

1. (2) U nekom trenutnu stanje sustava je sljedeće: programsko brojilo  $PC=0x1000$ , kazaljka stoga  $SP=0xF008$ , u memoriji su vrijednosti (adresa:vrijednost): 1000:[instrukcija RET], F000:5000, F004:4000, F008:3000, F00C:2000. Koje će vrijednosti biti u PC i SP nakon obavljanja iduće instrukcije (RET)?

PC = 3000

SP = F00C

2. (4) 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)`.

Primjer:      `uzlaz = 1000110100000001`  
              `izlaz = 000110110100101`  
              (bitovi) 543210987654321

```
unsigned int f(unsigned int ulaz) {
    unsigned int i, izlaz = ulaz;
    for (i = 2; i < 8*sizeof(unsigned int); i += 3)
        izlaz |= 1 << i;

    for (i = 4; i < 8*sizeof(unsigned int); i += 5)
        izlaz = izlaz & (~1 << i);
}
```

3. (2) Pristupni sklop s mogućnošću izravnog pristupa spremniku ima četiri registra. Koji su to registri?

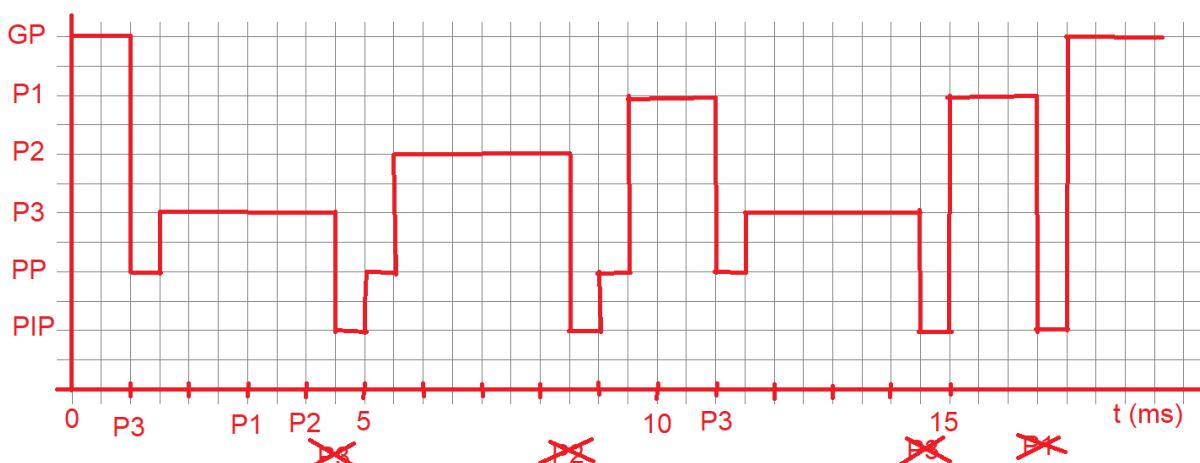
PR – podatkovni registar

RS – registar stanja

AR – adresni registar

BR – brojilo

4. (5) U nekom sustavu sa sklopom za prihvatanje prekida javljaju se prekidi:  $P_3$  u 1. i 11.,  $P_2$  u 4. te  $P_1$  u 3. ms. Prioritet prekida određen je brojem ( $P_3$  ima najveći prioritet). Procedura za prihvatanje prekida (PP) traje 0,5 ms a procedura za povratak iz prekida (PiP) 0,5 ms. Obrada svakog prekida traje po 3 ms. Grafički prikazati aktivnosti procesora u glavnom programu (GP), procedurama za obradu prekida ( $P_i$ ) te procedurama za prihvatanje prekida (PP) i povratak iz prekida (PiP).



5. (4) Koje zajedničke varijable koristi Lamportov algoritam međusobnog isključivanja? U nekom trenutku neka je dretva D4 u KO, dretva D2 uzima broj te dretve D1 i D3 "čekaju" na ulaz u KO, s time da D1 čeka duže. Koje su vrijednosti u tim varijablama u tom trenutku? Prije D4 nije bilo dretvi u KO.

```

ULAZ[1] = 0    BROJ[1] = 2
ULAZ[2] = 1    BROJ[2] = 0
ULAZ[3] = 0    BROJ[3] = 3
ULAZ[4] = 0    BROJ[4] = 1

```

6. (2) Navesti strukturu podataka jezgre (u prikazanom modelu u okviru predmeta).

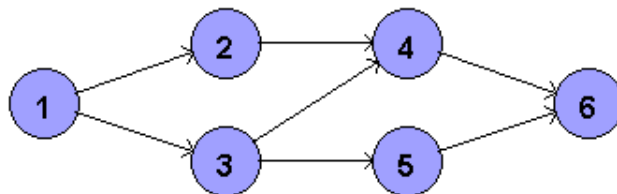
```

opisnici dretvi, UI naprava
liste za dretve: Aktivna_D, Pripravne_D, Odgođenje_D,
                 BSEM[], OSEM[], UI[]

```

7. (5) Neki posao podijeljen je u sedam zadataka  $Z_1$  do  $Z_6$ . Poznato je da će rezultat biti ispravan ako se zadaci izvode slijedno:  $Z_1 \rightarrow Z_2 \rightarrow Z_3 \rightarrow Z_4 \rightarrow Z_5 \rightarrow Z_6$ . 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.

	Z1	Z2	Z3	Z4	Z5	Z6
M1	D		K		K	D
M2	K	D				
M3		K		D		
M4			D	K		K



8. (5) 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.st.					
Aktivna_D	3	3	2	4	2	
Pripravne_D	1	2 1	1	2 1	1	
OSEM[1]	4	4	4	–	–	
Odgođene_D	2^1 5^2	5^2	5^2	5^2	5^2 4^3	
UI[1]	–	–	3	3	3	

9. (5) U nekom sustavu postoji više dretvi koje sudjeluju u posluživanju zahtjeva: više ulaznih, dvije radne i više izlaznih dretvi. Ulazne dretve zaprimaju zahtjev te preko zajedničke varijable  $Z$  predaju taj zahtjev radnim dretvama. Radne dretve zajedno (nezavisno) rade na istom zahtjevu te po dovršetku obrade predaju izlaznim dretvama preko zajedničke varijable  $R$  (obje dretve moraju napraviti `dodaj()`). Izlazne dretve prihvataju rezultat te ga pohranjuju (svaka dretva radi nad drugim rezultatom). Nepotpuni pseudokod takvih dretvi zadan je u nastavku. Proširiti zadani pseudokod pozivima semafora (binarnih i/ili općih) tako da radi prema navedenom tekstu. Operacije `dohvati()`, `obradi()` i `pohrani()` moraju se moći izvoditi paralelno (i od strane istih dretvi).

Zajedničke varijable su:  $Z$  i  $R$

Ulazne dretve:

```
z - lokalna
varijabla dretva

while(1)
{
    z = dohvati()

    ČekajBSEM(1)
    ČekajOSEM(1)
    ČekajOSEM(1)

    Z = z

    PostaviOSEM(2)
    PostaviOSEM(2)
    PostaviBSEM(1)
}
```

Radne dretve:

```
z, r - lokalne
varijable dretve

while(1)
{
    ČekajOSEM(2)
    //ČekajBSEM(2)

    z = Z

    //PostaviBSEM(2)
    PostaviOSEM(1)

    r = obradi(z)

    ČekajOSEM(3)
    ČekajBSEM(2)

    R = dodaj(r)

    PostaviBSEM(2)
    PostaviOSEM(4)
}
```

Izlazne dretve:

```
r - lokalna
varijabla dretve

while(1)
{
    ČekajBSEM(3)
    ČekajOSEM(4)
    ČekajOSEM(4)

    r = R

    PostaviOSEM(3)
    PostaviOSEM(3)
    PostaviBSEM(3)

    pohrani(r)
}
```