



Zaglavljivanje procesa

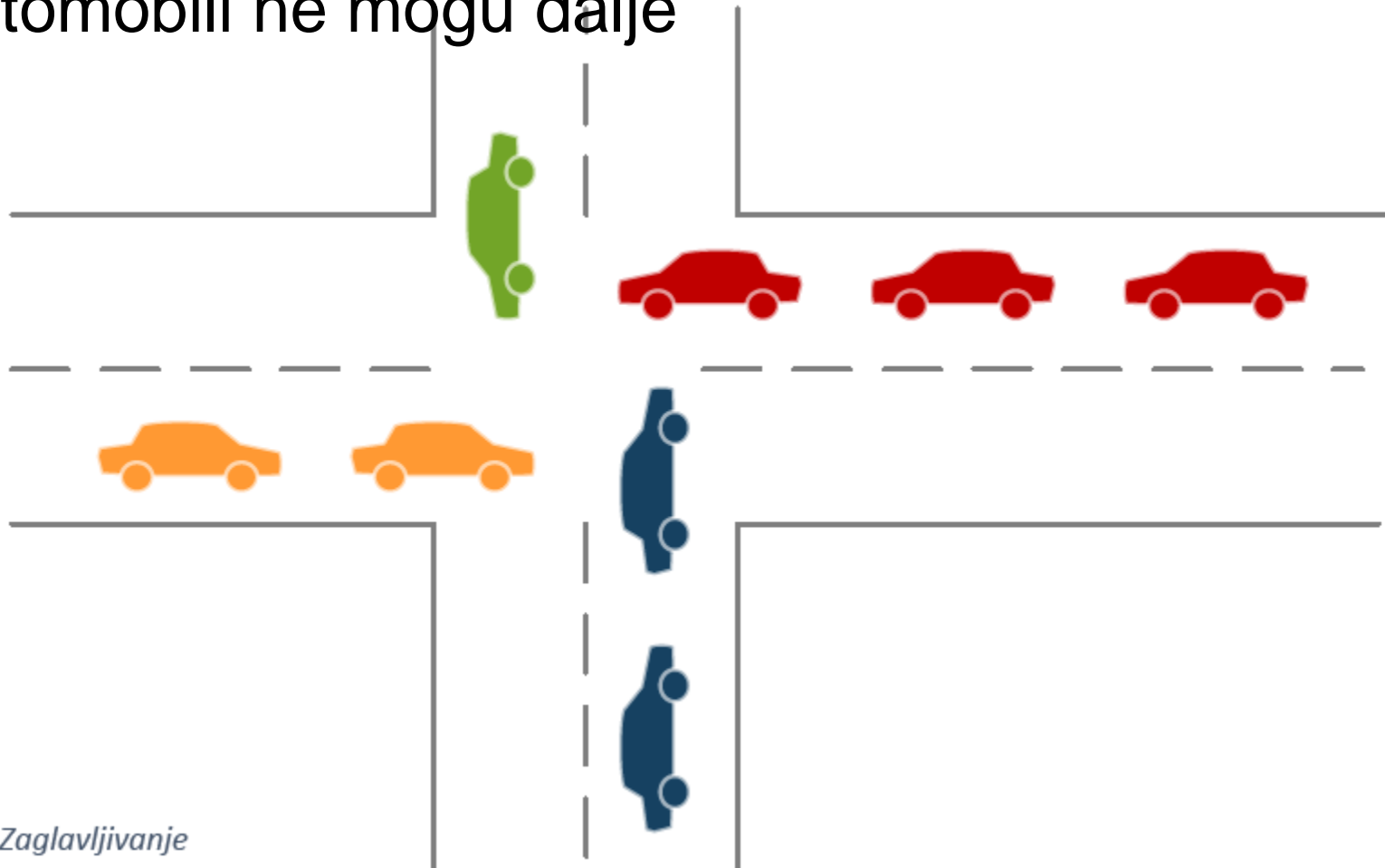
Vladimir Filipović

Zaglavljivanje

- Situacije u kojima dva ili više međusobno zavisnih procesa čekaju (blokirani) na resurse koje nikada neće dobiti se nazivaju **smrtonosno blokiranje** (deadlock, uzajamno blokiranje).
- Ovakvo zaglavljivanje je trajno i za posledicu često ima zaglavljivanje čitavog sistema.
- Ako sistem sadrži procese koji su zaglavljeni smatra se da je i on je zaglavljen.

Zaglavljivanje (2)

Slikovit primer smrtonosnog blokiranja (deadlock) – automobili ne mogu dalje

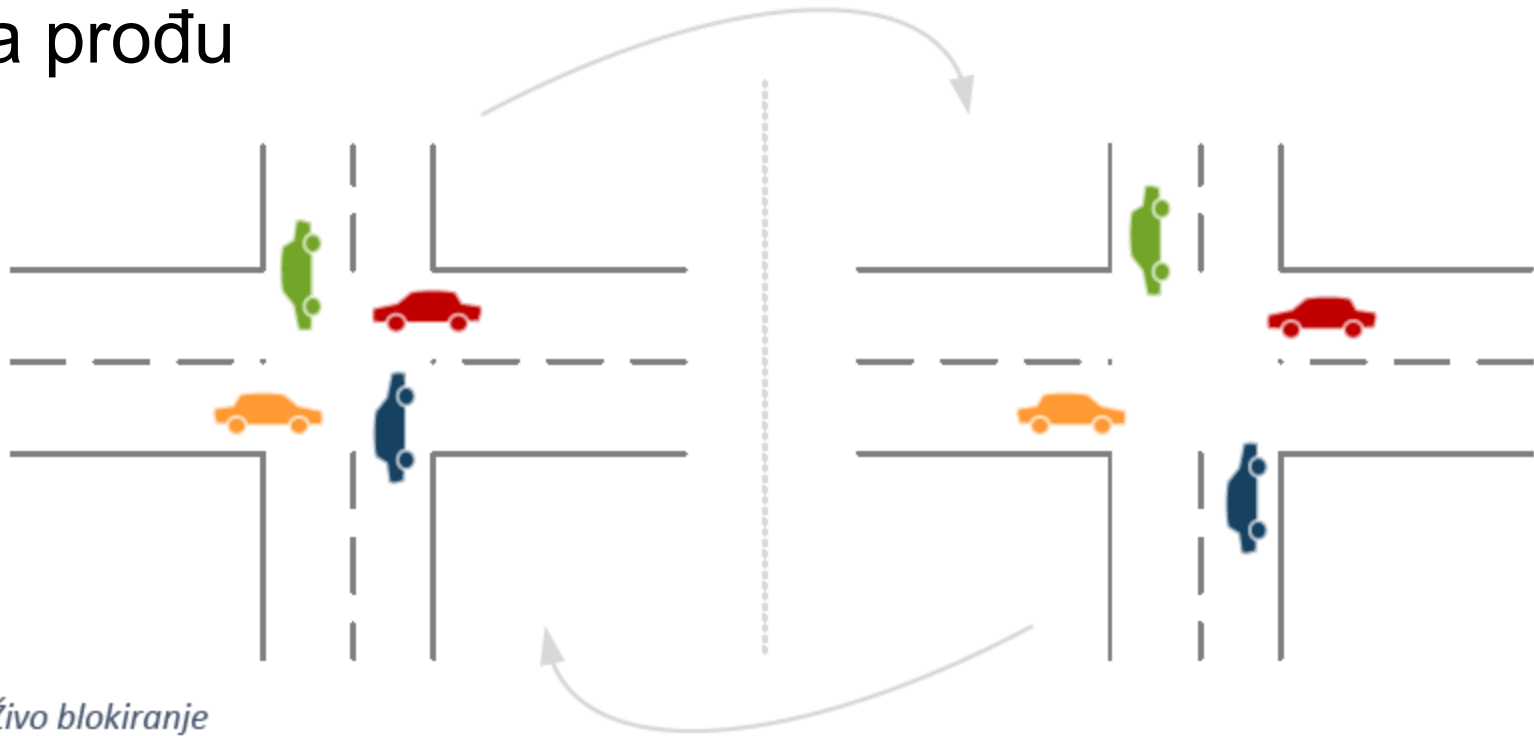


Zaglavljivanje (3)

- Situacije u kojima procesi nisu blokirani ali nemaju napretka u izvršavanju nazivaju se **živo blokiranje**.
- Najslikovitiji primer ovakve situacije u realnom životu je scena u kojoj u uskom prolazu treba da se mimoiđu dve osobe pri čemu se obe „džentlmenški“ istovremeno pomeraju u levu pa u desnu stranu pokušavajući da propuste jedna drugu. Pri tome, osobe ne napreduju ali ni ne stoje u mestu.

Zaglavljivanje (4)

Slikovit primer živog blokiranja (livelock) – automobili se vraćaju unazad pa istovremeno opet pokušavaju da prođu

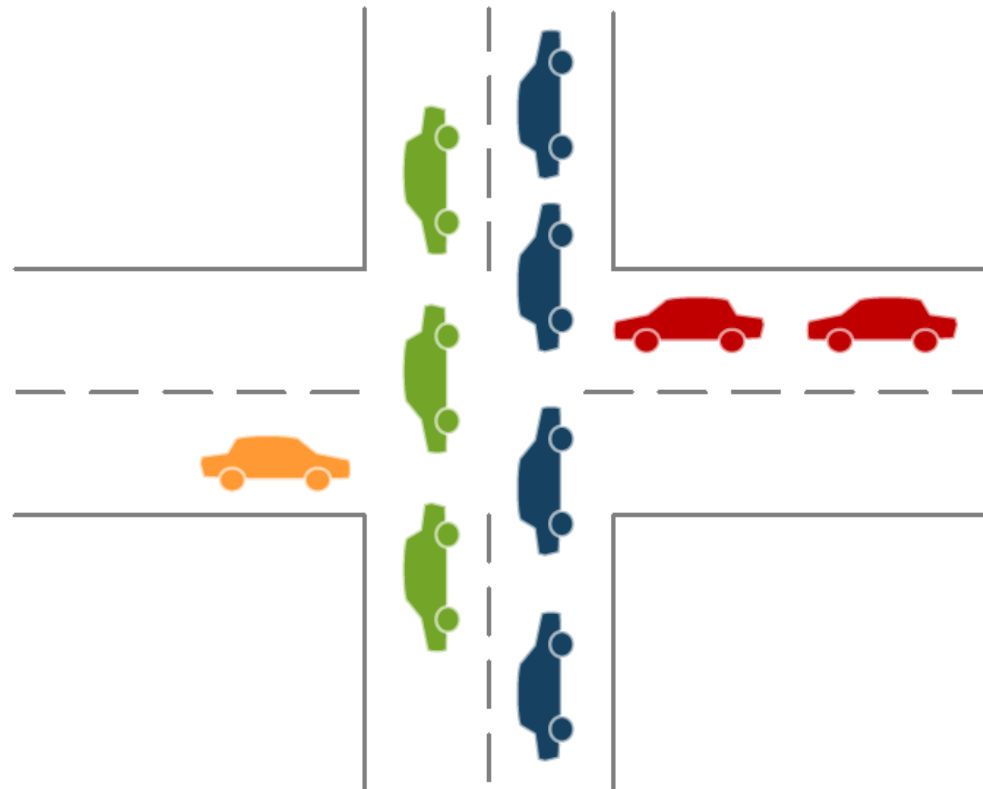


Zaglavljivanje (5)

- **Izgladnjivanje** se odnosi na situaciju koja podrazumeva da u sistemu generalno postoji napredak ali da neki procesi dosta dugo ne napreduju.
- Međutim, za razliku od zaglavljivanja i živog blokiranja, izgladnjivanje obično nije trajna situacija u sistemu.
- Često se događa da, posle određenog vremena (koje može biti proizvoljne dužine), procesi koji dugo čekaju dobiju priliku da nastave sa izvršavanjem.

Zaglavljivanje (6)

Slikovit primer izglednjivanja - ako su automobili koji se kreću u kolonama na putu sa prvenstvom prolaza ostali neće imati mogućnost da uđu u raskrsnicu.



Zaglavljivanje (7)

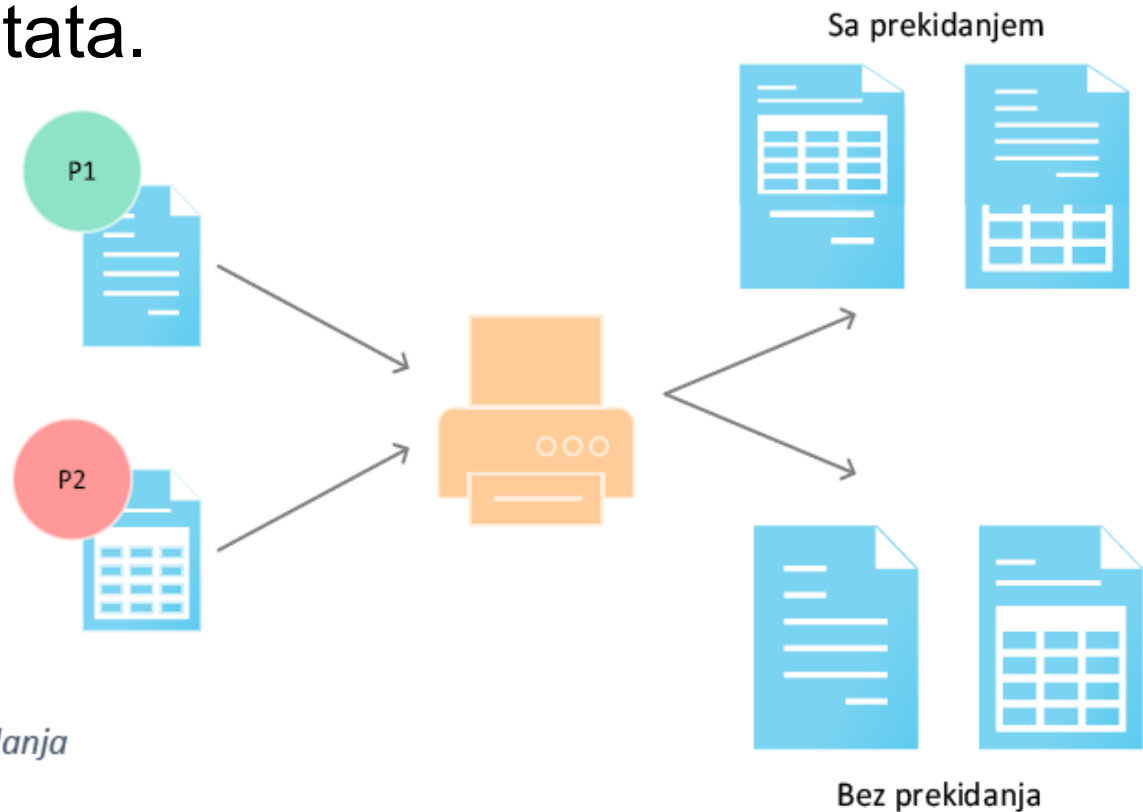
- U sistemima koji podržavaju multiprogramiranje i srodne koncepte lako se može dogoditi da dođe do zaglavljivanja.
- Ukoliko proces pri izvršavanju dospe u stanje Blokiran čekajući na resurse koje iz nekog razloga nikada neće dobiti smatra se da je proces zaglavljen.
- Pošto se procesi izvršavaju (pseudo) paralelno i konkurišu za resurse čiji je broj ograničen, do zaglavljivanja obično dolazi kada procesi imaju resurse koji su drugima potrebni, a drugi od njih očekuju resurse koji su njima već dodeljeni.

Zaglavljivanje (8)

- Resursi mogu biti hardverski poput diska, memorije, procesora, skenera, itd. i softverski poput semafora, datoteka, slogova u bazama podataka, itd.
- U obe grupe postoje oni koji se mogu i ne mogu „prekidati“, tj. koji se moraju iskoristiti do kraja kada se jednom do njih dođe.
- Do zaglavljivanja dolazi pri radu sa resursima koji kada se dobiju moraju biti iskorišćeni bez prekidanja, tj. oduzimanja tog resursa i predavanja drugom procesu pre nego što se potpuno iskoriste.

Zaglavljivanje (9)

Štampač je primer hardverskog resursa gde bi prekidanje jednog procesa pre završetka štampanja i puštanje drugog da nešto odštampa dovelo do neželjenih rezultata.



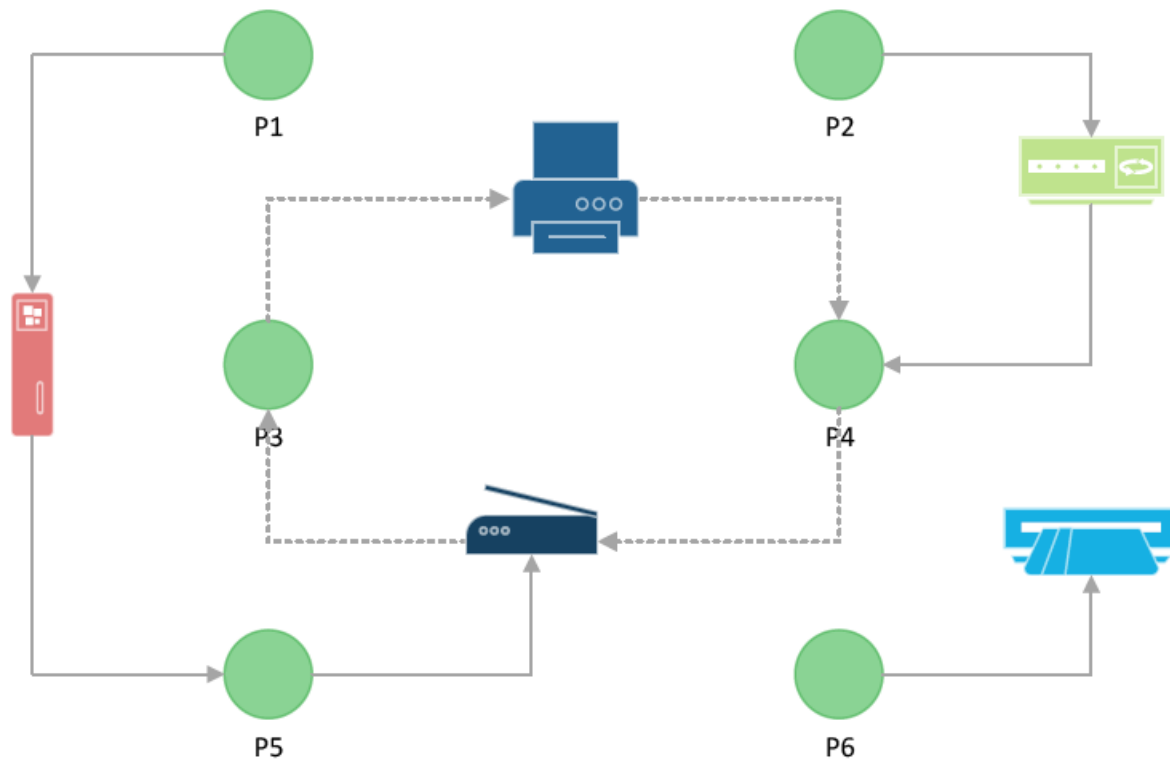
Rezultat štampanja sa i bez prekidanja

Zaglavljivanje (10)

- Da bi došlo do zaglavljivanja neophodno je da su u sistemu istovremeno ispunjena sledeća četiri uslova (tzv. Kohanovi uslovi):
 1. **Uzajamno isključivanje** (Mutual Exclusion) –u jednom trenutku tačno jedan proces može da koristi resurs, dok ostali moraju da sačekaju njegovo oslobađanje.
 2. **Čekanje i držanje** (Wait and Hold) – dok proces drži resurse sa kojima radi, može da zahteva nove koji su mu potrebni za dalje izvršavanje i da čeka one koji trenutno nisu raspoloživi.
 3. **Nemogućnost prekidanja** (No Preemption) –operativni sistem nema pravo da oduzme resurse koje je dodelio procesu i kasnije mu ih vrati, kako bi te resurse, u međuvremenu, dodelio drugim procesima.

Zaglavljivanje (11)

4. **Kružno čekanje** (Currier wait) –u sistemu postoji lanac procesa koji čekaju jedni na druge, tj. na resurse koji su im dodeljeni, čineći pri tome krug.



Kružno čekanje

Zaglavljivanje (12)

- Postojanje svih navedenih uslova u sistemu ne dovodi nužno i do zaglavljivanja već predstavlja neophodne uslove da do zaglavljivanja dođe.

Tretmani protiv zaglavljivanja

- Na osnovu mera koje se primenjuju tretmani protiv zaglavljivanja se mogu svrstati u sledeće grupe:
 1. **Sprečavanje** - pristup koji podrazumeva sistemske mere (implementiraju se u operativni sistem) i kojima se isključuje mogućnost zaglavljivanja.
 2. **Izbegavanje** - dinamičke mere kako bi se sistem vodio kroz stanja koja obezbeđuju da ne dođe do zaglavljivanja.
 3. **Detekcija i oporavljanje** - pristup koji dozvoljava da do zaglavljivanja dođe, ali ima mehanizme za otkrivanje da li je do zaglavljivanja došlo, tretman procesa koji su zaglavljeni i eventualno spašavanje dela rezultata.
 4. **Ne preduzimanje ničega** - mera koja prihvatljiva na sistemima na kojima do zaglavljivanja retko dolazi.

Mere za sprečavanje zaglavljivanja

- Mere sprečavanja su statičke systemske mere koje obično nisu algoritamski zahtevne, a za cilj imaju da u startu, definisanjem pravila za izvršavanje procesa, eliminišu neki od uslova koji dovode do zaglavljivanja.
- Najčešće su troškovi ovih algoritama mali ali su zato ograničenja velika.
- Jasno je da se prvi Kohmanov uslov ne može ukloniti tako da se mere mogu primeniti na preostala tri uslova.

Mere za sprečavanje zaglavljivanja (2)

- **Prevenција čekanja i držanja** - glavna ideja za eliminisanje čekanja i držanja je uvođenje ograničenja koje onemogućava da proces drži neke resurse dok traži nove.
- Obično se koriste sledeća dva pristupa za rešavanje ovakvih situacija:
 - 1) Pre nego što počne sa izvršavanjem, proces zahteva da mu se dodele svi potrebni resursi i da u izvršavanje krene tek kada ih sve dobije.
 - 2) Proces traži resurse u trenucima kada su mu potrebni, ali pre svakog uzimanja novog resursa one koje drži treba da vrati operativnom sistemu, kako bi operativni sistem doneo odluku da li će ih ponovo dodeliti njemu ili ih ustupiti nekom drugom procesu.

Mere za sprečavanje zaglavljivanja (3)

- 1) Loša strana prvog pristupa je to što za vreme svog izvršavanja proces **drži sve potrebne resurse** i time onemogućava druge procese da ih koriste u trenucima kada su njemu nepotrebni.

Na primer, ako proces treba da pročita fajl sa fleš memorije, snimi ga na disk, modifikuje i na kraju odštampa, ovakvim pristupom proces bi na početku zahtevao sve potrebne resurse (fleš memorija, disk, štampač) i držao ih do kraja, tj. završetka štampanja. Štampač bi u tom slučaju bio nedostupan ostalim procesima i u trenucima dok proces prenosi podatke ili dok se oni modifikuju.

Takođe, često na početku nije poznato koji će resursi biti potrebni procesu u toku izvršavanja što dodatno otežava upotrebu ovakvog algoritma.

Mere za sprečavanje zaglavljivanja (4)

- 2) Drugi pristup rešava problem prethodnog jer podrazumeva da proces traži resurse u trenucima kada su mu potrebni, ali da pre svakog uzimanja novog resursa ima obavezu da one koje drži vrati operativnom sistemu.

Na ovaj način postiže se da proces ne drži nijedan resurs u trenutku kada traži nove.

Kada je prethodni primer u pitanju, proces bi prvo zatražio fleš memoriju i disk kako bi obavio transfer.

Na kraju bi oslobodio te resurse kako bi zatražio štampač. Štampač bi tako sve do poslednje operacije procesa (štampanja) bio na raspolaganju drugim procesima.

Mere za sprečavanje zaglavljivanja (5)

- Primenom prethodno opisanih se mera eliminiše mogućnost da uslov čekanja i držanja bude ispunjen, pa do zaglavljivanja ne može doći.
- Neke od mana ovakvog pristupa su **slabije iskorišćenje** resursa, ali i **moguće „izgladnjivanje“** procesa koji ne mogu da dođu na red jer su u svakom trenutku neki od potrebnih resursa nedostupni.

Mere za sprečavanje zaglavljivanja (6)

- **Eliminisanje nemogućnosti prekidanja** – zasniva se na ideji da se u slučaju da se potrebe procesa ne mogu zadovoljiti otpuštaju svi resursi koje je taj proces držao.
- Naime, kada proces drži neke resurse a zahteva druge koji mu se ne mogu odmah dodeliti, on će otpustiti sve resurse koje drži i ustupiti ih drugim procesima na korišćenje.
- Pri tome, oduzeti resursi se dodaju na listu onih koje proces čeka.

Mere za sprečavanje zaglavljivanja (7)

- Proces će biti ponovo pokrenut u trenutku kada povрати svoje stare resurse, ali i dobije nove koje je zahtevao.
- Jasno je da ovakav pristup sigurno ne dovodi do zaglavljivanja, ali i da se vrlo lako može dogoditi da se zahtevniji procesi dugo čekaju da dođu u posed svih resursa koji su im potrebni.

Mere za sprečavanje zaglavljivanja (7)

- **Prevenција kružnog čekanja** - obično se koriste rešenja koja podrazumevaju označavanje tj. enumeraciju tipova (klasa) resursa.
- Ideja je da se svakom resursu dodeli jedan broj, a da proces koji drži određene resurse može da zahteva samo one resurse kojima je dodeljen veći broj od brojeva njegovih resursa.

Mere za sprečavanje zaglavljivanja (8)

Na primer, neka je skup $R = \{R_1, \dots, R_n\}$ skup svih resursa u sistemu. Svakom resursu se dodeljuje jedan ceo broj $F(R_i)=k$. Ako je R_i resurs sa najvećim brojem koji drži dati proces, u svom daljem izvršavanju taj proces može zahtevati samo resurse R_j takve da je $F(R_j) > F(R_i)$.

Pri tome, proces resurse nižeg ranga (manje vrednosti dodeljenog broja) može zahtevati u trenutku kada otpusti sve resurse višeg ranga.



Mere za sprečavanje zaglavljivanja (9)

- Kod ovakvog pristupa javljaju se teškoće oko numerisanja tj. dileme kako je najbolje napraviti enumeraciju.
- Takođe, jedan od problema je i potreba za ponovnom enumeracijom svaki put kada se u sistemu pojavi novi resurs.
- I pored navedenih problema primenom ovakvih mera se sigurno onemogućava kružno čekanje.

Mere za izbegavanje zaglavljivanja

- Za razliku od mera sprečavanja koje u startu eliminišu neki od neophodnih uslova da do zaglavljivanja dođe, **mere izbegavanja** su dinamičke mere koje se preduzimaju u određenim momentima u zavisnosti od stanja u sistemu.
- Iz tog razloga troškovi algoritma su obično veći nego kada su mere sprečavanja u pitanju, ali se sa druge strane povećava efikasnost sistema jer se resursi koriste optimalni.

Mere za izbegavanje zaglavljivanja (2)

- Jedan takav pristup, efikasan ali ne previše komplikovan, je **bankarev algoritam**.
 - Ime je dobio po principu davanja zajmova u bankama.
 - Korisnici traže od banke kredite kako bi mogli da realizuju svoje projekte.
 - Banka ima ograničene resurse, a korisnici znaju kolika je maksimalna količina novca koja im je potrebna.
 - Korisnici traže određenu količinu novca (kredite), a banke procenjuju da li će im i u kojem trenutku izaći u susret.
 - Novac banci mogu vratiti samo korisnici koji su uspeali da završe svoje projekte tako što su prethodno od banke dobili (kroz kredite) maksimalnu prijavljenu količinu novca.

Mere za izbegavanje zaglavljivanja (3)

- Bankarev algoritam podrazumeva da je pre početka izvršavanja svakog od procesa poznat maksimalni broj resursa svakog tipa koji će mu biti potreban.
- Ideja je da se dinamički ispituje pokušaj dodele resursa kako bi se obezbedilo da nikada ne dođe do kružnog čekanja.
- Stanje dodele resursa je definisano brojem raspoloživih, dodeljenih resursa i maksimalnim zahtevima procesa.

Mere za izbegavanje zaglavljivanja (4)

- Smatra se da je stanje sistema **bezbedno** ako sistem može u određenom redosledu da dodeljuje resurse kako bi svaki proces mogao da završi sa radom, tj. da ne dođe do zaglavljivanja, a da se pri tome svi procesi izvrše.
- Redosled kojim se vrši ovakva dodela resursa procesima naziva se **bezbedna sekvenca**.
- Ako stanje nije bezbedno to ne znači da će sigurno doći do zaglavljivanja, već da postoji mogućnost da do njega dođe.

Mere za izbegavanje zaglavljivanja (5)

- Bankarev algoritam je 1965. predložio Dijkstra.
- Osnovna ideja ovog algoritma je da se resursi procesima dodeljuju tako da se sistem sprovodi kroz niz bezbednih stanja kako bi se izbegla mogućnost da dođe do zaglavljivanja.
- Kao što je već napomenuto, banke vode računa da novac rasporede tako da zadovolje potrebe što većeg broja svojih klijenata.
- One će se truditi da kreditiraju perspektivne klijente i ne bi trebalo da dopuste da se novac dodeli jednom ili grupi klijenata, a da pri tome ne ostane sredstava za ostale.

Mere za izbegavanje zaglavljivanja (6)

- Analogno, kada je bankarev algoritam u pitanju, pri dodeli resursa vodi se računa o tome da sistem uvek bude u bezbednom stanju, odnosno po zahtevu procesa za dobijanje resursa vrši se provera da li postoji mehanizam kojim će se novododeljeni zajam kasnije (kada proces kojem je dodeljen završi svoj rad) vratiti i biti na raspolaganju drugim procesima.

Mere za detekciju zaglavljivanja

- **Detekcija zaglavljivanja** ne garantuje da do zaglavljivanja neće doći, već dopušta da do njega dođe, a onda se primenjuju mere da se takva situacija otkloni.
- Otkrivanje da je do zaglavljivanja došlo može biti **aktivno** i **pasivno**.
 - Aktivno otkrivanje podrazumeva da se, korišćenjem algoritama za detekciju, proverava da li je došlo do zaglavljivanja. Učestalost provere zavisi od frekvencije zaglavljivanja i može varirati na različitim sistemima.
 - Pasivan pristup podrazumeva da se provera vrši u situacijama kada se registruje da u sistemu nema aktivnosti ili da nešto nije u redu.

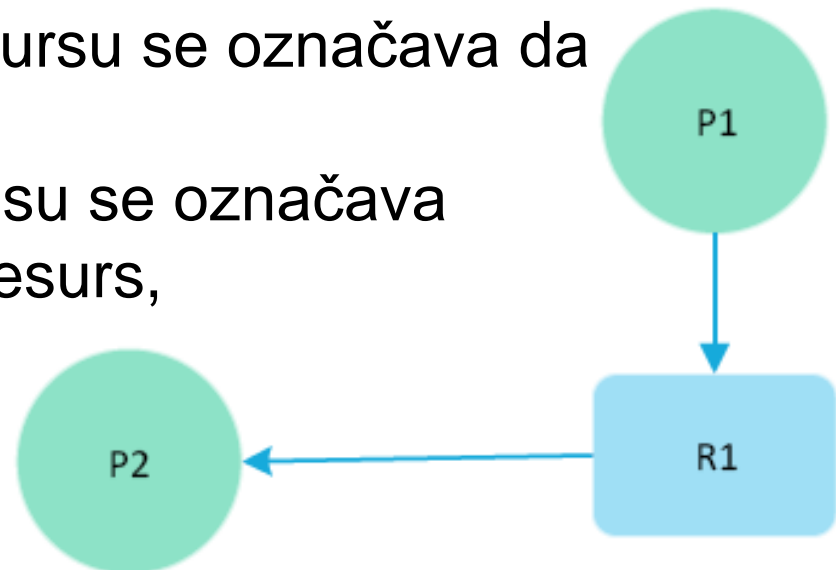
Mere za detekciju zaglavljivanja (2)

Stanje sistema se može modelovati korišćenjem **grafova**. Čvorovi ovakvih grafova su **procesi** (koji se obično predstavljaju krugovima) i **resursi** (predstavljani pravougaonikima).

Resursi sa više instanci se predstavljaju sa više tačaka u pravougaoniku.

Strelicom od procesa ka resursu se označava da neki proces drži resurs.

Strelica od resursa ka procesu se označava da neki proces zahteva taj resurs, ali da ga trenutno ne drži.



Primer stanja resursa i procesa u sistemu

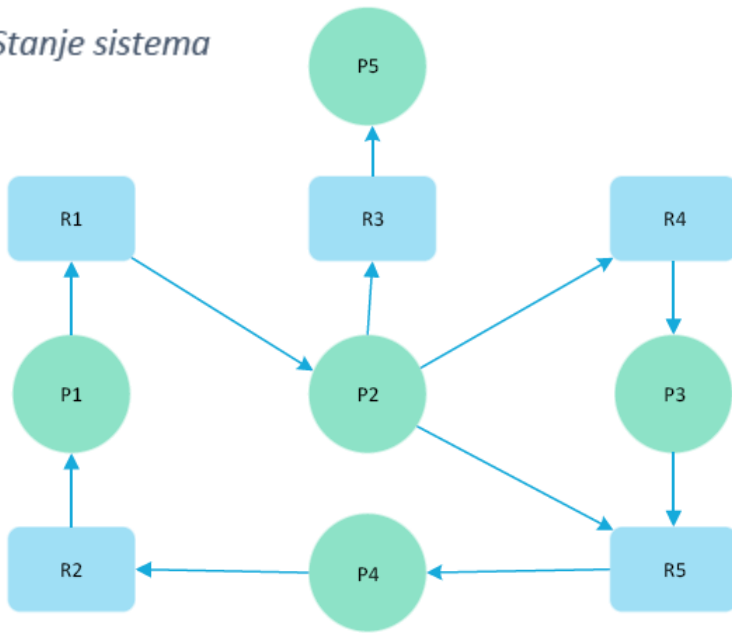
Mere za detekciju zaglavljivanja (3)

Detekcija zaglavljivanja kod sistema u kojem svi resursi imaju po jednu instancu se svodi na pronalaženje ciklusa u grafu.

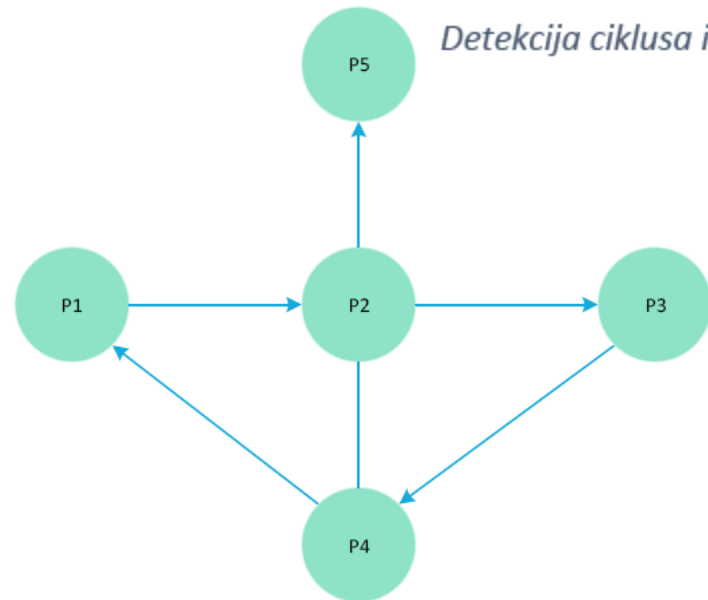
Graf koji se posmatra nastaje kada se iz grafa koji opisuje stanje sistema uklone resursi a grane povežu procese koji čekaju jedni na druge.

Ukoliko u grafu postoji ciklus to znači da postoji zaglavljivanje.

Stanje sistema

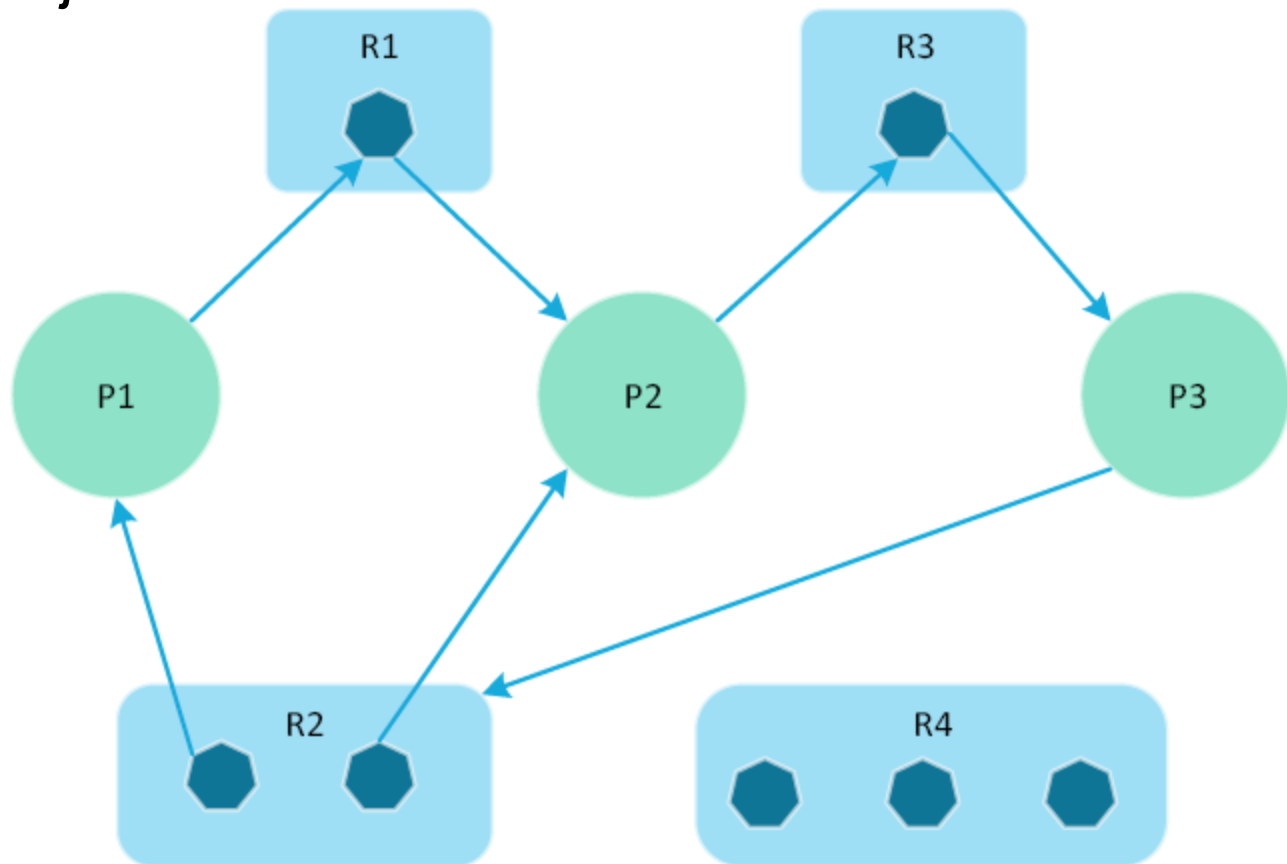


Detekcija ciklusa i zaglavljivanja



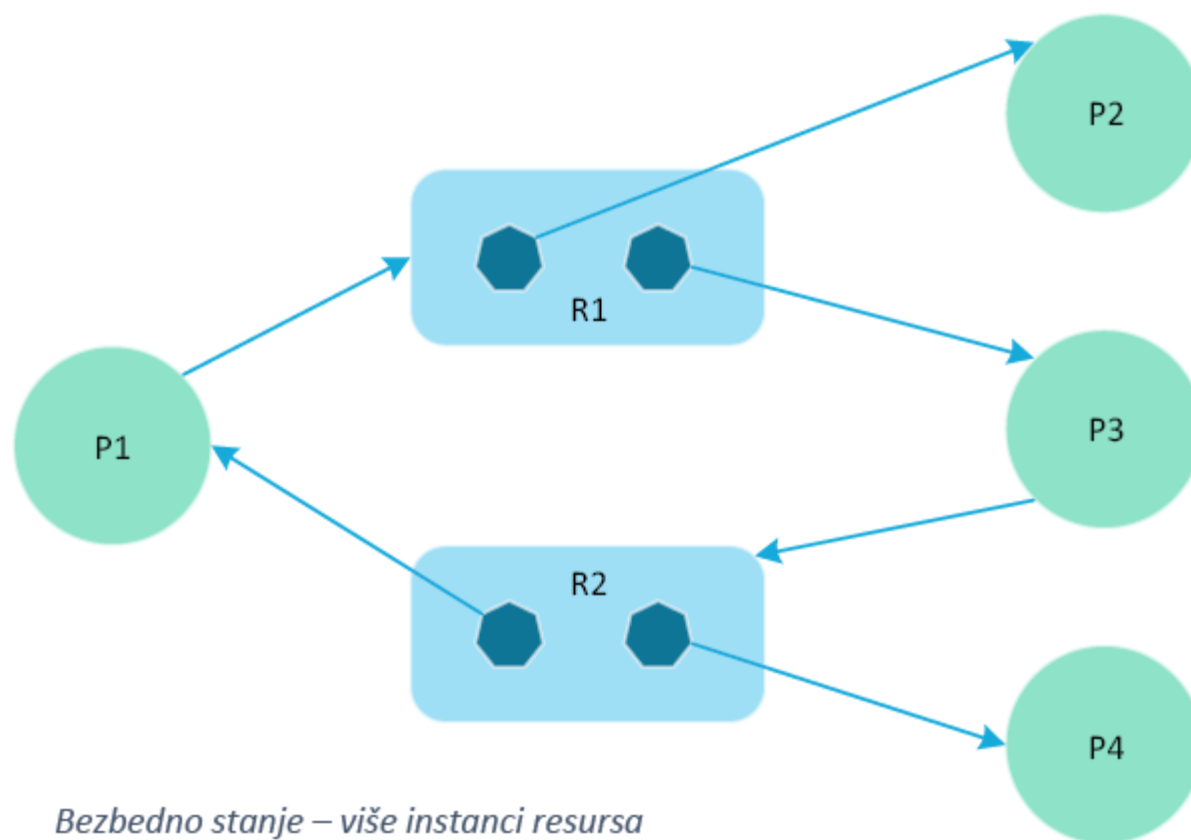
Mere za detekciju zaglavljivanja (4)

Kada su sistemi gde resursi mogu imati više instanci, postojanje ciklusa u grafu ne znači da je obavezno došlo do zaglavljivanja.



Zaglavljeno stanje – više instanci resursa

Mere za detekciju zaglavljivanja (5)



Mere za oporavak od zaglavljivanja

- Kada se utvrdi da je došlo do zaglavljivanja sistem bi trebalo da interveniše i pomogne da se ovakva situacija prevaziđe.
- Najjednostavnija akcija koju operativni sistem može da uradi je da suspenduje blokirane procese, oduzme im resurse a odluku o njihovoj sudbini prepusti korisnicima.
- U praksi se više koriste automatizovane operacije koje ne očekuju intervenciju korisnika.
- Najčešće se koriste dva pristupa: jedan je **prekidanje** jednog ili više zaglavljenih procesa, a drugi **oduzimanje resursa** zaglavljenim procesima.

Mere za oporavak od zaglavljivanja (2)

- Algoritam za **oporavak prekidanjem procesa**, koji se prvi nameće, podrazumeva da se uklone svi blokirani procesi.
- Ovo rešenje je veoma neefikasno jer se takvim postupkom gube rezultati do kojih su procesi došli dok se nisu zaglavili.
- Malo zahtevniji algoritam podrazumeva da se uklanja jedan po jedan od procesa u nadi da će se sistem izvesti iz zaglavljenog stanja. U tom slučaju, u svakom koraku, treba odabrati proces koji treba ukloniti, što obično nije lak zadatak.

Mere za oporavak od zaglavljivanja (3)

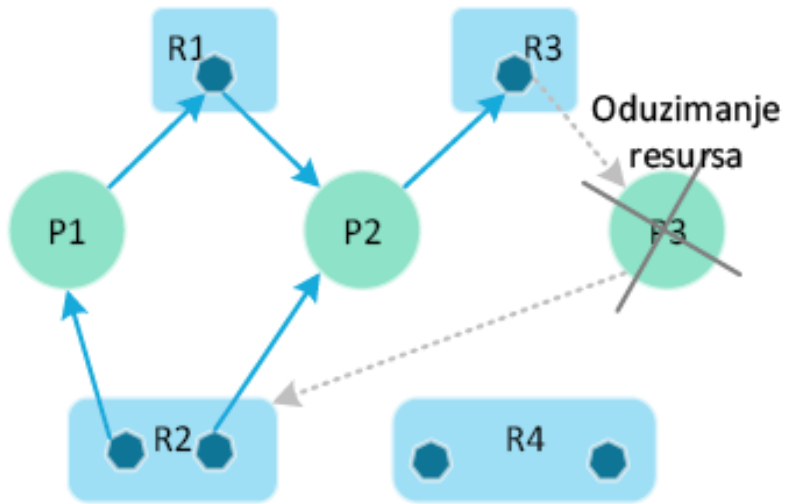
- Pri odabiru procesa koji će biti prekinut operativni sistem može uzimati neke od sledećih kriterijuma:
 - Prioritet procesa – treba bi prvo prekinuti procese nižeg prioriteta.
 - Koliko se dugo proces izvršava i koliko ima do završetka – ideja je da se prekidaju procesi koji su najkasnije počeli da se izvršavaju a da se prekidanje onih koji su pri kraju (ako je to moguće odrediti) odlaže do poslednjeg trenutka.
 - Koje resurse proces koristi a koji su mu potrebni da završi sa radom – teži se izbacivanju velikih potrošača ili procesa koji drže resurse koji su mnogim drugim procesima potrebni.
 - Koliko procesa treba da bude prekinuto.

Mere za oporavak od zaglavljivanja (3)

- Pristup koji podrazumeva **oduzimanje resursa** se zasniva na ideji da se određenim procesima oduzmu resursi kako bi se uz pomoć tih resursa otklonilo zaglavljivanje.
 - Pri tome bi trebalo da se vodi računa o tome da se takvi procesi ostave u stanju koje je pogodno da kasnije mogu nastaviti sa radom.
 - Trivijalno, proces se može vratiti na početak ali je korisnije pronaći tačku u izvršavanju od koje on može nastaviti kada se steknu uslovi i pri tome sačuvati rezultate izvršavanja pre oduzimanja resursa.
 - U ovakvim situacijama trebalo bi voditi računa i o tome da se istom procesu ne oduzimaju resursi više puta već da se poštuje neka vrsta pravednosti.

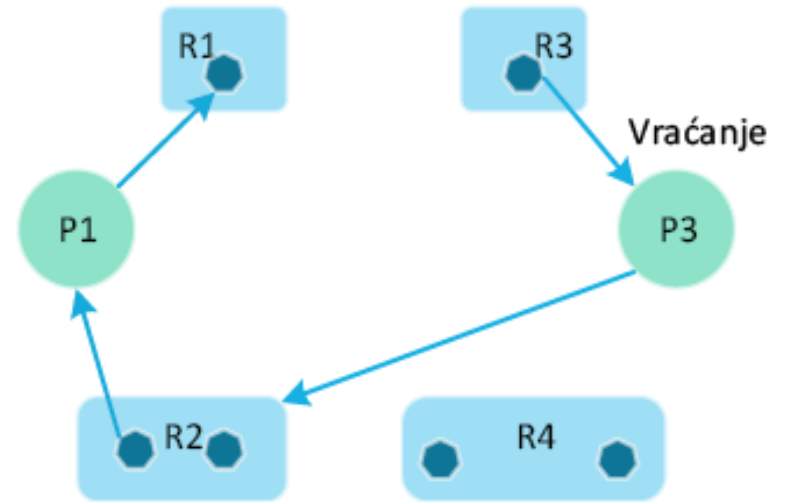
Mere za oporavak od zaglavljanja (4)

Prikazana je situacija u sistemu gde postoji zaglavljivanje. Problem se može rešiti suspendovanjem procesa P3 i njegovim kasnijim pokretanjem kada proces P2 završi sa radom.



Oduzimanje resursa i kasnije vraćanje

Nakon izvršavanja procesa P2



Zahvalnica

Najveći deo materijala iz ove prezentacije je preuzet iz knjige „Operativni sistemi“ autora dr Miroslava Marića i iz slajdova sa predavanja istog autora.

Hvala dr Mariću na datoj saglasnosti za korišćenje tih materijala.