```
def init():
    num_sensors = 3

    access_data = Semaphore(1)
    barrier = Barrier(num_sensors)
    turnstile = Semaphore(1)
    ls_monitor = Lightswitch()
    ls_sensor = Lightswitch()
    valid_data = Event()

    for i in range(8):
        t = Thread(monitor, i, access_data, turnstile, ls_monitor, valid_data)
    for i in range(3):
        t = Thread(sensor, i, access_data, barrier, turnstile, ls_sensor,
        valid_data)


''' Predtym, ako cidlo zvysi hodnotu semafora, a tym sa aktivuju monitory,
je nutne, aby sa vsetky cidla pockali
Na toto bola pouzita bariera
'''
class Barrier:
    def __init__(self, n):
        # ...

    def wait(self):
        # ...


''' Nakolko citanie moze vykonavat viacero monitorov naraz a zapisovanie
viacero cidiel, pouzijeme vypinac - teda staci, ked prvy zazne a posledny
zhasne
'''
class Lightswitch:
    def __init__(self):
        # ...

    def lock(self, sem):
        # ...
        return self.counter

    def unlock(self, sem):
        # ...


def monitor(monitor_id, access_data, turnstile, ls_monitor, valid_data):

    ''' Vsetky monitory cakaju, kym vsetky cidla nezapisu data
    '''
    valid_data.wait()

    ''' Zmenou oproti predch ulohe je, ze monitory neposielaju ziadost kazdych
    500ms, ale neustale
    '''
    while True:

        turnstile.wait()

        ''' Pomocou vypinaca pristupuju teraz k datam len monitory
        '''
        monitor_num = ls_monitor.lock(access_data)
        turnstile.signal()
        print('monit "%02d": pocet_citajucich_monitorov=%02d, trvanie_citania=%03d\n')

        ''' Cas citania
        '''
        sleep(50-60 ms)
        ls_monitor.unlock(access_data)


def sensor(sensor_id, access_data, barrier, turnstile, ls_sensor, valid_data):
    while True:
```

```
            turnstile.wait()

            sensor_num = ls_sensor.lock(access_data)

            ''' Zmenou oproti predch ulohe je, ze metody turniketu wait a signal
            nie su volane hned po sebe, ale bola vymenena turnstile.signal()
            a ls_sensor.lock(access_data), kedze tentokrat monitory necakaju 500ms,
            ale neustale posielaju ziadosti, tak by doslo k vyhladovaniu cidiel
            Vyvolanie metody signal nad turniketom musi teda nastat po vyvolani metody
            lock nad vypinacom
            '''
            turnstile.signal()

            if (sensor_id == 0):

                ''' Cas zapisovania pre cidlo H
                '''
                writing_time = (randint(20, 25) / 1000)
            else:

                ''' Cas zapisovania pre cidla P a T
                '''
                writing_time = (randint(10, 20) / 1000)
            print('cidlo "%02d": pocet_zapisujucich_cidiel=%02d, trvanie_zapisu=%03d\n')
            sleep(writing_time)

            ''' Predtym ako nastane udalost je potrebne, aby sa vsetky cidla pockali
            - teda vsetky cidla zapisu udaje, a az potom mozu monitory k nim pristupovat
            Na toto bola pouzita bariera
            '''
            barrier.wait()

            ''' Vyvolanie udalosti. Potom uz mozu monitory pristupovat k udajom
            '''
            valid_data.set()
            ls_sensor.unlock(access_data)

```