# **C语言文件操作**

## **什么是文件？**

**文件有不同的类型，在程序设计中，主要用到两种文件：**

1. **程序文件。包括源程序文件（后缀名为.c）、目标文件（后缀名为.obj）、可执行文件（后缀名为.exe）等。这一类型的文件主要用于存储程序代码。**
2. **数据文件。此文件的内容不是程序，而是程序运行时读写的数据，比如程序运行过程中输出到磁盘或其他设备上的数据，或在程序运行过程中供程序读取的数据。**

**这里C语言的文件操作主要是对数据文件的操作。**

**在之前程序中所处理的数据的输入和输出都是以终端为对象的，都是从键盘输入数据，然后运行结果输出到终端显示器上。实际上，我们有时候需要将一些数据（程序运行的最终结果或者中间数据）保存起来，方便以后需要时再调用，而这就需要用到磁盘文件了。**

## **文件的概念**

1. **文件名**

**每一个文件都需要一个唯一的文件标识，以便用户使用，就像我们的变量名一样，同一程序中不能有相同的变量名。**

**文件标识也称为文件名，它由3部分组成：**

**①文件路径：表示文件在外存设备中的存储位置；**

**②文件名主干：表示文件的名字，可由用户自定义，命名规则应遵循标识符的命名规则。**

**③文件后缀：表示文件的性质，也称为文件的格式，用于描述文件的类型（如txt、ppt等）。**

**文件路径能唯一标识文件在外存中的位置。**

**如：D:\C++\VSproject\TEXT\text.c**

### **文件的分类**

**根据数据的组织形式，数据文件可分为ASCII文件和二进制文件。**

**数据在内存中是以二进制形式存储的，如果不加转换的输出到外存，就是二进制文件，可以认为它是存储在内存的数据的映像，所以称之为映像文件。如果要求在外存上以ASCII码形式存储，就需要在存储前进行转换。ASCII文件又称为文本文件，每一个字节存放一个字符的ACSII码。**

### 3、文件存储方法的区别

**一个数据在磁盘上存储，字符一律以ASCII形式存储，数值型既可以用ASCII形式存储也可以用二进制形式存储。如整数10000，用ASCII码形式存储在磁盘上占5个字节（每个字符占一个字节），而用二进制形式存储在磁盘上只占4个字节（00000000 00000000 00100111 00010000‬）。用ASCII形式存储时字符与字节一一对应，一个字节代表一个字符，便于逐个处理，但占的存储空间较多，而且处理的时候要花费转换时间（二进制与ASCII码之间的转换）。二进制形式存储就相当于直接把内存中的内容原封不动存储在磁盘上，由于不需要转换，所以二进制文件更加方便计算机处理。**

## 指向文件的指针

**C语言要想操作内存，需要用到各种数据类型的指针，而文件也是需要调用到内存中才能够使用，所以就需要用到“指向文件的指针”，就是“文件类型的指针”，简称“文件指针”。**

**每一个被使用的文件都在内存中开辟一个相应的文件信息区，用来存放文件的相关信息（如文件的名字、文件的状态和文件的位置等）。这些信息是保存在一个结构体变量中的，此结构体类型是由系统声明的，取名为FILE，其被包含在stdio.h头文件中。**

### 文件指针的定义

**由于文件类型已经在stdio.h头文件中有声明了，所以我们不需要另外声明，直接使用就行了。**

**文件类型变量的定义格式为：**

**FILE f1;**

**定义一个结构体变量f1，用它来存放一个文件的相关信息。这些信息是在打开文件时由系统根据文件的情况自动放入的，用户不必过问。**

**我们一般不会用文件变量来访问文件，而使用文件指针来访问文件。**

**文件指针的定义格式为：**

**FILE \*fp;**

**定义一个指针fp用于指向FILE类型的数据，可以使fp指向某一个文件在内存中的文件信息区（结构体变量），通过此文件信息区能够访问此文件。也就是说通过文件指针变量能够找到并可以操作其指向的文件。**

### **打开与关闭文件**

1. **用fopen函数打开文件**

**C语言规定用文件标准输入输出函数fopen来实现打开文件。**

**fopen函数的调用方式为：**

**fopen（文件名，使用文件的方式）；**

**例如：fopen（“text1.txt”,”r”）；**

**意思是以只读的方式打开名为“text1.txt”文件。**

**此时fopen函数返回的是“text1.txt”文件的起始地址，我们通常将fopen函数的返回值赋值给一个文件指针，用文件指针指向此文件的地址。**

**例如：FILE \*fp;//定义一个文件指针fp**

**fp=fopen（“text1.txt”,”r”）；//使fp指向文件“text1.txt”的首地址**

**这样fp就与文件“text1.txt”有联系了。在打开一个文件时，通知编译系统一下3个信息：**

**①需要打开的文件名称；②文件的打开方式；③使用哪个文件指针指向被打开的文件。**

**（2）文件的打开方式**

文件打开方式与含义如下表所示：

|  |  |  |
| --- | --- | --- |
| 文件的打开方式 | 含义 | 如果指定的文件不存在 |
| “r”（只读） | 为了输入数据，打开一个已存在的文本文件 | 出错 |
| “w”（只写） | 为了输出数据，打开一个文本文件 | 新建文件 |
| “a”（追加） | 向文本文件尾部添加数据 | 出错 |
| “rb”（只读） | 为了输入数据，打开一个已存在的二进制文件 | 出错 |
| “wb”（只写） | 为了输出数据，打开一个二进制文件 | 新建文件 |
| “ab”（追加） | 向二进制文件尾部添加数据 | 出错 |
| “r+”（读写） | 为了读和写，打开一个文本文件 | 出错 |
| “w+”（读写） | 为了读和写，打开一个文本文件 | 新建文件 |
| “a+”（读写） | 为了读和写，打开一个文本文件 | 出错 |
| “rb+”（读写） | 为了读和写，打开一个二进制文件 | 出错 |
| “wb+”（读写） | 为了读和写，打开一个二进制文件 | 新建文件 |
| “ab+”（读写） | 为了读和写，打开一个二进制文件 | 出错 |

**（3）用fclose函数关闭文件**

**在使用完一个文件之后，为了防止它被误用，应该关闭它。“关闭”就是撤销文件信息区和文件缓冲区，使指针不再指向此文件了，也无法操作此文件了，除非重新打开此文件，使指针指向此文件。**

**关闭文件用fclose函数。**

**fclose函数的一般形式为：**

**fclose（文件指针）;**

**例如：fclose（fp）;**

**在每次程序终止之前都要养成习惯关闭所有的文件。**

**当fclose函数成功关闭文件时，返回0；否则返回EOF（-1）。**

### **顺序读写文件**

**文件打开完成之后就可以对它进行读写操作了。**

**（1）常用的文件操作函数**

**①字符输入和输出函数：**

**使用字符读取函数fgetc从文件读取一个字符**

**ch=fgetc（fp）；**

**从文件指针fp指向的位置读取一个字符存入字符变量ch中，**

**读取成功返回所读的字符，失败则返回为你文件结束标志EOF（-1）。**

**使用字符写入函数fputc向文件写入一个字符**

**fputc（ch，fp）；**

**向文件指针fp指向的位置写入字符ch，**

**写入成功返回输出的字符，失败则返回EOF（-1）。**

**②字符串输入和输出函数**

**使用字符串读取函数fgets从文件读取一个字符串**

**fgets（str，n，fp）；**

**从文件指针fp指向的位置读取一个长度位n-1的字符串（最后一位赋值‘\0’，用作字符串结束标志），存放在字符数组str中。**

**读取成功返回地址str，失败则返回NULL。**

**使用字符串写入函数fputs向文件写入一个字符串**

**fputs（str，fp）；**

**把str所指向的字符串写入文件指针fp指向的位置，**

**写入成功返回0，否则返回非0值。**

**③格式化输入和输出函数**

**使用格式化输出函数fprintf向文件写入数据**

**fprintf（文件指针，“格式化字符串”，输出列表）；**

**例如：fprintf（fp，“%d，%c”，x，y）；**

**将变量x以整型的形式写入文件指针fp指向的位置，把变量y以字符的形式写入文件指针fp指向的位置。**

**使用格式化输入函数fscanf从文件读取数据**

**fscanf（文件指针，“格式化字符串”，输入列表）；**

**例如：fscanf（fp，“%d，%f”，&x，&y）；**

**从文件指针fp指向的位置读入一个整型数据和一个单精度型数据，分别存入变量x和变量y中。**

**④以二进制的形式读写数据**

**fread（buffer，size，count，fp）；**

**buffer是一个地址（数组），用于存储从文件读取出来的数据，size为需要读取的字节数，count为需要读取数据项的个数（每个数据项的大小为size）。**

**fwrite（buffer，size，count，fp）；**

### 4、随机读写文件

**⑤文件指针位置标记及定位**

**使用rewind函数强制使文件指针fp指向文件开头的位置。**

**rewind（fp）；**

**使用fseek函数使文件指针指向文件中任意位置。**

**fseek（fp，位移量，起始点）；**

**起始点用0、1、2代替，0代表文件开始位置，1代表当前位置，2代表文件末尾位置。**

**位移量是指以起始点为基础，向前移动的字节数。位移量应为long型数据（在末尾加上一个字母L就表示是long型）。**

**例如：**

**fseek（fp，100，0）；将文件指针fp向前移动到离文件开头100个字节处；**

**fseek（fp，50，1）；将文件指针fp向前移动到离当前位置50个字节处；**

**fseek（fp，-10，2）；将文件指针fp向后移动到离文件末尾10个字节处；**

**fseek一般用于二进制文件。**

**用rewind和fseek函数实现随机读写。**

### 5、文件读写的出错检测

**ferror(fp)；**

**if(ferror(fp))**

**{**

**printf(“文件读写失败！”);**

**}**