Белорусский государственный технологический университет

Кафедра Информационных Систем и Технологий

**Курс «Операционные системы и системное программирование»**

**Лабораторная работа 14. Работа с файлами и каталогами**

Выполнил: Сазановец Я.И.

ФИТ 3 курс 4 группа

Минск 2018

1. Чтение текстового файла. Указание: файл должен быть заранее подготовлен.

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

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

{

FILE\* fp;

char str[80];

if((fp=fopen("d:\\temp\\1.txt","rt"))==NULL)

{

puts("Opening failed");

getchar();

}

else

{

fgets(str,79,fp);

printf("read: %s", str);

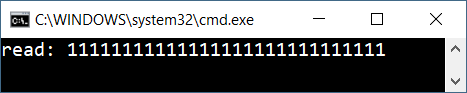
int z=fclose(fp);

getchar();

}

return 0;

}



2. Следующий пример демонстрирует запись с последующим чтением из текстового файла.

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

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

{

FILE\* fp;

char str[80]="glad to hear from you";

if((fp=fopen("d:\\temp\\1.txt","wt"))==NULL)

{

puts("Opening failed");

getchar();

}

else

{

fputs(str,fp);

int z=fclose(fp);

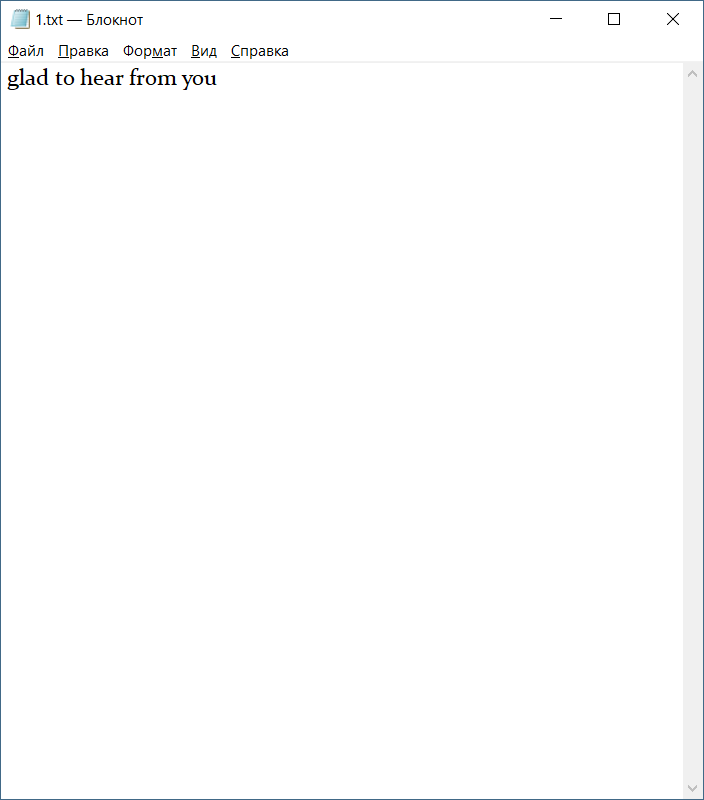
fp=fopen("d:\\temp\\1.txt","rt");

fgets(str,79,fp);

puts(str);

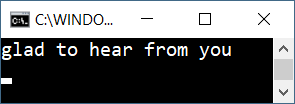
z=fclose(fp);

getchar();

 }

return 0;

}



3. Создание файла средствами API WINDOWS.

Следующий пример демонстрирует создание файла средствами API. Используется функция CreateFile. Если файл существует, то он пересоздается, а его содержимое стирается.

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include "windows.h"

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

{

TCHAR Buffer[256]=TEXT("Glad to hear from You");

DWORD nLenCurDir;

HANDLE hOut;

FILE\* fp;

TCHAR stdPath[30] = TEXT("d:\\temp\\1.txt");

hOut = CreateFile(stdPath, GENERIC\_WRITE, 0, NULL,

CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);

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

printf("ERROR %x \n",GetLastError());

getchar();

return 2;

}

else

{

//nLenCurDir = GetCurrentDirectory(256, Buffer);

// WriteFile(hOut, Buffer, 256, NULL, NULL );

printf("Created");

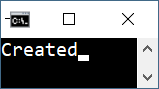
getchar();

CloseHandle(hOut);

}

return 0;

}



4. Следующий пример демонстрирует создание файла средствами API+ запись в файл текста из буфера. Используется функция WriteFile.

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include "windows.h"

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

{

TCHAR Buffer[256]=TEXT("Glad to hear from You\n");

HANDLE hOut;

DWORD dwNumberOfBytes;

FILE\* fp;

TCHAR stdPath[30] = TEXT("d:\\temp\\1.txt");

hOut = CreateFile(stdPath, GENERIC\_WRITE, 0, NULL,

CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);

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

printf("ERROR %x \n",GetLastError());

getchar();

return 2;

}

else

{

WriteFile(hOut, Buffer, 256, &dwNumberOfBytes, NULL );

printf("Created and Written!");

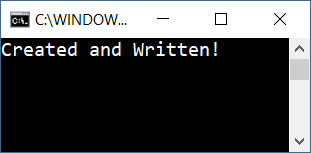
getchar();

CloseHandle(hOut);

}

return 0;

}



5. Следующий пример демонстрирует чтение ранее созданного файла средствами API+. Используется функция ReadFile.

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include "windows.h"

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

{

TCHAR Buffer[256];

HANDLE hIn;

DWORD dwNumberOfBytes;

FILE\* fp;

TCHAR stdPath[30] = TEXT("d:\\temp\\1.txt");

hIn = CreateFile(stdPath, GENERIC\_READ, 0, NULL,

OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

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

printf("ERROR %x \n",GetLastError());

getchar();

return 2;

}

else

{

ReadFile(hIn, Buffer, 256, &dwNumberOfBytes, NULL );

printf("DataRead!");

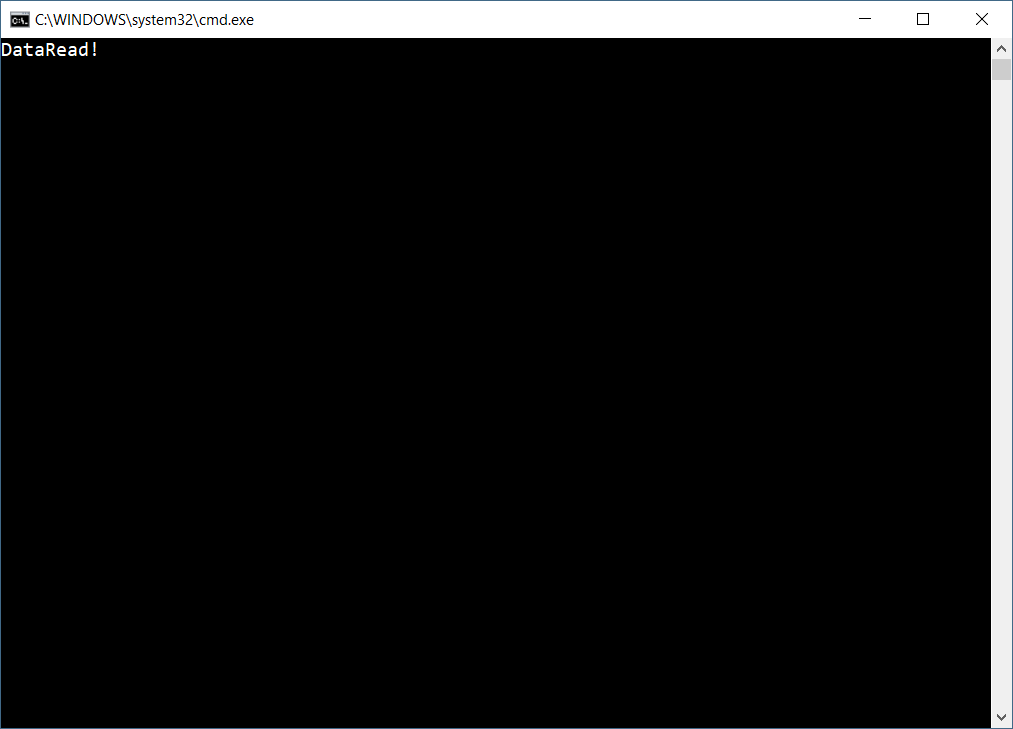
getchar();

CloseHandle(hIn);

}

return 0;

}



Данные считываются в массив символов Buffer. Покажем, как вывести этот массив на консоль.

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include "windows.h"

#include "iostream"

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

{

TCHAR Buffer[256];

HANDLE hIn;

DWORD dwNumberOfBytes;

FILE\* fp;

TCHAR stdPath[30] = TEXT("d:\\temp\\1.txt");

hIn = CreateFile(stdPath, GENERIC\_READ, 0, NULL,

OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

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

printf("ERROR %x \n",GetLastError());

getchar();

return 2;

}

else

{

ReadFile(hIn, Buffer, 256, &dwNumberOfBytes, NULL );

std::wcout << "Read: " << Buffer << '\n';

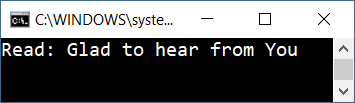
getchar();

CloseHandle(hIn);

}

return 0;

}



ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ API WINDOWS для переименования, перемещения и удаления файла.

КОПИРОВАНИЕ ФАЙЛА.

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include "windows.h"

#include "iostream"

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

{

TCHAR Buffer[256];

HANDLE hIn;

DWORD dwNumberOfBytes;

FILE\* fp;

BOOL pr2=0;

TCHAR stdPathA[30] = TEXT("d:\\temp\\1.txt");

TCHAR stdPathB[30] = TEXT("d:\\temp\\1\\1.txt");

BOOL pr = CopyFile(stdPathA,stdPathB,pr2);

if(pr)

{

printf("COPIED");

getchar();

}

else

{

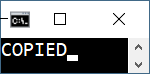
printf(" NOT COPIED");

getchar();

}

return 0;

}



ПЕРЕМЕЩЕНИЕ ФАЙЛА.

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include "windows.h"

#include "iostream"

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

{

TCHAR Buffer[256];

HANDLE hIn;

DWORD dwNumberOfBytes;

FILE\* fp;

BOOL pr2=0;

TCHAR stdPathA[30] = TEXT("d:\\temp\\1\\1.txt");

TCHAR stdPathB[30] = TEXT("d:\\temp\\1.txt");

BOOL pr= MoveFile(stdPathA,stdPathB);

if(pr)

{

printf("MOVED");

getchar();

}

else

{

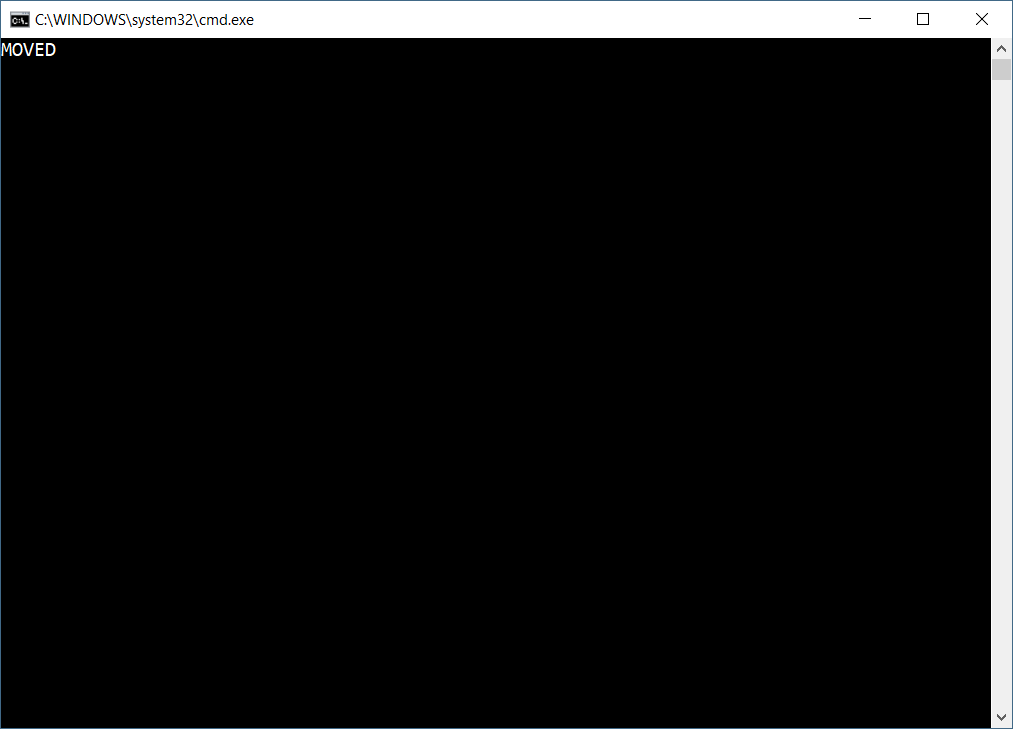
printf(" NOT MOVED");

getchar();

}

return 0;

}



УДАЛЕНИЕ ФАЙЛА.

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include "windows.h"

#include "iostream"

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

{

TCHAR Buffer[256];

HANDLE hIn;

DWORD dwNumberOfBytes;

TCHAR stdPathA[30] = TEXT("d:\\temp\\1.txt");

BOOL pr= DeleteFile(stdPathA);

if(pr)

{

printf("DELETED");

getchar();

}

else

{

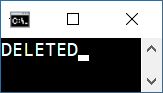
printf(" NOT DELETED");

getchar();

}

return 0;

}



ОПРЕДЕЛЕНИЕ ТЕКУЩЕГО КАТАЛОГА

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include "windows.h"

#include "iostream"

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

{

TCHAR Buffer[256];

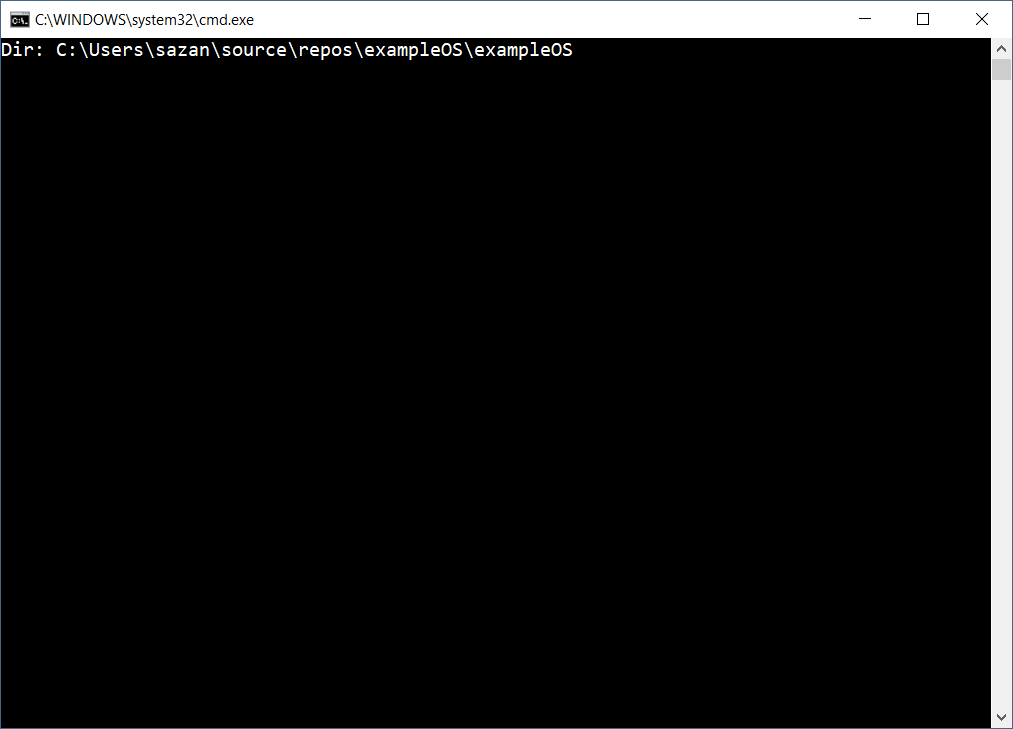
DWORD dwResult=GetCurrentDirectory(256,Buffer);

std::wcout << "Dir: " << Buffer << '\n';

getchar();

return 0;

}



СМЕНА КАТАЛОГА

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include "windows.h"

#include "iostream"

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

{

TCHAR Buffer[256];

TCHAR Buffer2[]=TEXT("d:\\temp");

DWORD dwResult=GetCurrentDirectory(256,Buffer);

std::wcout << "Dir: " << Buffer << '\n';

BOOL b=SetCurrentDirectory(Buffer2);

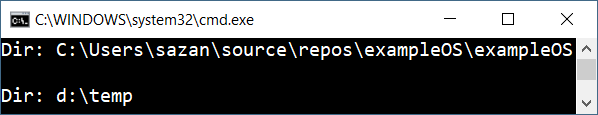
dwResult=GetCurrentDirectory(256,Buffer);

std::wcout << "Dir: " << Buffer << '\n';

getchar();

return 0;

}



СОЗДАНИЕ КАТАЛОГА

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include "windows.h"

#include "iostream"

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

{

TCHAR Buffer[256];

TCHAR Buffer2[]=TEXT("d:\\temp");

TCHAR Buffer3[]=TEXT("d:\\temp\\dir");

DWORD dwResult=GetCurrentDirectory(256,Buffer);

std::wcout << "Dir: " << Buffer << '\n';

BOOL b=SetCurrentDirectory(Buffer2);

dwResult=GetCurrentDirectory(256,Buffer);

std::wcout << "Dir: " << Buffer << '\n';

if (CreateDirectory(Buffer3,NULL))

std::wcout << "directory create" << '\n';

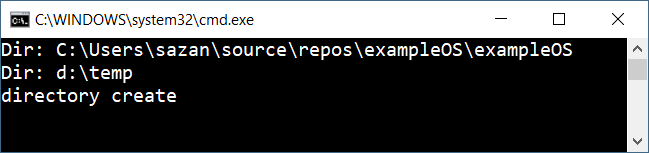
else

std::wcout << "error create directory" << '\n';

getchar();

return 0;

}



ЗАДАНИЕ.

1. Рассмотреть и выполнить приведенные примеры.
2. Создать программно два каталога.
3. В первом каталоге создать текстовый файл и записать в него текст.
4. Скопировать созданный файл из первого каталога во второй.
5. Удалить файл из первого каталога.
6. Переименовать файл во втором каталоге и прочитать его содержимое.

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include <iostream>

#include "Windows.h"

int choose;

void CreateDir();

void Text();

void Copy();

void Deleted();

void Move();

void Read();

void Menu();

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Menu();

std::cout << "Ввод: ";

do

{

std::cin >> choose;

switch (choose)

{

case 1:

CreateDir();

break;

case 2:

Text();

break;

case 3:

Copy();

break;

case 4:

Deleted();

break;

case 5:

Move();

break;

case 6:

Read();

break;

case 7:

Menu();

break;

}

std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << std::endl;

std::cout << "Ввод: ";

} while (choose != 0);

return 0;

}

void Menu()

{

std::cout << "1 - создание двух каталогов " << std::endl;

std::cout << "2 - создание текстового файла в 1м каталоге" << std::endl;

std::cout << "3 - копирование файла из 1го каталога во 2й " << std::endl;

std::cout << "4 - удаление файла из 1го каталога " << std::endl;

std::cout << "5 - переименование файла во 2м каталоге" << std::endl;

std::cout << "6 - чтение его содержимого файла во 2м каталоге" << std::endl;

std::cout << "7 - меню" << std::endl;

std::cout << "0 - выход" << std::endl;

std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << std::endl;

}

void CreateDir()

{

TCHAR Buffer2[] = TEXT("d:\\temp\\1");

TCHAR Buffer3[] = TEXT("d:\\temp\\2");

CreateDirectory(Buffer2, NULL);

CreateDirectory(Buffer3, NULL);

}

void Text()

{

TCHAR Buffer[256] = TEXT("Created");

HANDLE hOut;

DWORD dwNumberOfBytes;

TCHAR stdPath[30] = TEXT("d:\\temp\\1\\file.txt");

hOut = CreateFile(stdPath, GENERIC\_WRITE, 0, NULL,

CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);

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

std::cout << "Ошибка %x \n" << std::endl;

getchar();

}

else

{

WriteFile(hOut, Buffer, 256, &dwNumberOfBytes, NULL);

std::cout << "Создано и записано" << std::endl;

getchar();

CloseHandle(hOut);

}

}

void Copy()

{

TCHAR Buffer[256];

HANDLE hIn;

DWORD dwNumberOfBytes;

FILE\* fp;

TCHAR filea[30] = TEXT("d:\\temp\\1\\file.txt");

TCHAR fileb[30] = TEXT("d:\\temp\\2\\file.txt");

BOOL pr = CopyFile(filea, fileb, NULL);

if (pr)

{

std::cout << "Скопировано" << std::endl;

getchar();

}

else

{

std::cout << "Не скопировано" << std::endl;

getchar();

}

}

void Deleted()

{

TCHAR Buffer[256];

HANDLE hIn;

DWORD dwNumberOfBytes;

TCHAR file[30] = TEXT("d:\\temp\\1\\file.txt");

BOOL pr = DeleteFile(file);

if (pr)

{

std::cout << "Удалено" << std::endl;

getchar();

}

else

{

std::cout << "Не удалено" << std::endl;

getchar();

}

}

void Move()

{

TCHAR Buffer[256];

HANDLE hIn;

DWORD dwNumberOfBytes;

FILE\* fp;

BOOL pr2 = 0;

TCHAR fileA[30] = TEXT("d:\\temp\\2\\file.txt");

TCHAR fileB[30] = TEXT("d:\\temp\\2\\newfile.txt");

BOOL pr = MoveFile(fileA, fileB);

if (pr)

{

std::cout << "Переименовано" << std::endl;

getchar();

}

else

{

std::cout << "Не переименовано" << std::endl;

getchar();

}

}

void Read()

{

TCHAR Buffer[256];

HANDLE hIn;

DWORD dwNumberOfBytes;

FILE\* fp;

TCHAR file[30] = TEXT("d:\\temp\\2\\newfile.txt");

hIn = CreateFile(file, GENERIC\_READ, 0, NULL,OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

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

std::cout << "Данные не прочитаны" << std::endl;

getchar();

}

else

{

ReadFile(hIn, Buffer, 256, &dwNumberOfBytes, NULL);

std::cout << "Данные успешно прочитаны: " << std::endl;

std::wcout << Buffer << std::endl;

getchar();

CloseHandle(hIn);

}

}

