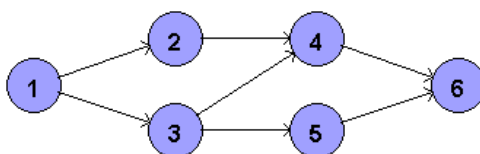


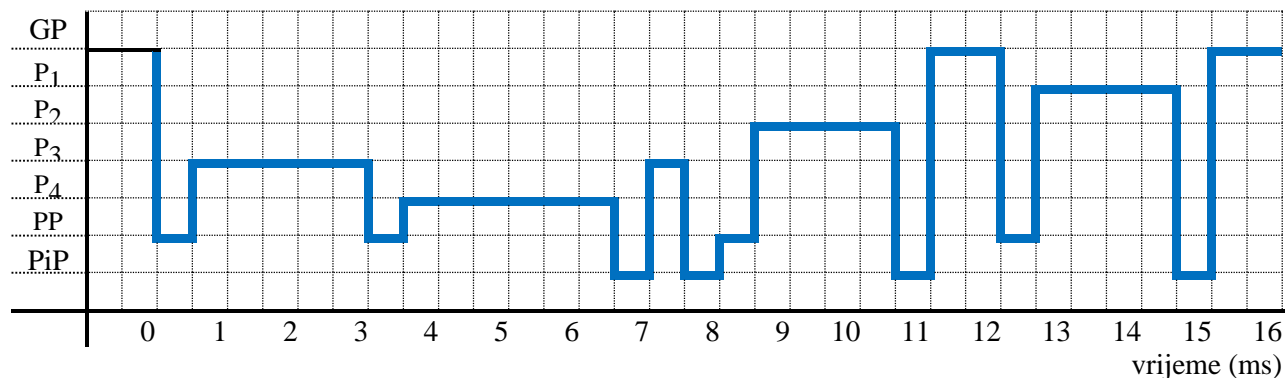
1. Napisati funkciju u C-u `unsigned int f(unsigned int ulaz)` koja će najprije postaviti svaki treći bit ulazne varijable u jedan, a potom obrisati svaki 5 bit (postaviti ga u nulu). Broj bitova ulazne varijable dobiva se sa `8*sizeof(unsigned int)`.

```
unsigned int f(unsigned int ulaz) {
    int i;
    unsigned int izlaz = ulaz
    for (i = 2; i < 8*sizeof(unsigned int); i++)
        izlaz = izlaz | (1 << i);
    for (i = 4; i < 8*sizeof(unsigned int); i++)
        izlaz = izlaz & ~(1 << i);
    return izlaz;
}
```

2. Neki posao podijeljen je u šest zadataka Z_1 do Z_6 . Poznato je da će rezultat biti ispravan ako se zadaci izvode slijedno: $Z_1 > Z_2 > Z_3 > Z_4 > Z_5 > Z_6$. Domene i kodomene zadataka su sljedeće: $Z_1:\{D_1=M_1; K_1=M_2\}$, $Z_2:\{D_2=M_2; K_2=M_3\}$, $Z_3:\{D_3=M_4; K_3=M_1\}$, $Z_4:\{D_4=M_3; K_4=M_4\}$, $Z_5:\{K_5=M_1\}$, $Z_6:\{D_6=M_1; K_6=M_4\}$. Odrediti maksimalno paralelni sustav zadataka (i prikazati ga usmjerenim grafom) uzimajući u obzir njihove domene i kodomene te međusobni odnos u lancu.



3. U nekom sustavu sa sklopom za prihvat prekida javljaju se prekidi P_3 u trenutku $t=0$ ms, P_2 u $t=1$ ms, prekid P_4 u $t=3$ ms te prekid P_1 u $t=12$ ms. Prioritet prekida određen je brojem (P_4 ima najveći prioritet). Grafički prikazati aktivnosti procesora u glavnom programu (GP), procedurama za obradu prekida (P_i) koje traju: 2 ms za P_1 i P_2 , 3 ms za P_3 i P_4 te procedurama za prihvat prekida (PP) koje traju 0,5 ms i povratak iz prekida (P_iP) koje traju 0,5 ms.



4. Stanje nekog sustava u promatranom trenutku je sljedeće: dretva D_3 je aktivna, D_1 je u redu pripravnih, D_4 u redu semafora $OSEM[1]$, te D_2 i D_5 u redu odgođenih: D_2 treba čekati još jedan otkucaj sata, a D_5 3 otkucaja (ukupno). Red pripravnih dretvi je uređen prema prioritetu. Prioritet dretve određen je njenim indeksom – dretve s većim indeksom imaju veći prioritet (D_5 ima najveći prioritet). U tom stanju dogodi se sljedeći niz poziva jezgrinih funkcija (svaki idući se događa neko kratko vrijeme nakon što je prethodni bio gotov): 1. Otkucaj_sata() 2. Započni_UI(1) 3. Postavi_OSEM(1) 4. Odgodi(5). Prikazati stanje sustava nakon svakog od tih poziva.

	poč. stanje	nakon 1.	nakon 2.	nakon 3.	nakon 4.
Aktivna_D	3	3	2	4	2
Pripravne_D	1	2 1	1	2 1	1
OSEM[1]	4	4	4	--	--
Odgodene_D	$2^1 5^2$	5^2	5^2	5^2	$5^2 4^3$
UI[1]	--	--	3	3	3

5. Neki sustav upravljanja treba ostvariti korištenjem tri dretve dohvati te jednom dretvom obradi. Svaka dretva dohvati provjerava svoj dio sustava funkcijom provjera(id), gdje id predstavlja identifikacijski broj dretve koja tu funkciju poziva (1, 2 ili 3). Kada funkcija provjera(id) vrati nulu tada taj dio sustava ne treba dodatnu pažnju te dotična dretva dohvati spava 5 sekundi prije ponovne provjere. U protivnom, kada je povratna vrijednost veća od nule, dretvi dohvati ta vrijednost predstavlja prioritet otkrivenog događaja te to treba dojaviti dretvi obradi. Dretva dohvati potom čeka potvrdu da je njen događaj obrađen prije nego li ponavlja provjeru. Dretva obradi čeka da joj netko dojavi potrebu za obradom. Kad joj se dojavi ona među dojavljenim događajima odabire događaj najvećeg prioriteta te ga obrađuje funkcijom obradi_događaj(prio). Po završetku obrade jednog događaja, dretva obradi ide na sljedeći ako takav postoji ili čeka na dojavu novog. Napisati pseudokod dretvi korištenjem monitora za sinkronizaciju. Akcijama koje su nezavisne dopustiti paralelni rad (provjera i obradi_događaj).

Primjer kostura rješenja uz zajedničku varijablu događaj[3] (ne mora se koristiti, a ako se koristi, onda tri točkice (...) se mogu zamijeniti s odgovarajućim kodom). Navedi korištene varijable, monitor i redove uvjeta.

```
dretva dohvati(id) {
    ponavljaaj {
        prio = provjera(id)
        ako je prio == 0 tada {
            spavaj(5)
        }
        inače {
            ...
            događaj[id] = prio
            ...
        }
    }
}

dretva obradi() {
    ponavljaaj {
        ...
        id = 1
        ako je događaj[2] > događaj[id] tada { id = 2 }
        ako je događaj[3] > događaj[id] tada { id = 3 }
        ako je događaj[id] > 0 tada {
            ...
            obradi_događaj(događaj[id])
            ...
            događaj[id] = 0
            ...
        }
        inače {
            ...
        }
        ...
    }
}
```

```
dretva dohvati(id) {
    ponavljaaj {
        prio = provjera(id)
        ako je prio == 0 tada {
            spavaj(5)
        }
        inače {
            Udi_u_monitor(m)
            događaj[id] = prio
            Oslobodi_iz_reda_uvjeta(novo)
            Čekaj_u_redu_uvjeta(potvrda[id])
            Izadi_iz_monitora(m)
        }
    }
}
```

```
dretva obradi() {
    ponavljaaj {
        Udi_u_monitor(m)
        id = 1
        ako je događaj[2] > događaj[id] onda
            id = 2
        ako je događaj[3] > događaj[id] onda
            id = 3
        ako je događaj[id] > 0 tada {
            Izadi_iz_monitora(m)
            obradi_događaj(događaj[id])
            Udi_u_monitor(m)
            događaj[id] = 0
            Oslobodi_iz_reda_uvjeta(potvrda[id])
        }
        inače {
            Čekaj_u_redu_uvjeta(novo)
        }
        Izadi_iz_monitora(m)
    }
}
```

6. U nekom sustavu poslužitelj radi s 75% opterećenjem te je prosječno zadržavanje poslova u sustavu dvije sekunde. Kada bi 20% dolaznih poslova preusmjerili na neki drugi poslužitelj, koliko bi nakon toga bilo prosječno trajanje zadržavanje poslova u ovom sustavu? Pretpostaviti Poissonovu razdiobu dolazaka i eksponencijalnu razdiobu za trajanja obrade.

$\rho_1 = 0,75$ $T_1 = 2 \text{ s}$ $\beta_2 = \beta_1$ $\alpha_2 = 0,8 * \alpha_1$ $T_2 = ?$ $T = 1 / (\beta - \alpha)$ trebaju nam β_2 i α_2 a za njih β_1 i α_1	$\rho = \alpha / \beta \Rightarrow \alpha = \beta * \rho$ $T = 1 / (\beta - \alpha) = 1 / (\beta - \beta * \rho) = 1 / (\beta * (1 - \rho))$ $\Rightarrow \beta = 1 / (T * (1 - \rho))$ $\beta_1 = 1 / (T_1 * (1 - \rho_1)) = 1 / (2 * (1 - 0,75)) = 1/0,5 = 2 \text{ s}^{-1}$ $\alpha_1 = 2 * 0,75 = 1,5 \text{ s}^{-1}$ $\alpha_2 = \alpha_1 * 0,8 = 1,2 \text{ s}^{-1}$ $T_2 = 1 / (\beta_2 - \alpha_2) = 1 / (2 - 1,2) = 1/0,8 = \mathbf{1,25 \text{ s}}$
---	---

7. Neki sustav koristi raspoređivanje prema prioritetu kao osnovnom kriteriju te prema podjeli vremena kao dodatnom koji se koristi kad prvi kriterij ne daje samo jednu dretvu (opisan kao SCHED_RR). U nekom trenutku t_0 u sustavu se nalaze dretve A, B, C, D, E. Dretva A ima najveći prioritet 5, slijede dretve B i D s prioritetom 4 te dretve C i E s prioritetom 3. Dretva A treba još 30 ms za završetak, dretva B 50 ms, dretva C 20 ms te dretve D i E po 30 ms. Prikazati stanje procesora od trenutka t_0 pa dok sve dretve ne završe radom ako je kvant vremena $T_q = 10 \text{ ms}$ (kada se koristi).

pripravne dretve (nije obavezno)	E	E	E																
	C	C	C	E	E	E	E	E	E										
	D	D	D	C	C	C	C	C	C	E	E								
	B	B	B	D	B	D	B	D	B	D	C	C	E	C	E				
aktivna	A	A	A	B	D	B	D	B	D	B	B	C	E	C	E	E			
				A						D	B				E	E			

8. U sustavu koji raspolaže s 20 okvira za programe pojavljuju se zahtjevi za pokretanje procesa i to redom: P1 koji ukupno treba 10 stranica, P2 koji treba 12 stranica, P3 koji treba 6 stranica te P4 koji treba 6 stranica. Svi su procesi već pripremljeni na pomoćnom spremniku (disku). Zahtjevi se poslužuju na način da se procesu daje najviše okvira koliko se može, ali ne manje od 6 ili više od 10 (niti jedan proces neće u memoriji imati manje od 6 i više od 10 stranica). Ako nema dovoljno slobodnih okvira, onda se okviri oslobađaju tako da se procesima koji zauzimaju više od 6 okvira oduzimaju okviri. Pri oduzimanju okvira odabire se proces s najviše okvira i njemu se oduzima jedan okvir (prvi s lijeva). Potom se postupak ponavlja dok se ne dobiju svi potrebni okviri za novi zahtjev ili se više ne može ništa nikome oduzeti. Oduzeti okviri se ne vraćaju početnim procesima čak i ako se novi zahtjev nije mogao poslužiti (nije se dobilo minimalno 6 okvira). Prikazati stanje sustava do (pokušaja) posluživanja svih procesa. U svakom retku tablice pokazati stanje nakon događaja (npr. nakon „pokreni P1“). Navedeni procesi ne završavaju za vrijeme ovih zahtjeva (svi se još uvijek izvode i za vrijeme zadnjeg zahtjeva). Učitani procesi se ne izbacuju iz memorije (mogu im se oduzeti neki okviri, ali ostaju u minimalno 6 okvira). U tablici upisivati samo broj procesa koji koristi zadane okvire.

poč. stanje																				komentar (po potrebi)
pokreni P1	1	1	1	1	1	1	1	1	1	1										
pokreni P2	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	
pokreni P3	3	3	3	1	1	1	1	1	1	1	3	3	3	2	2	2	2	2	2	
pokreni P4	3	3	3		1	1	1	1	1	1	3	3	3		2	2	2	2	2	nema dovoljno mjesta

9. Neki disk ima 6 površina na kojima su zapisani podaci. Na svakoj stazi nalazi se 2048 sektora, svaki sektor veličine 512 B. Disk se okreće s 6000 okr/min. Pri čitanju disk čita cijelu stazu u interni spremnik te prenosi potrebne sektore u radni spremnik brzinom 4,096 Gbita/s. Za vrijeme prijenosa ne može se čitati s diska, ali može se paralelno pomicati glavu. Datoteka veličine 25 MB kompaktno je smještena na disku. Koliko traje čitanje te datoteke u radni spremnik, ako je vrijeme traženja staze $T_{seek} = 5$ ms te vrijeme postavljanja na susjednu stazu $T_1 = 0,5$ ms.

$2048 * 512 \text{ B} = 1 \text{ MB po stazi}$

$25 \text{ MB} = 25 \text{ staza} \Rightarrow 4 \text{ puna cilindra po 6 staza} + 1 \text{ staza na 5. cilindru}$

$T_R = 1/6000 \text{ okr/min} = 1/100 \text{ okr/s} = 10 \text{ ms}$

$T_{R/2} = 5 \text{ ms} (T_R \text{ potez})$

$T_P = 1 \text{ MB}/4,096 \text{ Gbita/s} = 2,048 \text{ ms}$

$t_{\check{c}} = T_{seek} + [(T_{R/2} + T_R + T_P) * 5 + (T_{R/2} + T_R + \max(T_P, T_1))] * 4 + (T_{R/2} + T_R + T_P)$

obzirom da je $\max(T_P, T_1) = T_P$

$t_{\check{c}} = T_{seek} + (T_{R/2} + T_R + T_P) * 25 = 5 + (5 + 10 + 2,048) * 25 = 431,2 \text{ ms}$

10. Neka datoteka veličine 50 MB kompaktno je smještena na disku koji koristi NTFS particiju s veličinom bloka od 4 KB. Prvi blok datoteke nalazi se na disku u bloku rednog broja 50000.

a) Navesti dio opisnika datoteke koji opisuje njen smještaj.

b) Koji blok diska treba dohvatiti da bi se pročitao 7000. blok datoteke?

a) $50 \text{ MiB} / 4 \text{ KiB} = 50 * 1024 / 4 = 12800$ blokova za datoteku

VCN	LCN	#
1	50000	12800

b) 57000. blok diska