

Teoria Współbieżności Lab1 - sprawozdanie

Autor: Michał Flak

Zadanie 1

Poleceniem zadania było stworzenie dwóch wątków, które jednocześnie inkrementują i dekrementują jedną zmienną. Po wykonaniu określonej liczby iteracji wątek kończy zadanie.

Implementacja

Stworzono klasę, która opakowuje zmienną int, umożliwiając dzielenie jej przez wątki:

```
public class IntHolder
{
    public Integer value;

    IntHolder(Integer value) {
        this.value = value;
    }

    @Override
    public String toString() {
        return String.valueOf(value);
    }
}
```

Stworzono również klasę implementującą Runnable, która przyjmuje naszego opakowanego inta i zamuje się inkrementacją / dekrementacją.

```
public class IncrementRunnable implements Runnable{
    int delta;
    int reps;
    IntHolder holder;

    public IncrementRunnable(int delta, int reps, IntHolder holder) {
        this.delta = delta;
        this.reps = reps;
        this.holder = holder;
    }

    @Override
    public void run() {
        for(int i = 0; i < reps; i++){
            holder.value += delta;
        }
    }
}
```

Tworzymy dwa wątki w main i przekazujemy do nich tę samą instancję holdera, uruchamiamy operacje w pętli 10000 razy i zapisujemy wyniki do pliku:

```
public class Main {

    public static void main(String[] args) {
        try{
            FileWriter output = new FileWriter("out.txt");
```

```

        for(int i = 0; i < 10000; i++) {
            output.write(run() + "\r\n");
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

private static String run() throws InterruptedException{
    IntHolder holder = new IntHolder(0);

    Thread incrementer = new Thread(new IncrementRunnable(1, 1000000, holder));
    Thread decrementer = new Thread(new IncrementRunnable(-1, 1000000, holder));

    incrementer.start();
    decrementer.start();

    incrementer.join();
    decrementer.join();

    return holder.toString();
}
}

```

Wyniki

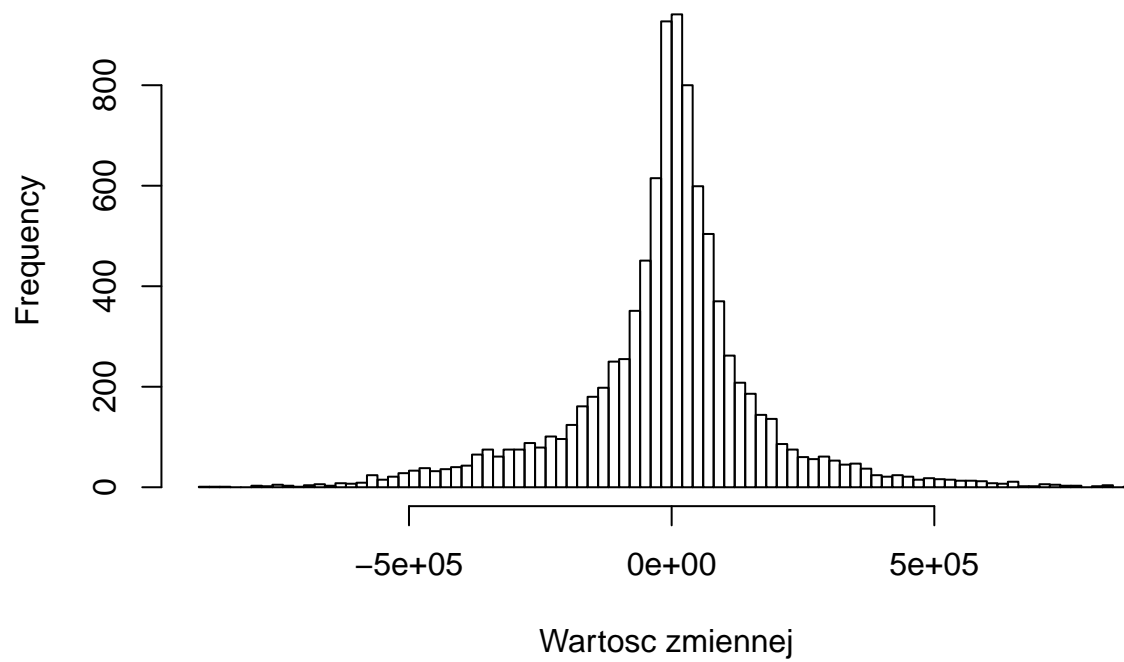
Przykładowy uzyskany plik przedstawiono w postaci histogramu częstości występowania wartości naszej zmiennej:

```

data <- read.table("out.txt", header=F)
hist(data[,1], 100, xlab="Wartość zmiennej", main="Zadanie 1")

```

Zadanie 1



Wnioski

Mamy do czynienia z sytuacją tzw. race condition - wątki ścigają się o dostęp do zmiennej.

W tym konkretnym przypadku wartości najczęściej oscylują wokół zera, a im bardziej oddalamy się od niego, tym rzadziej występują te wyniki.