Министерство общего и профессионального образования РФ

Московский Авиационный Институт

(Государственный Технический Университет)

Факультет  3

Кафедра  304

*ОТЧЕТ*

по

(наименование дисциплины)

Тема

Принял:

(фамилия, инициалы)

(подпись) (дата)

Выполнил:

(группа)

(фамилия, инициалы)

(подпись) (дата)

Москва 2012 г.

**Постановка задачи**

Написать программу поиска фалов по заданной дате создания с возможностью выбора директории поиска и рекурсивной проверки всех папок содержащихся в директории. Результат вывести в файл.

**Среда разработки**

Программа была реализована с помощью [интегрированной среды разработки](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) программного обеспечения **Microsoft Visual Studio**.

**Visual Studio** включает в себя [редактор исходного кода](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0) с поддержкой технологии [IntelliSense](http://ru.wikipedia.org/wiki/IntelliSense) и возможностью простейшего [рефакторинга кода](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3). Встроенный [отладчик](http://ru.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня.

В данном проекте использовались функции семейства **Windows API.**

**Windows API** — общее наименование целого набора базовых функций интерфейсов программирования приложений операционных систем *Microsoft Windows* корпорации «Майкрософт». Является самым прямым способом взаимодействия приложений с *Windows*.

**Список прецедентов**

1. Чтение начальных условий
2. Поиск в заданной директории других папок
3. Сравнение даты найденных файлов с заданной датой
4. Вывод в файл полный путь к файлам с требуемой датой

**Описание интерфейса программы**

Данная программа выполнена в виде консольного приложения Win32.

Все начальные значения задаются в командной строке с сопутствующими сообщениями о просьбе ввести тот или иной параметр.

При вводе пути к директории которая нас интересует необходимо учитывать следующее:

1 Вместо знака “\” вводится “\\”

2 Знак “\*” означает любую последовательность символов. Поэтому, если требуется найти например только файлы с исходным кодом нужно ввести “\*.cpp”.

**Текст программы**

Файл finddata.cpp

// finddata.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include "searchfiles.h"

#include <conio.h>

struct SFileData

{

WORD wDay;

WORD wMonth;

WORD wYear;

};

SFileData sFD;

void DoSomething(FILE \*log, wchar\_t file[], WIN32\_FIND\_DATA \*wfd)

{

SYSTEMTIME pSysTime;

FileTimeToSystemTime( &(wfd->ftCreationTime), &pSysTime );

if( pSysTime.wDay == sFD.wDay && pSysTime.wMonth == sFD.wMonth && pSysTime.wYear == sFD.wYear )

fwprintf( log, L"%s\t%i.%i.%i\t%i:%i\n", file, pSysTime.wDay, pSysTime.wMonth, pSysTime.wYear,

pSysTime.wHour, pSysTime.wMinute);

}

int main()

{

//if (!SetConsoleOutputCP(1251))

//{

// // попытка установить нужную страницу закончилась неудачно

// // выводим причину ошибки

// char msg[300];

// FormatMessageA(FORMAT\_MESSAGE\_IGNORE\_INSERTS

// | FORMAT\_MESSAGE\_FROM\_SYSTEM, 0, GetLastError(),

// 0, msg, sizeof(msg) - 1, 0);

// // если версия Windows локализованная то ошибка будет

// // на русском языке, поэтому

// CharToOemA(msg, msg);

// printf("SetConsoleOutputCP Error: %s\n", msg);

// return 1;

//}

setlocale(LC\_ALL,"Russian");

cout << "ПОИСК ФАЙЛОВ ПО ДАТЕ" << endl << "Выполнил: Гусев М.С 03-527" << endl;

cout << "Введите число ";

cin >> sFD.wDay;

cout << "Введите число ";

cin >> sFD.wMonth;

cout << "Введите число ";

cin >> sFD.wYear;

FILE \*log;

log = fopen( "log.txt", "w" );

if( log == NULL )

{

cout << "Произошла ошибка при открытии выходного файла";

return FALSE;

}

cout << "Введите маску для поиска файла ";

wchar\_t wstr[260];

fflush( stdin );

wscanf(L"%s", wstr );

SearchFiles( log, wstr, (LPSEARCHFUNC)DoSomething, true);

fclose( log );

return 0;

}

Файл searchfiles.cpp

#include "stdafx.h"

typedef void (\*LPSEARCHFUNC)(FILE \*log , wchar\_t file[], WIN32\_FIND\_DATA \*wfd);

BOOL SearchFiles( FILE \*log, LPCWSTR lpszFileName, LPSEARCHFUNC lpSearchFunc, BOOL bInnerFolders = TRUE)

{

LPWSTR part;

wchar\_t tmp[MAX\_PATH]; // временный массив

wchar\_t name[MAX\_PATH];

SYSTEMTIME pSysTime;

HANDLE hSearch = NULL;

WIN32\_FIND\_DATA wfd;

memset(&wfd, 0, sizeof(WIN32\_FIND\_DATA));

// сначало поиск внутри вложенных папках ...

if(bInnerFolders)

{

if(GetFullPathNameW(lpszFileName, MAX\_PATH, tmp, &part) == 0) return FALSE;

wcscpy(name, part);

wcscpy(part, L"\*.\*");

// если папки существуют, то делаем поиск

wfd.dwFileAttributes = FILE\_ATTRIBUTE\_DIRECTORY;

if (!((hSearch = FindFirstFile(tmp, &wfd)) == INVALID\_HANDLE\_VALUE))

do

{

// в каждой папке есть две папки с именами "." и ".."

// и эти папки мы не трогаем

// пропускаем папки "." и ".."

if (!wcsncmp(wfd.cFileName, L".", 1) || !wcsncmp(wfd.cFileName, L"..", 2))

continue;

if (wfd.dwFileAttributes & FILE\_ATTRIBUTE\_DIRECTORY) // если мы нашли папку

{

wchar\_t next[MAX\_PATH];

if(GetFullPathNameW(lpszFileName, MAX\_PATH, next, &part) == 0) return FALSE;

wcscpy(part, wfd.cFileName);

wcscat(next, L"\\");

wcscat(next, name);

SearchFiles( log, next, lpSearchFunc, TRUE);

}

}

while (FindNextFile(hSearch, &wfd)); // ищем следующий файл

FindClose (hSearch); // заканчиваем поиск

}

if ((hSearch = FindFirstFile(lpszFileName, &wfd)) == INVALID\_HANDLE\_VALUE)

return TRUE; // в противном случае выходим

do

if (!(wfd.dwFileAttributes & FILE\_ATTRIBUTE\_DIRECTORY)) // если мы нашли файл

{

wchar\_t file[MAX\_PATH];

if(GetFullPathNameW(lpszFileName, MAX\_PATH, file, &part) == 0) return FALSE;

wcscpy(part, wfd.cFileName);

lpSearchFunc( log, file, &wfd);

}

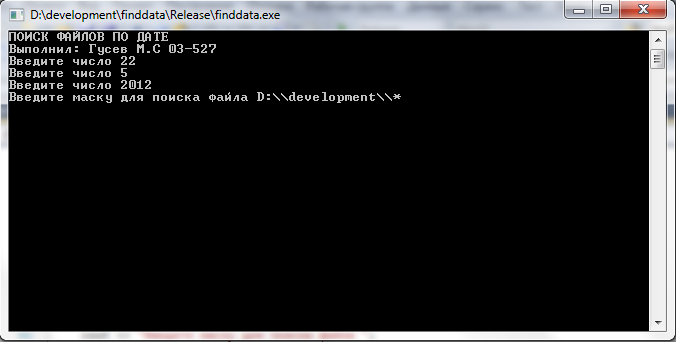
while (FindNextFile(hSearch, &wfd)); // ищем следующий файл

FindClose (hSearch); // заканчиваем поиск

return TRUE;

}

**Результат выполнения**



Пример выходного файла

D:\development\finddata\finddata\finddata.cpp 22.5.2012 18:57

D:\development\finddata\finddata\finddata.vcxproj 22.5.2012 18:57

D:\development\finddata\finddata\finddata.vcxproj.filters 22.5.2012 18:57

D:\development\finddata\finddata\finddata.vcxproj.user 22.5.2012 18:57

D:\development\finddata\finddata\ReadMe.txt 22.5.2012 18:57

D:\development\finddata\finddata\stdafx.cpp 22.5.2012 18:57

D:\development\finddata\finddata\stdafx.h 22.5.2012 18:57

D:\development\finddata\finddata\targetver.h 22.5.2012 18:57

D:\development\finddata\finddata.sdf 22.5.2012 18:57

D:\development\finddata\finddata.sln 22.5.2012 18:57

D:\development\finddata\finddata.suo 22.5.2012 18:57

D:\development\interface\doc\forreport.docx 22.5.2012 19:24

D:\development\interface\.gitignore 22.5.2012 19:46

**Вывод**

# В данной работе были рассмотрены такие функции WINAPI как FindFirstFile, FindNextFile, FindClose, предназначенные специально для поиска файлов в директории, а так же структура WIN32\_FIND\_DATA, содержащая информацию о файле, который был найден [**FindFirstFile**](http://msdn.microsoft.com/en-us/library/windows/desktop/aa364418(v=vs.85).aspx), **[FindNextFile](http://msdn.microsoft.com/en-us/library/windows/desktop/aa364428(v=vs.85).aspx)** функциями.

# В редких случаях или при сильно загруженной системе, информация о файле в вайловой системе  NTFS  может быть не актуальна на момент вызова FindFirstFile функции.

# Было замечено что, при использовании структуры WIN32\_FIND\_DATA, если файл имеет длинное имя, то полное имя появляется в члене структуры cFileName , а усеченная версия имя  в формате 8.3,  появится в члене cAlternateFileName. Это необходимо учитывать при работе в DOS, так как известно что в такой системе имя файла может иметь длину максимум 8 знаков. Если имя имеет более 8-ми знаков, то остальные знаки не принимаются в расчет.