







## Zintegrowany Program Rozwoju Akademii Górniczo-Hutniczej w Krakowie Nr umowy: POWR.03.05.00-00-Z307/17

## Instrukcja do ćwiczeń laboratoryjnych

Nazwa przedmiotu	Teoria współbieżności
Numer ćwiczenia	7
Temat ćwiczenia	Rozwiązanie problemu pięciu filozofów w różnych paradygmatach programowania współbieżnego

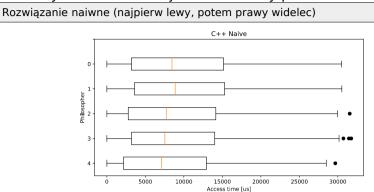
Poziom studiów	I stopień
Kierunek	Informatyka
Forma i tryb studiów	Stacjonarne
Semestr	5

Mateusz Wieczorek *Autor rozwiązania* 

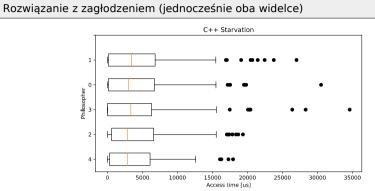


Wydział Informatyki, Elektroniki i Telekomunikacji

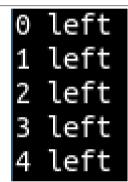
- 1. Kod źródłowy do zadania oraz inne pliki znajdują się w repozytorium pod adresem: <a href="https://github.com/MatWiecz/AGH-TW/tree/master/lab-7">https://github.com/MatWiecz/AGH-TW/tree/master/lab-7</a>
- 2. Rozwiązanie zrealizowałem w dwóch językach programowania. Jednym z nich jest C++ i stanowi on przykład programowania synchronicznego, drugim jest NodeJS i stanowi on przykład programowania asynchronicznego. W każdym z tych języków problem został zaimplementowany na 4 różne sposoby, dane w treści zadania.
- 3. Każdy przypadek testowy został uruchomiony dla 5 filozofów, którzy jedli i myśleli na przemian, przez 5 sekund. Czas jedzenia i myślenia jest losowany z przedziału [0, 10] ms.
- 4. Czasy oczekiwania na jedzenie zostały przedstawione na wykresach pudełkowych.



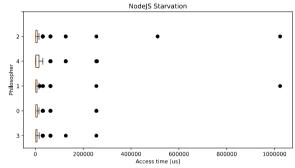
Czasy nie są najlepsze i są porozrzucane po całym przedziale [0, 30] ms, nie ma czasów odstających.



Jest to zdecydowanie najlepsze rozwiązanie dla C++, uzyskano najlepsze czasy, zagłodzenie pojawiało się rzadko, a i tak te wartości odstające nie były większe niż w przypadku naiwnego rozwiązania.



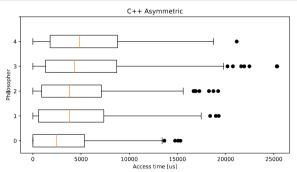
W przypadku NodeJS wszyscy filozofowie wzięli swój lewy widelec i będą w nieskończoność czekać na prawy



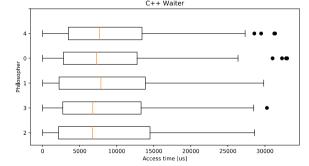
W przypadku NodeJS jest to dobre rozwiązanie z powodu dużej ilości drastycznie odstających wartości, ale pomimo tego średni czas jest bardzo mały. Ciekawością są powtarzające się wartości odstające dla filozofów.

Rozwiązanie asymetryczne (dla parzystego filozofa – prawy, dla nieparzystego – lewy widelec)

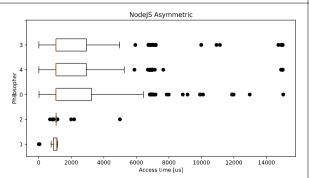




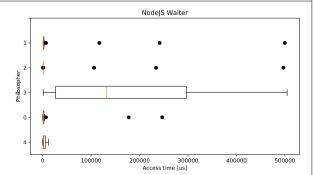
Jest to rozwiązanie jakościowo zbliżone do rozwiązania z zagłodzeniem dla języka C++, natomiast tutaj rozstęp międzykwartylowy jest nieco większy.



Jest to rozwiązanie zbliżone jakościowo do rozwiązania naiwnego w C++. Średnie wartości są niemal identyczne, tak samo położenie kwartyli.



Dla NodeJS jest to najlepsze rozwiązanie ze względu na bardzo małe średnie czasy. Pojawia się dużo wartości odstających, ale jednak one są o wiele razy mniejsze od czasów w innych rozwiązaniach NodeJS.



Jesto to najgorsze rozwiązanie dla NodeJS ze względu na ciągłe zagładzanie jednego filozofa. W przypadku innych filozofów sytuacja wygląda podobnie jak w rozwiązaniu z zagłodzeniem.