SVEUČILIŠTE U ZAGREBU
Fakultet elektrotehnike i računarstva

Diplomski studij
Računarstvo
Znanost o mrežama
Programsko inženjerstvo i informacijski sustavi
Računalno inženjerstvo
Ostali (slobodni izborni predmet)

Raspodijeljeni sustavi

6. Sinkronizacija procesa u vremenu

Ak. god. 2022./2023.

Creative Commons

- slobodno smijete:
  - **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
  - **prerađivati** djelo
- pod sljedećim uvjetima:
  - **imenovanje:** morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
  - **nekomercijalno:** ovo djelo ne smijete koristiti u komercijalne svrhe.
  - **dijeli pod istim uvjetima:** ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.

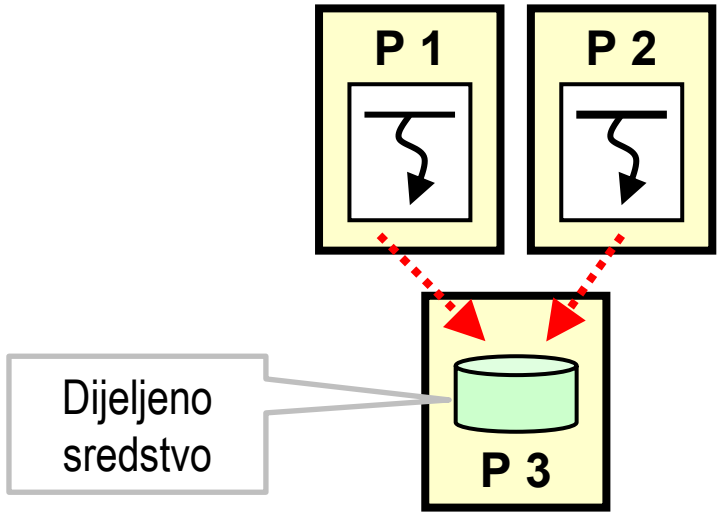
U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela.
Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.
Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava. *Tekst licence preuzet je s http://creativecommons.org/*

Sadržaj predavanja

- Motivacija: potreba za sinkronizacijom procesa u raspodijeljenoj okolini
- Primjena sata u jednoprocesorskoj okolini
- Primjena sata u raspodijeljenoj okolini
- Sinkronizacija tijeka izvođenja procesa
- Međusobno isključivanje procesa

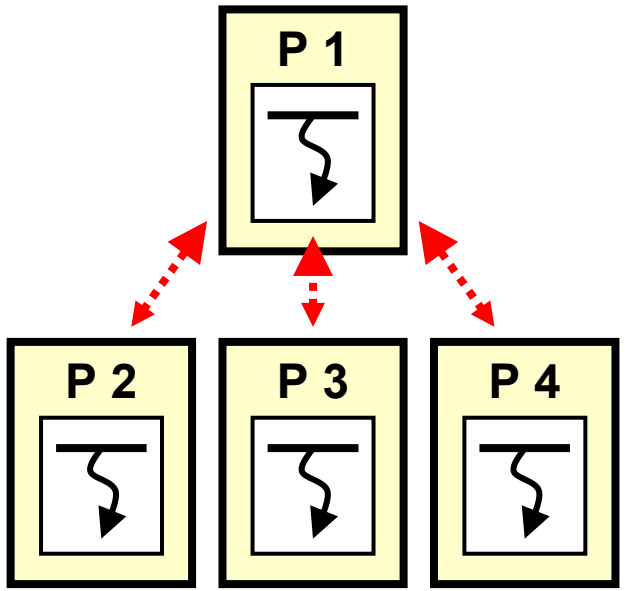
Potreba za sinkronizacijom procesa (1/4)

- **Uporaba dijeljenog sredstva u raspodijeljenoj okolini**
  - Procesi istodobno pristupaju dijeljenom sredstvu
  - Potrebno je ostvariti pristup dijeljenom sredstvu na međusobno isključiv način
  - Raspodijeljeni procesi moraju postići dogovor o redoslijedu pristupa sredstvu



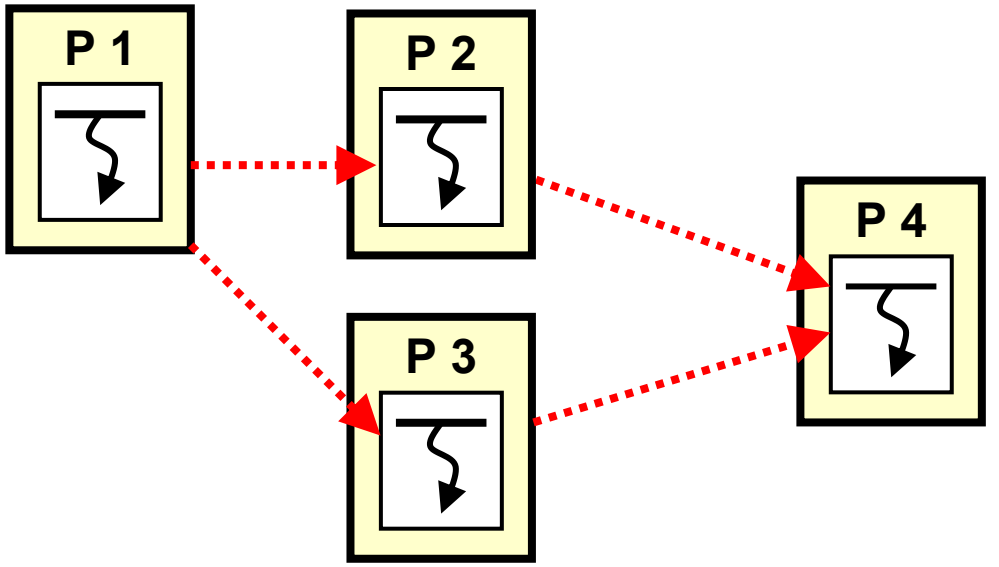
Potreba za sinkronizacijom procesa (2/4)

- **Nadgledanje i upravljanje nad izvođenjem poslova u raspodijeljenoj okolini**
  - Odabir upravljačkog procesa (sjetite se primjera s predavanja o modeliranju raspodijeljenog sustava)
  - Upravljački proces nadzire i određuje aktivnosti radnih procesa u raspodijeljenoj okolini



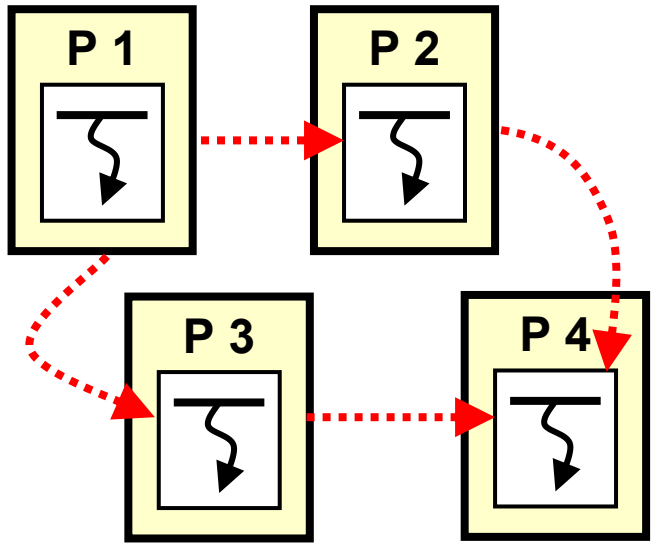
Potreba za sinkronizacijom procesa (3/4)

- **Usuglašavanje vremenskog redoslijeda izvođenja akcija**
  - Potrebno je omogućiti vremenski tijek izvođenja akcija na procesima u raspodijeljenoj okolini ako postoji međuovisnost među procesima



Potreba za sinkronizacijom procesa (4/4)

- **Uspostava suradnje skupa procesa u raspodijeljenoj okolini**
  - Ostvarivanje vremenski i prostorno usklađenog raspodijeljenog tijeka izvođenja (proširenje 3. primjera)
  - Primjer: P2P sustav za dijeljenje datoteka

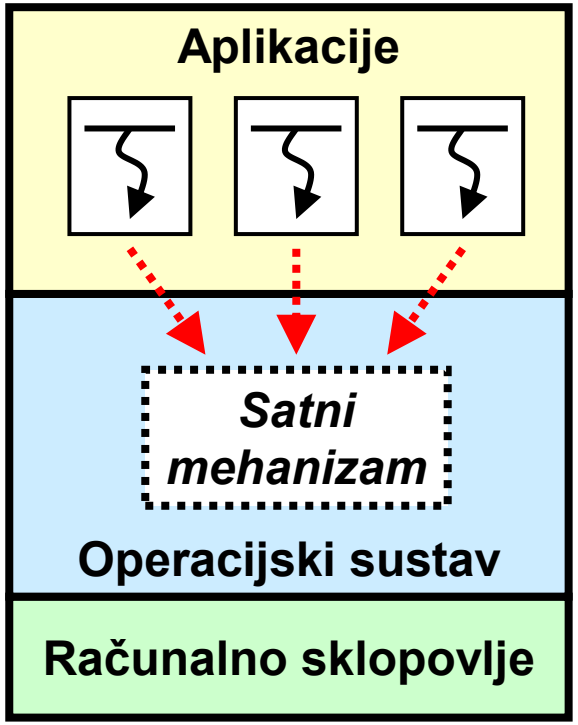


Sadržaj predavanja

- Motivacija: potreba za sinkronizacijom procesa u raspodijeljenoj okolini
- Primjena sata u jednoprocesorskoj okolini
- Primjena sata u raspodijeljenoj okolini
- Sinkronizacija tijeka izvođenja procesa
- Međusobno isključivanje procesa

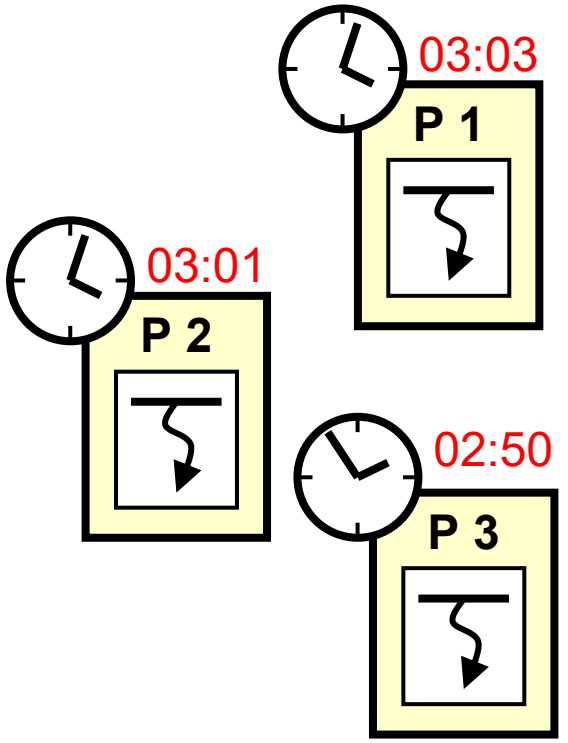
Primjena sata u jednoprocesorskoj okolini

- **Podsjetimo se: satni mehanizam operacijskog sustava**
  - Izveden uporabom kristala kvarca, osciliraju pod naponom zbog piezoelektričkog efekta
- **Aplikacije**
  - Procesi koriste i upravljaju mehanizmom sata
  - Primjena programskih knjižnica za uporabu satnog mehanizma
- **Zatvorena okolina**
  - Predvidiva vremena izvođenja procesa
  - Jednostavnija sinkronizacija procesa u vremenu



Fizički i logički sat

- **Svako računalo ima vlastiti satni mehanizam**
  - Satovi nisu usklađeni
  - Satovi imaju različiti takt
  - Satovi imaju različita odstupanja (pogrešku)
- **Usuglašavanje vremena**
  - Fizički sat u raspodijeljenoj okolini
  - Logički sat u raspodijeljenoj okolini



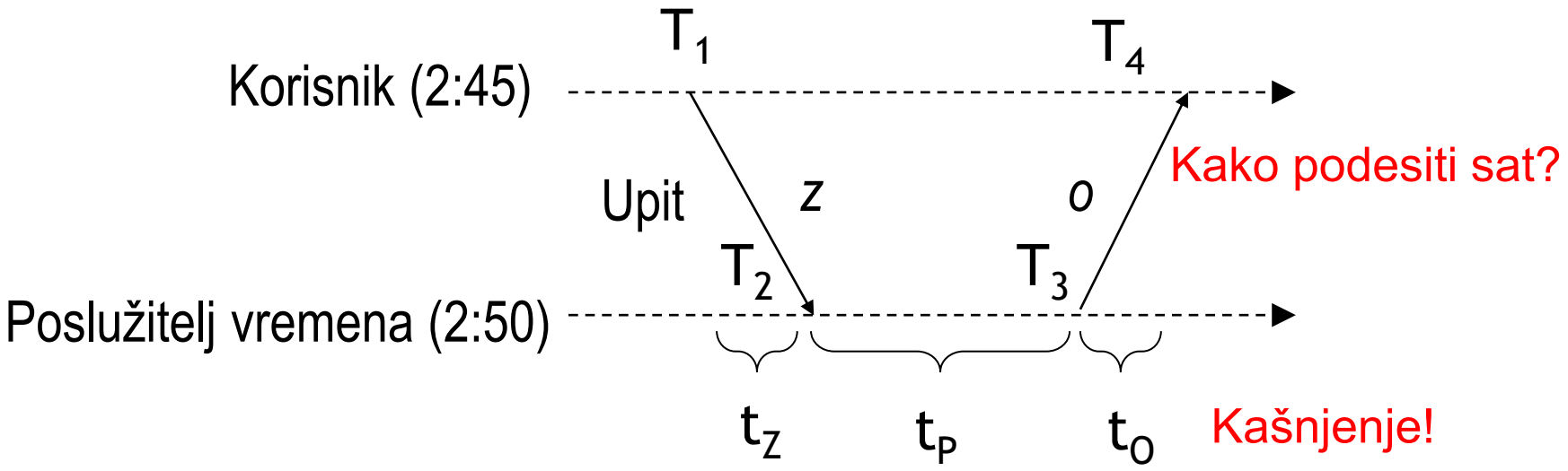
Fizički sat u raspodijeljenoj okolini

- **Cristianov algoritam**
  - Razvio ga je F. Cristian (1989)
  - Primjena poslužitelja s točnim vremenom, sinkronizacija prema vanjskom izvoru
  - Dohvaćanje informacije o vremenu prema potrebi

- **Algoritam Berkeley**
  - Razvili su ga R. Gusella i S. Zatti na University of California, Berkeley (1989)
  - Primjena upravitelja vremena, sinkronizacija unutar skupine procesa
  - Periodičko odašiljanje informacije o vremenu

Cristianov algoritam (1/2)

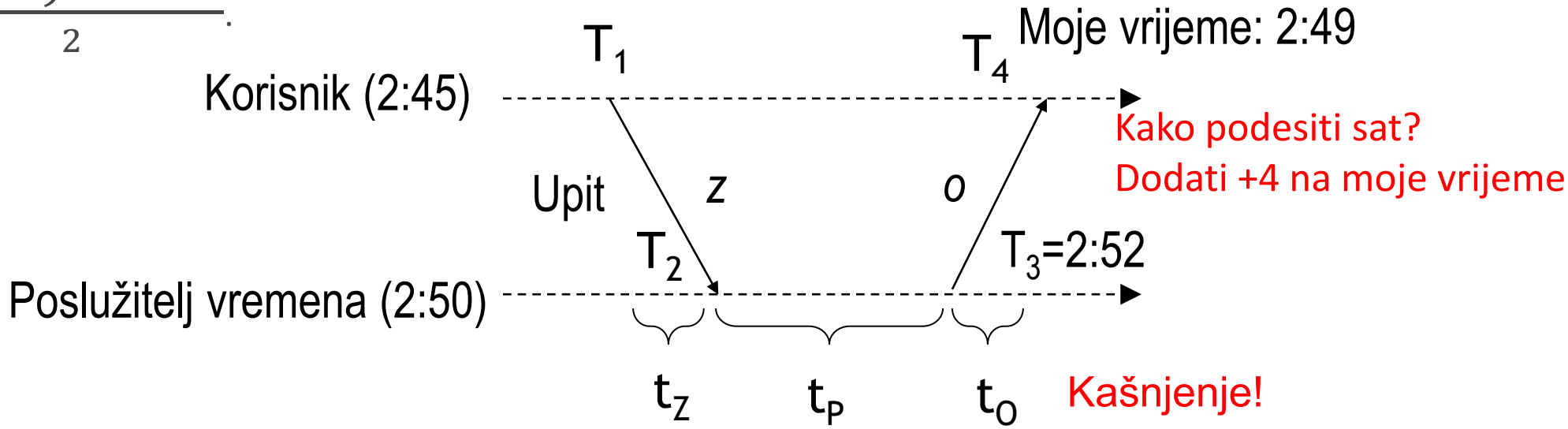
- **Primjena poslužitelja vremena**
- **Koraci algoritma**
  1) Korisnički proces upućuje zahtjev za dohvat vremena (*z*)
  2) Poslužitelj vremena prima i obrađuje zahtjev te šalje trenutno vrijeme (*o*)



Cristianov algoritam (1/2)

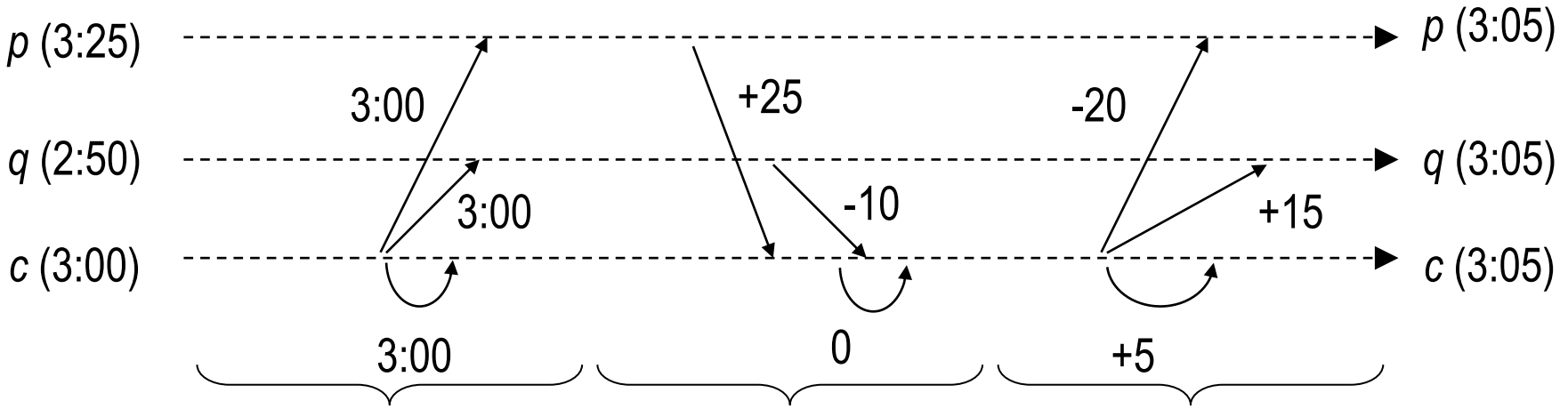
3. Odgovor sadrži T_2 i T_3 na poslužitelju u trenutku slanja odgovora.
4. Klijent na osnovi vremenskih trenutaka koje je primio u poruci i izmjerenih vremenskih trenutaka T_1 i T_4 pomiče svoje lokalno vrijeme za

$$\theta = T_3 + t_O - T_4 \approx T_3 + \frac{t_Z + t_O}{2} - T_4 = T_3 + \frac{(T_4 - T_1) - (T_3 - T_2)}{2} - T_4 = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}.$$



Algoritam Berkeley

- **Primjena upravitelja vremena**
- **Koraci algoritma**
  1) Upravljački proces *c* šalje vrijeme procesima *p*, *q*, *c*
  2) Procesi *p*, *q*, *c* šalju razliku vremena upravljačkom procesu *c*
  3) Upravljački proces *c* šalje pomak procesima *p*, *q*, *c*



Network Time Protocol (NTP)

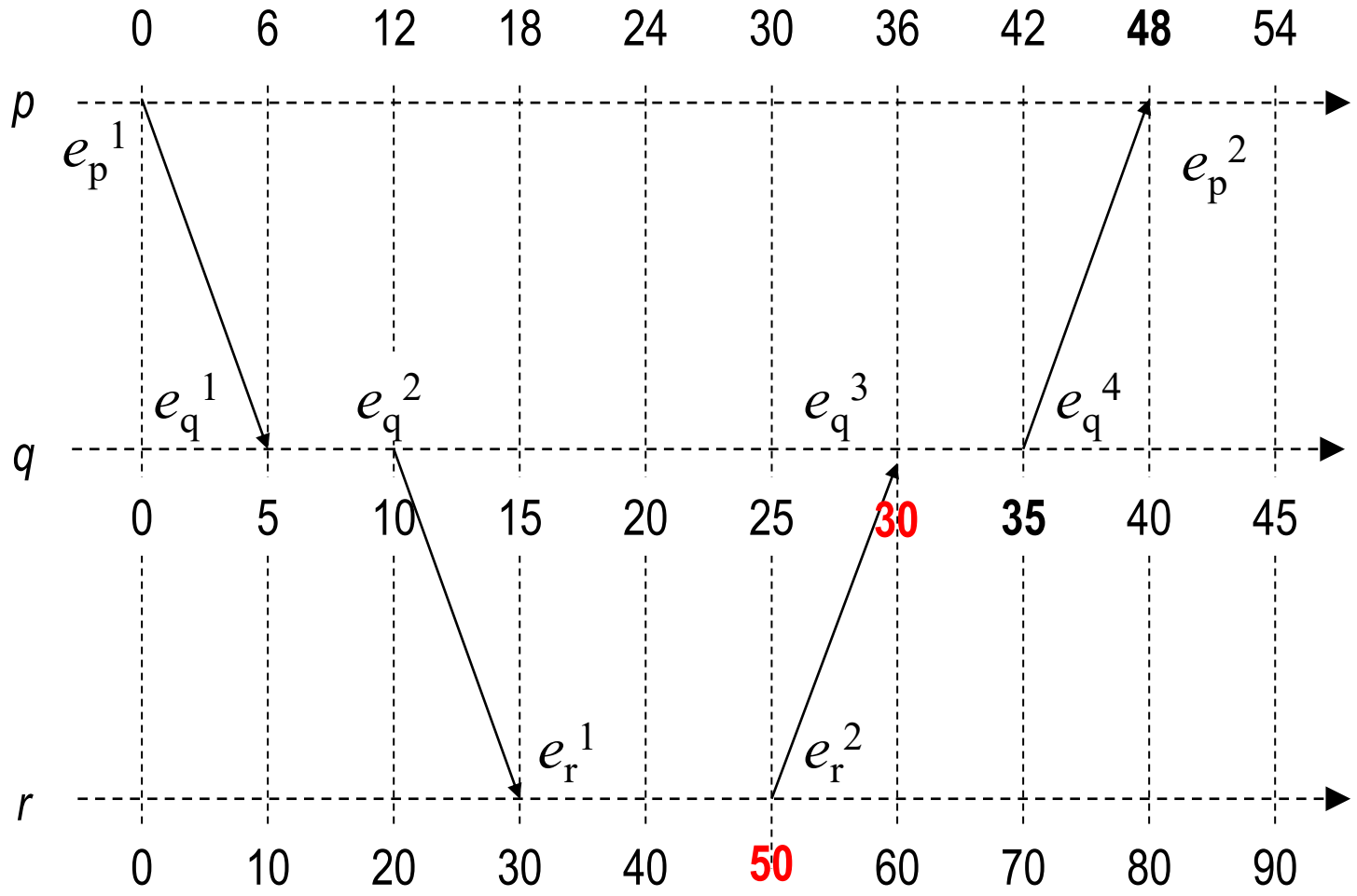
- Definira arhitekturu usluge za sinkronizaciju satnih mehanizama u raspodijeljenoj okolini i protokol za isporuku informacija o vremenu u Internetu
- Hijerarhijska organizacija NTP servisa
  - *primary servers*: povezani direktno na izvor sinkroniziran na UTC (Coordinated Universal Time)
  - *secondary servers*: sinkroniziraju se u odnosu na *primary servers*, itd.
  - preciznost: ~10 ms za računala na javnom Internetu, ~1 ms za računala u LAN-u
- RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification, 2010
- **Nedostatak fizičkog sata**: fizički satni mehanizmi su potpuno neovisni, stoga samo primjenom fizičkih satnih mehanizama nije moguće odrediti odnos događaja u vremenu (npr. redoslijed aktivnosti)

Primjer nedostatka fizičkog sata

$e_p^{\,1} \rightarrow e_q^{\,1}$ $\quad e_q^{\,2} \rightarrow e_r^{\,1}$

$e_r^{\,2} \rightarrow e_q^{\,3}$ $\quad e_q^{\,4} \rightarrow e_p^{\,2}$

$T(e_r^{\,2}) > T(e_q^{\,3})$



Logičke oznake vremena

- **Usklađivanje globalnog tijeka vremena**
  - Primjena logičkih oznaka vremena (*timestamps*)

- **Vrste logičkih oznaka**
  - Skalarne oznake vremena
  - Vektorske oznake vremena

Skalarne oznake vremena

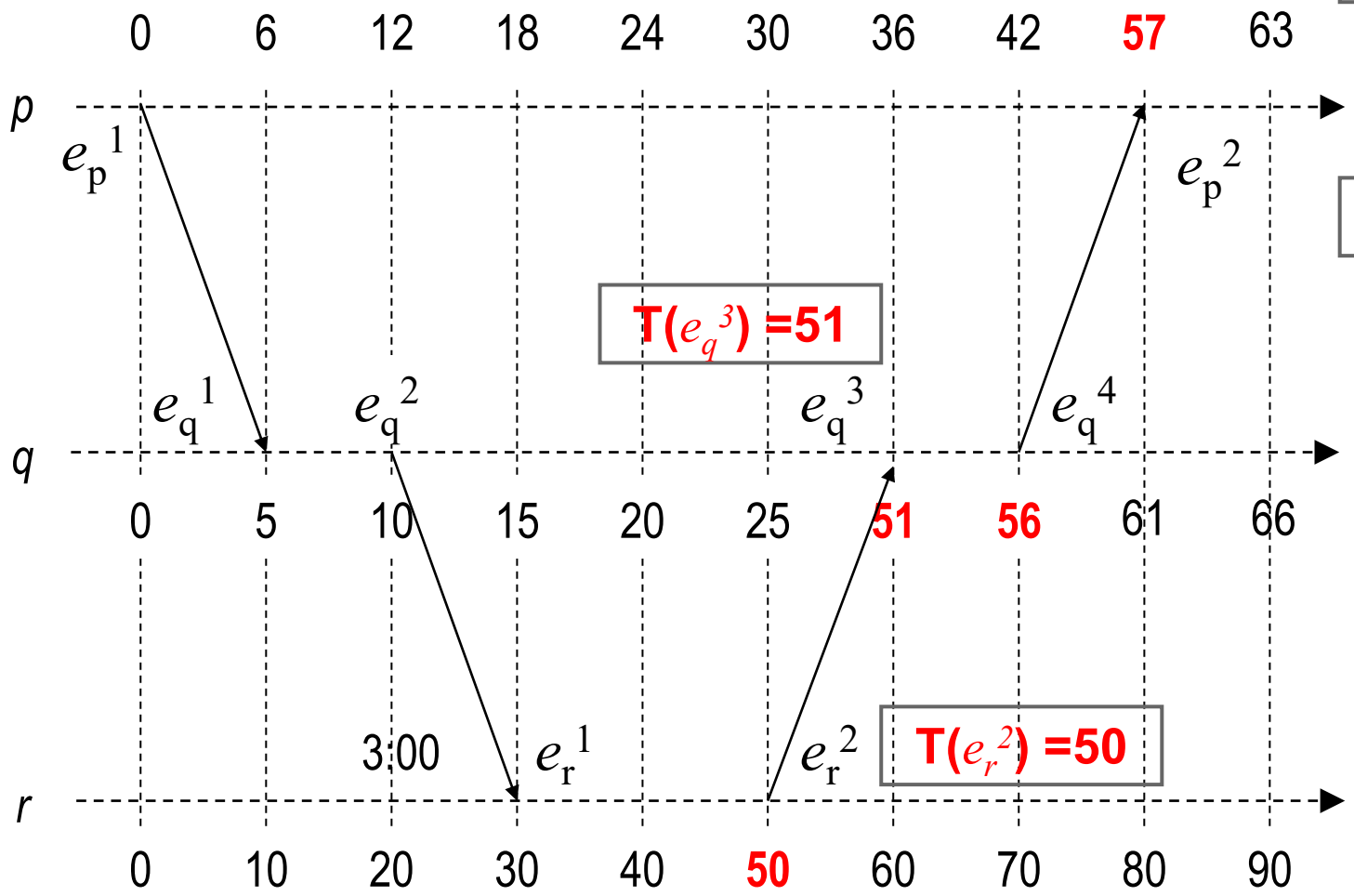
- **Globalno logičko vrijeme**
  - Sva računala na jednak način bilježe tijek globalnog logičkog vremena

- **Oznake logičkog vremena**
  - Svakoj akciji *a* koju provode procesi u raspodijeljenoj okolini pridružena je jedinstvena oznaka vremena T(*a*)
  - Ako za događaj *a* i *b* vrijedi uzročna relacija $a \rightarrow b$ tada vrijedi da je akcija *a* ostvarena u vremenu prije akcije *b* [T(*a*) < T(*b*)]

Primjer uporabe skalarnih oznaka vremena

$e_p^1 \rightarrow e_q^1$ $e_q^2 \rightarrow e_r^1$

$e_r^2 \rightarrow e_q^3$ $e_q^4 \rightarrow e_p^2$



$T(e_q^3) = 51$

$T(e_r^2) = 50$

$T(e_r^2) < T(e_q^3)$

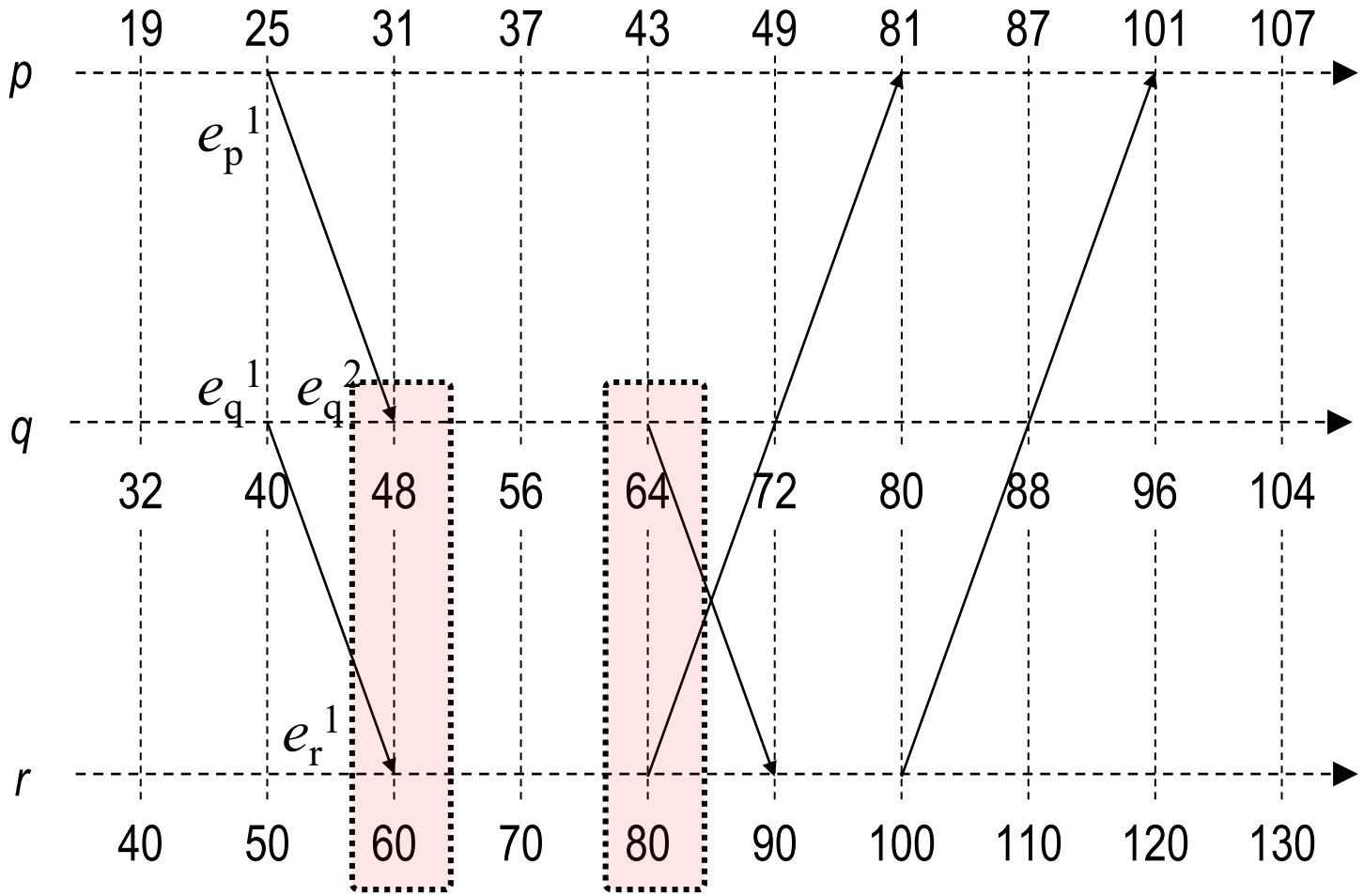
$T(e_q^4) < T(e_p^2)$

Obilježja skalarne oznake vremena

- **Prednosti primjene skalarnih oznaka**
  - Tijek vremena zasnovan je na jednostavnom modelu
  - Svi procesi usklađeni su s globalnim tijekom vremena
  - Usuglašeni su vremenski trenutci nastupanja akcija u raspodijeljenoj okolini

- **Nedostatci primjene skalarnih oznaka**
  - Ako za događaje *a* i *b* vrijedi da je vremenska oznaka od *a* manja od vremenske oznake od *b,* to ne povlači nužno da je događaj *a* nastupio u vremenu prije događaja *b*
  - $T(a) < T(b)$ *ne povlači* $a \rightarrow b$

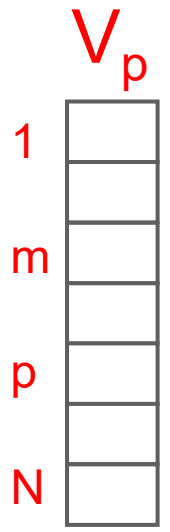
Primjer nedostatka skalarnih oznaka



T(e_q^2) = 48, T(e_r^1) = 60 ?

Vektorske oznake vremena (1/2)

- **Vektorska oznaka opisuje uzročno-posljedične veze između događaja u vremenu**
  - Polje elemenata $V[N]$ opisuje broj akcija (unutarnja akcija, slanje poruke, prijam poruke) provedenih od N procesa u raspodijeljenoj okolini
  - Procesi razmjenjuju vektorske oznake tijekom razmjene poruka
- **Vektorska oznaka**
  - $V_p[p]$ broj akcija koje je ostvario proces P_p
  - $V_p[m]$ broj akcija za koje proces P_p zna da su ostvarene od strane procesa P_m

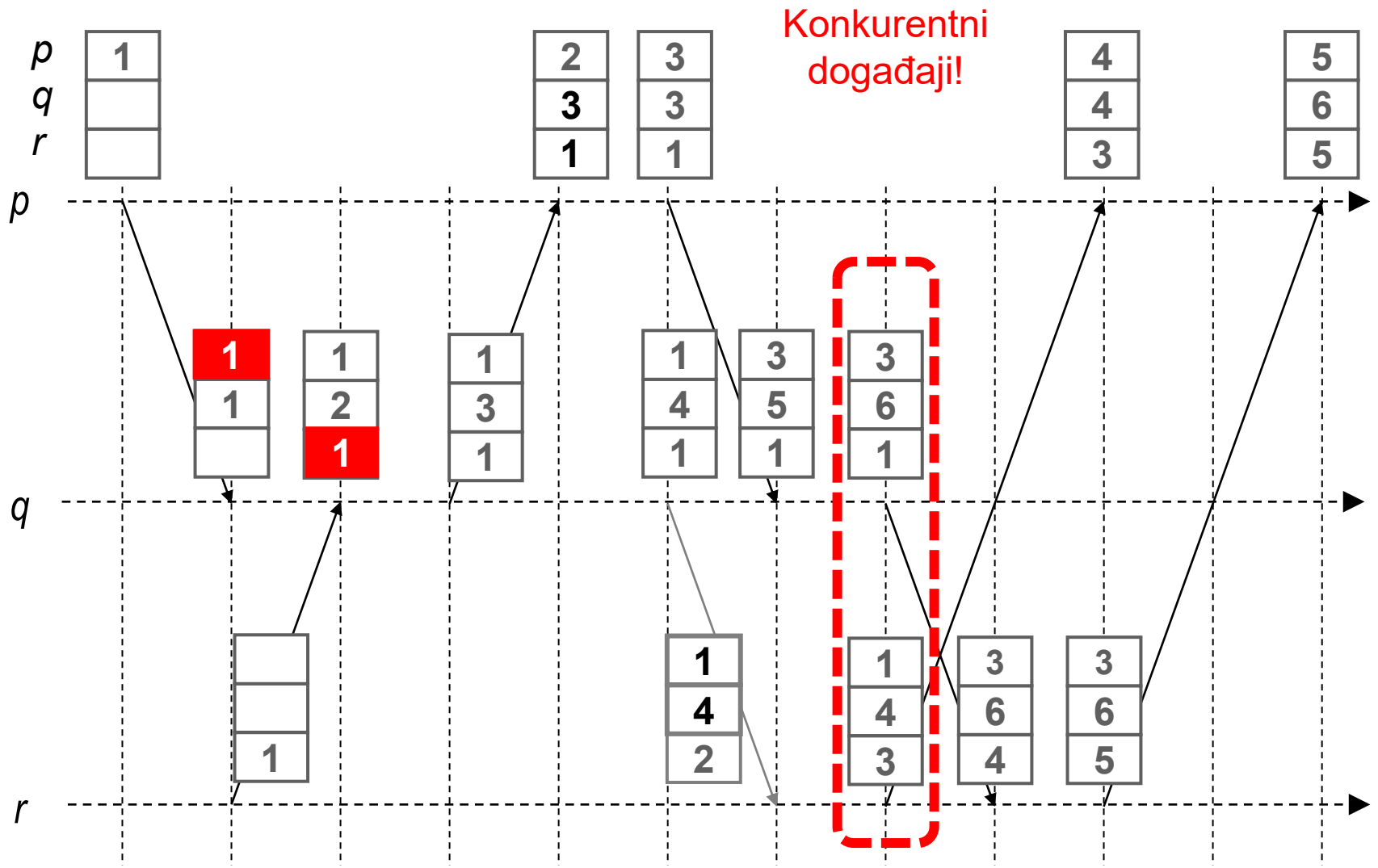


Vektorske oznake vremena (2/2)

- **Primjena vektorskih oznaka**
  - Ako za događaje *a* i *b* vrijedi $V(a) < V(b)$ tada vrijedi da je događaj *a* nastupio u vremenu prije događaja *b*, $a \rightarrow b$

- **Za dvije vektorske oznake V_i i V_j vrijedi $V_i < V_j$ ako:**
  - postoji barem jedan *k* za koji vrijedi $V_i[k] < V_j[k]$,
  - za sve ostale $l \neq$ k vrijedi $V_i[l] \leq V_j[l]$,
  - $i, j, k, l \in [0, N-1]$ i
  - broj procesa u raspodijeljenoj okolini je *N*

Primjer uporabe vektorskih oznaka



Vektorske oznake vremena

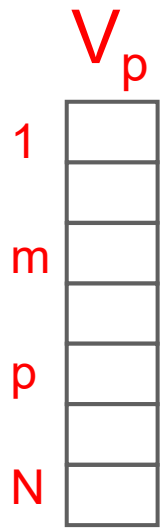
Koraci algoritma za održavanje vektorskih oznaka:

1) Početne vrijednosti svih vektorskih oznaka su postavljene na 0.

2) Za svaku unutarnju akciju procesa *p* uvećaj vremensku oznaku na procesu *p* pridijeljenu tome procesu za 1, tj. $V_p[p]+1$.

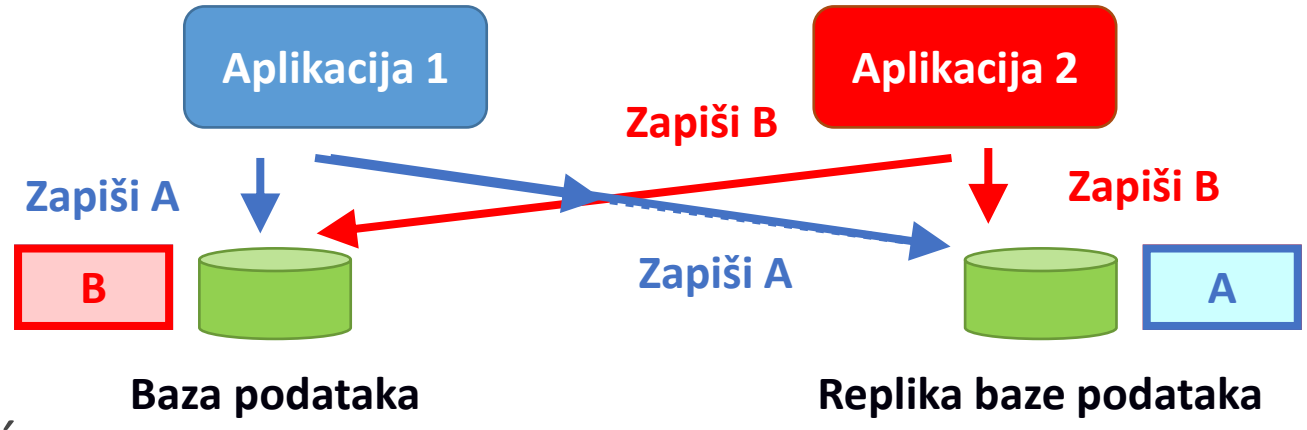
3) Prije slanja poruke na procesu *p* uvećaj oznaku $V_p[p]$ za 1 i poslanoj poruci pridruži izgrađeni vektor V_p.

4) Nakon primitka poruke od procesa *p* na procesu *k* uvećaj oznaku $V_k[k]$ za 1. Za ostale oznake $i \neq k$ postavi $V_k[i] = V_p[i]$ ako je $V_k[i] < V_p[i]$.



Primjena sata u raspodijeljenoj okolini

- **Uređena razmjena poruka**
  - Primjena skalarnih logičkih oznaka vremena
  - Svi procesi na isti način vide redoslijed događaja
- **Održavanje konzistentnosti**
  - Bez vremenskih oznaka nije moguće odrediti pravilni redoslijed akcija u vremenu



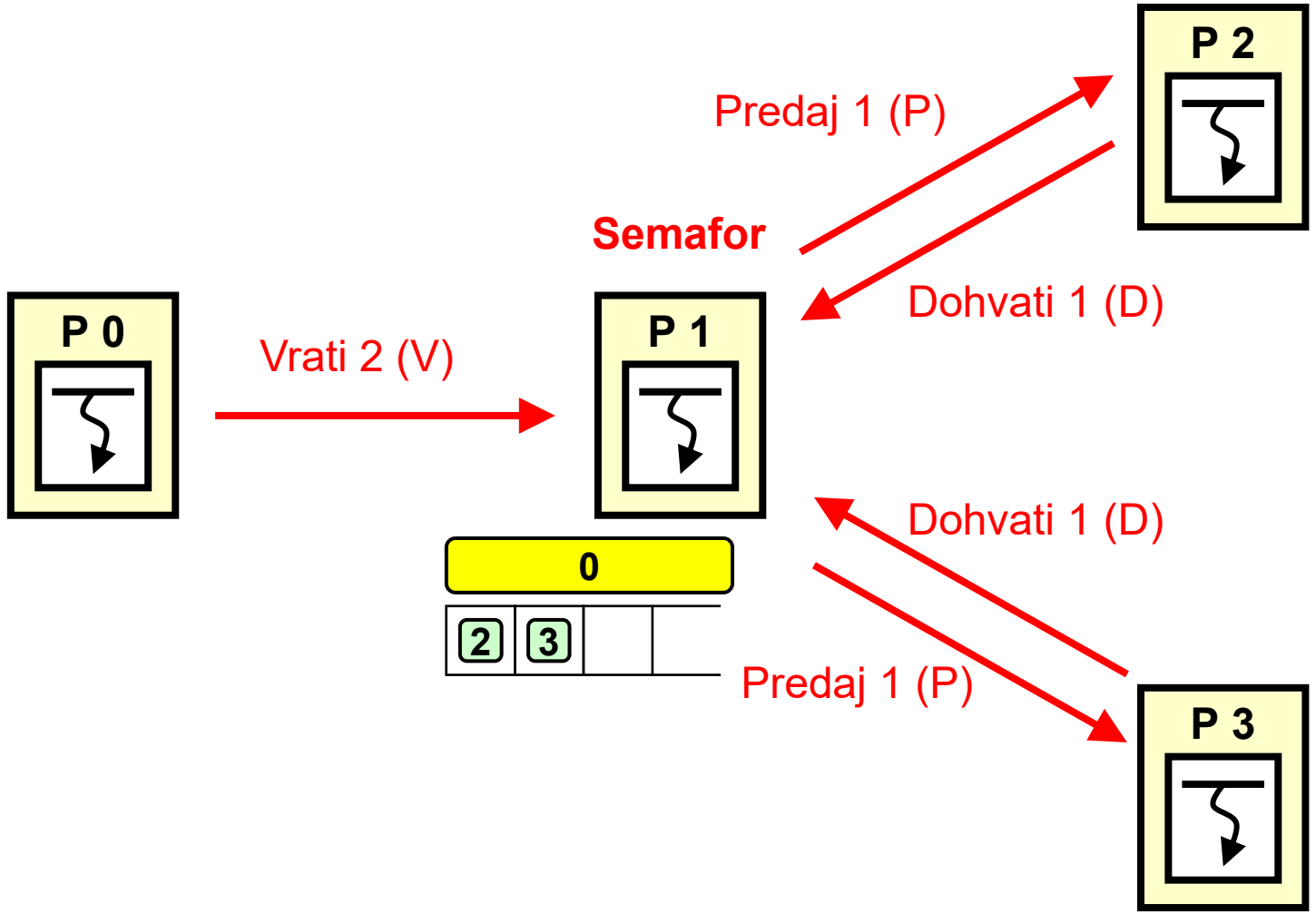
Sadržaj predavanja

- Potreba za sinkronizacijom procesa
- Primjena sata u jednoprocesorskoj okolini
- Primjena sata raspodijeljenoj okolini
- Sinkronizacija tijeka izvođenja procesa
  - Primjena semafora u raspodijeljenoj okolini
  - Sinkronizacija zasnovana na razmjeni obavijesti
- Međusobno isključivanje procesa

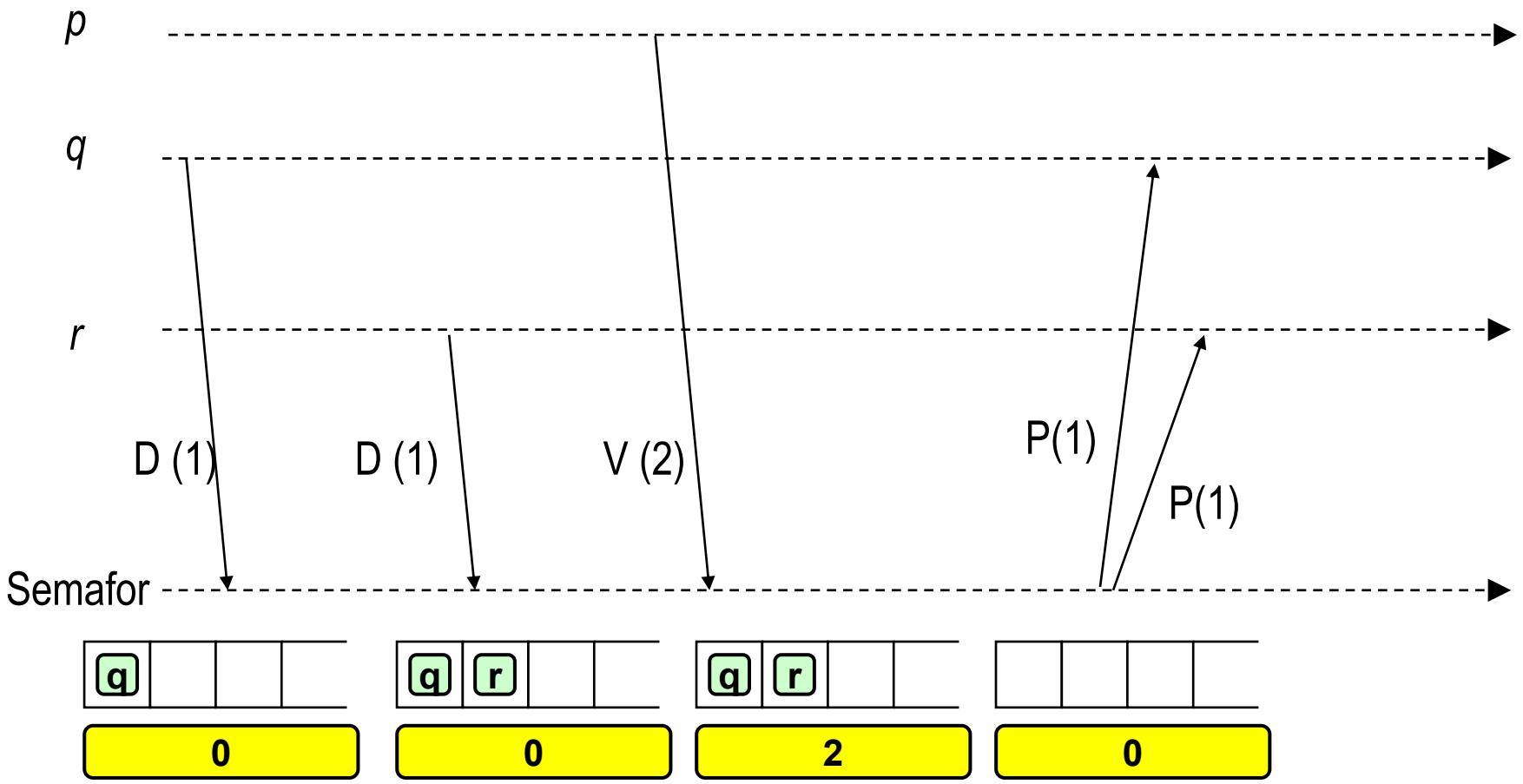
Primjena semafora u raspodijeljenoj okolini

- **Semafor**
  - Proces koji u spremniku čuva *N* znački (*token*)
  - Rep čekanja zasnovan na posluživanju zahtjeva prema redoslijedu prispijeća (*FIFO*)
- **Korisnici**
  - Procesi šalju poruke *zahtjev za dohvat* (*D*) *n* znački
  - Ako u spremniku postoji traženi broj znački, prosljeđuje se *potvrda za predaju* (*P*)
  - Ako u spremniku ne postoji traženi broj znački, zahtjev se stavlja u rep čekanja
  - Nakon završetka obrade, procesi vraćaju preuzete značke slanjem poruke *vrati* (*V*)

Semafor u raspodijeljenoj okolini



Primjer sinkronizacije tijeka izvođenja



D – Dohvati

V – Vrati

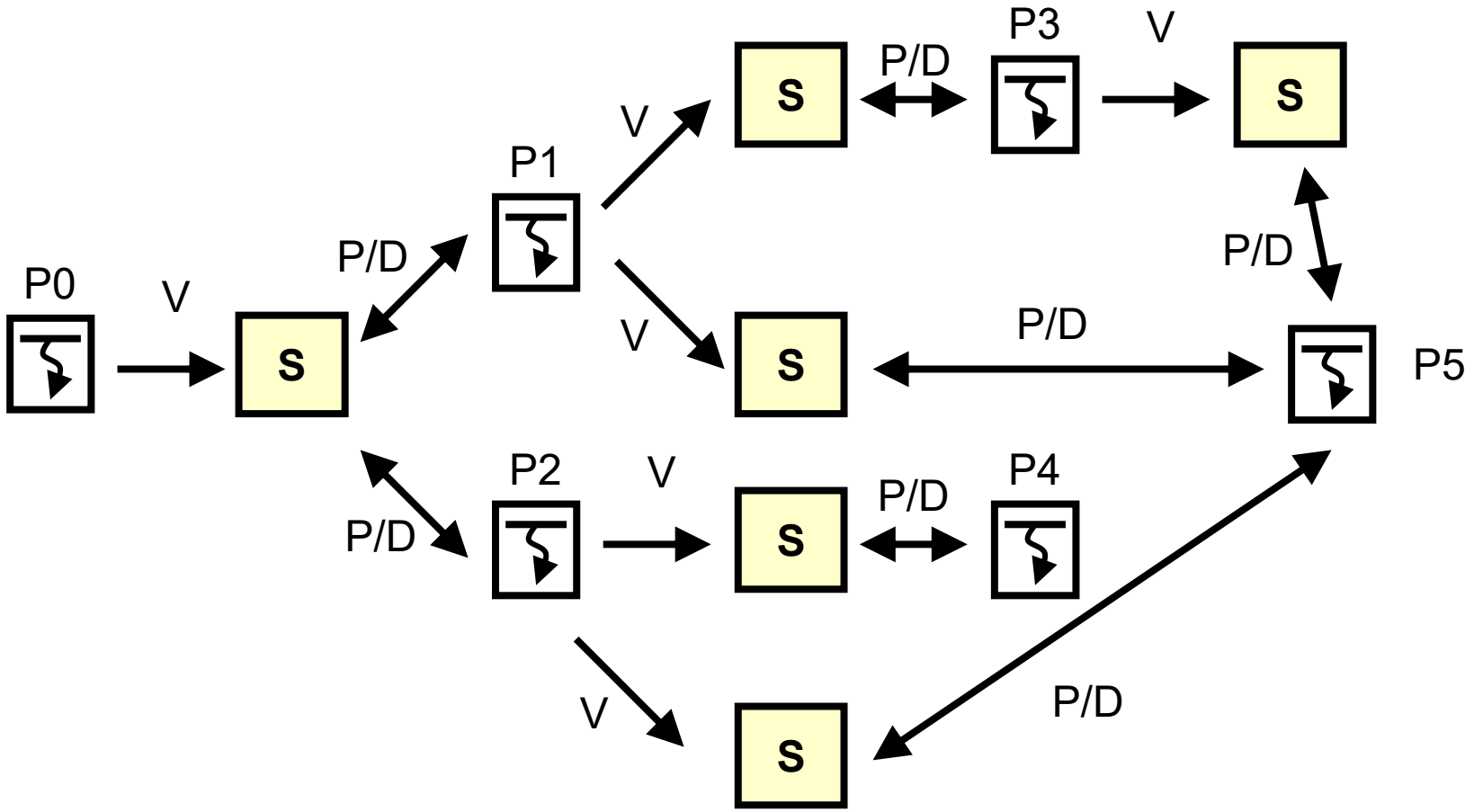
P – Predaj

Složeni obrasci sinkronizacije

- **Semafor je osnovni element za ostvarivanje složenih obrazaca sinkronizacije**

- **Graf raspodijeljenog tijeka izvođenja procesa**
  - Grananje tijeka izvođenja
  - Spajanje tijeka izvođenja
  - Ponavljanje tijeka izvođenja

Graf raspodijeljenog tijeka izvođenja



D – Dohvati

V – Vrati

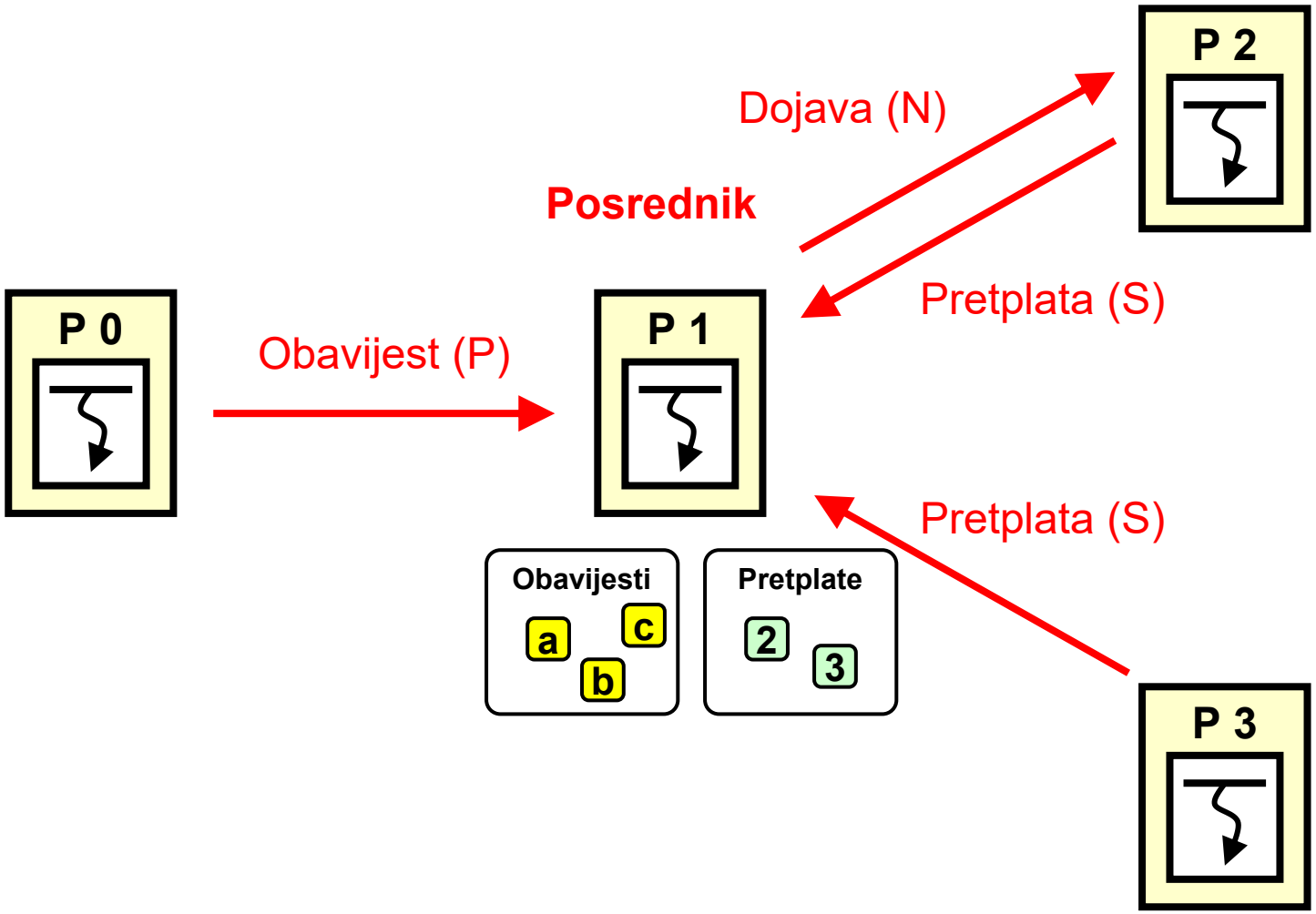
P – Predaj

Sinkronizacija razmjenom obavijesti

- **Posrednik**
  - Sadrži spremnik s obavijestima i spremnik pretplata na obavijesti
  - Ostvaruje postupak usporedbe obavijesti i pretplata prema modelu objavi - pretplati

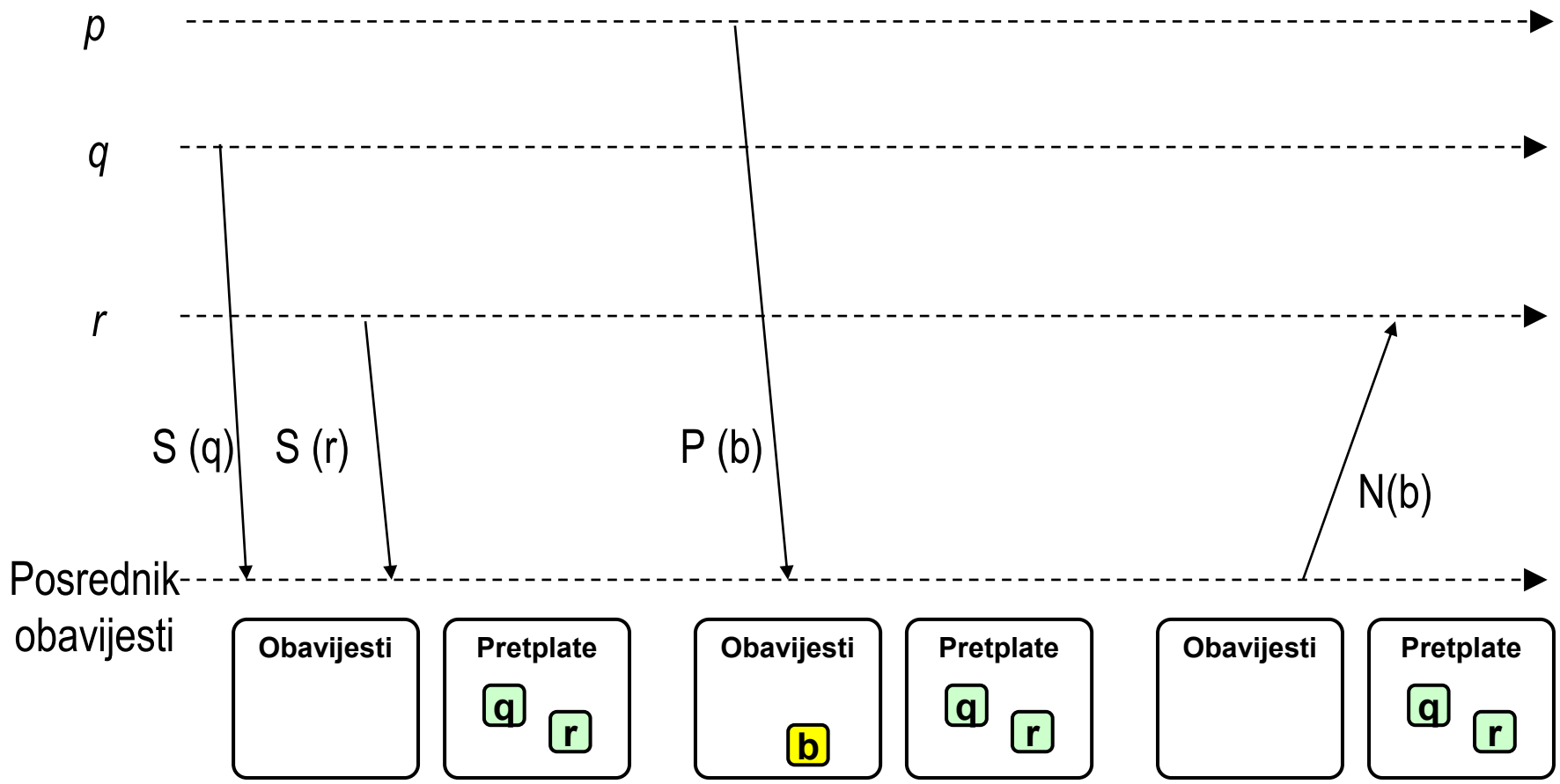
- **Korisnici**
  - Procesi šalju posredniku pretplate (*S*)
  - Procesi šalju posredniku obavijesti (*P*)
  - Ako posrednik ima aktivnu pretplatu na obavijest, ona se prosljeđuje procesu pretplatniku u poruci dojave (*N*)

Okolina posrednika obavijesti



Npr:
Proces P2 se pretplatio (*subscribe*) na obavijesti tipa A
Proces P0 objavljuje (*publish*) obavijest a posredniku
Posrednik prosljeđuje obavijest (notify) a procesu P2

Primjer sinkronizacije razmjenom obavijesti



S (x) – pretplata na obavijesti tipa x
P(x) – objavi obavijest x,
N(x) – dojava, tj. proslijedi obavijest x

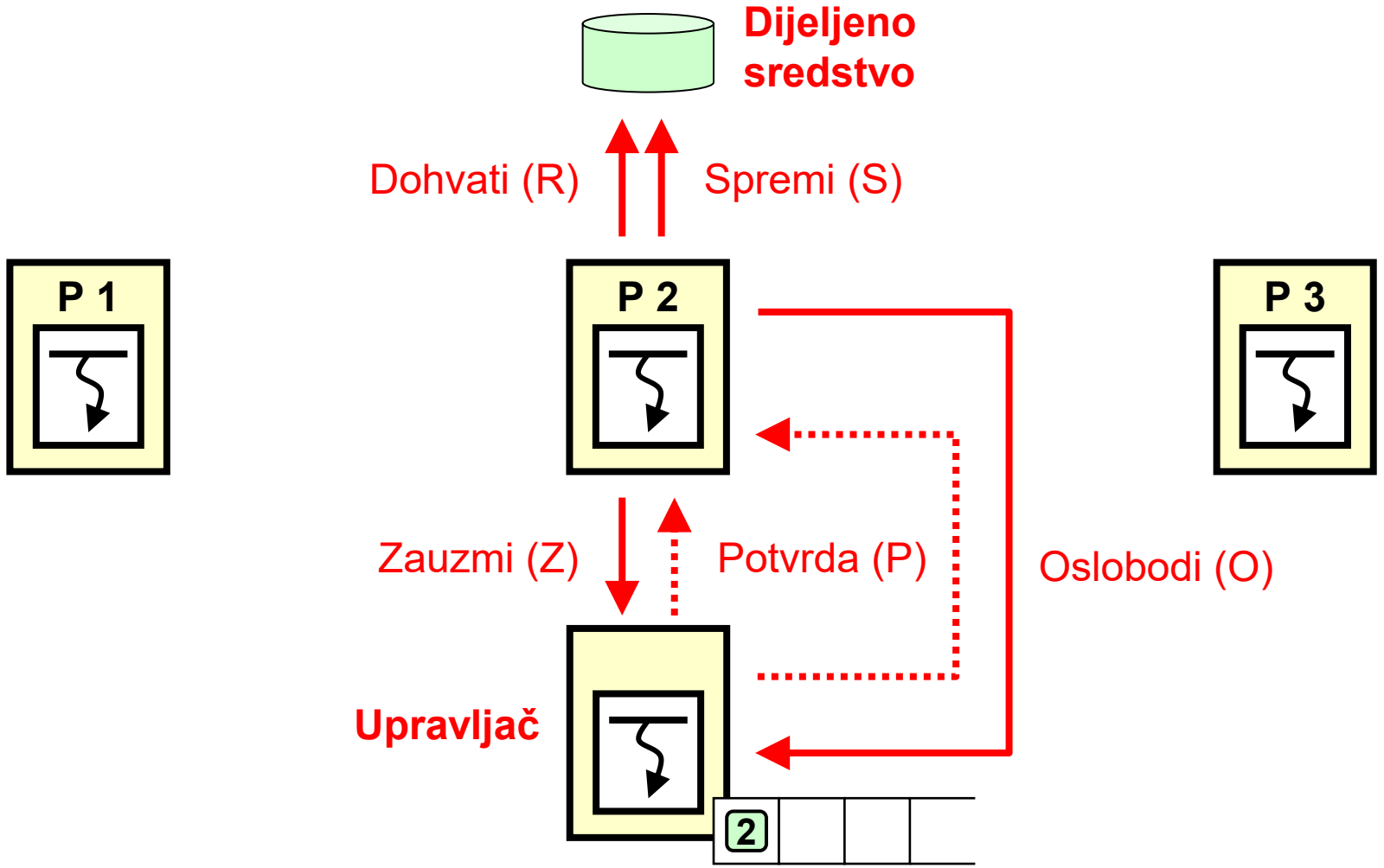
Sadržaj predavanja

- Potreba za sinkronizacijom procesa
- Primjena sata u jednoprocesorskoj okolini
- Primjena sata raspodijeljenoj okolini
- Sinkronizacija tijeka izvođenja procesa
- Međusobno isključivanje procesa
  - Središnji upravljač s repom čekanja
  - Decentralizirano međusobno isključivanje
  - Isključivanje zasnovano na primjeni prstena

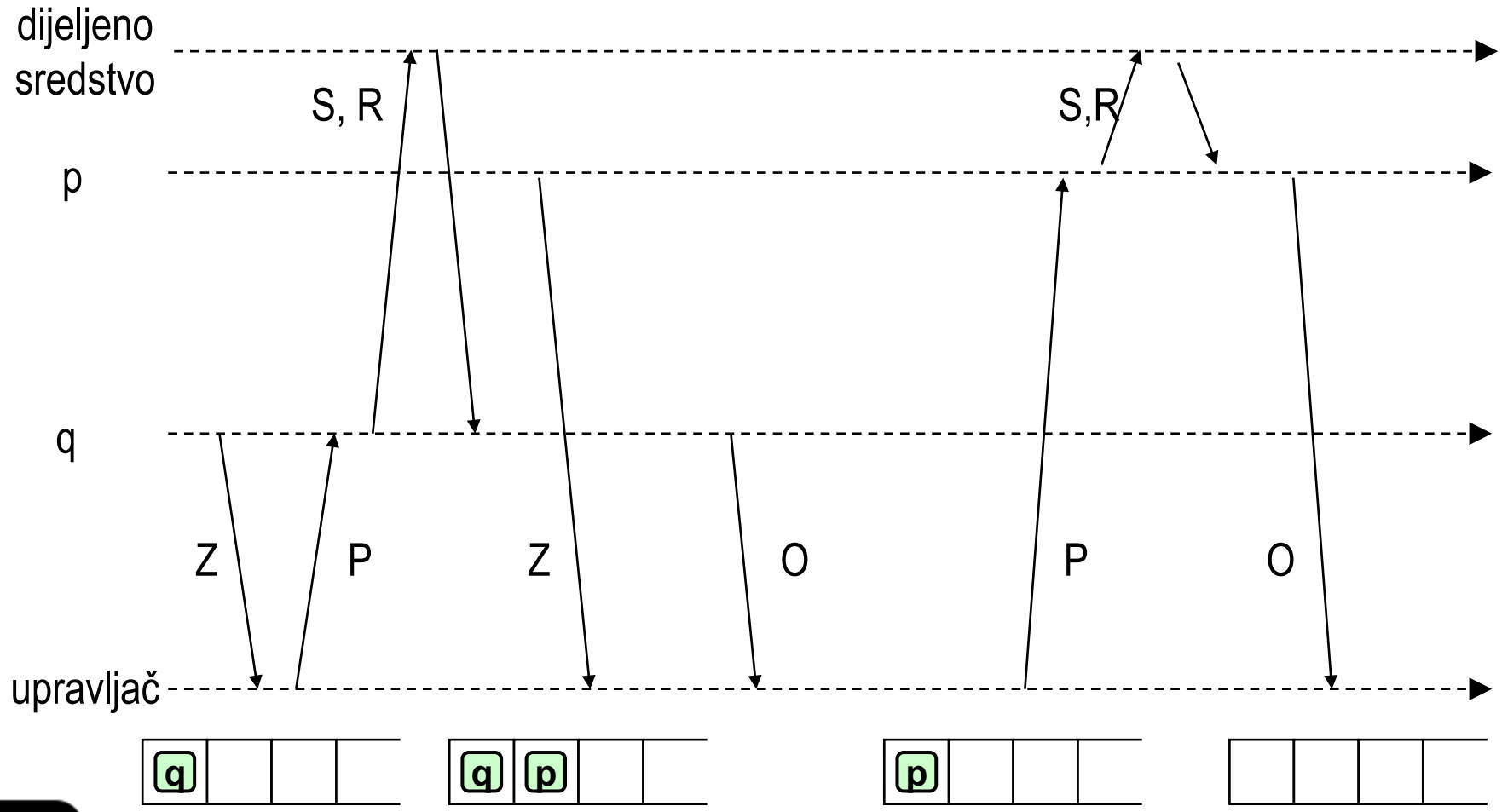
Međusobno isključivanje procesa

- **Središnji upravljač s repom čekanja**
  - Proces koji čuva stanje repa čekanja
  - Rep čekanja zasnovan na posluživanju zahtjeva prema redoslijedu prispijeća (*FIFO*)
- **Korisnici**
  - Procesi šalju poruke sa *zahtjevom za zauzimanje* (*Z*) tj. pristup sredstvu
  - Procesi ostvaruju pristup sredstvu nakon primitka poruke *potvrde* (*P*), te dohvaćaju (R) i/ili spremaju (S) podatke na dijeljeno sredstvo
  - Nakon završetka obrade, procesi otpuštaju zauzeto sredstvo slanjem poruke *oslobodi* (*O*)

Središnji upravljač s repom čekanja



Isključivanje putem središnjeg upravljača

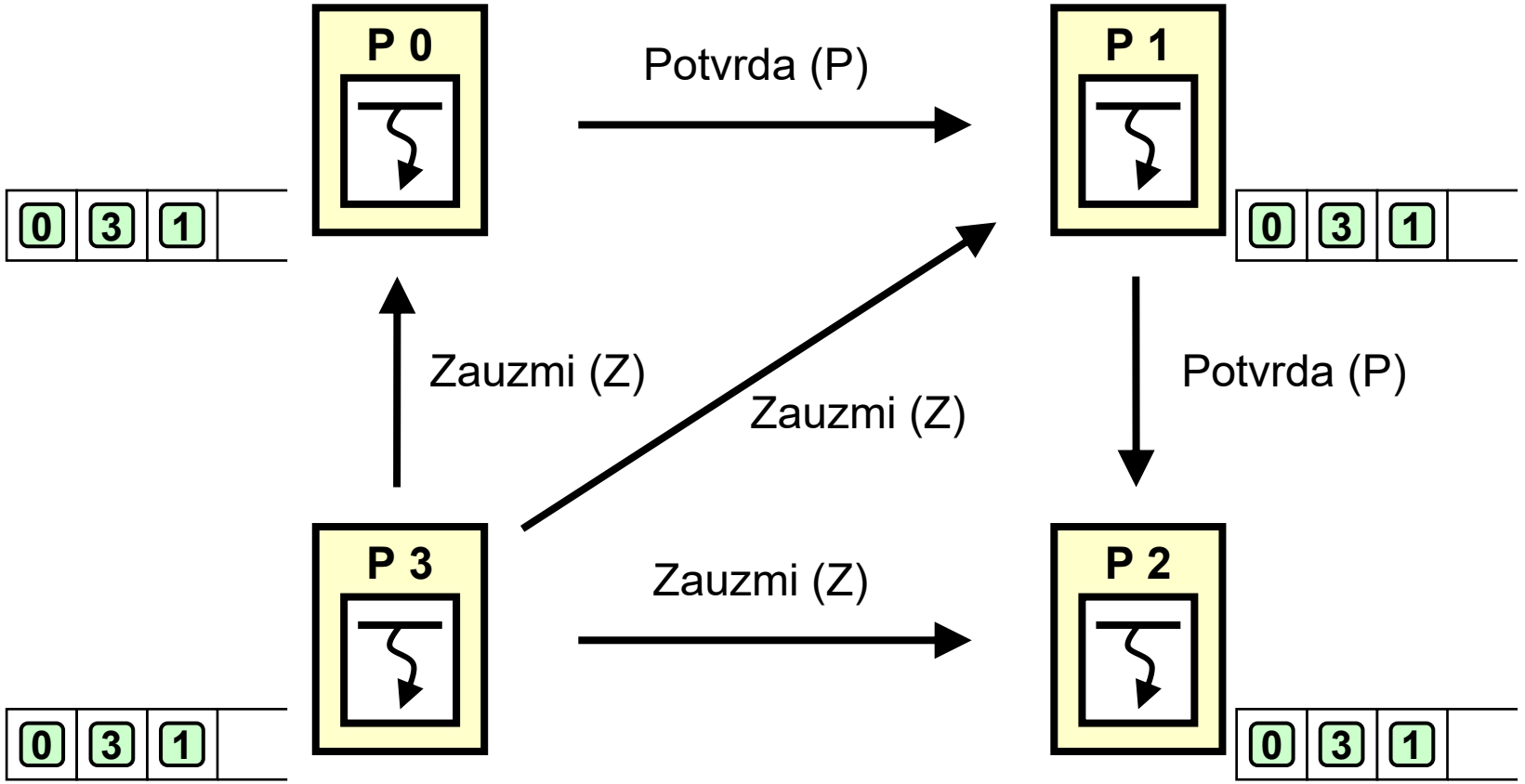


R – Dohvati
S – Spremi
Z – Zauzmi
P – Potvrda
O – Oslobodi

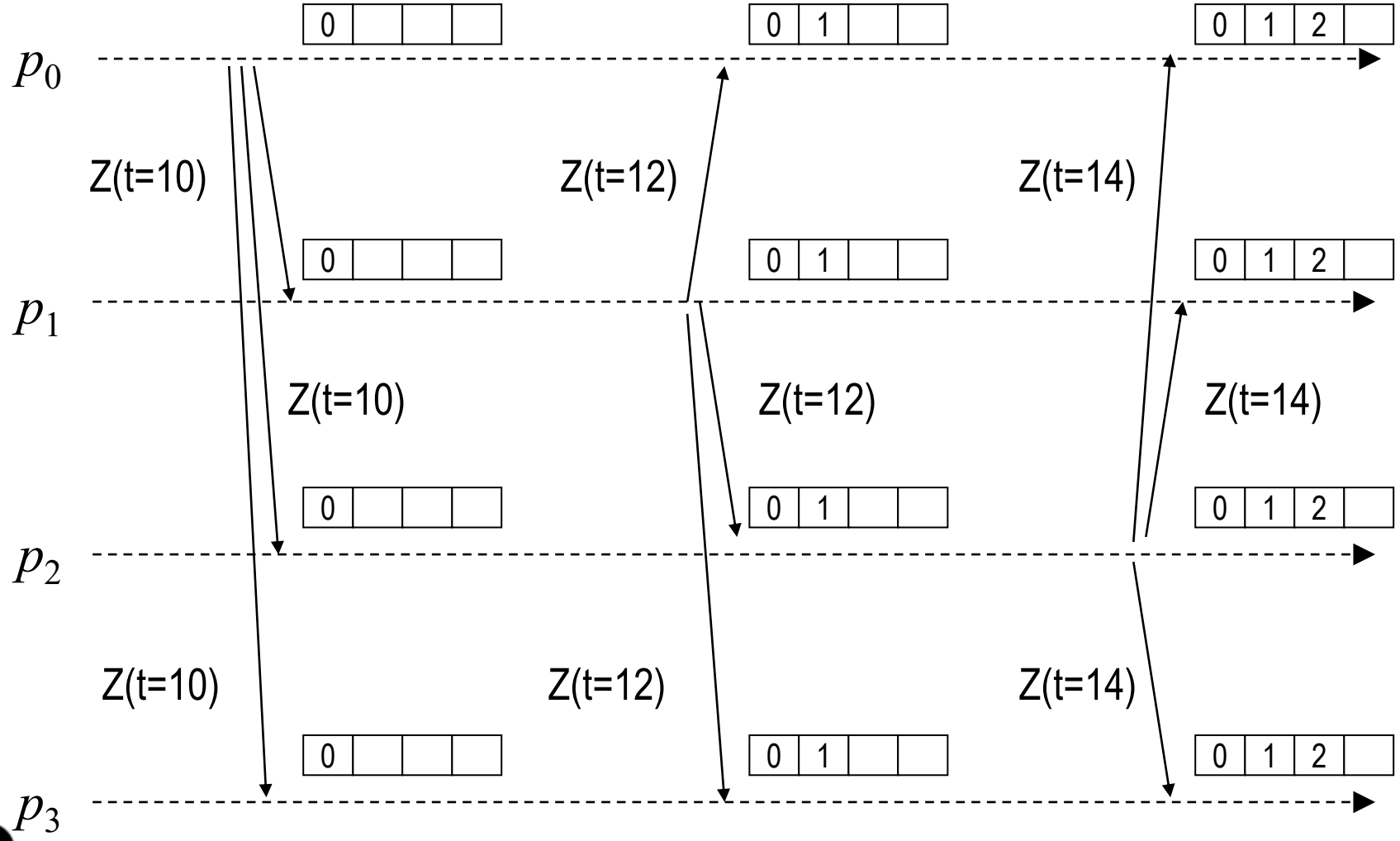
Decentralizirano međusobno isključivanje

- **Raspodijeljeni rep čekanja**
  - Svaki proces ima lokalni rep čekanja
  - Procesi razmjenjuju informacije potrebne za usklađivanje stanja svih repova čekanja u sustavu
- **Pretpostavke**
  - Svaki proces ima lokalni satni mehanizam koji je usklađen s ostalim procesima
  - Svaki zahtjev za pristup sredstvu uključuje oznaku trenutka u kojem je proces uputio zahtjev
  - Procesi ostvaruju pristup u skladu s vremenskim oznakama upućivanja zahtjeva

Elementi decentraliziranog isključivanja

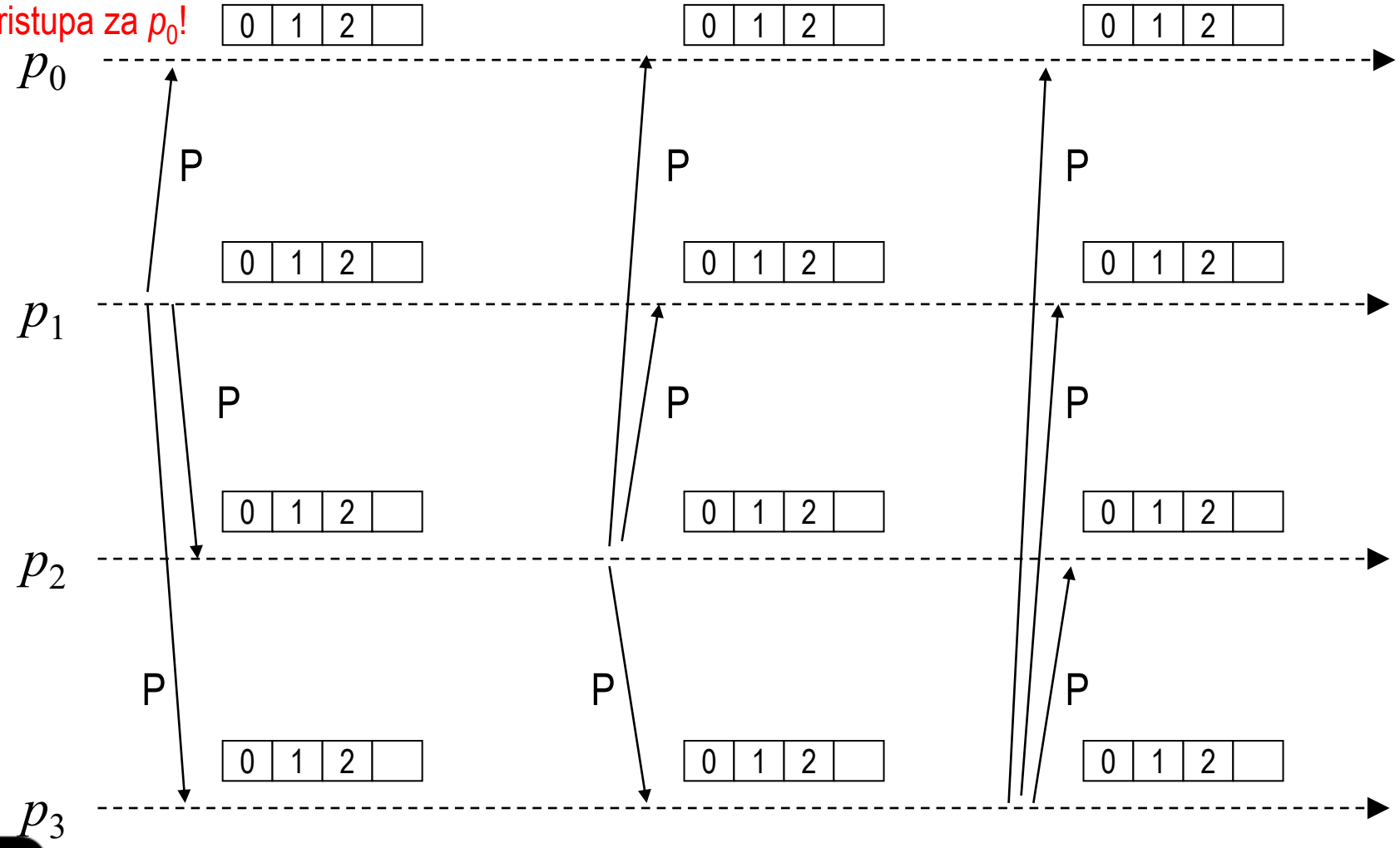


Decentralizirano isključivanje (1/4)



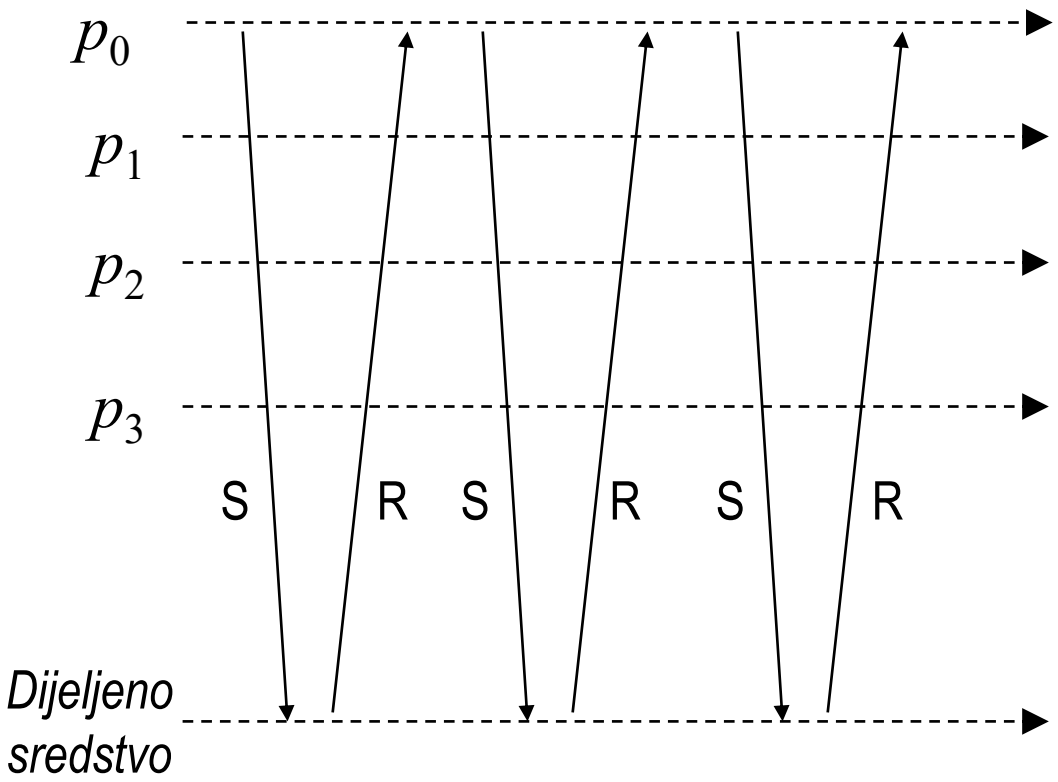
Z(t=x) – Zauzmi, vremenska oznaka x

Decentralizirano isključivanje (2/4)



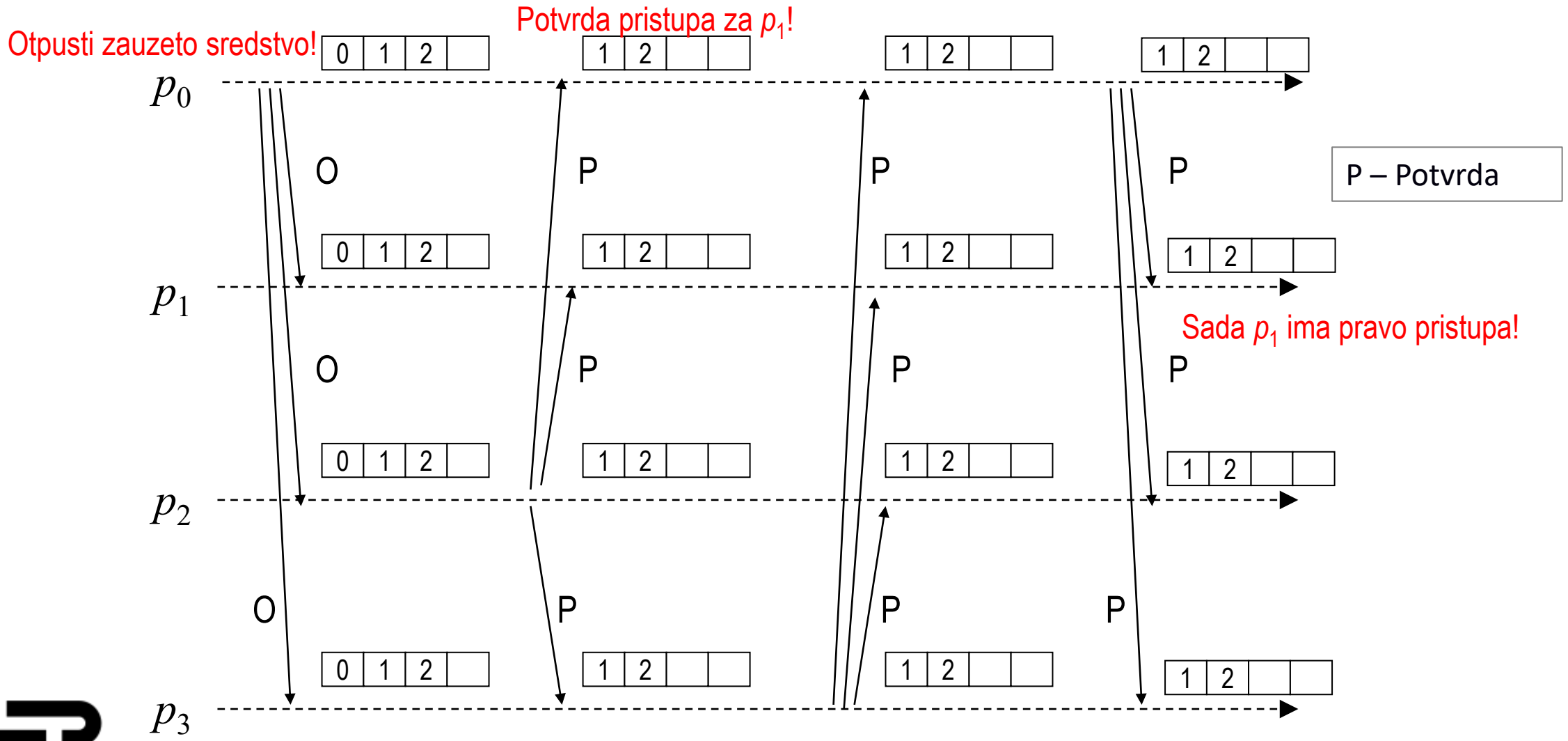
Decentralizirano isključivanje (3/4)

Pristup dijeljenom sredstvu!



R – Dohvati,
S – Spremi

Decentralizirano isključivanje (4/4)



Međusobno isključivanje primjenom prstena

- **Struktura prstena procesa**
  - Procesi su povezani u logičku mrežu zasnovanu na prstenu
  - Primjenjuju se identifikatori procesa za formiranje prstena
  - Duž prstena ostvaruje se razmjena jedne značke
  - I ovo je rješenje decentralizirano!
- **Pristup dijeljenom sredstvu**
  - Pristup ima samo proces koji u određenom trenutku ima značku
  - Nakon završetka pristupa, proces prosljeđuje značku susjednom procesu u prstenu

Akcije procesa u prstenu (1)

Proces *n* prima značku

1) Čitanje podataka iz spremnika
2) Pisanje podataka u spremnik
3) Prosljeđivanje značke procesu (n-1)

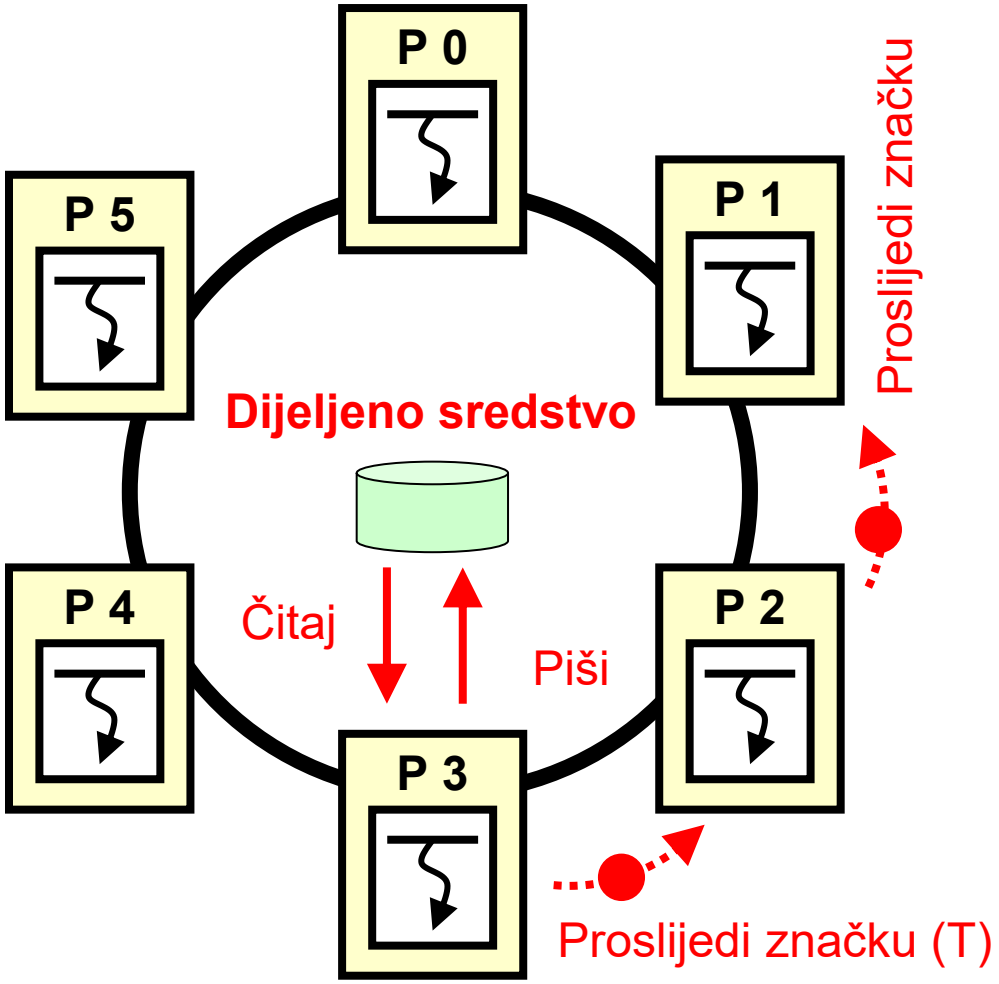
Proces (*n*-1) prima značku

4) Proces ne zahtjeva pristup spremniku
5) Prosljeđivanje značke procesu (n-2)

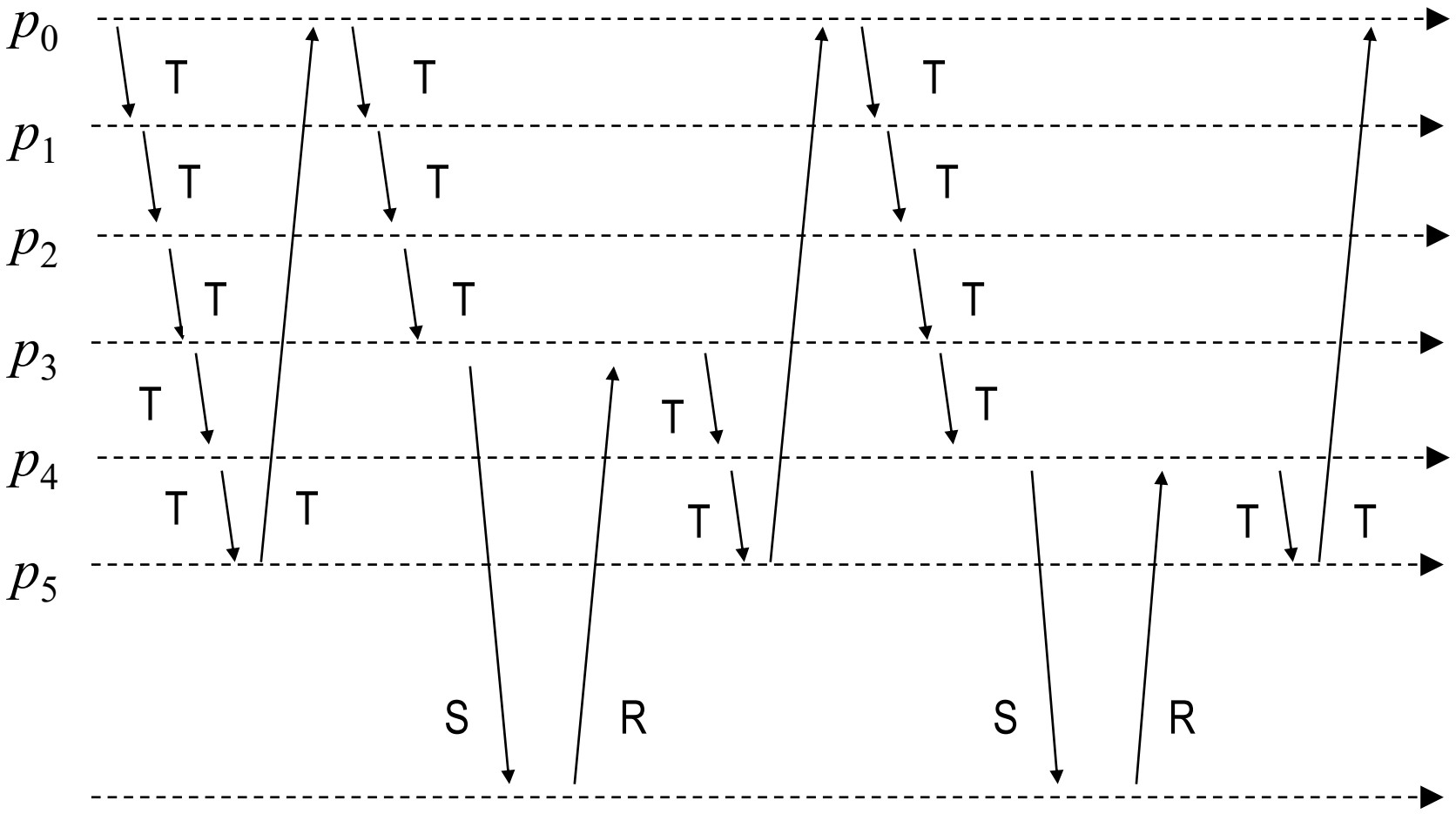
Proces (*n*-2) prima značku

6) …

Akcije procesa u prstenu (2)



Akcije procesa u prstenu (3)



T – Prijenos tokena,
S – Spremi,
R – Dohvati

Primjeri sustava za sinkronizaciju

Usluga ZooKeeper

- Usluga za pouzdanu koordinaciju tijeka izvođenja skupa procesa u raspodijeljenoj okolini
- Dodatne informacije: http://zookeeper.apache.org

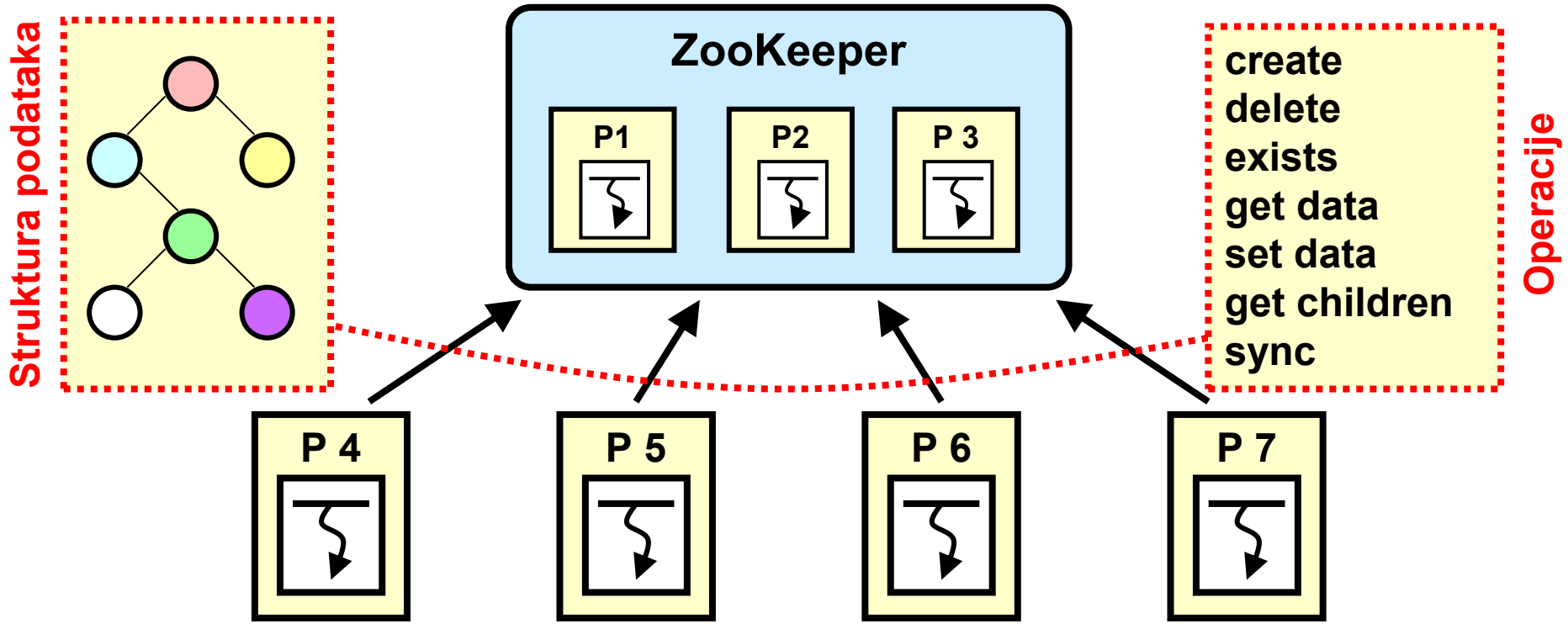
Okružje Hadoop

- Programsko okružje za provođenje paralelne obrade velike količine podataka (*Big Data*)
- Postoji potreba za sinkronizacijom procesa *map* i *reduce*
- Dodatne informacije: http://hadoop.apache.org

Usluga ZooKeeper (1)

Usluga opće namjene za koordiniranje skupa procesa u raspodijeljenoj okolini

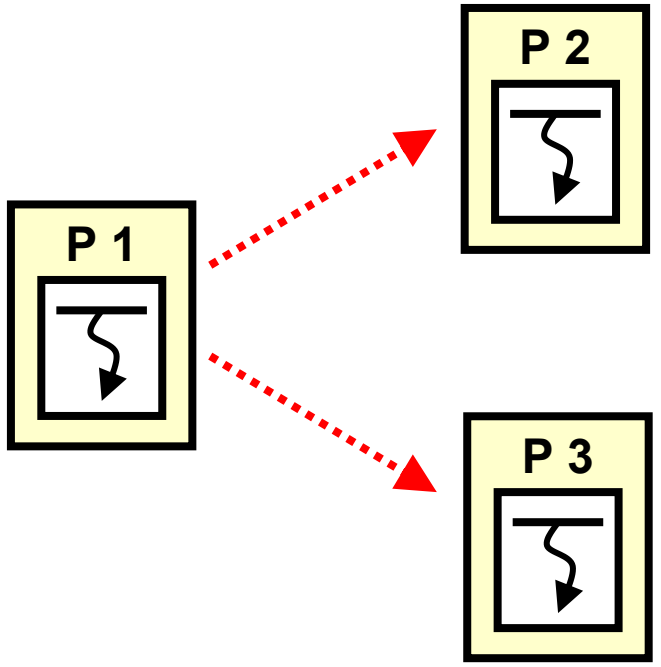
Imenovanje, sinkronizacija, upravljanje grupama, repovi, donošenje odluka, zaključavanje sredstava



Usluga ZooKeeper (2)

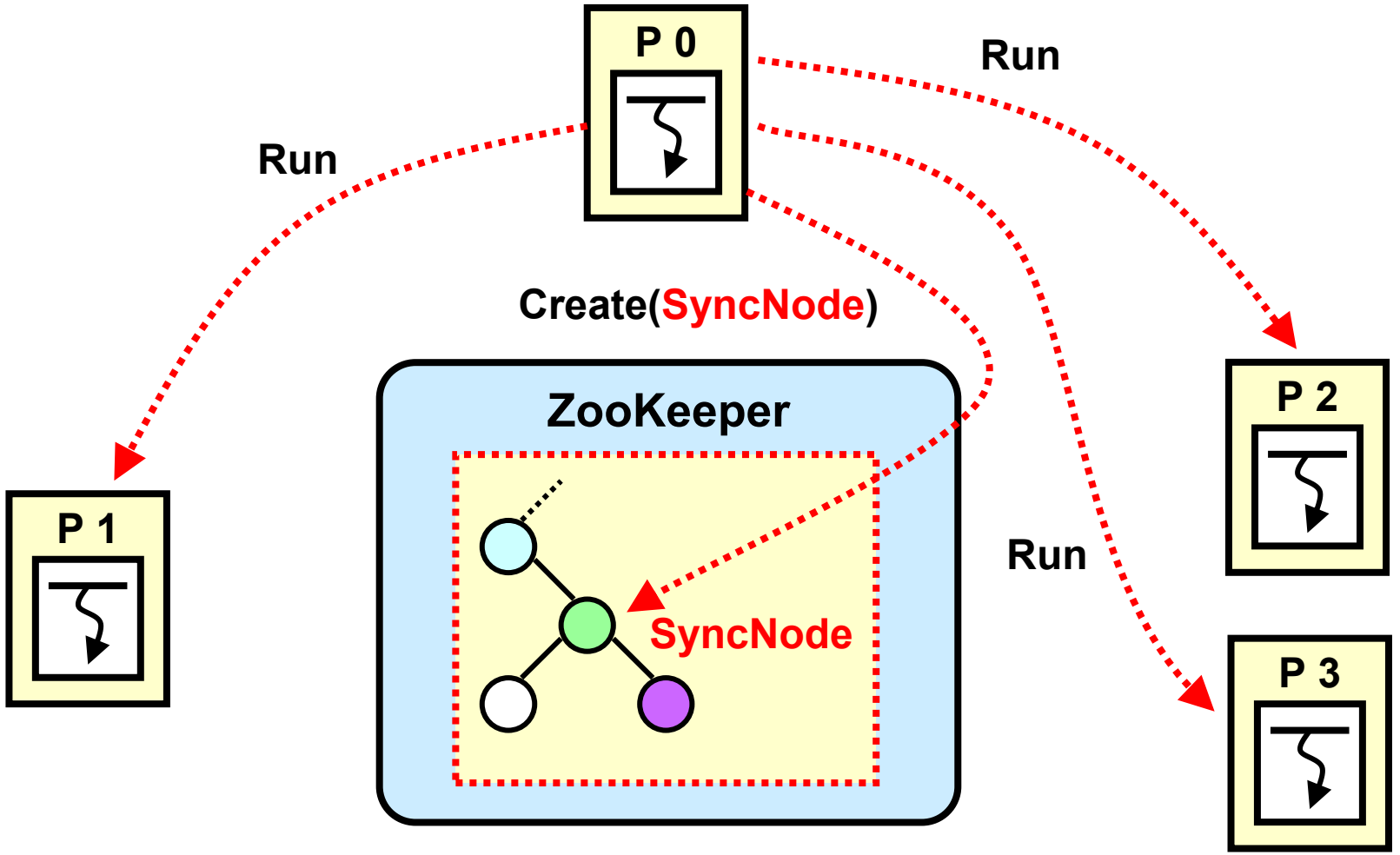
Primjer: Sinkronizacija procesa

Procesi P2 i P3 započinju s izvođenjem tek nakon što je proces P1 završio s izvođenjem



Usluga ZooKeeper (3)

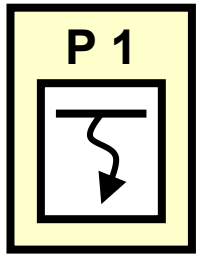
P0 je upravljački proces koji stvara sinkronizacijski čvor SyncNode

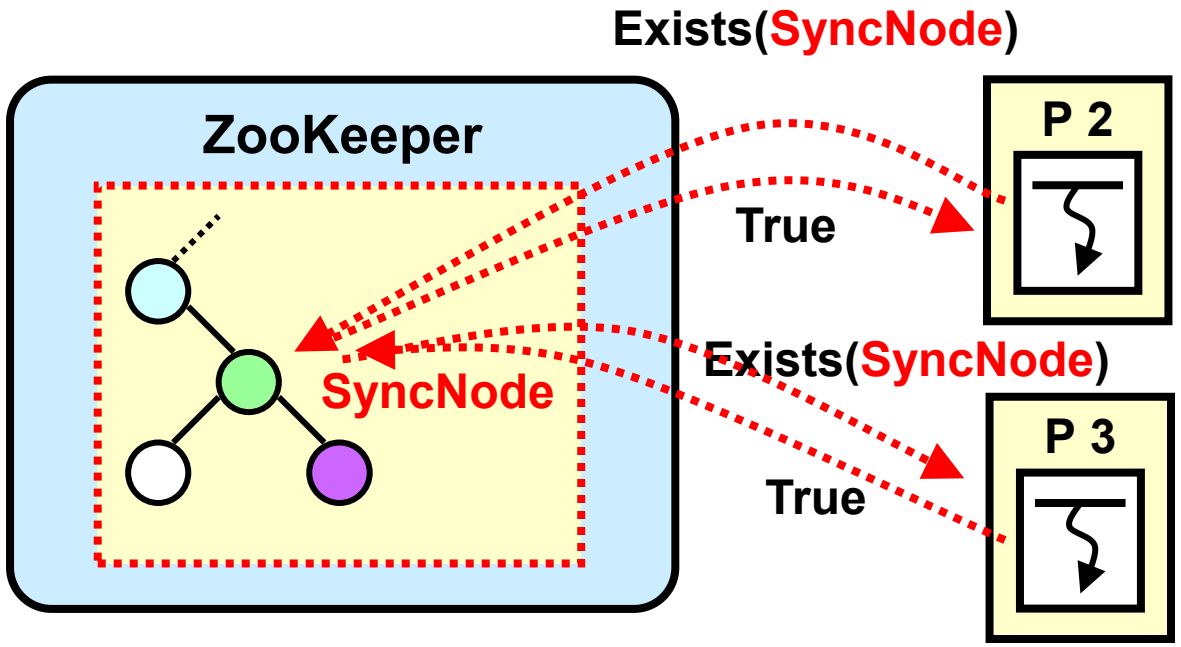


Usluga ZooKeeper (4)

Procesi P2 i P3 ispituju postoji li sinkronizacijski čvor SyncNode

- Ako čvor postoji, čekaju na dojavu o brisanju čvora
- jer kada se čvor izbriše, oni započinju izvođenje

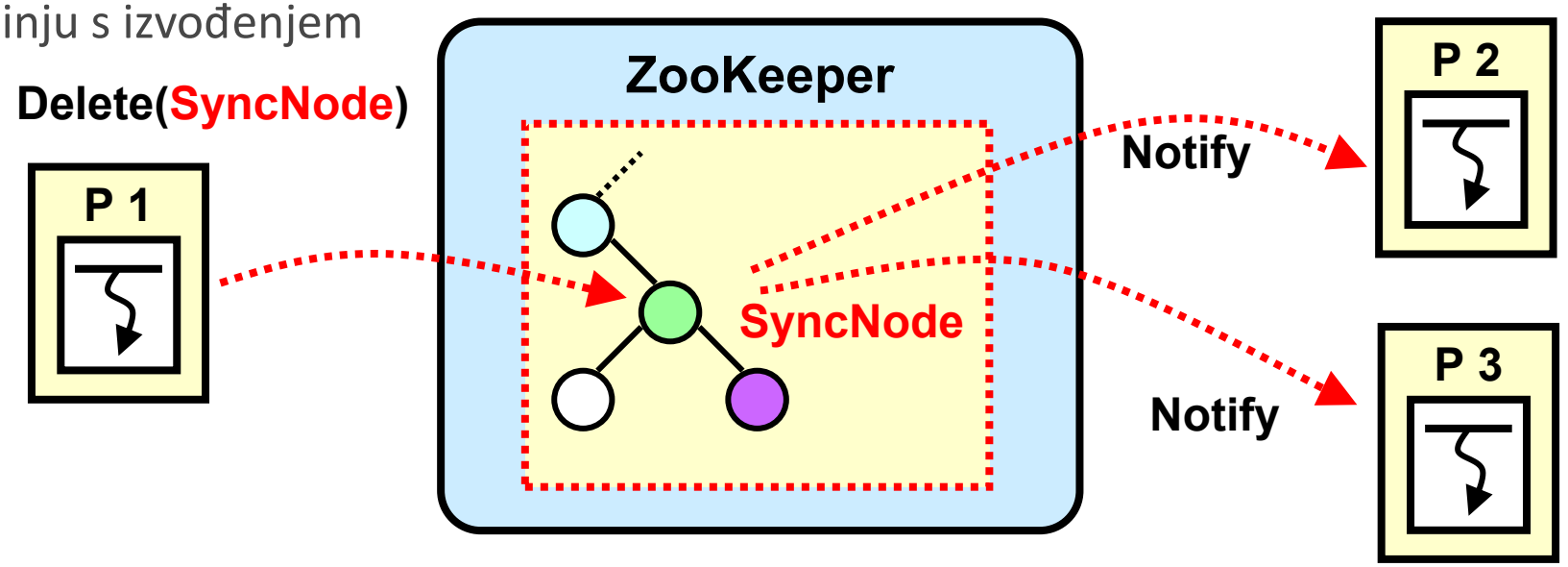




Usluga ZooKeeper (5)

Proces P1 završava s izvođenjem

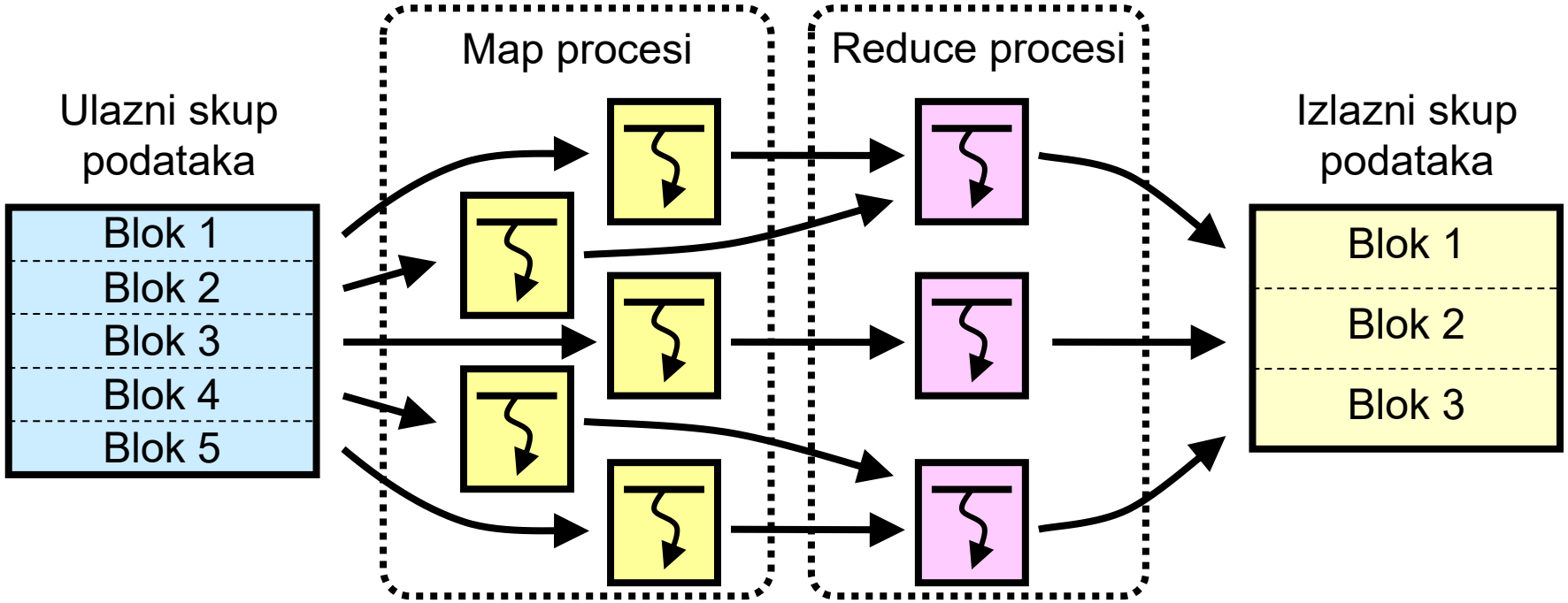
- Briše sinkronizacijski čvor **SyncNode**
- Usluga ZooKeeper dojavljuje brisanje čvora i stoga
- procesi P2 i P3 započinju s izvođenjem



Okružje Hadoop

MapReduce: analiza i obrada velikih skupova podataka

- Ulazni skup podataka dijeli se na blokove koje paralelno obrađuju nezavisni procesi
- Rezultati obrade grupiraju se u odredišni skup podataka



Literatura

1. Maarten van Steen, Andrew S. Tanenbaum (2017.), *Distributed Systems 3rd edition*, Createspace Independent Publishing Platform poglavlje 14 (bez 14.5 i 14.6)
2. H. Attya, J. Welch: **“Distributed Computing: Fundamentals, Simulations, and Advanced Topics"**, Wiley, 2004. (Poglavlje: *Causality and Time*)
3. N. A. Lynch: **“Distributed Algorithms"**, Morgan Kaufmann, 1997. (Poglavlje: *Logical Time*)

Dodatne informacije

- Kolegij na FER-u
  - **Raspodijeljena obrada velikih podataka**, 3. semestar
  - https://www.fer.unizg.hr/predmet/rovp
  - Sadržaj kolegija
    - Obrada velike količine podataka: MapReduce
    - Obrada nestrukturiranog teksta
    - Obrada toka podataka