

# Višedretvena sinkronizacija

svi zadaci ikad!!

# 2010/11 – 2KZ

6. U nekom sustavu zbivaju se događaji koji traže obradu. Dijelovi obrada traže sredstva zaštićena binarnim semaforima. Obrada događaja obavlja se u zasebnim dretvama s odgovarajućim prioritetima (pojava događaja višeg prioriteta istiskuje obradu manjeg prioriteta, tj. odgovarajuće dretve). Neka su događaji (za promatrani vremenski interval) zadani vremenima pojave, trajanjima obrade i zahtjevima za sredstvima i njihovim oslobađanjem (na početku se zauzimaju i/ili oslobađaju sredstva):

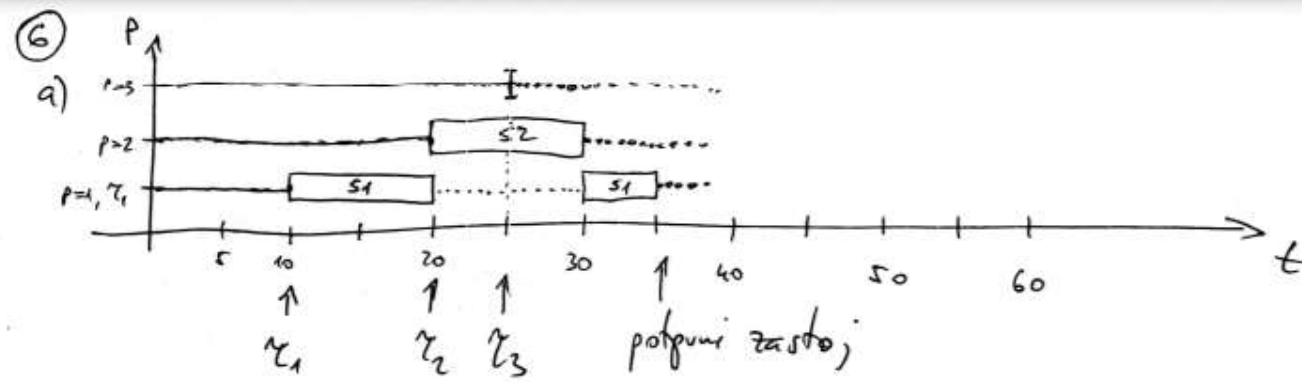
$\tau_1$ :  $t_1=10$ ; {[zauzmi( $s_1$ ); radi(15)]; [zauzmi( $s_2$ ); radi(5)]; [oslobodi( $s_1, s_2$ ); radi(5)];}

$\tau_2$ :  $t_2=20$ ; {[zauzmi( $s_2$ ); radi(10)]; [zauzmi( $s_1$ ); radi(5)]; [oslobodi( $s_1, s_2$ ); radi(5)];}

$\tau_3$ :  $t_3=25$ ; {[zauzmi( $s_2$ ); radi(5)]; [zauzmi( $s_1$ ); radi(5)]; [oslobodi( $s_1, s_2$ ); radi(5)];}

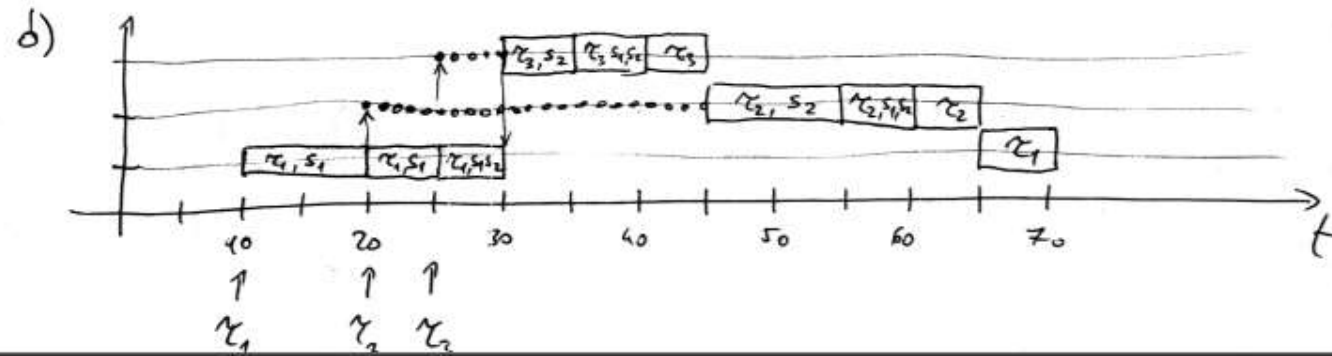
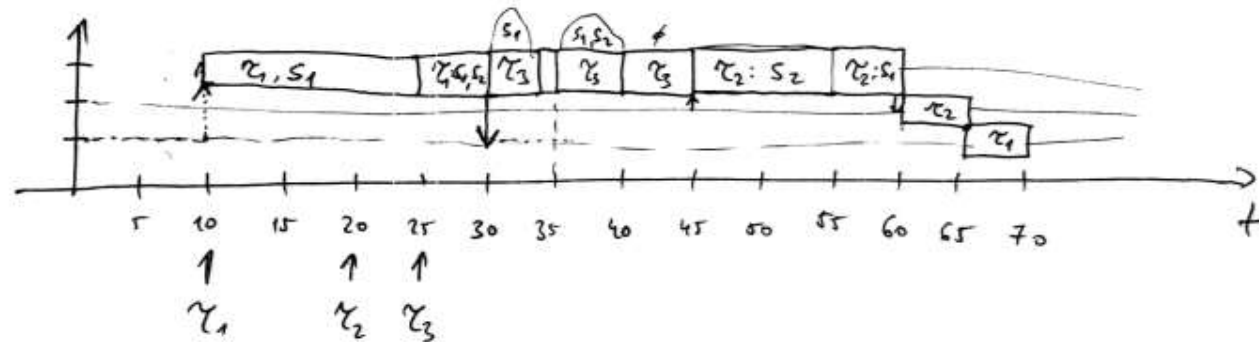
Prikazati što će se dogoditi u sustavu ako se koristi:

- a) (2) samo raspoređivanje prema prioritetu
- b) (2) protokol nasljeđivanja prioriteta
- c) (2) protokol stropnog prioriteta – jednostavnija inačica (navesti stropne prioritete sredstava)
- d) (2) protokol stropnog prioriteta – napredna inačica (navesti stropne prioritete sredstava)



b) isto kao i a), s time da: u  $t=25$   $\tau_2$  uadijubi prihvati od  $\tau_3$   
 u  $t=30$   $\tau_1$  —||—  $\tau_2$  (t. od  $\tau_3$ )

c)  $SP(S1) = 3$   
 $SP(S2) = 3$



# 2020 – ZI

5. [2 boda] Problem pet filozofa riješen je prikazanim kodom sa semaforima uz korištenje izvorna protokola stropnog prioriteta ("složenijeg"). Sve dretve imaju jednaki prioritet=10.

```
filozof (i)
    ponavlja
        razmišljaj
        ČekajSEM (i)
        ČekajSEM ((i+1) mod 5)
        jedi (traje 10 jedinica vremena)
        PostaviSEM ((i+1) mod 5)
        PostaviSEM (i)
```

Pokazati rad sustava za sljedeće scenarije:

- a) u  $t=0$  filozof 1 prvi hoće jesti (završava s razmišljanjem);  
u  $t=1$  filozof 3 hoće jesti (ostali filozofi još dugo razmišljaju)
- b) u  $t=0$  svi filozofi hoće jesti, ali je filozof 2 prvi pozvao ČekajSem (drugi je filozof 4).

Završiti s prikazom nakon što prvi filozof koji jede završi i postavi oba semafora (i pritom možda propusti nekog drugog) ili ako se dogodi potpuni zastoј.





0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6

f1 f3  
s1 s3  
s2 s4

- a) u  $t = 0$  filozof 1 može zauzeti oba semafora:
- za ČekajSEM(1) nema  $S^*$  i  $S1$  je slobodan, pa nema ograničenja
  - za ČekajSEM(2)  $S^*$  je  $S1$ , ali ovaj filozof ga je zauzeo
- u  $t = 1$  filozof 3 se blokira
- ČekajSEM(3):  $S^*$  je  $S1$ , filozof 3 nema veći prioritet!
- u  $t = 10$  filozof 1 oslobađa semafore i onda tek može filozof 3
- b) slično kao za a), samo što prvo kreće filozof 2, pa onda 4.



0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6

svi  
f2  
s2  
s3

f1 uzima i s2 -> spreman

f3 uzima s4 -> spreman

f3 uzima s3 -> čeka s4

f5 uzima s5 -> čeka s1

f4  
s4  
s5

f1  
s1, čeka s2

f3 čeka s3, s4

f5 čeka s5, s1

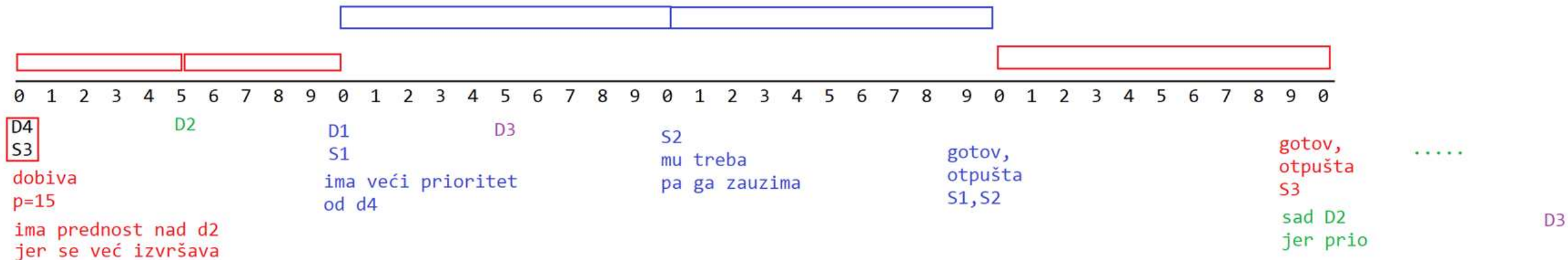
# 2017 – ZI

10. [2 boda] U nekom sustavu nalaze se dretve  $D_1 - D_4$  s prioritetima  $p_1 = 20$ ,  $p_2 = 15$ ,  $p_3 = 10$ ,  $p_4 = 5$ . Poznato je da će dretva  $D_1$  u svom radu trebati semafore  $S_1$  i  $S_2$ , dretva  $D_2$  semafore  $S_2$  i  $S_3$ , dretva  $D_3$  semafore  $S_3$  i  $S_4$  te dretva  $D_4$  semafore  $S_1$  i  $S_3$ . Početne vrijednosti svih semafora su 1. Ukoliko se koristi izvorni protokol stropnog prioriteta opisati zbivanja u sustavu za slijedeći scenarij:

- a) u  $t = 0$  se javlja  $D_4$  i treba  $S_3$
- b) u  $t = 5$  se javlja  $D_2$  i treba  $S_2$
- c) u  $t = 10$  se javlja  $D_1$  i treba  $S_1$
- d) u  $t = 15$  se javlja  $D_3$  i treba  $S_3$

Pretpostaviti da svaka dretva nakon što zauzme semafor isti i vrati nakon 20 ms korištenja, osim dretve  $D_1$  koja nakon 10 ms treba i semafor  $S_2$ , te nakon još 10 ms rada s oba semafora, oba i otpušta.

na sljedećem slajdu je službeno rješenje, ovo  
je naše



10.	$\Delta$	$p$	$S$	$S$	$SP$
	$\Delta_1$	20	$S_1, S_2$	$S_1$	20
	$\Delta_2$	15	$S_2, S_3$	$S_2$	20
	$\Delta_3$	10	$S_3, S_4$	$S_3$	15
	$\Delta_4$	5	$S_1, S_3$	$S_4$	10

$t=0$   $\Delta_4$  zauzima  $S_3$

$t=5$   $\Delta_2$  se blokira ( $P_2 > SP(S^* = S_2) = 15$  nije zad.)  $\Rightarrow P(\Delta_4) = P(\Delta_2) = 15$

$t=10$   $\Delta_1$  zauzima  $S_1$  ( $\Delta_4$  je do tada obradila 10 ws)

$t=15$   $\Delta_3$  se blokira ( $P(\Delta_4) = 15 > P(\Delta_3)$  pa  $\Delta_4$  u potpunosti preuzima od  $\Delta_3$ )

(pretpostavimo jeduprocesorski sustav)

$t=20$   $\Delta_1$  zauzima  $S_2$  ( $S^* = S_1$  koji drži brata  $\Delta_1$ !)

$t=30$   $\Delta_1$  otpušta  $S_1$  i  $S_2$  (<sup>izvršava</sup>),  $\Delta_4$  nastavlja s radom

$t=40$   $\Delta_4$  otpušta  $S_3 \Rightarrow$  radnja se prvo na 5,  $\Delta_2$  nastavlja s radom - zauzima  $S_2$

$t=60$   $\Delta_2$  otpušta  $S_2$ , završava,  $\Delta_3$  zauzima  $S_3$

$t=80$   $\Delta_3$  otpušta  $S_3$ , završava



2014 – ZI, nakon obrađenog B se semafor otpušta

8. [2 boda] U nekom sustavu javljaju se dretve:  $D_1$  u  $t_1 = 0$  ms,  $D_2$  u  $t_2 = 3$  ms te  $D_3$  u  $t_3 = 6$  ms. Svaka dretva sastoji se od tri dijela posla: A, B i C. Izvođenje A dijela traje 2 ms, B dijela 3 ms te C dijela 2 ms. Prije izvođenja B dijela dretve trebaju zauzeti semafore:  $D_1$  semafor  $S_1$ ,  $D_2$  semafor  $S_2$ ,  $D_3$  semafor  $S_1$ . Dretva  $D_3$  ima najveći prioritet, dok  $D_1$  ima najmanji. Ukoliko se koristi protokol nasljeđivanja prioriteta pokazati izvođenje zadanih dretvi, tj. što procesor radi u pojedinom trenutku, dok sve dretve ne završe sa svojim poslovima.

Rj.

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
D	1 <sub>A</sub>	1 <sub>A</sub>	1 <sub>B</sub>	2 <sub>A</sub>	2 <sub>A</sub>	2 <sub>B</sub>	3 <sub>A</sub>	3 <sub>A</sub>	1 <sub>B</sub>	1 <sub>B</sub>	3 <sub>B</sub>	3 <sub>B</sub>	3 <sub>B</sub>	3 <sub>C</sub>	3 <sub>C</sub>	2 <sub>B</sub>	2 <sub>B</sub>	2 <sub>C</sub>	2 <sub>C</sub>	1 <sub>C</sub>	1 <sub>C</sub>	1 <sub>C</sub>	–

mi smo tu dobili malo drugačije:

T	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
	A1	A1	B1	A2	A2	B2	A3	A3	B1	B1	B3	B3	B3	C3	C3	B2	B2	C2	C2	C1	C1	

# 2013 – MI, nema rj

13. (3) U nekom jednoprocesorskom sustavu u istom trenutku pokrenute su dretve:  $D_1$ ,  $D_2$ ,  $D_3$ ,  $D_4$  i  $D_5$ . Neka se posao dretvi može opisati sa (indeks  $i$  opisuje dretvu  $D_i$ ):

```
dretva D(i) {  
    PostaviSem(S[i+1]);  
    ČekajSem(S[i]);  
}
```

Početne vrijednosti semafora su nule. Indeks dretvi predstavlja prioritet (veći broj označava veći prioritet). Prikazati rad prioritetnog raspoređivača dok sve dretve ili ne završe ili se blokiraju u redu semafora.



## 2009 – 2MI, nema rj

4. U sustavu  $S_2$  se nalazi pet periodičkih zadataka:  $\tau_1$  do  $\tau_5$ . Raspoređivanje zadataka se obavlja po prioritetu (s dozvoljenim prekidanjem zadataka), a prioriteti su pridijeljeni statički:  $\tau_1$  ima najveći, a  $\tau_5$  najmanji prioritet. Vremena izvođenja zadataka su ista za sve zadatke i iznose:  $c_i = 5$ . Zadaci  $\tau_1$  i  $\tau_5$  u svom izvođenju trebaju sredstvo X (koriste ga u kritičnom odsječku zaštićenom istim semaforom), ali samo za vrijeme  $c_x = 1$  (od ukupnog  $c_i = 5$  vremena računanja u jednom pojavljivanju). Svi se zadaci periodički javljaju u sustav s periodom  $T = 100$ , ali ne nužno sinkronizirano, ne nužno svi u isto vrijeme.
- a) (2) Odrediti najveće moguće odgađanje završetka izvođenja zadatka  $\tau_1$  zbog problema inverzije prioriteta (u najgorem slučaju).
  - b) (1) Ako bi se koristio protokol stropnog prioriteta ili protokol nasljeđivanja prioriteta, koliko bi tada iznosilo najveće odgađanje završetka (zbog problema inverzije prioriteta)?

# 2017 – ZI, nema rj

7. [2 boda] U nekom jednoprosesorskom sustavu izvode se dretve  $A$ ,  $B$ ,  $C$  i  $D$ . Dretva  $A$  ima najveći prioritet, slijedi dretva  $B$ , pa  $C$  te  $D$  koja ima najmanji. Dretve koriste tri sredstva koja su zaštićena semaforima  $S_1$ ,  $S_2$  i  $S_3$ . Sve dretve mogu trebati bilo koje sredstvo u svom izvođenju. Ponekad, dretva koja već ima jedno sredstvo može tražiti i drugo, ali ne i treće (prije traženja trećeg otpušta sva zauzeta). Pretpostavka je da se potpuni zastoј neće dogoditi . Ako sustav koristi raspoređivanje prema prioritetu te protokol nasljeđivanja prioriteta, koliko se najviše (u najgorem slučaju) može zaustaviti dretva  $A$  zbog inverzije prioriteta? Opisati scenarij u kojem se to događa.



# Slijede zadaci iz skripte, nemaju rj

17. U nekom sustavu javljaju se dretve:  $D_1$  u  $t_1 = 0$  ms,  $D_2$  u  $t_2 = 3$  ms te  $D_3$  u  $t_3 = 6$  ms. Svaka dretva sastoji se od tri dijela posla: A, B i C. Izvođenje A dijela traje 2 ms, B dijela 3 ms te C dijela 2 ms. Prije izvođenja B dijela dretve trebaju zauzeti semafore: dretva  $D_1$  semafor  $S_1$ , dretva  $D_2$  semafor  $S_2$ , dretva  $D_3$  semafor  $S_1$ . Dretva  $D_3$  ima najveći prioritet, dok dretva  $D_1$  ima najmanji. Ako se koristi protokol nasljeđivanja prioriteta pokazati izvođenje zadanih dretvi, tj. što procesor radi u pojedinom trenutku, dok sve dretve ne završe sa svojim poslovima.

18. Za neki sustav poznati su događaji pokretanja dretvi, njihovi poslovi i trajanja.

U  $t = 1$  pokreće se dretva  $D_1$ . Nakon 1 ms rada dretva treba zauzeti semafor  $S_1$ . Nakon još 2 ms rada (uz semafor  $S_1$ ) treba semafor  $S_2$ . Nakon još 1 ms rada (uz semafore  $S_1$  i  $S_2$ ) otpušta oba semafora. Nakon još 1 ms rada dretva završava. Ukupno, dretva treba 5 ms procesorskog vremena.

U  $t = 3$  pokreće se dretva  $D_2$ . Nakon 1 ms rada treba zauzeti semafor  $S_2$ . Nakon još 2 ms rada (uz semafor  $S_2$ ) treba semafor  $S_3$ . Nakon još 1 ms rada (uz semafore  $S_2$  i  $S_3$ ) otpušta oba semafora. Nakon još 1 ms rada dretva završava. Ukupno, dretva treba 5 ms procesorskog vremena.

U  $t = 5$  pokreće se dretva  $D_3$ . Nakon 1 ms rada treba zauzeti semafor  $S_3$ . Nakon još 2 ms rada (uz semafor  $S_3$ ) treba semafor  $S_1$ . Nakon još 1 ms rada (uz semafore  $S_1$  i  $S_3$ ) otpušta oba semafora. Nakon još 1 ms rada dretva završava. Ukupno, dretva treba 5 ms procesorskog vremena.

Dretva  $D_3$  ima najveći prioritet, slijedi  $D_2$  dok  $D_1$  ima najmanji prioritet. Prikazati rad sustava dretvi na jednoprocorskom sustavu koji koristi prioritetno raspoređivanje te:

a) nikakve druge protokole (niti nasljeđivanje prioriteta niti stropne protokole)

b) ako koristi protokol nasljeđivanja prioriteta

c) ako koristi jednostavniju inačicu protokola stropnog prioriteta

d) ako koristi izvorni protokol stropnog prioriteta.

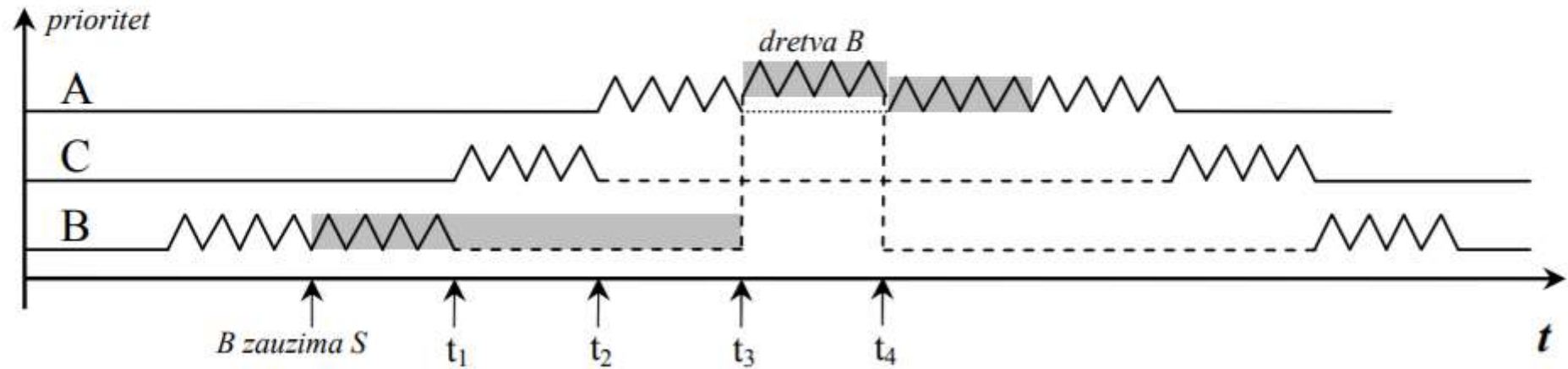


19. Dretve  $D_1 - D_4$  koriste semafor  $S$  po 100 ms, ali ne odmah po pokretanju već nakon 50 ms rada (svaka dretva najprije nešto radi 50 ms pa onda hoće semafor  $S$  za idućih 100 ms). Nakon otpuštanja semafora dretve rade nešto još 50 ms (dakle ukupno  $50+100+50$ , svaka dretva). Dretva  $D_2$  javlja se prva u  $t = 0$  ms. Slijedi  $D_3$  u  $t = 100$ ,  $D_4$  u  $t = 100$  te  $D_1$  u  $t = 150$ . Prioritet dretvi određen je njenim indeksom:  $D_4$  ima najveći a  $D_1$  namanji prioritet. Ako se dretve izvode na dvoprocesorskom sustavu i koristi se protokol nasljeđivanja prioriteta (uz prioritetni raspoređivač), pokazati rad sustava.
20. U nekom jednoprocorskom sustavu izvode se dretve  $A$ ,  $B$ ,  $C$  i  $D$ . Dretva  $A$  ima najveći prioritet, slijedi dretva  $B$ , pa  $C$  te  $D$  koja ima najmanji. Dretve koriste tri sredstva koja su zaštićena semaforima  $S_1$ ,  $S_2$  i  $S_3$ . Sve dretve mogu trebati bilo koje sredstvo u svom izvođenju. Ponekad, dretva koja već ima jedno sredstvo može tražiti i drugo, ali ne i treće (prije traženja trećeg otpušta sva zauzeta). Pretpostavka je da se potpuni zastoj neće dogoditi. Kada neka dretva zauzme sredstvo, ona ga ne koristi (aktivno) dulje od  $10 \mu s$ . Ako sustav koristi raspoređivanje prema prioritetu te protokol nasljeđivanja prioriteta, koliko se najviše (u najgorem slučaju) može zaustaviti dretva  $A$  zbog inverzije prioriteta? Opisati scenarij u kojem se to događa.

Slijedi dio teorije s ispita, ima točnih odgovora

# 2010 – ZI

5. Dretve A i B, različita prioriteta, koriste isto sredstvo S prema slici ispod. Osim njih u sustavu se pojavljuje i dretva C (u  $t_1$ ). Opisati što se dogodilo u trenucima  $t_1$ ,  $t_2$ ,  $t_3$  i  $t_4$ . Sustav koristi raspoređivanje prema prioritetu.





u t3 se pojavljuje inverzija prioriteta te se primjenjuje *protokol nasljeđivanja prioriteta*

# 2010 – ZI

6. Sinkronizacijski mehanizmi su često neophodni u višedretvenom programu. Međutim, njihova uporaba mora biti pažljivo analizirana jer može uzrokovati mnoge probleme. Koji su to problemi? Kako ih se može riješiti/izbjeći/ublažiti?

potpuni zastoј, inverzija prioriteta

2010 – 2MI

3. (2 boda) Zašto je *problem inverzije prioriteta* „problem“ za RT sustave?

*Problem je stoga što se zadatak većeg prioriteta ("bitniji zadatak") odgađa zbog zadatka manjeg prioriteta, i to ne samo zadatka koji trenutno ima traženo sredstvo, već i ostalih koji i ne moraju imati nikakve veze s tim sredstvom na kome se zadatak većeg prioriteta blokira.*



# Teorija bez točnih odgovora

11. (1) Što je to rekurzivno zaključavanje i zašto je ponekad potrebno?
  12. (1) Koja su uobičajena dodatna proširenja sinkronizacijskih i komunikacijskih mehanizama često potrebna u RTS okruženju (ali ne samo u njemu)?
6. Što je to potpuni zastoj? Kad se može pojaviti? Kako se izbjegava?
  7. Što je to rekurzivno zaključavanje i zašto je ponekad potrebno?
  8. Na primjeru prikazati nastajanje problema inverzije prioriteta.
  9. Opisati izvorni protokol stropnog prioriteta te njegovo pojednostavljenje.
  10. Opisati protokol nasljeđivanja prioriteta. Prikazati rad protokola nad primjerima.
  11. Nad kojim sinkronizacijskim mehanizmima definiranih POSIX normom su podržani protokoli stropnog prioriteta te protokol nasljeđivanja prioriteta?