Sprawozdanie cw3

Problem ucztujących filozofów Mateusz Nowak

Struktura projektu

W klasach "Arbiter" i "PhilosopherStrarving" są odpowiednio zaimplementowane algorytmy stosujące rozwiązanie z arbitrem oraz z możliwością zagłodzenia, czyli podnoszące dwa widelce na raz.

Klasa "*Test*" i "*BarChart*" to klasy pomocnicze. Pierwsza wykonuje pomiary, a druga używa tych pomiarów do tworzenia wykresów słupkowych.

"Main" klasa uruchomieniowa, uruchamia testy i zapisuje wykresy do plików.

Natomiast część JavaScriptowa znajduje się w pakiecie js (plik phil5.js).

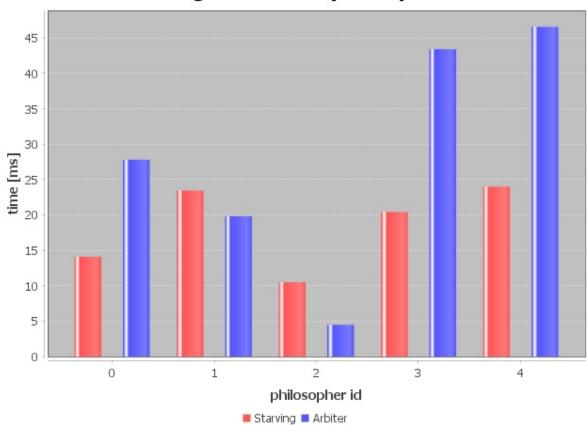
Java

5 filozofów

50 iteracji, aby zjeść cały posiłek

Wykres nr 1

Average times for 5 philosophers

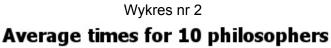


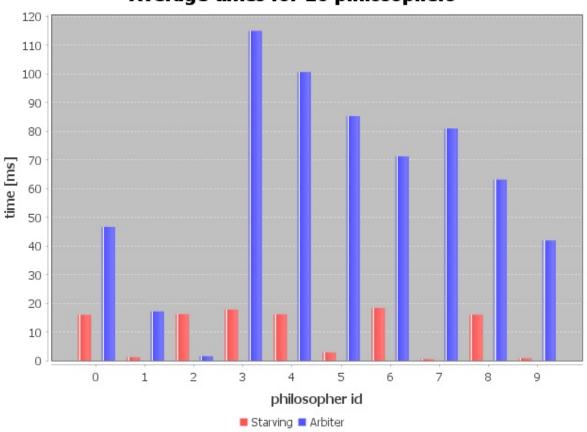
Dla małej liczby filozofów czasy obu rozwiązań wydają się być podobne z lekką korzyścią dla podnoszenia 2 widelców na raz

Java

10 filozofów

50 iteracji, aby zjeść cały posiłek





Gdy zwiększymy liczbę filozofów, czasy dla rozwiązania z arbitrem również się zwiększają, ponieważ kelner musi obsłużyć więcej osób, natomiast dla podnoszenia dwóch widelców na raz czasy pozostają względnie bez zmian, ponieważ filozofa obchodzi tylko czy lewy i prawy widelec jest wolny, czyli patrzy on tylko na swoich sąsiadów, a liczba sąsiadów jest niezmienna.

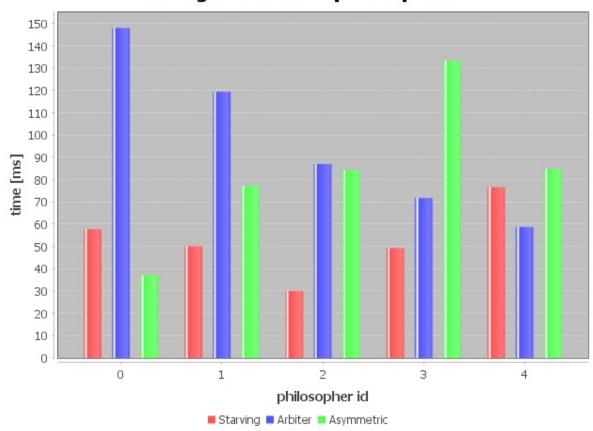
Mimo, że metoda 2 widelców jest teoretycznie najbardziej podatna na zagłodzenia, to w praktyce szansa, na to że filozofowi nie uda się podnieść widelców kilka razy pod rząd jest niewielka.

JavaScript

5 filozofów 50 iteracji, aby zjeść cały posiłek

Wykres nr 3

Average times for 5 philosophers



Dla rozwiązań w javascript otrzymujemy trochę gorsze czasy niż w javie. Jeśli chodzi o porównanie rozwiązań problemu ucztujących filozofów to ponownie otrzymujemy najlepsze wyniki dla jednoczesnego podnoszenia widelców, a najwyższe czasy dla rozwiązania z arbitrem. Dla małej ilości filozofów rozwiązanie asymetryczne wydaje się plasować pośrodku dwóch pozostałych.

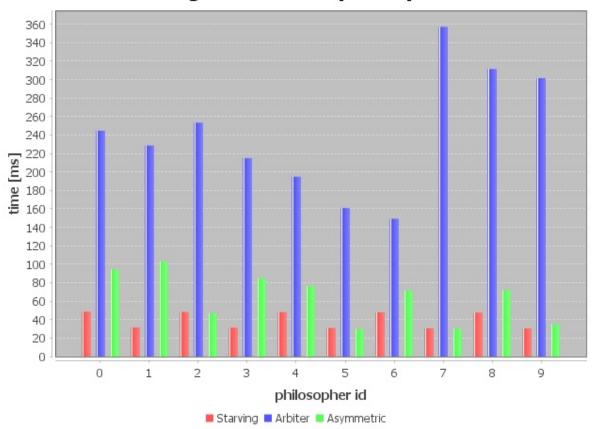
Natomiast jeśli chodzi o rozwiązanie naiwne to przy każdej próbie uruchomienia występowało zakleszczenie.

JavaScript

10 filozofów

50 iteracji, aby zjeść cały posiłek

Wykres nr 4 **Average times for 10 philosophers**



Ponownie przy większej liczbie filozofów, czasy rozwiązania z arbitrem mocno się zwiększyły a dla jednoczesnego podnoszenia jak i dla rozwiązania asymetrycznego pozostały stosunkowo bez zmian.