

**Отчет по лабораторной работе №1
«Копирование файлов»**

Выполнил: студент группы Р3217

Плюхин Дмитрий

Преподаватель: Зыков А. Г.

2017 год

1. Задание

Создайте файл, содержащий массив структур, каждая из которых содержит информацию о Ваших результатах за три семестра (фамилия, семестр, дисциплина, оценка).

Реализуйте простое последовательное копирование содержимого файла тремя различными способами:

1. С использованием библиотеки C.
2. С использованием Windows.
3. С использованием вспомогательной функции Windows — CopyFile.

Результат вывести на печать рез-ты копирования и средний балл.

Для анализа способов копирования создайте произвольный файл большого размера (экспериментально увеличивая для получения наглядных результатов) и определите временные характеристики каждого из способов. Результаты выведите в виде таблицы.

Проведите сравнительный анализ способов. Достоинства и недостатки.

2. Листинг основной части программы

Файл main.cpp реализует первую часть задания, а именно, создает массив структур данных, записывает его в файл, производит копирование тремя разными способами, а также демонстрирует использование Windows API для вывода на экран текущей директории. Код, не относящийся непосредственно к копированию файлов, опущен.

```
#include <iostream>
#include <cstdlib>
#include <stdio.h>
#include <string.h>
#include <fstream>
#include <windows.h>
#define BUF_SIZE 256
using namespace std;

struct mark{
    char surname[10], discipline[40];
    int semester, mark;
};

int copyByCLib(const char* src, const char* dest){
    FILE *in_file, *out_file;
    char rec[BUF_SIZE];
    size_t bytes_in, bytes_out;

    in_file = fopen(src,"rb");
    if (in_file == NULL){
        return 1;
    }

    out_file = fopen(dest,"wb");
    if (out_file == NULL){
        return 1;
    }

    while((bytes_in = fread(rec, 1, BUF_SIZE, in_file))>0){
        bytes_out = fwrite(rec, 1, bytes_in, out_file);
        if (bytes_out != bytes_in){
            return 2;
        }
    }

    fclose(in_file);
    fclose(out_file);

    return 0;
}

int copyBywin(const char* src, const char* dest){
    HANDLE hIn, hOut;
    DWORD nIn, nOut, nBytes;
    CHAR Buffer[BUF_SIZE];
```

```

CHAR DirName[BUF_SIZE];
nBytes = GetCurrentDirectory(BUF_SIZE, DirName);
cout << "Current directory : " << DirName << endl;
hIn = CreateFile(src, GENERIC_READ, 0, NULL, OPEN_EXISTING, 0, NULL);
if (hIn == INVALID_HANDLE_VALUE){
    return 1;
}
hOut = CreateFile(dest, GENERIC_WRITE, 0, NULL, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
if (hOut == INVALID_HANDLE_VALUE){
    return 1;
}
while(ReadFile(hIn, Buffer, BUF_SIZE, &nIn, NULL) && nIn > 0){
    WriteFile(hOut, Buffer, nIn, &nOut, NULL);

    if (nIn != nOut){
        return 2;
    }
}
CloseHandle(hIn);
CloseHandle(hOut);
return 0;
}

int copyByCopyFile(const char* src, const char* dest){
    if (!CopyFile(src, dest, FALSE)){
        return 1;
    }
    return 0;
}

float getAverage(const char* fileName){
    mark* marks_loaded = loadMarks(fileName);

    int sum = 0;
    for (int i = 0; i < 14; i++){
        sum += marks_loaded[i].mark;
    }

    return (float)sum/14;
}

int main(){
    mark* marks = getMarks();

    saveMarks("data/data.dat", marks);

    if (copyByWin("data/data.dat", "data/data_win_copied.dat")==0){
        cout << "Successfully copied by win API" << endl;
    } else {
        cout << "Error when copying by win API" << endl;
    }

    if (copyByCLib("data/data.dat", "data/data_c_copied.dat")==0){
        cout << "Successfully copied by C lib" << endl;
    } else {
        cout << "Error when copying by C lib" << endl;
    }

    if (copyByCopyFile("data/data.dat", "data/data_copyfile_copied.dat")==0){
        cout << "Successfully copied by Copy File" << endl;
    } else {
        cout << "Error when copying by CopyFile" << endl;
    }

    cout << "Average mark is " << getAverage("data/data_c_copied.dat") << endl;

    return 0;
}

```

Файл tester.cpp реализует вторую часть задания, а именно, производит замеры времени копирования больших файлов (в несколько тысяч килобайт) разными способами, и выводит их на экран в виде таблицы.

```

int createFile(int size){
    ofstream fout("data/file.dat");

```

```

    for (int i = 0; i < size; i++){
        fout.write("DATA",1024);
    }
    fout.close();
}

float getTimeExecuting(int (*func)(const char*, const char*),const char* src, const char*
dest){
    float fTimeStart = clock()/(float)CLOCKS_PER_SEC;

    func(src, dest);

    float fTimeStop = clock()/(float)CLOCKS_PER_SEC;
    return fTimeStop - fTimeStart;
}

int initializeCell(){
    cout << "| ";
    cout.setf(ios::left);
    cout.width(10);
    return 0;
}

int showCell(const char* content){
    initializeCell();
    cout << content;
}

int showCell(const char* content, const char* finalizer){
    initializeCell();
    cout << content << finalizer << endl;
}

int showCell(int content){
    initializeCell();
    cout << content;
}

int showCell(float content){
    initializeCell();
    cout << content;
}

int showCell(float content, const char* finalizer){
    initializeCell();
    cout << content << finalizer << endl;
}

int drawHorizontalLine(int length){
    cout.width(length);
    cout.fill('-');
    cout << '-' << endl;
    cout.fill(' ');
}

int showTableHead(){
    drawHorizontalLine(50);
    showCell("Size(Kb)");
    showCell("C lib");
    showCell("win API");
    showCell("CopyFile", " |");
    drawHorizontalLine(50);
    return 0;
}

int main(){
    int size = 0;
    showTableHead();
    for (int i = 1; i < 10; i++){
        size = i*10000;
        createFile(size);
        showCell(size);
        showCell(getTimeExecuting(copyByCLib,"data/file.dat","data/file1.dat"));
        showCell(getTimeExecuting(copyByWin,"data/file.dat","data/file2.dat"));
        showCell(getTimeExecuting(copyByCopyFile,"data/file.dat","data/file3.dat"), " |");
        drawHorizontalLine(50);
    }
}

```

3. Результаты работы программы

В результате работы программы получили следующий вывод:

```
build/main
Current directory : C:\Users\Zerbs\Desktop\lab1
Succesfully copied by Win API
Succesfully copied by C lib
Succesfully copied by Copy File
Average mark is 4.78571
g++ tester.cpp -o build/tester
build/tester
```

Size(Kb)	C lib	Win API	CopyFile
10000	0.118	0.535	0.248
20000	0.2	1.018	0.043
30000	0.392	1.542	0.0739999
40000	0.394	1.971	0.0869999
50000	0.475	2.616	0.118
60000	0.485001	3.091	0.244999
70000	0.68	3.571	0.151001
80000	0.754	4.342	0.228001
90000	0.861	4.679	0.190998

4. Вывод

Таким образом, существует три способа копирования файлов в системе Windows: первый, использующий библиотеку языка C, может быть применен в том случае, если требуется возможность как можно более быстрой адаптации программы под альтернативные операционные системы, однако не предъявляется особых требований к производительности; второй способ, использующий функцию CopyFile, является наиболее эффективным в плане быстродействия. Наименее «приоритетным» в рассмотренных отношениях представляется способ, использующий WindowsAPI, поскольку он как сильно проигрывает по времени работы, так и не предоставляет достаточной переносимости программ, хотя и обеспечивает высокую гибкость, если программа запускается только в рамках Windows.