

1. (5) U sustavu se nalaze ciklički zadaci dva proizvođača i jednog potrošača. Proizvođači generiraju poruke i stavljaju ih u prazna mjesta međuspremnika (ako je on pun, onda čekaju da se oslobodi jedno mjesto). Potrošač uzima dvije poruke iz međuspremnika te ih procesira (ako nema bar dvije poruke u međuspremniku onda čeka da se poruke pojave). Modelirati sustav *Petrijevom* mrežom. Veličina međuspremnika 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  $\mu$ 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  $\mu$ 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  $\mu$ s.

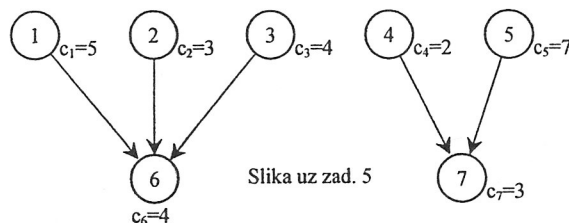
b) Ako se koristi DDS, pripralni su zadaci smješteni u jednu strukturu podataka sličnu listi, složenoj prema rastućim trenucima krajnjeg završetka zadataka (prvi je onaj s najskorijim *deadlineom* –  $d_i$ ). Za zadatak „i“ različita aktiviranja imaju različiti  $d_i$ , tj. pri svakom prebacivanju zadatka „i“ u red pripravnih, potrebno je izračunati  $d_i$ , za što se potroši 20  $\mu$ 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  $40 \cdot \log(n)$   $\mu$ s. Aktiviranje zadatka najvećeg prioriteta neka traje dodatno 2  $\mu$ s.

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

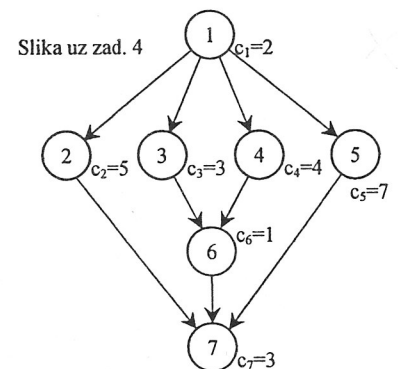
4. a) (3) Korištenjem općeg raspoređivanja (GS) napraviti raspored sustava zadataka (desno) koji je zadan necikličkim računalnim grafom. Raspoređivanje napraviti za dva procesora.

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  $\ell/c$  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?

