

Pisati čitko – nečitak odgovor ne donosi bodove.

1. [2 boda] Što je složenije za isprogramirati: sustav upravljanja koji koristi povratnu vezu ili sustav upravljanja koji NE koristi povratnu vezu? Obrazložiti.

onaj koji koristi povratnu vezu jer treba uzeti u obzir i stvarno stanje sustava koje dohvaća preko senzora

sustav koji ne koristi povratnu vezu pretpostavlja da je sustav u očekivanom stanju te samo izdaje naredbe u odgovarajućim trenucima

2. [2 boda] Zašto su sekvencijski UML dijagrami posebno zanimljivi s aspekta SRSV-a?

jer promatraju scenarije kroz vrijeme – ono je dio opisa, a vrijeme je vrlo bitno za SRSV

3. [2 boda] Zašto se svugdje ne koristi višedretvenost za upravljanje SRSV-a (već većinom obična upravljačka petlja)?

višedretvenost zahtijeva OS koji opet zahtijeva veće resurse (bolji procesor, više memorije)

također, višedretveni sustavi su složeniji, a većina problema je jednostavna gdje su jednostavna rješenja upravljanja bolja

4. [2 boda] Neki sustav od četiri zadatka ima iskoristivost $U = 0,8$. Ako se radi o raspoređivanju mjerom ponavljanja na jednoprocesorskom sustavu, što se sve može zaključiti o rasporedivosti samo na osnovu toga ($4(\sqrt[4]{2} - 1) = 0,75$)?

$U < 1 \Rightarrow$ nužni uvjet zadovoljen

$U > \text{lub}(U) \Rightarrow$ ne može se prema tome reći je li rasporediv, potrebno je nekom drugom metodom provjeriti rasporedivost

5. [2 boda] Zašto se raspoređivanje prema najmanjoj labavosti (LLF) ne koristi u praksi (a prema mjeri ponavljanja i prema kranjim trenucima završetaka ipak koristi)?

LLF koristi i C koji je vrlo promjenjiv u praksi te stoga ne daje smislene rezultate

također, kod LLF-a bi trebalo puno češće pozivati raspoređivač što povećava "overhead" samog raspoređivača

6. U nekom sustavu nalaze se četiri čvora: klijent K koji upućuje zahtjeve poslužitelju A, poslužitelj A koji zaprima i obrađuje zahtjeve klijenata, poslužitelj B s bazom podataka koju poslužitelj A kontaktira pri nekim operacijama s klijentom (prijava, izmjena postavki korisnika) te poslužiteljem C koji upravlja proizvodnim procesom na osnovu naredbi koje zaprima od poslužitelja A (na zahtjev klijenta). Operacije sustava uključuju prijavu korisnika PRIJAVA <id-korisnika>, promjene postavke korisnika POSTAVKA <id-korisnika>:{<ime>=vrijednost} te obrade zahtjeva korisnika: DOHVATI <id> (treba od C dohvatiti stanje objekata <id> sa svim njegovim vrijednostima) te POSTAVI <id>:{<ime>=vrijednost}.

- a) [2 boda] Za zadani sustav prikazati dijagram obrazaca upotrebe.
- b) [2 boda] Prikazati sekvencijski dijagram za scenarij sa slijedećim događajima:
- u $t=0$ korisnik X se prijavljuje u sustav
 - u $t=20$ X šalje naredbu `POSTAVI soba42:{rasvjeta=on}`.
 - u $t=30$ korisnik Y se prijavljuje u sustav

Pretpostaviti da poslužiteljima A i B treba po pet jedinica vremena za obradu zahtjeva te 20 poslužitelju C. Npr. ako u obradi zahtjeva korisnika poslužitelj A treba kontaktirati B, onda za cjelu obradu mu treba 5 (na A) + 5 (na B) + 5 (opet na A) = 15 jedinica vremena. Poslužitelji mogu paralelno obrađivati zahtjeve korisnika, višeprocorski su.

a)

```

PRIJAVA
korisnik      POSTAVKA      A
               DOHVATI
               POSTAVI

               dohvati podatke o korisniku
               pohrani podatke o korisniku      A B

               dohvati podatke stanje objekta
               pošalji naredbu                  A C

```

b) sekvencijski

```

      X      Y      A      B      C
t=0  *----->*
t=5              *--->*
t=10              *<---*
t=15 *<-----*
t=20 *----->*
t=25              *----->*
t=30          *--->*              *
t=35              *--->*              *
t=40              *<---*              *
t=45          *<---*<-----*
t=50 *<-----*

```

7. [2 boda] Neki PID regulator zadan je parametrima $K_P = 0,5$, $K_I = 0,2$ i $K_D = 0,05$. Ako je stanje sustava u nekom trenutku $y_k = 100$, željeno stanje $y_P = 200$ izračunati r_k (vrijednost s kojom se utječe na sustav)? Prethodno stanje sustava je $y_{k-1} = 80$, uz $I_{k-1} = 20$. Korak integracije je $T = 0,1$ s.

```

Kp=0,5; Ki=0,2; Kd=0,05; yk=100
yp=200      => ek=yp-yk=100
yk-1=80     => ek-1=120
              Dk = (ek-ek-1)/T = (100-120)/0,1 = -200
Ik-1=20     => Ik = Ik-1 + ek*T = 20 + 100*0,1 = 30
T = 0,1     rk = Kp*ek + Ki*Ik + Kd * Dk
              = 0,5*100 + 0,2*30 - 0,05*200 = 50+6-10 = 46

```

8. [3 boda] Neki upravljački program treba točno svakih $T=50$ ms (ili najbliže moguće tom trenutku) obaviti posao $op_a()$, jednom u svakom intervalu od 35 ms operaciju $op_b()$ te jednom u svakom intervalu od 150 ms posao $op_c()$. Prvo pokretanje proizvoljno odabrati (unutar prvog intervala). Pretpostaviti da operacije neće trajati duže od 5 ms. Posao $op_a()$ se ne smije prekidati prekidima u svom izvođenju dok poslovi $op_b()$ i $op_c()$ mogu. Pseudokodom napisati upravljački program ukoliko na raspolaganju stoji sučelje: $dohvati_vrijeme_ms()$, $zabrani_prekide()$, $dozvoli_prekide()$ te funkcije $op_a()$, $op_b()$ i $op_c()$ postoje.

```
t1=0
t2=10
t3=30
ponavljaaj {
    ako je dohvati_vrijeme() >= t1 {
        zabrani_prekide()
        op_a()
        dozvoli_prekide()
        t1 += 50
    }
    inače ako je dohvati_vrijeme() >= t2 {
        op_b()
        t2 += 35
    }
    inače ako je dohvati_vrijeme() >= t3 {
        op_c()
        t3 += 150
    }
}
```

9. [3 boda] Za zadani sustav zadataka grafičkim postupkom provjeriti rasporedivost postupkom prema mjeri ponavljanja (RMPA) za jednoprocesorski sustav te iz grafa odrediti implicitne trenutke krajnjeg dovršetka za sve zadatke.

$$\mathcal{T}_1: T_1 = 10 \text{ ms}, C_1 = 3 \text{ ms}$$

$$\mathcal{T}_2: T_2 = 15 \text{ ms}, C_2 = 5 \text{ ms}$$

$$\mathcal{T}_3: T_3 = 20 \text{ ms}, C_3 = 4 \text{ ms}$$

```
P:  1 1 1 2 2 2 2 2 3 3 1 1 1 3 3 2 2 2 2 2 1
t:  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
    1                               1                               1
    2                               2
    3                               |                               3
                                v
                                3 gotov, rasporedivo
```

Implicitni deadline:

T1: 10

T2: 15

t3: 15

10. [4 boda] Za zadani sustav zadataka grafički prikazati izvođenje sustava kada se za raspoređivanje koristi postupak prema krajnjem trenutku završetka (EDF + red prispjeća kao sekundarni kriterij) na dvoprocesorskom računalu do $t = 20$ ms.

$$\mathcal{T}_1: T_1 = 5 \text{ ms}, \quad C_1 = 3 \text{ ms}$$

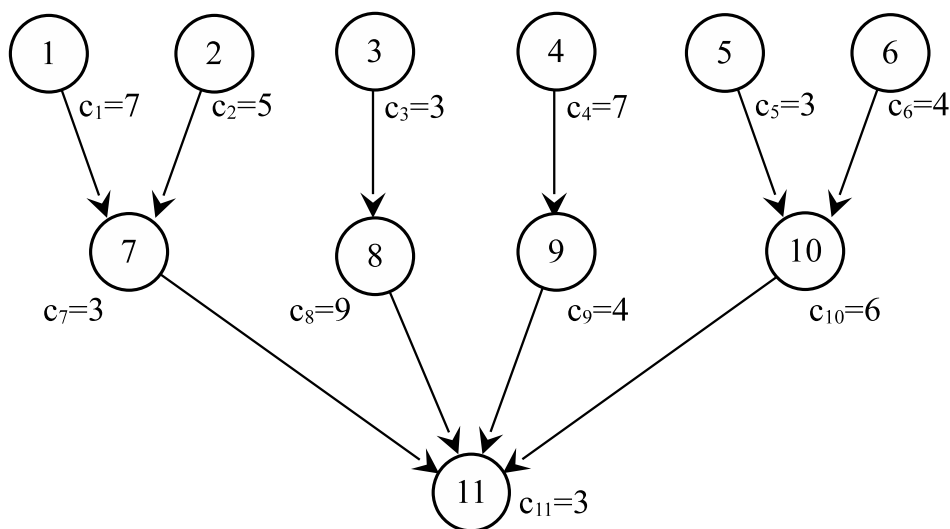
$$\mathcal{T}_2: T_2 = 10 \text{ ms}, \quad C_2 = 3 \text{ ms}$$

$$\mathcal{T}_3: T_3 = 15 \text{ ms}, \quad C_3 = 9 \text{ ms}$$

$$\mathcal{T}_4: T_4 = 20 \text{ ms}, \quad C_4 = 8 \text{ ms}$$

P1: 2 2 2 4 4 1 1 1 4 4 1 1 1 2 2 2 1 1 1 -
P1: 1 1 1 3 3 3 3 3 3 3 3 3 3 4 4 4 4 3 3 3 3
t: 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4

11. [4 boda] Za zadani sustav zavisnih zadataka pokazati raspoređivanje postupkom sa stabilnom strukturom na troprocesorskom računalu.



15 3 1
14 | 4 1
13 1 | 6 4/3
12 \ 8 | 5 5/3
11 / 2 | 10 6/3
10 / 10 6/3
9 5/3
8 5/3
7 5/3
5 4/3
5 4/3
4 4/3
3 11 1
2 | 1
1 | 1
0 * suma = 63/3 = 21