

Домашнее задание

1. Напишите многопоточную программу для подсчёта количества слов в тексте

- Используйте подход «производитель-потребитель» /*producer-consumer*/.
- Используйте классы стандартной библиотеки: `queue`, `thread`, `mutex`, `condition_variable`

2. Напишите параллельную программу для вычисления скалярного произведения двух векторов большого размера:

```
double scalar_product(const double* v1, const double* v2, int n) {  
    double result = 0.0;  
    for(int i = 0; i < n; ++i) result += v1[i] * v2[i];  
    return result;  
}
```

Используйте следующую функцию для подсчёта числа слов в строке

```
#include <string>
```

```
size_t countup_words( const std::string& str ) {  
    std::string seps = "\\t\\n !.?%^+-*/=() []{} , ; : - # $ & ` ~";  
    size_t count = 0;  
    bool inword = false;  
    for( auto it = str.cbegin(); it != str.cend(); ++it ) {  
        if( seps.find( *it ) == std::string::npos )  
            inword = true;  
        else if( inword ) {  
            inword = false;  
            count++;  
        }  
    }  
    return count;  
}
```

Домашнее задание (продолжение)

3. Перепишите данные программы с использованием шаблонов алгоритмов из Intel® Threading Building Blocks
4. Сравните производительность версий (например, с помощью `tbb::tick_count`)

```
using namespace tbb;  
tick_count t0 = tick_count::now();  
// ...action being timed...  
tick_count t1 = tick_count::now();  
printf("time span = %g seconds\n", (t1-t0).seconds());
```