|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ** |  |

Јована Михић, Сандра Бугарин и Младен Матић

**(НАСЛОВ РАДА)**

ИСПИТНИ РАД

- Дистрибуирани системи у реалном времену -

Нови Сад, (година)

**Sadržaj**

[1. Zadatak 1](#_Toc32751016)

[2. Analiza i dizajn 2](#_Toc32751017)

[3. Razvoj 6](#_Toc32751018)

[4. Testiranje 7](#_Toc32751019)

[5. Literatura 10](#_Toc32751020)

**Spisak slika**

Slika 1: Topologija iz primera u knjizi 2

Slika 2: Izgled stabla nakon inicijalizacije iz prvog čvora 4

Slika 3: Dijagram sekvence komunikacije unutar topologije prikazane na slici 1. 5

Slika 4: Tabela svih klijenata iz topologije 7

Slika 5: Tabela veza između klijenata 8

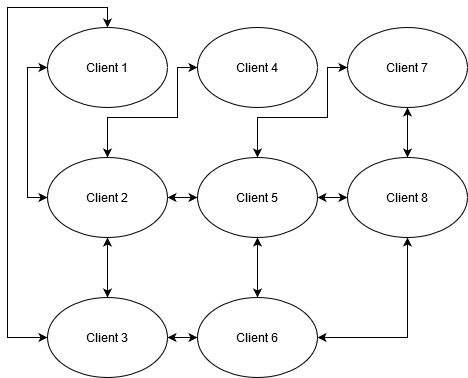
Slika 6: Osam klijenata pokrenutih u osam različitih terminala 9

# Zadatak

Koristeći okruženje za komunikaciju procesora u čvorovima mreže, koje je razvijeno na vežbama, potrebno je realizovati (projektovati, implementirati, istestirati I dokumentovati) izmenjeni algoritam plavljenja za konstruisanje razapinjućeg stabla, koji radi nad proizvoljnom mrežom sa n čvorova.

# Analiza i dizajn

Pre samog početka implemetacije algoritma za konstruisanje razapinjućeg stabla, bilo je potrebno osmisliti sam dizajn algoritma. Krenulo se od osnovnog zadatog testnog slučaja prikazanoj na slici 1.



Slika 1: Topologija iz primera u knjizi [[1]](#footnote-1)

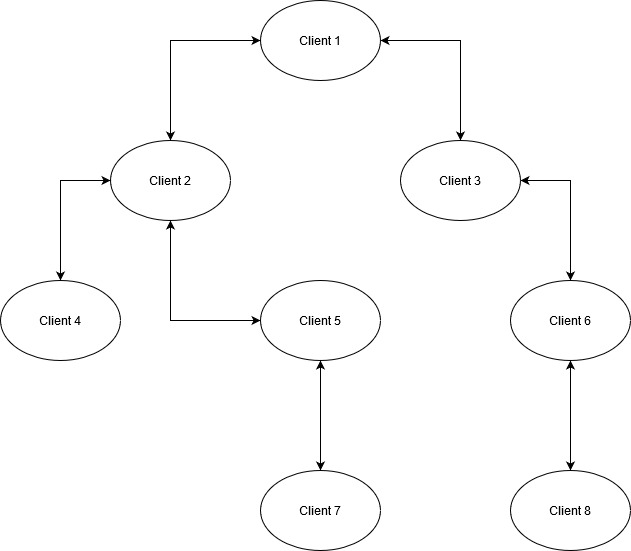
Ideja algoritma je da se krene od jednog čvora od kog će se inicijalizovati stablo. Stablo može inicijalizovati bilo koji čvor, iz topologije, koji prvi pošalje poruku “init\_tree”. On bi poslao poruku svim čvorovima u topologiji sa kojim je povezan. Inicijalno, ni jedan čvor u topologiji nema roditelja. Kada čvoru stigne poruka “init\_tree”, ukoliko on nema roditelja, postaviće pošiljaoca poruke za roditelja i uzvratiti mu poruku “tree\_parent” da bi on znao da inicijalizuje taj čvor za svoje dete. Ukoliko čvor već ima roditelja, po prijemu poruke “init\_tree”, on će pošiljaocu te poruke poslati poruku “tree\_reject”, sa ciljem da on zna zanemariti taj čvor u komunikaciji. To dovodi do rasterećenja mreže jer, pri slanju svima, svaki čvor će primiti poruku samo jednom.

Napravljen je dijagram sekvence, koji bi lakše prikazao slanje poruka između čvorova. Dijagram je prikazan na slici 3. Nakon inicijalizacije stabla iz prvog čvora, ono će izgledati kao na slici 2.

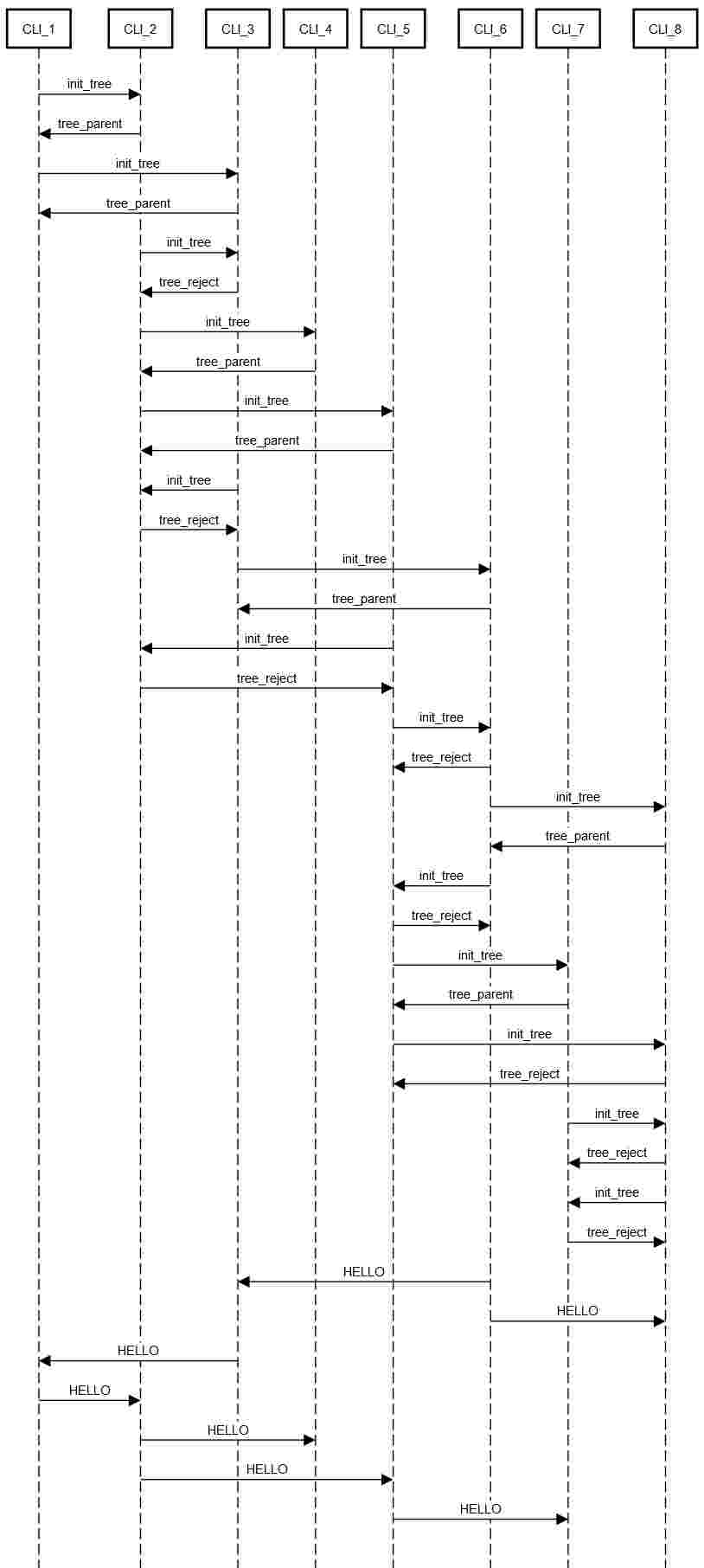
Nakon razrade samog algoritma, bilo je potrebno naći način za inicijalizaciju same topologije. Odlučeno je da će se za to koristiti baza podataka[[2]](#footnote-2). U bazi se nalaze dve tabele. Prva tabela sadrži identifikacioni broj klijenta, njegovu IP adresu i port. Druga tabela sadrzi sve relacije između čvorova. U toj tabeli, polja idu sledećim redosledom:

1. identifikacioni broj čvora
2. IP adresa čvora sa kojim je povezan
3. Port istog tog čvora

Da bi se inicijalizovala topologija, potrebno je bazu izmeniti dodavanjem ili uklanjanjem klijenata iz tabele klijenata, kao i dodavanje ili uklanjanje veza iz tabele relacija.



Slika 2: Izgled stabla nakon inicijalizacije iz prvog čvora



Slika 3: Dijagram sekvence komunikacije unutar topologije prikazane na slici 1.

# Razvoj

Razvoj samog programa podrazumevao je izmenu primera sa vezbi na način da može slati i primati poruke bez čekanja. Inicijalizacija stabla implementirana je slanjem poruka u tekstualnom formatu, kojima je, kao prvi karakter, dodat identivikacioni broj pošiljaoca. Time program zna da poruku prosleđuje svom roditelju i svoj deci, isključujući izvor te poruke. Na taj način, izbegnuta je beskonačna petlja slanja poruka, kao i redundantno slanje više poruka jednom čvoru.

Funkcije za slanje i primanje poruka pokrenute su u posebnim procesima. Za same poruke korišćen je red[[3]](#footnote-3) prosleđen kao argument pri kreiranju procesa. Kako bi se informacije o roditelju i deci videli između procesa za slanje i primanje poruka, korišcen je niz menadžer objekta.

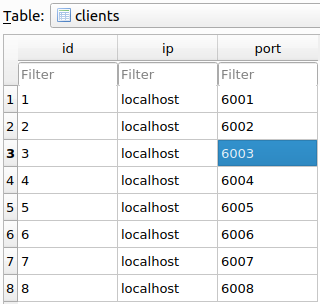
# Testiranje

Testiranje je vršeno na primeru topologije prikazane na slici 1. Isprobano je više različitih čvorova kao osnovni čvor, kao i zvezdasta topologija sa četiri čvora.

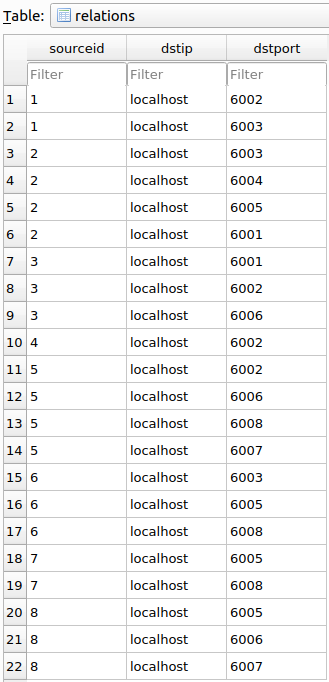
Skripte je potrebno pokrenuti u odvojenim terminalima sa jednim argumentom koji nosi identifikacioni broj klijenta. Trenutno podržani broj klijenata je deset, identifikacioni brojevi od nula do devet. U postavci baze u trenutku pisanja, upisana je topologija sa slike 1, gde su klijenti sa identifikacionim brojevima od jedan do osam. Primer pokretanja prvog klijenta:

$python3 example\_complete\_graph.py 1

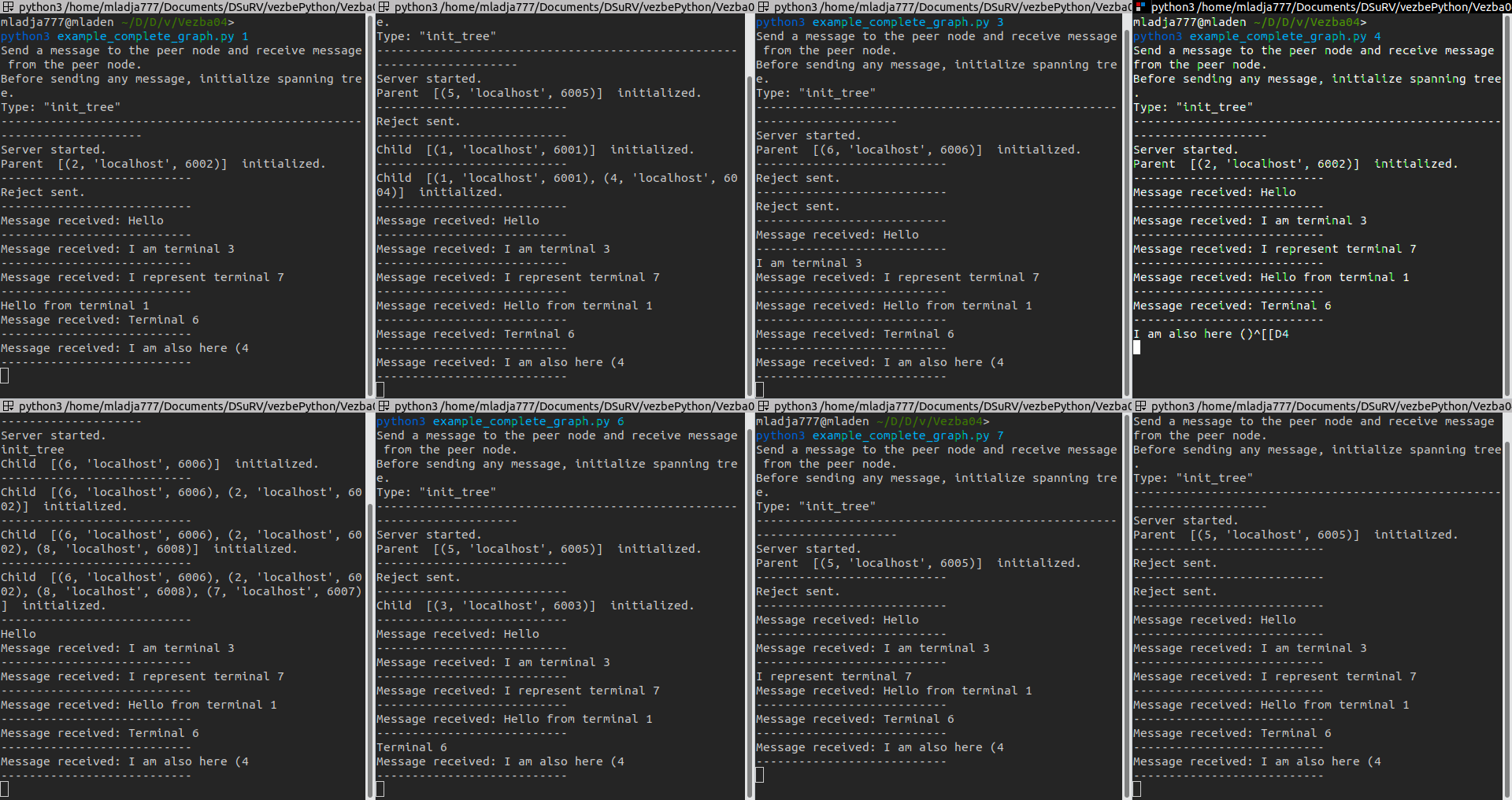
Nakon napravljene topologije mreže sa primera iz knjige, pokrenuto je, u osam različitih terminala, osam klijenata. Testirano je inicijalizovanje sa svih čvorova, kao i slanje poruka sa bilo kog čvora. Primer se može videti na slici 6.



Slika : Tabela svih klijenata iz topologije



Slika : Tabela veza između klijenata



Slika : Osam klijenata pokrenutih u osam različitih terminala

# Literatura

1. Osnove distribuiranih sistema u realnom vremenu, Miroslav Popović

1. Slika 2.5, strana 15, Osnove distribuiranih sistema u realnom vremenu, Miroslav Popović [↑](#footnote-ref-1)
2. SQLite [↑](#footnote-ref-2)
3. Multiprocessing queue [↑](#footnote-ref-3)