Задание 1

```cpp
#include <iostream>
#include <vector>
#include <fstream>
#include <sstream>
#include <thread>
#include <future>

void count(std::string file_path, std::string word, int& counter) {
    std::ifstream stream(file_path);
    std::vector<std::string> words;
    counter = 0;
    if (stream.is_open()) {
        while (!stream.eof()) {
            std::string temp_;
            stream >> temp_;
            if (temp_ == word) {
                counter++;
            }
        }
    }
}

int main()
{
    std::ifstream fin("files.txt");
    std::vector<std::string> files;

    while (!fin.eof()) {
        std::stringstream buffer;
        std::string file_path;
        fin >> file_path;
        files.push_back(file_path);
    }

    std::vector<int> counters(files.size());

    /*for (auto & file : files)
    {
        std::cout << file << "\n";
    }*/

    std::string word = "std::string";
    std::vector<std::thread> threads(files.size());

    auto start1 = std::chrono::system_clock::now();
    for (int i = 0; i < files.size(); i++)
    {
        threads[i]=std::thread(count, files.at(i), word,
                               std::ref(counters[i]));
    }

    for (int i = 0; i < files.size(); i++)
    {
        threads[i].join();
    }
    auto stop1 = std::chrono::system_clock::now();
    std::cout << "Thread time: " << std::chrono::duration<double>(stop1 - start1).count() << " sec.\n";

    for (size_t i = 0; i < counters.size(); i++)
    {
        std::cout << "counter " << i <<" "<< counters[i] << "\n";
    }
```

```
    counters.clear();
    counters.resize(files.size());

    // async
    auto start2 = std::chrono::system_clock::now();
    for (int i = 0; i < files.size(); i++)
    {
        auto res = std::async(std::launch::async, count, files.at(i), word, std::ref(counters[i]));
    }
    auto stop2 = std::chrono::system_clock::now();
    std::cout << "Async time: " << std::chrono::duration<double>(stop2 - start2).count() << " sec.\n";

    for (size_t i = 0; i < counters.size(); i++)
    {
        std::cout << "counter " << i <<" "<< counters[i] << "\n";
    }
}
```

Результаты измерений:

размерность	Thread (в сек.)	Async (в секундах)
2 файла	0.000148195	0.000121301
4 файла	0.000208526	0.000201303
8 файлов	0.000486216	0.000522642

Получились примерно одинаковые результаты.

Задание 2

```
#include <iostream>
#include <string>
#include <queue>
#include <fstream>
#include <chrono>
#include <thread>

using namespace std;

string word_for_find = "quo";
pthread_mutex_t pmutex;
int n = 20;
int p = 4;
queue<string> fileNames;

void ThreadFunction() {
    while (true)
    {
        pthread_mutex_lock(&pmutex);
        if (fileNames.empty()) {
            pthread_mutex_unlock(&pmutex);
            break;
        }
        string fileName = fileNames.front();
        fileNames.pop();
        pthread_mutex_unlock(&pmutex);
        ifstream fin;
        fin.open(fileName);
        int counter = 0;
        while (!fin.eof())
        {
            string word;
```

```
            fin >> word;
            if (word == word_for_find) {
                counter++;
            }
        }
        fin.close();
        pthread_mutex_lock(&pmutex);
        cout << fileName << ": " << counter << endl;
        pthread_mutex_unlock(&pmutex);
    }
}

int main() {
    pthread_mutex_init(&pmutex, nullptr);

    ifstream fin("files.txt");
    for (int i = 0; i < n; i++) {
        string word;
        fin >> word;
        fileNames.push(word);
    }
    fin.close();

    std::vector<std::thread> threads(fileNames.size());

    auto start = chrono::system_clock::now();

    for (int k = 0; k < p; k++) {
        threads[k] = std::thread(ThreadFunction);
        threads[k].join();
    }

    auto stop = chrono::system_clock::now();

    cout << "Time: " << chrono::duration<double>(stop - start).count() << " sec.\n";

    pthread_attr_destroy(reinterpret_cast<pthread_attr_t *>(&pmutex));
    return 0;
}
```

Результаты измерений:

размерность	C++ Thread (в сек.)	WinApi (в секундах)
2 файла	0.000201456	0.000190235
4 файла	0.000256232	0.000265321
8 файлов	0.000502145	0.000513325

Реализации на базе потоков c++ и WinApi сработали примерно одинаковое время.