Diplomski studij

Informacijska i komunikacijska tehnologija
Telekomunikacije i informatika

Računarstvo
Računarska znanost
Programsko inženjerstvo i informacijski sustavi

Raspodijeljeni sustavi

Pitanja za provjeru znanja
2. blok predavanja

Ak.g. 2008./2009.

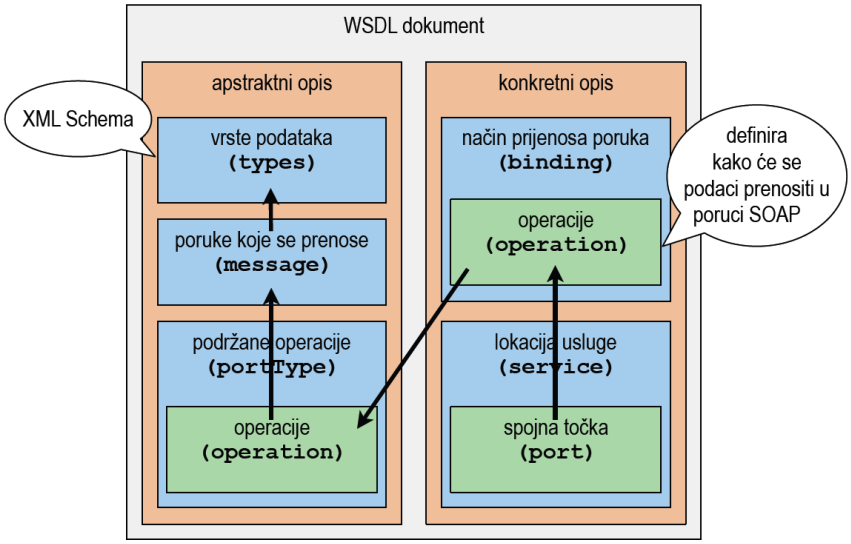
Napomena: *Preporučena literatura su bilješke s predavanja.*

Zadatak 1 Navedite dva osnovna načina rada protokola SOAP i objasnite kako se poruka SOAP šalje pomoću protokola HTTP.

Dva osnovna načina rada koje podržava protokol SOAP su poziv udaljene procedure te razmjena dokumenata i poruka. Poruka SOAP, koja je pisana jezikom XML, se sastoji od zaglavlja i tijela. Prilikom slanja poruke SOAP protokolom HTTP, i zaglavlje i tijelo poruke SOAP se nalaze u tijelu poruke HTTP.

Zadatak 2 Objasnite i skicirajte sadržaj apstraktnog i konkretnog opisa u strukturi dokumenta WSDL.

Dokument WSDL se sastoji od apstraktnog i konkretnog opisa kao što se vidi na slici.



Apstraktan opis u WSDL-u se sastoji od 4 elementa:

1. *types*: definira vrste podataka neovisne o platformi i jeziku (koristi se XML Schema),
2. *message*: definiraju ulazne i izlazne poruke koje se mogu koristiti kao parametri usluge,
3. *operation*: predstavlja jednu operaciju/metodu/proceduru koja je definirana u usluzi, a sastoji se od definicija ulaznih, izlaznih i iznimnih poruka koje se mogu razmjenjivati korištenjem ove operacije i
4. *portType*: koristi poruke (pod 2) da bi opisao sve operacije koje pruža usluga.

Konkretni opis se sastoji od 2 dijela:

1. *binding*: definira kako je konkretna implementacija povezana s operacijama u apstraktnom opisu i definira fromat u kojem će se poruke prenositi (protokol i elemente) i
2. *service*: definira URI gdje je usluga isporučena tj. na kojoj adresi se može pozvati usluga (taj URI je definiran u spojnoj točci).

Zadatak 3 Objasnite svojstvo slabe povezanosti usluga kod uslužno orijentirane arhitekture.

Svojstvo slabe povezanosti usluga kod uslužno orijentirane arhitekture (SOA – Service Oriented Architecture) odnosi se na dizajn programske izvedbe usluga. U sustavu SOA, usluge trebaju biti izvedene tako da promjena u jednoj usluzi ne zahtijeva promjenu neke druge usluge. Pri tome, svaka usluga može i dalje nesmetano koristiti neku drugu uslugu. Npr. algoritam koji se koristi u usluzi se može promijeniti bez znanja drugih usluga i druge usluge ju mogu nesmetano koristiti. Bitno je da se sučelja opisana WSDL-om ne promijene.

Zadatak 4 Definirajte što su to nestrukturirana imena i navedite barem 2 načina pronalaženja pristupnih točaka pomoću ovakvih imena.

Nestrukturirana imena su najčešće slučajan niz bitova. Ona sama ne sadrže informacije o tome kako pronaći pristupnu točku pa se zbog toga koriste sljedeći načini pronalaženja pristupnih točaka s nestrukturiranim imenima[1]:

1. jednostavni načini (samo u lokalnim mrežama)
    a. opće razašiljanje (*broadcast*) i
    b. višeodredišno razašiljanje (*multicast*),
2. pokazivači prema naprijed (*forwarding pointers*),
3. domaća lokacija (*home location*),
4. raspodijeljene *hash* tablice (*distributed hash tables*) i
5. hijerarhijski pristup.

Zadatak 5 Objasnite što je to razlučivanje imena na primjeru strukturnog imenovanja.

Razlučivanje imena je proces pronalaženja čvora iz zadanog puta. Primjer strukturnog imenovanja je datotečni sustav koji je zapravo usmjereni graf kod kojega se do svakog čvora može doći preko puta koji je predstavljen nizom oznaka odijeljenih specijalnim znakom („\“ ili „/“). Ovisno o početnom čvoru put može biti apsolutni (počinje specijalnim znakom) ili relativni (ne počinje specijalnim znakom). U datotečnom sustavu postoje alternativna imena (*aliasi*) tako da se do nekih čvorova može doći preko 2 ili više različitih puteva.

Zadatak 6 Objasnite što je replika podatka, a što je nekonzistentnost replike podatka.

Replika podatka je jedna kopija podatkovnog objekta u raspodijeljenoj okolini. Nekonzistentnost replika podataka se javlja kada dvije ili više replika u raspodijeljenoj okolini u nekom trenutku u vremenu se nalaze u različitim stanjima.

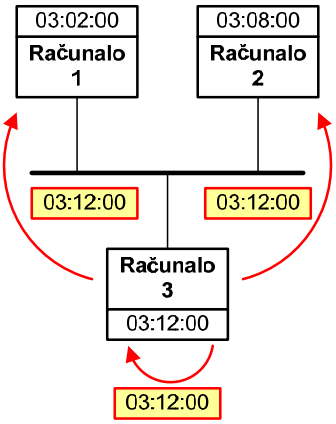
Zadatak 7 Navedite i opišite značajke tri osnovna razreda replika podataka u raspodijeljenim sustavima.

U raspodijeljenim sustavima koriste se sljedeća tri razreda replika:

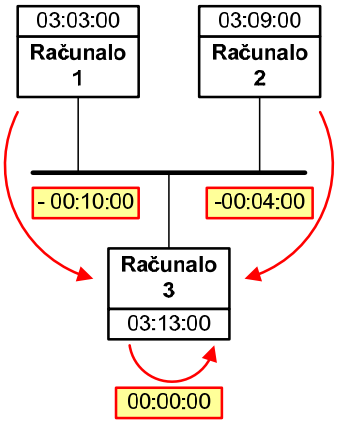
1. **Trajne replike:** Osnovni primjerci podataka koji su trajno postavljeni na poslužiteljima. Njihove postavke su statičke i upravljane od strane vlasnika podataka. Nad njima se najčešće ostvaruju operacije čitana podataka i poslužuju primjenom grozdova poslužitelja i replika poslužitelja.
2. **Poslužiteljske replike:** Poslužitelji podataka stvaraju nove replike trajnih replika i raspoređuju ih na dostupne poslužitelje u mreži. Odabir i raspoređivanje replika ostvaruje se u stvarnom vremenu tijekom rada poslužitelja.
3. **Korisničke replike: P**ostupak repliciranja iniciran je odstrane korisnika putem korisničkih programa. Korisnički programi u lokalne spremnike spremaju pribavljene podatke. Potrebno je održavati konzistentnost lokalnog spremnika s poslužitelje od kojeg su podatci dohvaćeni.

[1] U odgovoru je potrebno navesti samo dva od pet mogućih načina pronalaženja pristupnih točaka s nestrukturiranim imenima.

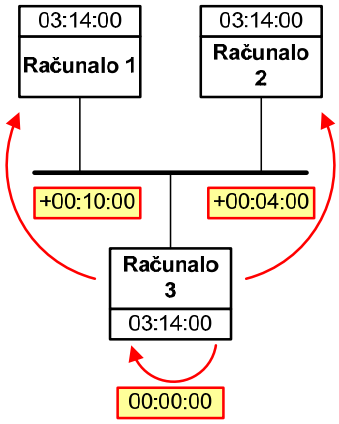
Zadatak 8 Prikažite i objasnite korake algoritma Berkeley za usklađivanje satnih mehanizama tri računala u raspodijeljenoj okolini. Računala imaju sljedeće vrijednosti satova T1=03:02:00, T2=03:08:00 i T3=03:12:00. Upravitelj je treće računalo. Pretpostavite da prijenos poruke između 2 računala traje 1 minutu.



Upravitelj šalje poruku s trenutnim vremenom svim računalima.

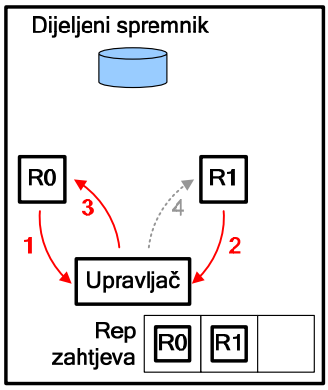


Poslane poruke putuju 1 minutu i nakon primitka poruka, računala odgovaraju s porukom koja sadrži razliku lokalnog vremena u odnosu na primljeno vrijeme.

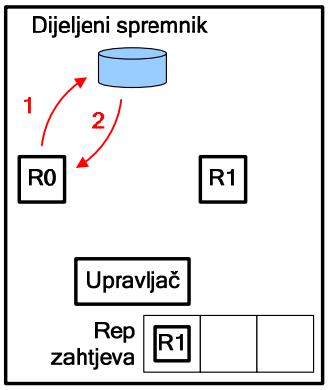


Nakon primitka poruka odgovora, upravitelj šalje poruke zahtjeva koje sadrže vremenski pomak za svako računalo. Poruke zahtjeva putuju 1 minutu te nakon primitka poruke zahtjeva, svako računalo usklađuje lokalni satni mehanizam.

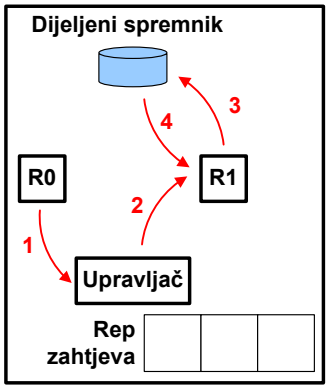
Zadatak 9 Opišite postupak međusobnog isključivanja dvaju procesa (R0 i R1) primjenom središnjeg upravljača s repom čekanja tako da nacrtate redoslijed operacija i objasnite ih. Nakon zauzimanja dijeljenog spremnika, proces provodi jednu operaciju čitanja ili pisanja nad dijeljenim spremnikom.



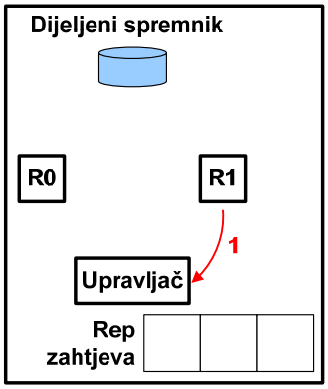
1 – R0 šalje zahtjev za zauzimanje sredstva, zahtjev se sprema u rep
2 – R1 šalje zahtjev za zauzimanje sredstva, zahtjev se stavlja u rep
3 – Kako je zahtjev od R0 stigao prije, upravljač šalje potvrdu R0 i uklanja njegov zahtjev iz repa



1 – R0 provodi operaciju pisanja
2 – R0 prima potvrdu

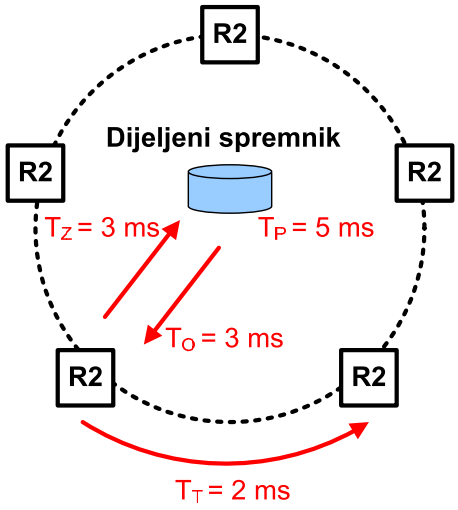


1 – R0 šalje poruku Upravljaču i otpušta pristup
2 – Upravljač šalje poruku dojave R1 te mu dodjeljuje pristup dijeljenom spremniku. Iz repa zahtjeva uklanja se zahtjev od R1
3 – R1 provodi operaciju pisanja
4 – R1 prima potvrdu



1 – R1 šalje poruku Upravljaču i otpušta pristup

Zadatak 10 Pet procesa postavljenih na različita računala u raspodijeljenoj okolini ostvaruje međusobno isključivanje primjenom prstena. Vrijeme prijenosa poruke zahtjeva i odgovora pri pristupu dijeljenom sredstvu jednako je 3 ms, vrijeme obrade poruke zahtjeva na sredstvu je 5 ms, vrijeme prijenosa *tokena* između dva susjedna procesa u prstenu je 2 ms. Kada primi *token*, proces može maksimalno jednom ostvariti pristup dijeljenom sredstvu prije nego što proslijedi *token* idućem susjedu. Prikažite strukturu prstena i naznačite navedena vremena na slici. Koje je minimalno, a koje maksimalno vrijeme čekanja bilo kojeg procesa u prstenu za pristup dijeljenom sredstvu.



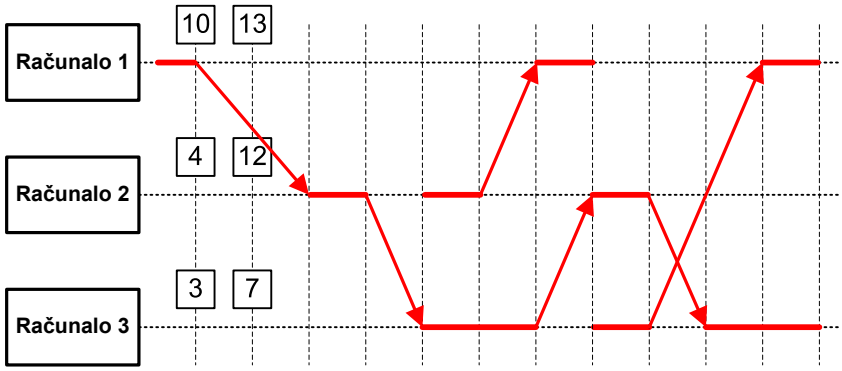
1. Minimalno vrijeme čekanja na pristup

U najboljem slučaju, proces koji želi ostvariti pristup čeka T=0 sekundi. Naime, taj slučaj nastupa kada proces uđe u stanje u kojem želi ostvariti pristup sredstvu netom prije nego što je primio token.

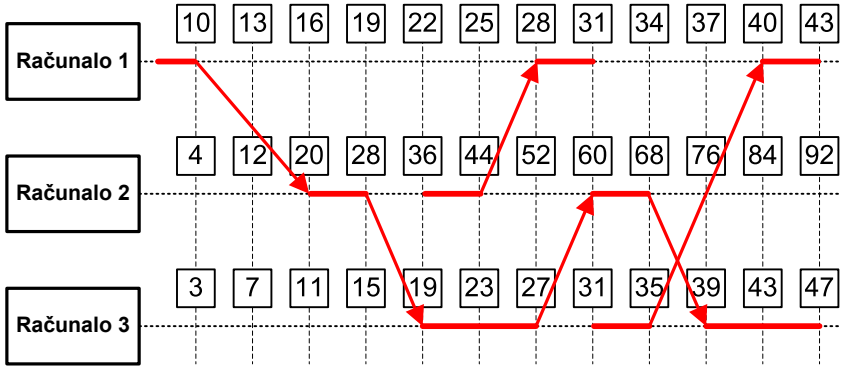
2. Maksimalno vrijeme čekanja na pristup

U najgorem slučaju, proces ulazi u stanje u kojem želi ostvariti pristup sredstvu netom nakon što je proslijedio token svojem susjedu. U tom slučaju, proces mora čekati da svi ostali procesi prime token i ostvare pristup dijeljenom sredstvu. Maksimalno vrijeme čekanja u tom slučaju iznosi $T = 5 * T_T + 4 * (T_Z + T_O + T_P) = 10 + 44 = 54$ ms.

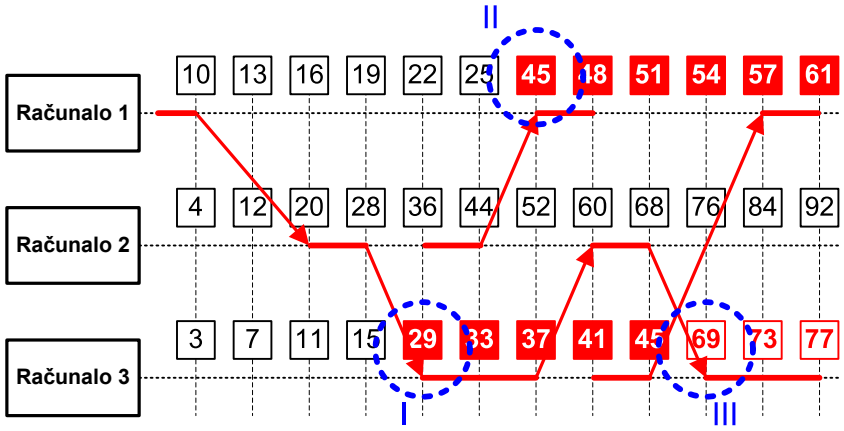
Zadatak 11 Za slijed razmjene poruka između tri računala prikazan na slici uspostavite globalni tijek vremena primjenom skalarnih oznaka logičkog vremena. Navedite i opišite trenutke u kojima se ostvaruje korekcija lokalnih satnih mehanizama.



1. Određivanje vrijednosti satnih mehanizama u trenutcima razmjene poruka



2. Primjena skalarnih oznaka vremena



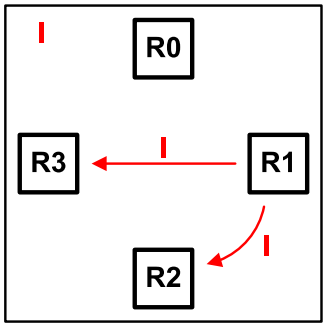
Trenutak I: Računalo 3 prima poruku od računala 2 s oznakom vremena T_P=28 koja je veća od lokalne oznake vremena T_L= 19. Lokalni sat se pomiče na vrijednost **T_P+1=29**.

Trenutak II: Računalo 1 prima poruku od računala 2 s oznakom vremena T_P=44 koja je veća od lokalne oznake vremena T_L= 28. Lokalni sat se pomiče na vrijednost **T_P+1=45**.

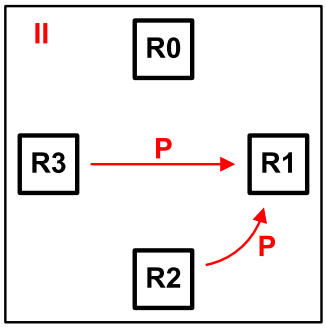
Trenutak III: Računalo 3 prima poruku od računala 2 s oznakom vremena T_P=68 koja je veća od lokalne

oznake vremena T_L= 49. Lokalni sat se pomiče na vrijednost **T_P+1=69**.

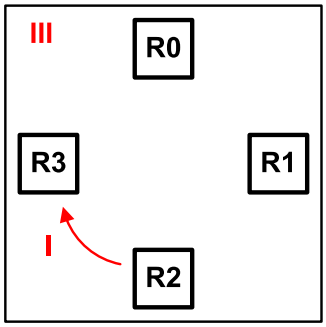
Zadatak 12 Prikažite postupak određivanja upravitelja između četiri računala (R_0, R_1, R_2, R_3) u raspodijeljenoj okolini. Postupak odlučivanja započinje računalo R_1.



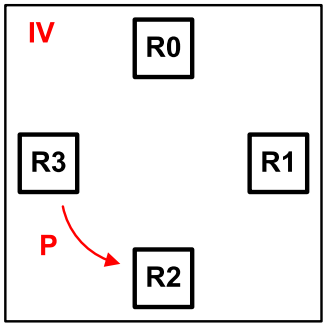
Postupak izbora započinje računalo R_1 slanjem poruke *izbor* (I) svim računalima većeg rednog broja od računala R_1. Poruka izbora poslana je računalima R_2 i R_3.



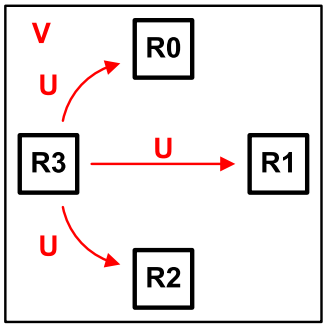
Računala R_2 i R_3 primaju poruku izbora. Obzirom da su oba računala aktivna, oba računala šalju poruku *potvrda* (P) računalu R_1.



Obzirom da je računalo R_3 računalo s najvećim rednim brojem, računalo R_3 niti jednom računalu ne šalje poruku *izbor*. Međutim, računalo R_1 nije računalo s najvećim rednim brojem te stoga šalje poruku *izbor* (I) računalu R_3.



Računalo R_3 prima poruku *izbor* (I) od računala R_2 i šalje poruku *potvrda* (P) računalu R_2.



Obzirom da je računalo R_3 računalo s najvećim rednim brojem, računalo R_3 postaje novi upravitelj te šalje poruku *upravitelj* (U) svim računalima (R_0, R_1, R_2).