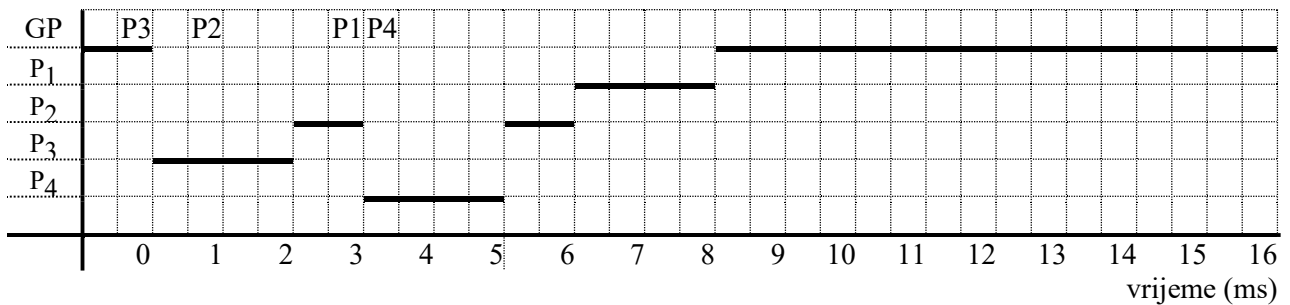


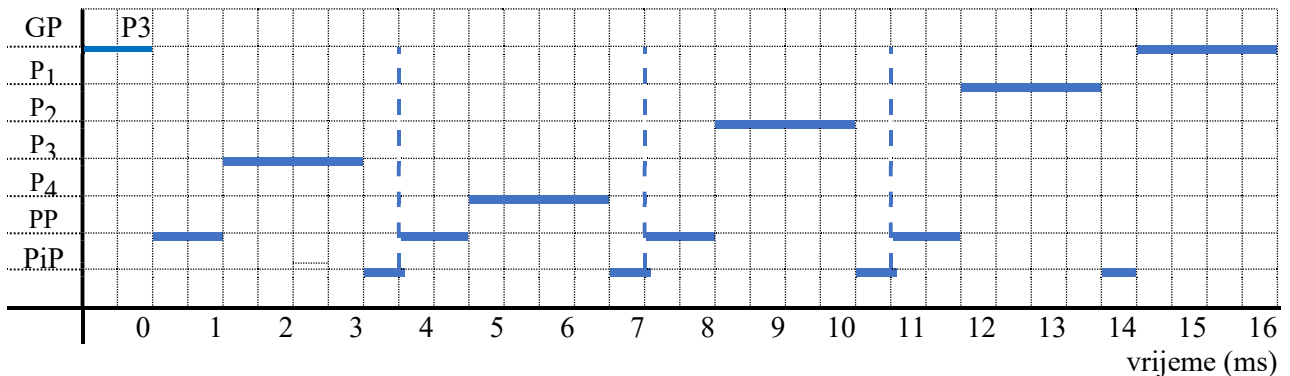
RJEŠENJA

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:

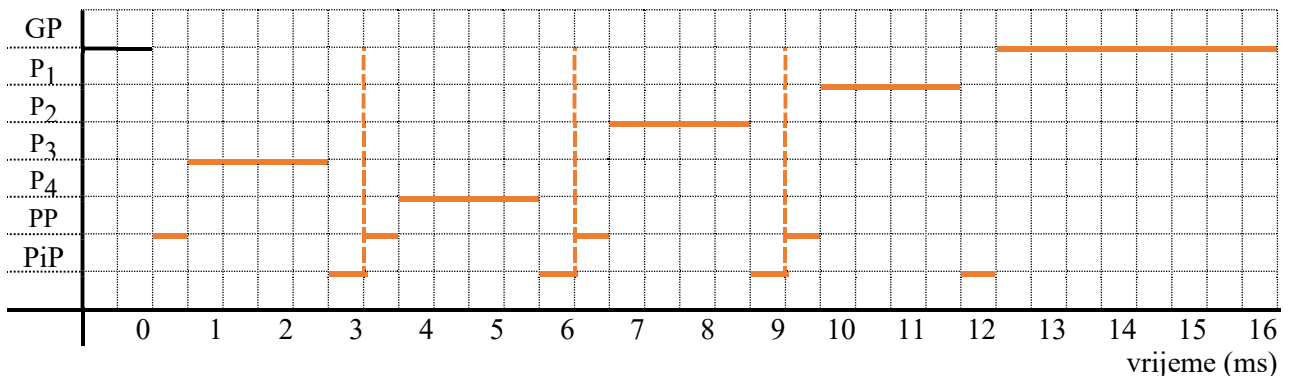
a) (1) u idealnom slučaju



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

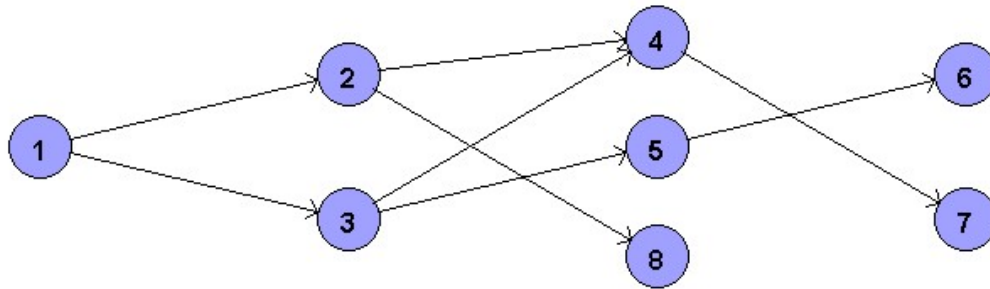
Red Pripravne_D: 6, 7, 1, 2

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 proslijeđuje 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**