

7.二进制文件的读写

要根据 **文件的定义格式** 对二进制文件进行读写。

比如**BMP**位图文件，是典型的二进制文件。其文件头部是格式固定的信息，其中

前2字节用来记录文件为BMP格式，
接下来的8个字节用来记录文件长度，
再接下来的4字节用来记录BMP文件头的长度，等等。

因此，**BMP**文件的读取方法是依次读取2字节、8字节、4字节的数据，再转化为字符或整数。

二进制文件读写函数

- ▶ 对二进制文件进行操作时，打开文件要指定方式 `ios::binary`
- ▶ 从二进制文件输入数据可调用istream流类提供的成员函数，函数原型为：

`istream& read(char* buffer, int len)`

- ▶ 向二进制文件输出数据可调用ostream流类提供的成员函数，函数原型为：

`ostream& write(const char* buffer, int len)`

两个函数格式上差不多，第一个参数是一个字符指针，用于指向输入输出数据所放的内存空间的地址。第二个参数是一个整数，表示要输入输出的数据的字节数。

【例9】 将学生信息存入二进制文件再读取出来。

建立学生信息类，包含姓名、班级、性别、年龄四个私有属性。有构造函数、输出自身信息的函数。在主函数中创建3个对象，而后按二进制形式存入文件，最后再读出该文件信息并显示。

```
#include<iostream>
#include<fstream>
using namespace std;
class Student      // 定义类
{
    char Name[10];
    char Class[10];
    char Sex;
    int Age;
public:
    Student() { }
    Student( char *Name, char *Class, char sex, int age) {
        strcpy(this->Name,Name);
        strcpy(this->Class,Class);
        Sex=sex;
        Age=age;
    }
    void Showme() {
        cout<<Name<<'\t'<<Class<<'\t'<<Sex<<'\t'<<Age<<endl;
    }
};
```

```

int main()
{
    Student    stu[3]={
        Student("王二小","电气11",'m',27),
        Student("刘大明","机械01",'f',24),
        Student("李文化","生物12",'m',39)    };
    //打开文件
    ofstream file1("file.dat",ios::binary);
    if(!file1) {    cout<<"文件打开失败!";    return 1;    }
    //写文件
    for(int i=0;i<3;i++)
        file1.write((char*)&stu[i],sizeof(stu[i]));
    file1.close(); //关闭文件

    //////////以下为读文件并显示出来//////////
    Student    stu2; //建立对象
    //打开文件
    ifstream file2("file.dat", ios::binary);
    if(!file2) {
        cout<<"文件打开失败!"; return 1;
    }
}

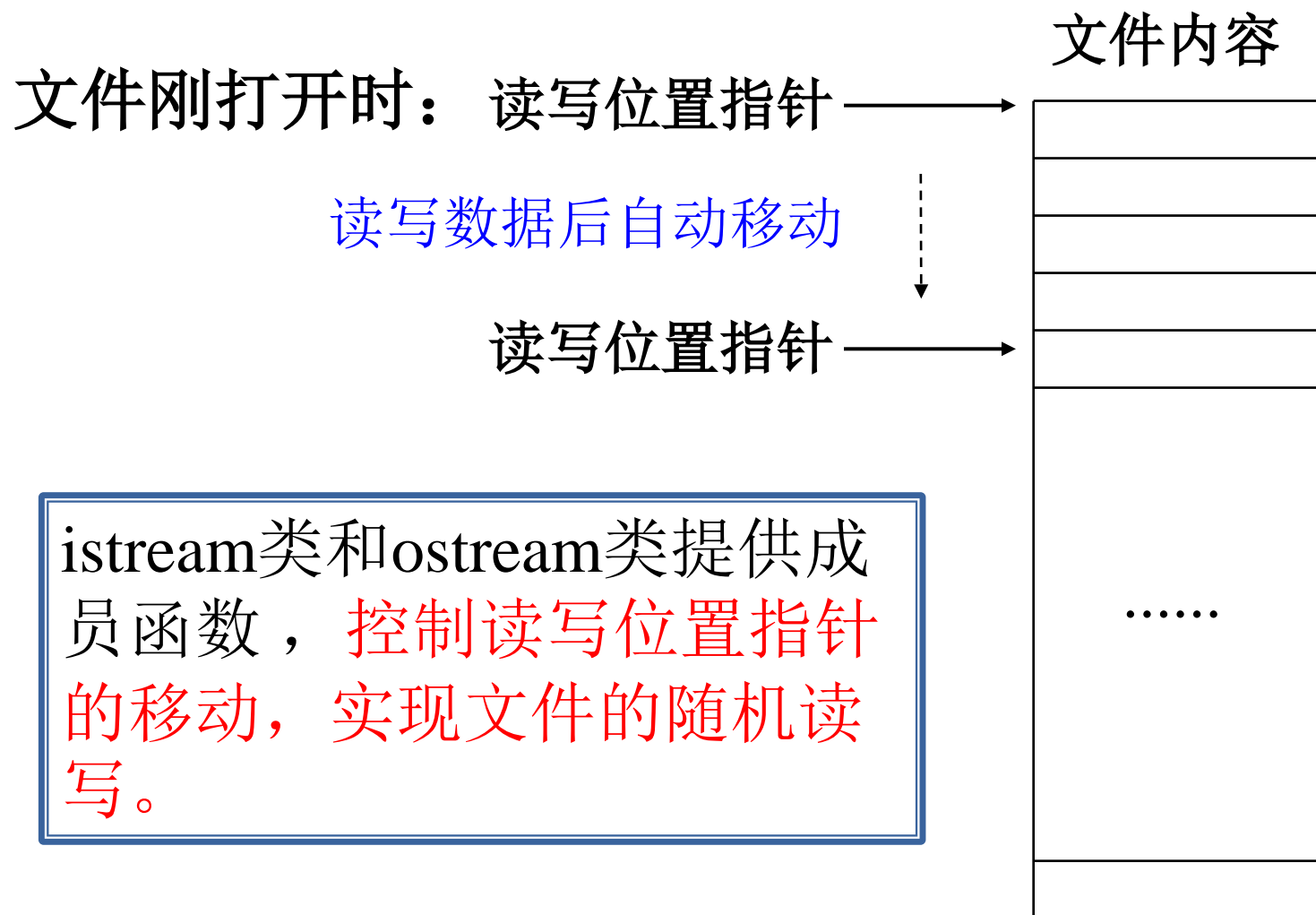
```

```
//读文件
while(file2)
{
    file2.read((char*)&stu2,sizeof(stu2));
    if(file2) stu2.Showme();
}
//关闭文件
file2.close();
return 0;
}
```

程序运行后，先创建文件并写入信息，而后从文件读出信息并显示如下：

王二小	电气	27
刘大明	机械	24
李文化	生物	39

二进制文件的顺序读写、随机读写



istream 类操作流读指针的成员函数

➤ `istream & istream::seekg (long pos);`

读指针从流的起始位置向后移动由`pos`指定字节

➤ `istream & istream::seekg (long off, ios::seek_dir);`

读指针从流的 `seek_dir` 位置移动, `off` 指定字节

`ios::seek_dir` 值:

`cur` 当前读指针所在的当前位置

`beg` 文件流的开始位置 (文件头部)

`end` 文件流的结尾处

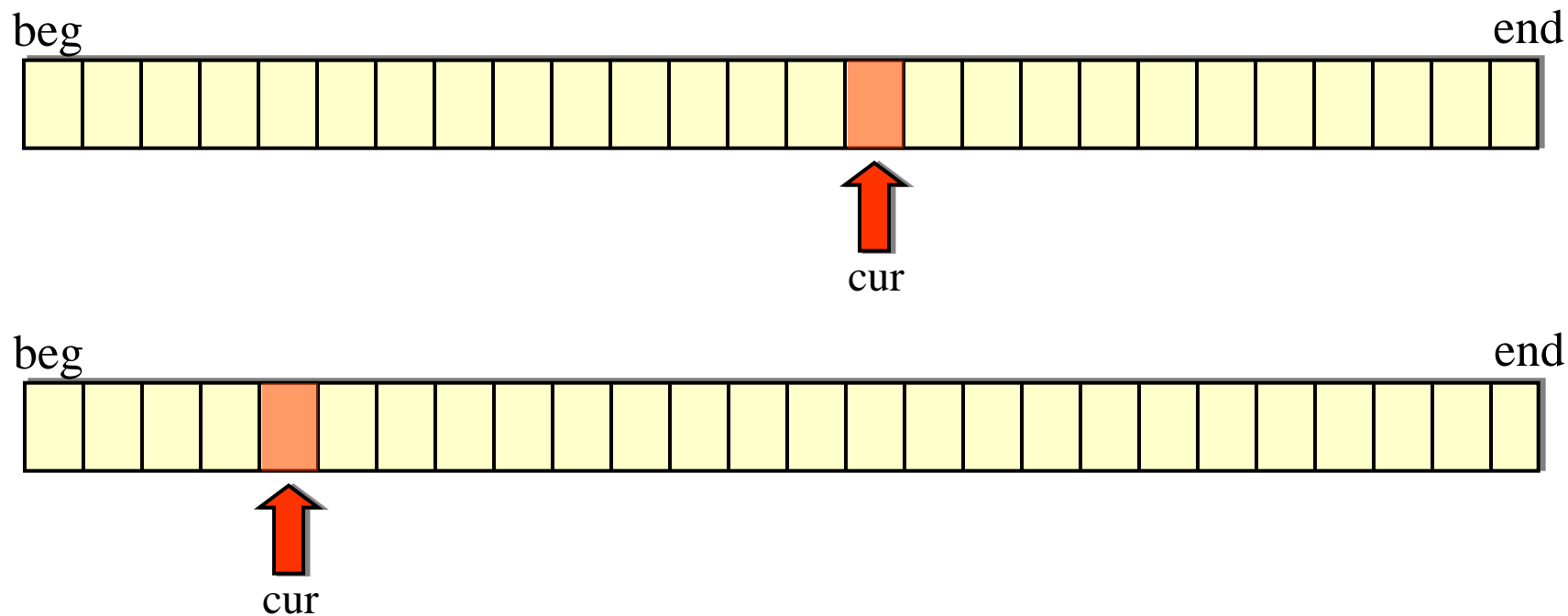
例如：

```
istream input ;
```

.....

```
input . seekg ( - 10 , ios :: cur ) ;
```

// 读指针以当前位置为基准，向前移动 10 个字节



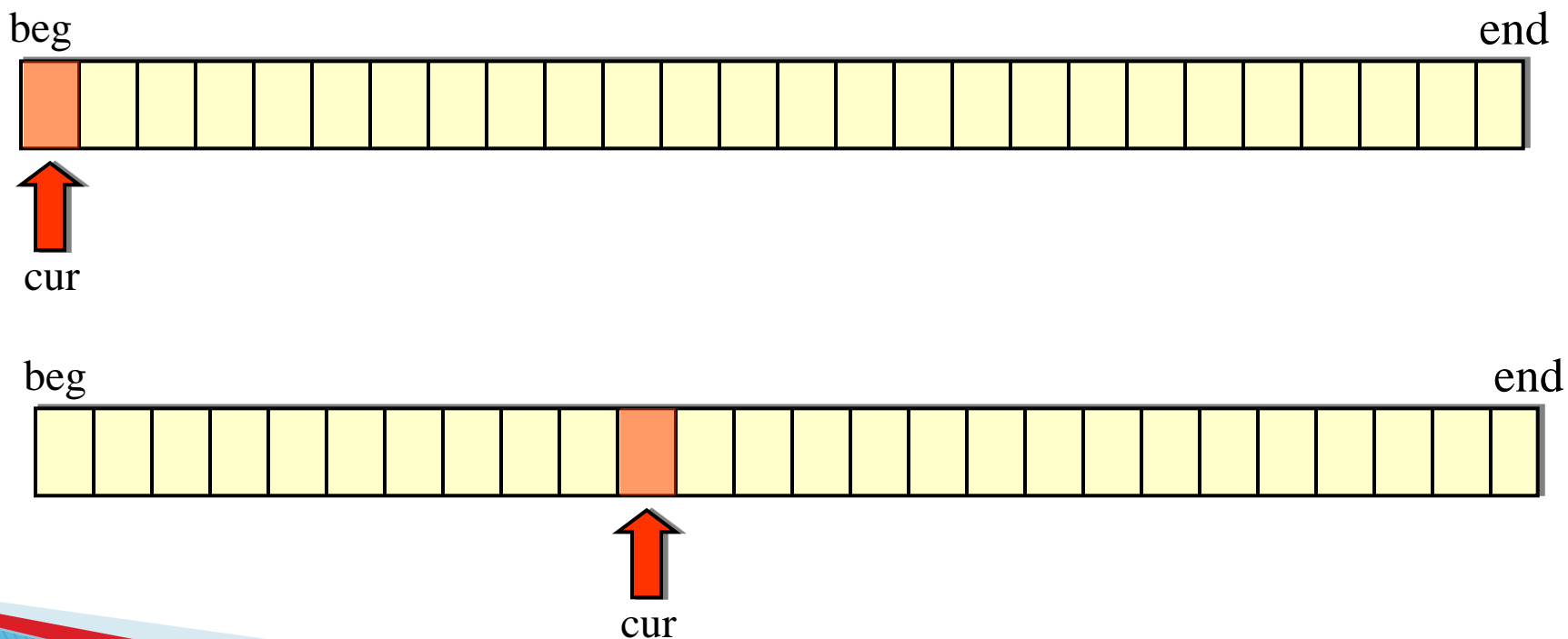
例如:

```
istream input ;
```

.....

```
input . seekg ( 10 , ios :: beg ) ;
```

// 读指针从流的开始位置，向后移动 10 个字节



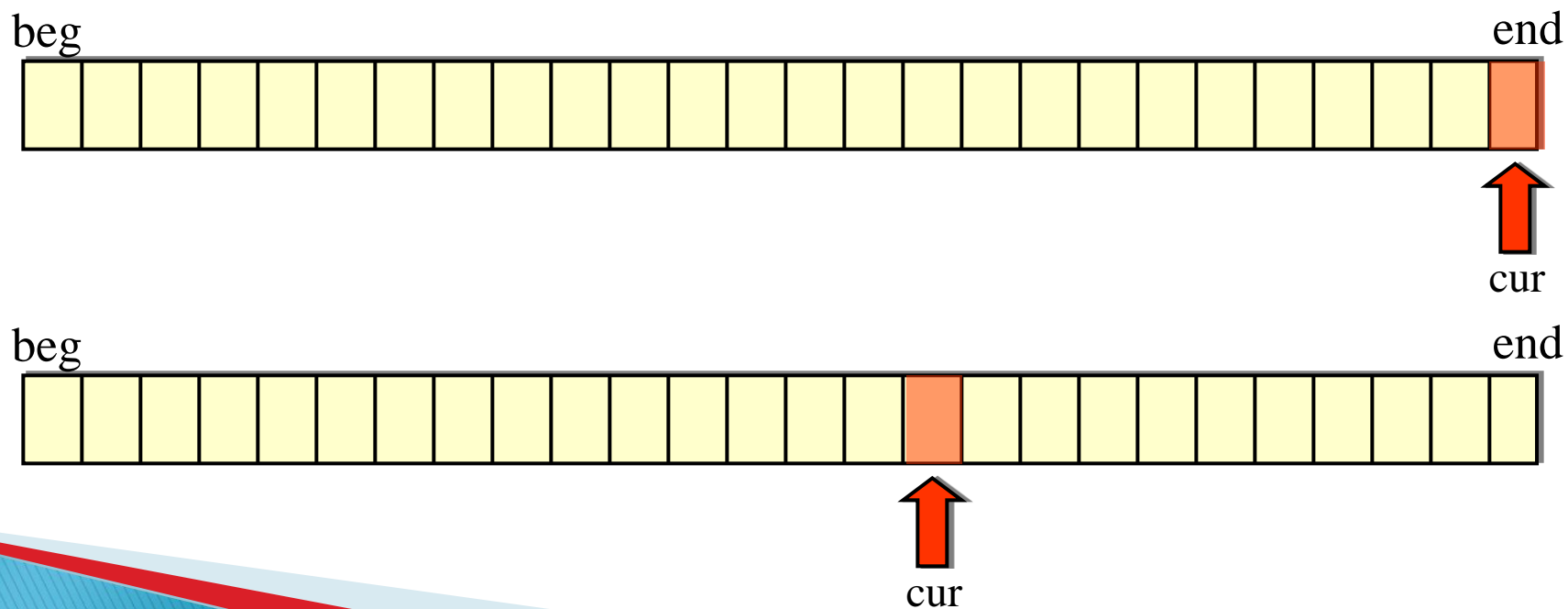
例如:

```
istream input ;
```

.....

```
input . seekg ( -10 , ios :: end ) ;
```

// 读指针从流的结尾，向前移动 10 个字节



ostream 类操作流写指针的成员函数

➤ `ostream & ostream::seekp (long pos);`

写指针从流的起始位置向后移动由参数指定字节

➤ `ostream & ostream::seekp (long off, ios::seek_dir);`

写指针从流的 `seek_dir` 位置移动由 `off` 指定字节

➤ `ostream & ostream::tellp ();`

返回写指针当前所指位置值

【例10】从二进制文件中读取倒序读取信息。

本例读取上一个例所生成的学生信息文件，并且从最后一个记录倒序读取并输出。

```
#include<iostream>
#include<fstream>
using namespace std;
class Student
{ 略  };

int main()
{   Student stu;    //建立对象
    //打开文件
    ifstream file("file.dat",ios::binary);
    if(!file)  { cout<<"文件打开失败!";
                return 1;
            }
}
```

```
file.seekg(0,ios::end);    //定位文件指针到文件末尾
int len=file.tellg();      //得到文件指针位置（文件大小）
//读文件
for(int k=len/sizeof(stu)-1;k>=0;k--)
{
    file.seekg(k*sizeof(stu));
    file.read((char*)&stu,sizeof(stu));
    stu.Showme();
}
file.close(); //关闭文件
return 0;
}
```

本例题运行结果显示如下：

李文化	生物	39
刘大明	机械	24
王二小	电气	27