

Urobte analýzu, o aké typy synchronizačných úloh (prípadne ich modifikácie či kombinácie) sa v tejto úlohe jedná.

Ide o úlohu vzájomného vylúčenia kategórií procesov. Monitory tvoria jednu kategóriu, čidlá tvoria druhú kategóriu. Pre obe kategórie platí, že viacerí členovia kategórie môžu naraz pristupovať k údajom, a to z toho dôvodu, že:

1. monitory údaje iba čítajú, nijako ich nemodifikujú, takže viaceré naraz môžu sprístupňovať údaje toho istého čidla,
2. keďže má každé čidlo svoj priestor, kam ukladá údaje, nemôžu si čidlá navzájom svoje údaje prepisovať; to znamená, že aj čidlá môžu aktualizovať svoje údaje viaceré súčasne.

Taktiež ide aj o problém prednostného riadenia prístupu. Keďže chceme aby boli aktuálne údaje vždy zapísane a senzorov je menej, musia mať väčšiu prioritu.

Presne namapujte vami zvolené synchronizačné úlohy (primitívy) na jednotlivé časti zadania.

Jednu kategóriu tvoria monitory a druhú čidlá. V prípade že nastane čas na aktualizáciu tak chceme aby tie prebrali tok riadenia a vykonali sa všetky zápisy. Pokiaľ však niekto číta, nechceme aby bolo umožnené v tom čase zapisovať a naopak. To zabezpečuje LS.

Monitory vykonávajú svoju činnosť neustále, ich činnosť trvá kratšie ako čidlá a zároveň je ich viac. Aby nenastalo vyhľadovanie a zároveň sme zapísali aktuálne namerané hodnoty čo najskôr ako je to možné, preto je potrebné vyriešiť aj problém prednostného riadenia prístupu. Toto je zabezpečuje LS na oboch stranách. (podľa prednášky)

```

class LightSwitch():
    // Standard implementation from previous assignments

class SimpleBarrier ():
    // Standard implementation from previous assignments


class PowerPlant():
    ls_monitor = LightSwitch()
    ls_sensor = LightSwitch()
    monitor_access = Semaphore(1)
    sensor_access = Semaphore(1)
    barrier = SimpleBarrier(3)

def monitor(monitor_id):
    while True:
        // monitor nemôže pracovať, kým všetky 3 senzory nezapísali
        barrier.event.wait()

        //začiatok turniketu v prípade prednostého prístupu senzorov tu
        //budú čakať
        monitor_access.wait()

        //počet čítajúcich monitorov a zapnutie LS pre ne
        n_monitors_reading = ls_monitor.lock(sensor_access)

        //pustí ďalší monitor
        monitor_access.signal()

        print(
            'monit "%02d": pocet_citajucich_monitorov=%02d,
            trvanie_citania=%03f\n')

        // „čítanie dát“
        sleep(40-50 ms)

        //ukončenie LS
        ls_monitor.unlock(sensor_access)

```

```

// time_range_for_actualisation - range pre daný typ senzora
def sensor(sensor_id, time_range_for_actualisation):
    while True:
        // „čas medzi aktualizáciami“
        sleep(40-50ms)

        // počet čítajúcich senzorov a zapnutie LS pre ne, a zároveň si
        // vyžiada prístup
        n_sensors_writing = ls_sensor.lock(monitor_access)

        //čáká ak ešte číta monitor
        sensor_access.wait()
        print(
            'cidlo "%02d": pocet_zapisujucich_cidiel=%02d, '
            'trvanie_zapisu=%03f\n')

        // „zapisavanie“ dát
        sleep(time_range_for_actualisation)

        sensor_access.signal()

        //uvolnenie bariéry pre čakajúce monitory
        barrier.wait_with_events()
        ls_sensor.unlock(monitor_access)

```

```

def init():
    p = PowerPlant()
    for i in range(2):
        create_and_run_thread(p. sensor, monitor_id, pt_range)

    for i in range(1):
        create_and_run_thread(p. sensor, monitor_id, h_range)

    for i in range(8):
        create_and_run_thread(p.monitor, monitor_id)

```