

Липецкий государственный технический университет  
Факультет автоматизации и информатики  
Кафедра Автоматизированных систем управления

Лабораторная работа №1  
по дисциплине «Операционные системы»  
Разработка программ обработки файлов в ОС Windows

Студент

Усанов А.В.

Группа АИ-19

Руководитель

Доцент

Журавлёва М.Г.

Липецк 2021г.

## Цель работы

Получение навыков использования функций Windows API, обеспечивающих обработку файлов.

## Задание кафедры

Написать программу вставки  $n$  строк исходного файла в заданную позицию результирующего файла и последующей установки/получения атрибутов результирующего файла. Исходные данные указываются пользователем в аргументах командной строки: количество строк – в качестве второго аргумента, номера строк – в качестве третьего –  $[(n+3)-1]$ -го аргументов, далее следуют имена исходного и результирующего файлов, следующий аргумент – позиция, в которую следует вставлять строки.

№ вар.	Количество строк, $n$	Номера строк	Позиция в результирующем файле	Установка атрибутов
1	3	1,3,9	Начало файла	«только для чтения», «скрытый»
2	3	4,7,10	Текущая позиция	«архивный», «скрытый»
3	3	2,3,10	Конец файла	«системный», «скрытый»
4	3	3,5,6	Начало файла	«только для чтения», «архивный»
5	3	1,8,10	Текущая позиция	«только для чтения», «системный»
6	4	2,4,6,8	Конец файла	«только для чтения», «скрытый»
7	4	1,5,7,8	Начало файла	«только для чтения», «зашифрованный»
8	4	2,3,7,8	Текущая позиция	«зашифрованный», «скрытый»
9	4	4,7,9,10	Конец файла	«зашифрованный», «архивный»
10	4	1,3,6,7	Начало файла	«только для чтения», «скрытый», «архивный»
11	3	6,8,10	Текущая позиция	«только для чтения», «скрытый», «системный»
12	3	7,9,12	Конец файла	«зашифрованный», «скрытый», «системный»
13	3	4,5,10	Начало файла	«зашифрованный», «скрытый», «архивный»
14	3	2,7,11	Текущая позиция	«только для чтения», «зашифрованный», «скрытый»
15	3	3,8,10	Конец файла	«только для чтения», «зашифрованный», «системный»
16	4	1,2,5,6	Начало файла	«только для чтения», «зашифрованный», «архивный»

Рисунок 1 – Задание к лабораторной работе №1. Вариант №12

## Код

```
#include <iostream>
#include <Windows.h>
#include <vector>
#include <string>

using namespace std;

wstring s2ws(const std::string& s)
{
    int len;
    int slength = (int)s.length() + 1;
    len = MultiByteToWideChar(CP_ACP,
        0,
        s.c_str(),
        slength,
        0,
        0
    );
    wchar_t* buf = (wchar_t*)malloc(len * sizeof(int));
    MultiByteToWideChar(CP_ACP, 0, s.c_str(), slength, buf, len);
    std::wstring r(buf);
    free(buf);
    return r;
}

string GetLineForWrite(HANDLE fileHANDLE)
{
    DWORD BytesReadDWORD = NULL;
    string result;
    DWORD fileSize = GetFileSize(fileHANDLE, NULL);
    int amountOfReadLines = 0;
    char symbolBuffer;
    bool isTargetLine = false;

    for (int i = 0; i < fileSize; ++i)
    {
        try {
            ReadFile(fileHANDLE, &symbolBuffer, 1, &BytesReadDWORD,
                NULL);
        }
        catch (exception ex) {
            throw ex;
        }
        if (symbolBuffer == '\n')
        {
            amountOfReadLines += 1;
            if (amountOfReadLines == 7 || amountOfReadLines == 9 ||
amountOfReadLines == 12)
                isTargetLine = true;
            else
                isTargetLine = false;
        }
        if (isTargetLine == true)
        {
            result += symbolBuffer;
        }
    }
    return result;
}

int main(int argc, char* argv[])
{

```

```

HANDLE sourceHandle = CreateFile(s2ws(argv[1]).c_str(),
    GENERIC_READ,
    FILE_SHARE_READ,
    NULL,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    NULL);
HANDLE targetHandle = CreateFile(s2ws(argv[2]).c_str(),
    GENERIC_WRITE,
    0,
    NULL,
    OPEN_ALWAYS,
    FILE_ATTRIBUTE_NORMAL,
    NULL);

string lineForWrite;
try
{
    lineForWrite = GetLineForWrite(sourceHandle);
}
catch (exception)
{
    cout << "Error";
    CloseHandle(sourceHandle);
    CloseHandle(targetHandle);
    return 0;
}

SetFilePointer(targetHandle,
    0,
    0,
    FILE_END);

WriteFile(targetHandle,
    lineForWrite.c_str(),
    lineForWrite.size(),
    NULL,
    NULL);

SetFileAttributes(s2ws(argv[2]).c_str(), FILE_ATTRIBUTE_ENCRYPTED |
FILE_ATTRIBUTE_HIDDEN | FILE_ATTRIBUTE_SYSTEM);

CloseHandle(sourceHandle);
CloseHandle(targetHandle);
system("pause");
return 0;
}

```