

Snapshots

Domaći 2

1 Pregled zadatka

Zadatak je u velikoj meri zasnovan na primerima koji su rađeni na vežbi 8. Preporučuje se temeljno izučavanje tih primera pre čitanja teksta zadatka.

Implementirati distribuiran sistem koji podržava sledeće funkcionalnosti:

- Implementiran u Java ili go programskom jeziku. Dozvoljena je upotreba pomoćnih biblioteka za komunikaciju između čvorova.
- Potpuno asinhrona ne-FIFO komunikacija.
- Proizvoljan broj čvorova koji su povezani na proizvoljan način (graf nije kompletan). Ako bilo koji čvor bilo kada ima potrebu da pošalje neku poruku svim čvorovima u sistemu, to mora da funkcioniše isključivo preko veza koje su date u konfiguraciji sistema.
- Svaki čvor ima svoj port na kojem prihvata poruke od svojih suseda, i svi slušaju na localhost.
- Radi se jedan snapshot u jednom trenutku, na nivou čitavog sistema. Sistem treba da podrži izradu novog snapshota na proizvoljnom čvoru, nakon što je prethodni snapshot završen.
- Komunikacija sa korisnikom preko CLI ili tekstualnih (skript) datoteka.

Dozvoljeno je koristiti projekat sa vežbi kao polaznu tačku za izradu domaćeg zadatka, kao i implementirati čitav sistem od nule, dok god zadovoljava sve funkcionalne i nefunkcionalne zahteve.

Funkcionalni zahtevi za sistem su opisani u odeljku 2.

Nefunkcionalni zahtevi za sistem su opisani u odeljku 3.

Bodovanje zadatka, kao i instrukcije za predaju zadatka su dati u odeljku 4.

2 Funkcionalni zahtevi

Svaki čvor u sistemu počinje sa unapred određenom količinom bitcake-ova, kao u primerima sa vežbe 8.

Neophodno je da bude moguće napraviti snapshot sistema sa proizvoljnog čvora u sistemu. Strategija za pravljenje snapshota treba da bude po jednom od dva algoritma za snimanje stanja u sistemima sa kauzalnom isporukom koji su opisani u knjizi, i to su:

1. Acharya-Badrinath algoritam.
2. Alagar-Venkatesan algoritam.

Sistem treba da podrži rad oba ova algoritma, a koji se koristi u konkretnom pokretanju treba da se bira na osnovu podešavanja u konfiguracionoj datoteci. Nove vrednosti za "snapshot" atribut treba da budu: ab (Acharya-Badrinath) i av (Alagar-Venkatesan).

Svi čvorovi, će na osnovu komandi od korisnika da veoma često razmenjuju svoje bitcake zalihe. Rezultat snapshot algoritma treba da bude trenutno bitcake stanje u sistemu, kao što je bio slučaj na vežbi 8 kod Lai-Yang algoritma. Kod Alagar-Venkatesan algoritma nema potrebe prikazati stanje u kanalima na čvoru koji je inicirao snapshot algoritam, nego je dovoljno da svaki čvor ispiše stanje za svoje kanale.

Dozvoljeno je da se svi čvorovi startuju na istoj mašini i slušaju na različitim portovima na localhost. Pritom je neophodno da postoji veštački uvedeno nasumično kašnjenje pri slanju svake poruke, kako bi se simuliralo kašnjenje na mreži.

Dva čvora smeju da razmenjuju poruke samo ako su navedeni kao susedi u konfiguracionoj datoteci.

Neophodno je da sistem podrži "skriptovano" pokretanje više čvorova, gde se komande za svaki čvor čitaju iz tekstualne datoteke, i izlazi za svaki čvor se pišu u zasebnim datotekama.

3 Nefunkcionalni zahtevi

Neophodno je imati podršku za sledeće situacije, sa navedenim rešenjima. Probleme koji nisu navedeni u tabeli nema potrebe rešavati.

R. Br.	Problem	Rešenje
1.	Korisnik traži izradu snapshota na čvoru koji je već započeo izradu snapshota, ali taj snapshot nije završen.	Prijaviti grešku na konzoli i nastaviti normalno sa radom.
2.	NAPOMENA: ako korisnik započne izradu snapshota na više čvorova konkurentno, dozvoljeno je da se sistem ponaša nepredvidivo.	Ne rešava se.

Neophodno je da broj poruka kod ovih algoritama bude isti kao u opisu algoritma u udžbeniku - $4n$ za Acharya-Badrinath i $3n$ za Alagar-Venkatesan.

Pomoću konfiguracione datoteke se navodi:

- Koliko čvorova ima u sistemu.
- Port na kojem svaki čvor sluša.
- Lista suseda za svaki čvor.

Na svim čvorovima treba da postoje poruke ispisa koje daju informacije o tome kako teče komunikacija, kako bi posmatranjem ispisa mogla da se rekonstruiše istorija rada sistema.

4 Predaja zadatka

4.1 Način predaje zadatka

Zadatak se predaje putem mail-a na bmilojkovic@raf.rs. Java projekat imenovati na sledeći način: "kids_d2_ime_prezime_ind". Npr. "kids_d2_student_studentic_rn0101".

Arhivirati ovaj direktorijum (.zip), okačiti na svoj drive i u mail-u poslati link ka arhivi, pošto će Google mail verovatno blokirati direktan attachment.

U tekstu mail-a obavezno navesti:

- Ime i prezime
- Broj indeksa
- Grupa, po zvaničnom spisku
- Programski jezik (java / go) i razvojno okruženje (Eclipse / Intellij)

Subject mail-a mora da bude u obliku: "[KiDS] D2 ime_prezime_ind".

Npr. "[KiDS] D2 student_studentic_rn0101"

Naziv arhive mora da bude u obliku: "kids_d2_ime_prezime_ind.zip"

Npr. "kids_d2_student_studentic_rn0101.zip"

Rok za predaju je:

- Petak, 14. maj 23:59:59 za grupu koja sluša KiDS petkom
- Ponedeljak, 17. maj 23:59:59 za grupe koje slušaju KiDS ponedeljkom
- Utorak, 18. maj 23:59:59 za grupu koja sluša KiDS utorkom
- Utorak, 18. maj 23:59:59 za ponovce

Rok je definisan po grupi kojoj student zvanično pripada, tj. na čijem spisku se nalazi.

4.2 Odbrana i bodovanje

Odbrana domaćih zadataka je obavezna. Termin za odbranu drugog domaćeg zadatka će biti naknadno objavljen. Odbrane će se vršiti individualno i preko Zoom. Ako ste iz bilo kog razloga sprečeni da prisustvujete odbrani, obavezno to najavite što pre, kako bismo mogli da zakažemo vanredni termin za odbranu.

Svrha odbrane je da se pokaže autentičnost zadatka. Ovo podrazumeva odgovaranje na pitanja u vezi načina izrade zadatka, ili izvršavanje neke izmene nad zadatkom na licu mesta. U slučaju da odbrana nije uspešna, dodeljuje se -15 poena na domaćem zadatku.

Zadatak se boduje na sledeći način:

- | | | |
|--|---|---------|
| ● Snimanje lokalnog stanja | - | 7 poena |
| ● Snimanje kanala po Acharya-Badrinath | - | 4 poena |
| ● Snimanje kanala po Alagar-Venkatesan | - | 4 poena |

Zadatak je moguće raditi parcijalno, ali obavezno je da se kompajluje i da može da se pokrene, kao i da je moguće pokazati da implementirana stavka ispunjava funkcionalne i nefunkcionalne zahteve.