

**Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный университет нефти и газа  
(национальный исследовательский университет)  
имени И. М. Губкина»**

**Кафедра Автоматизированных систем управления**

Отчет по лабораторной работе № 2  
дисциплины *Основы организации операционных систем*

**Процессы и потоки**

Группа: АС-23-04

Студент: Ханеский Ярослав  
Александрович

К.т.н., доцент Фридлянд  
Александр Михайлович

Москва

2024 г.

**Цель работы:** познакомиться с основными принципами программного создания и управления процессами и потоками, получить практические навыки синхронизации потоков.

1. Написать программу, выполняющую расчет некоторого алгоритма в отдельном потоке:

- a. имена файлов с входными данными передаются через параметры командной строки;
- b. для каждого входного файла с данными запускается отдельный поток, т.е. количество потоков определяется количеством файлов с исходными данными;
- c. на выходе – время считывания данных из файла и время работы алгоритма с указанием размера входных данных (используя `std::chrono`);
- d. синхронизация доступа к разделяемым ресурсам осуществляется через `std::mutex`.

2. Скомпилировать и запустить программу в ОС Windows и в ОС Linux.

**Ход работы:**

- 1) Напишем программу, считающую сумму чисел в файле и время работы потока и выводит соответствующие данные:

```

1  #include <iostream>
2  #include <string>
3  #include <fstream>
4  #include <chrono>
5  #include <sstream>
6  #include <filesystem>
7  #include <thread>
8  #include <mutex>
9  #include <vector>
10
11  std::mutex coutMutex;
12
13  void summ(std::string fileName){
14      std::ifstream file;
15      std::stringstream ss;
16      long long int summ = 0;
17      int temp;
18      auto fileSize = std::filesystem::file_size(fileName);
19
20      std::chrono::time_point<std::chrono::system_clock> readingStart, readingEnd;
21      readingStart = std::chrono::system_clock::now();
22      file.open(fileName);
23      ss << file.rdbuf();
24      file.close();
25      readingEnd = std::chrono::system_clock::now();
26      std::chrono::duration<double> readingDuration = readingEnd - readingStart;
27
28      std::chrono::time_point<std::chrono::system_clock> summStart, summEnd;
29      summStart = std::chrono::system_clock::now();
30      while (ss >> temp) summ = summ + temp;
31      summEnd = std::chrono::system_clock::now();
32      std::chrono::duration<double> summDuration = summEnd - summStart;
33
34      std::chrono::duration<double> totalDuration = summEnd - readingStart;
35
36      coutMutex.lock();

```

Рисунок 1. Фрагмент кода №1

```

37      std::cout << fileName << "\n";
38      std::cout << "Size of file = " << fileSize << " Bytes\n";
39      std::cout << "Summ = " << summ << "\n";
40      std::cout << "Time spent reading from the file = " << readingDuration.count() << "s\n";
41      std::cout << "Time spent on summation = " << summDuration.count() << "s\n";
42      std::cout << "Total time = " << totalDuration.count() << "s\n\n";
43      coutMutex.unlock();
44  }
45
46  int main(int argc, char* argv[]){
47      std::vector<std::thread> threads;
48      for (int i = 1; i < argc; i++){
49          threads.push_back(std::thread(summ, argv[i]));
50      }
51      for (int i = 1; i < argc; i++){
52          threads[i-1].join();
53      }
54
55      return 0;
56  }

```

Рисунок 2. Фрагмент кода №2

2) Создадим в папке с программой несколько текстовых файлов и заполним их случайными числами, для этого напишем следующий код:

```
1 #include <iostream>
2 #include <fstream>
3
4 int main(int argc, char* argv[]){
5     std::ofstream file;
6     for (int j = 1; j < argc; j++){
7         file.open(argv[j]);
8         for (int i = 0; i < 1000000; i++){
9             file << std::rand() << " ";
10        }
11        file.close();
12    }
13    return 0;
14 }
```

Рисунок 3. Код программы для создания файлов с псевдослучайными числами

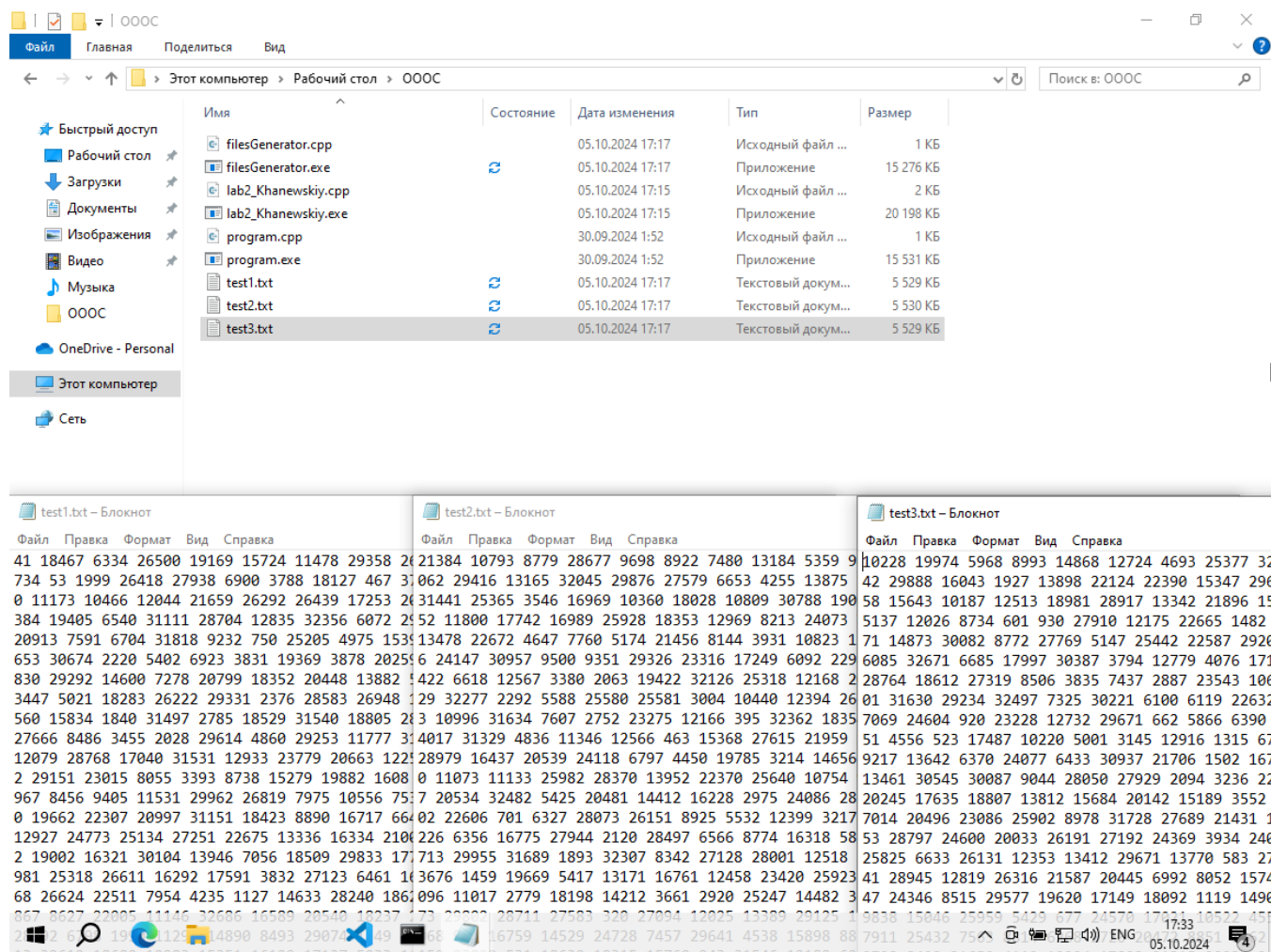


Рисунок 4. Тестовые файлы

3) Запустим программу через командную строку:

```
Командная строка
C:\Users\yaros\OneDrive\Рабочий стол\000C>lab2_Khanewskiy.exe test1.txt test2.txt test3.txt
test3.txt
Size of file = 5661261 Bytes
Summ = 16383699737
Time spent reading from the file = 0.01809s
Time spent on summation = 0.041068s
Total time = 0.059158s

test1.txt
Size of file = 5661529 Bytes
Summ = 16387983044
Time spent reading from the file = 0.018668s
Time spent on summation = 0.040718s
Total time = 0.059386s

test2.txt
Size of file = 5662297 Bytes
Summ = 16402252288
Time spent reading from the file = 0.018679s
Time spent on summation = 0.041119s
Total time = 0.059798s

C:\Users\yaros\OneDrive\Рабочий стол\000C>
```

Рисунок 5. Результат выполнения программы в Windows

```
Командная строка
C:\Users\yaros\OneDrive\Рабочий стол\000C>lab2_Khanewskiy.exe test1.txt test2.txt test3.txt
test1.txt
Size of file = 5661529 Bytes
Summ = 16387983044
Time spent reading from the file = test2.txt
Size of file = 5662297 Bytes
Summ = 16402252288
Time spent reading from the file = test3.txt
Size of file = 5661261 Bytes
Summ = 16383699737
Time spent reading from the file = 0.019415s
Time spent on summation = 0.042368s
Total time = 0.061784s

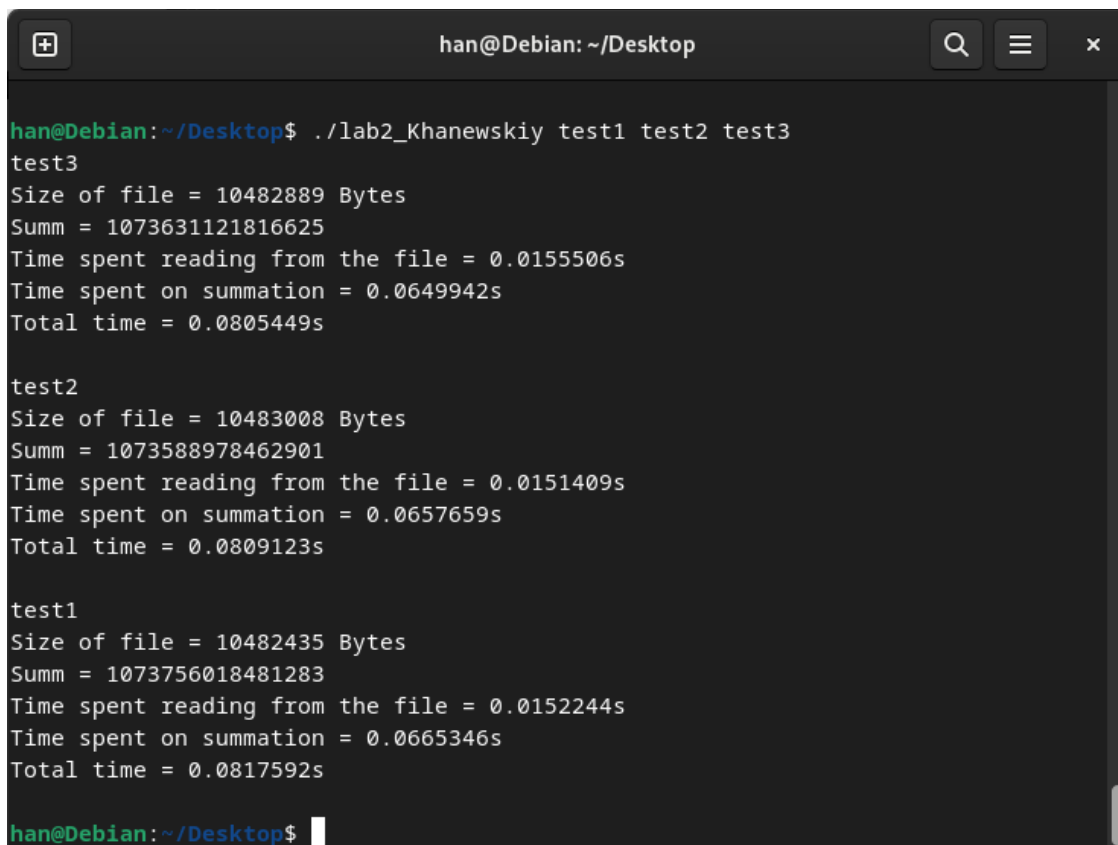
0.019411s
Time spent on summation = 0.041321s
Total time = 0.060732s

0.019491s
Time spent on summation = 0.041458s
Total time = 0.060949s

C:\Users\yaros\OneDrive\Рабочий стол\000C>
```

Рисунок 6. Результат выполнения программы в Windows без использования mutex

4) Та же программа написана в ОС Linux:

A terminal window titled 'han@Debian: ~/Desktop' with search, menu, and close buttons. It shows the execution of a program with three test cases. Each test case displays file size, summation, and timing data.

```
han@Debian:~/Desktop$ ./lab2_Khanewskiy test1 test2 test3
test3
Size of file = 10482889 Bytes
Summ = 1073631121816625
Time spent reading from the file = 0.0155506s
Time spent on summation = 0.0649942s
Total time = 0.0805449s

test2
Size of file = 10483008 Bytes
Summ = 1073588978462901
Time spent reading from the file = 0.0151409s
Time spent on summation = 0.0657659s
Total time = 0.0809123s

test1
Size of file = 10482435 Bytes
Summ = 1073756018481283
Time spent reading from the file = 0.0152244s
Time spent on summation = 0.0665346s
Total time = 0.0817592s

han@Debian:~/Desktop$
```

Рисунок 7. Результат выполнения программы в Linux

Библиотека `chrono` – работа со временем.

Библиотека `thread` – работа с потоками.

Библиотека `mutex` – тип синхронизации потоков.

**Вывод:** в ходе работы изучены основные принципы программного создания и управления процессами и потоками, получены практические навыки синхронизации потоков.