目录

[C 语言 stdio.h 文件操作函数 2](#_Toc207224375)

[**一、文件打开与关闭** 2](#_Toc207224376)

[**1. FILE \*fopen(const char \*filename, const char \*mode);** 2](#_Toc207224377)

[**2. int fclose(FILE \*stream);** 2](#_Toc207224378)

[**二、文本读写函数** 3](#_Toc207224379)

[**1. int fputc(int c, FILE \*stream);** 3](#_Toc207224380)

[**2. int fgetc(FILE \*stream);** 3](#_Toc207224381)

[**3. char \*fgets(char \*str, int n, FILE \*stream);** 3](#_Toc207224382)

[**4. int fputs(const char \*str, FILE \*stream);** 4](#_Toc207224383)

[**三、二进制读写函数** 4](#_Toc207224384)

[**1. size\_t fread(void \*ptr, size\_t size, size\_t nmemb, FILE \*stream);** 4](#_Toc207224385)

[**2. size\_t fwrite(const void \*ptr, size\_t size, size\_t nmemb, FILE \*stream);** 4](#_Toc207224386)

[**四、文件位置控制** 5](#_Toc207224387)

[**1. int fseek(FILE \*stream, long offset, int whence);** 5](#_Toc207224388)

[**2. long ftell(FILE \*stream);** 5](#_Toc207224389)

[**3. void rewind(FILE \*stream);** 5](#_Toc207224390)

[**五、状态检查和缓冲区操作** 6](#_Toc207224391)

[**1. int feof(FILE \*stream);** 6](#_Toc207224392)

[**2. int ferror(FILE \*stream);** 6](#_Toc207224393)

[**3. void clearerr(FILE \*stream);** 6](#_Toc207224394)

[**4. int fflush(FILE \*stream);** 6](#_Toc207224395)

[基于 <fstream>的C++ 文件操作详解 7](#_Toc207224396)

[**一、头文件和命名空间** 7](#_Toc207224397)

[**二、文件打开与关闭** 7](#_Toc207224398)

[**1. 打开文件** 7](#_Toc207224399)

[**2. 关闭文件** 7](#_Toc207224400)

[**三、打开模式（ios::）** 7](#_Toc207224401)

[**示例：组合模式** 8](#_Toc207224402)

[**四、常用操作函数** 8](#_Toc207224403)

[**1. 检查文件状态** 8](#_Toc207224404)

[**五、读写操作** 8](#_Toc207224405)

[**1. 文本写入** 8](#_Toc207224406)

[**2. 文本读取** 8](#_Toc207224407)

[**3. 字符读写** 8](#_Toc207224408)

[**4. 整行读取** 9](#_Toc207224409)

[**六、二进制文件读写（结构体、数组）** 9](#_Toc207224410)

[**示例：写入结构体到文件** 9](#_Toc207224411)

[**示例：读取结构体** 9](#_Toc207224412)

[**七、定位操作（文件指针）** 9](#_Toc207224413)

[**用于文本或二进制文件随机访问** 9](#_Toc207224414)

# C 语言 stdio.h 文件操作函数

**一、文件打开与关闭**

**1. FILE \*fopen(const char \*filename, const char \*mode);**

**功能：**

打开一个文件并返回 FILE\* 文件指针。

**参数说明：**

* filename：要打开的文件名。
* mode：文件的打开模式，常见有：

|  |  |
| --- | --- |
| **模式** | **含义** |
| "r" | 以只读方式打开文件（文件必须存在） |
| "w" | 以只写方式打开文件（文件存在则清空，不存在则创建） |
| "a" | 以追加方式写入文件 |
| "r+" | 以读写方式打开文件（文件必须存在） |
| "w+" | 以读写方式打开文件（存在则清空） |
| "a+" | 以读写方式打开，追加写入 |

* + 可加 b：如 "rb"，表示以二进制方式打开。

**返回值：**

* 成功：返回 FILE\* 指针。
* 失败：返回 NULL。

**示例：**

FILE \*fp = fopen("example.txt", "r");

if (fp == NULL) {

perror("File open error");

}

**2. int fclose(FILE \*stream);**

**功能：**

关闭文件，释放文件资源。

**返回值：**

* 成功：0
* 失败：EOF（通常是 -1）

**示例：**

fclose(fp);

**二、文本读写函数**

**1. int fputc(int c, FILE \*stream);**

**功能：**

将字符 c 写入文件。

**返回值：**

* 成功：返回写入的字符。
* 失败：返回 EOF

**示例：**

fputc('A', fp);

**2. int fgetc(FILE \*stream);**

**功能：**

从文件读取一个字符。

**返回值：**

* 成功：读取的字符。
* 失败或结束：EOF

**示例：**

int ch = fgetc(fp);

**3. char \*fgets(char \*str, int n, FILE \*stream);**

**功能：**

从文件中读取一行字符串，最多读取 n-1 个字符，最后添加 \0。

**返回值：**

* 成功：返回 str
* 失败或 EOF：返回 NULL

**示例：**

char buffer[100];

fgets(buffer, 100, fp);

**4. int fputs(const char \*str, FILE \*stream);**

**功能：**

将字符串写入文件，不写入结尾的 \0。

**返回值：**

* 成功：非负数
* 失败：EOF

**示例：**

fputs("Hello, world!", fp);

**三、二进制读写函数**

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

**功能：**

从文件中读取 nmemb 个元素，每个元素大小为 size，存入 ptr 指向的内存。

**返回值：**

实际读取的元素数量（不是字节数）

**示例：**

int nums[10];

size\_t read\_count = fread(nums, sizeof(int), 10, fp);

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

**功能：**

将内存数据写入文件，共写入 nmemb \* size 字节。

**返回值：**

成功写入的元素数量

**示例：**

int nums[10] = {1,2,3};

fwrite(nums, sizeof(int), 10, fp);

**四、文件位置控制**

**1. int fseek(FILE \*stream, long offset, int whence);**

**功能：**

移动文件指针。

**参数 whence：**

|  |  |
| --- | --- |
| **值** | **含义** |
| SEEK\_SET | 文件开头 |
| SEEK\_CUR | 当前指针位置 |
| SEEK\_END | 文件末尾 |

**返回值：**

* 成功：0
* 失败：非0

**示例：**

fseek(fp, 0, SEEK\_END); // 移动到文件末尾

**2. long ftell(FILE \*stream);**

**功能：**

返回文件指针当前位置相对于文件起始的偏移字节数。

**返回值：**

偏移值，失败返回 -1L

**示例：**

long pos = ftell(fp);

**3. void rewind(FILE \*stream);**

**功能：**

将文件指针重置到文件开头，相当于 fseek(fp, 0, SEEK\_SET);

**五、状态检查和缓冲区操作**

**1. int feof(FILE \*stream);**

**功能：**

判断是否到达文件末尾。

**返回值：**

是 EOF 返回非 0，否则返回 0

**示例：**

while (!feof(fp)) { ... }

**2. int ferror(FILE \*stream);**

**功能：**

检查文件是否发生错误。

**返回值：**

* 有错误：非0
* 无错误：0

**3. void clearerr(FILE \*stream);**

**功能：**

清除文件的错误和 EOF 标志。

**4. int fflush(FILE \*stream);**

**功能：**

刷新缓冲区，把缓冲区内容写入文件，常用于写文件后立即保存。

# 基于 <fstream>的C++ 文件操作详解

C++ 使用三个主要类处理文件输入输出，均定义在 <fstream> 中：

|  |  |  |
| --- | --- | --- |
| 类名 | 功能 | 继承自 |
| ifstream | 读文件（输入流） | istream |
| ofstream | 写文件（输出流） | ostream |
| fstream | 读写文件（输入输出流） | iostream |

**一、头文件和命名空间**

#include <fstream>

#include <iostream>

#include <string>

using namespace std;

**二、文件打开与关闭**

**1. 打开文件**

三种方式均可打开文件：

ifstream inFile("file.txt"); // 自动打开，默认 ios::in

ofstream outFile("file.txt"); // 默认 ios::out

fstream file("file.txt", ios::in | ios::out); // 自定义模式

也可以先定义对象，再调用 open()：

fstream file;

file.open("file.txt", ios::in | ios::out);

**2. 关闭文件**

file.close();

**三、打开模式（ios::）**

文件打开时可以指定模式（常用组合使用 | 运算符）：

|  |  |
| --- | --- |
| 模式 | 含义 |
| ios::in | 输入（读）模式 |
| ios::out | 输出（写）模式 |
| ios::app | 追加到文件末尾 |
| ios::ate | 打开时指针移动到文件末尾 |
| ios::trunc | 截断文件（默认写入模式） |
| ios::binary | 二进制模式 |

**示例：组合模式**

fstream file("data.dat", ios::in | ios::out | ios::binary);

**四、常用操作函数**

**1. 检查文件状态**

|  |  |
| --- | --- |
| 函数 | 功能 |
| is\_open() | 检查文件是否成功打开 |
| eof() | 检查是否到达文件末尾 |
| fail() | 检查读取或格式错误 |
| good() | 文件流是否处于正常状态 |
| clear() | 清除错误状态位 |

if (!inFile.is\_open()) {

cerr << "无法打开文件" << endl;

}

**五、读写操作**

**1. 文本写入**

ofstream out("test.txt");

out << "Hello, world!" << endl;

out << 42 << " " << 3.14 << endl;

out.close();

**2. 文本读取**

ifstream in("test.txt");

string line;

while (getline(in, line)) {

cout << line << endl;

}

in.close();

**3. 字符读写**

char ch;

while (in.get(ch)) {

cout << ch;

}

**4. 整行读取**

string str;

getline(in, str);

**六、二进制文件读写（结构体、数组）**

**示例：写入结构体到文件**

#include <fstream>

struct Student {

char name[20];

int age;

};

int main() {

Student s = {"Tom", 18};

ofstream out("student.dat", ios::binary);

out.write((char\*)&s, sizeof(s));

out.close();

}

**示例：读取结构体**

Student s;

ifstream in("student.dat", ios::binary);

in.read((char\*)&s, sizeof(s));

cout << s.name << " " << s.age << endl;

**七、定位操作（文件指针）**

**用于文本或二进制文件随机访问**

|  |  |
| --- | --- |
| 函数 | 功能 |
| seekg(pos) / seekg(offset, dir) | 设置读取位置 |
| seekp(pos) / seekp(offset, dir) | 设置写入位置 |
| tellg() | 获取读取位置 |
| tellp() | 获取写入位置 |

**示例：**

file.seekg(0, ios::end); // 移动到文件末尾

int size = file.tellg(); // 获取文件大小

file.seekg(0, ios::beg); // 回到文件开头