**Автоматически распараллелить алгоритм std::uninitialized\_copy**

**выполнить вычислительные эксперименты**

**Код программы:**

#include <iostream>

#include <vector>

#include <random>

#include <algorithm>

#include <execution>

using namespace std;

int main() {

vector<int> d(50000000);

mt19937 gen;

uniform\_int\_distribution<int> dis(0, 500000);

auto rand\_num([=]() mutable { return dis(gen); });

printf("Starting random\n");

generate(execution::par, begin(d), end(d), rand\_num);

vector<int> e = d;

vector<int> buffer1(50000000);

vector<int> buffer2(50000000);

printf("Starting normal algo\n");

unsigned int start\_time = clock();

uninitialized\_copy(execution::seq, begin(d), end(d), begin(buffer1));

unsigned int end\_time = clock();

unsigned int search\_time = end\_time - start\_time;

printf("Total time in milliseconds: %d\n", search\_time);

printf("Starting parallel algo\n");

start\_time = clock();

uninitialized\_copy(execution::par, begin(e), end(e), begin(buffer2));

end\_time = clock();

search\_time = end\_time - start\_time;

printf("Total time in milliseconds: %d\n", search\_time);

return 0;

}

**Результат выполнения программы:**

Starting random

Starting normal algo

Total time in milliseconds: 1097

Starting parallel algo

Total time in milliseconds: 1096