

C++ 文件操作

C++ 文件操作

例题讲解：

C++ 文件流：

fstream // 文件流

ifstream // 输入文件流

ofstream // 输出文件流

//创建一个文本文件并写入信息

//同向屏幕上输出信息一样将信息输出至文件

```
#include<iomanip.h>
```

```
#include<fstream.h>
```

```
void main()
```

```
{
```

```
ofstream f1("d:\\me.txt"); //打开文件用于写，若文件不存在就创建它
```

```
if(!f1)return; //打开文件失败则结束运行
```

```
f1<<setw(20)<<" 姓名： "<<" 张三 "<<endl; //使用插入运算符写文件内容
```

```
f1<<setw(20)<<" 家庭地址： "<<" 河南郑州 "<<endl;
```

```
f1.close(); //关闭文件
```

```
}
```

运行后打开文件 d:\me.txt，其内容如下：

姓名：张三

家庭地址：河南郑州

文件操作：

打开文件

文件名

注意路径名中的斜杠要双写，如：

```
"D:\\MyFiles\\ReadMe.txt"
```

文件打开方式选项：

ios::in = 0x01, //供读，文件不存在则创建 (ifstream 默认的打开方式)

ios::out = 0x02, //供写，文件不存在则创建，若文件已存在则清空原内容

(ofstream 默认的打开方式)

ios::ate = 0x04, //文件打开时，指针在文件最后。可改变指针的位置，常和 in、out 联合使用

ios::app = 0x08, //供写，文件不存在则创建，若文件已存在则在原文件内容后写入新的内容，指针位置总在最后

ios::trunc = 0x10, //在读写前先将文件长度截断为 0（默认）

ios::nocreate = 0x20, //文件不存在时产生错误，常和 in 或 app 联合使用

ios::noreplace = 0x40, //文件存在时产生错误，常和 out 联合使用

ios::binary = 0x80 //二进制格式文件

文件保护方式选择项：

filebuf::openprot; //默认的兼容共享方式

filebuf::sh_none; //独占，不共享

filebuf::sh_read; //读共享

filebuf::sh_write; //写共享

打开文件的方法

调用构造函数时指定文件名和打开模式

ifstream f("d:\\12.txt",ios::nocreate); //默认以 ios::in 的方式打开文件，文件不

存在时操作失败

ofstream f("d:\\12.txt"); //默认以 ios::out 的方式打开文件

fstream f("d:\\12.dat",ios::in|ios::out|ios::binary); //以读写方式打开二进制文

件

使用 Open 成员函数

fstream f;

f.open("d:\\12.txt",ios::out); //利用同一对象对多个文件进行操作时要用到

open 函数

检查是否成功打开

成功：

if(f){...} //对 ifstream 、 ofstream 对象可用，fstream 对象不可用。

if(f.good()){...}

失败：

if(!f){...} //! 运算符已经重载

if(f.fail()){...}

读写操作

使用 << , >> 运算符

只能进行文本文件的读写操作，用于二进制文件可能会产生错误。

使用函数成员 get 、 put 、 read 、 write 等

经常和 read 配合使用的函数是 gcount() ，用来获得实际读取的字节数。

读写二进制文件注意事项

打开方式中必须指定 ios::binary ，否则读写会出错

用 read\\write 进行读写操作，而不能使用插入、提取运算符进行操作，否则会出
错。

使用 eof() 函数检测文件是否读结束，使用 gcount() 获得实际读取的字节数

关闭文件

使用成员函数 close ，如：

```
f.close();
```

利用析构函数

对象生命期结束时检查文件是否关闭，对没有关闭的文件进行关闭操作。

随机读写文件

通过移动文件读写指针，可在文件指定位置进行读写。

```
seekg( 绝对位置 ); //绝对移动, //输入流操作
```

```
seekg( 相对位置 ,参照位置 ); //相对操作
```

```
tellg(); //返回当前指针位置
```

```
seekp( 绝对位置 ); //绝对移动, //输出流操作
```

```
seekp( 相对位置 ,参照位置 ); //相对操作
```

```
tellp(); //返回当前指针位置
```

参照位置：

```
ios::beg = 0 //相对于文件头
```

```
ios::cur = 1 //相对于当前位置
```

```
ios::end = 2 //相对于文件尾
```

读写文本文件的示例：

//为能够正确读出写入文件的各数据，各数据间最好要有分隔

```
#include<fstream.h>
```

```
void main()
```

```
{
```

```
fstream f("d:\\try.txt",ios::out);
```

```
f<<1234<<' '<<3.14<<'A'<<"How are you"; // 写入数据
```

```
f.close();
```

```
f.open("d:\\try.txt",ios::in);
```

```
int i;
```

```
double d;
```

```
char c;
```

```
char s[20];
```

```
f>>i>>d>>c; //读取数据
```

```
f.getline(s,20);
```

```
cout<<i<<endl; //显示各数据
```

```
cout<<d<<endl;
```

```
cout<<c<<endl;
```

```
cout<<s<<endl;
```

```
f.close();
```

```
}
```

运行结果：

1234

3.14

A

How are you

Press any key to continue

显示文本文件的内容

//使用 get() 一次读一个字符 ----- 方案一

```
#include<fstream.h>
```

```
void main()
```

```
{
```

```
    简介.txt",ios::nocreate);
```

```
if(!fin){
```

```
cout<<"File open error!\n";
```

```
return;
```

```
}
```

```
char c;
```

```
while((c=fin.get())!=EOF)cout<<c; // 注意结束条件的判断
```

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

```
}
```

//使用 get(char *,int n,char delim='\n') 一次读多个字符 ---- 方案二

//巧妙利用文本文件中不会有字符 '\0' 的特点进行读取

```
#include<fstream.h>
```

```
void main()
```

```
{
```

```
    简介.txt",ios::nocreate);
```

```
if(!fin){
```

```
cout<<"File open error!\n";
```

```
return;
```

```
}
```

```
char c[80];
```

```
while(fin.get(c,80,'\0')!=NULL)cout<<c; // 注意结束条件的判断
```

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

```
}
```

//使用 read(char *,int n) 读文件 ----- 方案三

```
#include<fstream.h>
```

```
void main()
```

```
{
```

```
    简介.txt",ios::nocreate);
```

```
if(!fin){
```

```
cout<<"File open error!\n";
return;
}
char c[80];
while(!fin.eof())    // 判断文件是否读结束
{
    fin.read(c,80);
    cout.write(c,fin.gcount());
}
fin.close();
}
拷贝文件
//二进制文件操作示例
#include<fstream.h>
void main()
{
    ifstream fin("C:\\1.exe",ios::nocreate|ios::binary);
    if(!fin){
        cout<<"File open error!\n";
        return;
    }
    ofstream fout("C:\\2.exe",ios::binary);
    char c[1024];
    while(!fin.eof())
    {
        fin.read(c,1024);
        fout.write(c,fin.gcount());
    }
    fin.close();
    fout.close();
    cout<<"Copy over!\n";
```