FeaturesBusinessExplorePricing

This repository

Search

Sign in or Sign up

wichtounet / articles

Watch12

Star72

Fork35

<> Code

🕒 Issues 0

🔗 Pull requests 0

📁 Projects 0

📈 Pulse


📊 Graphs

Branch: master ▾

articles / src / threads / part4 / AtomicCounter.cpp

Find file

Copy path

wichtounet Set the code to MIT license

393b5ec on 17 Sep 2014


1 contributor


51 lines (39 sloc) 1.01 KB

Raw

Blame

History





```
1 //=====
2 // Copyright (c) 2014 Baptiste Wicht
3 // Distributed under the terms of the MIT License.
4 // (See accompanying file LICENSE or copy at
5 // http://opensource.org/licenses/MIT)
6 //=====
7
8 #include <thread>
9 #include <atomic>
10 #include <iostream>
11 #include <vector>
12
13 struct AtomicCounter {
14     std::atomic<int> value;
15
16     AtomicCounter() : value(0) {}
17
18     void increment(){
19         ++value;
20     }
21
22     void decrement(){
23         --value;
24     }
25
26     int get(){
27         return value.load();
28     }
29 };
30
31 int main(){
32     AtomicCounter counter;
33
34     std::vector<std::thread> threads;
35     for(int i = 0; i < 10; ++i){
36         threads.push_back(std::thread([&counter]() {
37             for(int i = 0; i < 500; ++i){
38                 counter.increment();
39             }
40         }));
41     }
42
43     for(auto& thread : threads){
44         thread.join();
45     }
46
47     std::cout << counter.get() << std::endl;
48
49     return 0;
50 }
```

