

Zaawansowane C++

Lista 6: `std::thread` , static thread pool

Zadanie 1 `std::thread`

Stwórz duży `std::vector<int>` (np. 1000 elementowy) losowych liczb całkowitych. Za pomocą dwóch wątków z klasy `std::thread` oras funkcji `std::sort()` posortuj w pierwszym wątku pierwszą połowę wektora, a w drugim wątku drugą połowę wektora. Czy taki wektor będzie posortowany w całości?

Zadanie 2 `static thread pool`

Napisz klasę `thread_pool`, która będzie zawierać listę zadań do wykonania - obiekty klasy `std::function<double()>` - oraz zestaw wątków procesujących, które będą te zadania pobierać z listy i wykonywać. Konstruktor tej klasy powinien wystartować zadaną w argumencie liczbę wątków procesujących. Klasa ma mieć trzy publiczne metody:

- `void add_task(std::function<double()> task)` Dodająca nowe zadanie do listy
- `double average()` Zwracająca wartość średnią wyników wykonanych zadań
- `void stop()` Zatrzymująca procesowanie, powinna poczekać na zakończenie wszystkich wątków

Należy zapewnić, żeby wszystkie zadania dodane do listy przed wykonaniem metody `stop()` zostały wykonane przez wątki procesujące. W trakcie wykonywania zadania z listy wątek procesujący nie powinien blokować dostępu do pozostałych zadań innym wątkom.

Materialy pomocnicze:

- `std::threads` https://www.youtube.com/watch?v=wXBcwHwIt_I&t=595s
- threading introduction series https://www.youtube.com/watch?v=TPVH_coGAQs&list=PLk6CEY9XxSIAeK-EAh3hB4fgNvYkYmghp
- Thread pool example
<https://www.youtube.com/watch?v=eWTGtp3HXiw&t=132s>
- `std::condition_variable`
https://www.youtube.com/watch?v=eh_9zUNmTig
- Uzupełnienia ogólne cppcon17
https://www.youtube.com/watch?v=PNRju6_yn3os
- `std::async` elementary (do wiedzy ogólnej)
<https://www.youtube.com/watch?v=ehMg6zvXuMY&t=378s>
- `std::async` advanced (do wiedzy ogólnej)
<https://www.youtube.com/watch?v=5HWCsmE9DrE&t=865s>