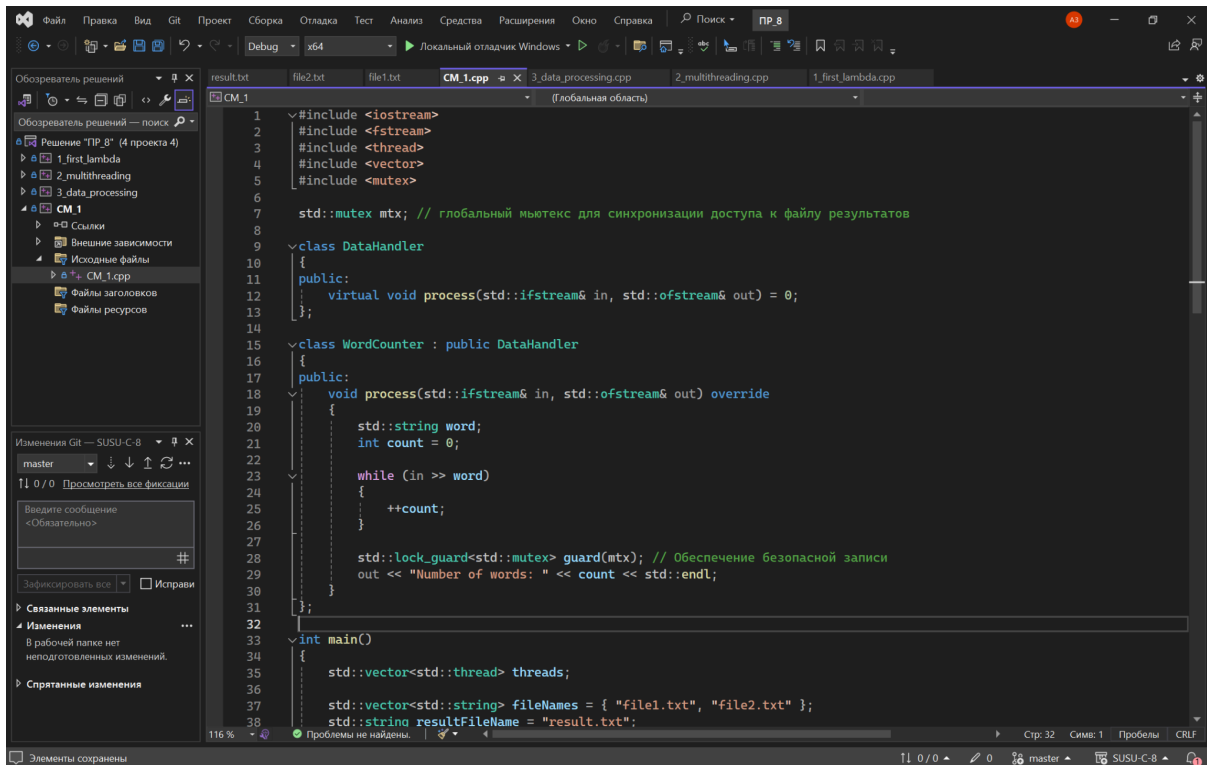
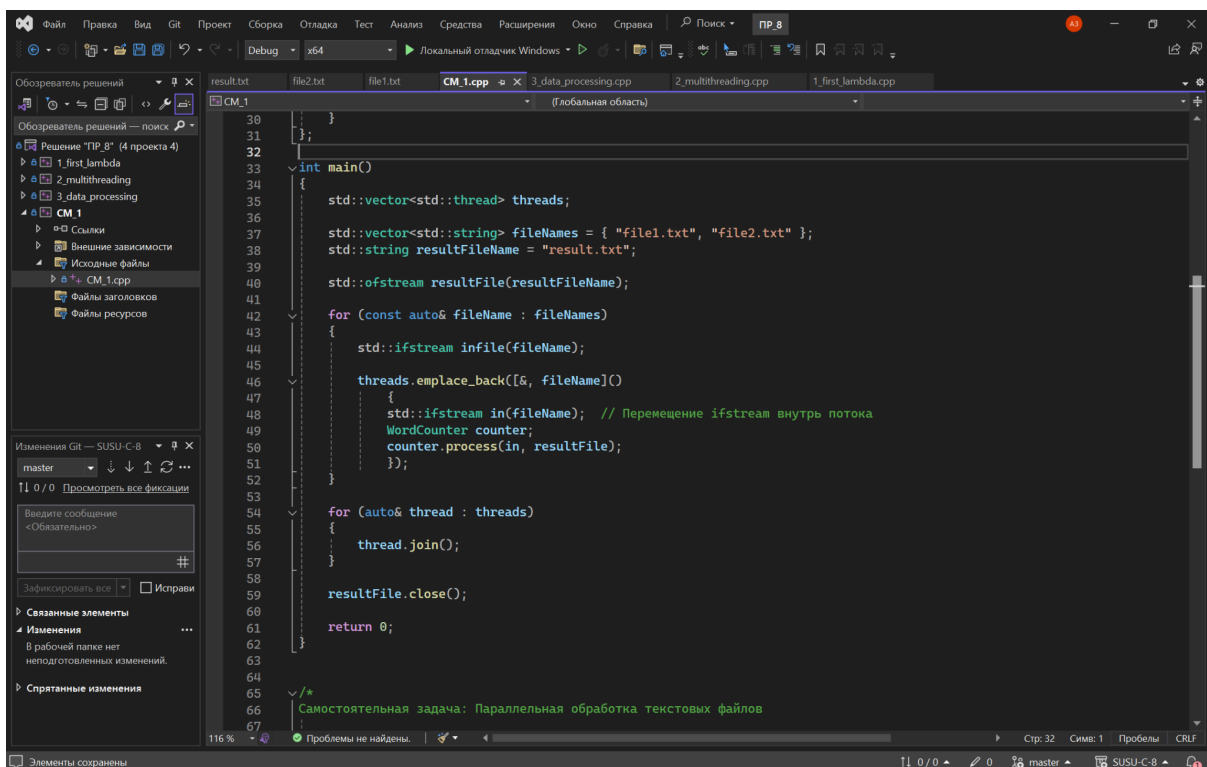


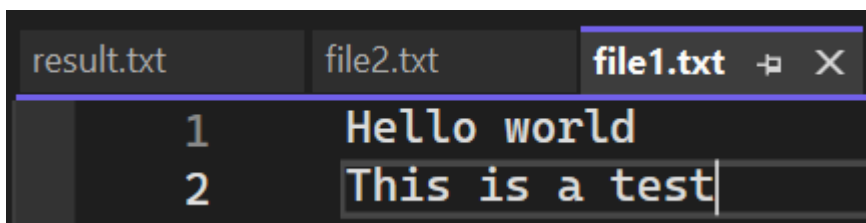
Самостоятельная работа

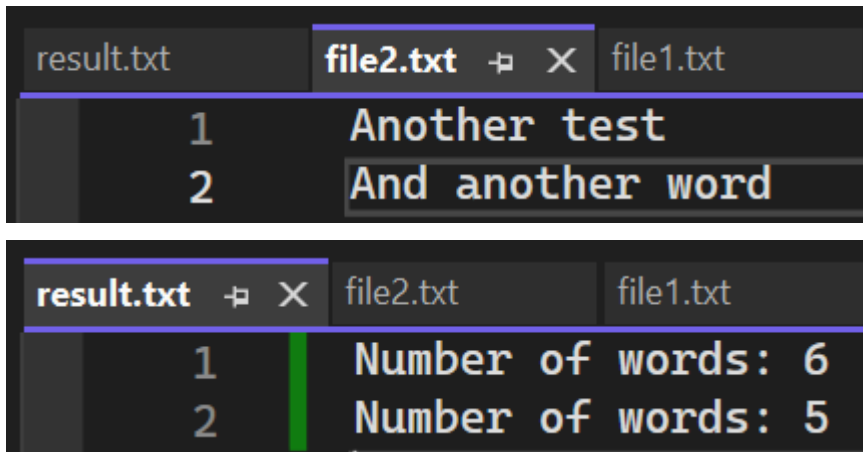


```
1 #include <iostream>
2 #include <fstream>
3 #include <thread>
4 #include <vector>
5 #include <mutex>
6
7 std::mutex mtx; // глобальный мьютекс для синхронизации доступа к файлу результатов
8
9 class DataHandler
10 {
11 public:
12     virtual void process(std::ifstream& in, std::ofstream& out) = 0;
13 };
14
15 class WordCounter : public DataHandler
16 {
17 public:
18     void process(std::ifstream& in, std::ofstream& out) override
19     {
20         std::string word;
21         int count = 0;
22
23         while (in >> word)
24         {
25             ++count;
26         }
27
28         std::lock_guard<std::mutex> guard(mtx); // Обеспечение безопасной записи
29         out << "Number of words: " << count << std::endl;
30     }
31 };
32
33 int main()
34 {
35     std::vector<std::thread> threads;
36
37     std::vector<std::string> fileNames = { "file1.txt", "file2.txt" };
38     std::string resultFileName = "result.txt";
```



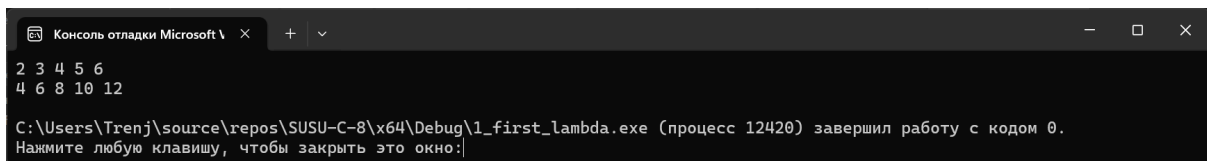
```
30 }
31 };
32
33 int main()
34 {
35     std::vector<std::thread> threads;
36
37     std::vector<std::string> fileNames = { "file1.txt", "file2.txt" };
38     std::string resultFileName = "result.txt";
39
40     std::ofstream resultFile(resultFileName);
41
42     for (const auto& fileName : fileNames)
43     {
44         std::ifstream inFile(fileName);
45
46         threads.emplace_back([& fileName]()
47         {
48             std::ifstream in(fileName); // Перемещение ifstream внутрь потока
49             WordCounter counter;
50             counter.process(in, resultFile);
51         });
52     }
53
54     for (auto& thread : threads)
55     {
56         thread.join();
57     }
58
59     resultFile.close();
60
61     return 0;
62 }
63
64 /*
65 Самостоятельная задача: Параллельная обработка текстовых файлов
66 */
67
```



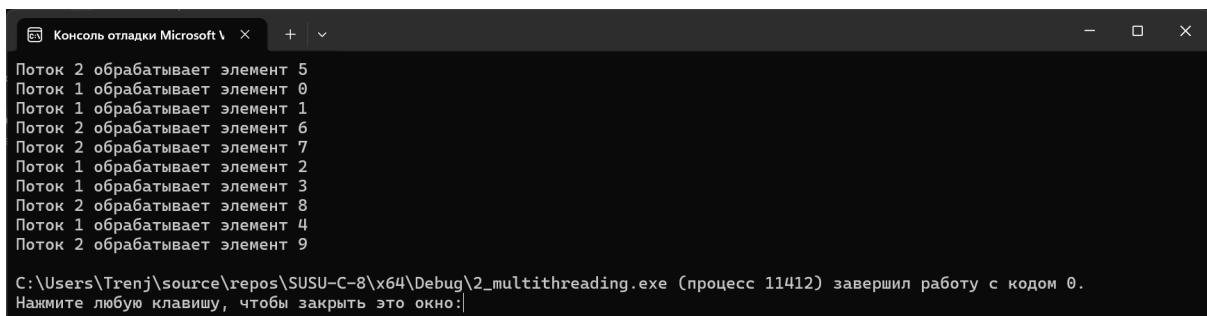


Реализация данных в файле программ

1. Введение в лямбда-выражения с использованием наследования



2. Основы многопоточности



3. Реализация параллельной обработки данных с использованием лямбда-выражений, многопоточности и наследования

