

Odgovarati kratko, jasno i čitko. Nečitki odgovori su "krivi" (ne donose bodove).

1. [2 boda] Što je najbitnije za SRSV-e koji svoje upravljanje temelje na obradi događaja?

što prije prihvatiti i obraditi događaj

2. [2 boda] Navesti osnovne razloge korištenja UML dijagrama: obrazaca uporabe, sekvencijskih, kolaboracijskih i dijagrama stanja (čemu služi svaki).

obrasci uporabe: identifikacija funkcija sustava  
sekvencijski: vremensko uređenje događaja u scenariju izvođenja  
kolaboracijski: identifikacija tko s kim komunicira  
dijagram stanja: stanja sustava i promjene stanja

3. [2 boda] Što su to regulacijski zadaci?

automatski upravljaju sustavom (prema zadanom algoritmu, npr. PID)

4. [2 boda] Kada se za neki sustav zadataka kaže da "u potpunosti iskorištava procesor" (u kontekstu raspoređivanja tih zadataka nekim postupkom)?

bilo kakvo povećanje potrebna procesorska vremena za bilo koji zadatak izazvati će da sustav postane nerasporediv (u kritičnom slučaju zadatak najmanja prioriteta se jedva stigne obaviti)

5. [2 boda] Opisati kriterije i načela raspoređivanja običnih (vremenski nekritičnih) dretvi prema višerazinskom raspoređivanju s povratnom vezom (engl. *multilevel feedback queue – MFQ*).

preferiraju se kratki poslovi, poslovi koji rade s UI napravama  
postupak raspoređivanja:  
- kad se stvori nova dretva, ona ide u ``najviši`` red (na kraj)  
- uvijek se uzima i izvodi prva dretva iz ``najvišeg`` reda  
- ako ta dretva nije gotova unutar kvanta vremena, miče se u prvi niži red (na kraj)  
- dretve u najnižem redu se poslužuju podjelom vremena

6. Neki sustav se početno nalazi u stanju A. Nakon 5 sekundi ili nakon primitka poruke od čvora X prelazi u stanje B. Ako je prijelaz A→B uzrokovan porukom, onda u stanju B ostaje 10 sekundi (ako nema novih poruka) te onda prelazi u stanje C. Ako je prijelaz A→B uzrokovan protokom 5 sekundi, onda ostaje u stanju B dok ne primi poruku od čvora Y. U stanjima B i C ignoriraju se poruke od čvora X, ali s porukama od čvora Y prelazi se u stanje D. U stanju D ostaje dok ne dođe nova poruka (bilo kakva). Kada dođe poruka, onda sustav prelazi u stanje A.

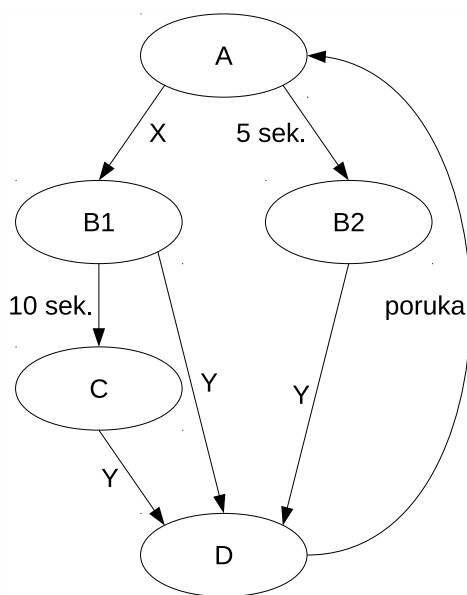
a) [1 bod] Opisati prikazani sustav UML dijagramom stanja.

b) [2 boda] Prikazati sustav UML sekvencijskim dijagramom za slučaj kada čvor X pošalje poruku u 3. i 20. sekundi (i nema ostalih poruka).

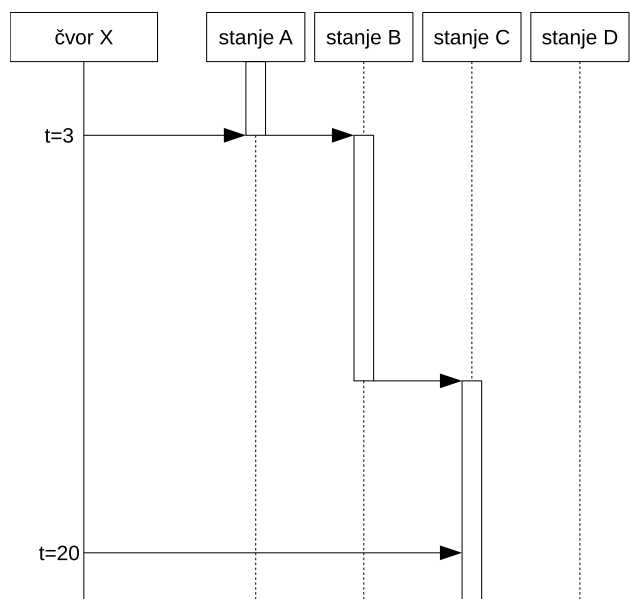
c) [2 boda] Prikazati sustav vremenskom Petrijevom mrežom.

Stanja A, B, C i D su stanja sustava – ne moraju UML dijagrami i Petrijeva mreža imati upravo takva stanja, ali se mora iz njihovih stanja jasno odrediti u kojem je sustav stanju.

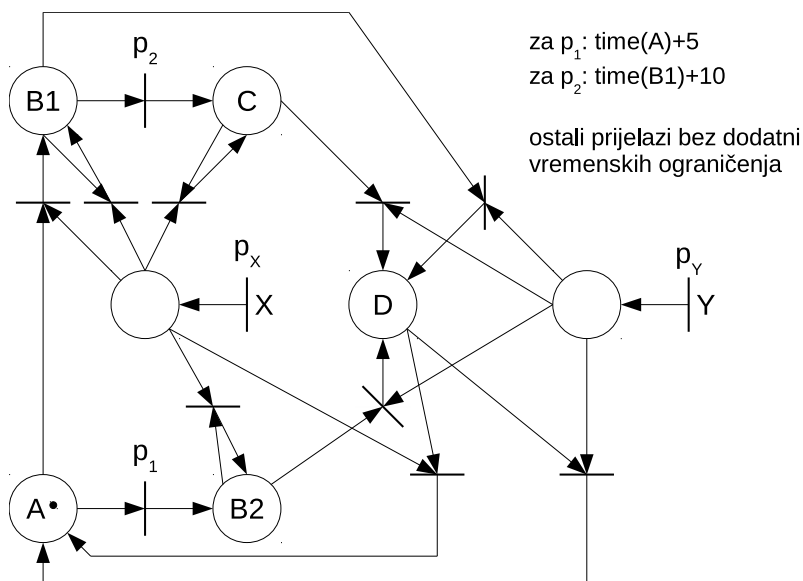
Primjer rješenja:



a) UML dijagram stanja



b) UML sekvencijski dijagram



c) Vremenska Petrijeva mreža

7. [3 boda] Neko računalo treba upravljati s nekoliko aktivnosti. Za aktivnost A treba jednom svakih 30 ms pozvati  $A()$ , za B treba jednom u 50 ms pozvati  $B()$ . Aktivnost C upravlja se u obradama prekida 27 funkcijom  $C()$  koja traje vrlo kratko. Za aktivnost D treba pratiti ulaz STANJE\_D i ako je ono postavljeno (nije nula) onda treba pozvati funkciju  $D()$  i to **ne kasnije od 5 ms nakon promjene** (obrada  $D()$  traje vrlo kratko). Obrade  $A()$  i  $B()$  mogu potrajati i do 10 ms, ali se mogu prekidati u obradi ( $C()$  i  $D()$  se nesmiju prekidati). Opisati ostvarenje upravljanja ako sustav posjeduje prekidni podsustav ( $registriraj\_prekid(irq, handler)$ ) i podsustav upravljanja vremenom ( $trenutno\_vrijeme()$ ,  $postavi\_alarm(perioda, funkcija)$ ). Pretpostaviti da alarmi prekidaju izvođenje upravljačke petlje, ali ne i obrade drugih alarma (već čekaju na njihov završetak).

```
C obrađivati iz prekida
D obrađivati preko alarma
A i B pozivati iz upravljačke petlje
(inače bi D mogao biti previše odgođen)

d() {
    zabrani_prekidanje()
    ako je ( STANJE_D != 0 ) tada
        D()
    dozvoli_prekidanje()
}
program {
    registriraj_prekid ( 27, C )
    postavi_alarm ( 5 ms, d ) //poželjno i kraće od 5 ms
    a = b = trenutno_vrijeme()
    ponavlja {
        ako je ( trenutno_vrijeme() >= a ) tada {
            a = a + 30 ms
            A()
        }
        ako je ( trenutno_vrijeme() >= b ) tada {
            b = b + 50 ms
            B()
        }
    }
}
```

8. [1 bod] Neki PID regulator sastoji se samo od P komponente s  $K_P = 0,5$ . Ako je stanje sustava u tom trenutku  $y_k = 10$ , željeno stanje  $y_P = 20$  koliko će biti  $r_k$  (vrijednost s kojom se utječe na sustav)?

$$e_k = y_P - y_k = 20 - 10 = 10$$
$$r_k = K_P \cdot e_k = 0,5 \cdot 10 = 5$$

9. [2 boda] U sustavu se neizrazitim zaključivanjem postoji pravilo:

$$p_2 : \text{ ako } (a \in A_1) \text{ i } (b \in B_2) \text{ tada } (c \in C_0)$$

Opisati primjenu tog pravila, ukoliko se koristi MIN\_MAX načelo.

$$\min(\{a \text{ na } A_1\}, \{b \text{ na } B_2\}) \Rightarrow \text{površina na } C_0 \text{ ulazi u sumu površina}$$



11. [2 boda] Sustav zadataka  $\mathcal{S}_2$  sličan je  $\mathcal{S}_1$  iz prethodnog zadatka, ali uz trajanja:  $C_1 = 6$  ms,  $C_2 = 6$  ms,  $C_3 = 10$  ms. Pokazati raspoređivanje prema krajnjim trenucima završetaka za  $\mathcal{S}_2$  na dvoprocesorskom sustavu. Sekundarni kriterij, ako bude potreban je mjera ponavljanja.

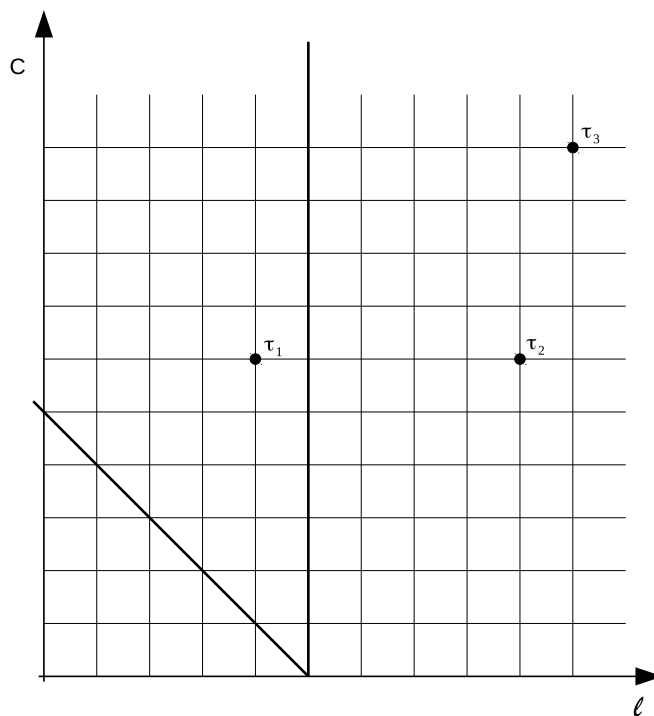
a)

```

P1 | 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 3 3 3 3 3
P2 | 2 2 2 2 2 2 - - - - 1 1 1 1 1 1 - - - - 1 1 1 1 1 1 -
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
0           5           10          15          20          25
1           1           2           1
2           3
3
kraj:      |           |
           1           1
           2           3
                    (tu se može stati)

```

12. [2 boda] Za sustav zadataka  $\mathcal{S}_2$  iz prethodnog zadatka izračunati zalihost računalne snage za prvih 5 sekundi (počevši u kritičnom slučaju, kad se poklope pojave sva tri zadatka) na dvoprocesorskom sustavu.



Iz gornjeg  $\ell/c$  grafa vidi se da treba 1 ms odvojiti za  $\tau_1$  te preostaje 9 ms (zalihost na dva procesora u prvih 5 ms)

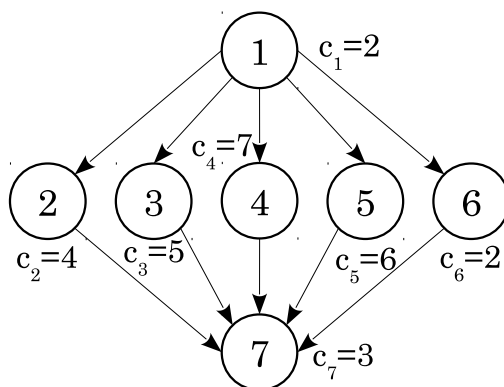
Primjer rasp. prema EDF+LLF, kada bi uzeli 9 ms u početku

```

P1 | x x x x 1 1 1 1 1 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1
P2 | x x x x x 2 2 2 2 2 2 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
1           1           2           1
2           -1          -1          3
3           -2          -3

```

13. [2 boda] Korištenjem općeg raspoređivanja napraviti raspored sustava zadataka koji je zadan necikličkim računalnim grafom prema slici ispod. Raspoređivanje napraviti za tri procesora. Prikazati mogući raspored zadataka po procesorima **korištenjem** rezultata općeg raspoređivanja.



**Sustav zadataka uz zad. 13.**

$t = 0 \Rightarrow$  samo  $\mathcal{T}_1$  te  $\alpha_1 = 1, \Delta t = c_1$

$$t = 2 \Rightarrow \mathcal{T}_2 - \mathcal{T}_6:$$

$$\Delta t = (c_2 + c_3 + c_4 + c_5 + c_6)/3 = (4 + 5 + 7 + 6 + 2)/3 = 24/3 = 8$$

$$\alpha_i = c_i / \Delta t$$

$$\alpha_2 = 4/8, \quad \alpha_3 = 5/8, \quad \alpha_4 = 7/8, \quad \alpha_5 = 6/8, \quad \alpha_6 = 2/8$$

$t = 10 \Rightarrow$  samo  $\mathcal{T}_7$  te  $\alpha_7 = 1, \Delta t = c_7$

t = 13 => sve gotovo

## Mogući raspored po procesorima

P1	1	1	2	2	2	2	3	3	3	3	7	7	7	
P2	-	-	3	4	4	4	4	4	4	4	-	-	-	
P3	-	-	5	5	5	5	5	5	6	6	-	-	-	
	+	-	+	-	+	-	+	-	+	-	+	-	+	+
	0					5					10			