10. Komunikacija između procesa

1. Poruka se ne šalje izravno procesu nego:

U red poruka

1. Što se dodaje na IP razini?

adresa odredišta i izvorišta

1. Ako redom poruka glumimo semafor onda funkciji čekaj\_semafor odgovora:

uzimanje\_iz\_reda

(*btw,* postavi\_semafor = stavljanje\_u\_red)

|  |  |
| --- | --- |
| čekaj\_semafor | uzimanje\_iz\_reda |
| postavi\_semafor | stavljanje\_u\_red |

1. Prema pravilima gl. logičkog sata proces povećava svoj log. sat kada?

/\* između svaka dva događaja \*/

1. Izmedu svaka svoja 2 karakteristicna dogadaja
2. Po primitku poruke

1. Kako se osvježava vrijednost lokalnog log. sata? (Cj=max...)

Cj = max { Cj(prijašnji),Tm}+1 gdje je Tm Ci od procesa koji je poslao zahtjev

1. Koliko se poruka razmjeni u Ricart-Agrawala protokolu ako je N čvorova?

2\*(N-1) (ako bude pitanje koliko točno kojih, N-1 ZAHTJEVA I N-1 ODGOVORA)

1. Koliko poruka u Lamportovom protokolu, 10 čvorova.

3\*(N-1) = 27 (N-1 ZAHTJEVA, N-1 ODGOVORA i N-1 IZLAZAKA)

1. Čitanje cjevovoda je za razliku od čitanja datoteke:

destruktivno

1. Okolina (u kontekstu var. okoline) je:

Niz nizova znakova oblika "ime=vrijednost" koji se predaje svakom programu prilikom pokretanja (samo niz znakova = jedan *entry* u okolini)

1. Neimenovan cjevovod ima koliko opisnika:

Dva

Koliko ima imenovani?

Jedan, pise dolje

1. Ako je vrijeme 712, a proces primi poruku 718, onda je nova vrijednost sata: (brojeve napamet bacam)

C = max {712,718}+1=719

12. a) Lamport, kada proces *i* želi ući u KO, onda svima šalje i stavlja u svoj red:

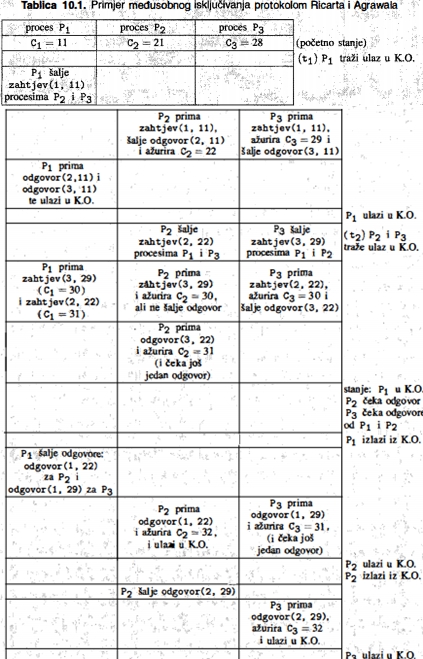
stavlja u svoj red zahtjev(i,T(i))

b) Kada proces Pj primi tu poruku onda napravi:

-namjesti svoj sat prema pravilima GLS-a

-u svoj red čekanja stavlja zahtjev

-šalje odgovor (i,T(j)) procesu Pi gdje je T(j) = novi Cj

13. Ricart-Agrawala, zadano C1,C2,C3, P1 želi ući, P2 i P3 žele ući nakon što P1 uđe, koje se poruke razmijenjuju i stanja logičkih satova po ulasku u KO

Knjiga, 10.3., str 275.

14. U postupku povezivanja cjevovoda na standardni ulaz treba zatvoriti standardni ulaz ili izlaz?

Standardni ulaz

15. Prema pravilima globalnog logičkog sata kada proces šalje poruku on uz nju pridodaje

svoj ID procesa i vrijednost lokalnog logičkog sata

zar nije samo vremensku oznaku Tm = Ci? -> nije samo Tm jer je moguće da je Tm isti onda se rješava tako da prioritet ima onaj sa manjim ID-om

16. Na koju će vrijednost postaviti lokalni logički sat proces ako proces primi poruku s vremenskom oznakom 178, a vrijednost njegovog logičkog lokalnog sata je 177?

max(178, 177) + 1 = 179

17. Navesti broj poruka koji treba razmijeniti u sustavu s 10 računala gdje se na svakom računalu nalazi po jedan proces koji se natječe za ulazak u kritični odsječak ako se koristi raspodijeljeni Lamportov protokol:

● 3 \* (N - 1) -> 3 \* 9 = 27 -> ovo bi bilo ako samo jedan pokusava ući

● 3 \* (N-1) \* 10 -> mora ici puta 10 samo -> 270 , tj treba pomnožiti s brojem procesa koji žele ući u KO (neovisno o broju računala?) --> ovisno o broju računala, tj. Zapravo ovisno o ukupnom broju procesa u tom zadatku(10 racunala \* 1 proces = 10)

18. Koje informacije u Internet protokolu se dodaju bloku podataka?

izvorišnu i odredišnu IP adresu

19. Navedite tri načina na koje dretve različitih procesa mogu komunicirati.

● razmjenom poruka

● datotekom

● cjevovodi

● uz pomoć dijeljenog spremnika

● uz pomoć varijabli okoline

Ukoliko se radi o raspodjeljenim sustavima, onda samo razmjenom poruka (drugi mehanizmi se oslanjaju na poruke). To su RPC, raspodjeljeni adresni prostor

20. Okolina (environment) je?

Niz nizova znakova oblika “ime=vrijednost” (bez navodnika) koji se predaje svakom programu prilikom pokretanja

21. U postupku povezivanja dva procesa uz pomoć reda poruka okolina može poslužiti za razmjenu \_\_\_\_\_ .

identifikatora reda poruka.

22. Nedostaci običnog cjevovoda: \_\_\_\_

● procesi moraju biti u srodstvu

● ne može se stvoriti nakon što su stvoreni (child?) procesi (misli se na random stvorene, koji nisu u srodstvu)

● Običan cjevovod ima ograničenu veličinu međuspremnika na 10 stranica (cca 40kB, 1 stranica je 4kB)

22b. Prednosti imenovanog cjevovoda:

* Rad s imenovanim cjevovodom odgovara radu s datotekama
* Procesi koji komuniciraju preko imenovanog ne moraju biti u srodstvu
* Nema ograničenja na duljinu (kod običnog je 10 stranica)

23. S koliko je opisnika opisan IMENOVAN cjevovod?

Jednim

24. Lamportov protokol, procesi P1, P2 i P3 uz početne vrijednosti C1=12, C2=7, C3=20. P1 i P2 istovremeno šalju zahtjeve, odredite logičke satove i redove zahtjeva u trenutku kada P1 uđe u kritični odsječak.

<https://www.fer.unizg.hr/_download/repository/NOS_aud_zad_10.1.pdf> -> novi link

- *da li netko može objasniti zašto se u primjeru mijenja vrijednost satova C2,C3 nakon poruke izlazak?* Zato što dobijaju od procesa P1 poruku Izlazak() a kad dobiju bilo kakvu poruku, ažuriraju svoj lokalni sat

-to ne pise u algoritmu(iako je napomenuto negdje iznad) x

25. Kako se zove mehanizam unutar jednog računala koji broji događaje?

lokalni logički sat

26. Prema pravilima globalnog logičkog sata na koju vrijednost proces Pi postavlja lokalni logički sat kada primi poruku zahtjev(j,T(j))?

Ci = max (Ci, T(j)) + 1

27. Navesti ispravan redoslijed koraka ako želimo napraviti cjevovod tako da roditelj čita, a dijete piše u cjevovod:

1. U djetetu zatvoriti kraj cjevovoda za čitanje
2. U procesu roditelja pozvati stvaranje djeteta
3. Stvoriti cjevovod
4. U roditelju zatvoriti kraj cjevovoda za pisanje

C->B->D->A

//moze li C->B->A->D ? - mislim da može, taj poredak je u pripremi za lab

28. Isto s cjevovodima, ali je bilo s funkcijama, kojim redoslijedom treba nešto napraviti, i koji korak se može preskočiti da bi cjevovod i dalje bio ispravan:

Ovakve slične funkcije su bile:

1. close(1);

2. dup( cj[1] );

3. close( cj[1] );

4. close( cj[0] );

Ovo je vjerojatno povezivanje cjevovoda na standardni izlaz:

close( cj[0] ); //zatvaranje cjevovoda za čitanje (cj[0] je za čitanje)

close(std\_out); //zatvaranje STD\_OUT (1)

dup( cj[1] ); //dupliciranje -> sigurno će dobiti identifikator od upravo zatvorenog STD\_OUT

close( cj[1] ); //zatvaranje originalnog cjevovoda za pisanje ~~- pretpostavljam da se ovaj korak može izostaviti~~ - bez ovoga bi cjevovod imao dva handlera za pisanje

^^Golub je rekao da opisnik koji dupliciramo ne moramo zatvarati

29. Kako proces može znati da je poruka u redu poruka namijenjena za njega?

Prema tipu poruke

30. Što se dodaje na TCP razini

Redni broj i zaštita

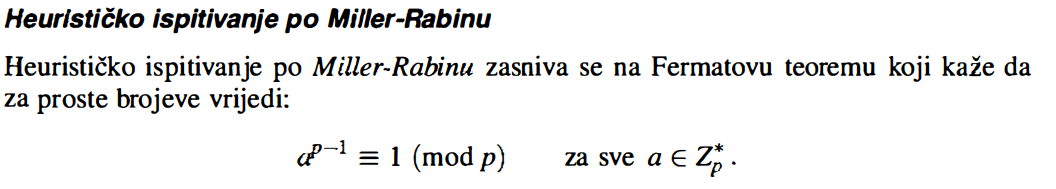
* Redni broj bloka
* Zaštita u obliku paritetnih bitova

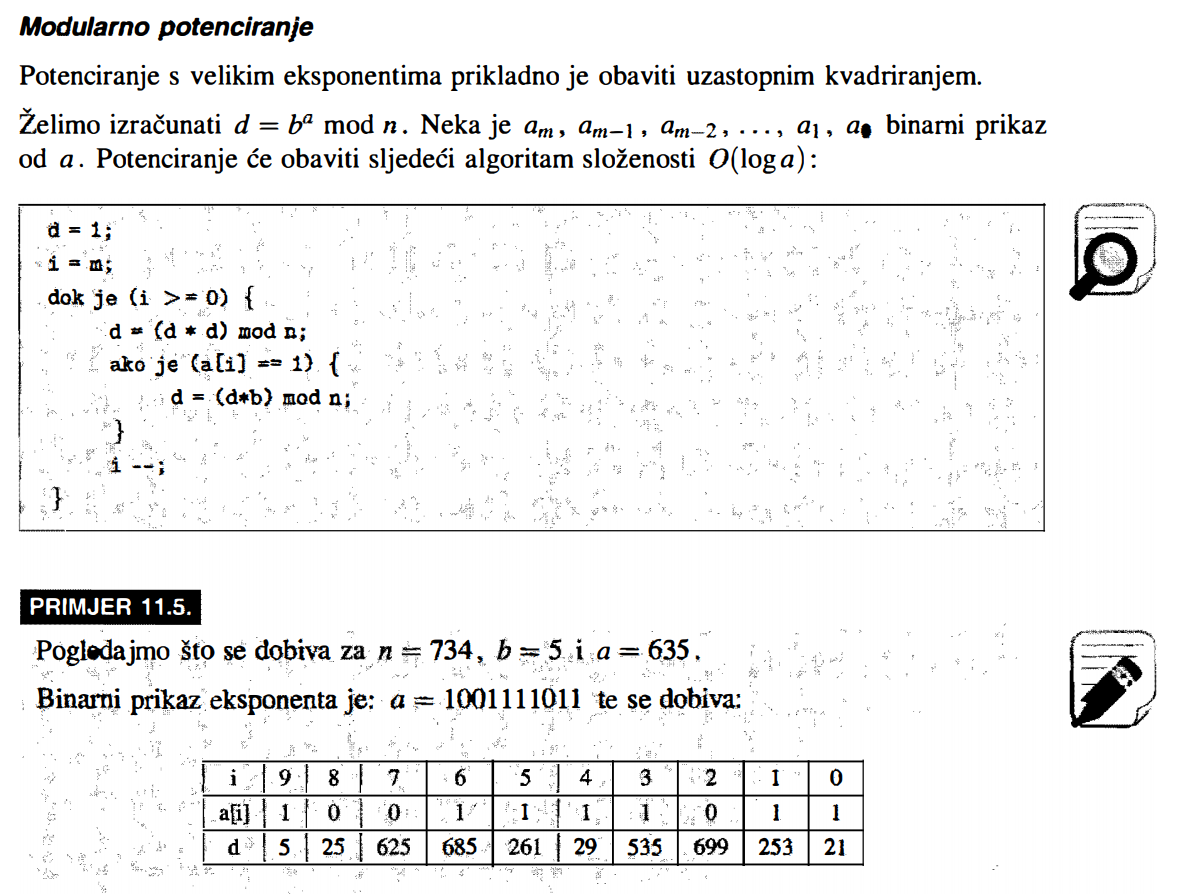
31. U postupku povezivanja dva procesa okolina može poslužiti za razmjenu

identifikatora reda poruka

32. Pronaći prvi broj p veći od 7700 koji prolazi ispitivanje radi li se o prostom broju Fermatovim teoremom. U postupku ispitivanje umjesto proizvoljnog broja nasumično odabranih brojeva a∈Zp\* koristite samo dva broja i to 2 i 7. (ako se za prvi broj 2 utvrdi složenost ne treba provjeravati za drugi broj 7. Napomena. U svim zadacima u ovom ispitu gdje treba izračunati izraz ab mod n treba koristiti modularno potenciranje.

Pls zna li netko ovo rijesiti?





Izračunavaš 27699 mod 7700 = 1 i 77699 mod 7700 = 1

Ako obje jednakosti vrijede, broj je vjerojatno prost (bar za potrebe zadatka). Ako makar jedna ne drži, broj je sigurno složen. Dakle prvo izračunaš za 2, ako dobiješ 1 nastavljaš računati za 7, ako ne, složen je, ne maltretiraš se dalje.  
  
Ovo je provjera za 7700 što nema smisla, ali process je isti za 7701, 7703, 7707, etc

Kako ste dobili taj 7.699? 7700-1

Ne znam zašto su uzimali 7699, trebalo bi se naći broj veći od 7700, pa bi se kao redom trebalo provjeravat za 7701, 7702 (<- nije prost), a prvi prosti će biti 7703. Onda bi provjeru trebalo raditi: 27702 mod 7703 = 1.4

Ricard i lamport s predavanja - isto kao [ovo](https://www.fer.unizg.hr/_download/repository/NOS_aud_zad_10.1.pdf)