



第11章 流类库和输入输出



本章重点

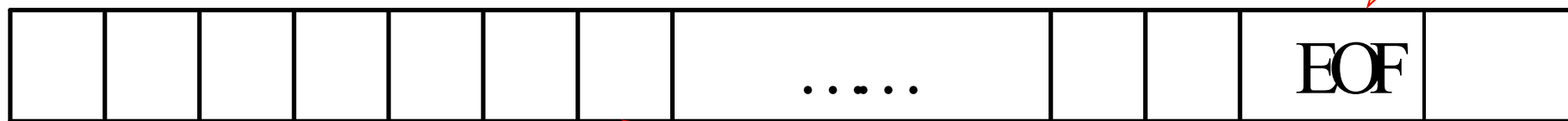
- 文件和流的概念
- 使用FILE结构操作文件
- 使用特殊的类和对象操作文件
- 文件对于实际系统的开发非常有用，而且不可替代

基本概念 (1/3)

■ 文件是什么？

- 文件是外存数据的组织单位。一个文件对应一批存放在外存（如硬盘、软盘、U盘）的数据，计算机以及计算机的使用者通过文件名实现对文件的操作（如复制、删除、内容修改）。
- C++把一个文件看成是一个字节序列。

0 1 2 3 4 5 6 n-2 n-1



结束符

字节中的内容？



基本概念 (2/3)

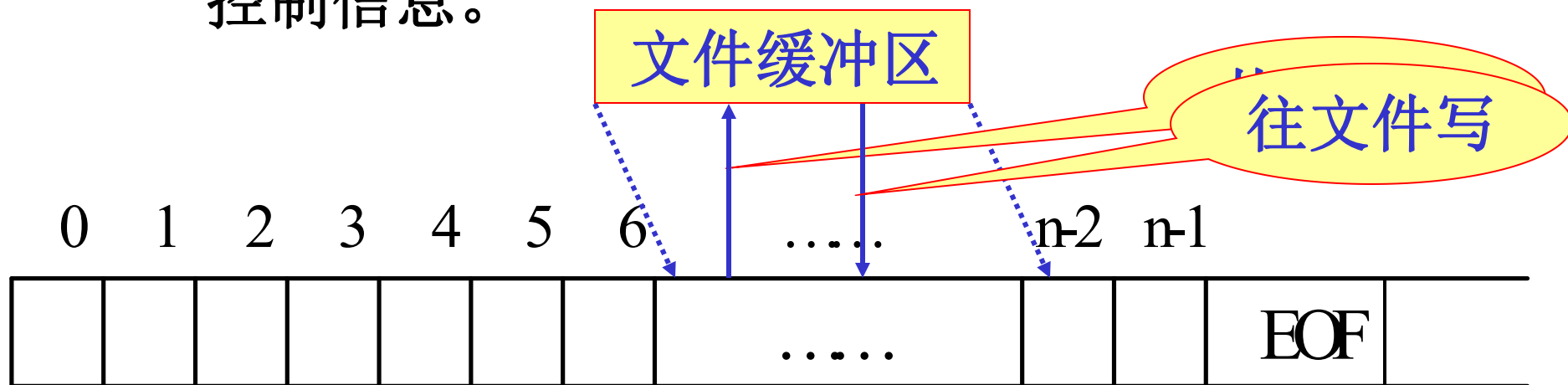
■ C++的文件类型

- 文本文件（也称**ASCII**文件）。文件中每个字节的内容是**ASCII**码（0~127），便于阅读和编辑。文件名通常以**.txt**为后缀
- 二进制文件。文件中每个字节的内容不限（八个二进制位的任意0/1组合），无法阅读和编辑。文件名通常以**.dat**为后缀

基本概念 (3/3)

■ 流是什么？

- 流是对文件中字节流向及顺序的描述。一个流对应一个正在工作的文件（指文件被打开后，被关闭前），实质上是计算机操作系统为文件建立的一个保存文件部分内容的缓冲区及有关控制信息。





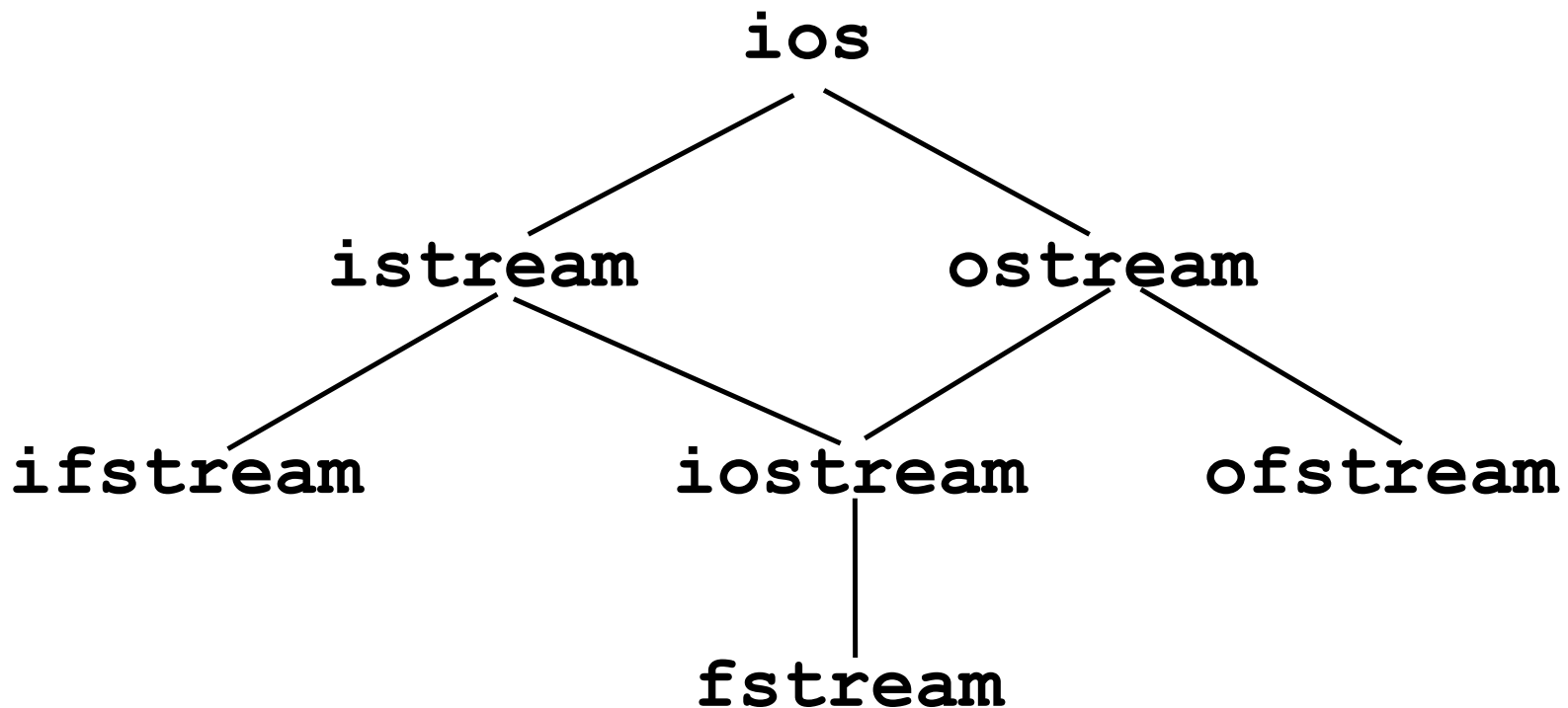
用FILE结构操作文件(1/9)

■ 三部曲

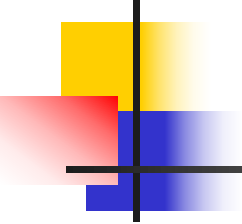
- 打开一个已存在的文件或建立并打开一个新文件
 - 关键函数: `fopen`
- 对该文件进行读/写操作
 - 关键函数: `fgetc`、`fputc`、`fgets`、`fputs`、`fscanf`、`fprintf`、`feof`、`rewind`、`fseek`、`fread`、`fwrite`
- 关闭该文件
 - 关键函数: `fclose`
- ❖ 以上函数都在库文件`stdio.h`中定义

使用类和对象操作文件(1/6)

- 文件流类的继承关系



使用类和对象操作文件(2/6)

- 
- 几个特殊的流对象
 - 标准输入流对象`cin`
(类`istream`的对象)
 - 标准输出流对象`cout`
 - 非缓冲标准错误流对象`cerr`
 - 缓冲标准错误流对象`clog`
(类`ostream`的对象)

使用类和对象操作文件(3/6)

■ 三个文件流类

- 类**`ifstream`**—文件的输入
- 类**`ofstream`**—文件的输出
- 类**`fstream`**—文件的输入输出
- 上述三个类都在头文件**`fstream.h`**中定义

△ 例子：使用文件流对象新建或打开一个文件，并写入字符串"This is a test file. "

```
#include <fstream.h>
main()
{ //声明文件流对象
ofstream outFile("outFile.txt", ios::out);
if (!outFile)
    //使用错误流对象输出错误信息
    cerr<<"Open or create file error."
        <<endl;
else
    //输出数据到与对象outFile 关联的文件中
    outFile << "This is a test file.";
return 0; //文件关闭?
}
```

使用类和对象操作文件(4/6)

打开方式	描述 (打开方式可以用 作无冲突组合)
ios::in	打开一个供读取的文件
ios::out	打开一个供写入的文件（默认方式）
ios::app	打开一个供写入的文件，写入的所有数据将被追加到文件的末尾
ios::ate	打开一个已有的文件，文件指针指向文件末尾
ios::trunc	废弃当前文件内容或建立新文件, 供读写
ios::nocreate	如果要打开的文件不存在，则打开失败。 （不建立新文件）
ios::noreplace	如果要打开的文件已存在，则打开失败。
ios::binary	打开一个二进制文件 (默认是文本文件)

使用类和对象操作文件(5/6)

■ 例子6中的

`ofstream`

```
outFile("outFile.txt", ios::out);
```

可以改写为:

```
ofstream outFile;
```

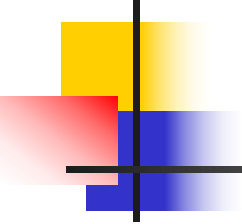
```
outFile.open("outFile.txt", ios::out);
```

或

```
ofstream outFile;
```

```
outFile.open("outFile.txt");
```

使用类和对象操作文件(6/6)

- 
- 文件可以显式关闭：
`outFile.close();`
 - 问题：声明对象时使用文件流类 `ifstream`、`ofstream`、`fstream` 中的哪一个？
 - 取决于文件的打开方式

例子：使用文件流对象将整数、浮点数、字符串等类型的数据写入文件

```
#include <fstream.h>
main()
{ofstream
    outFile ("outFile.txt", ios::out);
    if (!outFile )
        cerr<<"Open file or create file error."
            <<endl;
    else
        {outFile <<5<<"string"<<" "<<1.2;
          outFile.close();
        }
    return 0;
}
```

例子：使用文件流对象读取上例中程序创建的文件数据，跳过字符串 “string”

```
#include <fstream.h>
main()
{ ifstream
    inFile ("outFile.txt", ios::in);
    int a; float c;
    if (!inFile)
        cerr<<"File open error."<<endl;
    else
    { inFile>>a; inFile.seekg(7,ios::cur);
      inFile>>c;
      cout<<"INT--"<<a<<endl<<"FLO--"
          <<c<<endl;
      inFile.close();
    }
    return 0; }
```

例子：使用文件保存和恢复类的对象(1/3)

```
#include <fstream.h>
class Test
{public:
    Test(int m, double n)
        {a = m; b = n;}
    void set(int m, double n)
        {a = m; b = n;}
    void show()
        {cout<<"int--"<<a<<" double--"
          << b <<endl;
        }
private:
    int a;
    double b;
};
```


例子：使用文件保存和恢复类的对象(2/3)

```
main()  
{Test st(10,10.5);  
  cout<<"Object st:"<<endl;  
  st.show();  
  ofstream outFile("outFile.dat",  
                   ios::out|ios::binary);  
  if (!outFile )  
    cerr<<"Open file or create file error."  
    <<endl;  
  else  
    {outFile.write((char *)&st,sizeof(st));  
    outFile.close();}  
  st.set(1, 2.3);  
  cout<<"After set, object st:"<<endl;  
  st.show(); //输出int--1 double--2.3
```

例子：使用文件保存和恢复类的对象(3/3)

```
//创建文件流对象，再次打开该文件用于读
ifstream infile("outFile.dat",
               ios::in|ios::binary);
if (! infile )
    cerr<<"File open error."<<endl;
else
{ //从文件中恢复对象st
    infile.read((char *)&st,sizeof(st));
    infile.close();
}
cout<<" After read from file:"<<endl;
st.show(); //输出int--10 double--10.5
return 0;
}
```



学习目的检测

- 掌握文件和流的基本概念
- 掌握使用FILE结构操作文本文件和二进制文件的方法
- 了解文件流类的继承关系
- 掌握使用文件流对象操作文本文件和二进制文件的方法