Домашнее задание 26.11

Доскоч Роман 13 группа 3 курс

#include <iostream>  
#include <vector>  
#include <future>  
#include <chrono>  
  
std::vector<int> linspace(int a, int b, int N) {  
 int h = (b - a) / (N - 1);  
 std::vector<int> xs(N);  
 typename std::vector<int>::iterator x;  
 int val;  
 for (x = xs.begin(), val = a; x != xs.end(); ++x, val += h)  
 \*x = val;  
 return xs;  
}  
  
void linear\_primes(int a, int b) {  
 std::vector<int> v;  
 bool isPrime = true;  
 for (int i = a; i < b; ++i) {  
 for (int j = 2; j < i; ++j)  
 if (i % j == 0) {  
 isPrime = false;  
 break;  
 }  
  
 if (isPrime && i > 1) v.push\_back(i);  
 isPrime = true;  
 }  
}  
  
  
int main() {  
 int a = 1;  
 int b = 1000;  
 /\*linear\*/  
 auto start = std::chrono::high\_resolution\_clock::now();  
 linear\_primes(a, b);  
 auto end = std::chrono::high\_resolution\_clock::now();  
  
 auto res = std::chrono::duration\_cast<std::chrono::microseconds>(end - start).count();  
 std::cout << "Sequential algorithm execution: " << res << "ms" << std::endl;  
  
 /\*async\*/  
 auto range = linspace(a, b, 4);  
  
 start = std::chrono::high\_resolution\_clock::now();  
 for (int i = 1; i < range.size(); ++i)  
 async(std::launch::*async*, linear\_primes, range[i - 1], range[i]).wait();  
 end = std::chrono::high\_resolution\_clock::now();  
  
 res = std::chrono::duration\_cast<std::chrono::microseconds>(end - start).count();  
 std::cout << "Async algorithm execution: " << res << "ms" << std::endl;  
}

Тесты: все в миллисекундах

