
Системное ПО

Работа с файлами

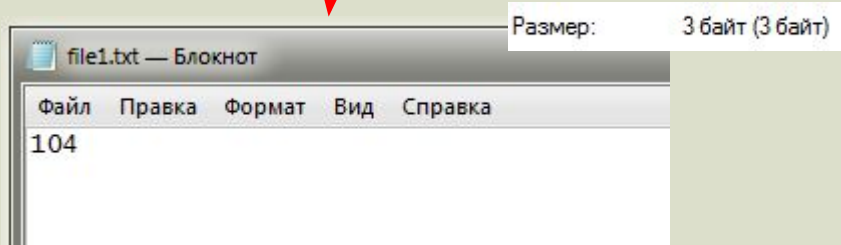
Системное ПО

Работа с файлами

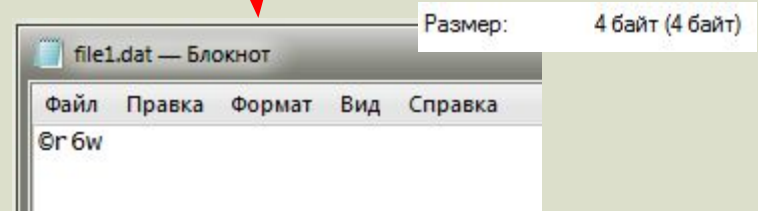
Как вывести данные в файл

Записать данные в файл в виде текста, используя например **fprintf**

```
int k=104;  
FILE *f=fopen("file1.txt","w+");  
fprintf(f,"%d",k);
```



Записать в файл двоичные данные так же, как они представлены в ЭВМ



Системное ПО

Работа с файлами

Минимум функций WinAPI для работы с файлами:

CreateFile – создать файл или открыть существующий

ReadFile – прочитать данные

WriteFile – записать данные

SetFilePointer – переместить указатель текущей позиции для чтения/записи

CloseHandle – закрыть хэндл (в том числе файла)

Системное ПО

Работа с файлами

```
HANDLE WINAPI CreateFile(  
    _In_      LPCTSTR lpFileName,  
    _In_      DWORD dwDesiredAccess,  
    _In_      DWORD dwShareMode,  
    _In_opt_  LPSECURITY_ATTRIBUTES lpSecurityAttributes,  
    _In_      DWORD dwCreationDisposition,  
    _In_      DWORD dwFlagsAndAttributes,  
    _In_opt_  HANDLE hTemplateFile  
);
```

CreateFile

Функция кроме имени файла получает еще 6 параметров, определяющих все нюансы работы с файлом. Возвращаемое значение имеет тип HANDLE (хэндл файла).

Системное ПО

Работа с файлами

dwDesiredAccess определяет желаемые права доступа к файлу. Параметр представляет собой целое число (**DWORD=unsigned int**), некоторые биты которого определяют соответствующее право доступа

Константа	Значение
GENERIC_READ	0x80000000
GENERIC_WRITE	0x40000000
GENERIC_EXECUTE	0x20000000
GENERIC_ALL	0x10000000

В двоичном представлении каждой константы есть только одна единица

Права доступа можно комбинировать, используя побитовое ИЛИ:
GENERIC_READ | GENERIC_WRITE

Системное ПО

Работа с файлами

dwShareMode определяет способ совместного использования файла с другими процессами. Ноль означает отсутствие совместного доступа (работать с файлом может только наш процесс)

Константа	Описание
FILE_SHARE_READ	Другие процессы могут открыть файл для чтения
FILE_SHARE_WRITE	Другие процессы могут открыть файл для записи

Системное ПО

Работа с файлами

dwCreationDisposition определяет поведение, в случае, если файл существует, или не существует

Константа	Описание
CREATE_ALWAYS	Создает новый файл. Если файл существует, уничтожает его содержимое
CREATE_NEW	Создает новый файл. Если файл существует, функция завершится с ошибкой
OPEN_ALWAYS	Открывает существующий файл. Если файл не существует – создает его
OPEN_EXISTING	Открывает существующий файл. Если файл не существует – функция завершается с ошибкой
TRUNCATE_EXISTING	Открывает существующий файл и уничтожает его содержимое

Системное ПО

Работа с файлами

```
BOOL WINAPI WriteFile(  
    _In_      HANDLE hFile,  
    _In_      LPCVOID lpBuffer,  
    _In_      DWORD nNumberOfBytesToWrite,  
    _Out_opt_ LPDWORD lpNumberOfBytesWritten,  
    _Inout_opt_ LPOVERLAPPED lpOverlapped  
);
```

WriteFile

Функция записывает в файл любые данные, находящиеся в памяти.
lpBuffer – указатель на начало записываемой области.
nNumberOfBytesToWrite – количество байт для записи.
lpNumberOfBytesWritten – указатель на переменную, куда будет помещен реально записанный объем информации в байтах.

Системное ПО

Работа с файлами

```
BOOL WINAPI ReadFile(  
    _In_      HANDLE hFile,  
    _Out_    LPVOID lpBuffer,  
    _In_     DWORD nNumberOfBytesToRead,  
    _Out_opt_ LPDWORD lpNumberOfBytesRead,  
    _Inout_opt_ LPOVERLAPPED lpOverlapped  
);
```

ReadFile

Функция читает данные из файла, находящиеся в памяти.

lpBuffer – указатель, куда записать прочитанные данные.

nNumberOfBytesToRead – сколько байт читать.

lpNumberOfBytesRead – указатель на переменную, куда будет помещен реально прочитанный объем информации в байтах.

Системное ПО

Работа с файлами

SetFilePointer

```
DWORD WINAPI SetFilePointer(  
    _In_        HANDLE hFile,  
    _In_        LONG lDistanceToMove,  
    _Inout_opt_ PLONG lpDistanceToMoveHigh,  
    _In_        DWORD dwMoveMethod  
);
```

dwMoveMethod –
точка отсчета
нового значения
указателя
(**FILE_BEGIN**,
FILE_CURRENT,
FILE_END)

При открытии файла чтение/запись будут происходить с начала файла. Указатель на следующую позицию чтения/записи будет сдвигаться автоматически. Для установки указателя на желаемую позицию используется данная функция.

Системное ПО

Пример

```
typedef struct __MyStruct
{
    int a;
    int b;
    double c;
} MyStruct;

int _tmain(int argc, _TCHAR* argv[])
{
    MyStruct s1,s2,s3;
    //Заполняем две структуры для записи
    s1.a=19;s1.b=34;s1.c=40.1;
    s2.a=8;s2.b=10;s2.c=16.98;

    HANDLE hFile;
    DWORD Bytes;
    hFile=CreateFile(L"file1.dat",GENERIC_READ | GENERIC_WRITE,0,NULL,CREATE_ALWAYS,FILE_ATTRIBUTE_NORMAL,NULL);
    //Записываем s1 и s2
    WriteFile(hFile,&s1,sizeof(MyStruct),&Bytes,NULL);
    WriteFile(hFile,&s2,sizeof(MyStruct),&Bytes,NULL);
    CloseHandle(hFile);

    hFile=CreateFile(L"file1.dat",GENERIC_READ | GENERIC_WRITE,0,NULL,OPEN_EXISTING,FILE_ATTRIBUTE_NORMAL,NULL);
    //Необходимо считать вторую записанную структуру
    SetFilePointer(hFile,sizeof(MyStruct),0,FILE_BEGIN);
    ReadFile(hFile,&s3,sizeof(MyStruct),&Bytes,NULL);
    CloseHandle(hFile);
    printf("%d, %d, %f",s3.a,s3.b,s3.c);//8,10,16.980000
```

Системное ПО

Работа с файлами

Получение информации о файле

```
DWORD WINAPI GetFileAttributes(  
    _In_ LPCTSTR lpFileName  
);
```

Возвращаемое значение – целое число, отдельные биты которого определяют наличие или отсутствие атрибута файла. Для работы с атрибутами Microsoft определила константы.

FILE_ATTRIBUTE_ARCHIVE – архивный

FILE_ATTRIBUTE_HIDDEN – скрытый

FILE_ATTRIBUTE_READONLY – только для чтения

Системное ПО

Работа с файлами

Пример

```
DWORD attr;  
attr=GetFileAttributes(L"file1.dat");  
//Имеет ли файл атрибут "Только для чтения"  
if(attr&FILE_ATTRIBUTE_READONLY)  
    printf("Файл только для чтения!");
```

Используя константу **FILE_ATTRIBUTE_READONLY** проверяем, установлен ли бит, отвечающий за атрибут «Только для чтения»