

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное учреждение высшего образования
«Чувашский государственный университет И.Н. Ульянова»
Факультет информатики и вычислительной техники
Кафедра вычислительной техники

Системное программирование
Лабораторная работа 1
«Копирование файлов в Unix системе»

Выполнил:

Студент группы ИВТ-41-20
Галкин Д.С.

Проверил:

Яковлев С.В.

Цель работы:

Написать, ввести, построить и выполнить три программы копирования файлов **срС**, **срW**, **срCF** с указанной строкой запуска.

1. срС: копирование файлов с использованием стандартной библиотеки C, которая поддерживает объекты потокового ввода-вывода FILE. (копирует файл1 в файл2)
2. Выяснить, как поведет себя программа срС, если входной файл открыть в двоичном режиме, а выходной - в текстовом, и наоборот.
3. Измерить быстродействие программ копирования на файлах размером 10K, 100K, 1M, 10M, 100M, возможно, с применением функции DWORD GetTickCount(VOID)

Полный текст программы:

cpC_files.h

```
//  
// Created by Dmitry Galkin on 02.02.2024.  
//  
  
#ifndef CPC_FILES_H  
#define CPC_FILES_H  
#include <string>  
#include <fstream>  
  
namespace Lab1 {  
    class File {  
    public:  
        std::string read_file();  
        void write_file(std::string data);  
        std::string read_file(std::string nameFile);  
        void write_file(std::string data, std::string nameFile);  
        static long long getCurrentTimeMillis();  
    private:  
        const std::string IN_PATH_FILE = "../Lab1/resources/";  
        const std::string OUT_PATH_FILE = "../Lab1/resources/";  
    };  
}  
  
#endif //CPC_FILES_H
```

```
//
// Created by Dmitry Galkin on 02.02.2024.
//

#include "cpC_files.h"
#include <vector>
#include <sys/time.h>
using namespace Lab1;
using namespace std;

// Функция для измерения текущего времени в миллисекундах
long long File::getCurrentTimeMillis() {
    timeval tv{};
    gettimeofday(&tv, nullptr);
    return (tv.tv_sec * 1000LL) + (tv.tv_usec / 1000);
}

string File::read_file() {

    string line;
    string data;
    ifstream in(IN_PATH_FILE, ios::binary);

    if (in.is_open()) {
        while (getline(in, line)) {
            data += line;
        }
    }

    in.close();

    return data;
}

string File::read_file(string nameFile) {

    string line;
    string data;
    ifstream in(IN_PATH_FILE + nameFile, ios::binary);

    if (in.is_open()) {
        while (getline(in, line)) {
            data += line;
        }
    }

    in.close();

    return data;
}

void File::write_file(string data) {
```

```
    ofstream out(OUT_PATH_FILE);
    if(out.is_open()) {
        out << data;
    }

    out.close();
}

void File::write_file(string data, string nameFile) {

    ofstream out(OUT_PATH_FILE + nameFile);
    if(out.is_open()) {
        out << data;
    }

    out.close();
}
```

```
#include "Lab1/cpC_files.h"
#include <iostream>
#include <vector>

using namespace std;

void start_lab1();

int main() {

    start_lab1();
    return 0;
}

void start_lab1() {

    vector<std::string> files = {"10K_file", "100K_file", "1M_file", "10M_file",
"100M_file"};
    vector<long long> times;

    auto *file = new Lab1::File();

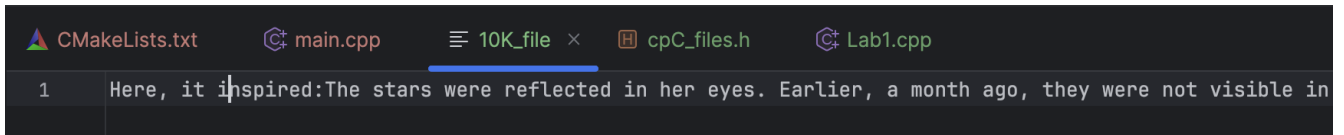
    for (const auto &name_file : files) {
        string sourcePath = name_file; // Укажите путь к исходному файлу
        string destinationPath = name_file + "_copy"; // Путь к файлу назначения

        long long startTime = Lab1::File::getCurrentTimeMillis();
        string data = file->read_file(name_file);
        file->write_file(data, destinationPath);
        long long endTime = Lab1::File::getCurrentTimeMillis();

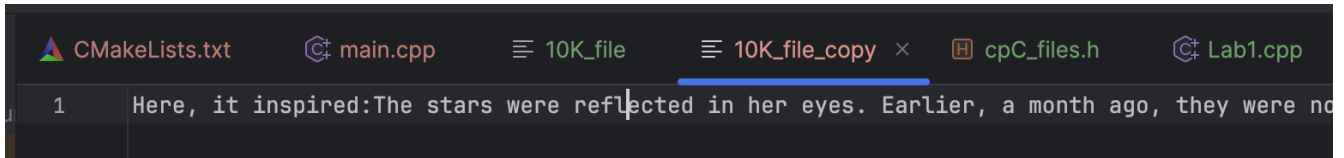
        times.push_back(endTime - startTime);
    }

    // Вывод результатов в виде таблицы
    std::cout << "File Size\tTime (ms)\n";
    for (size_t i = 0; i < files.size(); ++i) {
        std::cout << files[i] << "\t" << times[i] << "\n";
    }
}
```

Пример работы:



A screenshot of a code editor interface. The top bar shows five tabs: CMakeLists.txt, main.cpp, 10K_file (active), cpC_files.h, and Lab1.cpp. The 10K_file tab is highlighted with a blue underline. The editor area shows a single line of text: "Here, it inspired:The stars were reflected in her eyes. Earlier, a month ago, they were not visible in". The line number "1" is visible on the left side of the editor.



A screenshot of a code editor interface. The top bar shows six tabs: CMakeLists.txt, main.cpp, 10K_file, 10K_file_copy (active), cpC_files.h, and Lab1.cpp. The 10K_file_copy tab is highlighted with a blue underline. The editor area shows a single line of text: "Here, it inspired:The stars were reflected in her eyes. Earlier, a month ago, they were not visible in". The line number "1" is visible on the left side of the editor.