

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра информационных систем управления**

**ЛАБОРАТОРНАЯ РАБОТА № 2
По дисциплине “Операционные системы”**

Возовикова Никиты Александровича
студента 2 курса группы 10
специальности «Компьютерная
Безопасность»
дневной формы получения
высшего образования

Научный руководитель:
старший преподаватель
Безверхий Александр Анатольевич

Минск, 2021

Постановка задачи

Необходимо реализовать два консольных приложения под OS Windows на языке C++. Оба задания (А и Б) необходимо сдавать одновременно. Так же **необходимо** написать краткий отчет по выполняемой работе и сдать его в виде PDF файла. В отчете должен быть описан математический алгоритм, который вы использовали (его суть, условие завершения вычислений, условие отсутствия решения).

Задание А:

Реализовать программу, которая создает вспомогательный поток, выполняющий некоторые вычисления в фоне, и ожидает его завершения. Приоритет вспомогательного потока не должен превосходить normal. Вспомогательный поток внутри себя исполняет функцию, **аргументами** которой являются имена двух текстовых файлов (конфигурационный и выходной). По завершении вычислений необходимо из главного потока вывести в консоль время, затраченное на вычисления.

Вычисление содержит следующие этапы:

- 1) Чтение необходимых параметров задачи из конфигурационного текстового файла.
- 2) Произведение необходимых вычислений (решаемая задача определяется вашим вариантом).
- 3) Сохранение результата в выходной текстовый файл.

Задание Б:

Реализовать программу, работающую по нижеприведенному сценарию:

- 1) Из консоли принимаются два аргумента TaskCount и ThreadCount.
- 2) Основной поток генерирует TaskCount наборов параметров (для задачи из задания А), и сохраняет их в некоторый контейнер (контейнер с параметрами один для всех потоков). При генерации параметров использовать датчик случайных чисел. Алгоритм генерации должен быть такой, что бы решения всех задач не были одинаковыми (например 0 или Nosolution).
- 3) Создается ThreadCount одинаковых вспомогательных потоков, которые берут задания из контейнера, и выполняют их асинхронно (т.е. как только поток выполнил свое текущее задание, он берет следующее и выполняет его. И так, пока все задания не будут выполнены.). Результаты своих вычислений все потоки складывают в один **общий** контейнер.

- 4) После того, как все задания будет вычислены, основной поток сохраняет все результаты попарно (параметры задачи, ответ) в выходной текстовый файл.
- 5) После основным потоком на экран выводится статистика: сколько задач всего решено, сколько не решилось из-за ошибки в процессе вычисления, сколько не имеют решений, сколько задач решил каждый из потоков, сколько времени он на это затратил, минимальное и максимальное время решения задачи, время затраченное на запись данных в файл. Статистика должна быть выведена на экран в удобочитаемом виде.

Важно (для каждого задания):

- 1) В вашей реализации для хранения параметров задачи необходимо создать подходящую структуру, предусмотреть для нее операции ввода/вывода из/в поток.
- 2) Предусмотреть обработку исключений в коде, исполняемом вспомогательным потоком.

Вариант задания № 26

26	Зашифровать заданный текст.	Первый файл - текстовый файл с произвольным текстом, не более 1М символов. Второй файл – корректный ключ для шифрования. В нем содержится две строки символов (один над одним), разделенных пробелами. Шифрование производится заменой верхнего символа нижним.	Зашифрованный файл (с сохранением строчной структуры исходного файла).
----	-----------------------------	---	--

Алгоритм

Для решения поставленной задачи необходимо считать две строки символов из конфигурационного файла, где первая строка содержит список символов, которые необходимо заменить на символы, содержащиеся во второй строке. По окончании считывания символов из конфигурационного файла, каждому символу из верхней строки, ставится в соответствие символ из нижней строки. Каждая пара, поставленных в соответствие символов, сохраняется в соответствующей структуре данных.

При наличии в строке не одиночного символа, а множества символов, не разделённых пробелами, или при не соответствии количества символов в верхней строке и количества символов в нижней строке, алгоритм завершается и выводятся соответствующие ошибки.

При успешном считывании конфигурационного файла и построении структуры данных, считывается текст из входного файла. Далее последовательно считывается каждый символ текста и при совпадении данного символа с “первым” символом из списка пар символов, символ текста заменяется на “второй” символ из пары, иначе символ не изменяется. По окончании считывания символов текста из входного файла, в выходной файл записывается зашифрованный текст.

Листинг программ

Задача А

```
1  #include <fstream>
2  #include <sstream>
3  #include <thread>
4  #include <chrono>
5  #include <iostream>
6  #include <Windows.h>
7  #include <string>
8  #include <exception>
9  #include <vector>
10 #include <List>
11 #include <mutex>
12
13 using namespace std;
14
15 string configPath = "config.txt";
16 string inputPath = "input.txt";
17 string outputPath = "output.txt";
18
19 mutex mute;
20 vector<exception_ptr> threadsExceptions;
21
22 vector<string> lineSplit(string str)
23 {
24     istringstream strStream(str);
25     vector<string> symbols;
26     string sym;
27
28     while (strStream >> sym)
29     {
30         if (sym.length() > 1) {
31             throw new exception("Found invalid character for encoding!");
32         }
33         else
34             symbols.push_back(sym);
35     }
36
37     return symbols;
38 }
39
40 vector<pair<string, string>> getCryptoConfig(string configFile) {
41     ifstream configStream(configFile);
42
43     string oldSymbols;
44     string newSymbols;
45
46     if (!getline(configStream, oldSymbols) || !getline(configStream, newSymbols))
47         throw exception("Something wrong with configure file!");
48
49     vector<string> oldList = lineSplit(oldSymbols);
50     vector<string> newList = lineSplit(newSymbols);
51
52     if (oldList.size() != newList.size())
53         throw exception("Characters for encoding are missing!");
54
55     vector<pair<string, string>> cryptoConfig;
56     for (int i = 0; i < oldList.size(); i++)
57         cryptoConfig.push_back(make_pair(oldList[i], newList[i]));
58
59     configStream.close();
60     return cryptoConfig;
61 }
62
```

```

63 void encryption(string configFile, string inputFile, string outputFile) {
64
65     ifstream inputStream(inputFile);
66     ofstream outputStream(outputFile);
67
68     try {
69         vector<pair<string, string>> cryptoConfig = getCryptoConfig(configFile);
70
71         string strLine;
72
73         while (getline(inputStream, strLine)) {
74             string outline = "";
75
76             for (int i = 0; i < strLine.length(); i++) {
77                 for (int j = 0; j < cryptoConfig.size(); j++)
78                     if (cryptoConfig[j].first[0] == strLine[i])
79                     {
80                         outline += cryptoConfig[j].second;
81                         goto exit;
82                     }
83
84             outline += strLine[i];
85         exit:
86             continue;
87         }
88         outputStream << outline + '\n';
89     }
90
91     catch(const exception& e){
92         lock_guard<mutex> lock(mute);
93         threadsExceptions.push_back(current_exception());
94     }
95
96     inputStream.close();
97     outputStream.close();
98 }
99
100 class Timer {
101 private:
102     chrono::time_point<chrono::high_resolution_clock> start;
103
104 public:
105     Timer() :start(chrono::high_resolution_clock::now()) {
106     }
107
108
109     void resetTimer() {
110         start = chrono::high_resolution_clock::now();
111     }
112
113     double getTime() {
114         return chrono::duration_cast<chrono::duration<double, std::ratio<1>>> (chrono::high_resolution_clock::now() - start).count();
115     }
116 };
117
118
119 int main() {
120
121     Timer one;
122
123     thread encryptThread(encryption, configPath, inputPath, outputPath);
124     SetThreadPriority(encryptThread.native_handle(), THREAD_PRIORITY_BELOW_NORMAL);
125     encryptThread.join();
126
127     cout << "Elapsed time: " << one.getTime() << endl;
128
129     for (auto& e : threadsExceptions)
130     {
131         try
132         {
133             if (e != nullptr)
134                 rethrow_exception(e);
135         }
136         catch (const exception& e)
137         {
138             cout << e.what() << endl;
139         }
140     }
141
142     return 0;
143 }

```

Задача В

```
1 // commit line - debug lines for additional info
2 #include <fstream>
3 #include <sstream>
4 #include <thread>
5 #include <chrono>
6 #include <iostream>
7 #include <Windows.h>
8 #include <string>
9 #include <exception>
10 #include <vector>
11 #include <List>
12 #include <mutex>
13 #include <stack>
14 #include <random>
15
16 using namespace std;
17 void encryption(int);
18
19 class Timer { ... };
20
21 class safetyStack { ... };
22
23 struct xxxThread { ... };
24
25 class safetyVector { ... };
26
27 string outputPath = "output.txt";
28 string commonWords = "a ability able about above accept according acco
29
30 safetyStack dataBase;
31 safetyStack outputBase;
32
33 safetyVector Threads;
34
35 vector<string> lineSplit(string str, bool symbolsCheck) { ... }
36
37 vector<pair<string, string>> getCryptoConfig(string configFile) { ... }
38
39 void encryption(int ID) { ... }
40
41 int random(const int a, const int b) { ... }
42
43 void taskGenerator(int taskCount) { ... }
44
45 void threadsGenerator(int threadCount) { ... }
46
47 string report(double procTime) { ... }
48
49 int main() {
50     Timer one;
51     int taskCount = 0;
52     int threadCount = 0;
53
54     cout << "Task count:\n";
55     cin >> taskCount;
56     cout << "Thread count:\n";
57     cin >> threadCount;
58
59     if (taskCount <= 0 || threadCount <= 0) {
60         cout << "Wrong data entered!" << endl;
61         return 0;
62     }
63 }
```



```

411     one.startTimer();
412     taskGenerator(taskCount);
413     threadsGenerator(threadCount);
414
415     bool aim = true;
416
417     while (aim) {
418         for (int i = 0; i < Threads.size(); i++) {
419             if (Threads[i].getAlive())
420                 break;
421             if (i == Threads.size() - 1)
422                 aim = false;
423         }
424         Sleep(1);
425     }
426
427     cout << report(one.getTime());
428
429     one.resetTimer();
430     ofstream out(outputPath);
431     pair<string, string> currentPair;
432     while (outputBase.pop(currentPair)) {
433         out << "CONFIGURE DATA\n#####\n\n" + currentPair.first + '\n'
434             << "OUTPUT DATA\n#####\n\n" + currentPair.second + '\n';
435     }
436     cout << "Writefile time: " << one.getTime() << " (s)" << endl;
437
438     return 0;
439 }
440

```


Результаты выполнения программы А

Входной файл:

```
1 This is a classic problem, faced by many while in Competitive Coding Competitions ,
2 in Coding Rounds of Technical Companies for placements, etc. We're supposed to take the input
3 in a continuous manner and further "break" the input into a suitable form to be manipulated, mostly as strings.
4
5 After facing the problem quite a number of times on CodeChef, Hackerrank and my College exams,
6 I found a perfect solution to the same, which will be explained in great depth by me here.
7
8 Note: Do not get scared, looking at terms such as Vectors, stringstream, etc. All of them will be explained in understandable language, after the code.
```

Конфигурационный файл:

```
1 a b c d i f g v
2 @ # $ % ^ * = 1
```

Выходной файл:

```
1 This is a classic problem, faced by many while in Competitive Coding Competitions ,
2 in Coding Rounds of Technical Companies for placements, etc. We're supposed to take the input
3 in a continuous manner and further "break" the input into a suitable form to be manipulated, mostly as strings.
4
5 After facing the problem quite a number of times on CodeChef, Hackerrank and my College exams,
6 I found a perfect solution to the same, which will be explained in great depth by me here.
7
8 Note: Do not get scared, looking at terms such as Vectors, stringstream, etc. All of them will be explained in understandable language, after the code.
```

Результаты выполнения программы В

```
Task count:
5
Thread count:
2
=====SUMMARY=====
Total program time: 22.071083 (s)
Total solved problems: 5
Total unsolved problems: 0
Total problems with errors: 0
Maximum solve time: 13.750119 (s)
Minimum solve time: 0.593968 (s)
=====
=====THREAD(0)=====
Total solved problems: 3
Total unsolved problems: 0
Total solve time: 14.631844 (s)
Maximum solve time: 9.264627 (s)
Minimum solve time: 0.593968 (s)
=====
=====THREAD(1)=====
Total solved problems: 2
Total unsolved problems: 0
Total solve time: 21.377136 (s)
Maximum solve time: 13.750119 (s)
Minimum solve time: 7.623608 (s)
=====
Writefile time: 0.0103844 (s)
```