

Systemy Operacyjne

Laboratoria nr 6

Problem ucztujących filozofów

Mikołaj Nowak

280082

1 Wstep

Należało rozwiązać problem uczujących filozofów.

2 Kod

By uniknąć problemów użyłem mutex'ów i semafora ograniczającego dostęp filozofów do widełców. Pełny kod załączam w mailu.

```
1 void filozof(int i) {
2     while (true) {
3         // Filozof sko czy my le i jest g odny
4         {
5             lock_guard<mutex> lock(printm); // Ensure exclusive
6             // access to print operations.
7             mvprintw(i, 0, "Filozof %d is hungry %d ", i,
8                 zag odzenie[i]);
9             refresh();
10        }
11
12        // ograniczenie ilo ci filozof w jednocze nie jedz cych
13        semaphore.acquire();
14
15        // zabranie widełcy, parzysty filozof bierze najpierw tego po
16        // swojej lewej stronie a nieparzysty odwrotnie by unikn
17        // sytuacji w ktorej trzymaj tylko jeden widelec i nie
18        // chc go zwolni
19        if (i % 2 == 0) {
20            widelec[i].lock();
21            widelec[(i + 1) % N].lock();
22        } else {
23            widelec[(i + 1) % N].lock();
24            widelec[i].lock();
25        }
26
27        // filozof je
28        {
29            lock_guard<mutex> lock(printm);
30            mvprintw(i, 0, "Filozof %d is eating %d ", i,
31                zag odzenie[i]);
32            refresh();
33        }
34
35        // odczekanie czasu jedzenia
36        this_thread::sleep_for(chrono::milliseconds(1500, 1700));
37
38        // od o enie widełca
39        widelec[i].unlock();
40        widelec[(i + 1) % N].unlock();
41
42        // zniesienie ograniczenia po sko czonym jedzeniu
43        semaphore.release();
44    }
45}
```

```

39         // Filozof my li
40     {
41         lock_guard<mutex> lock(printm);
42         mvprintw(i, 0, "Filozof %d is thinking %d", i,
43             zag odzenie[i]);
44         refresh();
45     }
46
47     // oczekiwanie losowego czasu my lenia
48     this_thread::sleep_for(chrono::milliseconds(los(1600, 1900)));
49
50     // zwi kszenie licznika g odu
51     zag odzenie[i]++;
52 }

```

3 Wykonanie

```

userbrigh@Hp15Ub:~/CLionProjects/Ucztujący_filozofowie$ g++ -std=c++20 Stół.c
pp -o uczta -lncurses
userbrigh@Hp15Ub:~/CLionProjects/Ucztujący_filozofowie$ ./uczta

```

Kompilacja pliku

```

Filozof 0 is hungry 40
Filozof 1 is eating 40
Filozof 2 is thinking 40
Filozof 3 is eating 40
Filozof 4 is thinking 39

```

Wykonanie pliku

```

Filozof 0 is eating 1677
Filozof 1 is hungry 1609
Filozof 2 is thinking 1609
Filozof 3 is eating 1532
Filozof 4 is thinking 1532

```

Wykonanie dużej ilości symulacji

4 Wnioski

Powyższy program rozwiązuje problem uczujących filozofów zapobiega zakleszczaniu się filozofów, sytuacji w której to nie będzie możliwy progress oraz w której filozofowie będą głodować ze względu na bardzo nierównomierne rozłożenie surowców.

5 Bibliografia

- <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/files/so/w9.pdf>
- https://pl.wikipedia.org/wiki/Problem_uczujacych_filozofow