

Problem pospane frizerke

(Izvorno *Sleeping barber problem*)

Zadatak

U nekom frizerskom salonu radi jedna frizerka. Salon ima odvojenu čekaonicu s N mjesta gdje čekaju klijenti. Frizerka ima samo jedan stolac u koji može primiti klijenta i raditi na njegovoj frizuri.

Frizerka ujutro otvara vrata salona i do kraja radnog vremena obavlja sljedeću cikličku aktivnost: sjeda u stolac i spava dok ju ne probudi dolazak klijenta ili joj sat zazvoni kada je kraj radnog vremena. Ako je došao klijent uzima ga i radi na njegovoj frizuri. Potom, ako ima još klijenata uzima ih redom, jednog po jednog. Ako je čekaonica prazna, opet sjeda u stolac i spava. Kada u čekaonici ima klijenata frizerka ne smije spavati.

Novi klijent prvo uđe u čekaonicu. Ako nema praznih mjesta, tj. u čekaonici već čeka N klijenata, klijent odlazi bez obavljenog posla. Ako ima praznih stolaca, sjeda i signalizira svoj dolazak frizerki. Kada ga frizerka pozove dolazi k njoj na frizuru. Potom odlazi iz salona.

Simulirati prikazani sustav dretvom/procesom *frizerka* te dretvama/procesima *klijent* koje stvara pomoćna (može i početna ili glavna) dretva/proces, koja signalizira i kraj radnog vremena nakon čega se više ne primaju novi klijenti.

Prijedlog moguće strukture rješenja (rješenje ne mora izgledati ovako)

```
dretva frizerka ()
    otvori salon i postavi znak "OTVORENO"
    ponavljaj
        ako je kraj radnog vremena tada
            postavi znak "ZATVORENO"

        ako ima klijenata u čekaonici tada
            uzmi prvog klijenta iz čekaonice //semafor
            radi na frizuri //sleep
        inače ako nije kraj radnog vremena
            spavaj //semafor
        inače
            // kraj radnog vremena i salon je prazan
            zatvori salon
            završi s radom

dretva klijent (x)
    ispiši da si se pojavio(x)
    ako je znak "OTVORENO" postavljen i ima mjesta u čekaonici tada {
        ažuriraj broj mjesta u čekaonici
        signaliziraj frizerki //semafor
```

```

        čekaj na svoj red //semafor
        // frizerka te pozvala, odi na frizuru, ali prije ažuriraj broj
mjestu u čekaonici
        // - nije potrebno dodatno sinkronizirati čekanje da frizura bude
gotova
        // - dovoljno je ispisati stanje i završiti s dretvom klijenta
    }
    inače {
        danas ništa od frizure //ispis + završi s radom
    }
}

```

Napomene

Korištenje zajedničkih varijabli (npr. *otvoreno*, *br_mjesta*) zaštititi binarnim semaforom!

Neka početna dretva/proces stvara sve ostale dretve/procese, najprije *frizerku* a potom i dretve/procese *klijente* te označava kraj radnog vremena. Trajanje rada na frizuri i raspored dolazaka klijenata odredite sami.

Primjer ispisa (pet klijenata je stvoreno odmah, a kasnije još nekoliko):

```

Frizerka: Otvaram salon
Frizerka: Postavljam znak OTVORENO
Frizerka: Spavam dok klijenti ne dođu
    Klijent(1): Želim na frizuru
    Klijent(1): Ulazim u čekaonicu (1)
Frizerka: Idem raditi na klijentu 1
    Klijent(1): frizerka mi radi frizuru
    Klijent(4): Želim na frizuru
    Klijent(4): Ulazim u čekaonicu (1)
    Klijent(3): Želim na frizuru
    Klijent(3): Ulazim u čekaonicu (2)
    Klijent(2): Želim na frizuru
    Klijent(2): Ulazim u čekaonicu (3)
    Klijent(5): Želim na frizuru
    Klijent(5): Nema mjesta u čekaoni, vratit ću se sutra
Frizerka: Klijent 1 gotov
Frizerka: Idem raditi na klijentu 4
    Klijent(4): frizerka mi radi frizuru
Frizerka: Klijent 4 gotov
Frizerka: Idem raditi na klijentu 3
    Klijent(3): frizerka mi radi frizuru
Frizerka: Klijent 3 gotov
Frizerka: Idem raditi na klijentu 2
    Klijent(2): frizerka mi radi frizuru
Frizerka: Klijent 2 gotov
Frizerka: Spavam dok klijenti ne dođu
    Klijent(6): Želim na frizuru
    Klijent(6): Ulazim u čekaonicu (1)
Frizerka: Idem raditi na klijentu 6
    Klijent(6): frizerka mi radi frizuru
Frizerka: Klijent 6 gotov
Frizerka: Spavam dok klijenti ne dođu

```

```

    Klijent(7): Želim na frizuru
    Klijent(7): Ulazim u čekaonicu (1)
Frizerka: Idem raditi na klijentu 7
    Klijent(7): frizerka mi radi frizuru
Frizerka: Klijent 7 gotov
Frizerka: Spavam dok klijenti ne dođu
Frizerka: Postavljam znak ZATVORENO
Frizerka: Zatvaram salon
-----

```

Kanibali i misionari

Zadatak

Rješiti problem prijevoza kanibala i misionara. Na obali neke široke rijeke postoji čamac koji prevozi kanibale i misionare na drugu stranu obale. Kapacitet čamca je 7 putnika. U čamcu moraju biti najmanje 3 putnika da on može krenuti. U čamcu ne smije biti više kanibala od misionara, dok su sve ostale kombinacije putnika dozvoljene (npr. u čamcu mogu biti i samo kanibali). Kanibali i misionari dolaze sa obje strane rijeke. Jedan misionar dolazi svake dvije sekunde, a kanibal svake sekunde (odabir obale je slučajna). Nakon što pređu preko rijeke putnici odlaze dalje (nema ih više u sustavu). U sustavu neka postoji samo jedan čamac, a svaki misionar i kanibal predstavljaju po jednu dretvu/proces. Čamac je također jedna dretva/proces koji pri svakom prijelazu ispisuje koga prevozi (npr. "Prevezeni: misionar, kanibal, misionar, misionar"). Pretpostaviti da je čamac u početku na desnoj obali. Nakon što se u čamcu popune tri (ili više) mjesta, čamac pričekava još sekundu, u kojoj se još netko može ukrcati u čamac, prema navedenim pravilima, te potom kreće preko rijeke - što traje dvije sekunde. Dretve/procese misionare i kanibale stvara pomoćna dretva/proces. Ispravno sinkronizirati dretve/procese kanibale, misionare i čamac.

Primjer ispisa

```

Legenda: M-misionar, K-kanibal, C-čamac,
          LO-lijeva obala, DO-desna obala
          L-lijevo, D-desno

```

```

C: prazan na desnoj obali
C[D]={} LO={} DO={}

```

```

M1: došao na lijevu obalu
C[D]={} LO={M1} DO={}

```

```

K1: došao na desnu obalu

```

C[D]={} LO={M1} DO={K1}

K1: ušao u čamac

C[D]={K1} LO={M1} DO={}

K2: došao na lijevu obalu

C[D]={K1} LO={M1 K2} DO={}

M2: došao na desnu obalu

C[D]={K1} LO={M1 K2} DO={M2}

M2: ušao u čamac

C[D]={K1 M2} LO={M1 K2} DO={}

K3: došao na desnu obalu

C[D]={K1 M2} LO={M1 K2} DO={K3}

M3: došao na desnu obalu

C[D]={K1 M2} LO={M1 K2} DO={K3 M3}

M3: ušao u čamac

C[D]={K1 M2 M3} LO={M1 K2} DO={K3}

K3: ušao u čamac

C[D]={K1 M2 M3 K3} LO={M1 K2} DO={}

C: tri putnika ukrcana, polazim za jednu sekundu

C[D]={K1 M2 M3 K3} LO={M1 K2} DO={}

M4: došao na desnu obalu

C[D]={K1 M2 M3 K3} LO={M1 K2} DO={M4}

M4: ušao u čamac

C[D]={K1 M2 M3 K3 M4} LO={M1 K2} DO={}

K4: došao na desnu obalu

C[D]={K1 M2 M3 K3 M4} LO={M1 K2 K4} DO={}

C: prevozim s desne na lijevu obalu: K1 M2 M3 K3 M4

K5: došao na desnu obalu

C[D]={K1 M2 M3 K3 M4} LO={M1 K2 K4} DO={K5}

M5: došao na lijevu obalu

C[D]={K1 M2 M3 K3 M4} LO={M1 K2 K4 M5} DO={K5}

C: preveo s desne na lijevu obalu: K1 M2 M3 K3 M4

C: prazan na lijevoj obali

C[L]={} LO={M1 K2 K4 M5} DO={K5}

M1: ušao u čamac

C[L]={M1} LO={K2 K4 M5} DO={K5}

K2: ušao u čamac

C[L]={M1 K2} LO={K4 M5} DO={K5}

M5: ušao u čamac

$C[L] = \{M1 \ K2 \ M5\}$ $LO = \{K4\}$ $DO = \{K5\}$

C: tri putnika ukrcana, polazim za jednu sekundu

K5: ušao u čamac

$C[L] = \{M1 \ K2 \ M5 \ K4\}$ $LO = \{\}$ $DO = \{K5\}$

C: prevozim s lijeve na desnu obalu: M1 K2 M5 K4

...