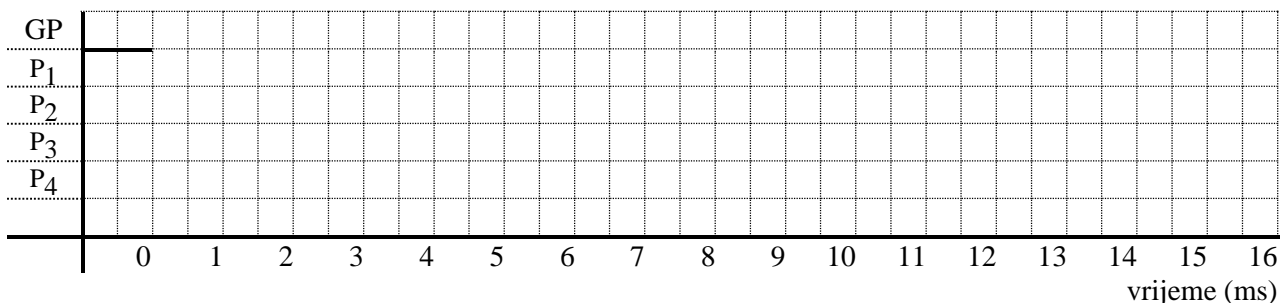


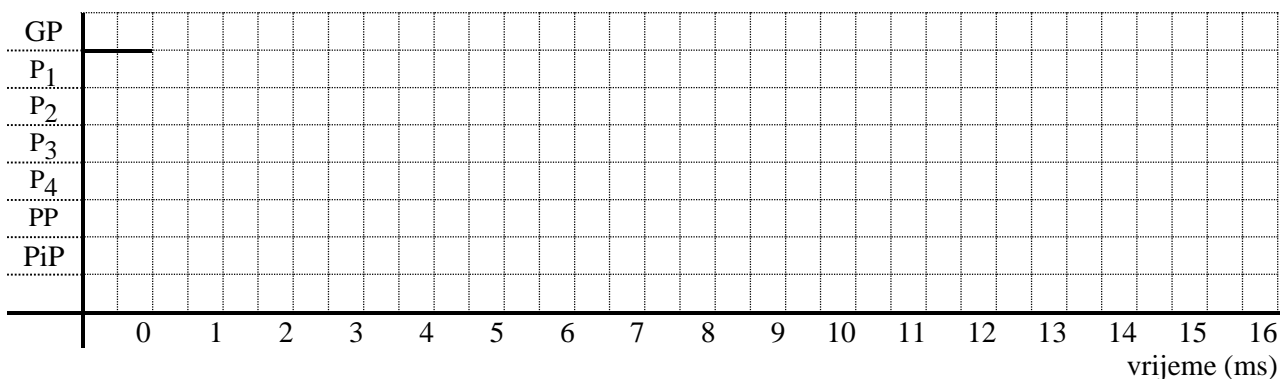
ZEMRIS	Ime i prezime	JMBAG
Operacijski sustavi, međuispit		

1. (Zadatak nosi ukupno 7 bodova.) U nekom sustavu javljaju se prekidi P3 u 0 ms, P2 u 1 ms i oba prekida P1 i P4 istovremeno u 3 ms. Prioritet prekida određen je brojem (P4 ima najveći prioritet). Obrada svakog prekida traje po 2 ms. Grafički prikazati aktivnosti procesora u glavnom programu (GP), procedurama za obradu prekida (Pi) te procedurama za prihvat prekida (PP) i povratak iz prekida (PiP) i to:

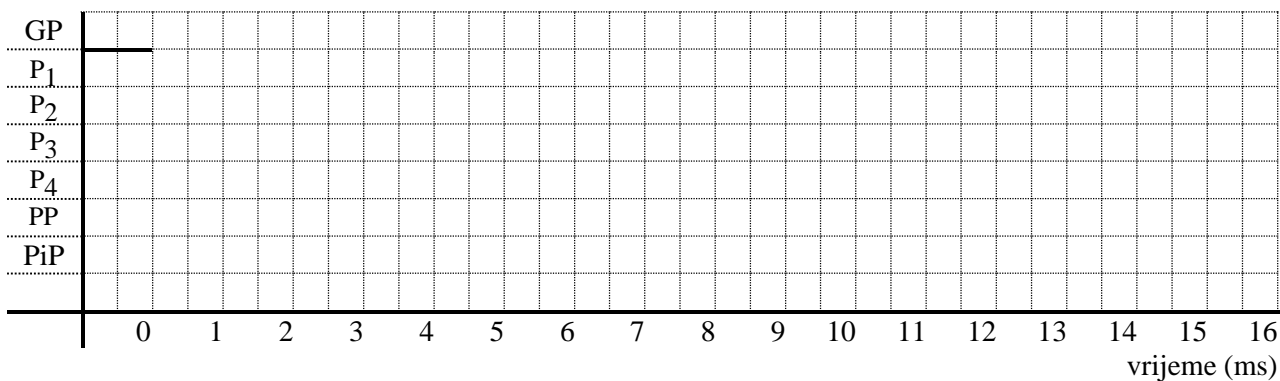
a) (1) u idealnom slučaju



b) (2) **bez sklopa za prihvat prekida, bez programske potpore i obrada uz zabranjeno prekidanje** uz trajanje prihvata prekida (PP) od 1 ms te trajanja povratka iz prekida (PiP) od 0,5 ms



c) (2) **sa sklopom za prihvat prekida** uz trajanje prihvata prekida od 0,5 ms (PP) te trajanje povratka iz prekida od 0,5 ms (PiP)



d) (2) Navesti sve registre **sklopa za prihvat prekida** i njihov sadržaj kao i stanje na sustavskom stogu u trenutku 2ms:

sustavski stog: _____

registri: _____

Prostor za postupak:

2. (Zadatak nosi ukupno 6 bodova.)

(0,5) U sustavu sa sklopom za prihvata prekida prioritet prekida određuje **prekidna rutina / procesor / sklop za prihvata prekida / jezgra OS-a / pristupni sklop**, a za vrijeme obrade prekida prekidanje je / nije omogućeno (zaokružiti ili potcrtati točne odgovore.).

(0,5) Prekidna rutina na svom početku prvo _____

(0,5) Sklop s neposrednim pristupom spremniku ima sljedeća 4 registra (dovoljno je navesti samo kratice)

(0,5) Koje registre sadrži sklop za prihvata prekida? _____

(0,5) Navesti strukture podataka koje koristi prekidna rutina za obradu prekida prema prioritetima u sustavu bez sklopa za prihvata prekida _____

(0,5) Što je hipervizor? _____

(0,5) Jedan od uvjeta koji mora zadovoljavati algoritam međusobnog isključivanja je i da dretva koja zastane u kritičnom odsječku ne smije spriječiti drugu dretvu da uđe u kritični odsječak? **DA / NE**

(0,5) Ako se pozove `j_fja Postavi_BSEM(i)` kada je semafor neprolazan i u redu `BSEM(i)` je jedna dretva tada će se dogoditi sljedeće _____

(1) Ulazak u jezgru zbiva se kada se dogodi _____ i tada se poziva _____

(1) Za ispravnu sinkronizaciju tri potrošača i jednog proizvođača preko ograničenog međuspremnik potrebno je _____ opća semafora i _____ binarna semafora.

3. (3) (Postupak ili obrazloženje obavezno navesti na košuljici) Navedeni program će slovo A ispisati _____ puta, slovo B _____ puta i slovo C _____ puta (navesti koliko puta).

```
#include <stdio.h>
int main() {
    fork();
    if (fork() == 0) {
        printf("A\n");
        if (fork() != 0) printf("B\n");
    }
    printf("C\n");
    return 0;
}
```

4. (4) U promatranom trenutku stanje sustava je sljedeće: dretva 1 je aktivna; dretve 2, 3 i 4 su u redu općeg semafora S (dretva 2 je prva u redu, a dretva 4 zadnja) te dretve 5, 6 i 7 su u redu pripravnih dretvi (dretva 5 je prva u redu, a dretva 7 je zadnja). Svi redovi organizirani su po redu prispjeća (FIFO). Ako tada dretva 1 pozove jezgrinu funkciju `PostaviOSem(S)`, kako će izgledati struktura podataka jezgre nakon poziva?

red Aktivna_D: _____	red Pripravne_D: _____
red OSEM[S]: _____	OSEM[1].v = _____

Postupak:

5. (Zadatak nosi ukupno 3 boda.) Sustav zadataka je zadan u obliku lanca: $Z_1 \rightarrow Z_2 \rightarrow Z_3 \rightarrow Z_4 \rightarrow Z_5 \rightarrow Z_6 \rightarrow Z_7 \rightarrow Z_8$, a zadaci imaju domene (D) i kodomene (K) smještene u memorijske lokacije M_1, M_2, M_3, M_4 , i M_5 prema tablici:

	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8
M1			K		K	D		
M2	D	K						K
M3		D	D	D,K			K	
M4	K		D		D	K		
M5							D	D

- a) (2) Skicirati maksimalno paralelni sustav zadataka uzimajući u obzir njihov međusobni odnos u lancu.
Mjesto za postupak i rješenje:

- b) (0,5) Za sinkronizaciju maksimalno paralelnog sustav zadataka uzimajući u obzir njihov međusobni odnos u lancu potrebno je _____ općih semafora.
c) (0,5) Zadaci 3 i 8 su **zavisni** / **nezavisni** (zaokružiti).

- ```
Dretva D \mathbf{x} {
 za(i=1 do \mathbf{x}){
 čekajBSEM(S);
 piši(P \mathbf{x});
 PostaviBSEM(S);
 piši(Z \mathbf{x});
 }
}
```

- | podatak ulaz, izlaz; //globalne varijable                                                                                                                                                                                                         |                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                          |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dretva ulazna(){<br>podatak a; //lokalna varijabla<br>while(1){<br><br>.....<br><br>.....<br><br>.....<br><br>a = dohvati_podatak();<br><br>.....<br><br>.....<br><br>.....<br><br>ulaz = a;<br><br>.....<br><br>.....<br><br>.....<br><br>}<br>} | dretva radna(){<br>podatak a; //lokalna varijabla<br>while(1){<br><br>.....<br><br>.....<br><br>.....<br><br>a = obradi_podatak(ulaz);<br><br>.....<br><br>.....<br><br>.....<br><br>izlaz = a;<br><br>.....<br><br>.....<br><br>.....<br><br>}<br>} | dretva izlazna(){<br>podatak a; //lokalna varijabla<br>while(1){<br><br>.....<br><br>.....<br><br>.....<br><br>A = izlaz;<br><br>.....<br><br>.....<br><br>.....<br><br>pohrani(a);<br><br>.....<br><br>.....<br><br>.....<br><br>}<br>} |

---

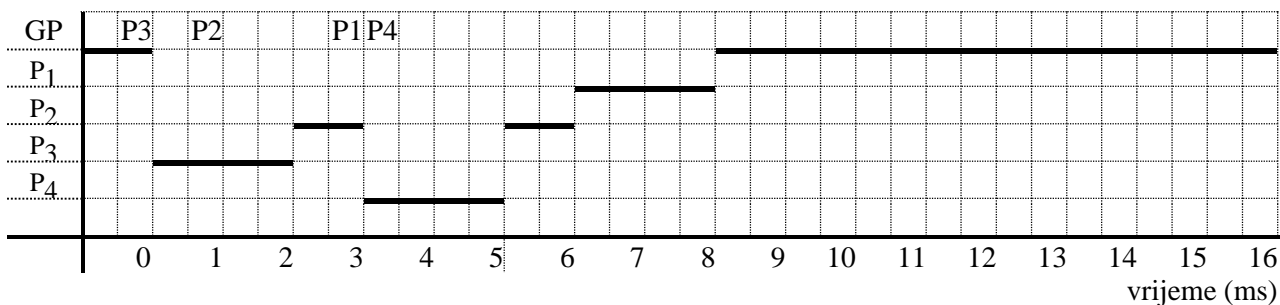


---

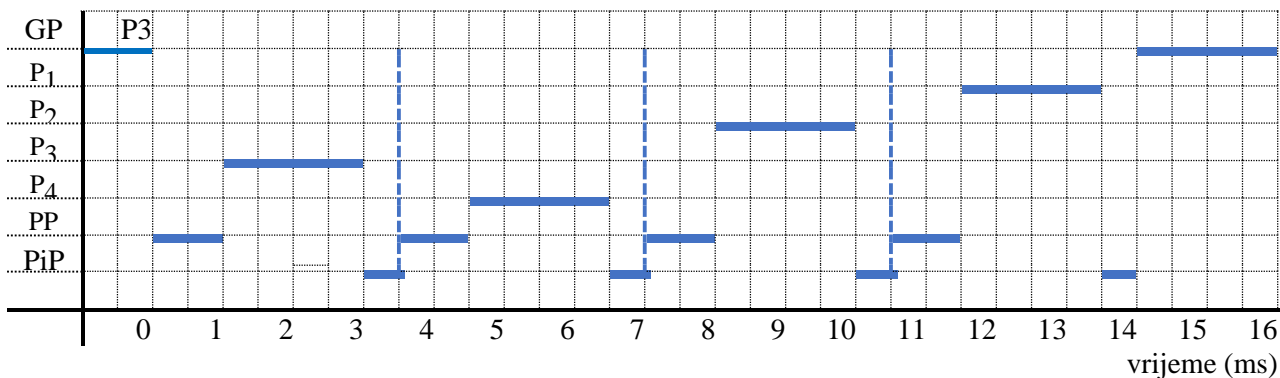
## RJEŠENJA

1. (Zadatak nosi ukupno 7 bodova.) U nekom sustavu javljaju se prekidi P3 u 0 ms, P2 u 1 ms i oba prekida P1 i P4 istovremeno u 3 ms. Prioritet prekida određen je brojem (P4 ima najveći prioritet). Obrada svakog prekida traje po 2 ms. Grafički prikazati aktivnosti procesora u glavnom programu (GP), procedurama za obradu prekida (Pi) te procedurama za prihvata prekida (PP) i povratka iz prekida (PiP) i to:

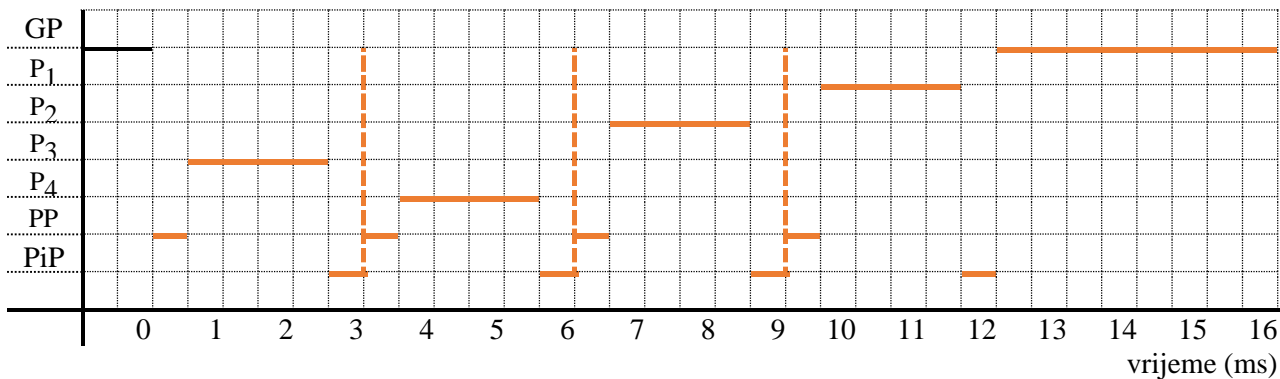
a) (1) u idealnom slučaju



b) (2) bez sklopa za prihvata prekida, bez programske potpore i obrada uz zabranjeno prekidanje uz trajanje prihvata prekida (PP) od 1 ms te trajanja povratka iz prekida (PiP) od 0,5 ms



c) (2) sa sklopom za prihvata prekida uz trajanje prihvata prekida od 0,5 ms (PP) te trajanje povratka iz prekida od 0,5 ms (PiP)



d) (2) Navesti sve registre sklopa za prihvata prekida i njihov sadržaj kao i stanje na sustavskom stogu u trenutku 2ms:

sustavski stog: 00000, reg[0]

registri: K\_Z=0100 i T\_P=0010

| vrijeme | K_Z  | T_P  | sustavski stog |
|---------|------|------|----------------|
| t < 0   | 0000 | 0000 | —              |
| t = 0   | 0010 | 0000 | —              |
|         | 0000 | 0010 | 00000, reg[0]  |
| t = 1   | 0100 | 0010 | 00000, reg[0]  |
| t = 2   | 0100 | 0010 | 00000, reg[0]  |

2. (Zadatak nosi ukupno 6 bodova.)

(0,5) (Zaokružiti točne odgovore.) U sustavu sa sklopom za prihvatanje prekida prioritet prekida određuje **prekidna rutina / procesor / sklop za prihvatanje prekida / jezgra OS-a / pristupni sklop**, a za vrijeme obrade prekida prekidanje **je / nije** omogućeno.

(0,5) Prekidna rutina na svom početku prvo **pohrani kontekst**

(0,5) Sklop s neposrednim pristupom spremniku ima sljedeća 4 registra (dovoljno je navesti samo kratice)  
\_\_\_\_\_ **PR, RS, AR i BR**

(0,5) Koje registre sadrži sklop za prihvatanje prekida? **K\_Z i T\_P (kontrolne zastavice i tekući prioritet)**

(0,5) Navesti strukture podataka koje koristi prekidna rutina za obradu prekida prema prioritetima u sustavu bez sklopa za prihvatanje prekida

\_\_\_\_\_ **T\_P, K\_Z i KON[N] (tekući prioritet, kontrolne zastavice i prostor za pohranu konteksta)**

(0,5) Što je hipervizor? **Upravljač virtualnim strojevima**

(0,5) Jedan od uvjeta koji mora zadovoljavati algoritam međusobnog isključivanja je i da dretva koja zastane u kritičnom odsječku ne smije spriječiti drugu dretvu da uđe u kritični odsječak? **DA / NE**

(0,5) Ako se pozove j\_fja Postavi\_BSEM(i) kada je semafor neprolazan i u redu BSEM(i) je jedna dretva tada će se dogoditi sljedeće **dretva iz reda BSEM(i) premjestiti u red Pripravne\_D**

(1) Ulazak u jezgru zbiva se kada se dogodi \_\_\_\_\_ **prekid** \_\_\_\_\_ i tada se poziva j\_fja \_\_\_\_\_

(1) Za ispravnu sinkronizaciju tri potrošača i jednog proizvođača preko ograničenog međuspremnik potrebno je \_\_\_\_\_ **2** \_\_\_\_\_ opća semafora i \_\_\_\_\_ **1** \_\_\_\_\_ binarna semafora.

3. (3) Navedeni program će slovo A ispisati \_\_\_\_\_ **2** \_\_\_\_\_ puta, slovo B \_\_\_\_\_ **2** \_\_\_\_\_ puta i slovo C \_\_\_\_\_ **6** \_\_\_\_\_ puta.

```
#include <stdio.h>
int main(){
 fork();
 if (fork() == 0) {
 printf("A\n");
 if (fork() != 0) printf("B\n");
 }
 printf("C\n");
 return 0;
}
```

4. (4) U promatranom trenutku stanje sustava je sljedeće: dretva 1 je aktivna; dretve 2, 3 i 4 su u redu općeg semafora S (dretva 2 je prva u redu, a dretva 4 zadnja) te dretve 5, 6 i 7 su u redu pripravnih dretvi (dretva 5 je prva u redu, a dretva 7 je zadnja). Svi redovi organizirani su po redu prispjeka (FIFO). Ako tada dretva 1 pozove jezgrinu funkciju PostaviOSem(S), kako će izgledati struktura podataka jezgre nakon poziva?

red Aktivna\_D: \_\_\_\_\_ **5** \_\_\_\_\_                      red Pripravne\_D: \_\_\_\_\_ **6, 7, 1, 2** \_\_\_\_\_  
red OSEM[1]: \_\_\_\_\_ **3, 4** \_\_\_\_\_                      OSEM[1].v = \_\_\_\_\_ **0** \_\_\_\_\_

Prije poziva PostaviOSem(S):

Aktivna\_D: 1

Red Pripravne\_D: 5, 6, 7

Red OSEM[S]: 2, 3, 4

Međurezultati:

-

5, 6, 7, 1

2, 3, 4

-

5, 6, 7, 1, 2

3, 4

Poslije poziva PostaviBSem(S):

Aktivna\_D: 5

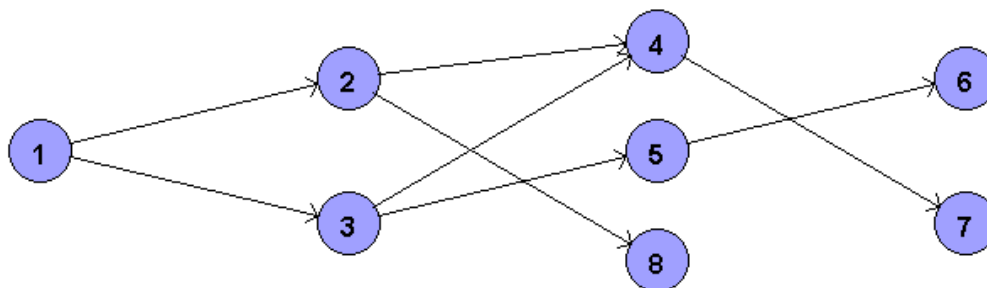
Red Pripravne\_D: 6, 7, 1, 2

Red OSEM[S]: 3, 4 i OSEM[S].v=0

5. (Zadatak nosi ukupno 3 boda.) Sustav zadataka je zadan u obliku lanca:  $Z_1 \rightarrow Z_2 \rightarrow Z_3 \rightarrow Z_4 \rightarrow Z_5 \rightarrow Z_6 \rightarrow Z_7 \rightarrow Z_8$ , a zadaci imaju domene (D) i kodomene (K) smještene u memorijske lokacije  $M_1, M_2, M_3, M_4$ , i  $M_5$  prema tablici:

|    | Z1 | Z2 | Z3 | Z4  | Z5 | Z6 | Z7 | Z8 |
|----|----|----|----|-----|----|----|----|----|
| M1 |    |    | K  |     | K  | D  |    |    |
| M2 | D  | K  |    |     |    |    |    | K  |
| M3 |    | D  | D  | D,K |    |    | K  |    |
| M4 | K  |    | D  |     | D  | K  |    |    |
| M5 |    |    |    |     |    |    | D  | D  |

d) (2) Skicirati maksimalno paralelni sustav zadataka uzimajući u obzir njihov međusobni odnos u lancu.



e) (0,5) Za sinkronizaciju maksimalno paralelnog sustav zadataka uzimajući u obzir njihov međusobni odnos u lancu potrebno je 7 općih semafora.

f) (0,5) Zadaci 3 i 8 su zavisni / **nezavisni** (zaokružiti).

6. (3) (Postupak navesti na košuljici.) U jednoprocorskom računalu pokrenut je sustav dretvi  $D1, D2$  i  $D3$  s prioritetima 1, 2 i 3, tim redom. Najviši prioritet je 3. Svi zadaci koje obavljaju dretve su istog oblika  $Dx$ . Red pripravnih dretvi i red semafora su prioritetni. Aktivna je dretva koja je prva u redu pripravnih (nema posebnog reda aktivnih dretvi). Prije pokretanja sustava dretvi semafor  $S$  je bio neprolazan. Nakon nekog vremena sve dretve se nađu u redu semafora  $S$ . Ako se tada pozove procedura PostaviBSEM( $S$ ) na zaslonu će se ispisati do završetka rada svih dretvi (važan je i redoslijed ispisa):

```

Dretva Dx{
 za(i=1 do x){
 ČekajBSEM(S);
 piši(Px);
 PostaviBSEM(S);
 piši(Zx);
 }
}

```

      P3 Z3 P2 P3 Z3 Z2 P1 P3 Z3 P2 Z2 Z1      

| Red semafora | BSEM.v | Red pripravnih      | P | Z |
|--------------|--------|---------------------|---|---|
| 321          | 0      | -                   |   |   |
| 21           | 0      | 3a                  | 3 |   |
| 1            | 0      | 3b2a                |   | 3 |
| 31           | 0      | 2a                  | 2 |   |
| 1            | 0      | 3a2b                | 3 |   |
| -            | 0      | 3b2b1a              |   | 3 |
| 3            | 0      | 2b1a                |   | 2 |
| 32           | 0      | 1a                  | 1 |   |
| 2            | 0      | 3a1b                | 3 |   |
| -            | 0      | 3b2a1b (3 završava) |   | 3 |
| -            | 0      | 2a1b                | 2 |   |
| -            | 1      | 2b1b (2 završava)   |   | 2 |
| -            | 1      | 1b (1 završava)     |   | 1 |

7. (4) Neki problem riješen je s pomoću četiri dretve: jednom ulaznom dretvom, dvije radne dretve i jednom izlaznom dretvom. Ulazna dretva dobavlja podatke preko globalne varijable *ulaz* te podatke prosljeđuje radnim

dretvama na obradu. Radna dretva rezultat zapisuje u globalnu varijablu *izlaz* koju na kraju izlazna dretva pohranjuje. Sinkronizirati dretve binarnim semaforima, tj. nadopuniti prikazani kod **isključivo jezgrinim funkcijama** **ČekajBSEM(i)** i **PostaviBSEM(j)** ('i' i 'j' zamjeniti brojevima) te navesti početne vrijednosti semafora.

|                                                                                                                                                                  |                                                                                                                                                                                                                   |                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> podatak ulaz, izlaz; //globalno dretva ulazna(){     podatak a;     while(1){         a = dohvati_podatak();         ulaz = a;     } } </pre>              | <pre> dretva radna(){     podatak a;     while(1){         a = obradi_podatak(ulaz);         izlaz = a;     } } </pre>                                                                                            | <pre> dretva izlazna(){     podatak a;     while(1){         a = izlaz;         pohrani(a);     } } </pre>                                               |
| <pre> dretva ulazna(){     podatak a;     while(1){         a =dohvati_podatak();         ČekajBSEM(1);         ulaz = a;         PostaviBSEM(2);     } } </pre> | <pre> dretva radna(){     podatak a;     while(1){         ČekajBSEM(2);         a =obradi_podatak(ulaz);         PostaviBSEM(1);         ČekajBSEM(3);         izlaz = a;         PostaviBSEM(4);     } } </pre> | <pre> dretva izlazna(){     podatak a;     while(1){         ČekajBSEM(4);         a = izlaz;         PostaviBSEM(3);         pohrani(a);     } } </pre> |

Semafori njihove i početne vrijednosti: **BSEM(2).v=BSEM(4).v= 0, a BSEM(1).v=BSEM(3).v=1**