Козунов Алексей, 12 группа

Отчет

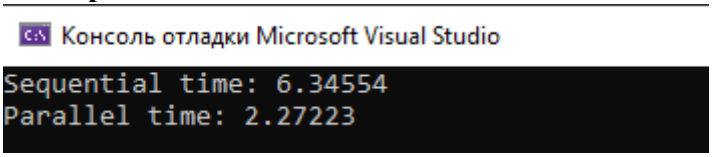
Код программы:

```
#include <iostream>
#include <chrono>
#include <vector>
#include <future>

using namespace std::chrono;

const int interval = 20;
const int number_thread = 4;
const long size = 10000000;

bool is_prime(int x)
{
    for (int i = 2; i * i <= x; ++i)
        if (x % i == 0)
            return false;
    return x == 1;
}

void find_sequential(int a, int b, std::vector<int>* primes)
{
    for (int i = a; i <= b; ++i)
        if (is_prime(i))
            primes->push_back(i);
}

void find_parallel(int size) {
    std::vector<std::future<void>> futures;
    std::vector<std::vector<int>> future_primes;

    int step = size / number_thread;
    auto t1 = system_clock::now();
    for (int i = 0; i < number_thread; i++)
    {
        future_primes.push_back(std::vector<int>());
        futures.push_back(std::async(find_sequential, interval + i * step, interval + (i
+ 1) * step - 1, &future_primes.back()));
    }
    for (auto& future : futures)
        future.wait();
    auto t2 = system_clock::now();
    auto time = 1.0 * (t2 - t1).count() * system_clock::period::num /
system_clock::period::den;

    std::cout << "Parallel time: " << time << std::endl;
}

int main()
{
    std::vector<int> answer;
    auto t1 = system_clock::now();
    find_sequential(interval, interval + size, &answer);
    auto t2 = system_clock::now();

    auto time = 1.0 * (t2 - t1).count() * system_clock::period::num /
system_clock::period::den;

    std::cout << "Sequential time: " << time << std::endl;
```

```
    find_parallel(size);
}
```

Результат:

- **Размер 10000000:**



- **Размер 1000000:**