

Zadatak 5.2.x

Sličan zadatak Zadatku 5.2. ali „zdraviji“ (možda).

U neku trgovinu ulaze kupci koje si žele napraviti sendvič. Za napraviti sendvič te osobe trebaju kruh, šunku i sir. Svaka osoba već ima jedan sastojak i trebaju joj iduća dva. Trgovac u trgovini ima sva tri sastojka u velikoj količini. Trgovac nasumice odabire dva različita sastojka i stavlja ih na stol. Kupac koji je u trgovini i koji ima različit sastojak od onih na stolu, uzima sastojke sa stola, izlazi iz trgovine, slaže si sendvič i jede. U trgovini u svakom trenutku može biti najviše tri kupca i to s različitim sastojcima. Tek kad neki kupac izađe, u trgovinu može ući drugi, ali samo takav koji treba isti sastojak kao i ovaj koji izlazi. Riješiti sinkronizaciju semaforima.

Početno rješenje:

- ulaz u trgovinu štititi s tri binarna semafora: `ulaz_kruh, ulaz_šunka i ulaz_sir`
- signal trgovcu da je stol prazan preko binarnog semafora: `stol_prazan`
- za svaki sastojak na stolu po jedan binarni semafor: `kruh, šunka, sir`

Početne vrijednosti semafora:

```
BSEM[ulaz_kruh].v = BSEM[ulaz_šunka].v = BSEM[ulaz_sir] = BSEM[stol_prazan] = 1  
BSEM[kruh].v = BSEM[šunka].v = BSEM[sir] = 0
```

dretva trgovac

```
{  
  ponavljaj {  
    ČekajBSEM(stol_prazan)  
    s1,s2 = nasumice odaberi dva različita sastojka //dva od: kruh, šunka, sir  
    PostaviBSEM(s1)  
    PostaviBSEM(s2)  
  }  
}
```

dretve osoba_s_kruhom

```
{  
  ČekajBSEM(ulaz_kruh)  
  uđi u trgovinu  
  
  ČekajBSEM(šunka)  
  ČekajBSEM(sir)  
  uzmi sastojke sa stola  
  PostaviBSEM(stol_prazan)  
  
  odi van trgovine  
  PostaviBSEM(ulaz_kruh)  
  napravi sendvič  
  jedi  
}
```

dretve osoba_sa_šunkom

```
{  
  ČekajBSEM(ulaz_šunka)  
  uđi u trgovinu  
  
  ČekajBSEM(kruh)  
  ČekajBSEM(sir)  
  uzmi sastojke sa stola  
  PostaviBSEM(stol_prazan)  
  
  odi van trgovine  
  PostaviBSEM(ulaz_šunka)  
  napravi sendvič  
  jedi  
}
```

dretve osoba_sa_sirom

```
{  
  ČekajBSEM(ulaz_sir)  
  uđi u trgovinu  
  
  ČekajBSEM(kruh)  
  ČekajBSEM(šunka)  
  uzmi sastojke sa stola  
  PostaviBSEM(stol_prazan)  
  
  odi van trgovine  
  PostaviBSEM(ulaz_sir)  
  napravi sendvič  
  jedi  
}
```

Problem prikazanog rješenja: može se dogoditi *potpuni zastoj!*

Dva različita kupca mogu uzeti različite sastojke (proći samo prvi od **crvenih** semafora), ali ni jedan neće imati sva tri te neće dojaviti trgovcu da je stol prazan. Trgovac neće staviti nove sastojke i svi će čekati beskonačno.

Problem potpuna zastoja će detaljnije biti obrazložen u okviru 6. poglavlja.

Potpuni zastoj nije dozvoljen u ispravnom rješenju kada se koriste semafori (i monitori)!

Rješenje s provjerama

- ulaz u trgovinu štititi s tri binarna semafora: `ulaz_kruh, ulaz_šunka i ulaz_sir`
- signal trgovcu da je stol prazan preko binarnog semafora: `stol_prazan`
- signal kupcima da su na stolu sastojci: `stol_pun`
- zajedničke varijable: `s1, s2` – označavaju sastojke koji su na stolu

Početne vrijednosti semafora:

```
BSEM[ulaz_kruh].v = BSEM[ulaz_šunka].v = BSEM[ulaz_sir] = BSEM[stol_prazan] = 1  
BSEM[stol_pun].v = 0
```

dretva trgovac

```
{  
  ponavlja {  
    ČekajBSEM(stol_prazan)  
    s1,s2 = nasumice odaberi dva različita sastojka() //dva od: kruh, šunka, sir  
    PostaviBSEM(stol_pun)  
  }  
}
```

dretve osoba_s_kruhom

```
{  
  ČekajBSEM(ulaz_kruh)  
  uđi u trgovinu  
  
  ponavlja {  
    imam_sve = LAŽ //imam_sve: lokalna varijabla, svaka dretva ima svoju  
    ČekajBSEM(stol_pun)  
    ako je ( s1 != "kruh" I s2 != "kruh" ) onda {  
      uzmi sastojke sa stola  
      imam_sve = ISTINA  
      PostaviBSEM(stol_prazan)  
    }  
    inače {  
      PostaviBSEM(stol_pun)  
    }  
  }  
  dok je imam_sve == LAŽ //ponavlja petlju dok je uvjet ispunjen  
  
  odi van trgovine  
  PostaviBSEM(ulaz_kruh)  
  napravi sendvič  
  jedi  
}
```

Sličan kod je za **osoba_sa_šunkom** i **osoba_sa_sirom**, uz odgovarajuće zamjene (obojani tekst).

Problem: radno čekanje

Primjerice, ukoliko trgovac na stol stavi sastojke za kupca koji nije u trgovini, ostala dva kupca (ili i samo jedan) će svejedno izvoditi dok je petlju!

Radno čekanje nije dozvoljeno u ispranom rješenju kada se koriste semafori (i monitori)!

Rješenje s više semafora

- ulaz u trgovinu štititi s tri binarna semafora: `ulaz_kruh, ulaz_šunka i ulaz_sir`
- signal trgovcu da je stol prazan preko binarnog semafora: `stol_prazan`
- signal kupcima da su na stolu sastojci: `kruh, šunka, sir`
- zajedničke varijable: `s1, s2` – označavaju sastojke koji su na stolu
- korištenje zajedničkih varijabli zaštićeno kritičnim odsječkom, semaforom `KO`

Početne vrijednosti semafora:

```
BSEM[ulaz_kruh].v = BSEM[ulaz_šunka].v = BSEM[ulaz_sir] = BSEM[stol_prazan] = 1
BSEM[KO].v = 1
BSEM[kruh].v = BSEM[šunka].v = BSEM[sir] = 0
```

dretva trgovac

```
{
  ponavljaj {
    ČekajBSEM(stol_prazan)

    ČekajBSEM(KO)
    s1,s2 = nasumice odaberi dva različita sastojka() //dva od: kruh, šunka, sir
    PostaviBSEM(KO)

    PostaviBSEM(kruh)           // svaki puta kad se nešto stavi na stol
    PostaviBSEM(šunka)          // samo se jednom javi „svakom kupcu“
    PostaviBSEM(sir)
  }
}
```

dretve osoba_s_kruhom

```
{
  ČekajBSEM(ulaz_kruh)
  uđi u trgovinu

  ponavljaj {
    imam_sve = LAŽ //imam_sve: lokalna varijabla, svaka dretva ima svoju
    ČekajBSEM(kruh)
    ČekajBSEM(KO)
    ako je ( s1 != "kruh" I s2 != "kruh" ) onda {
      uzmi sastojke sa stola
      imam_sve = ISTINA
      PostaviBSEM(stol_prazan)
    }
    PostaviBSEM(KO)
  }
  dok je imam_sve == LAŽ //ponavljaj petlju dok je uvjet ispunjen

  odi van trgovine
  PostaviBSEM(ulaz_kruh)
  napravi sendvič
  jedi
}
```

Sličan kod je za **osoba_sa_šunkom** i **osoba_sa_sirom**, uz odgovarajuće zamjene (obojani tekst).

Algoritam radi. Ali treba 5 semafora (+3 za kontrolu ulaska u trgovinu)!