

RMA I VREMENSKI POBUĐENI SISTEMI

◆ RMA analiza, II deo

Sprečavanje inverzije prioriteta

- ◆ U cilju sprečavanja pojave neograničene inverzije prioriteta, u sistemima koji rade u realnom vremenu, sa čvrstim vremenskim ograničenjima, se za potrebe međusobne sinhronizacije zadataka koriste binarni semafori sa plafonom prioriteta.

Pojam inverzije prioriteta (1/2)

- ◆ Posmatraju se tri procesa, označena prema prioritetima kao V, S, N (visok, srednji i nizak).
- ◆ Procesi V i N dele resurs, koji je štićen binarnim semaforom B.
- ◆ Neka proces N uspešno zauzme resurs sa P(B), zatim ga prekida V, koji takođe želi da zauzme isti resurs P(B), ali ne uspeva, pošto je resurs već zauzet.
- ◆ Tako dolazi do inverzije prioriteta: proces V mora sačekati da proces N oslobodi resurs.

Pojam inverzije prioriteta (2/2)

- ◆ Problem nije jako izražen ako su u sistemu prisutni samo V i N , pošto u tom slučaju čim N izađe iz svoje kritične sekcije, tj. obavi $V(B)$, on će prepustiti procesor procesu V .
- ◆ Tada je vreme trajanja inverzije prioriteta kratko, i jednako je vremenu potrebnom da N izvrši svoju kritičnu sekciju.

Pojam neograničene inverzije prioriteta (1/2)

- ◆ Problem je izražen ukoliko se nakon blokiranja procesa V , pojavi proces S .
- ◆ U tom slučaju dolazi do inverzije prioriteta u trajanju potrebnom da se završi S i da zatim N izađe iz kritične sekcije.
- ◆ Problem postaje opasan ako postoji puno procesa sa prioritetima između V i N , jer će se tada trajanje inverzije prioriteta veoma produžiti.

Pojam neograničene inverzije prioriteta (2/2)

- ◆ Ovaj problem se u teoriji OS naziva problem neograničene inverzije prioriteta (engl. unbounded priority inversion problem).
- ◆ Praktično, inverzija prioriteta može veoma lako dovesti do pojave međusobnog blokiranja (deadlock).

Osnovni protokol nasleđivanja prioriteta

- ◆ Delimično rešenje ovih problema je osnovni protokol nasleđivanja prioriteta (engl. basic priority inheritance protocol, PIP).
- ◆ On rešava problem neograničene inverzije prioriteta, ali ne rešava problem međusobnog blokiranja.

Protokol sa plafonom prioriteta

- ◆ Potpuno rešenje oba problema je tzv. protokol sa plafonom prioriteta (engl. priority ceiling protocol, PCP).
- ◆ Dve važne osobine:
 - onemogućava međusobno blokiranje procesa
 - onemogućava neograničenu invreziiju prioriteta, naime, najviše jedan proces nižeg prioriteta može blokirati proces višeg prioriteta za vreme dok proces nižeg prioriteta ne napusti svoju kritičnu sekciju.

Dve centralne ideje PCP-a

- ◆ I) nasleđivanja prioriteta: kad proces p blokira izvršenje prioritetnijeg procesa, on prelazi na nivo prioriteta procesa koji blokira.
- ◆ II) ulaz u kritičnu sekciju je dozvoljen samo ako će se ta kritična sekcija izvršavati na nivou prioriteta, koji je veći od (nasleđenih) nivoa prioriteta svih istisnutih kritičnih sekcija.

Binarni semafor sa plafonom prioriteta (1/2)

- ◆ Plafon prioriteta binarnog semafora B se definiše kao najviši prioritet od svih procesa koji mogu zaključati B.
- ◆ Kad proces p pokuša da izvrši neku od svojih kritičnih sekcija, on se blokira, osim ako je njegov prioritet veći od plafona prioriteta svih semafora zaključanih od strane drugih procesa.

Binarni semafor sa plafonom prioriteta (2/2)

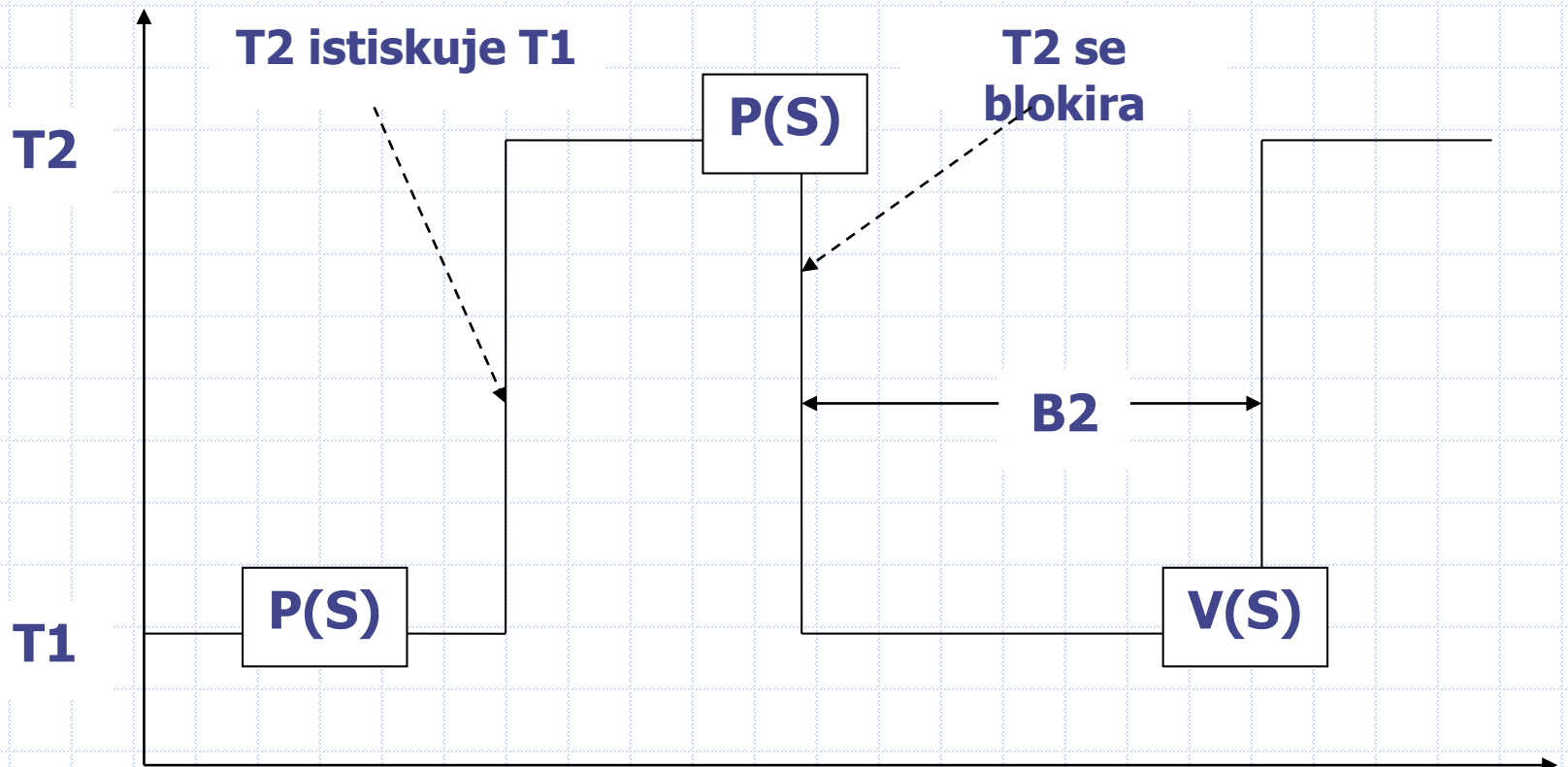
- ◆ Ako se proces p blokira na ulasku u kritičnu sekciju, za proces koji je zaključao semafor, sa najvišim plafonom prioriteta, se kaže da blokira proces p , i iz tog razloga nasleđuje prioritet procesa p .
- ◆ Sve dotle dok proces p ne pokušava da uđe u neku od svojih kritičnih sekcija, on istiskuje procese nižeg prioriteta.

Prošireni RM test rasporedivosti

- ◆ engl. extended rate monotonic schedulability test.
- ◆ Ovaj test računava moguća blokiranja zadataka na semaforima koji kontrolišu deljene resurse.
- ◆ Za potrebe ovog testa se definiše B_i kao najduže ukupno vreme blokiranja zadatka τ_i unutar njegove periode.

Pojam "istiskivanja" i "blokiranja"

(B2 je vreme blokiranja zadatka T2)



Proširen test rasporedivosti (engl. Extended RMS Test)

$$C_1/T_1 + B_1/T_1 \leq 1(2^{1/1}-1)$$

$$C_1/T_1 + C_2/T_2 + B_2/T_2 \leq 2(2^{1/2}-1)$$

...

$$C_1/T_1 + C_2/T_2 + \dots + C_k/T_k + B_k/T_k \leq k(2^{1/k}-1)$$

...

$$C_1/T_1 + C_2/T_2 + \dots + C_n/T_n \leq n(2^{1/n}-1)$$

Proširen RM test rasporedivosti skupa od n zadataka.

Problem promene režima

- ◆ Kod nekih sistema sa radom u realnom vremenu, skup zadataka u sistemu, kao i osobine pojedinih zadataka, se menjaju u toku rada sistema.
- ◆ Promena režima izvršenja se može zamisliti kao brisanje nekih zadataka i dodavanje drugi zadatka, ili kao promena parametara određenih zadataka.

Zahtevi za promenu režima

- ◆ 1) Kompatibilnost: moraju se sačuvati dve važne osobine protokola plafona prioriteta: (1) nemoguće je međusobno blokiranje zadataka i (2) zadatak nižeg nivoa prioriteta može blokirati najviše jednom zadatak višeg nivoa prioriteta.
- ◆ 2) Podrška održavanju: mora dozvoliti dodavanje novih zadataka
- ◆ 3) Performansa: bar kao kod monitora

Rešenje: Protokol za promenu režima (engl. Mode Change Protocol, MCP):

- ◆ To je drugi osnovni protokol RMS teorije (prvi je Priority Ceiling Protocol, PCP).
- ◆ On se sastoji iz tri dela koji odgovaraju originalnim zahtevima:
 - aspekt kompatibilnosti
 - aspekt održavanja
 - aspekt performanse

MCP, aspekt kompatibilnosti (1/2)

- ◆ Za svaki **nezaključan** semafor S čiji plafon prioriteta mora da se **podigne**, plafon se odmah podiže (na neprekidiv način).
- ◆ Za svaki **zaključan** semafor S čiji plafon prioriteta mora da se **podigne**, plafon se podiže (neprekidivom operacijom) odmah nakon otključavanja semafora S.

MCP, aspekt kompatibilnosti (2/2)

- ◆ Za svaki semafor S čiji plafon prioriteta treba **sniziti**, plafon se snižava **nakon brisanja svih zadataka** koji mogu zaključati semafor S , i koji imaju prioritet veći od novog plafona prioriteta semafora S .
- ◆ Ako je prioritet zadatka t veći od plafona prioriteta zaključanih semafora S_1, \dots, S_k , koje on može da zaključava, plafoni prioriteta semafora S_1, \dots, S_k **moraju biti podignuti pre dodavanja zadatka t .**

MCP, aspekt održavanja (1/2)

- brisanje zadatka

- ◆ Zadatak čije izvršenje nije započelo u tekućoj periodi može se obrisati odmah. Njegov budžet se može odmah dodeliti drugim zadacima.
- ◆ Zadatak čije izvršenje u tekućoj periodi jeste započelo može se obrisati tek u narednoj periodi. Tek nakon brisanja njegov budžet se dodeljuje drugima.

MCP, aspekt održavanja (1/2)

- dodavanje zadatka

- ◆ Zadatak se može dodati u sistem samo ako postoji dovoljan slobodnog procesorskog kapaciteta.

MCP, aspekt performanse (1/2)

- ◆ Isto tako je dokazano da je trajanje promene radnog režima ograničeno većim od sledeća dva broja:
 - (1) perioda zadatka sa najvećom periodom od svih zadataka koje treba obrisati, i
 - (2) najkraća perioda pridružena semaforu koji ima najniži plafon prioriteta, koji treba da bude menjan.
- ◆ Ovaj broj je obično puno manji, a nikad nije veći, od najmanjeg zajedničkog umnoška (engl. LCM - Least Common Multiple) svih perioda.

MCP, aspekt performanse (2/2)

- ◆ Nasuprot tome, u pristupu koji se zasniva na cikličkom egzikjutivu, glavni ciklus je LCM svih perioda, a promena radnog režima je moguća tek po isteku glavnog ciklusa.
- ◆ Pored toga, protokol promene radnog režima omogućava dodavanje i izvršavanje većine urgentnih zadataka novog režima u toku promene režima, tj. pre završetka promene radnog režima.

Problematične pretpostavke osnovne RMS teorije:

- ◆ smena zadatka je trenutna
- ◆ vreme rada OS nije uračunato
- ◆ nije dozvoljena međusobna interakcija zadatka
- ◆ zadaci postaju pripravnici za izvršenje tačno na početku svoje periode
- ◆ krajnji rok je jednak periodu izvršenja
- ◆ zadaci sa kraćom periodom dobijaju veći prioritet a da se ne razmatra kritičnost zadatka
- ◆ izvršenje zadatka je uvek konzistentno sa RM prioritetima

Problematični aspekti U/I radnji

- ◆ Prekidi. Najčešće signaliziraju kraj U/I. Imaju visok prioritet koji nije u skladu sa RM dodelom prioriteta.
- ◆ DMA. Efektivno obavlja U/I radnju. Zadatak koji je inicirao U/I se blokira.
- ◆ Neprekidivost zadatka u rutinama OS. Predstavlja oblik inverzije prioriteta.

Da li je RM analiza primenljiva i u tim slučajevima?

- ◆ Pokazuje se da jeste.
- ◆ U tom cilju razmatraju se:
 - stvarno vreme smene zadataka
 - suspenzija zadatka za vreme U/I radnje
 - zadaci koji se izvršavaju na prioritetima koji nisu dodeljeni po RM principu

Modeliranje smene zadatka

- ◆ Smena zadatka se može modelirati dodavanjem odgovarajućeg člana.
- ◆ Konkretno, neka je C_i vreme izvršenja zadatka i i neka je C_s vreme smene zadatka u najgorem slučaju, onda je vreme izvršenja zadatka, koje uračunava smenu zadatka, dato kao $C' = C_i + 2C_s$.
- ◆ Prema tome, vreme smene zadatka se može lako uključiti u osnovni RM test rasporedivosti.

Modeliranje U/I radnje

Proces
započinje U/I
radnju

Gomilanje vremena
izvršavanja!

- ◆ Negativan efekat suspenzije zadatka višeg prioriteta na zadatke nižeg prioriteta je tzv. efekat odloženog izvršenja (džiter).
- ◆ Intuitivno se vidi da suspenzija zadatka višeg prioriteta može izazvati "gomilanje vremena izvršavanja".
- ◆ No ipak, pojava "gomilanja" se može uzeti u obzir dodavanjem još po jednog člana u nejednačinama koje su pridružene zadacima nižeg prioriteta!

Modeliranje inverzije prioriteta

- ◆ PCP je rešenje za sinhronizaciju zadataka.
- ◆ Postoje i drugi izvori inverzije prioriteta, kao što su intervali nesmenjivosti zadatka i prekidi.
- ◆ Sve ove slučajeve inverzije prioriteta je moguće uračunati uključivanjem dodatnih članova u osnovnu RM nejednačinu!

Model rasporedivosti

- ◆ To je u osnovi skup RM nejednačina, po jedna za svaki zadatak.
- ◆ Svaka nejednačina se sastoji od članova koji opisuju ili modeliraju različite faktore koji utiču na rasporedivost zadatka.
- ◆ Npr. član koji uračunava izvršenje, istiskivanje, blokiranje (zbog deljenja resursa ili inverzije prioriteta) i negativan uticaj džitera.

Konstruisanje modela rasporedivosti

- ◆ Model rasporedivosti je moguće formirati tzv. inkrementalnim putem. Ovaj pristup se zasniva na pokušaju da se nađu odgovori na dva osnovna pitanja, za svaki zadatak t_i :
 - Kako drugi zadaci utiču na rasporedivost t_i ?
 - Kako zadatak t_i utiče na rasporedivost drugih zadataka?