

1. [2 boda] Lift u nekoj zgradi ima dvije brzine kretanja: normalnu, kada prevozi putnike te brzu kada je prazan. Pri pokretanju lifta, on se naprije ubrzava na potrebnu brzinu kretanja te potom nastavlja konstantnom brzinom skoro do odredišna kata, kada započinje sa smanjivanjem brzine do zaustavljanja. Po zaustavljanju započinje s otvaranjem vrata, pa neko vrijeme stoji s otvorenim vratima te ih konačno zatvara. Ako nema novih zahtjeva lift stoji. Za opisani lift napraviti UML dijagram stanja te sekvencijski dijagram za scenarij u kojem je lift početno na 5. katu, putnik ga pozove s 2. kata za gore i kada lift dođe putnik ide na 6. kat.
2. [2 boda] Neki kontroler treba na temelju ulaznih vrijednosti a i b izračunati izlaznu i . Poznato je da je izlaz (otprilike) obrnuto proporcionalan ulazu a te (otprilike) proporcionalan ulazu b . Za najmanju vrijednost od $a = A_{MIN}$ i najveću vrijednost od $b = B_{MAX}$ izlaz treba biti $i = I_{MIN}$ i obratno, za $a = A_{MAX}$ i $b = B_{MIN}$ izlaz treba biti $i = I_{MAX}$. Ostvariti kontroler korištenjem neizrazitog upravljanja.
3. [2 boda] Korištenjem raspoređivanja prema mjeri ponavljanja (*rate monotonic scheduling*) provjeriti rasporedivost sustava zadataka na jednoprocorskom sustavu korištenjem:
 - a) procesorske iskoristivosti (nužni uvjet, lub)
 - b) simulacijom (grafičkim postupkom).

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

$$\mathcal{T}_2 : T_2 = 15 \text{ ms}, C_2 = 3 \text{ ms}$$

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

$$\mathcal{T}_4 : T_4 = 30 \text{ ms}, C_4 = 6 \text{ ms}$$

4. [2 boda] Zadaci $\mathcal{T}_1 - \mathcal{T}_4$ javljaju se u trenucima 1. ms, 2. ms, 3. ms te 4. ms, respektivno. Obrada svakog zadatka traje po 3 ms. Svi zadaci (i pojedinačno) moraju biti gotovi do 7. ms. Pokazati rad raspoređivača po najmanjoj labavosti (LLF) nad tim sustavom zadataka ako se koristi dvoprocorski sustav. Naznačiti posebne trenutke u tom izvođenju (dolasci, odlasci, ...).
5. [2 boda] U nekom sustavu javljaju se dretve A, B, C i D. Dretva A ima najveći prioritet, slijede dretve B i C koje imaju jednaki prioritet, te dretva D koja ima najmanji prioritet. Dretva A se javlja u $t=4$. [s], B u $t=2$. [s], C u $t=5$. [s] te D u $t=1$. [s]. Svaka dretva treba 5 [s] procesorskog vremena. Sustav koristi raspoređivanje SCHED_RR. Prikazati rad sustava.
6. [2 boda] Što se sve može dogoditi s dretvom pri izvođenju koda (očekivano i neočekivano)?

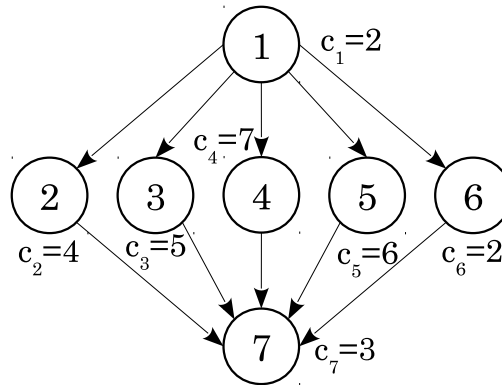
```
struct timespec t;  
t.tv_sec = 0; t.tv_nsec = 100000; /* 100 mikrosekundi */  
clock_nanosleep ( CLOCK_REALTIME, 0, &t, NULL ); /* odgodi za t */
```

7. [2 boda] U nekom jednoprocorskom sustavu izvode se dretve A, B, C i D. Dretva A ima najveći prioritet, slijedi dretva B, pa C te D koja ima najmanji. Dretve koriste tri sredstva koja su zaštićena semaforima S_1 , S_2 i S_3 . Sve dretve mogu trebati bilo koje sredstvo u svom izvođenju. Ponekad, dretva koja već ima jedno sredstvo može tražiti i drugo, ali ne i treće (prije traženja trećeg otpušta sva zauzeta). Pretpostavka je da se potpuni zastoj neće dogoditi. Ako sustav koristi raspoređivanje prema prioritetu te protokol nasljeđivanja prioriteta, koliko se najviše (u najgorem slučaju) može zaustaviti dretva A zbog inverzije prioriteta? Opisati scenarij u kojem se to događa.

8. [2 boda] Navesti nedostatke navedena koda (u raznim "okruženjima") i mogućnosti za njihovo rješavanje.

```
int op ( ... ) {  
    static int brojilo = 0;  
    ... /* neka operacija, bez korištenja varijable brojilo */  
    brojilo = brojilo + 1;  
    return brojilo;  
}
```

9. [2 boda] Korištenjem općeg raspoređivanja napraviti raspored sustava zadataka koji je zadan ne-cikličkim računalnim grafom prema slici ispod. Raspoređivanje napraviti za tri procesora. Prikazati mogući raspored zadataka po procesorima korištenjem rezultata općeg raspoređivanja.



Sustav zadataka uz zad. 9.

10. [2 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 obrada 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).