

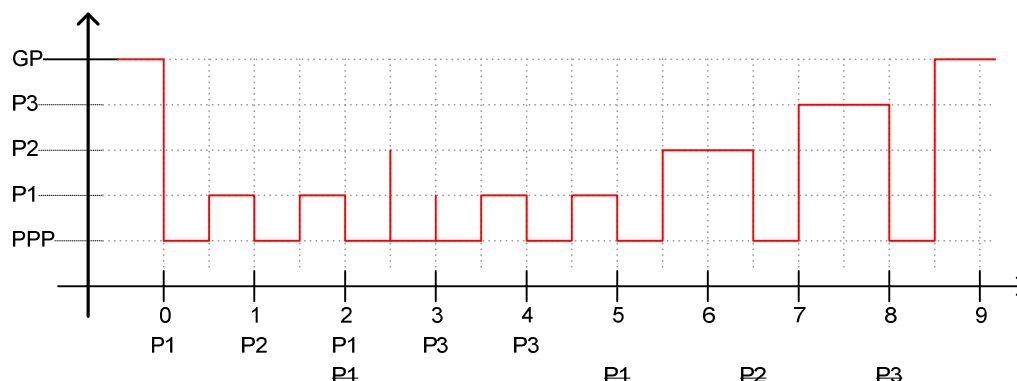
Prezime, ime, MBR	Grupa
	2.R____

Rbr	Bodovi

Grupa B

Prvi međuispit iz predmeta Operacijski sustavi

- U sustavu s minimalnim sklopovljem (bez sklopa za prihvat prekida) javljaju se prekidi: P1 u trenucima 0. i 2. ms; P2 u 1. ms te P3 u 3. i 4. ms. Prioritet prekida određen je brojem: P1 ima najviši prioritet, slijedi P2 te P3 koji ima najmanji. Obrada svakog prekida traje 1 ms, dok procedura za prihvat prekida (PPP) traje 0,5 ms.
- Grafički prikazati aktivnosti procesora u glavnom programu (GP), procedurama za obradu prekida (P_i) te proceduri za prihvat prekida (PPP).



- Navesti strukture podataka koje koristi procedura za prihvat prekida (PPP) i njihove vrijednosti u trenutku $t = 6$ ms.

 $T_P = 2$ $K_Z[1] = 0, K_Z[2] = 0, K_Z[3] = 1$ $KON[3] =$ nije definirano (nije korišteno u primjeru do 6. ms) $KON[2] = \{ 0, reg[0] \}$ $KON[1] =$ ništa/ne koristi se (već je iskorišteno: bilo je $\{ 0, reg[0] \}$ ali je maknuto u $KON[2]$)

- UNIX sustavi koriste 30-tak signala. Navedite bar 4 od njih: **SIGINT, SIGUSR1, SIGUSR2, SIGTERM, SIGKILL, SIGALRM, SIGQUIT, ...**

- Dva su zadatka nezavisna ako: $(X_i \cap Y_j) \cup (X_j \cap Y_i) \cup (Y_i \cap Y_j) = \emptyset$.

(ili ako im je presjek domene prvog i kodomene drugog zadatka prazan skup, i obratno, presjek kodomene prvog i domene drugog prazan skup te presjek kodomena također mora biti prazan skup)

- Navesti 4 zahtjeva na algoritme međusobnog isključivanja:

- * u kritičnom odsječku u svakom trenutku smije biti najviše jedna dretva

- * mehanizam međusobnog isključivanja mora djelovati i u uvjetima kada su brzine izvođenja dretvi proizvoljne

- * kada neka od dretvi zastane u svom nekritičnom dijelu ona ne smije spriječiti ulazak druge dretve u svoj kritični odsječak

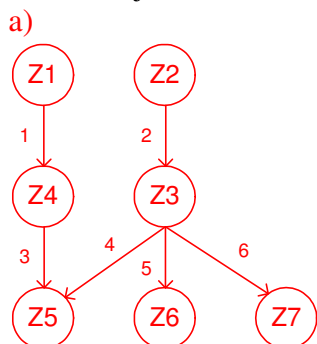
- * izbor jedne od dretvi koja smije ući u kritični odsječak treba obaviti u konačnom vremenu

2. Sustav zadataka zadan je u obliku lanca: $Z1 \rightarrow Z2 \rightarrow Z3 \rightarrow Z4 \rightarrow Z5 \rightarrow Z6 \rightarrow Z7$. Zadaci koriste određene spremničke segmente prema tablici.

- a) (2) Odrediti i nacrtati maksimalno paralelni sustav zadataka.
b) (1) Sinkronizirati sustav sa minimalnim brojem binarnih semafora, odnosno, ukoliko je T_i tekst zadatka i proširiti zadatke $Z1$, $Z4$ i $Z7$ pozivima procedura $\text{ČekajBSem}(j)$ i $\text{PostaviBSem}(k)$, te odrediti početne vrijednosti semafora.

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

- c) (1) Ako se sustav izvodi na 8-procesorskom sustavu (i nema drugih procesa/zadataka), odrediti trajanje izvođenja sustava zadataka sinkroniziranih prema b) dijelu zadatka, uz pretpostavku da svaki zadatak traje 200 ms.



- b)
- T1': T1; PostaviBSem(1);**
T2': T2; PostaviBSem(2);
T3': ČekajBSem(2); T3; PostaviBSem(4); PostaviBSem(5); PostaviBSem(6);
T4': ČekajBSem(1); T4; PostaviBSem(3);
T5': ČekajBSem(3); ČekajBSem(4); T5;
T6': ČekajBSem(5); T6;
T7': ČekajBSem(6); T7;

Početne vrijednosti: $\text{BSem}[i].v = 0$ za i iz skupa $\{1..6\}$

- c)
- Na višeprocorskom sustavu gdje za svaki zadatak postoji zaseban procesor (i još je jedan slobodan), ukupno trajanje biti će jednako najduljem putu na grafu. Najdulji put ima tri čvora, pa je prema tome trajanje: $3 * 200 \text{ ms} = 600 \text{ ms}$.

3. (3) U nekoj utrci sudjeluju puž, kornjača i jež. Pretpostavimo da je kornjača duplo brža od puža, a jež duplo brži od kornjače. Staza po kojoj se životinje utrkuju je kružna i duljine 1 metar. Svaka životinja pri prolasku kroz ciljnu ravninu ispisuje svoju oznaku i broj prijedjenih metara (prema primjeru sa strane). Simulirati životinje dretvama, tj. sinkronizirati dretve sukladno pretpostavljenim brzinama tako da one ispisuju potrebne poruke odgovarajućom frekvencijom i redoslijedom (kornjača prolazi jedan metar tek nakon što jež prođe dva uzastopna metra, a puž prolazi jedan metar tek nakon što kornjača prođe dva uzastopna metra). *Nadopuniti slijedeći kod ili rješavati na košuljici.*

Primjer ispisa:

```
Jež: 1 m
Jež: 2 m
Kornjača: 1 m
Jež: 3 m
Jež: 4 m
Kornjača: 2 m
Puž: 1 m
Jež: 5 m
...
```

```
Dretva Puž(){
    p = 0;
    ponavljaaj {
        p++; //prolazi krug

        ČekajBSem(1);

        Ispiši("Puž:", p);

        PostaviBSem(3);

    } dok je (1);
}
```

```
Dretva Kornjača(){
    k = 0;
    ponavljaaj {
        k++; //prolazi krug

        ČekajBSem(2);

        Ispiši("Kornjača:", k);

        ako je ('k' paran)
            PostaviBSem(1);
        inače
            PostaviBSem(3);

    } dok je (1);
}
```

```
Dretva Jež(){
    j = 0;
    ponavljaaj {
        j++; //prolazi krug

        ako je ('j' neparan)
            ČekajBSem(3);

        Ispiši("Jež:", j);

        ako je ('j' paran)
            PostaviBSem(2);

    } dok je (1);
}
```

Početne vrijednosti semafora: $\text{BSem}[1].v = 0$; $\text{BSem}[2].v = 0$; $\text{BSem}[3].v = 1$;