

C++文件读写

文件的使用步骤如下：

1. 说明一个文件流对象，这又被称为内部文件：

```
1 ifstream ifile; //只输入用
2 ofstream ofile; //只输出用
3 fstream iofile; //既输入又输出用
```

2. 使用文件流对象的成员函数打开一个磁盘文件。这样文件流对象和磁盘文件名之间建立联系。文件流中说明了三个打开文件的成员函数。

```
1 void ifstream::open(const char*, int=ios::in, int=filebuf::openprot);
2 void ofstream::open(const char*, int=ios::out, int=filebuf::openport);
3 void fstream::open(const char*, int, int=filebuf::openport);
```

第一个参数为要打开的磁盘文件名。第二个参数为打开方式，有输入（in），输出（out）等，打开方式在ios基类中定义为枚举类型。第三个参数为指定打开文件的保护方式，一般取缺省。

所以第二部可如下进行：

```
1 iofile.open("TEST.txt", ios::in | ios::out);
```

打开文件也应该判断是否成功，若成功，文件流对象值为非零值，不成功则为0（NULL），文件流对象值物理上就是指它的地址。

因此打开一个文件完整的程序为：

```
1 fstream iofile("TEST.txt", ios::in | ios::out);
2 if(!iofile)
3 {
4     cout<<"打开文件失败！"<<"TEST.txt"<<endl;
5     exit(1); //失败退回操作系统
6 }
```

3. 使用提取和插入运算符对文件进行读写操作，或者使用成员函数进行读写。

4. 关闭文件。三个文件流类各有一个关闭文件的成员函数。

```
1 void ifstream::close();
2 void ofstream::close();
3 void fstream::close();
```

文件打开过程示例代码如下

```
1 #include<iostream>
2 #include<fstream>
3 using namespace std;
4
5 void main()
6 {
7     int ar[] = {65,66,67,68,69,1134688,71,70,73,74}, br[10];
8     ofstream ofile; //①
```

```

9      ofile.open("TEST.txt", ios::out);//②
10     if(!ofile)
11     {
12         cerr<<"Open File Fail."<<endl;
13         exit(1);
14     }
15     for(int t=0; t<sizeof(ar)/sizeof(int); ++t)//③
16         ofile<<ar[t]<<" ";//向文件内写入
17     ofile.close();//④
18
19     ifstream ifile;
20     ifile.open("TEST.txt", ios::in);
21     if(!ifile)
22     {
23         cerr<<"Open File Fail."<<endl;
24         exit(1);
25     }
26     for(int t=0; t<10; ++t)
27         ifile>>br[t];//从文件内读出并写入br中
28     ifile.close();
29 }

```

查看是否生成文件TEST.txt和br[10]内是否有数据

Debug	2019/2/24 13:40	文件夹	
2019_02_21.vcxproj	2019/2/21 12:17	VC++ Project	4 KB
2019_02_21.vcxproj.filters	2019/2/21 12:17	VC++ Project Fil...	1 KB
2019_02_21.vcxproj.user	2019/2/21 12:02	Visual Studio Pr...	1 KB
TEST.txt	2019/2/24 13:42	文本文档	1 KB

```

for(int t=0; t<10; ++t)
    ifile>>br[t];//从文件内读出并写入br中
ifile.close();
}

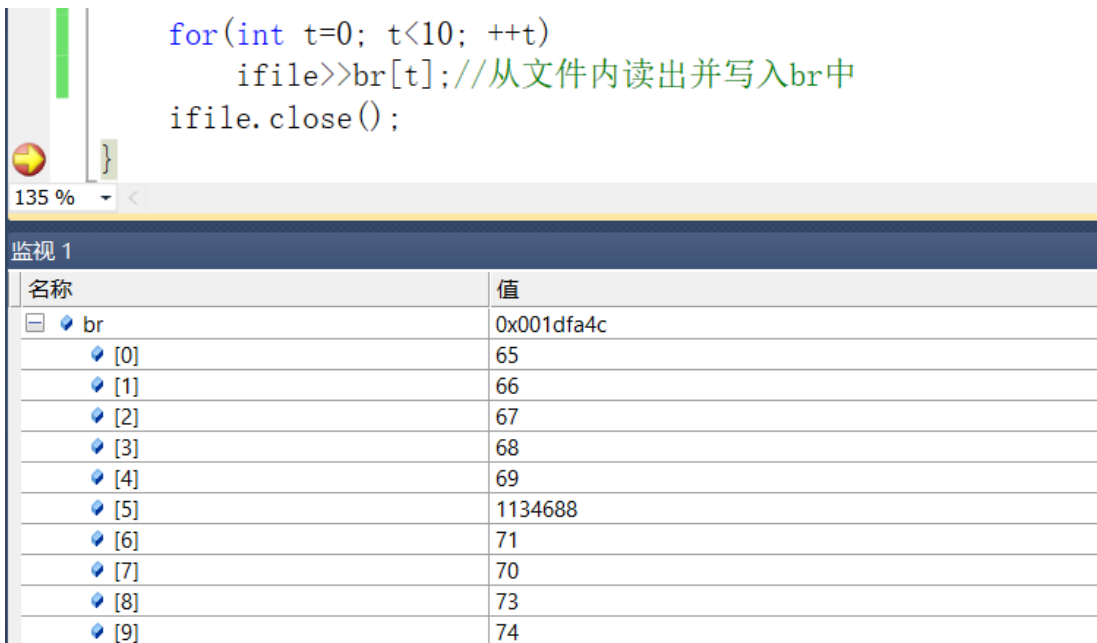
```



135 %

监视 1

名称	值
br	0x001dfa4c
[0]	65
[1]	66
[2]	67
[3]	68
[4]	69
[5]	1134688
[6]	71
[7]	70
[8]	73
[9]	74



文件打开方式是在 `ios` 类中定义的，公有枚举成员：

```
enum open_mode{
    in=0x01, //打开文件用于输入操作(从文件读取)，文件指针在文件头
    out=0x02, /*打开文件用于写入文件。如文件不存在，则建立，但指定目
               录必须存在，否则建立文件失败。如文件存在，未同时设
               app,ate,in则文件清空*/
    ate=0x04,
    //打开文件用于输入/输出,文件指针在文件尾,但新数据可写到任何位置
    app=0x08, //打开文件用于输出,但从尾部添加,新数据只能添加在尾部
    truncate=0x10, //打开文件,并清空它,以建立新文件
    nocreate=0x20, //如文件存在则打开,不存在并不创建新文件
    noreplace=0x40,
    //如文件不存在则创建,如文件存在则只能设为ate及app方式
    binary=0x80 }; //以二进制方式打开文件
```

使用方式与格式控制符相同。

在C++中可以又程序移动文件指针，从而实现文件的随机访问，即可读写流中任意一段内容。一般文件很难准确定位，所以随机访问多用于二进制文件。

在 `ios` 类中说明了一个公有枚举类型：

```
1  enum seek_dir
2  {
3      beg=0, //文件开头
4      cur=1, //文件指针的当前位置
5      end=2 //文件结尾
6  }
```

```
1  #include<iostream>
2  #include<fstream>
3  using namespace std;
4
5  void main()
6  {
7      int ar[] = {65,66,67,68,69,70,71,70,73,74}, br[10], pos,value;
8      ofstream ofile;
```

```
9      ofile.open("TEST.txt", ios::out);
10      if(!ofile)
11      {
12          cerr<<"Open File Fail."<<endl;
13          exit(1);
14      }
15      for(int i=0; i<sizeof(ar)/sizeof(int); ++i)
16          ofile<<ar[i]<<" ";
17      ofile.close();
18
19      ifstream ifile;
20      ifile.open("TEST.txt", ios::in);
21      if(!ifile)
22      {
23          cerr<<"Open File Fail."<<endl;
24          exit(1);
25      }
26      while(1)
27      {
28          cout<<"input pos = ";
29          cin>>pos;
30          ifile.seekg(3*pos, ios::beg);
31          ifile>>value;
32          cout<<value<<endl;
33      }
34      ifile.close();
35  }
```

运行结果

E:\C程序\2019_02_21\Debug\2019_02_21.exe

```
input pos = 0
65
input pos = 1
66
input pos = 2
67
input pos = 3
68
input pos =
```

不积小流无以成江河