	
9	100.12s	255 (-9 ↓)	89.93 KB (-2.98 KB ↓)	
➤ 10	131.57s	253 (-2 ↓)	89.58 KB (-0.35 KB ↓)	

Slika 7

Slika 7 prikazuje različite trenutke rada programa tokom kod su bile pokrenute dve instance klijenta i jedna instanca servera.

Posmatrajući ID-jeve sa leve strane događaji u tim trenucima su:

1. Inicijalizacija servera
2. Pokretanje prvog klijenta
3. Pokretanje drugog klijenta
4. Konektovanje prvog klijenta na grupu
5. Konektovanje drugog klijenta na grupu
6. Slanje poruke u grupu od strane prvog klijenta
7. Slanje poruke u grupu od strane drugog klijenta
8. Diskonektovanje prvog klijenta
9. Diskonektovanje drugog klijenta

ZAKLJUČAK

Na osnovu rezultata testiranja može se primetiti da se većina zauzete memorije tokom rada programa na kraju oslobađa. Ono što se ne oslobodi pripisuje se tome da nije bilo moguće uraditi snapshot na serveru pre oslobađanja memorije zauzete hash tabelom.

Potencijalna unapređenja

Kreiranje niti je vremenski zahtevna operacija. Kreiranjem Thread Pool-a, niti bi se kreirale pri pokretanju servera i bile bi spremne za obradu klijentskih zahteva. Na taj način bi se izbeglo okupiranje procesorskog vremena i aplikacija bi mogla da opslužuje veći broj Klijenata.

Kod kreiranja hash tabele hash funkcija koliko god složena bila u jednom trenutku dovešće do kolizije kada će namapirati dve različite grupe na isti indeks. Ovo se može izbeći proveravanjem da li postoji grupa na namapiranom indeksu i ako postoji pozivati hash funkciju iznova i iznova dok ne pronađe slobodan indeks. Takođe može se implementirati rešenje pomoću linked lista, gde kada se namapiraju dve grupe na isti indeks pravi se lista koja će sadržati sve grupe na tom indeksu.