

C语言程序设计(第8章输入输出和文件系统) PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/135/2021\\_2022\\_C\\_E8\\_AF\\_AD\\_E8\\_A8\\_80\\_E7\\_A8\\_8B\\_c97\\_135072.htm](https://www.100test.com/kao_ti2020/135/2021_2022_C_E8_AF_AD_E8_A8_80_E7_A8_8B_c97_135072.htm) 8.1 缓冲文件系统缓冲

文件系统的特点是：在内存开辟一个“缓冲区”，为程序中的每一个文件使用，当执行读文件的操作时，从磁盘文件将数据先读入内存“缓冲区”，装满后再从内存“缓冲区”依此读入接收的变量。执行写文件的操作时，先将数据写入内存“缓冲区”，待内存“缓冲区”装满后再写入文件。由此可以看出，内存“缓冲区”的大小，影响着实际操作外存的次数，内存“缓冲区”越大，则操作外存的次数就少，执行速度就快、效率高。一般来说，文件“缓冲区”的大小随机器而定。

8.1.1 文件的打开与关闭 任何关于文件的操作都要先打开文件，再对文件进行读写，操作完毕后，要关闭文件

。1. 文件类型指针 人们在操作文件时，通常都关心文件的属性，如文件的名称、文件的性质、文件的当前状态等。对缓冲文件系统来说，上述特性都是要仔细考虑的。ANSI C为每个被使用的文件在内存开辟一块用于存放上述信息的小区，利用一个结构体类型的变量存放。该变量的结构体类型由系统取名为FILE，在头文件stdio.h中定义如下：

```
typedef struct{
int_fd. /*文件号*/ int_cleft. /*缓冲区中的剩余字符*/ int_mode.
/*文件的操作模式*/ char *_next. /*下一个字符的位置*/ char
*_buff. /*文件缓冲区的位置*/}FILE.
```

在操作文件以前，应先定义文件变量指针：FILE \*fp1,fp2. 按照上面的定义，fp1和fp2均为指向结构体类型的指针变量，分别指向一个可操作的文件，换句话说，一个文件有一个文件变量指针，今后对文件

的访问，会转化为针对文件变量指针的操作。

### 2. 文件的打开

ANSI C 提供了打开文件的函数：`FILE *fopen(char *fname, char *mode)` 函数原型在 `stdio.h` 文件中，`fopen()` 打开一个 `fname` 指向的外部文件，返回与它相连接的流。`fname` 是字符串，应是一个合法的文件名，还可以指明文件路径。对文件的操作模式由 `mode` 决定，`mode` 也是字符串，由表 8-1 给出 `mode` 的取值表。

Mode	含义
<code>r</code>	打开一个文本文件只读
<code>w</code>	打开一个文本文件只写
<code>a</code>	打开一个文本文件在尾部追加
<code>rb</code>	打开一个只读的二进制文件
<code>wb</code>	打开一个只写的二进制文件
<code>ab</code>	对二进制文件追加
<code>r+</code>	打开一个可读/写的文本文件
<code>w+</code>	创建一个新的可读/写的文本文件
<code>a+</code>	打开一个可读/写的文本文件
<code>rb+</code>	打开一个可读/写的二进制文件
<code>wb+</code>	创建一个新的可读/写的二进制文件
<code>ab+</code>	打开一个可读/写的二进制文件

如表 8-1 所示，文件的操作方式有文本文件和二进制文件两种，打开文件的正确方法如下例所示：

```
#include FILE
*fp.If((fp=fopen("test.txt","w"))==NULL){ /*创建一个只写的新文本文件*/ printf("cannot open file \n"). exit(0).} 这种方法能发现打开文件时的错误。在开始写文件之前检查诸如文件是否有写保护，磁盘是否已写满等，因为函数会返回一个空指针 NULL，NULL 值在 stdio.h 中定义为 0。事实上打开文件是要向编译系统说明三个信息：需要访问的外部文件是哪一个。打开文件后要执行读或写即选择操作方式。确定哪一个文件指针指向该文件。对打开文件所选择的操作方式来说，一经说明不能改变，除非关闭文件后重新打开。是只读就不能对其写操作，对已存文件如以新文件方式打开，则信息必丢失。

### 3. 文件的关闭



ANSI C 提供了关闭文件的函数：int


```

fclose(FILE \*stream)fclose( )函数关闭与stream相连接的文件，并把它的缓冲区内容全部写出。在fclose( )函数调用以后，流stream与此文件无关，同时原自动分配的缓冲区也失去定位。fclose( )函数关闭文件操作成功后，函数返回0；失败则返回非零值。[例8-1] 打开和关闭一个可读可写的二进制文件

```
: #include main( ){ FILE *fp. If  
((fp=fopen("test.dat","rb"))==NULL) { printf("cannot open file\n").  
exit(0). } /*写入对文件执行读写的代码 */ if (fclose(fp))  
printf("file close error!\n").}
```

8.1.2 文件的读写 当文件按指定的工作方式打开以后，就可以执行对文件的读和写。下面按文件的性质分类进行操作。针对文本文件和二进制文件的不同性质，对文本文件来说，可按字符读写或按字符串读写；对二进制文件来说，可进行成块的读写或格式化的读写。 100Test  
下载频道开通，各类考试题目直接下载。详细请访问  
[www.100test.com](http://www.100test.com)