

1. (5) U sustavu se nalaze ciklički zadaci dva proizvođača i jednog potrošača. Proizvođači generiraju poruke i stavljaju ih u prazna mjesta međuspremnik (ako je on pun, onda čekaju da se oslobodi jedno mjesto). Potrošač uzima dvije poruke iz međuspremnik te ih procesira (ako nema bar dvije poruke u međuspremniku onda čeka da se poruke pojave). Modelirati sustav *Petrijevom* mrežom. Veličina međuspremnik je 7 poruka.
2. (5) Zadan je sustav zadataka: $\tau_1: T_1=10, c_1=3$; $\tau_2: T_2=15, c_2=3$; $\tau_3: T_3=20, c_3=3$; $\tau_4: T_4=25, c_4=3$. Grafički provjeriti rasporedivost prema RMPA te odrediti „*implicitni deadline*“ za sve zadatke.
3. (5) Jedan se sustav sastoji od 20 poslova. Prosječno se svaki posao aktivira jednom u 100 ms. Raspoređivač se poziva iz prekidne funkcije kojom se signalizira pojava nekog zadatka (npr. *timer* je istekao te se odblokirao neki zadatak), kao i kada aktivni zadatak završi ili se blokira. Spremanje konteksta prekinute dretve neka traje 30 μ s (kao i obnova), dok trajanje prebacivanja dretvi u red pripravnih ovisi o raspoređivaču.

a) Ako se koristi RMPA raspoređivanje, onda su pripralni zadaci složeni po prioritetima, svaki u svojoj listi obzirom na prioritet (npr. opisnik zadatka s prioritetom 5 nalazi se u elementu polja *aktivni[5]*). Prebacivanje zadatka iz nekog neaktivnog (ili blokiranog) stanja u red pripravnih dretvi je stoga složenosti $O(1)$ te neka traje 10 μ s. Aktiviranje zadatka najvećeg prioriteta zbog korištenja dodatnog polja s bitovima prisutnosti zadataka u poljima je također neovisno o broju zadataka te neka traje dodatno 10 μ s.

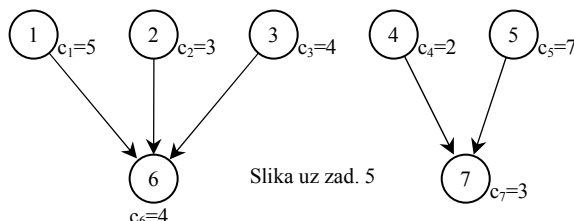
b) Ako se koristi DDS, pripralni su zadaci smješteni u jednu strukturu podataka sličnu listi, složenoj prema rastućim trenucima krajnjeg završetka zadataka (prvi je onaj s najskorijim *deadlineom* – d_i). Za zadatak „i“ različita aktiviranja imaju različiti d_i , tj. pri svakom prebacivanju zadatka „i“ u red pripravnih, potrebno je izračunati d_i , za što se potroši 20 μ s. Samo prebacivanje opisnika zadatka iz nekog neaktivnog (ili blokiranog) stanja u red pripravnih dretvi, zbog same strukture podataka je složenosti $O(\log n)$, gdje je n broj pripravnih zadataka (pretpostaviti da je prosječno $n=5$) te neka traje $40 \cdot \log(n)$ μ s. Aktiviranje zadatka najvećeg prioriteta neka traje dodatno 2 μ s.

Potrebno je odrediti opterećenje (u postocima) koje generira raspoređivač (zajedno s ostalim kućanskim poslovima) za a) i b) (*overhead* raspoređivanja).

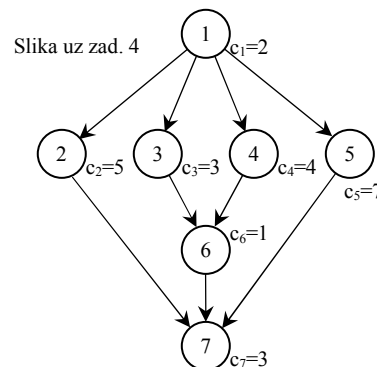
4. a) (3) Korištenjem općeg raspoređivanja (GS) napraviti raspored sustava zadataka (desno) koji je zadan necikličkim računalnim grafom. Raspoređivanje napraviti za dva procesora.

b) (2) Prikazati mogući raspored zadataka po procesorima korištenjem rezultata općeg raspoređivanja (napraviti PS - *preemptive scheduling*).

5. a) (3) Sustav zadataka S_4 zadan je usmjerenim grafom (slika ispod). Korištenjem postupka raspoređivanja sa stablenom strukturom (*rooted computation tree*) izgraditi računalno stablo. Na stablu naznačiti visine (h).



Slika uz zad. 5



Slika uz zad. 4

b) (2) Izračunati trajanje izračunavanja cijelog sustava zadataka raspoređenog ovom metodom za slučajeve kada u sustavu postoje tri procesora.

6. (4) Sustav zadataka u trenutku $t=0$ prikazan je ℓ/c grafom (desno). Grafički prikazati rad EDF te LLF metoda (do kraja svih zadataka ili do prekoračenja *deadline-a*), ako u sustavu postoje dva procesora.

(1) Izračunati zalihost procesorske snage u prve 4 jedinice vremena.

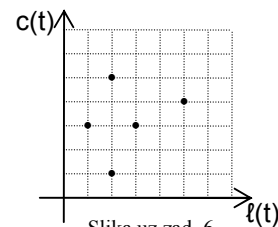
7. a) (1) Ako bi se CAN protokol pokušao ostvariti preko ethernet preklopnika (switcha), kako treba izgledati jedan takav ethernet paket koji prenosi CAN paket (izvorna i odredišna ethernet adresa)?

b) (1) Jednostavni način sinkronizacije sata klijenta sa satom poslužitelja (preko mreže) obavlja se formulom za računanje razlike u satovima: $c = (t_2 + t_3)/2 - (t_1 + t_4)/2$ (t_1 – vrijeme slanja zahtjeva (klijentsko vrijeme), t_2 – vrijeme primanja zahtjeva (poslužiteljsko vrijeme), t_3 – vrijeme slanja odgovora (poslužiteljsko vrijeme), t_4 – vrijeme primanja odgovora (klijentsko vrijeme)). Uz koje je pretpostavke izračun dobar?

c) (1) U sustavu koji koristi RTP/RTCP za komunikaciju, za jednu stanicu je izmjereno da prosječno šalje 1 paket veličine 1024 B svakih 100 ms. Što se iz tog podatka može zaključiti o RTP-u?

d) (1) Jedan zadatak treba svakih 100 ms nešto obaviti. Je li slijedeći kôd dobar za to? Obrazložiti.
ponavljaj { nešto_obavi; spavaj(100ms); }

e) (1) Dva zadatka na različitim računalima u istoj lokalnoj gigabitnoj mreži razmjenjuju podatke veličine 512 B. Koliko minimalno traje prijenos (od trenutka kada jedan zadatak pozove *pošalji*, a drugi primi podatak sa *primi*)? Koriste se 48-bitovne MAC adrese, te IPv4 (ostale podatke u zaglavlju zanemariti). Što još može utjecati da to vrijeme bude znatno veće?

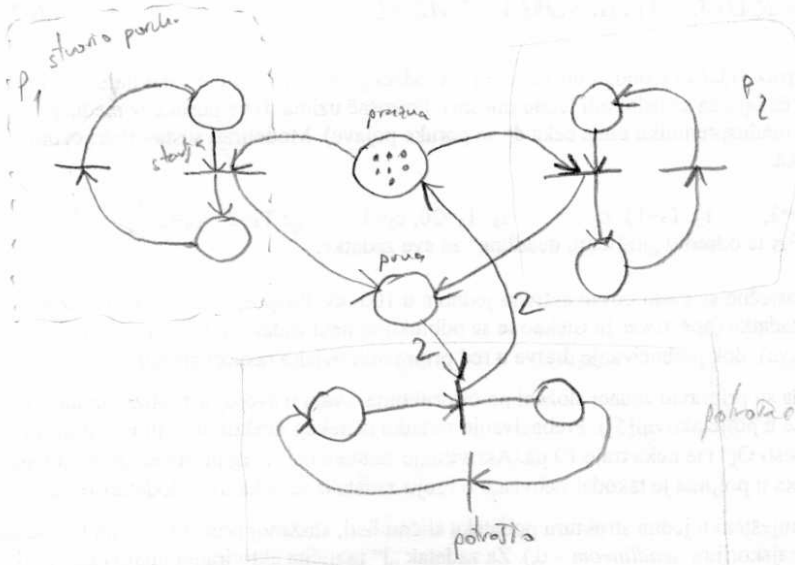


Slika uz zad. 6

Rješenja („jedna od mogućih točnih“):

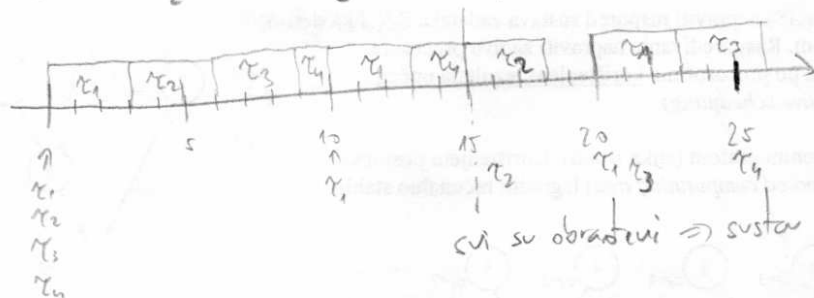
21. SZRSU 23. 6.

①



②

$$\begin{matrix} \tau_1 = 10 & \tau_2 = 15 & \tau_3 = 20 & \tau_4 = 25 \\ c_1 = 3 & c_2 = 3 & c_3 = 3 & c_4 = 3 \end{matrix}$$



implicitni deadline-i:

$$\begin{aligned} d_1 &= 10 \\ d_2 &= 15 \\ d_3 &= 20 \\ d_4 &= 20 \end{aligned}$$

iza 20-te jedinice vremena više nema mjesta za zadatak 4

③ $m = 20$ poruka
 $T = 100 \text{ ms}$
 $t_k = 30 \mu\text{s}$

a) $t_p = 10 \mu\text{s}$
 $t_a = 10 \mu\text{s}$

$$t = m(t_p + t_a + t_k \cdot 2)$$

$$= 20(10 + 10 + 30 \cdot 2)$$

$$= 1,6 \text{ ms}$$

$$\frac{t_{100}}{T} = 1,6 \%$$

b) $t_b = 20 \mu\text{s}$
 $t_p = 40 \cdot \log_2(4) \mu\text{s}$
 $m = 5$
 $t_a = 2 \mu\text{s}$

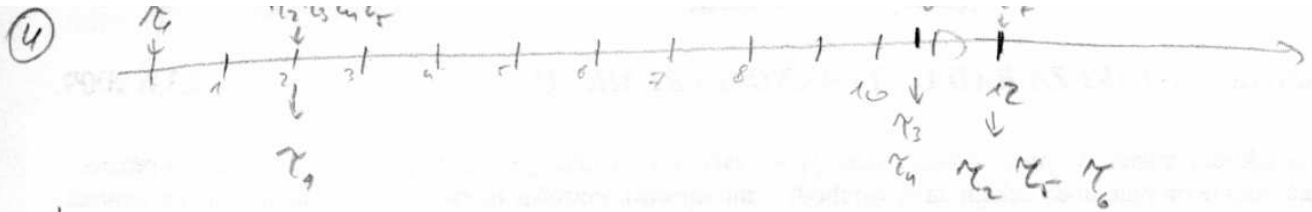
$$t_{100} = (t_b + t_p + t_a + t_k \cdot 2) \cdot m$$

$$= (20 + 40 \log_2 5 + 2 + 30 \cdot 2) \cdot 20$$

$$= 2,2 \text{ ms}$$

$$\frac{t_{100}}{T} = 2,2 \%$$

SVE PUTA DVA !
(početak i kraj)



$$t=0$$

$$\tau_1$$

$$t=2$$

$$\tau_2 \tau_3 \tau_4 \tau_5$$

$$\tau_2 c_2 = 5$$

$$\tau_6 c_6 = c_3 + c_4 + c_5 = 8$$

$$\tau_5 c_5 = 7$$

$$\frac{5+8+7}{2} = 10$$

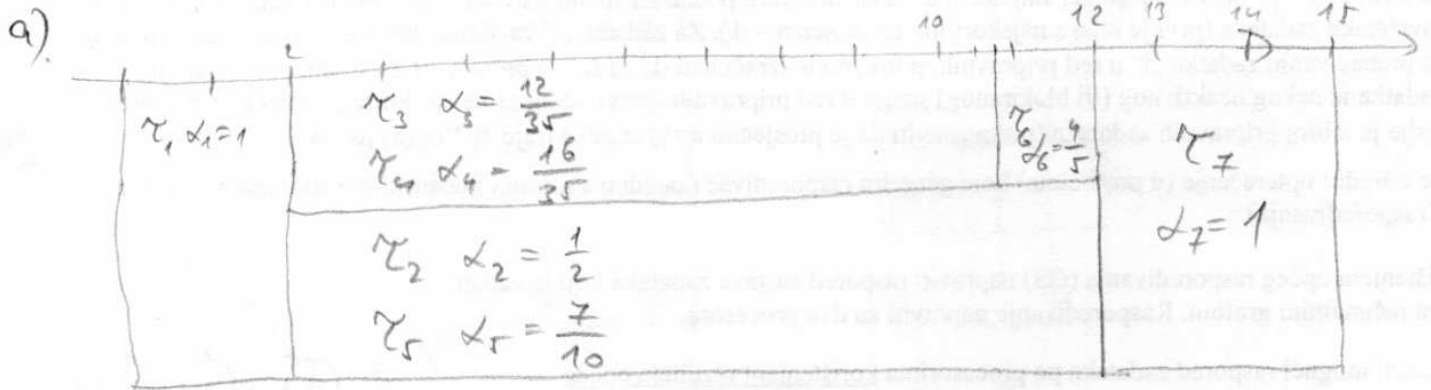
$$\alpha_2 = \frac{5}{10} = \frac{1}{2}$$

$$\alpha_4 = \frac{8}{10}$$

$$\alpha_5 = \frac{7}{10}$$

$$\sigma = \frac{8}{10} \Rightarrow \alpha_3 = \frac{3}{7} \cdot \frac{8}{10} = \frac{12}{35} \Rightarrow \Delta t_3 = \frac{c_3}{\alpha_3} = \frac{3}{\frac{12}{35}} = \frac{35}{4} = 8 \frac{3}{4}$$

$$\alpha_4 = \frac{4}{7} \cdot \frac{4}{5} = \frac{16}{35}$$



b)

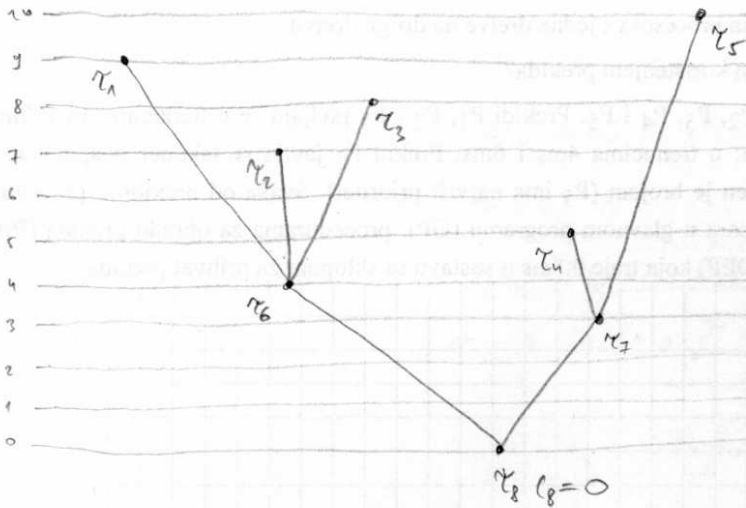
$\tau_1 c_1 = 2$	$\tau_3 c_3 = 3$	$\tau_4 c_4 = 1$	$\tau_2 c_2 = 5$	$\tau_6 c_6 = 1$	$\tau_7 c_7 = 3$
	$\tau_5 c_5 = 4$	$\tau_5 c_5 = 6$			

5)

3)

a)

b)



- 1
- 1
- 4
- 4/3
- 4/3
- 5/3
- 1
- 1
- 1
- 1

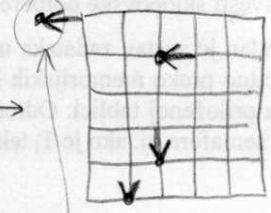
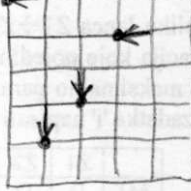
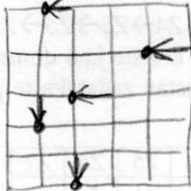
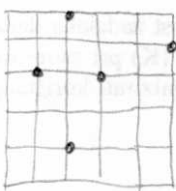
3 procesora

$$\Sigma = 7 + \frac{13}{3} = 11 \frac{1}{3}$$

6.

a)

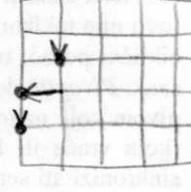
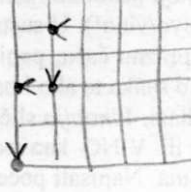
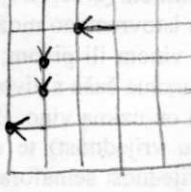
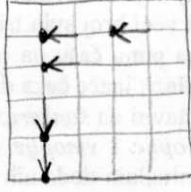
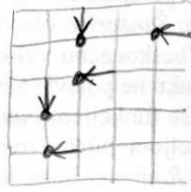
EDF



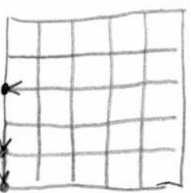
ovaj se više ne može rasporediti unutar deadline-a

LLF

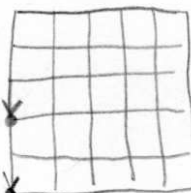
11



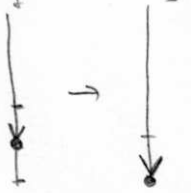
→



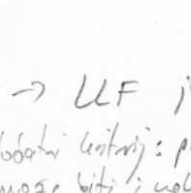
→



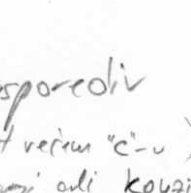
→



→



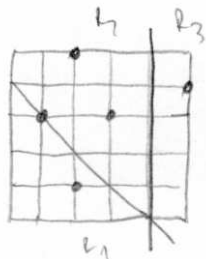
→



→ LLF je rasporediv

(dobro je unutra: prednost većim "c"-u)
može biti i malo druga, ali konstantno

b)



$$F(x, t) = ux - \sum_{x \in l_1} c_i(t) - \sum_{x \in l_3} (x - l_2(t))$$

$$F(4, t) = 2 \cdot 4 - (1 + 3) - [(4 - 3) + (4 - 2)] = 8 - 4 - 3 = 1$$

- ⑦ a) određena adresa $0x\text{ffff}\text{ffff}\text{ffff}$, izvor nije bitan
 b) trajanje prijenosa paketa u oboj smjeru isto (simetrična mreža)
 ! c) RTP troši barem $\frac{0,95}{0,05}$ puta više bandwidtha = $\frac{0,95}{0,05} \cdot 1024 = 19 \text{ kB}$ svakih 100 ms
 d) nije, period da širi uoči od 100 ms
 e) $v = 512 \text{ B}$

dh_A	dh_B	IP_A	IP_B	paket=512
--------	--------	--------	--------	-----------

$$v_p = 6 + 6 + 4 + 4 + 512 = 532 \text{ B} = 532 \cdot 8 \text{ bits}$$

$$r = 1 \text{ Gbit/s} = 2^{30} \text{ bit/s}$$

$$t = \frac{v_p}{r} = \frac{532 \cdot 8}{2^{30}} = 3,96 \cdot 10^{-6} \text{ s} = 3,96 \mu\text{s}$$

→ dodatna kašnjenja - obrada paketa u uređajnom području

Ili digitalno:

a)
 Dovoljno je za određenu ethernet (MAC) adresu koristiti broadcast, tada će svi primiti, a sam sadržaj (korisni) ethernet paketa može biti isti kao i na CAN mreži.

Naravno, ovakva bi realizacija bila inferiornija pravom CANu (zbog npr. nedeterminizma koji unosi preklopnik, poruke ne moraju biti isporučene vremenski istim redoslijedom kojim su i poslane)

b)
 Mreža treba biti simetrična: trajanje prijenosa u jednom i drugom smjeru treba biti jednako.

c)
 Gore navedeno rješenje pretpostavlja malo drukčije zadan zadatak, tj. umjesto „šalje 1 paket“, autor je mislio „šalje 1 RTP paket“, ali je greška nekako prošla neopaženo (do početka ispravljanja).

Uz zadano (bez „1 RTP paket“) može se zaključiti da je brzina od 10 kB/s mala, eventualno se prenosi samo audio (80 kbita/s), ili se radi o nekoj simulaciji ili drugim RT primjenama ovog protokola, ili stanica ne proizvodi RTP pakete već ih samo prima i povremeno šalje RTP pakete. Ako pretpostavimo ovo zadnje, možemo i izračunati minimalni „bandwidth“ RTP-a koje stanica prima, prema gore izračunatih 19 kB po 100 ms, tj. 190 kB/s (1520 kbita/s što možda sugerira da se ipak radi o nečem drugom, ne videokonferenciji nego videu veće kvalitete ili ipak nešto od prva dva prijedloga).

d)
 Problem onakve realizacije nije samo u „sleep(100ms)“ i trajanju izračunavanja (za koje se i u idealnoj situaciji povećava period). Problem je u nedeterminizmu cijelog sustava. Bilo kakav prekid u izvođenju programa (zbog npr. obrade nekog drugog zadatka većeg prioriteta) dodatno odgađa iduću pojavu programa.

e)
 Uz pretpostavku da su računala izravno spojena na isti gigabitni preklopnik, gore izračunatu brojku treba pomnožiti sa dva (od prvog računala do preklopnika + od preklopnika do drugog računala). Ako se između računala fizički nalazi više preklopnika, onda se gore izračunata brojka množi s brojem segmenata koje podaci moraju prijeći, tj. sa $(N+1)$, gdje je N broj preklopnika.

Osim ovih „najočitijih točnih rješenja“, studentima su se priznavala i druga, relativno logična (barem za 50% bodova).