**Доскоч Роман 3 курс 13 группа ТП**

**Задание 19.11.2021**

#include <iostream>

#include <fstream>

#include <queue>

#include <tuple>

#include <condition\_variable>

#include <thread>

#include <algorithm>

#include <chrono>

using namespace std;

using namespace chrono\_literals;

ifstream in("Source.txt");

ofstream out("Result.txt");

queue<string> q;

mutex mut;

condition\_variable cv;

bool finished{ false };

static void producer() {

string line;

while (in >> line) {

{

lock\_guard<mutex> lk{ mut };

q.push(line);

}

cv.notify\_all();

}

{

lock\_guard<mutex> lk{ mut };

finished = true;

}

cv.notify\_all();

}

static void consumer() {

while (!finished) {

unique\_lock<mutex> l{ mut };

cv.wait(l, [] { return !q.empty() || finished; });

string tmp = q.front();

q.pop();

l.unlock();

reverse(tmp.begin(), tmp.end());

out << tmp << "\n";

}

}

void linnear() {

string line;

while (in >> line) {

reverse(line.begin(), line.end());

out << line << "\n";

}

}

int main() {

auto start = std::chrono::high\_resolution\_clock::now();

thread t1{ producer };

thread t2{ consumer };

t1.join(); t2.join();

auto stop = std::chrono::high\_resolution\_clock::now();

cout << std::chrono::duration\_cast<std::chrono::microseconds>(stop - start).count() / 1e6 << endl;

/////////////////////////////////////////////////////////////////////////////

start = std::chrono::high\_resolution\_clock::now();

linnear();

stop = std::chrono::high\_resolution\_clock::now();

cout << std::chrono::duration\_cast<std::chrono::microseconds>(stop - start).count() / 1e6 << endl;

}

Результаты на 100 000 строках



1. Модель производитель – потребитель
2. Линейная.

Выигрыш по времени не получен.