

Pisati čitko – nečitak odgovor ne donosi bodove.

1. [2 boda] Koji se kriterij najčešće koristi da bi se odredilo je li nešto pogodno za primjenu u sustavima za rad u stvarnom vremenu?

vrijeme reakcije na neki događaj iz okoline kojom se upravlja

komentar na rješenja:

često je odgovor bio općenito ``vremenska ispravnost`` ili slično, međutim on se ne koristi ``najčešće`` nego uvijek!

2. [2 boda] Programsko inženjerstvo bi trebalo primijenjivati u svim procesima izrade programske potpore. Međutim, to je posebno bitno ako se to radi za sustave za rad u stvarnom vremenu. Koji su sve razlozi za to?

posljedice grešaka su veće, a programskim inženjerstvom će se smanjiti broj grešaka u sustavu

3. [2 boda] Koja su ograničenja regulatora (PID, neizrazitih)? Zašto se oni ne koriste svugdje u upravljanju?

oni su za jednostavna upravljanja, nisu za složena

4. [2 boda] Opći kriterij provjere rasporedivosti sustava zadataka koji se raspoređuje prema mjeri ponavljanja koristi provjere u "točkama raspoređivanja" ( $D_i$ ). Što su to točke raspoređivanja kad se ispituje rasporedivost zadatka  $T_i$ ?

trenuci pojave prioritetnijih zadataka  $T_j$  gdje je  $j \leq i$

5. U kojim klasama raspoređivača su optimalni:

a) [1 bod] raspoređivanje mjerom ponavljanja (RMPA),

b) [1 bod] raspoređivanje prema krajnjim trenucima završetaka (EDF)?

RMPA: statička pridjela prioriteta na jednoproc. rač.

EDF: dinamički raspoređivači (i općenito) na jednoproc. rač.

6. U nekom proizvodnom procesu jedno mikroračunalo upravlja jednim dijelom procesa. Na početku treba zatvoriti odvod iz kotla preko ventila *IZLAZ* (postavljenom vrijednosti 0 u *IZLAZ*). Potom treba otvoriti dovod smjese u kotao otvaranjem ventila *ULAZ* (postavljenom vrijednosti 0 u *ULAZ*). Je li dovoljna količina smjese u kotlu treba provjeravati preko senzora *MAX* (1 kad je). Kad se to dogodi ulazni ventil treba zatvoriti te početi s procesom "kuhanja". Na kuhanje se utječe preko tri izvora koje treba periodički obavljati. Posao očitavanja odgovarajućih senzora te proračuna novih postavki za svaki od tih izvora izvodi se u funkcijama *obavi\_1()*, *obavi\_2()* i *obavi\_3()* koje treba pozvati svakih  $T_1$ ,  $T_2$ , odnosno  $T_3$ , respektivno. Nakon  $T_k$  jedinica vremena kuhanje se zaustavlja sa *zaustavi\_sve()*. Nakon još  $T_h$  vremena otvara se izlazni ventil. Kad se smjesa odvede iz kotla, što se očitava preko senzora *MIN*, izlazni ventil se zatvara te ulazni otvara – započinje novi proces kuhanja.

a) [2 boda] Modelirati sustav UML dijagramom stanja.

b) [3 boda] Modelirati sustav vremenskom Petrijevom mrežom.

c) [3 boda] Napisati pseudokod za navedeno mikroračunalo, uz pretpostavku da

funkcija *dohvati\_vrijeme()* postoji.

a) UML dijagram stanja

stanja idu uglavnom slijedno

```
S0      - ULAZ i IZLAZ zatvoreni
S1      - ULAZ otvoren
S2      - senzor MAX, zatvori ULAZ
S3      - ULAZ i IZLAZ zatvoreni, kreće postupak
S3-1    - obavi_1 -> S3
S3-2    - obavi_1 -> S3
S3-3    - obavi_1 -> S3
S4      - zaustavi sve
S5      - otvori izlaz
S6      - zatvori izlaz -> S0
```

iz S3 ide se u S3-1 i -2 i -3

b) Vremenska Petrijeva mreža

```

                                Tk      Th
S0->|->S1    ->|->S2->|->S3->|->S4->|->S5->|-> S6->|->S0
    |->SM->/      \\      ///<      |->Sm->/
    MAX                      MIN
                                T1
                                |->S31->|->S31o->|->S31
                                T1
                                |->S32->|->S32o->|->S32
                                T2
                                |->S33->|->S33o->|->S33
                                T3
```

c) pseudokod

```
ponavljaaj {
    IZLAZ = 0
    ULAZ = 0
    dok je MAX == 0 radi
        ništa
    ULAZ = 1
    t0 = dohvati_vrijeme()
    t1 = t0 + T1
    t2 = t0 + T2
    t3 = t0 + T3
    dok je dohvati_vrijeme() < t0 + Tk radi {
        ako je dohvati_vrijeme() >= t1 tada {
            obavi_1()
            t1 = t1 + T1
        }
        ako je dohvati_vrijeme() >= t2 tada {
            obavi_2()
```

```

        t2 = t2 + T2
    }
    ako je dohvati_vrijeme() >= t3 tada {
        obavi_3()
        t3 = t3 + T3
    }
}
dok je dohvati_vrijeme() < t0 + Tk + Th radi
    ništa
zaustavi_sve()
IZLAZ = 1
dok je MIN == 0 radi
    ništa
}

```

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. Zadan je sustav zadataka:

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

$$\mathcal{T}_2: T_2 = 15 \text{ ms}, C_2 = 6 \text{ ms}$$

$$\mathcal{T}_3: T_3 = 20 \text{ ms}, C_3 = 3 \text{ ms}$$

- a) [2 boda] Grafičkim postupkom provjeriti rasporedivost postupkom prema mjeri ponavljanja za jednoprosorski sustav te iz grafa odrediti implicitne trenutke krajnjeg dovršetka za sve zadatke.
- b) [3 boda] Grafički prikazati izvođenje sustava kada se koristi postupak prema krajnjem trenutku završetka na dvoprosorskom računalu do  $t = 20$  ms (red prispijeća te mjera ponavljanja kao dodatni kriteriji, ako je potrebno), uz dvostruko veće potrebe zadataka za procesorskim vremenom ( $C'_i = 2 \cdot C_i$ ).

a) RMPA grafički

1 1 1 2 2 2 2 2 2 3 1 1 1 3 3 2 2 2 2 2 1

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

3 gotov, rasporedivo

Implicitni deadline:

T1: 10

T2: 15

t3: 15

b) prema krajnjem trenutku završetka

2 2 2 2 2 2 2 2 2 2 2 - - - 2 2 2 2 2

1 1 1 1 1 1 3 3 3 3 3 3 1 1 1 1 1 - -

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

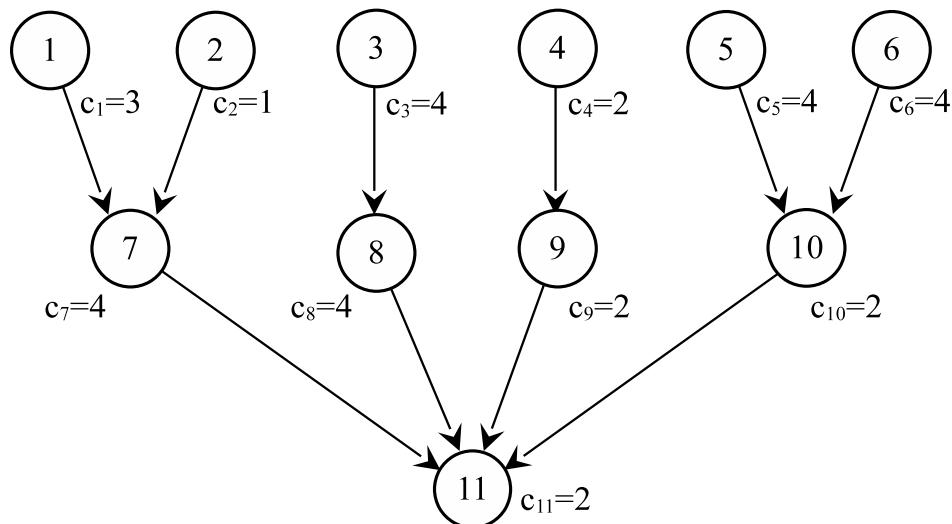
3 3

dodatni kriterij: red prispjeća

9. Za zadani sustav zavisnih zadataka pokazati:

a) [3 boda] opće raspoređivanje na troprocesorskom sustavu

b) [2 boda] raspoređivanje postupkom sa stablenom strukturom na troprocesorskom sustavu.



a) opće raspoređivanje

t1=0

grupiranje zadataka: TS\_1={1,2,7}; TS\_2={3,8}, TS\_3={4,9},  
TS\_4={5,6,10}

S1={1,2,7} CS1=8

S2={3,8} CS2=8

S3={4,9} CS3=4

$S4=\{5,6,10\}$   $CS3=10$

$CS1+CS2+CS3+CS4=30$

na troprocesorskom računalu to traje  $DT=10$

$A\_CS1 = CS1/DT = 8/10 = 4/5$

$A\_CS2 = CS2/DT = 8/10 = 4/5$

$A\_CS3 = CS3/DT = 4/10 = 2/5$

$A\_CS4 = CS4/DT = 10/10 = 1$

zadaci 1 i 2 dijele  $A\_CS1$ :

$A\_1 = A\_CS1 * C1/(C1+C2) = 4/5 * 3 / 4 = 3/5$

$A\_2 = A\_CS1 * C2/(C1+C2) = 1/5 * 1 / 4 = 1/5$

zadaci 3 i 4 koriste cijele  $A\_CS2$  i  $A\_CS3$

$A\_3 = A\_CS2 = 4/5$

$A\_4 = A\_CS3 = 2/5$

zadaci 5 i 6 dijele  $A\_CS4$  na pola jer traju jednako

$A\_4 = A\_CS4/2 = 1/2$

$A\_5 = A\_CS4/2 = 1/2$

zadatak 1 će biti gotov za:  $C1/A\_1 = 3/(3/5) = 5$

zadatak 2 će biti gotov za:  $C2/A\_2 = 1/(1/5) = 5$  (očekivano)

zadatak 3 će biti gotov za:  $C3/A\_3 = 4/(4/5) = 5$

zadatak 4 će biti gotov za:  $C4/A\_4 = 2/(2/5) = 5$

zadatak 5 će biti gotov za:  $C5/A\_5 = 4/(1/2) = 8$

zadatak 6 će biti gotov za:  $C6/A\_6 = 4/(1/2) = 8$  (očekivano)

$\min = 5$

$t2 = 5$ :

$Z1, Z2, Z3, Z4$  - gotovi

kreću:

$Z7$  sa  $A\_7 = A\_CS1 = 4/5$

$Z8$  sa  $A\_8 = A\_CS2 = 4/5$

$Z9$  sa  $A\_9 = A\_CS2 = 2/5$

zadaci 5 i 6 su do  $t=5$  izveli:

$c5' = 5*A\_5 = 5*1/2 = 2,5$ , ostaje još 1,5 (3 s potrebne)

$t = 8$ :

gotovi 5 i 6, počinje 10

zad. 7 od 5-8 (3 s) odradio:  $3*4/5 = 12/5 = 2 \frac{2}{5}$ , ostaje 1  $\frac{3}{5}$

zad. 8 isto

zad. 9  $3*2/5 = 6/5$  ostaje  $4/5$

$t = 10$ : svi gotovi, počinje 11 s  $A\_11 = 1$

$t = 12$ : 11 gotov, sustav gotov

b) stablena struktura

					3 proc	
10			3		1	(1 zadatak)
9	1				1	(2 zadatka)
8	\			5	4/3	(5 zadataka)
7	\	2		\	5/3	(5 zadataka)
6	7		8	4	5/3	(5 zadataka)
5	\				5/3	(5 zadataka)
4	\			9	4/3	(4 zadatka)
3	\				4/3	(4 zadatka)
2		----	11	----	1	(1 zadatak)
1					1	(1 zadatak)
0			-		ukupno:   13	