

文件与流的关系

- 程序通过**文件打开**操作将流与设备联系起来，文件打开后，可在程序和文件之间交换数据
 - * 由程序在磁盘上建立文件(建立文件)
 - * 文件打开后，通过写操作将数据存入该文件(写入数据)
 - * 由程序打开磁盘上的某个已有文件，通过读操作将文件中的数据读入内存供程序使用(读取数据)
- 程序通过**文件关闭**操作断开流与文件的联系

文件指针(File Pointer)

- C程序中流的打开和关闭是通过文件指针实现的
- 文件指针的类型为**FILE ***
 - * **FILE** * fp ;
 - 定义了**FILE**型指针变量*fp, 标识一个特定的磁盘文件
 - 与文件相关联的每个流都有一个**FILE**类型的控制结构, 定义有关文件操作的信息, 用户绝对不应修改

文件的打开

- 格式: `FILE *fopen(const char *filename, const char *mode);`

- * `FILE *fp;`

- * `fp = fopen("test.txt", "r");`

- * 返回值为指向此文件的指针 `*fp`

- * 如果打开失败(文件损坏或不存在), 返回值为 **NULL**

- 文件打开后一定要检查是否打开成功

```
if (fp == NULL)
{
    printf("Failure to open test.txt!\n");
    exit(0);
}
```

文件的打开

- **filename**是文件名

- * 包含路径。如果不含路径，表示打开当前目录下的文件

- * `fp = fopen("D:\newproject\test.txt", "r");`

- * 编译器会将‘\’看成转义字符，例如：`\n`和`\t`，为此“\\”

- * `fp = fopen("D:\\newproject\\test.txt", "r");`

- Windows用反斜杠\分割路径

- UNIX用斜杠/

文件的打开

文件打开方式(mode)

r	只读	必须是已存在的文件
w	只写	无论该文件是否存在，都新建一个文件
a	追加	向文本文件尾添加数据，该文件必须已经存在
r+	读写	打开一个已存在的文件，用于读写
w+	读写	建立一个新文件，可读可写
a+	读写	向文件尾追加数据，也可读

对应文本文件

以二进制方式打开文件
原汁原味地体现文件内容

wb

ab

rb+

wb+

ab+

文件的打开

■ 文件打开方式(mode)

r	只读	必须是已存在的文件
w	只写	不论该文件是否存在, 都新建一个文件
a	追加	向文本文件尾添加数据, 该文件必须存在
r+	读写	打开一个已存在的文件, 用于读写
w+	读写	建立一个新文件, 可读可写
a+	读写	向文件尾追加数据, 也可读

若文件不存在, **w**新建一个文件,
若文件存在, **w**会将原文件内容覆盖

用**a**打开文件, 要求该文件必须存在,
保留原文件内容, 在文件末尾添加

w和**a**的
区别?



文件的打开

■ 文件打开方式(mode)

r	只读	必须是已存在的文件
w	只写	不论该文件是否存在, 都新建一个文件
a	追加	向文本文件尾添加数据, 该文件必须存在
r+	读写	打开一个已存在的文件, 用于读写
w+	读写	建立一个新文件, 可读可写
a+	读写	向文件尾追加数据, 也可读

若文件不存在: **r+** 打开会失败,
w+ 则会新建一个文件

若文件存在: **r+** 不清空文件,
w+ 则清空文件, 因此
当文件存在时要慎用 **w+**

r+和**w+**的
区别?



文件的关闭

- 格式: `int fclose(FILE *fp);`
 - * 把遗留在缓冲区中的数据写入文件, 实施操作系统级的关闭操作
 - * 同时, 释放与流联系的文件控制块FCB, 以便以后重复使用
- `fclose`函数的返回值
 - * 若成功执行了关闭操作, 返回值为0
 - * 否则返回为非零值, 表示关闭时有错误

例如: 驱动器中无盘或盘空间不够时文件操作失败, 文件关闭失败会导致数据丢失、文件破坏, 甚至程序出现随机错误

文件的关闭

- 文件用完一定要关闭。
 - * 否则, 可能引起数据丢失
 - * 甚至影响其他文件的打开
 - * 多数情况下, 系统限制同时打开状态的文件总数。
 - * 因此, 打开文件前先关闭无用文件是必要的。

