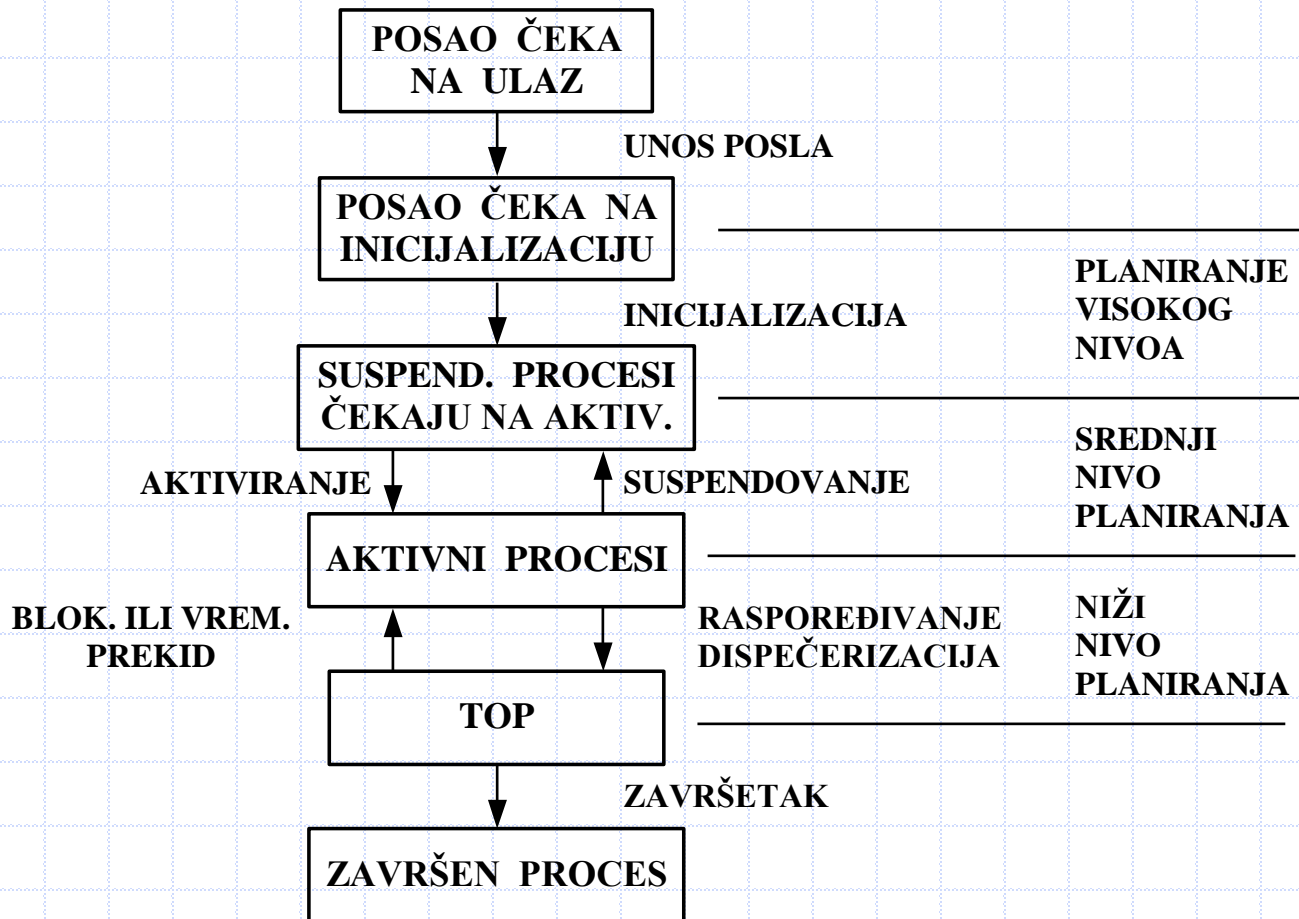


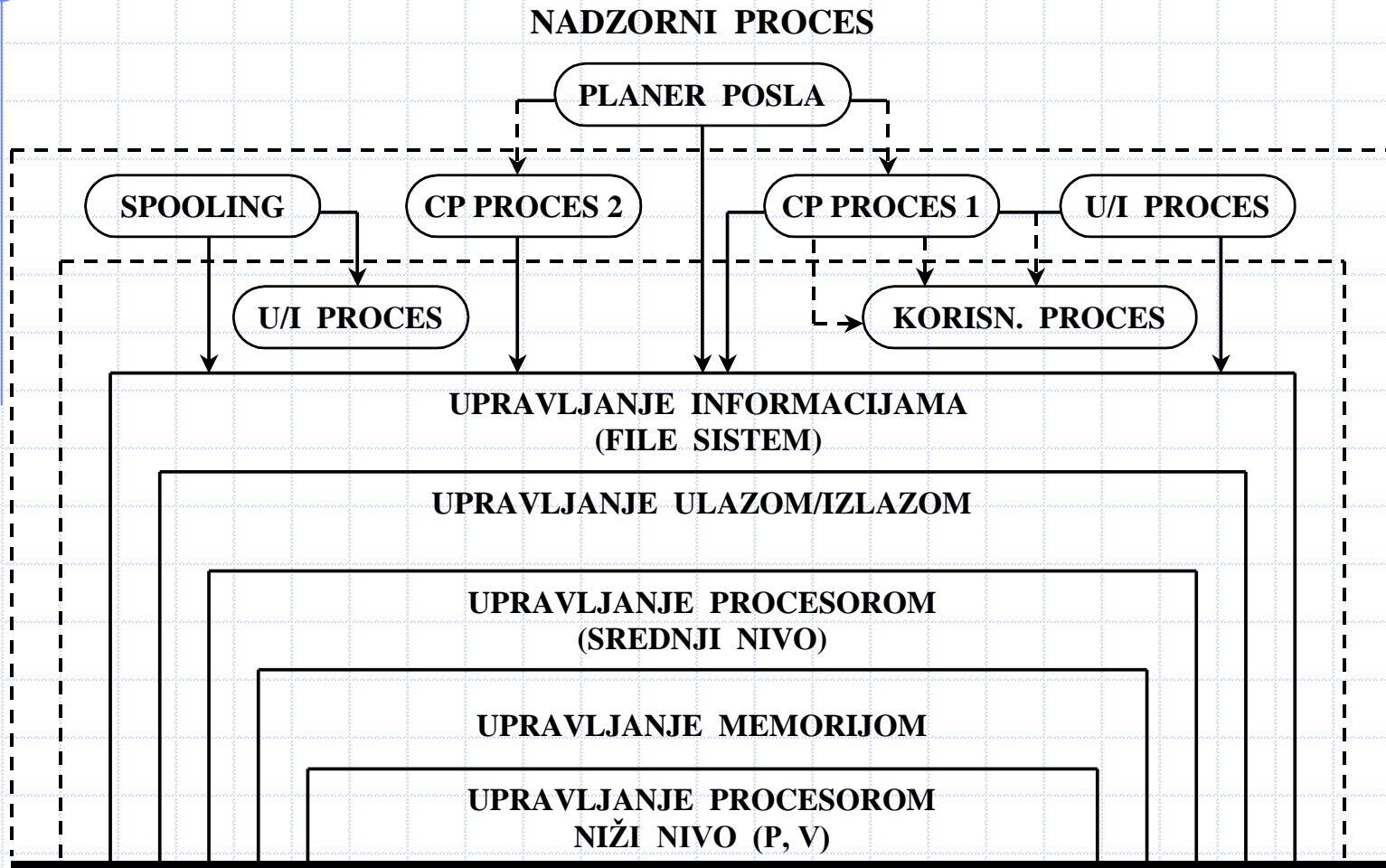
UKLOP PROCESOROM

- ◆ Algoritmi planiranja
- ◆ Organizacija planera
- ◆ Međusobno blokiranje

Nivoi planiranja



Nivoi planiranja u odnosu na planiranje posla



PLANER POSLOVA (1/2)

- ◆ prati stanje svih poslova, označavanjem koji posao traži opsluživanje kao i stanje svih koji su opsluživani (pripravni, obrađivani ili blokirani)
- ◆ izabira politiku ulaska poslova u sistem (prelazak iz "uneto" u stanje pripravnosti), a na bazi karakteristika tipa prioriteta, traženih resursa ili ravnoteže opterećenosti sistema

PLANER POSLOVA (2/2)

- ◆ dodeljuje potrebne resurse planiranom poslu, pozivanjem rukovaoca memorije, perifernih organa i procesora
- ◆ oslobađa zauzete resurse po obavljenom poslu

Kontrolni blok posla

IDENTIFIKACIJA POSLA

TEKUĆE STANJE

PRIORITET

VREMENSKA OCENA

OSTALO

Ciljevi planiranja

- ◆ procesi moraju biti tretirani ravnomerno
- ◆ maksimizirati propusnost sistema
- ◆ maksimizirati broja interaktivnih korisnika
- ◆ poslovi moraju da se izvršavaju u tačno određeno vreme
- ◆ minimiziranje praznog hoda resursa

PRIORITETI procesa

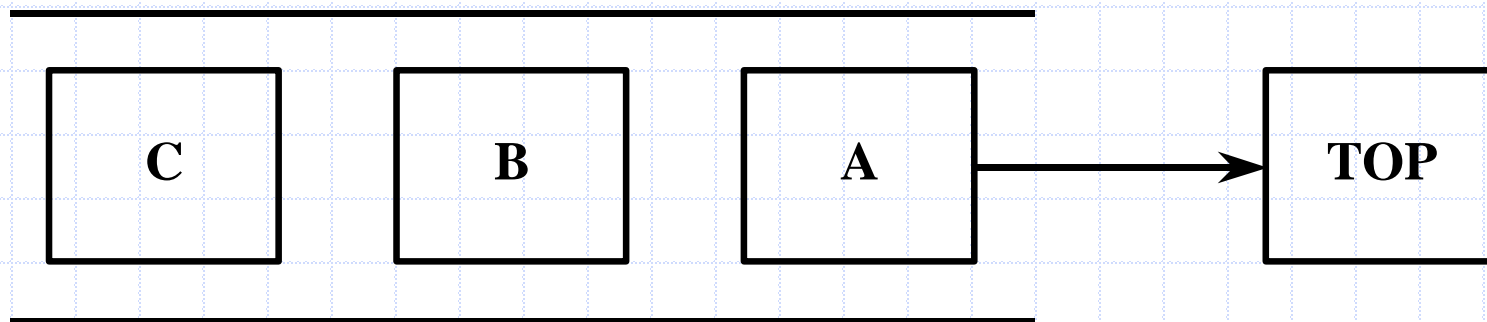
- ◆ Mogu da se dodele automatski od strane sistema ili se mogu spolja zadati.
- ◆ Mogu biti statički ili dinamički.

ALGORITMI PLANIRANJA PROCESA

- ◆ PRVI U LISTU PRVI IZ LISTE
- ◆ REDOM U KRUG (ROUND ROBIN-RR)
- ◆ PLANIRANJE DAVANJEM PREDNOSTI KRAĆIM POSLOVIMA
- ◆ PLANIRANJA NA BAZI NAJKRAĆEG PREOSTALOG VREMENA
- ◆ PLANIRANJE NA BAZI NAJVEĆEG ODNOSA VREMENA ODZIVA: $p = (t_w + t_s) / t_s$
- ◆ VIŠENIVOVSKI REDOVI ČEKANJA SA POVRATNOM SPREGOM

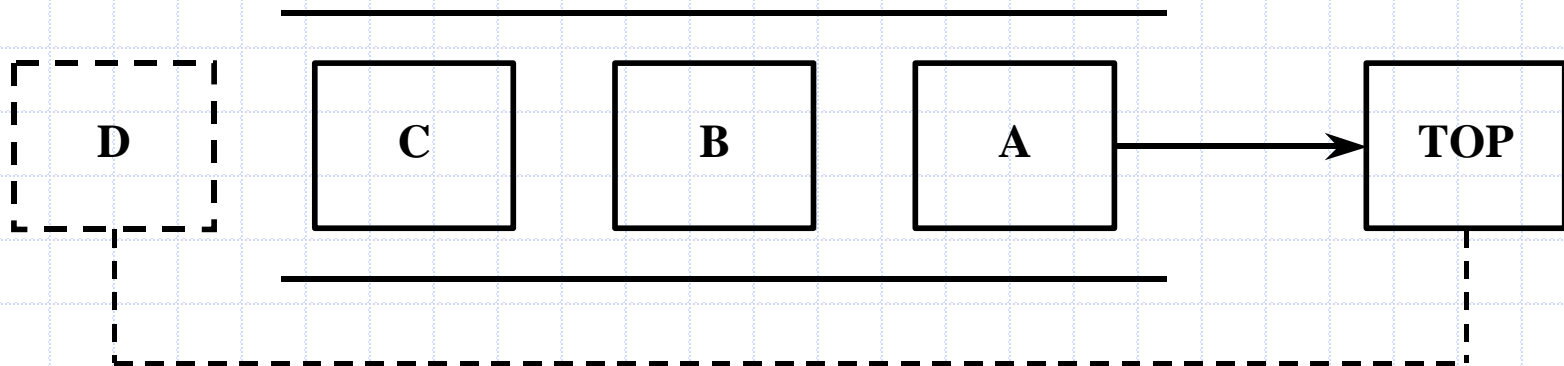
Algoritam prvi u listu prvi iz liste

LISTA PRIPRAVNOSTI

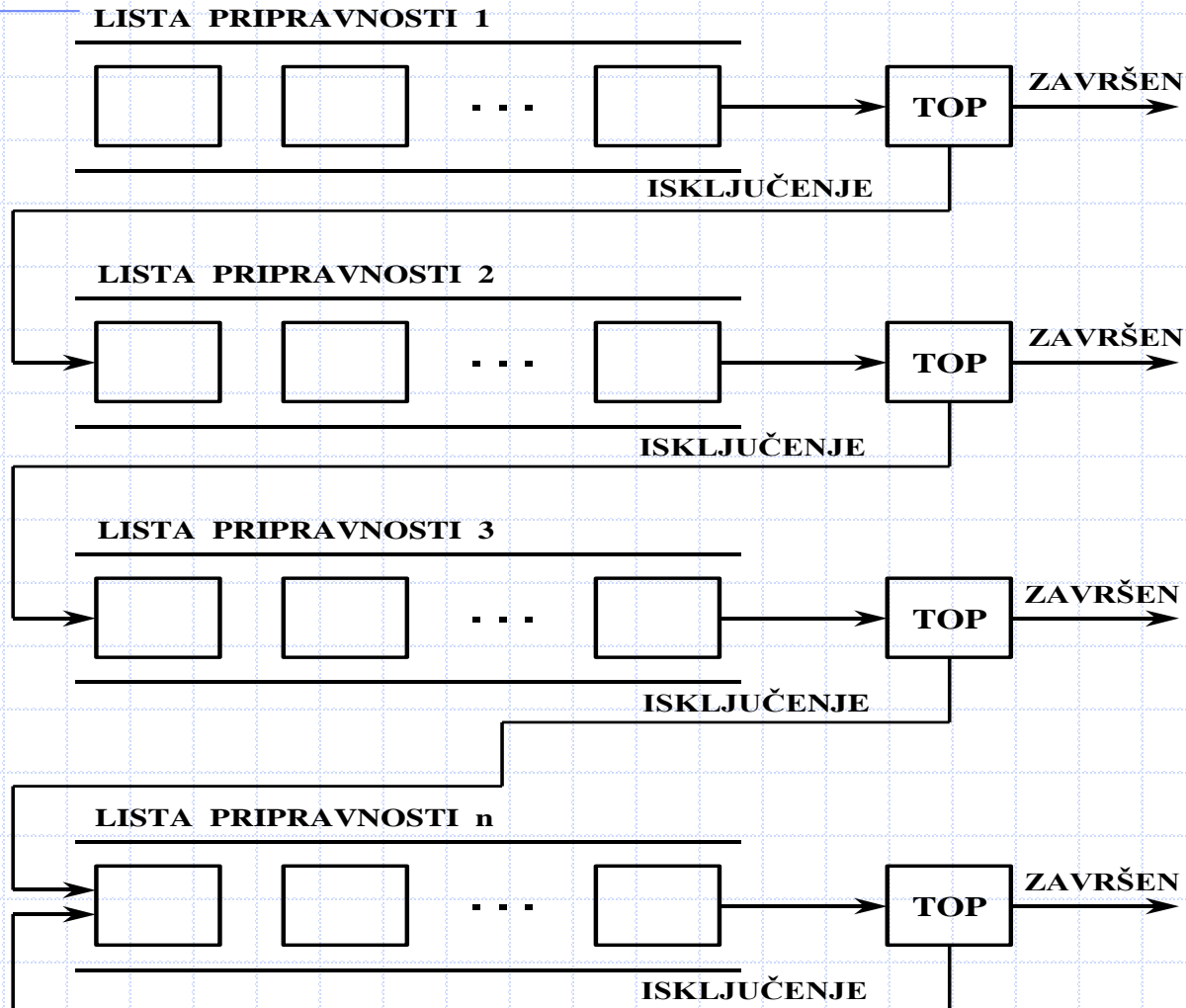


Planiranje - REDOM U KRUG (RR)

LISTA PRIPRAVNOSTI



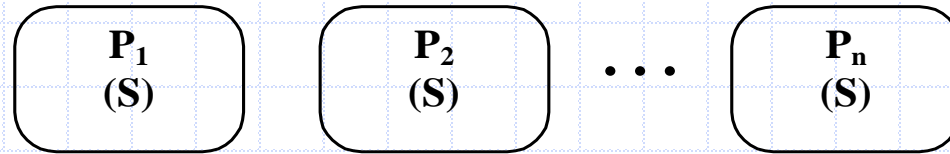
Višenivojski redovi čekanja



ORGANIZACIJA PLANERA PROCESA

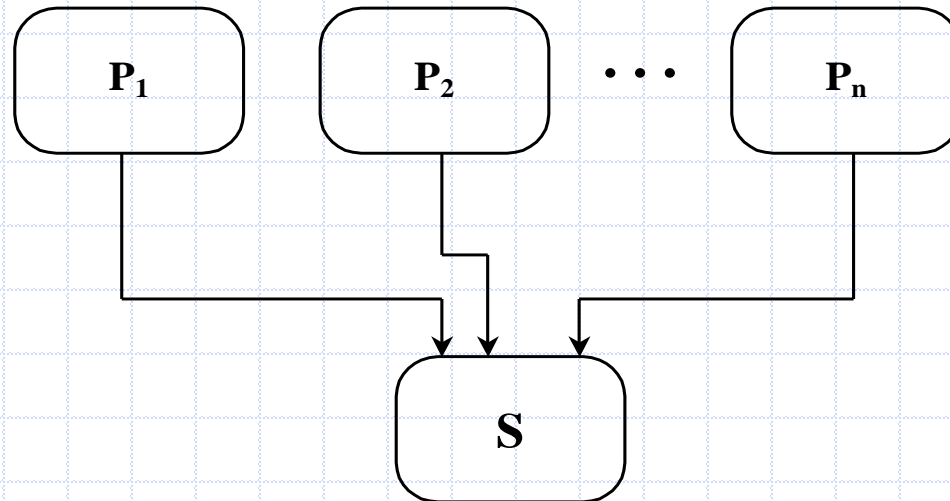
- ◆ Distribuiran planer
- ◆ Centralizovan planer

Dve organizacije planera CP-a



P_i : proces
 S : planer

a)



b)

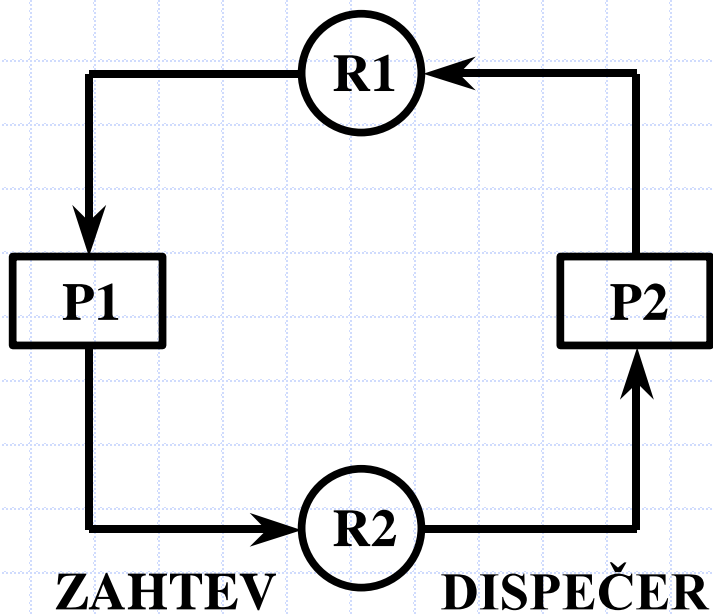
MEĐUSOBNO BLOKIRANJE PROCESA (DEADLOCK)

- ◆ Za proces u multiprogramskom sistem se kaže da je u stanju međusobnog blokiranja ukoliko čeka na neki poseban događaj koji se ne može pojaviti.

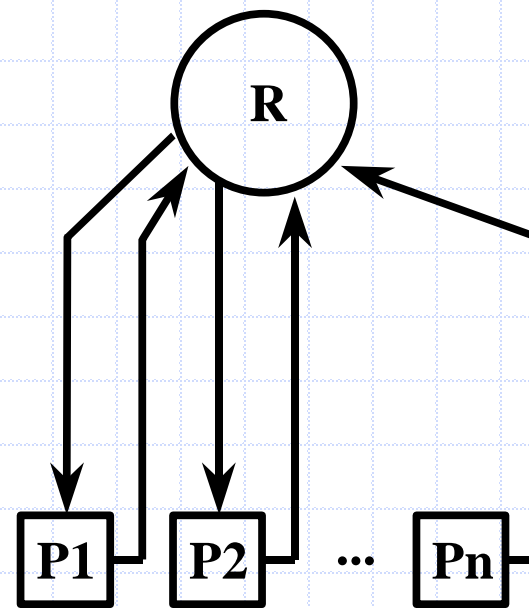
Tehnike za borbu protiv međusobnog blokiranja

- ◆ SPREČAVANJE
- ◆ IZBEGAVANJE
- ◆ DETEKCIJA
- ◆ OPORAVAK

Primeri međusobnog blokiranja

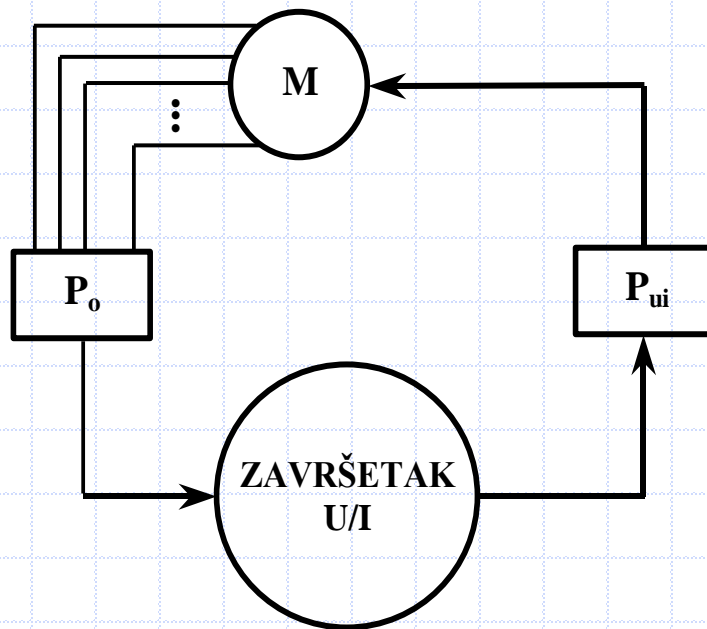


a)

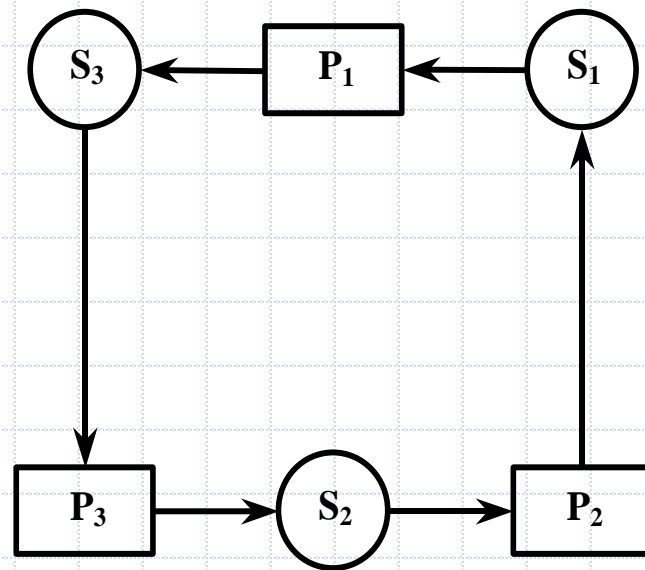


b)

Još primera međusobnog blokiranja



a)



b)

Neophodni uslovi za međusobno blokiranje

- ◆ a) proces zahteva isključivu kontrolu nad resursima koje traži (*uslov međusobne isključivosti*)
- ◆ b) proces drži njemu već dodeljene resurse za vreme čekanja na dodatne resurse (*uslov čekanja*)
- ◆ c) resurs ne može da bude uzet procesu koji ga drži sve dok je resurs potreban radi završetka procesa (*uslov nesmenjivosti*)
- ◆ d) postoji kružni lanac procesa u kome svaki proces drži jedan ili više resursa koji se traže od strane sledećeg procesa u lancu (*uslov kružnog čekanja*)

Sprečavanje međusobnog blokiranja

- ◆ Svaki proces može da traži sve njemu potrebne resurse odjednom i ne može da nastavi izvršenje sve dotle dok mu svi oni ne budu dodeljeni.
- ◆ Ukoliko proces koji drži neke resurse odustane od zahteva, mora osloboditi svoje polazne resurse i ako je potrebno zatražiti ih sve ponovo i sa dodatnim resursima (izbegavanje nesmenjivosti resursa).
- ◆ Uslovljavanje linearnog traženja tipa resursa za sve procese, tj. ako proces ima dodeljene resurse datog tipa može tražiti samo one resurse tog tipa koji slede po redosledu.

Izbegavanje blokiranja: Bankarov algoritam

- ◆ Ako su se pojavili potrebni uslovi za blokiranje, moguće ga je izbeći pažljivim rasporedom resursa koji se dodeljuju.
- ◆ Jedan od takvih algoritama je Bankarov Algoritam, ovako nazvan stoga što se odnosi na bankara koji pozajmljuje i prima isplate iz definisanog izvora kapitala.

Primer sigurnog stanja

STANJE 1.

| | Tekuća pozajmnica | Maksimalna potreba |
|-------------------|----------------------|-----------------------|
| KORISNIK (1) | 1 | 4 |
| KORISNIK (2) | 4 | 6 |
| KORISNIK (3) | 5 | 8 |
| NA RASPOLAGANJU 2 | | |

Primer nesigurnog stanja

STANJE 2.

| | Tekuća pozajmnica | Maksimalna potreba |
|-------------------|----------------------|-----------------------|
| KORISNIK (1) | 8 | 10 |
| KORISNIK (2) | 2 | 5 |
| KORISNIK (3) | 1 | 3 |
| NA RASPOLAGANJU 1 | | |

Primer prelaza iz bezbednog stanja u nebezbedno: STANJE 3 je bezbedno

STANJE 3.

| | Tekuća pozajmnica | Maksimalna potreba |
|-------------------|----------------------|-----------------------|
| KORISNIK (1) | 1 | 4 |
| KORISNIK (2) | 4 | 6 |
| KORISNIK (3) | 5 | 8 |
| NA RASPOLAGANJU 2 | | |

Primer prelaza iz bezbednog stanja u nebezbedno: STANJE 4 nije bezbedno

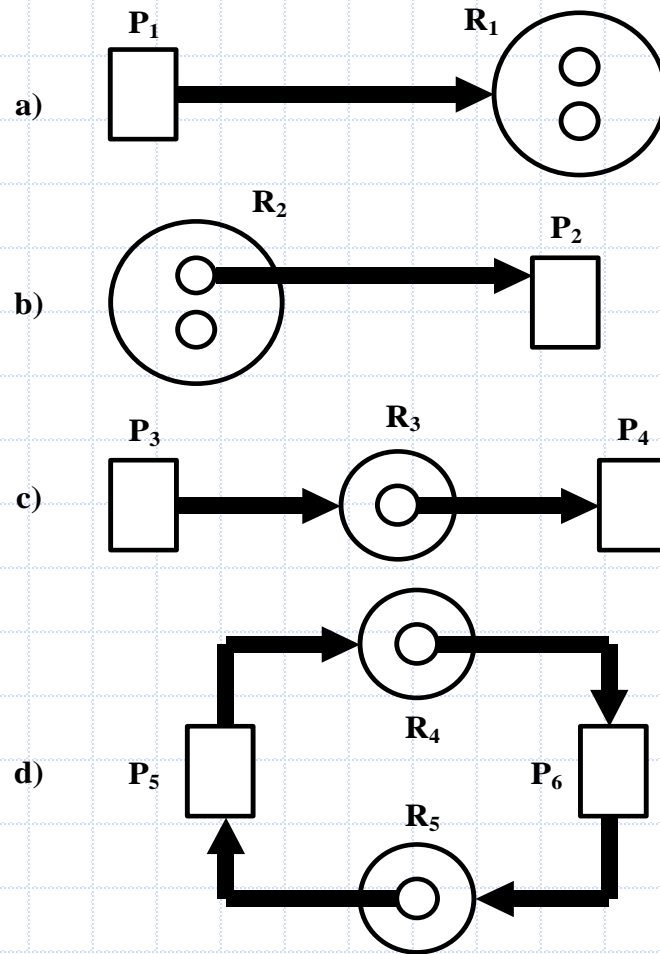
STANJE 4.

| | Tekuća pozajmnica | Maksimalna potreba |
|-------------------|----------------------|-----------------------|
| KORISNIK (1) | 1 | 4 |
| KORISNIK (2) | 4 | 6 |
| KORISNIK (3) | 6 | 8 |
| NA RASPOLAGANJU 1 | | |

Detekcija međusobnog blokiranja

- ◆ Predstavlja proces otkrivanja stanja međusobnog blokiranja i identifikacije procesa i resursa koji su uključeni u međusobno blokiranje.

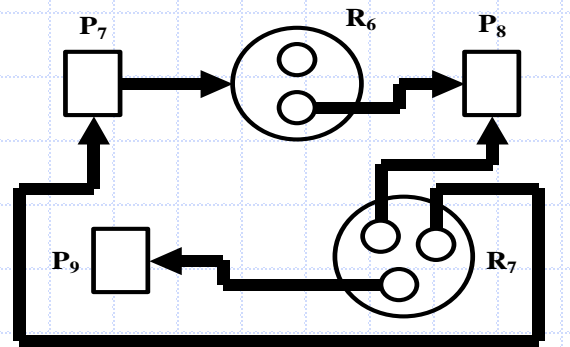
Graf dodele resursa



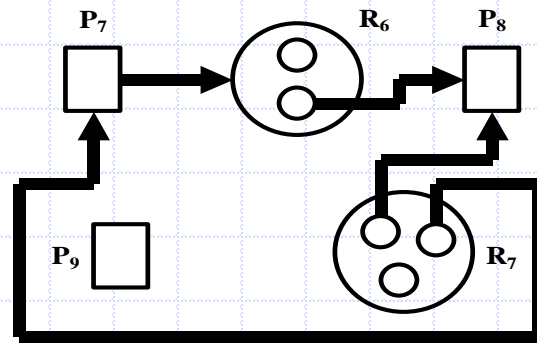
Redukovani graf dodele resursa

- ◆ Koristi se za određivanje da li postoji situacija međusobnog blokiranja i jedna je od tehnika koja je korisna u detekciji međusobnog blokiranja.

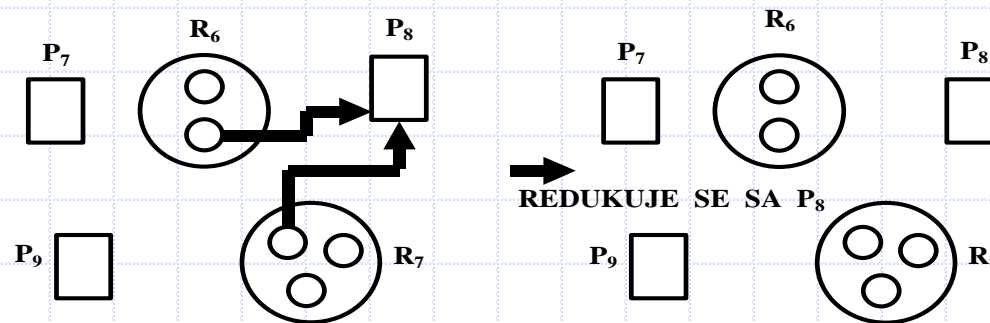
Ilustracija procesa redukcije grafa



REDUKUJE SE SA P_9



REDUKUJE SE SA P_7



REDUKUJE SE SA P_8

Oporavak od međusobnog blokiranja

- ◆ Pošto sistem uđe u stanje blokiranja, iz njega može da izađe izuzimanjem jednog ili više neophodnih uslova blokiranja.
- ◆ Time neki od procesa gube ili deo ili ceo posao koji je dotle urađen.

Višeprocorsko raspoređivanje

- ◆ Odeljak 3.4 u knjizi.
- ◆ Pročitati za lektiru.