

Лабораторная работа №5
Управление файлами и каталогами.
(2 часа)

Содержание: реализация основных файловых операций в WinAPI.

Цель: изучить основные функции WinAPI, предназначенные для работы с объектами файловой системы и организации файлового ввода-вывода, сформировать навыки чтения и записи данных в файлы в синхронном режиме.

Практически каждая программа нуждается в долговременном хранении какой-либо информации на внешних носителях. За задачу размещения информации на носителе и предоставления к ней доступа отвечает файловая система. В Windows используется несколько стандартных файловых систем, например, NTFS, различные варианты реализации FAT. Основными объектами файловой системы являются файл и каталог.

В Windows используется понятие **текущего каталога**. Текущий каталог используется для выполнения файловых операций с относительными именами файлов и входит в список поиска при запуске нового процесса. Для определения какой каталог является сейчас текущим, используется вызов функции GetCurrentDirectory:

```
DWORD WINAPI GetCurrentDirectory(  
    __in  DWORD nBufferLength,  
    __out LPTSTR lpBuffer  
);
```

здесь nBufferLength – размер буфера для получения имени текущего каталога в символах, включая место для терминального нуля (помните о размере символа для различных кодировок);

lpBuffer – указатель на буфер в который будет помещен абсолютный путь для текущего каталога.

После выполнения функция возвращает количество символов, скопированное в буфер при успешном завершении или 0 в случае ошибки. Интересным является поведение функции в случае, если был выделен буфер недостаточного для размещения ответа размера. В этом случае вызов GetCurrentDirectory возвращает нужное значение размера буфера, включая терминальный ноль, для корректного размещения строки. Таким образом, удобной является следующая последовательность вызовов:

- 1) Вызов GetCurrentDirectory с минимальным размером буфера и получение точного значения размера для размещения результата;
- 2) Выделение блока памяти нужного размера и повторный вызов GetCurrentDirectory для получения нужного значения.

Следует заметить, что подобная методика достаточно широко встречается в вызовах WinAPI.

Для смены текущего каталога может использоваться функция SetCurrentDirectory:

```
BOOL WINAPI SetCurrentDirectory(  
    __in LPCTSTR lpPathName  
);
```

, здесь lpPathName – новое значение текущего каталога.

Отдельного упоминания заслуживают функции, служащие для получения расположения некоторых известных каталогов:

GetSystemDirectory, возвращающая путь к системному каталогу (содержащему DLL и драйвера, обычно Windows\system32);

GetWindowsDirectory, возвращающая путь к каталогу Windows (обычно C:\Windows);

GetSystemWow64Directory, возвращающая путь к системному каталогу, используемому WOW64 (присутствует на 64-битных версиях Windows);

SHGetKnownFolderPath, возвращающую путь к хорошо известным каталогам (таким как каталог пользовательского профиля, каталог Application Data для текущего пользователя, каталог Program Files и т.д.).

Для создания и удаления каталогов используются вызовы функции:

```
BOOL WINAPI CreateDirectory(  
    __in LPCTSTR lpPathName,  
    __in_opt LPSECURITY_ATTRIBUTES lpSecurityAttributes  
);
```

для создания каталога, который задан путем lpPathName; lpSecurityAttributes, как обычно, задает указатель на структуру атрибутов безопасности для создаваемого каталога (NULL – допустимое значение) и

```
BOOL WINAPI RemoveDirectory(  
    __in LPCTSTR lpPathName  
);
```

для удаления каталога заданного параметром lpPathName.

Задание 1. (Разминка). Напишите программу, выводющую на экран имя текущего каталога, заменяющую его на D:\!TEMP (или C:\Temp).

Для высокоуровневой работы с файлами используются функции CopyFile, DeleteFile, MoveFile и др.

Для создания и открытия файла, как Вам известно из лекционного курса, используется функция:

```

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
);

```

здесь lpFileName имя файла (или устройства) для открытия или создания;

dwDesiredAccess задает желаемые права доступа к файлу (обычно это GENERIC_READ, GENERIC_WRITE или их комбинация GENERIC_READ | GENERIC_WRITE);

dwShareMode задает режим разделения доступа к файлу (может принимать значения: 0 – монопольный режим (другим процессам будет отказано в удалении, чтении или записи файла), FILE_SHARE_DELETE – разрешает операции удаления или переименования файла, FILE_SHARE_READ – разрешает открытие файла на чтение, FILE_SHARE_WRITE – разрешает открытие файла на запись));

dwCreationDisposition – указывает поведение функции в зависимости от того, существует файл или нет (некоторые из возможных значений – CREATE_ALWAYS (создает новый файл всегда, если он существовал, то он перезаписывается), CREATE_NEW (создает новый файл, если он существует, то возвращается ошибка), OPEN_ALWAYS (открывает файл всегда, если он не существует – создает его), OPEN_EXISTING (открывает файл только если он существует);

dwFlagsAndAttributes – задает атрибуты создаваемого файла. Обычно принимает значение FILE_ATTRIBUTE_NORMAL или может содержать комбинацию флагов, некоторые из которых приведены ниже: FILE_ATTRIBUTE_ARCHIVE (установлен атрибут архивный), FILE_ATTRIBUTE_HIDDEN (установлен атрибут скрытый), FILE_ATTRIBUTE_READONLY (установлен атрибут только для чтения) и т.д. Среди прочих значений этого параметра следует отметить флаг FILE_FLAG_OVERLAPPED, который используется для открытия или создания файла, предназначенного для обработки в асинхронном режиме;

hTemplateFile – может задавать дескриптор файла с доступом GENERIC_READ, который будет использован для создания нового файла с такими же атрибутами как у него.

Например, для открытия файла C:\Windows\log.log для записи в него информации в синхронном режиме может быть использован следующий вызов:

```
CreateFile("C:\\Windows\\log.log", GENERIC_WRITE,  
FILE_SHARE_READ, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL,  
NULL);
```

Для закрытия файла после выполнения нужных действий используется знакомая Вам функция `CloseHandle`.

Для чтения и записи данных используются функции `ReadFile` и `WriteFile`. Обычный порядок их использования следующий:

- Выделить буфер нужного размера;
 - Открыть файл с нужными правами;
 - Установить указатель в файле в нужную позицию (если файл читается или записывается последовательно от самого начала – это делать не нужно);
 - Прочитать данные;
 - Обработать содержимое буфера;
- (последние два пункта в случае записи в файл поменяются местами:
- Обработать и записать данные в буфер;
 - Записать данные в файл;)
 - Закрыть файл.

Для установки указателя в файле используется функция `SetFilePointer`:

```
DWORD WINAPI SetFilePointer(  
    __in HANDLE hFile,  
    __in LONG lDistanceToMove,  
    __inout_opt PLONG lpDistanceToMoveHigh,  
    __in DWORD dwMoveMethod  
);
```

здесь `hFile` – дескриптор открытого файла;

`lDistanceToMove` – младшая 32-битная часть знакового числа, задающего дистанцию перехода в пределах файла;

`lpDistanceToMoveHigh` – старшая 32-битная часть знакового числа, задающего дистанцию перехода в пределах файла;

`dwMoveMethod` – указывает как отсчитывается начало перехода: `FILE_BEGIN` – от начала файла, `FILE_CURRENT` – от текущего положения, `FILE_END` – от конца файла.

Задание 2. Напишите программу, которая выполняет следующие действия:

- Создаёт временный каталог на локальном жестком диске (например, в `D:\\!Temp\\`);
- копирует из каталога `Z:\\Новый\\2kurs\\SysProg` файл `test.info` во временный каталог;

- открывает этот файл на чтение;
- читает из этого файла блок информации в 126 байт начиная с 271014 байта от начала файла;
- записывает его в файл answer.txt в том же каталоге.

Задание 3. (исследовательское, по желанию). Напишите программу, выполняющую создание копии указанного файла при помощи последовательности: создание нового файла, чтение данных из исходного файла и запись в новый файл с помощью функции CreateFile, ReadFile и WriteFile. Сравните быстродействие полученной программы со стандартной функцией копирования файлов.