

Dokumentácia - zadanie elektrární

Rovnako ako v predchádzajúcom zadaní, ide o úlohu vzájomného vylúčenia kategórií procesov.

Monitory tvoria jednu kategóriu, čidlá tvoria druhú kategóriu. Pre obidve kategórie platí, že viacerí členovia môžu naraz pristupovať k údajom. Čidla zapisovať a monitory čítať.

V tomto zadaní máme 3 druhy čidiel. Všetky vykonávajú aktualizáciu každých 50-60ms, ale ich trvanie aktualizácie je rozdielne. Prvým dvom druhom to trvá 10-20ms a tretiemu 20-25ms. Okrem čidiel tam máme aj operátorov, ktorí čítajú dáta neustále a každé čítanie trvá 40-50ms.

Potrebujeme teda dosiahnuť aby pri zápise prestali monitory čítať a všetky čidlá stihli zapísať údaje. Taktiež nemôžeme zabudnúť na to aby monitory začali až keď tam budú nejaké údaje.

Na vyriešenie tohto použijeme prepínače (Lightswitch) v kombinácii so semaformi pre monitory a čidlá, čo nám zabezpečí, že sa nebudú navzájom prerušovať a teda vždy všetky monitory prečítajú a všetky čidlá zapíšu údaje. Problém s tým aby nenastalo vyhľadovanie zapisovateľov, keďže ich je viac, vyriešime pridaním bariéry, na ktorej budú monitory stáť. Táto bariéra nám zároveň aj rieši to, že sa pri spustení najskôr zapíšu údaje.

Pseudo kód

```
1. class PowerPlant:
2.     def __init__(self):
3.         self.barrier = Barrier(3)
4.         self.monitor_ls = LightSwitch()
5.         self.sensor_ls = LightSwitch()
6.         self.no_sensors = Semaphore(1)
7.         self.no_monitors = Semaphore(1)
8.
9.     def monitor(self, monitor_id):
10.        while True:
11.            //Monitor caka na bariere
12.            self.barrier.event.wait()
13.            //Monitor caka na ukocenie lightswitchu cidla
14.            self.no_monitors.wait()
15.
16.            //Ziskame pristup k ulozeniu
17.            number_of_monitors_reading =
                self.monitor_ls.lock(self.no_sensors)
18.
19.            self.no_monitors.signal()
20.
21.            print('monit "%02d":  pocet_citajucich_monitorov=%02d,
                    trvanie_citania=%03d\n')
22.
23.            //Dlžka trvania citania
24.            sleep(400-500 ms)
25.            //odblokujeme aby mohol nastat zapis
26.            self.monitor_ls.unlock(self.no_sensors)
27.
```

```
28. def sensor(self, sensor_id, duration):
29.     while True:
30.         sleep(500-600 ms)
31.
32.         //Ziskame pristup k ulozišku
33.         number_of_sensors_writing =
34.             self.sensor_ls.lock(self.no_monitors)
35.         //Zablokujeme monitory aby nemohli citat
36.         self.no_sensors.wait()
37.
38.         print('cidlo "%02d": pocet_zapisujucich_cidiel=%02d,
39.               trvanie_zapisu=%03d\n')
40.         //Dlžka trvania zapisu
41.         sleep(duration)
42.
43.         //odblokujeme monitory
44.         self.no_sensors.signal()
45.         //bariera sa uvolni, ked prejde posledne cidlo
46.         self.barrier.wait_with_events()
47.         self.sensor_ls.unlock(self.no_monitors)
48.
49.     for monitor_id in <0,7>:
50.         create_and_run_thread(monitor, monitor_id)
51.     for cidlo_id in <0,2>:
52.         create_and_run_thread(cidlo, cidlo_id, 10-15 ms)
53.     create_and_run_thread(cidlo, cidlo_id=3, 20-25 ms)
```