

## DISTRIBUIRANI SISTEMI

### Teorija:

1. Interface definition language (IDL) se koristi da:
  - (a) Omogućuje generisanje stub funkcija.
  - (b) Implementira serversku stranu udaljene funkcije.
  - (c) Pristupi udaljenim objektima
  - (d) Definiše interfeise za sve lokalne i udaljene klase u programu.
2. Kako se može locirati server koji implementira udaljenu proceduru?
3. Nad objektom x obavljene su sledeće operacije u distribuiranom skladištu:
  - P1: write(x)A
  - P2: write(x)B, read(x) C
  - P3: read(x) B, read(x) A, write(x)C
  - P4: read(x) B, read(x) A

P1 w(x)A  
 P2 w(x)B r(x)C  
 P3 r(x)B r(x)A w(x)C  
 P4 r(x)B r(x)A

Koji od navedenih redosleda događaja je korektan ako je skladište sekvencijalno konzistentno:

(a) P2: x.write(B) P3: x.read() = B P4: x.read() = B P1: x.write(A) P3: x.read() = A P4: x.read() = A P3: x.write(C) P2: x.read() = C	(b) P1: x.write(A) P2: x.write(B) P3: x.read() = B P4: x.read() = B P3: x.read() = A P4: x.read() = A P3: x.write(C) P2: x.read() = C	(c) P2: x.write(B) P3: x.read() = B P4: x.read() = B P1: x.write(A) P4: x.read() = A P3: x.read() = A P3: x.write(C) P2: x.read() = C
--	--	--

P1 w(x)A  
 P2 w(x)B  
 P3 r(x)B r(x)A  
 P4 r(x)B r(x)A  
 P1 w(x)A  
 P2 w(x)B  
 P3 r(x)B r(x)A  
 P4 r(x)B r(x)A

(d) ni jedan od ponuženih odgovora nije korektan

4. Šta je definicija potpuno uređene grupne komunikacije? Kako se ovo ostvaruje u distribuiranom sistemu?

5. Koji system je otporan na greške Vizantijskog tipa
  - a. Sistem koji je otporan na otkaz više čvorova
  - b. Sistem koji je otporan na gubitak više poruka
  - c. Sistem koji je otporan na podeľu mreže
  - (d) Sistem koji je otporan na maliciozne čvorove

6. Kada je reč o sinhronizaciji međusobno uslovljenih događaja, vektorske markice se u prijemniku koriste da se:

- (a) Obezbedi da se sve poruke prime u istom redosledu u svim članovima grupe.
- (b) Kreiraju vremenske markice koje govore kada je svaka poruka primljena.
- (c) Identifikuje da li su sve prethodne uslovljene poruke primljene
- (d) Da se kreira zapis o tome kada su sve poruke primljene.

7. Ako događaj A ima Lamportovu markicu 2 a događaj B Lamportovu markicu 3, tada smo sigurni da:

- (a) A se desilo pre B ako su se ovi događaji desili u istom procesu.
- (b) A se desio pre B nezavisno od toga u kojim procesima su nastupili A i B.
- (c) A i B su konkurentni.
- (d) B se desio pre A, nezavisno od toga u kojim procesima su nastupili A i B.

8. HDFS fajlovi su namenjeni za:



- (a) Više klijenata koji vrše upis i omogućava upis na proizvoljnu lokaciju u fajlu
- (b) Dodavanje podataka samo na kraj fajla
- (c) Upis u fajl samo jedanput
- (d) Nisku latenciju pristupa podacima

### Zadaci:

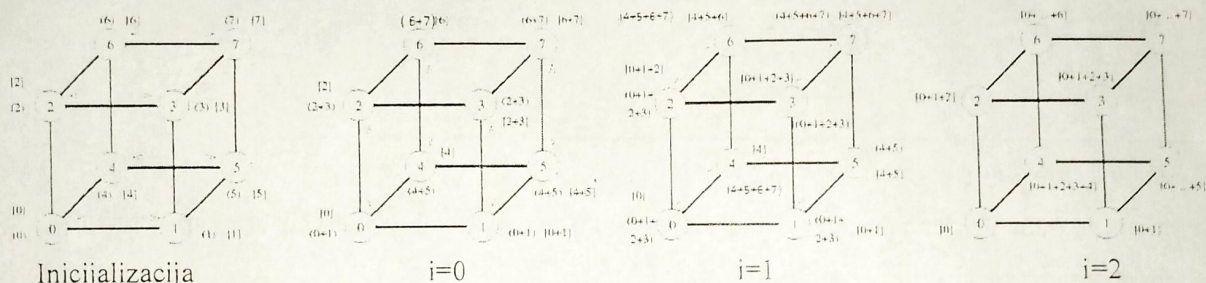
1. [30 poena]

a) Napisati MPI program kojim se vrši realizacija sumiranja opisanog sa:

```
for (i=c; i<c+N; i++)
    for (j=N-1; j>=0; j--)
        sum+=i+j;
```

ravnomernom cikličnom raspodelom posla između  $p$  procesa. Ovo znači da ako je  $k$  identifikator procesa, onda proces  $P_k$  ( $0 \leq k \leq p-1$ ) treba da izvrši  $k, k+p, k+2p, \dots, k+N^2-p$  sumiranje po redu u sekvencijalnom izvršenju programa. Pretpostaviti da je  $N$  veće od broja procesa  $p$  i da je  $N$  deljivo sa  $p$ , a da je  $c$  zadata konstanta. Rezultat programa prikazati u procesu koji sadrži najveći broj sabiraka koji su prosti brojevi. Nije dozvoljeno korišćenje indeksiranih promenljivih. Zadatak rešiti isključivo korišćenjem grupnih operacija

b) Napisati MPI program koji korišćenjem Point-to-Point operacija realizuje proces sumiranja celih brojeva korišćenjem  $p=n$  procesa ( $p$ -broj procesa) uređenih u hiperkub kao na Slici 1. Nakon izvršenja programa proces  $P_k$  ( $k=0, p-1$ ) treba da sadrži i prikaže sumu prvih  $k$  brojeva. Algoritam se odvija u  $\log_2 p$  koraka. Proces sabiranja po koracima je prikazan na Slici 1. U uglastim zagradama se nalaze vrednosti rezultata u svakom koraku. U obliku zagradama se nalaze vrednosti koje se šalju svakom procesu u odgovarajućem koraku. U svakom koraku proces sa rangom  $k$  razmenjuje svoje podatke sa procesom  $l=k \text{ XOR } 2^i$  ( $i$ -oznaka koraka).



Slika 1.

Koja grupna operacija se može iskoristiti za dobijanje istog rezultata?

2. [30 poena] Koristeći JMS napisati jednostavni mail server. Svaki korisnik pri startovanju klijenta unese svoj nadimak. Korisnik se može prijaviti sa istim nadimkom sa više različitih klijenata. Korisnik pod svojim nadimkom šalje poruku drugom klijentu tako što navede njegov nadimak (podržati više primalaca istovremeno), naziv poruke i sadržaj poruke. Servis isporučuje poruke svim korisnicima koji su trenutno aktivni. Ukoliko korisnik nije aktivan, poruka mu neće biti dostavljena. Klijent po prijemu poruke prikazuje poruku sledećim parametrima: ko je poslao, kad je poslata, kome je sve poslata, naslov i sadržaj poruke. Voditi računa da se poruke dostavljaju samo zainteresovanim stranama, a ne svima. Navesti na početku sve komunikacione kanale koje koristite i njihovu namenu.

flag za  
prijavu  
poruke  
treba da  
bude ispravan

3. [30 poena] Koristeći WCF napisati sistem za rad aritmetičkih operacija nad matricama. Operacije koje sistem treba da podrži su: Setovanje matrice na serverskoj strani, Sabiranje, Množenje matrice skalarom, Množenje matrice matricom i transponovanje matrice (bez pomoćne matrice), preuzimanje matrice sa servisa. Operacije kao jedan od operand uzimaju matricu koja je setovana na serverskoj strani, drugi operand se prosleđuje ako je potreban. Sve operacije treba da imaju zajedničku strukturu kojom vraćaju rezultat i obavezno mora sadržati flag da li je došlo do greške, opis greške (ako je do nje došlo) i sam rezultat operacije ako je uspešno izvršena. Klijent mora prikazati poziv svake operacije i prikaz rezultata. Obavezno napisati odgovarajuće konfiguracije.

**NAPOMENA: Radovi koji budu sadržali tragove grafitne olovke će biti diskvalifikovani!**