

TW-Filozofowie

Maksymilian Zawiślak

October 2023

1 Wstęp

Problem pięciu filozofów jest jednym z klasycznych problemów teorii współbieżności. Podstawowe sformułowanie problemu jest następujące :

- N filozofów zasiada przy okrągłym stole
- Pomiędzy sąsiednimi filozofami leży widelec (łącznie jest N widelców)
- Każdy filozof działa ciągle według schematu „myślenie - jedzenie - myślenie - jedzenie - ...”. Każdy z etapów (myślenie i jedzenie) jest skończony.
- Aby zjeść, filozof musi podnieść oba sąsiadujące widelce

Jest wiele rozwiązań zadanego problemu. Do rozwiązywania problemu został wykorzystany język programowania Java wykorzystaniem mechanizmów takich jak: `ReentrantLock`, `Synchronized` oraz `Semaphore`. Aby zapobiec sytuacji, w której jeden z filozofów ciągle podnosi swoje sąsiadujące widelce po odłożeniu ich, filozof jest uśpiony na okres 10 milisekund przed próbą ponownego podniesienia widelców. Symulacja działa przez 5 sekund, potem wszystkie wątki są przerywane. Warianty rozwiązania będą testowane dla $N = 5$ oraz $N = 20$.

2 Rozwiązania problemu

2.1 Wariant 1 - rozwiązanie naiwne (z możliwością blokady)

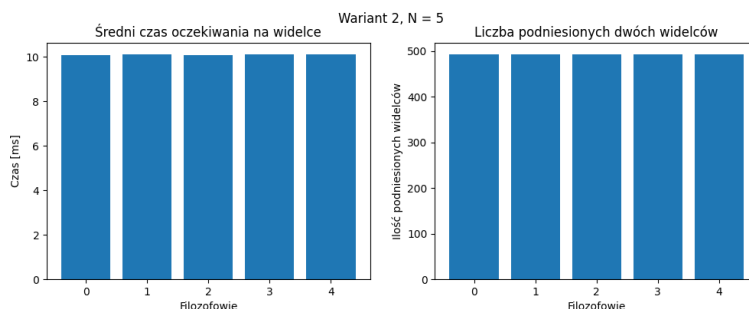
Każdy filozof czeka, aż wolny będzie lewy widelec, a następnie go podnosi (zajmuje), następnie podobnie postępuje z prawym widelcem.

To podejście prowadzi niemal natychmiast do zakleszczenia, niezależnie od liczby filozofów (czy to $N = 5$, czy $N = 20$). Każdy filozof podnosi lewy widelec i nie jest w stanie podnieść prawego, co powoduje zakleszczenie w procesie. Przy implementacji został wykorzystany mechanizm `Synchronized`.

2.2 Wariant 2 - rozwiązanie z możliwością zagłodzenia

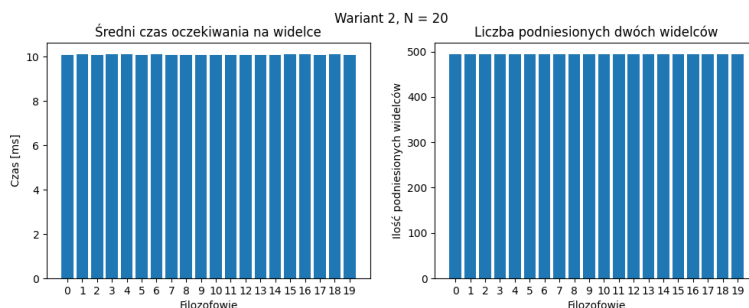
Każdy filozof sprawdza czy oba sąsiednie widelce są wolne i dopiero wtedy zajmuje je jednocześnie. Rozwiązanie to jest wolne od blokady, jednak w przypadku, gdy zawsze któryś z sąsiadów będzie zajęty jedzeniem, nastąpi zagłodzenie, gdyż oba widelce nigdy nie będą wolne.

2.2.1 $N = 5$



Rysunek 1

2.2.2 $N = 20$



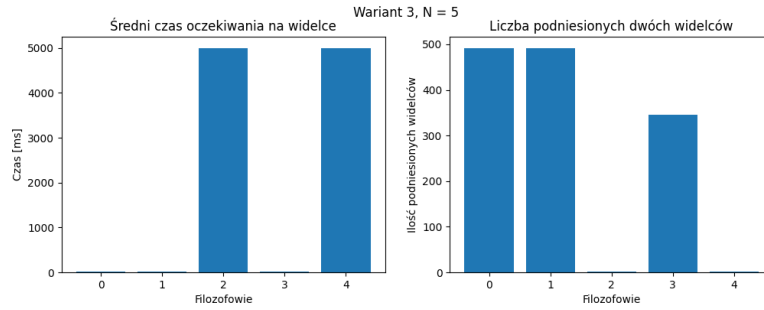
Rysunek 2

Dla testowanych N oraz przy wykorzystaniu uśpienia filozofa na 10 milisekund, nie udało się zaobserwować zagłodzenia żadnego z filozofów. Przy implementacji został wykorzystany mechanizm ReentrantLock.

2.3 Wariant 3 - rozwiązanie asymetryczne

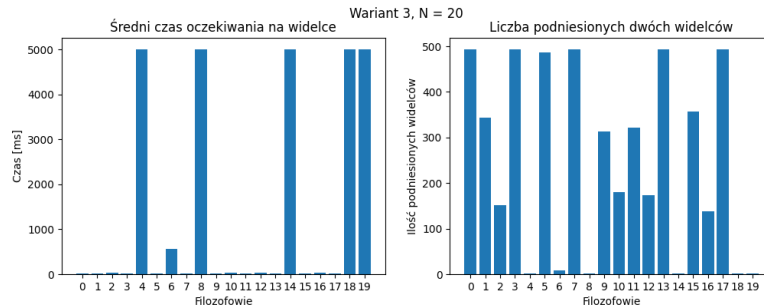
Filozofowie są ponumerowani. Filozof z parzystym numerem najpierw podnosi prawy widelec, filozof z nieparzystym numerem najpierw podnosi lewy widelec.

2.3.1 $N = 5$



Rysunek 3

2.3.2 $N = 20$



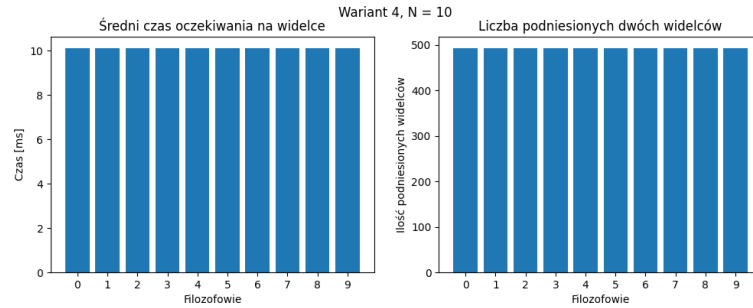
Rysunek 4

Na wykresach dobrze widać że w tym przypadku zachodzi do zagłódzenia, niektórych z filozofów. Dla przypadku $N = 5$, filozofowie 2 oraz 4 mają zdecydowanie większy średni czas oczekiwania na widelce oraz podnoszą widelce rzadziej. Analogiczną sytuację można zaobserwować dla $N = 20$. Przy implementacji został wykorzystany mechanizm Synchronized.

2.4 Wariant 4 - rozwiązanie stochastyczne

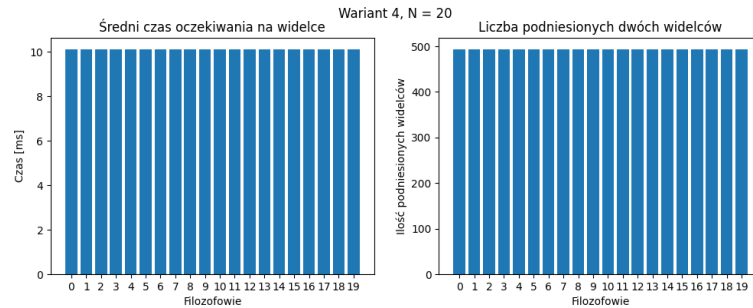
Każdy filozof rzuca monetą tuż przed podniesieniem widelców i w ten sposób decyduje, który najpierw podnieść - lewy czy prawy.

2.4.1 $N = 10$



Rysunek 5

2.4.2 $N = 20$



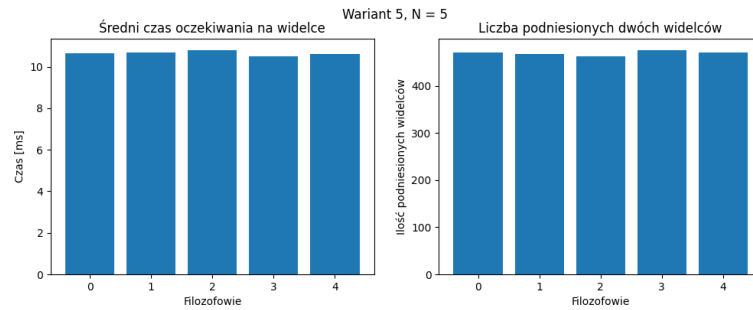
Rysunek 6

W przypadku $N = 5$ notorycznie dochodziło do zakleszczenia. Dopiero wykorzystanie $N = 10$ niepowodowało takiego problemu. Dla $N \geq 10$ ten wariant działał poprawnie. Niższe N często kończyło się zakleszczeniem w czasie trwania symulacji. Przy implemetanacji został wykorzystany mechanizm Synchronized oraz Random.

2.5 Wariant 5 - rozwiązanie z arbitrem

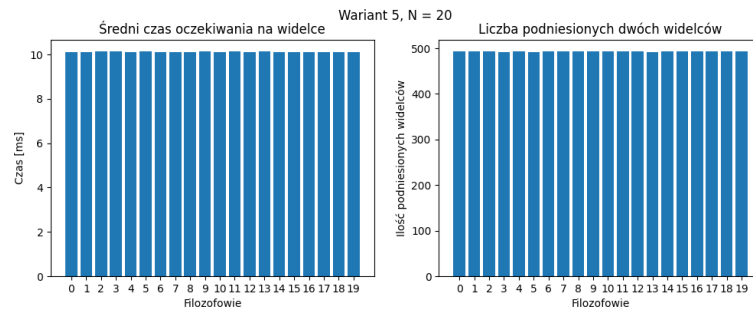
Zewnętrzny arbiter (lokaj, kelner) pilnuje, aby jednocześnie co najwyżej czterech (w ogólnym przypadku $N-1$) filozofów konkurowało o widelce. Każdy podnosi najpierw lewy a potem prawy widelec. Jeśli naraz wszyscy filozofowie będą chcieli jeść, arbiter powstrzymuje jednego z nich aż do czasu, gdy któryś z filozofów skończy jeść.

2.5.1 $N = 5$



Rysunek 7

2.5.2 $N = 20$



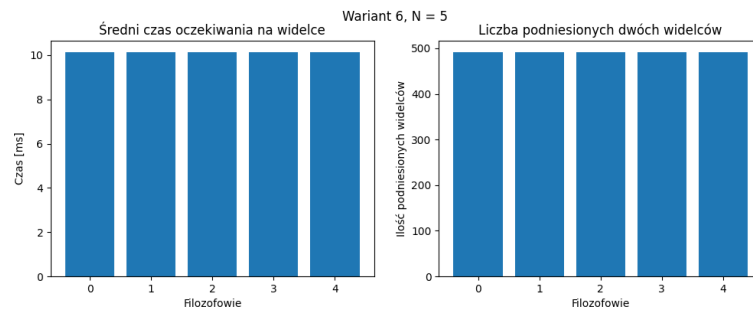
Rysunek 8

Wariant z lokajem również działa poprawnie. Żaden filozof nie jest zagłodzony oraz nie pojawia się zakleszczenie. Przy implemetanacji został wykorzystany mechanizm Synchronized oraz Semaphore.

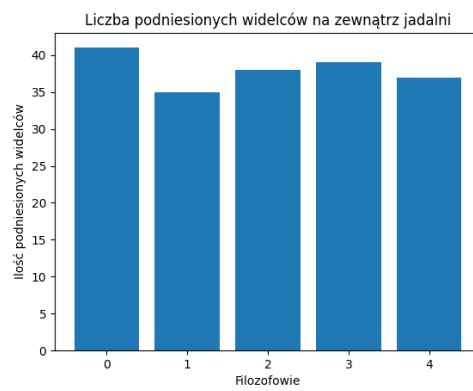
2.6 Wariant 6 - rozwiązanie z jadalnią

Rozwiązanie jest modyfikacją wersji z arbitrem. Filozof, który nie zmieści się w jadalni (czyli arbiter nie pozwolił mu jeść) je "na korytarzu" podnosząc jednocześnie widelce w odwrotnej kolejności (do reszty filozofów w jadalni).

2.6.1 $N = 5$

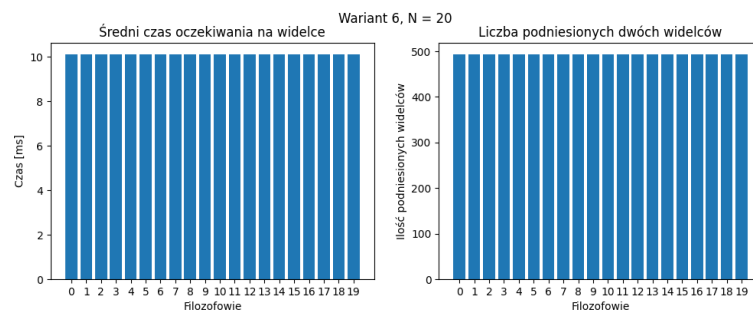


Rysunek 9

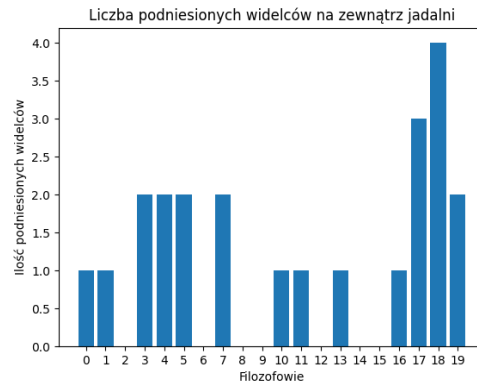


Rysunek 10

2.6.2 $N = 20$



Rysunek 11



Rysunek 12

Ten wariant to ulepszona wersja wcześniejszego wariantu. Nie zmusza jednego z filozofów do czekania aż zwolni się dla niego miejsce w semaforze. Przy mniejszej wartości N częściej dochodziło do "jedzenia na zewnątrz jadali". Również działa poprawnie. Przy implemetanacji został wykorzystany mechanizm Synchronized oraz Semaphore.

3 Wnioski

W większości przypadków średni czas oczekiwania na widelce często był równy 10 milisekundom (czas, na który był usypiany filozof (wątek) po udanym jedzeniu). Mimo wykorzystania innych pomysłów oraz innym mechanizmów zapewniających odpowiednie podnoszenie widelców, średnie czasy oczekiwania są bardzo porównywalne. Jedynie w wariacie 3 z rozwiązaniem asymetrycznym, gdzie dochodziło do zagłodzenia filozofów, nie można poczynić takich obserwacji. Trzeba zachować ostrożność w przypadku wariantu 4 z podejściem stochastycznym, ponieważ przy niskich wartościach parametru N istnieje ryzyko, że symulacja się zakleszczy.