

C++ 基础学习

[TOC]

整理自Olwiki, 始于2024.7.10晚

文件操作

文件之概念

文件是**根据特定的目的而收集在一起的有关数据的集合**。C/C++ 把每一个文件都看成是一个**有序的字节流**，每个文件都是以 **文件结束标志 (EOF)** 结束，如果要操作某个文件，程序应该首先打开该文件，每当一个文件被打开后（请记得关闭打开的文件），该文件就**和一个流关联起来**，这里的**流实际上是一个字节序列**。

C/C++ 将文件分为**文本文件**和**二进制文件**。文本文件就是**简单的文本文件（重点）**，另外二进制文件就是**特殊格式的文件或者可执行代码文件等**。

文件的操作步骤

1. 打开文件，将**文件指针**指向文件，决定打开**文件类型**；
2. 对文件进行**读、写**操作（比赛中主要用到的操作，其他一些操作暂时不写）；
3. 在使用完文件后，**关闭文件**。

freopen 函数

函数简介

函数用于**将指定输入输出流以指定方式重定向到文件**，包含于头文件 **stdio.h (cstdio)** 中，该函数可以在**不改变代码原貌的情况下改变输入输出环境**，但使用时应当保证流是可靠的。其主要含有三种操作，即读、写和附加

命令之格式

```
FILE* freopen(const char* filename, const char* mode, FILE* stream);
```

FILE*者，C语言的stdio.h里面预定义的一个结构体，是管理文件流的一种结构。建立流之前要先定义出 **FILE**类型的指针变量[?]

参数说明

- **filename**: 要打开的文件名
- **mode**: 文件打开的模式，表示文件访问的权限
- **stream**: 文件指针，通常使用标准文件流 (**stdin/stdout**) 或标准错误输出流 (**stderr**)
- **返回值**: 文件指针，指向被打开文件

文件打开格式

标识	解释
r	以只读方式打开文件，文件必须存在，只允许读入数据（常用）
r+	以读/写方式打开文件，文件必须存在，允许读/写数据
rb	以只读方式打开二进制文件，文件必须存在，只允许读入数据
rb+	以读/写方式打开二进制文件，文件必须存在，允许读/写数据
rt+	以读/写方式打开文本文件，允许读/写数据
w	以只写方式打开文件，文件不存在会新建文件，否则清空内容，只允许写入数据（常用）
w+	以读/写方式打开文件，文件不存在将新建文件，否则清空内容，允许读/写数据
wb	以只写方式打开二进制文件，文件不存在将会新建文件，否则清空内容，只允许写入数据
wb+	以读/写方式打开二进制文件，文件不存在将新建文件，否则清空内容，允许读/写数据
a	以只写方式打开文件，文件不存在将新建文件，写入数据将被附加在文件末尾（保留 EOF 符）
a+	以读/写方式打开文件，文件不存在将新建文件，写入数据将被附加在文件末尾（不保留 EOF 符）
at+	以读/写方式打开文本文件，写入数据将被附加在文件末尾
ab+	以读/写方式打开二进制文件，写入数据将被附加在文件末尾

使用方法

读入文件内容

```
freopen("data.in", "r", stdin);  
// data.in 就是读取的文件名，要和可执行文件放在同一目录下
```

输出到文件

```
freopen("data.out", "w", stdout);  
// data.out 就是输出文件的文件名，和可执行文件在同一目录下
```

关闭标准输入输出流

```
fclose(stdin);  
fclose(stdout);
```

注意

`printf/scanf/cin/cout` 等函数默认使用 `stdin/stdout`，将 `stdin/stdout` 重定向后，这些函数将输入/输出到被定向的文件

使用模板

注意上文提到的注意项

```
#include <cstdio>
#include <iostream>

int main(void) {
    freopen("data.in", "r", stdin);
    freopen("data.out", "w", stdout);
    /*
    中间的代码不需要改变，直接使用 cin 和 cout 即可
    */
    fclose(stdin);
    fclose(stdout);
    return 0;
}
```

fopen 函数

大致与 `freopen` 相同，函数将**打开指定文件并返回打开文件的指针**

函数原型

```
FILE* fopen(const char* path, const char* mode)
```

参数含义与`freopen`中的相同

可用读写函数

- `fread/fwrite`
- `fgetc/fputc`
- `fscanf/fprintf`
- `fgets/fputs`

fopen之使用方法

```
FILE *in, *out; // 定义文件指针
in = fopen("data.in", "r");
out = fopen("data.out", "w");
/*
do what you want to do
*/
```

```
fclose(in);  
fclose(out);
```

C++之 ifstream/ofstream 文件输入输出流

ifstream/ofstream之使用方法

读入文件内容：

```
ifstream fin("data.in");  
// data.in 就是读取文件的相对位置或绝对位置
```

输出到文件：

```
ofstream fout("data.out");  
// data.out 就是输出文件的相对位置或绝对位置
```

关闭标准输入输出流：

```
fin.close();  
fout.close();
```

模板

```
#include <fstream>  
using namespace std; // 两个类型都在 std 命名空间里  
  
ifstream fin("data.in");  
ofstream fout("data.out");  
  
int main(void) {  
    /*  
    中间的代码改变 cin 为 fin , cout 为 fout 即可  
    */  
    fin.close();  
    fout.close();  
    return 0;  
}
```

来自CSDN之补充

[来源](#)

初解释

```
FILE *fp
```

FILE 是C语言的`stdio.h`里面预定义的一个结构体，是管理文件流的一种结构。本质是对设备的文件IO操作的一种封装，根据操作类型的不同有`stdin`,`stdout`,`stderr`

操作

标准库中提供了通用的函数来读取和写入流，如`fopen`,`fclose`等等，如果要进行具体的操作，那么接下来就是如此书写：

```
FILE *fp;    //建立流之前要先定义出 FILE类型的指针变量
```

接下来对文件的操作有以下几种： 1.打开文件：

```
fp = fopen("a.txt","r")    //a.txt是文件名,r表示只读, read
```

2.关闭文件：

```
fclose(fp);
```

3.读文件：

```
fread(&x,sizeof(int),1,fp);    //sizeof是c语言内置的关键字, sizeof(int)是计算int类型的大小
```

从流中读一个整数，存放在x中，如果成功，返回值为1

4.写文件：

```
fwrite(&x,sizeof(int),1,fp);
```

把整型变量x写入流中，如果成功，返回1

FILE *fp 是声明，声明fp是指针，用来指向FILE类型的对象

`fp=fopen("yssysj.txt","r");` `fopen`标准函数，打开磁盘文件`yssysj.txt`，用于读，送回指针，指针再指向FILE类型对象。

示例

```
#include <stdio.h>

int main()
{
    FILE *fp;
    char str_buf[11];

    fp = fopen("test_file.txt", "r"); //这里要确保test_file.txt的存在

    if (fp)
        fgets(str_buf, 10, fp);
    else
        printf("Cannot find file test_file.txt\n")

    return 0;
}
```

以上基本整理于2024.7.10晚
