

# 同步异步IO



日成蝶—Windows 高级编程

七日做茧，一朝成蝶！



主讲：袁春旭

**打开关闭设备**

# 打开关闭设备

设备	常见用途
文件	永久存储任何数据
目录	属性和文件压缩的设置
逻辑磁盘驱动器	格式化驱动器
物理磁盘驱动器	访问分区表
串口	通过电话线传输数据
并口	将数据传输至打印机

# 打开关闭设备

设备	常见用途
邮件槽	一对多数据传输，通常是通过网络达到另一台运行 windows 的机器上。
命名管道	一对一数据传输，通常是通过网络达到另一台运行 windows 的机器上。
匿名管道	单机上一对一传输，不会跨网络传输。
套接字	报文或数据流的传输，通常是通过网络传到任何支持套接字的机器上（可以是 windows 或其它操作系统）
控制台	文本串口的屏幕缓存。

# 打开关闭设备

设备	打开关闭文件所用API
文件	CreateFile CloseHandle
目录	CreateFile CloseHandle
逻辑磁盘驱动器	CreateFile CloseHandle
物理磁盘驱动器	CreateFile CloseHandle
串口	CreateFile CloseHandle
并口	CreateFile CloseHandle

# 打开关闭设备

设备	打开文件所用API
邮件槽	CreateMailslot      CloseHandle
命名管道	CreateNamedPipe    CloseHandle
匿名管道	CreateFile            CloseHandle
套接字	socket, accept    closesocket
控制台	CreateConsoleScreenBuffer    GetStdHandle    CloseHandle

# 打开关闭设备

HANDLE WINAPI CreateFile(

LPCTSTR lpFileName, //设备类型, 设备实例

//0, GENERIC\_READ, GENERIC\_WRITE, GENERIC\_READ|GENERIC\_WRITE

DWORD dwDesiredAccess,

//0, FILE\_SHARE\_READ, FILE\_SHARE\_WRITE, FILE\_SHARE\_DELETE

DWORD dwShareMode,

LPSECURITY\_ATTRIBUTES lpSecurityAttributes, //安全属性, NULL为默认

//CREATE\_NEW, CREATE\_ALWAYS, OPEN\_EXISTING, OPEN\_ALWAYS, TRUNCATE\_EXISTING

DWORD dwCreationDisposition,

DWORD dwFlagsAndAttributes, //状态微调, 设置属性

HANDLE hTemplateFile

);



# 打开关闭设备

dwFlagsAndAttributes高速缓存相关的标志

取值	描述
FILE_FLAG_NO_BUFFERING	不使用缓存，读取大数据时使用
FILE_FLAG_SEQUENTIAL_SCAN	顺序访问时高效
FILE_FLAG_RANDOM_ACCESS	随机访问时使用
FILE_FLAG_WRITE_THROUGH	修改时不缓存直接写入设备

# 打开关闭设备

dwFlagsAndAttributes其它标志

取值	描述
FILE_FLAG_DELETE_ON_CLOSE	关闭后自动删除，常FILE_ATTRIBUTE_TEMPORARY共用
FILE_FLAG_BACKUP_SEMANTICS	用于备份恢复
FILE_FLAG_POSIX_SEMANTICS	创建及查找文件区分大小写
FILE_FLAG_OPEN_REPARSE_POINT	忽略文件重解析属性
FILE_FLAG_OPEN_NO_RECALL	不从脱机存储恢复到联机存储
<b>FILE_FLAG_OVERLAPPED</b>	<b>异步访问</b>

# 打开关闭设备

## dwFlagsAndAttributes文件属性标志

取值	描述
FILE_ATTRIBUTE_ARCHIVE	文件为存档文件
FILE_ATTRIBUTE_ENCRYPTED	文件为加密文件
FILE_ATTRIBUTE_HIDDEN	文件为隐藏属性
FILE_ATTRIBUTE_NORMAL	正常文件
FILE_ATTRIBUTE_NOT_CONTENT_INDEXED	内容索引服务不对此文件做索引

# 打开关闭设备

## dwFlagsAndAttributes文件属性标志

取值	描述
FILE_ATTRIBUTE_OFFLINE	文件虽然在，内容已经转移到脱机存储器
FILE_ATTRIBUTE_READONLY	文件是只读的。
FILE_ATTRIBUTE_STEM	文件专供操作系统使用
FILE_ATTRIBUTE_TEMPORARY	临时文件

# 获取设备类型

# 获取设备类型

```
DWORD GetFileType(  
    HANDLE hFile  
);
```


返回值	描述
FILE_TYPE_UNKNOWN	未知类型 0x0000
FILE_TYPE_DISK	磁盘文件 0x0001
FILE_TYPE_CHAR	并口设备或控制台 0x0002
FILE_TYPE_PIPE	命名管道或匿名管道 0x0003

# 打开文件的正确姿势

# 打开文件的正确姿势

```
#include <windows.h>
int main(int argc, char *argv[])
{
    HANDLE hFile = CreateFile(...);
    if(NULL != hFile)
    {
    }
    .....
}
```

```
#include <windows.h>
int main(int argc, char *argv[])
{
    HANDLE hFile = CreateFile(...);
    if(INVALID_HANDLE_VALUE != hFile)
    {
    }
    .....
}
```





# 同步设备IO

# 同步设备IO

```
BOOL ReadFile(  
    HANDLE    hFile,  
    LPVOID    lpBuffer,  
    DWORD     nNumberOfBytesToRead,  
    LPDWORD   lpNumberOfBytesRead,  
    LPOVERLAPPED lpOverlapped  
);
```

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

# 取得文件大小

# 取得文件大小

```
BOOL GetFileSizeEx(  
    HANDLE hFile,  
    PLARGE_INTEGER pliFileSize  
)
```

```
typedef union _LARGE_INTEGER {  
    struct {  
        DWORD LowPart;  
        LONG HighPart;  
    }  
    LONG LONG QuadPart;  
} LARGE_INTEGER, PLARGE_INTEGER;
```

# 设置文件指针位置

# 设置文件指针位置

```
BOOL SetFilePointerEx(  
    HANDLE      hFile,  
    LARGE_INTEGER liDistanceToMove, //指针移动的字节数  
    PLARGE_INTEGER lpNewFilePointer, //移动后的新位置  
    DWORD      dwMoveMethod        //FILE_BEGIN FILE_CURRENT FILE_END  
);
```

dwMoveMethod取值	含义
FILE_BEGIN	liDistanceToMove的值即为文件当前指针
FILE_CURRENT	liDistanceToMove与当前指针相加作为新的文件指针
FILE_END	liDistanceToMove与文件逻辑大小相加作为新的文件指针

# 设置文件尾

# 设置文件尾

```
BOOL SetEndOfFile(HANDLE hFile)
```

该函数可以强制使文件变得更大或更小。

例如，可以强制将文件大小设置为1024

```
HANDLE hFile = CreateFile(...);
```

```
LARGE_INTEGER liDistanceToMove;
```

```
liDistanceToMove.QuadPart = 1024;
```

```
SetFilePointerEx(hFile, liDistanceToMove, NULL, FILE_BEGIN);
```

```
SetEndOfFile(hFile);
```

```
CloseHandle(hFile);
```



# 编码实战



# Thank You !