СПб НИУ ИТМО

кафедра ИПМ

Системное программное обеспечение

Лабораторная работа № 2

Копирование файлов

Работу выполнил:

Студент II курса

Группы № 2120

Журавлев Виталий

Санкт-Петербург

2014 г.

**Цель работы:**

Создайте файл, содержащий массив структур, каждая из которых содержит поля:

ФИО;

Дату рождения;

Семестр (1,2 или 3);

Названия предметов (Дисциплин ) и полученные оценки по ним;

Одна структура на семестр. Выведите массив структур на экран в окно.Скопируйте созданный файл в другую директорию тремя способами:

1. С использованием библиотеки С

2. С использованием средств Windows

3. С использованием вспомогательных функций windows –copyfile

Оцените достоинства и недостатки используемых методов

**Код программы:**

#include "stdafx.h"

#include "windows.h"

#include "stdafx.h"

#include "time.h"

#include "fstream"

#include "string"

#include "map"

#include "array"

#define BUF\_SIZE 256

using namespace std;

struct Semestr

{

public:

 string Name, Date;

 int Sem;

 map <string, char> Rating;

 Semestr()

 {

 Semestr::Name = "";

 Semestr::Date = "";

 Semestr::Sem = 0;

 }

 Semestr(string name, string date, int sem, map <string, char> rating)

 {

 Semestr::Name = name;

 Semestr::Date = date;

 Semestr::Sem = sem;

 Semestr::Rating = rating;

 }

};

void MakeArray(array<Semestr, 3>& Journal)

{

 array<string, 15> subj = { "Информатика", "Основы программирования", "Физика", "Математика", "Основы вычислительной техники",

 "Дискретная математика", " Основы вычислительной техники", "Основы программирования", "Физика", "Математика",

 "Вычислительная математика", "Языки системного программирования", "Системное программное обеспечение", "Физика", "Математика" };

 array<char, 15> rates = { 'A', 'A', 'A', 'A', 'A', 'A', 'C', 'A', 'B', 'A', 'A', 'A', 'A', 'C', 'B' };

 map <string, char> Rating;

 string Name = "Журавлев Виталий Владимирович";

 string Date = "6 апреля 1994 года";

 for (int NSem = 1; NSem < 4; NSem++)

 {

 Rating.insert(pair<string, char>(subj[5 \* NSem - 5], rates[5 \* NSem - 5]));

 Rating.insert(pair<string, char>(subj[5 \* NSem - 4], rates[5 \* NSem - 5]));

 Rating.insert(pair<string, char>(subj[5 \* NSem - 3], rates[5 \* NSem - 5]));

 Rating.insert(pair<string, char>(subj[5 \* NSem - 2], rates[5 \* NSem - 5]));

 Rating.insert(pair<string, char>(subj[5 \* NSem - 1], rates[5 \* NSem - 5]));

 Semestr sem(Name, Date, NSem, Rating);

 Journal[NSem - 1] = sem;

 Rating.clear();

 }

}

int WriteArray(LPCTSTR file\_str, array<Semestr, 3>& Journal)

{

 ofstream f(file\_str, ios::out | ios::binary);

 f.write((char\*)&Journal, sizeof(Journal));

 f.close();

 return 1;

}

int cpC(LPCTSTR in\_file\_str, LPCTSTR out\_file\_str)

{

 ifstream in(in\_file\_str, ios::in | ios::binary);

 ofstream out(out\_file\_str, ios::out | ios::binary);

 if (!in || !out)

 return 2;

 char buffer[BUF\_SIZE];

 while (in)

 {

 in.read(buffer, BUF\_SIZE);

 streamsize readed = in.gcount();

 if (readed > 0)

 out.write(buffer, readed);

 }

 return 1;

}

int cpW(LPCTSTR in\_file\_str, LPCTSTR out\_file\_str) {

 HANDLE hIn, hOut;

 DWORD nIn, nOut;

 CHAR Buffer[BUF\_SIZE];

 hIn = CreateFile(in\_file\_str, GENERIC\_READ, FILE\_SHARE\_READ, NULL, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

 if (hIn == INVALID\_HANDLE\_VALUE) {

 printf("Error: invalid hadle value! Code: %x\n", GetLastError());

 return 2;

 }

 hOut = CreateFile(out\_file\_str, GENERIC\_WRITE, 0, NULL, CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);

 if (hOut == INVALID\_HANDLE\_VALUE) {

 printf("Error: error read file! Code: %x\n", GetLastError());

 return 3;

 }

 while (ReadFile(hIn, Buffer, BUF\_SIZE, &nIn, NULL) && nIn > 0) {

 WriteFile(hOut, Buffer, nIn, &nOut, NULL);

 if (nIn != nOut) {

 printf("Error: error write file! Code: %x\n", GetLastError());

 return 4;

 }

 }

 CloseHandle(hIn);

 CloseHandle(hOut);

 return 1;

}

int cpCF(LPCTSTR in\_file\_str, LPCTSTR out\_file\_str) {

 CopyFile(in\_file\_str, out\_file\_str, FALSE);

 return 0;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

 LPCTSTR OrigFName = L"D:/SPO/Original/Journal.bin";

 LPCTSTR CFName = L"D:/SPO/C/JournalC.bin";

 LPCTSTR CopyFName = L"D:/SPO/copyfile/JournalCopy.bin";

 LPCTSTR WindowsFName = L"D:/SPO/Windows/JournalWindows.bin";

 SYSTEMTIME st, ft;

 array<Semestr, 3> Journal;

 MakeArray(Journal);

 WriteArray(OrigFName, Journal);

 GetSystemTime(&st);

 cpC(OrigFName, CFName);

 GetSystemTime(&ft);

 printf("Copy with C libraries time: %d ms\n", (ft.wMilliseconds - st.wMilliseconds));

 GetSystemTime(&st);

 cpW(OrigFName, WindowsFName);

 GetSystemTime(&ft);

 printf("Copy with Windows time: %d ms\n", (ft.wMilliseconds - st.wMilliseconds));

 GetSystemTime(&st);

 cpCF(OrigFName, CopyFName);

 GetSystemTime(&ft);

 printf("Copy with CopyFile time: %d ms\n", (ft.wMilliseconds - st.wMilliseconds));

 return 0;

}

**Результат работы програмы:**

**Вывод:**

В ходе выполнения лабораторной работы были изучены основные типы копирования файлов: с использованием встроенной библиотеки С, с использованием Windows.и с использованием вспомогательной функции Windows – CopyFile.

В бинарный файл Journal.bin записывался массив из структур Semestr методом C++ с помощью fstream в виде потока байтов. Копирование происходило в такие же файлы, переименованные по имени метода и находящиеся в разных директориях.

Основным критерием для выяснения эффективности работы метода служило время выполнения метода, т.к. все скопированные файлы были точными копиями оригинала и данные не пропадали. В ряде проделанных экспериментов выяснилось, что стандартный метод Windows работает всегда быстрее остальных. Следом за ним, не на много отставая находится метод из библиотеки C. На последнем же месте, постоянно значительно отставая находится метод CopyFile.

Что касается простоты реализации, то CopyFile является лидером, т.к. там нужно просто вызвать функцию, передав ей в качестве аргументов имена файлов, и логическую переменную проверки. Зато метод из библиотеки C является более гибким и понятным, легче поймать ошибку (потоки возвращают логические переменные) и можно реализовать дополнительный функционал. Метод из Windows же является самым непонятным и неудобным, много проблем с установкой параметров и типами переменных.