

Zaawansowane C++

Lista 7: `promise-future`, `std::allocator<T>`

Zadanie 1 `std::promise` `std::future`

Napisz funkcję liczącą iloczyn skalarny dwóch wektorów przekazanych jako `std::vector<double>`. Jako trzeci argument funkcja ma przyjmować obiekt typu `std::promise`, który zostanie użyty do zwrócenia wyniku lub wyjątku w przypadku przekazania wektorów pustych lub różniących się rozmiarem.

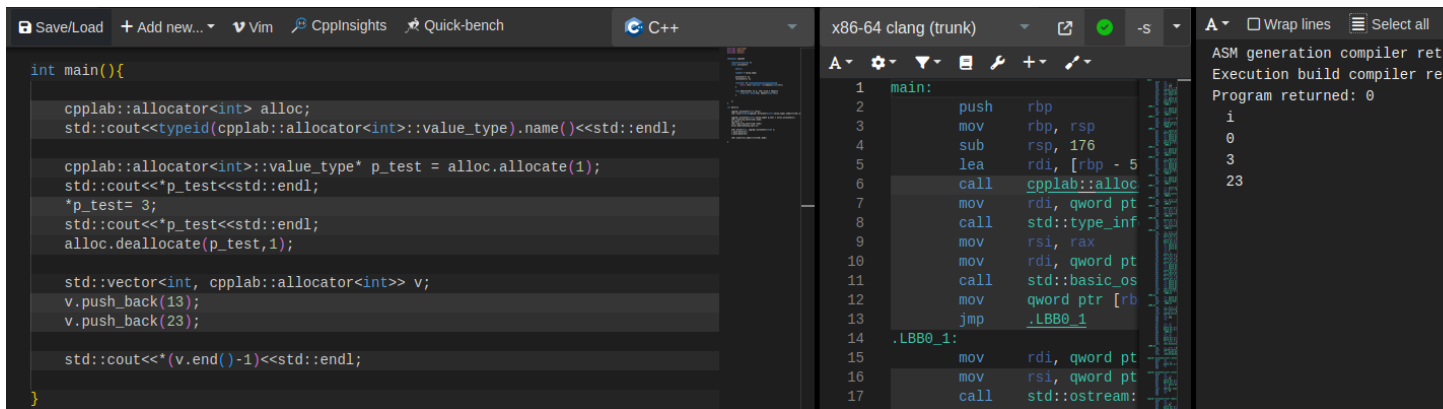
Funkcja `main()` powinna uruchomić 10 wątków (obiekty typu `std::thread`), które policzą iloczyny skalarne przy użyciu powyższej funkcji. Następnie poczekać na wyniki obliczeń (używając mechanizmu `future`) i wypisać na ekran ich sumę.

Zadanie 2 `cpplab::allocator<T>`

Napisz szablon klasy `cpplab::allocator<T>` służącej do zarządzania pamięcią na sterzie. Domyślny obiekt tej klasy powinien być bezstanowy. Dodatkowo klasa powinna zawierać następujące składniki

- `value_type`, przechowującą informację o alokowanym typie `T`
- Metodę `T* allocate(std::size_t n)` alokującą na sterzie za pomocą operatora `placement new` pamięć na przechowanie `n` elementów typu `T`
- Metodę `void deallocate(T* p, std::size_t n)` zwalnającą pamięć wspazaną przez wskaźnik `p` za pomocą operatora `placement delete()`

Przetestuj klasę zgodnie z kodem z rysunku poniżej.



Rysunek 1: Przykład użycia alokatora (zad. 2).

Materiały pomocnicze:

- `std::promise` `std::future`
<https://www.youtube.com/watch?v=XDZkyQVsbDY&t=307s>
- A quick start from Jason Turner
<https://www.youtube.com/watch?v=iCL6GYoi1RU>
with a correction
<https://www.youtube.com/watch?v=ZgN1-i095QY>
- niezwiązana ciekawostka od Jasona
<https://www.youtube.com/watch?v=86Pa973BW4Y>