Козунов Алексей, 12 группа

Отчет

Код программы:

```cpp
#include <iostream>
#include <queue>
#include <tuple>
#include <condition_variable>
#include <thread>
#include <cstdio>
#include <string>
#include <fstream>
using namespace std;
using namespace chrono_literals;
#pragma warning(disable:4996)

queue<string> q;
mutex mut;
condition_variable cv;
bool finished{ false };
ofstream output;
ifstream input;


static void producer() {
        input.open(R"(words.txt)");
        while (!input.eof()) {
                lock_guard<mutex> lk{ mut };
                string s;
                input >> s;
                q.push(s);
                cv.notify_all();
        }
        lock_guard<mutex> lk{ mut };
        finished = true;
        cv.notify_all();
        input.close();
}

static void consumer() {
        output.open(R"(inverted_words.txt)");
        while (!finished) {
                unique_lock<mutex> l{ mut };
                cv.wait(l, [] { return !q.empty() || finished; });
                while (!q.empty()) {
                        string s = q.front();
                        reverse(s.begin(), s.end());
                        output << s << endl;
                        q.pop();
                }
        }
        output.close();
}

void parallel() {
        thread t1{ producer };
        thread t2{ consumer };
        t1.join();
        t2.join();
}
```

Козунов Алексей, 12 группа

```cpp
void sequential() {
      input.open(R"(words.txt)");
      output.open(R"(inverted_words.txt)");
      while (!input.eof()) {
            string s;
            input >> s;
            reverse(s.begin(), s.end());
            output << s << endl;
      }
      input.close();
      output.close();
}

int main() {
      auto start_time = clock();
      parallel();
      auto end_time = clock();
      double parallel_time = (double)(end_time - start_time);
      std::cout << "Parallel time :" << parallel_time << endl;


      start_time = clock();
      sequential();
      end_time = clock();
      double sequent_time = (double)(end_time - start_time);
      std::cout << "Sequent time :" << sequent_time << endl;

      double acceleration_factor = sequent_time / parallel_time;
      std::cout << "Acceleration : " << acceleration_factor << endl;
}
```
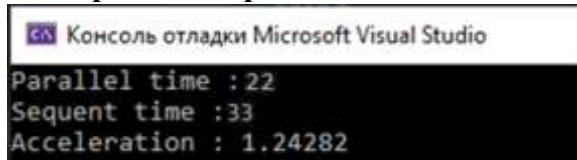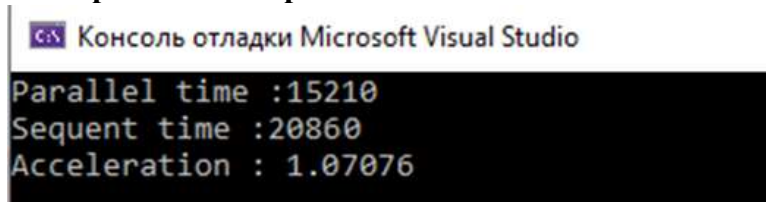
Результат:

- **Размер 10 000 строк:**


```
Консоль отладки Microsoft Visual Studio
Parallel time :22
Sequent time :33
Acceleration : 1.24282
```

- **Размер 1 200 000 строк:**


```
Консоль отладки Microsoft Visual Studio
Parallel time :15210
Sequent time :20860
Acceleration : 1.07076
```