

# Laboratorium 01: Miłe Złego Początki

Maciej J. Mrowiński

5 grudnia 2018

Zestaw do rozwiązania na pierwszych zajęciach laboratoryjnych. Niezbędna do tego będzie podstawowa wiedza na temat wątków i umiejętność korzystania z jakiegoś programu do robienia wykresów<sup>1</sup>.

## Problemy

- **Problem 1:** Sprawdź, ile jest procesorów w komputerze, z którego korzystasz<sup>2</sup>.
- **Problem 2:** Napisz program, który tworzy nowy wątek wyświetlający "Hello World!" i kończący na tym swoją pracę.
- **Problem 3:** Napisz program, który sumuje sekwencyjnie<sup>3</sup> logarytmy elementów w zadanej przez użytkownika tablicy double o dowolnej długości. Sprawdź, jak zależy czas<sup>4</sup> potrzebny na sumowanie tych elementów od długości tablicy<sup>5</sup>. Pamiętaj, że czas należy uśrednić a liczba prób ma być jednym z parametrów programu. Wyniki zapisz do pliku i wyświetl na wykresie.
- **Problem 4:** To samo co **Problem 3**, ale wielowątkowo. Napisz program, który tworzy tyle wątków, ile jest procesorów w maszynie. Każdy wątek ma sumować logarytmy względnie równych części tablicy. Wyniki muszą potem być oczywiście w jakiś sposób konsolidowane. Wyznacz średni czas niezbędny do wykonania sumowań<sup>6</sup> w funkcji długości tablicy. Wyniki zapisz do pliku i nanieś na stworzony podczas rozwiązywania poprzedniego problemu wykres, porównaj i wyciągnij wnioski.
- **Problem 5:** To samo, co poprzednio - program sumujący logarytmy elementów w tablicy - ale wykorzystujący stałą pulę wątków o tylu wątkach, ile jest procesorów w maszynie<sup>7</sup>. Wyniki znowu uśrednić, nanieść na wykres, porównać i wyciągnąć wnioski. Należy pamiętać o tym, że pula ma być stworzona tylko **raz** i ta sama pula ma być wykorzystywana podczas uśredniania<sup>8</sup>.
- **Problem 6:** Jeszcze raz sumowanie, tylko znowu trochę inaczej. Program ma tworzyć tyle wątków, ile jest procesorów w maszynie. Wątki mają się ściągać w sumowaniu logarytmów elementów tablicy - każdy wątek trzyma swoją lokalną sumę i dodaje do niej po kolei logarytmy kolejnych elementów tablicy, o ile żaden inny wątek już tego nie zrobił. Wyniki znowu na wykres, porównanie i wnioski.

<sup>1</sup> Jeżeli ktoś nie zna jeszcze żadnego programu do robienia wykresów, to na początek najprostszy chyba będzie `gnuplot`. Są również dostępne biblioteki dla Pythona. Warto też rozważyć, jeżeli ktoś tylko będzie zajmować się analizą danych, naukę środowiska `R`.

<sup>2</sup> Hint: Runtime.

```
getRuntime().availableProcessors()
```

<sup>3</sup> Czytaj: po kolei.

<sup>4</sup> Hint: `System.nanoTime()`

<sup>5</sup> Dla uproszczenia przyjmijmy, że długości testowanych tablic są kolejnymi potęgami 2.

<sup>6</sup> Ten czas ma uwzględniać wszystkie niezbędne operacje, w tym stworzenie wątków. Pamiętaj o uśrednianiu wyników.

<sup>7</sup> **Dokładnie** analogia tego, co było w poprzednim problemie - do puli za każdym razem ma trafiać tyle zadań do wykonania, co wątków w puli.

<sup>8</sup> Hint: `CountDownLatch`

- **Problem 7:** To samo, co w **Problem 6**, tylko wątki nie konkurują ze sobą podczas sumowania logarytmów pojedynczych elementów tablicy, ale następujących po sobie bloków o zadanej długości w tablicy. Tego już nie nanosicie na poprzedni wykres, ale dla zadanej długości tablicy<sup>9</sup> robicie wykres czasu wykonania w funkcji długości bloku<sup>10</sup>. Na wykres należy też nanieść linie odpowiadające czasowi niezbędnemu na sekwencyjne (**Problem 3**) i proste wielowątkowe (**Problem 4**) wykonanie analogicznego sumowania. Porównanie i wnioski.
- **Problem 8:** Potencjalnie. Na razie tajemnica, będzie wynikać bezpośrednio z potrzeby wyjaśnienia wyników poprzedniego problemu.

<sup>9</sup> Tablica ma być wystarczająco długa, a słowo "wystarczająco" stanie się jasne po wykonaniu poprzednich problemów.

<sup>10</sup> Znowu dla wygody niech to będą potęgi 2.