



Diplomski studij
Informacijska i
komunikacijska tehnologija:
Telekomunikacije i informatika
Računarstvo:
Programsko inženjerstvo i
informacijski sustavi
Računarska znanost

Ak.god. 2009./2010.

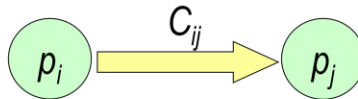
Raspodijeljeni sustavi

6.1

Model raspodijeljenog sustava

- ◆ Osnovni model raspodijeljenog sustava
- ◆ Model raspodijeljenog izvođenja
- ◆ Uzročna ovisnost događaja
- ◆ Globalno stanje raspodijeljenog sustava
- ◆ Proširenje osnovnog modela (sinkroni i asinkroni model)

- skup autonomnih procesa p_1, p_2, \dots, p_n
- C_{ij} – kanal koji povezuje procese p_i i p_j
- m_{ij} – poruka od p_i za p_j

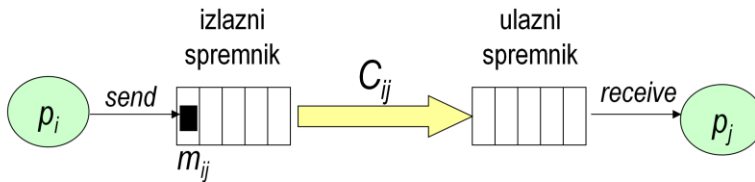


Raspodijeljeni sustav se sastoji od skupa autonomnih i asinkronih procesa p_1, p_2, \dots, p_n koji komuniciraju razmjenom poruka preko komunikacijske mreže. Komunikacijsku poveznicu između 2 procesa p_i i p_j možemo modelirati pomoću kanala C_{ij} koji opisuje virtualnu poveznicu, a ne fizički link.

- ◆ Izvođenje procesa i prijenos poruka su asinkroni
- ◆ Procesi ne dijele zajednički memorijski prostor
- ◆ Pri komunikaciji procesa neminovno se javlja kašnjenje
- ◆ Procesi ne koriste jedinstveni globalni sat

Ako pretpostavimo da se procesi izvode na različitim procesorima, oni su međusobno autonomni i asinkroni, a prijenos poruka je također asinkron. Procesi također ne dijele zajednički memorijski prostor. Zbog kašnjenja paketa pri prijenosu na fizičkom liku, neminovno se javlja kašnjenje poruka pri komunikaciji procesa. Procesi ne koriste jedinstveni zajednički sat te se javlja problem sinkronizacije vremena u raspodijeljenim sustavima.

- procesi međusobno komuniciraju razmjenom poruka (*message passing*) preko komunikacijskog medija (komunikacijske mreže)



- procesi koriste operatore *send* i *receive*
- *send*: pohranjuje poruku u izlazni spremnik i priprema za prijenos preko kanala
- *receive*: čita poruku iz dolaznog spremnika i prosljeđuje procesu

- ◆ Izvođenje procesa: slijedno izvođenje akcija procesa
- ◆ Akcije se modeliraju sljedećim događajima:
 - unutarnji događaj
 - slanje poruke
 - primanje poruke
- ◆ Događaj mijenja stanje procesa i komunikacijskog kanala
- ◆ Slijed događaja na procesu p_i :

$$e_i^1, e_i^2, e_i^3, \dots, e_i^x$$

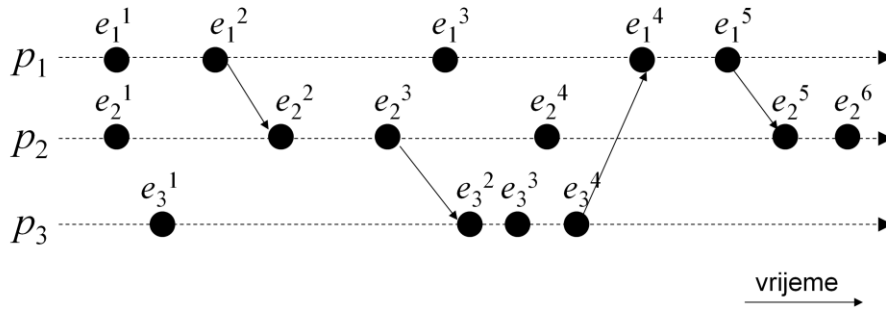
(e_i^3 se dogodio prije e_i^2)

Izvođenje procesa sastoji se od slijednog izvođenja njegovih akcija. Akcije mogu biti unutarnji događaju, slanje poruke i primanje poruke. Izvođenje događaja mijenja stanje procesa i komunikacijskog kanala (slanje i primanje poruke utječe na relevantni komunikacijski kanal), a interni događaj utječe samo na stanje procesa.

Primjer raspodijelnog izvođenja



Zavod za telekomunikacije



Koji događaji na slici predstavljaju unutarnje događaje, slanje poruke i primanje poruke?

Uzročna ovisnost događaja (1)



Zavod za telekomunikacije

- ◆ Uzročna relacija \rightarrow
 - izražava uzročnu ovisnost između dva događaja tijekom raspodijeljenog izvođenja, uzročnost može biti direktna ili tranzitivna
- ◆ $e_i^x \rightarrow e_i^y$
 - događaj e_i^x je izvršen na procesu p_i prije događaja e_i^y te su oni uzročno povezani
- ◆ $send(m) \rightarrow_{msg} receive(m)$
 - uzročna ovisnost vezana uz slanje i primanje poruke, da bi poruka bila primljena, mora prethodno nužno biti poslana na kanal
- ◆ $e_i^x \rightarrow e_k^z \wedge e_k^z \rightarrow e_i^y$
 - primjer tranzitivne uzročnosti događaja izvršenih na 3 različita procesa

Uzročna relacija označava uzročnu ovisnost dva događaja tijekom raspodijeljenog izvođenja. Lamport je naziva relacijom “*happens before*”. Na primjeru s prethodne slike vrijedi $e_1^1 \rightarrow e_3^3, e_3^3 \rightarrow e_2^6$.

Tranzitivna uzročnost: $e_i^x \rightarrow e_k^z \wedge e_k^z \rightarrow e_i^y$

Uzročna ovisnost događaja (2)



Zavod za telekomunikacije

- ◆ Uzročna relacija \nrightarrow
 - Označava neovisnost dvaju događaja tijekom raspodijeljenog izvođenja
- ◆ $e_i \nrightarrow e_j$
 - događaj e_j nije ovisan o događaju e_i
- ◆ Vrijede sljedeća pravila
 - za 2 događaja e_i i e_j , $e_i \nrightarrow e_j \nrightarrow e_j \nrightarrow e_i$
 - za 2 događaja e_i i e_j , $e_i \rightarrow e_j \Rightarrow e_j \nrightarrow e_i$
 - ako za 2 događaja e_i i e_j , vrijedi $e_i \nrightarrow e_j$ i $e_j \nrightarrow e_i$, onda su e_i i e_j konkurentni događaji i to možemo napisati na sljedeći način $e_i \parallel e_j$

Na primjeru s prethodne slike vrijedi $e_1^3 \nrightarrow e_3^3$, $e_2^4 \nrightarrow e_3^1$.

- ◆ Određeno lokanim stanjima procesa i kanala
- ◆ Stanje procesa određeno je stanjem lokalne memorije i lokalnim izvođenjem akcija
- ◆ Stanje kanala određeno skupom primljenih i poslanih poruka
- ◆ Izvođenje akcije mijenja lokano stanje procesa/kanala te istovremeno i globalno stanje raspodijeljenog sustava

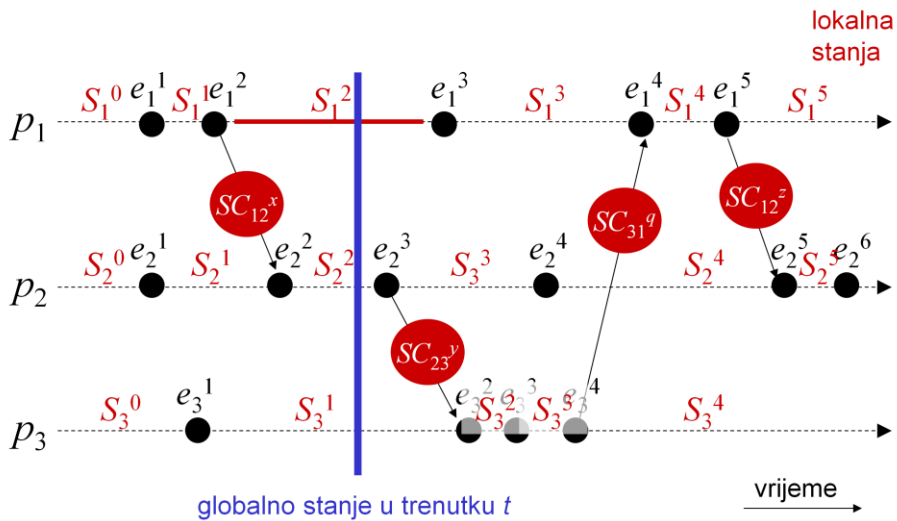
Globalno stanje raspodijeljenog sustava određeno je stanjem pojedinih procesa i komunikacijskih kanala. Stanje procesa određeno je stanjem lokane memorije i lokalnim izvođenjem akcija. Stanje kanala određeno je skupom primljenih i poslanih poruka.

Izvođenje bilo koje akcije mijenja lokano stanje procesa/kanala, ali istovremeno i globalno stanje raspodijeljenog sustava.

Primjer lokalnog/globalnog stanja



Zavod za telekomunikacije



$$GS(t) = \{S_1^2, S_2^2, S_3^1, SC_{12}^x, SC_{23}^y, SC_{31}^q\}$$

Na slici su naznačena lokalna stanja procesa i kanala. Na promjenu stanja utječu događaji. Svaki proces možemo dodatno opisati automatom stanja gdje prijelaze između stanja čine događaji.

Globalno stanje je skup lokalnih stanja procesa i kanala te se kontinuirano mijenja u vremenu. Na primjeru pretpostavljamo da se koriste samo 3 kanala te na globalno stanje sustava utječe promjena stanja na procesima i navedenim kanalima.

Sinkroni model

- ◆ Pretpostavka
 - Svi procesi raspodijeljenog sustava izvode korake istovremeno
 - pojednostavljenje koje nije realno za raspodijeljene sustave, ali može biti korisno za njihovo modeliranje i razumijevanje

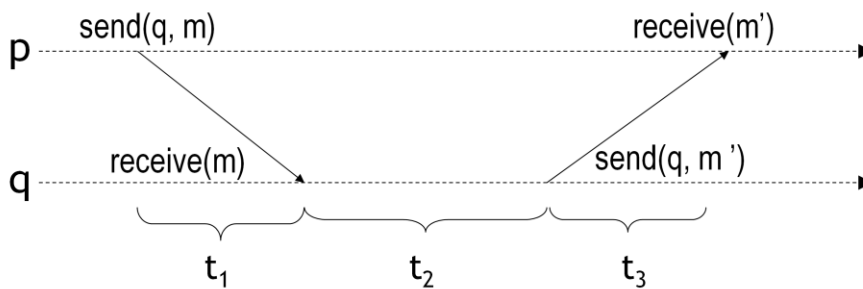
Asinkroni model

- ◆ Pretpostavka
 - komponente izvode akcije u proizvoljnom slijedu
 - postoji neodređenost vezana uz slijed događaja
 - realna situacija

Osnovni model može se proširiti tako da se raspodijeljeni sustav modelira kao sinkroni ili asinkroni sustav.

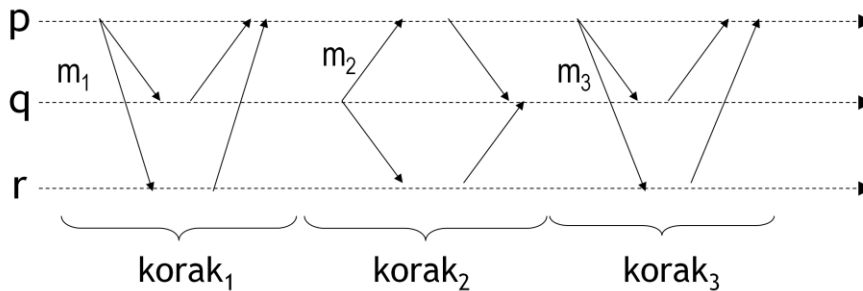
Sinkroni model se koristi isključivo radi pojednostavljenja osnovnog modela, pretpostavka je da svi procesi izvode korake istovremeno.

- ◆ Poznata je gornja vremenska granica za
 - izvođenje prijelaza nekog procesa
 - trajanje prijenosa poruke kanalom
- ◆ Pretpostavka
 - procesi imaju potpuno sinkronizirana lokalna vremena



Primjer sinkrone komunikacije

- ◆ izvođenje algoritma može se organizirati u koracima
 - pošalji poruke procesima u sustavu
 - primi poruke od drugih procesa u sustavu
 - promijeni stanje na temelju primljenih poruka



Algoritmi koji se temelje na sinkronom modelu se izvode u *koracima*. Inicijalno su svi procesi u početnom stanju i svi su kanali prazni. Nakon toga se izvode koraci.

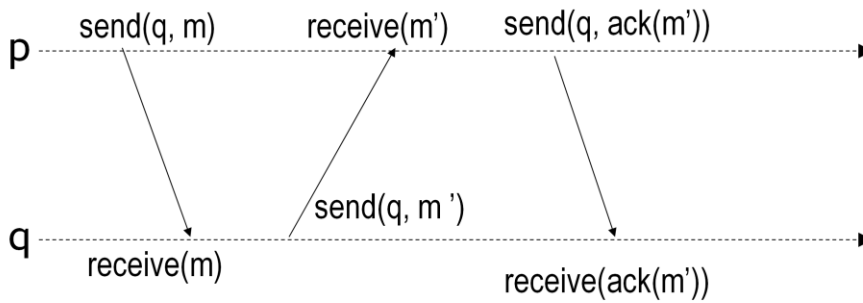
Korak se sastoji od 2 faze i izvodi se na svim procesima istovremeno!

Faza 1: Svaki proces generira poruke koje će biti poslane izlaznim susjedima i postavlja te poruke na izlazne kanale.

Faza 2: Svaki proces na temelju trenutnog stanja i primljenih poruka izvodi prijelaz (procesira primljene poruke) i određuje sljedeće stanje procesa. Briši sve poruke na kanalima.

- ◆ Ne postoji gornja vremenska granica za
 - izvođenje prijelaza nekog procesa (no trajanje prijelaza je uvijek konačno)
 - trajanje prijenosa poruke kanalom
- ◆ Pretpostavka
 - procesi **nemaju** sinkronizirana lokalna vremena
- ◆ Realni slučaj koji ćemo najčešće razmatrati, znatno komplicira model raspodijelnog sustava

Primjer asinkrone komunikacije



nepouzdana komunikacijski medij, potrebno je modelirati mogućnost gubitka poruke na kanalu