

Teoria Współbieżności Lab2 - sprawozdanie

Autor: Michał Flak

Ćwiczenie

Poleceniem zadania było zaimplementowanie semafora w Javie z użyciem monitora, następnie zastosowanie go do załączonego problemu wyścigu.

Implementacja semafora

Zaimplementowano semafor binarny z użyciem metod *synchronized*:

```
public class Semafor {
    private boolean _stan = true;

    public Semafor(boolean initial) {
        _stan = initial;
    }

    public synchronized void P() {
        while (!_stan) {
            try{
                wait();
            } catch (InterruptedException ex) {

            }
        }
        _stan = false;
    }

    public synchronized void V() {
        _stan = true;
        notify();
    }
}
```

Zastosowanie semafora do problemu

Zmodyfikowano klasę licznika w następujący sposób, z użyciem semafora chroniącego wartość licznika:

```
class Counter {
    private int _val = 0;
    private Semafor _sem;

    public Counter(Semafor _sem) {
        this._sem = _sem;
    }

    public void inc(){
        _sem.P();
        ++_val;
        _sem.V();
    }

    public void dec(){
        _sem.P();
    }
}
```

```

        --_val;
        _sem.V();
    }
    public int value() {
        return _val;
    }
}

```

Wątki dekrementujące oraz inkrementujące (bez zmian):

```

class IThread extends Thread {
    private Counter _cnt;
    public IThread(Counter c) {
        _cnt = c;
    }
    public void run() {
        for (int i = 0; i < 1000000; ++i) {
            try {
                _cnt.inc();
            } catch (Exception ex) {

            }
        }
    }
}

class DThread extends Thread {
    private Counter _cnt;
    public DThread(Counter c) {
        _cnt = c;
    }
    public void run() {
        try {
            _cnt.dec();
        } catch (Exception ex) {

        }
    }
}

```

Stworzenie semafora i uruchomienie wątków w main:

```

public static void main(String[] args) {
    Counter cnt = new Counter(new Semafor(true));
    IThread it = new IThread(cnt);
    DThread dt = new DThread(cnt);

    it.start();
    dt.start();

    try {
        it.join();
        dt.join();
    } catch (InterruptedException ie) { }
}

```

```
        System.out.println("value=" + cnt.value());  
    }
```

Wyniki

Przed zmianami, załączony kod zwracał wartość 0, co wskazuje ewidentnie na wyścig danych.

Po zastosowanych zmianach wartość licznika na wyjściu to 999999.

Wnioski

Zaimplementowany semafor skutecznie rozwiązał problem wyścigu. Wątki czekają na swoją kolej w dostępie do licznika.