

## Poznámka

Dokumentácia môže riešenie často porovnávať s riešením atómovej elektrárne #1 (ďalej iba AE1), ktoré bolo ukázané na seminári a ktoré bolo použité ako základ pri riešení atómovej elektrárne #2.

## Analýza

O aké synchronizačné úlohy sa bude jednať?

- Lightswitch (mutex, počítadlo, semafor)
- Turniket (semafor(1))
- Event

Vďaka nepomeru čidiel a monitorov sme v AE1 museli, aby neprišlo k vyhladoveniu, použiť **turniket**. Kedykoľvek potreboval monitor čítať, prešiel cez turniket, čo zastavilo všetky čidlá, ktoré by chceli zapisovať do čakajúcej fronty. Po prečítaní bol turniket opäť uvoľnený.

Teraz sa situácia otočila. Máme príliš veľa monitorov na počet čidiel. Tentokrát hrozí vyhladovenie u čidiel. Analogicky teda otočíme aj funkcionality. A teda, vykonávanie čidiel uzavrieme do KO, opäť pomocou turniketu.

Kvôli prehľade výpisov potrebujeme evidenciu toho, koľko entít naraz zapisuje, prípadne číta. Na to sme, ako v AE1, použili **lightswitch** s počítadlom. Tentokrát evidujeme aj počet monitorov, aj počet čidiel. Implementáciu Lightswitch sme upravili tak, aby pri lock() vracala counter.

V AE1 mohli monitory čítať akonáhle bol dostupný aspoň jeden údaj čidla. V tejto úlohe musia monitory počkať, kým prvýkrát zapíšu všetky čidlá. Toto sme vyriešili výmenou riadkov, kde signalizácia **event**-u nastane až "keď posledné čidlo zhasne".

## Postup

Čidlá aj monitory reprezentujeme ako Thread objekty. Thready pre čidlá spúšťajú funkciu pre čidlá, Thready pre monitory spúšťajú funkciu pre monitory.

Definujeme si triedu Lightswitch, ktorá bude v sebe obsahovať funkcionality "posledný nech zhasne".

Zabezpečíme, aby čidlá pri zapisovaní uzamkli monitory.

Zabezpečíme, aby monitory začali s čítaním až keď každé čidlo zapíše aspoň raz.

Implementujeme čakacie doby a doby trvania čítania/zápisu monitorov a rôznych čidiel.

## Pseudokód

Je definovaný veľmi symbolicky, nakoľko pseudokód má vyjadrovať iba myšlienku implementácie. Nevieme, akú knižnicu na synchronizačné objekty používa používateľ, takže jednotlivým triedam sme sa implementačne v pseudokóde nevenovali. Konkrétnu implementáciu v jazyku Python možno nájsť v repozitári.

```
turniket := new Turniket
cidloLightswitch := new Lightswitch
monitorLightswitch := new Lightswitch
dataPresent := new Event

for i := 0 to 3 do
    t := new Thread
    t.launch(fcidlo)
endfor

for i := 0 to 8 do
    t := new Thread
    t.launch(fmonitor)
endfor

func fcidlo()
    while 1 equal to 1 do
        sleep(Interval<50ms,60ms>)
        turniket.stop()
        pocetCidiel := cidloLightswitch.lock()

        if t is last do
            sleep(Interval<20ms,25ms>)
        else do
            sleep(Interval<20ms,25ms>)
        end ifelse

        turniket.go()
        cidloLightswitch.unlock()
        dataPresent.signal()
    endwhile
endfunc

func fmonitor()
    dataPresent.wait()
    while 1 equal to 1 do
        pocetMonitorov := monitorLightswitch.lock()
        sleep(Interval<40ms,50ms>)
        monitorLightswitch.unlock()
    endwhile
endfunc
```

## Beh programu

Program spustíme a sledujeme výstup:

```
cidlo "00": pocet_zapisujucich_cidiel=00, trvanie_zapisu=0.018
cidlo "01": pocet_zapisujucich_cidiel=01, trvanie_zapisu=0.018
cidlo "02": pocet_zapisujucich_cidiel=02, trvanie_zapisu=0.021
monit "00": pocet_citajucich_monitorov=00 trvanie_citania=0.050
monit "02": pocet_citajucich_monitorov=01 trvanie_citania=0.042
monit "01": pocet_citajucich_monitorov=02 trvanie_citania=0.041
monit "04": pocet_citajucich_monitorov=03 trvanie_citania=0.047
monit "06": pocet_citajucich_monitorov=04 trvanie_citania=0.045
monit "05": pocet_citajucich_monitorov=05 trvanie_citania=0.046
monit "07": pocet_citajucich_monitorov=06 trvanie_citania=0.046
monit "03": pocet_citajucich_monitorov=07 trvanie_citania=0.043
monit "01": pocet_citajucich_monitorov=07 trvanie_citania=0.041
cidlo "00": pocet_zapisujucich_cidiel=00, trvanie_zapisu=0.012
cidlo "01": pocet_zapisujucich_cidiel=01, trvanie_zapisu=0.018
cidlo "02": pocet_zapisujucich_cidiel=02, trvanie_zapisu=0.023
monit "06": pocet_citajucich_monitorov=00 trvanie_citania=0.041
monit "02": pocet_citajucich_monitorov=01 trvanie_citania=0.044
monit "01": pocet_citajucich_monitorov=02 trvanie_citania=0.048
monit "00": pocet_citajucich_monitorov=03 trvanie_citania=0.044
monit "05": pocet_citajucich_monitorov=04 trvanie_citania=0.040
monit "03": pocet_citajucich_monitorov=05 trvanie_citania=0.050
monit "04": pocet_citajucich_monitorov=06 trvanie_citania=0.043
monit "07": pocet_citajucich_monitorov=07 trvanie_citania=0.045
cidlo "00": pocet_zapisujucich_cidiel=00, trvanie_zapisu=0.015
cidlo "02": pocet_zapisujucich_cidiel=01, trvanie_zapisu=0.022
cidlo "01": pocet_zapisujucich_cidiel=02, trvanie_zapisu=0.019
monit "05": pocet_citajucich_monitorov=00 trvanie_citania=0.048
monit "07": pocet_citajucich_monitorov=01 trvanie_citania=0.049
```

Hneď po spustení programu monitory **nič nerobia**. Najprv musia počkať, kým všetky čidlá napíšu.

Po zapísaní čidlá zapisujú najsôr až o 50 ms. Medzitým čítajú monitory.

Potom zase čidlá potrebujú zapisovať, tak prejdú cez turniket a monitory sú uzamknuté. Čidlá dozapisujú a monitory prevezmú riadenie.