



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

编程操作永久性数据

- ◆ 1、文件系统概念
- ◆ 2、流式文件
- ◆ 3、文件指针
- ◆ 4、文件打开

第10章 文件

- ▶ 计算机信息根据存储时间，可以分为临时性信息和永久性信息。临时性信息存储于计算机系统临时存储设备（如内存）中，在程序运行结束或系统断电时数据信息会消失。永久性信息存储于计算机系统持久性存储设备（如磁盘）中，在这些设备上的数据信息可以长久地保存下来。

10.1 文件概述

- ▶ 程序常常需要将一些数据信息（如运行结果）永久性地保存下来，或者从永久性信息中读取有用的数据（如历史记录），这些都需要进行文件操作。C语言文件操作是通过函数实现的。

10.1.1 文件系统

- ▶ 文件是指存放在磁盘上数据的集合。操作系统以文件为单位对这些数据进行管理。
- ▶ 如果想得到存在磁盘上的数据，必须先按文件名找到指定的文件，然后再从该文件中读取数据。要向磁盘上存放数据也必须先以文件名为标识创建一个文件，才能向它输出数据。

10.1.1 文件系统

▶ 1. 缓冲文件系统

- ▶ 缓冲文件系统是指系统自动地在内存中为每一个正在使用的文件开辟一个缓冲区。从内存向磁盘输出数据时先存放到这个缓冲区，装满缓冲区或发出强制写入指令后才将数据存储到磁盘上。如果从磁盘向内存读入数据，则是一次从磁盘文件中读入一批数据到内存缓冲区中充满，然后再从缓冲区逐个将数据送到程序中。内存缓冲区的大小可以调节，一般为1024字节（1KB）。

10.1.1 文件系统

- ▶ 缓冲文件系统的优点是存取效率高，缺点是当输出数据时，在缓冲区尚未装满且发生系统断电等情况，数据会丢失，因此需要程序有特别的处理策略。
- ▶ C语言标准文件处理采用缓冲文件系统。

10.1.1 文件系统

- ▶ 2. 非缓冲文件系统
- ▶ 非缓冲文件系统是指系统不使用缓冲区作为程序数据和磁盘文件数据的交换区，写入数据时直接存放到磁盘文件上，读入数据时也是直接从磁盘文件上取出。
- ▶ 非缓冲文件系统的优点是不会发生数据丢失，缺点是存取效率低。

10.1.1 文件系统

- ▶ 3. 文本文件
- ▶ 文本（text）文件对于ASCII字符集而言，文件中每个字节存放的是一个ASCII码，表示一个字符；对于像汉字、日韩文字等字符集而言，使用双字节存放字符。文本文件有UTF-8、UTF-16、UNICODE等编码格式。本书不讨论复杂编码格式，仅以ASCII文本文件（又称DOS格式文本文件）为例。

10.1.1 文件系统

- ▶ 例如字符型数据123输出到文本文件中，存放的是"123"的ASCII码序列，即31H、32H、33H（十六进制），占用文件3个字节。
- ▶ ASCII文本文件，数据以字符形式表示，因而便于按字符方式逐个处理，也便于打印输出字符。但ASCII文本文件一般占用存储空间较多，且存在编码转换的运行开销。

10.1.1 文件系统

- ▶ 4. 二进制文件
- ▶ 二进制文件是将数据以内存中的存储形式直接存放到磁盘上。用二进制形式输出数据，可以节省存储空间和避免编码转换。由于一个字节并不对应一个字符，所以不能直接打印输出或编辑二进制文件。
- ▶ 例如字符型数据123输出到二进制文件中，存放的是123的值，即7BH（十六进制），占用文件1个字节。

10.1.2 流式文件

- ▶ 流是磁盘或其它外围设备存取数据的逻辑接口，它提供或存储数据。提供数据的称为输入流，存储数据的称为输出流。
- ▶ C语言在处理文件时，并不区分文件类型，都将其看作是一个字符流，按字符进行处理。即文件由一个一个字符（或字节）的数据顺序组成。输入输出字符流的开始和结束只由程序控制而不受物理符号（如回车符）的控制，因此也把这种文件称作流式文件。

10.1.3 文件指针

- ▶ 在C语言中用一个指针指向一个文件，称为文件指针（或文件句柄）。文件指针是一个指向结构体类型的指针，其类型声明在头文件stdio.h中，一般形式为：

```
typedef struct _iobuf {  
    char *_ptr; //文件位置指针  
    int _cnt; //缓冲区剩余字符数  
    char *_base; //缓冲区首地址  
    int _flag; //文件状态标志  
    int _file; //文件有效性检查  
    int _charbuf; //若无缓冲区则不读取字符  
    int _bufsiz; //缓冲区大小  
    char *_tmpfname; //临时文件名  
} FILE;
```

10.1.3 文件指针

- ▶ 对于每个正在使用的文件，都有一个已分配内存的文件结构用来存放文件的状态、当前位置、缓冲区大小等相关信息。一般情况下，程序很少直接使用这些信息，但文件操作的各种函数都需要它们（通过文件指针引用）。

10.1.3 文件指针

- ▶ 例如定义一个文件指针变量
- ▶ `FILE *fp;` // 定义一个文件指针变量
- ▶ 一旦通过函数打开或创建一个文件时，fp就指向这个文件，通过fp就可以对文件进行各种操作。也就是说，通过文件指针变量能够找到与之关联的文件。
- ▶ 显然，程序若要使用多个文件，就应定义多个文件指针变量指向这些文件，且通过关联的文件指针变量来操作具体的一个文件。

10.1.3 文件指针

- ▶ C语言文件操作函数定义在标准输入输出库中，其头文件为 `stdio.h`。

10.2 文件打开与关闭

- ▶ 文件操作之前需要打开该文件，在使用结束之后需要关闭该文件。所谓打开文件，实际上是建立文件的各种信息，使文件指针指向该文件，以便进行其它操作。关闭文件则断开文件指针与文件之间的联系，也就禁止再对该文件进行操作。

10.2.1 文件打开

- fopen函数用来打开或创建一个文件，其函数原型为：

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

- 参数filename表示文件名，可以是相对路径，如
- "A.TXT"（当前目录下A.TXT文件）
 - "SUB\\A.TXT"（SUB子目录下A.TXT文件）
 - "..\\A.TXT"（上一级目录下A.TXT文件）
 - 也可以是绝对路径，如
 - "C:\\A.TXT"（C盘根目录下A.TXT文件）
 - "D:A.TXT"（D盘当前目录下A.TXT文件）

10.2.1 文件打开

- ▶ 请注意，文件系统路径表示的反斜线（\）写成字符串时应该用转义字符（\\）。参数filename可以是字符串常量，也可以是字符串变量。
- ▶ 参数mode表示操作模式，只能是下表所列的模式字符串。

10.2.1 文件打开

表10-1 文件工作模式

模式字符串		含义
文本文件	二进制文件	
"r"或"rt"	"rb"	打开一个文件用于只读数据，文件不存在时打开失败。
"r+"或"rt+"或 "r+t"	"rb+"或"r+b"	打开一个文件用于读写数据，文件不存在时打开失败。
"w"或"wt"	"wb"	创建一个新文件用于只写数据，新文件会覆盖旧文件。
"w+"或"wt+"或 "w+t"	"wb+"或"w+b"	创建一个新文件用于读写数据，新文件会覆盖旧文件。
"a"或"at"	"ab"	打开一个文件用于追加数据，文件不存在时创建新文件。

10.2.1 文件打开

续表10-1 文件工作模式

模式字符串		含义
"a+"或"at+"或 "a+t"	"ab+"或"a+b"	打开一个文件用于追加和读数据，文件不存在时 创建新文件。

10.2.1 文件打开

- ▶ fopen函数根据操作模式，打开或创建指定文件名的文件，返回与之关联的文件指针。如果打开或创建文件成功，则返回有效指针，否则返回NULL指针。程序可以根据该指针是否有效（非NULL）决定后续操作。

10.2.1 文件打开

► 示例

```
FILE *fp1,*fp2; //定义文件指针变量  
fp1=fopen("IN.TXT","r"); //打开一个（已存在的）文本文件IN.TXT  
fp2=fopen("OUT.DAT","wb"); //创建一个新二进制文件OUT.DAT
```

10.2.1 文件打开

- ▶ 关于操作模式的说明：
- ▶ (1) 将文件打开只读和读写时，如果文件不存在时会失败，函数返回NULL指针，后续不能对这个文件作任何操作。
文件不存在的原因有：
- ▶ ①文件名是错误的；
- ▶ ②按文件名中的路径找不到该文件；
- ▶ ③磁盘设备不存在或物理损坏。

10.2.1 文件打开

- ▶ (2) 通常，创建新文件总是成功的，除非存储设备物理异常。例如往系统不存在的F盘里写文件，往只读设备里写文件。创建新文件时若磁盘上已有同名文件，则旧文件会被删除，重建一个新文件。

10.2.1 文件打开

- ▶ (3) 追加数据是指打开文件将文件位置指针移动到文件末尾，向文件新增数据。

10.2.1 文件打开

- ▶ 通常，系统会为每个程序自动定义并打开三个标准文件，分别是标准输入文件stdin（键盘）、标准输出文件stdout（显示器）、标准出错输出stderr（显示器），程序可直接使用它们。例如从文件指针stdin中读取数据就是从键盘上输入数据。

CP 程序设计