

1. (3) Što se nalazi između slojeva u računalnom sustavu (korisnik $\leftrightarrow$ programi $\leftrightarrow$ operacijski sustav $\leftrightarrow$ sklopovlje)?

**sučelje**

2. (2) Što je to kontekst dretve?

**sadržaj registara procesora u trenutku izvođenja dretve, ili pohranjeni pri prekidu izvođenja dretve**

3. (2) Navesti svojstva upravljanja ulazno izlaznim napravama korištenjem metode "radnog čeka-nja".

**prednosti: jednostavno sklopovlje**

**nedostaci: neproduktivan rad**

4. (2) Zadatak Z1 koristi objekte A1, A2, A3, ali promjene radi samo u A3. Zadatak Z2 koristi iste objekte ali radi promjene samo u A2. Zadatak Z3 koristi objekte A1 i A4, s time da radi promjene samo u A4. Koji su zadaci međusobno zavisni a koji nezavisni?

**Z1, Z2 => zavisni**

**Z1, Z3 => nezavisni**

**Z2, Z3 => nezavisni**

5. (3) Koje zahtjeve za algoritme međusobnog isključivanja ne zadovoljava algoritam?

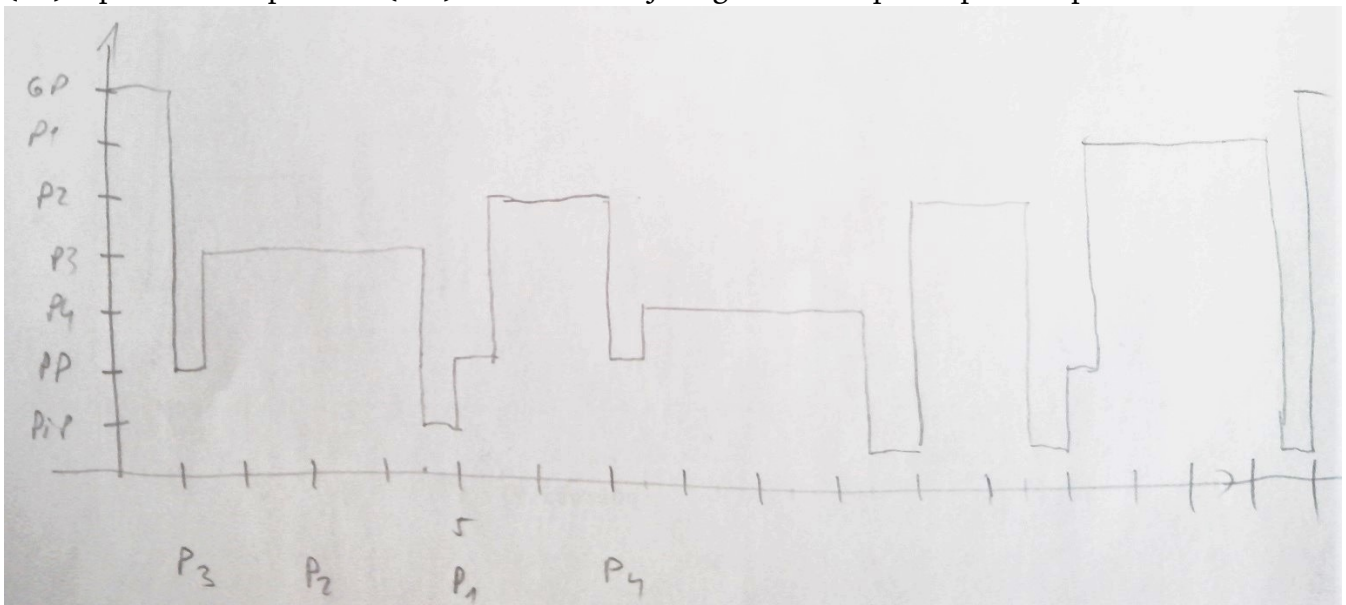
```

Uđi_u_KO(I){//I je ili 0 ili 1
    dok je PRAVO == J
        ;
    }
Izađi_iz_KO(I){
    PRAVO = J
}
    
```

**algoritam dopušta samo naizmjenični ulazak u kritični odsječak**

**dretva zaustavljena izvan KO utječe i na onu drugu: 2. i 3. zahtjevi nisu ispunjeni**

6. (3) U sustavu koji ima sklop za prihvat prekida pojavljuju se zahtjevi: P1 u 5. ms, P2 u 3. ms, P3 u 1. ms te P4 u 7. ms. Indeks zadatka označava prioritet (P4 ima najveći). Ukoliko prihvat prekida (PP) i povratak iz prekida (PIP) traju po 0.5 ms, a obrade prekida traju po 3 ms, pokazati rad procesora u glavnom programu (GP), obradama prekida (Pi) i procedurama za prihvat prekida (PP) i povratak iz prekida (PIP). Navesti stanje registara sklopa za prihvat prekida u t=6. ms.



7. (3) Početno stanje sustava je:

Red	početno	nakon a)	nakon b)	nakon c)	nakon d)	nakon e)	nakon f)
Aktivna_D:	4	3	3	4	3	4	3
Pripravne_D:	3 2 8	2 8 4 7	2 8	3 2 8	2 8 4 6	3 2 8	2 8
Odgodjene_D:	1 <sup>2</sup> , 5 <sup>1</sup>	1 <sup>2</sup> , 5 <sup>1</sup>	1 <sup>2</sup> , 5 <sup>1</sup>	1 <sup>2</sup> , 5 <sup>1</sup>	1 <sup>2</sup> , 5 <sup>1</sup>	1 <sup>1</sup> , 5 <sup>1</sup>	1 <sup>2</sup> , 5 <sup>1</sup> , 4 <sup>2</sup>
UI[1]:	6	6	6	6	6	6	6
OSEM[1]:	7 (.v=0)	-. (.v=0)	7 4 (.v=0)	7 (.v=0)	7 (.v=0)	7 (.v=0)	7 (.v=0)
OSEM[2]:	-. (.v=1)	-. (.v=1)	-. (.v=1)	-. (.v=0)	-. (.v=1)	-. (.v=1)	-. (.v=1)

Pokazati iduće stanje sustava ako se u tom stanju dogodi poziv: a) PostaviOSEM(1), b) ČekajOSEM(1), c) ČekajOSEM(2), d) Prekid\_UI(1), e) Otkucaj\_sata(), f) Odgodi(5).  
Navedeno su odvojeni slučajevi, svaki poziv raditi u odnosu na početno stanje.

8. (3) Za komunikaciju između ulazne, radne i izlazne dretve stoji na raspolaganju samo jedan međuspremnik M u koji stane samo jedna poruka. Taj se međuspremnik najprije koristi za komunikaciju između ulazne i radne dretve, te potom za komunikaciju između radne i izlazne te opet za ulaznu-radnu i tako dalje. Korištenjem binarnih semafora sinkronizirati rad tih dretvi. Nadopuniti idući kod.

```

dretva ulazna () {
    U = dohvati()
    ČekajBSEM(1)
    M = U
    PostaviBSEM(2)
}

dretva radna () {
    ČekajBSEM(2)
    R = M
    R = obrada(R)
    M = R
    PostaviBSEM(3)
}

dretva izlazna () {
    ČekajBSEM(3)
    I = M
    PostaviBSEM(1)
    pohrani(I)
}
    
```

9. (3) Dvije dretve proizvođača komuniciraju s jednom dretvom potrošača preko ograničenog međuspremnika M veličine N=10 poruka. Sinkronizirati te dretve korištenjem općih i binarnih semafora te dodatnih varijabli. Navesti početne vrijednosti.

```

dretva proizvođač () {
    ponavljaaj {
        P = dohvati()
        ČekajOSEM(1)
        ČekajBSEM(2)
        M[ULAZ] = P
        ULAZ = ( ULAZ + 1 ) MOD N
        PostaviBSEM(2)
        PostaviOSEM(3)
    }
}

dretva potrošač () {
    ponavljaaj {
        ČekajOSEM(3)
        P = M[IZLAZ]
        IZLAZ = ( IZLAZ + 1 ) MOD N
        PostaviOSEM(1)
    }
    potroši(P)
}

Početne vrijednosti:
OSEM[1].v = N
BSEM[2].v = 1
OSEM[3].v = 0
ULAZ = IZLAZ = 0
    
```