# 一、文件及基本读写

## （一）文件的基本概念（File Basics）

文件（File） 是存储在电脑硬盘上的一块“信息盒子”，用来**保存数据**，数据可以是文字、图片、声音等。文件可以是文本文件（Text）或二进制文件（Binary）。

- 读写文件是程序与电脑硬盘“对话”，实现数据的保存和读取。
- 就像你写日记，保存在一本书里；电脑的文件就是电脑里的日记本。你可以写日记（写文件）、也可以翻看以前的日记（读文件）。
- C++ 通过标准库中的**文件流（File Stream）**进行文件操作。

常用的头文件是：

```
#include <fstream>
```

## （二）文本文件类型与二进制文件类型（Text vs Binary Files）

**☑ 文本文件（Text File）：**

- 是由**普通字符组成**的文件，可以用记事本打开看见文字。比如 .txt 文件、.in 文件、.out 文件、源代码文件 . 都是文本文件。

**C++ 打开、写入、读取文本文件示例：**

```
#include <iostream>
#include <fstream>   // 文件操作头文件
using namespace std;
int main() {
    // 写文件
    ofstream fout("test.txt"); // 创建输出文件流，打开文件
    fout << "你好，小明！" << endl;
    fout << "这是我的第一份文件。" << endl;
    fout.close();   // 关闭文件，保存
    // 读文件
```

```
    ifstream fin("test.txt"); // 创建输入文件流，打开文件
    string line;
    while (getline(fin, line)) {   // 按行读取文件
        cout << line << endl;
    }
    fin.close();   // 关闭文件
    return 0;
}
```

- ofstream（输出文件流）就像“写笔”，用它把内容写进日记本。
- ifstream（输入文件流）就像“读眼镜”，用它看日记本里的内容。
- close() 就是“合上书”，保存内容。

**☑ 二进制文件（Binary File）**

- 文件内容是以**二进制格式（0 和 1）**保存的，机器读取速度快，但人类不可直接读懂。
- 常用于存储图片、音频或结构体、数组等大数据。

```
// 示例：写入二进制文件
ofstream fout("data.bin", ios::binary);
int x = 1234;
fout.write((char*)&x, sizeof(x));
fout.close();
```

## （三）文件重定向（File Redirection）

文件重定向是让程序的输入或输出，不是从键盘或者屏幕，而是从文件读写。

- **输入重定向**：程序的输入来自一个文件，而不是键盘。
- **输出重定向**：程序的输出写到一个文件，而不是屏幕。

**什么是文件重定向（File Redirection）？**

- 默认情况下，程序的输入来自键盘（标准输入 stdin），输出到屏幕（标准输出 stdout）。
- 使用 **freopen()** 可以将输入输出重定向到文件。
- 这在写程序做测试、处理大量数据时非常方便。

**示例解释：**

```
freopen("apple.in", "r", stdin);
```

意思是：以后读 cin 的内容，其实是从文件 apple.in 中读。

```
freopen("apple.out", "w", stdout);
```

意思是：以后所有的 cout 输出都写入 apple.out 文件中。

就像你原本和老师当面交流（屏幕和键盘），但现在你通过信件来沟通（文件），把你的输入输出“写进信封”或者“从信封读”。

## （四）文件读写操作（File Read and Write）

### 1. 普通方式（使用 ifstream 和 ofstream）：

文件读写是程序主动打开文件，往里写内容或从里面读内容。

- **写文件**：ofstream（输出文件流）
- **读文件**：ifstream（输入文件流）

需要包含头文件：

```
#include <fstream>
```

**写文件示例**

```
#include <fstream>
using namespace std;
int main() {
    ofstream fout("output.txt"); // 打开文件准备写
    fout << "你好，小明！" << endl;
```

```
    fout << "这是文件写入的内容。" << endl;
    fout.close();  // 关闭文件
    return 0;
}
```

**读文件示例**

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main() {
    ifstream fin("output.txt"); // 打开文件准备读
    string line;
    while (getline(fin, line)) { // 一行行读
        cout << line << endl;
    }
    fin.close();
    return 0;
}
```

**读写文件示例**

```
#include <fstream>
using namespace std;
int main() {
    ifstream fin("apple.in");
    ofstream fout("apple.out");
```

```
        int x;
        fin >> x;
        fout << x * 2 << endl;
        fin.close();
        fout.close();
    }
```

**2. 使用 freopen() 重定向的方式：**

```
    freopen("apple.in", "r", stdin);
```

- 将标准输入 stdin 重定向为文件 apple.in。
- 从现在开始，所有的 cin 输入都会从这个文件中读取数据。
- r 表示以“读”的方式打开文件（Read）。

```
    freopen("apple.out", "w", stdout);
```

- 将标准输出 stdout 重定向为文件 apple.out。
- 所有的 cout 输出内容会写入这个文件，而不是显示在屏幕上。
- w 表示以“写”的方式打开文件（Write）。如果文件已存在，原内容会被清空。

**程序示例**

```
    #include <bits/stdc++.h>
    using namespace std;
    int main() {
        freopen("apple.in", "r", stdin);
        freopen("apple.out", "w", stdout);
        int x;
        cin >> x;
```

```
    cout << x * 2 << endl;
    return 0;
}
```