



知识点总结



- 文件：

C语言中，文件是**字节流**（**数据序列**），从**存储形式**上划分，主要有**文本文件**和**二进制文件**两种类型。

当数据**以字符形式存储**，便于人直接阅读和编辑，那就是文本文件，像**.txt文档**。

当数据**按字节原样存储**，不进行字符转换等处理，就是二进制文件，像.exe可执行文件、图片、音频都是典型的二进制文件。

- 文件路径：用于在计算机存储系统中定位文件位置，主要分为**绝对路径**和**相对路径**。（在Linux、macOS等**类Unix系统**中，路径分隔符使用斜杠'/'; 在**Windows系统**中，路径分隔符使用反斜杠'\'。）



- **绝对路径**：从文件系统的根目录开始，完整地描述文件的位置。

例：若在C盘的Projects文件夹下有一个example.txt文件，其绝对路径可表示为C:\Projects\example.txt。

- **相对路径**：基于当前目录来描述文件的位置。

在相对路径中，用.代表当前目录，用..代表上一层的父目录。

例：若当前目录为C:\Projects，对于上述example.txt文件，可以用相对路径.\example.txt来表示，而.\是可以省略的，所以也可直接用example.txt来表示。

- 注意：在Windows系统，因为反斜杠在C语言字符串中具有转义功能，因此需要用**双反斜杠**'\\'来表示真正的反斜杠，如"C:\\Projects\\example.txt"。此外，还有一种更为简洁的表示方式，即采用**正斜杠**'/'来分隔路径，如"C:/Projects/example.txt"。



- **文件指针 (FILE*)**：在文件操作函数中起着关键作用，使得程序能够对文件进行正确的读写操作。
- FILE是stdio库中定义的一个**结构体类型**，用于表示一个文件，它包含文件的**当前位置、文件状态**等信息。
- 文件指针 (FILE*) 是指向FILE结构体的**指针**，能够**维护文件的读写状态**，确保读写等操作的正确执行。

例：`FILE* fp`；定义了一个名为fp的文件指针。

- **fopen函数**和**fclose函数**：用于对文件进行打开和关闭操作，这两个函数均位于stdio库中。

fopen函数：用于打开一个文件，若文件成功打开，函数会返回一个指向该文件的文件指针。若文件打开失败，函数将返回NULL。

调用fopen函数的语法格式：`fopen("文件路径", "打开模式")`；

其中，"文件路径"，可以用**相对路径**表示，也可以用**绝对路径**表示。

| 模式 | 说明 |
|---------|--|
| r | 只读（文本文件）。文件必须存在，否则打开失败。 |
| w | 只写（文本文件）。若文件存在则清空内容；若不存在，则创建新文件。 |
| a | 追加（文本文件）。若文件存在，写入数据追加到文件末尾；若不存在，则创建新文件。 |
| r+ | 读写（文本文件）。文件必须存在，可读取和写入。 |
| w+ | 读写（文本文件）。若文件存在则清空内容；若不存在，则创建新文件。 |
| a+ | 读写（文本文件）。若文件存在，可读取和追加写入。若不存在，则创建新文件。 |
| rb | 只读（二进制文件）。文件必须存在，否则打开失败。 |
| wb | 只写（二进制文件）。若文件存在则清空内容；若不存在，则创建新文件。 |
| ab | 追加（二进制文件）。若文件存在，写入数据追加到文件末尾；若不存在，则创建新文件。 |
| rb+或r+b | 读写（二进制文件）。文件必须存在，可读取和写入。 |
| wb+或w+b | 读写（二进制文件）。若文件存在则清空内容；若不存在，则创建新文件。 |
| ab+或a+b | 读写（二进制文件）。若文件存在，可读取和追加写入。若不存在，则创建新文件。 |

| 模式 | 说明 |
|----|---|
| r | 只读（文本文件）。文件必须存在，否则打开失败。 |
| w | 只写（文本文件）。若文件存在则清空内容；若不存在，则创建新文件。 |
| a | 追加（文本文件）。若文件存在，写入数据追加到文件末尾；若不存在，则创建新文件。 |
| r+ | 读写（文本文件）。文件必须存在，可读取和写入。 |
| w+ | 读写（文本文件）。若文件存在则清空内容；若不存在，则创建新文件。 |
| a+ | 读写（文本文件）。若文件存在，可读取和追加写入。若不存在，则创建新文件。 |

| 模式 | 说明 |
|---------|--|
| rb | 只读（二进制文件）。文件必须存在，否则打开失败。 |
| wb | 只写（二进制文件）。若文件存在则清空内容；若不存在，则创建新文件。 |
| ab | 追加（二进制文件）。若文件存在，写入数据追加到文件末尾；若不存在，则创建新文件。 |
| rb+或r+b | 读写（二进制文件）。文件必须存在，可读取和写入。 |
| wb+或w+b | 读写（二进制文件）。若文件存在则清空内容；若不存在，则创建新文件。 |
| ab+或a+b | 读写（二进制文件）。若文件存在，可读取和追加写入。若不存在，则创建新文件。 |



- 例: `FILE* fp = fopen("example.txt", "r");` 就是以只读的方式, 打开当前目录下的example.txt文件, 并把fopen函数返回的指向该文件的指针赋给文件指针fp。

- **fclose函数**: 用于关闭一个已打开的文件, 释放与该文件相关的资源, 并将文件缓冲区中尚未写入的数据写入文件。

如果文件成功关闭, 函数会返回0。若文件关闭失败, 函数会返回EOF (通常为-1), 表示关闭过程中出现错误。

- 调用fclose函数的语法格式: **fclose(文件指针变量);**

这里的"文件指针变量"必须是一个有效的指针, 并且指向一个表示已打开文件流的FILE结构体。一般情况下, 这个指针是通过调用fopen函数获取的。

例: 假设已经定义了一个文件指针fp, 它指向一个已成功打开的文件example.txt。若要关闭该文件, 可以用fclose(fp);。

- 注意: 每一个成功打开的文件都应该在使用完毕后**及时关闭**, 以**释放系统资源并确保数据正确保存**。忘记关闭文件可能导致数据丢失或其他意想不到的问题。



文件是由数据序列的，可以有文本文件和二进制文件两种类型



题目1、下列关于C语言文件的叙述中，正确的是（ C ）

- A. 文件由一系列数据依次排列组成，只能构成二进制文件
- B. 文件由结构序列组成，可以构成二进制文件或文本文件
- C. 文件由数据序列组成，可以构成二进制文件或文本文件
- D. 文件由字符序列组成，其类型只能是文本文件



文件指针是用FILE类型定义的指针变量

变量，在使用的时候就需要在内存里给它分配空间

文件指针变量的值是它所指向的FILE结构体的地址，而文件的当前读取位置只是FILE结构体里的一个成员

一个文件指针在同一时间只能指向一个FILE 结构体



题目2、以下叙述正确的是（ A ）

- A. 文件指针是程序中用FILE定义的指针变量
- B. 在使用文件指针时，不需要在内存中为其分配空间
- C. 文件指针变量的值是文件的当前读取位置
- D. 文件指针可同时指向不同文件

 fopen函数的返回值必须赋给一个指向文件类型的指针变量，而不是任意类型的指针变量



题目3、以下叙述正确的是 (A)

- A. 在C语言中调用fopen函数就可把程序中要读、写的文件与磁盘上实际的数据文件联系起来
- B. fopen函数的调用形式为：fopen("文件路径");
- C. fopen函数的返回值为NULL时，则成功打开指定的文件
- D. fopen函数的返回值必须赋给一个任意类型的指针变量



“w”模式是只写模式，如果这个文件存在，会先清空这个文件的原有内容，再从这个文件的起始位置开始写，如果文件不存在，就会创建一个新文件



✓ 题目4、设文件指针fp已定义，执行语句fp = fopen("file","w");后，以下针对文本文件file操作叙述的选项中正确的是（ B ）

- A. 写操作结束后可以从头开始读
- B. 只能写不能读
- C. 可以在原有内容后追加写
- D. 可以随意读和写



只要对文件进行了操作，都要调用fclose函数关闭文件，这样能有效避免数据丢失和防止资源浪费



✓ 题目5、以下关于fclose(fp)函数的叙述正确的是 (A)

- A. 当程序中对文件的所有写操作完成之后，必须调用fclose(fp)函数关闭文件
- B. 当程序中对文件的所有写操作完成之后，不一定要调用fclose(fp)函数关闭文件
- C. 只有对文件进行输入操作之后，才需要调用fclose(fp)函数关闭文件
- D. 只有对文件进行输出操作之后，才能调用fclose(fp)函数关闭文件