1. fopen打开一个文件

FILE\* fp;

fp = fopen("文件路径",r);

if(fp == NULL) {

printf("文件打开错误！\n");

return;

}

2、fgetc读取一个字符

//返回字符，文件尾返回EOF；#define EOF -1

int c**;**

**while((**c **=** fgetc**(**fp**))!=**EOF**)**

printf**(**"%c"**,**c**);**

3、fgets读取一个字符串

读取成功，返回缓冲区地址，读取失败或文件尾，返回NULL

缓冲区10个单元，但是存储9个字符，最后一个存储’\0’

char str[10];

char\* str0;

while((str0 = fgets(str,10,fp))!=NULL) {

while(\*str0!='\0') {

printf("%c",str0);

str0++;

}

}

4、fseek重定位文件流的读写位置

int fseek("fp",long offset,int whence);

将读写位置重定位在whence后的offset处。

操作成功返回0，失败返回-1

1) 欲将读写位置移动到文件开头时:fseek(FILE \*stream, 0, SEEK\_SET);

2) 欲将读写位置移动到文件尾时:fseek(FILE\*stream, 0, SEEK\_END);

5、ftell 当前文件流指针位置

long ftell()返回FILE[指针](http://baike.baidu.com/view/159417.htm)当前位置。

6、fputc 将一个字符写入文件

int fputc(int c,fp) 返回写入的字符，失败返回EOF

7、fputs 将一个字符串写入文件

int fputs(const char\* str,fp) 返回写入的字符数，失败，返回EOF

8、fread和发fwrite

(1)size\_t fread ( void \* ptr, size\_t size, size\_t count, FILE \* stream );

ptr：指向保存结果的指针；size：每个数据类型的大小；count：数据的个数；stream：文件指针

函数返回读取数据的个数。

(2)size\_t fwrite ( const void \* ptr, size\_t size, size\_t count, FILE \* stream );

ptr：指向保存数据的指针；size：每个数据类型的大小；count：数据的个数；stream：文件指针

函数返回写入数据的个数。

C语言采用文本方式和二进制方式打开文件的区别分析

稍微了解C程序设计的人都知道，文本文件和二进制文件在计算机上面都是以0,1存储的，那么两者怎么还存在差别呢？对于编程人员来说，文本文件和二进制文件就是一个声明，指明了你应该以什么方式(文本方式/二进制)打开这个文件，用什么函数读写这个文件(读写函数)，怎么判断读到这个文件结尾等。

具体分析如下：

**一、以哪种方式打开一个文件**：

ANSI C规定了标准输入输出函数库，用 fopen()函数打开文件。fopen()函数的调用方式一般为：  
FILE \*fp;  
fp=fopen(文件名，使用文件方式)；  
使用文件方式见下表：

|  |  |
| --- | --- |
| 使用文件方式 | 含义 |
| "r"（只读） | 为输入打开一个文本文件 |
| "w"（只写） | 为输出打开一个文本文件 |
| "a"（追加） | 为追加打开一个文本文件 |
| "rb"（只读） | 为输入打开一个二进制文件 |
| "wb"（只写） | 为输出打开一个二进制文件 |
| "ab"（追加） | 为追加打开一个二进制文件 |
| "r+"（读写） | 为读／写打开一个文本文件 |
| "w+"（读写） | 为读／写创建一个文本文件 |
| "a+"（读写） | 为读／写打开一个文本文件 |
| "rb+"（读写） | 为读／写打开一个二进制文件 |
| "wb+"（读写） | 为读／写创建一个二进制文件 |
| "ab+"（读写） | 为读／写打开一个二进制文件 |

同一个文件从磁盘读取文件到内存(程序数据区或者缓存区)时，两种方式下，内存中的内容一般不相同，这就是两种打开方式的实质性差别。  
这里要说一个背景，那就是在windows下，它会做一个处理，就是写文件时，换行符会被转换成回车，换行符存在磁盘文件上，而读磁盘上的文件时，它又会进行逆处理，就是把文件中连续的回车，换行符转换成换行符。  
因此，在读取一个磁盘文件时，文本方式读取到文件内容很有可能会比二进制文件短，因为文本方式读取要把回车，换行两个字符变成一个字符，相当于截短了文件。但是为什么仅仅是可能呢？因为可能文本中不存在连着的45，42这两个字节(45是CR回车的ASCII码，42是换行符CL的ASCII码)，也就不存在“截短”操作了，因此读到的内容是一样的。  
具体的来说，文件文件(以文本方式写的)，最好以文本方式读。二进制文件(以二进制方式写的)，最好以二进制方式读。不然可能会不正确。

**二、以什么函数读写文件**

数据怎么在磁盘上写不是由文件打开方式决定的，而是由写函数决定的。数据怎么从磁盘上读也不是由文件打开方式决定的，而是由读函数决定的。  
上面说的数据怎么写是指，一种类型的变量是怎么存的？比如int 12，可以直接存12的二进制码(4个字节)，也可以存字符1，字符2.  
数据怎么读的是指，我要读一个int变量，是直接读sizeof(int)个字节，还是一个字符一个字符的读，直到读到的字符不是数字字符。

C里面有两组文件读写函数恰好支持上面两种方式的读写：

1.fread(buffer,size,count,fp)，fwrite(buffer,size,count,fp)。用来读写一个数据块。它对应的是第一种存储方式。直接按类型的字节长度指定读写的字节数。

2.fprintf函数和fscanf函数.它对应的是第二种读写方式。即以字符的方式读写。(fprintf函数、fscanf函数与printf函数、scanf函数的作用相仿，都是格式化读写函数。fprintf和fscanf函数的读写对象是磁盘文件，而printf和scanf函数的读写对象是终端。)  
它们的一般调用格式为：

[?](http://www.jb51.net/article/52302.htm)

|  |  |
| --- | --- |
| 1  2 | fprintf(文件指针，格式字符串，输出列表);  fscanf (文件指针，格式字符串，输入列表); |

**三、怎么判断文件尾**

在C语言，或更精确地说成 C标准函式库中，有一个特别的字符EOF(stdio.h中这个定义 #define EOF (-1) )，它表示:文件结束符(end of file)。在while循环中以EOF作为文件结束标志，这种以EOF作为文件结束标志的文件，必须是文本文件。在文本文件中，数据都是以字符的ASCII代码值的形式存放。我们知道，ASCII代码值的范围是0~255，不可能出现-1，因此可以用EOF作为文件结束标志。  
  
但是，C语言中，当把数据以二进制形式存放到文件中时，就会有-1值的出现，此时不能采用EOF作为二进制文件的结束标志。为解决这个问题，ANSI C提供一个feof函数，用来判断文件是否结束。如果遇到文件结束，函数feof(fp)的值为1，否则为0.  
feof函数既可用以判断二进制文件是否结束，也可以用以判断文本文件是否结束。但是要注意feof用以判断文本文件尾时，如果代码编写不当，可能会把文本文中中的文件结束符EOF也读取出来了;具体可以参考http://baike.baidu.com/view/656648.htm中feof函数的用法。

**四、知道一个文件是文本文件，还是二进制文件，更多的“提醒”我们，应该选择哪种读写函数。**

正如前文所说的，数据怎么存不是由文件打开方式决定的，而是由读写函数决定的。  
比如说，我们以二进制文件的方式打开一个文件(实际上只是指明了要进行换行符的转换)，它更多的是代表一种理念(虚的)：我“希望”这个文件里面的数据是这样的，int类型占4字节，char占1个字节。这种模式下，我用fread(buffer,size0f(int),1,fp)读取一个int到int变量中。

**这里需要记住：**

我们在对一个文件进行操作以前，首先，我们要清楚这个文件到底是文本文件还是二进制文件。文件文件用文本方式打开，二进制文件用二进制方式打开  
如果我们要操作一个二进制文件，那么我们就以二进制方式打开(理论上也可以以文件方式打开，但是如果写的二进制数据里面有45时，会转化成45,42存储，如前文所述。这是很有可能发生的)。同时读写的时候用fread，fwrite这两个函数。  
如果我要操作一个文本文件，那么我们就以文本的方式打开(理论上也可以以二进制方式打开，但是不保险)。同时读写的时候用读写字符的那些函数fprintf,fscanf ,fgetc,fputc,putw,getw,fgetc,fputs.