

郭 婕 刻家琪

# 文件操作



# 数据的层次

- 位 bit
- 字节 byte
- 域/记录:

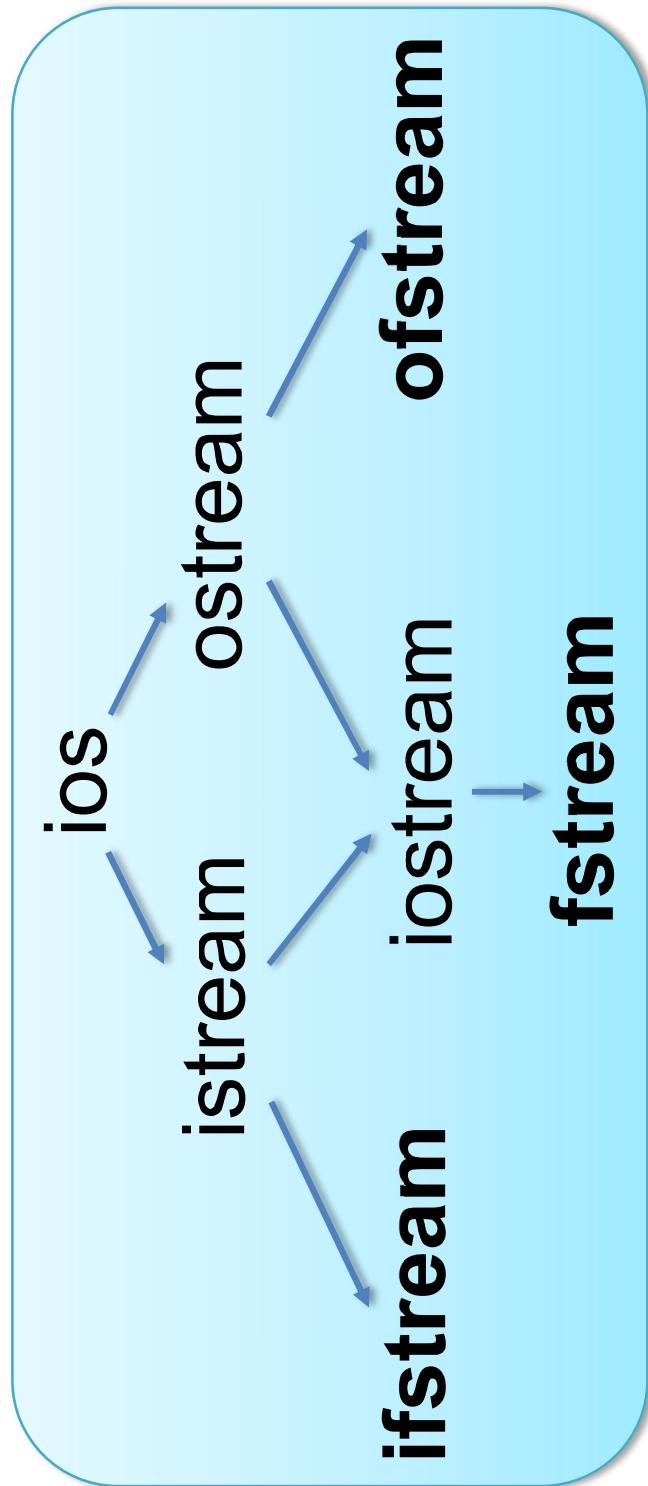
例如：学生记录

```
int ID;  
char name[10];  
int age;  
int rank[10];
```

- 将所有记录顺序地写入一个文件 → 顺序文件

# 文件和流

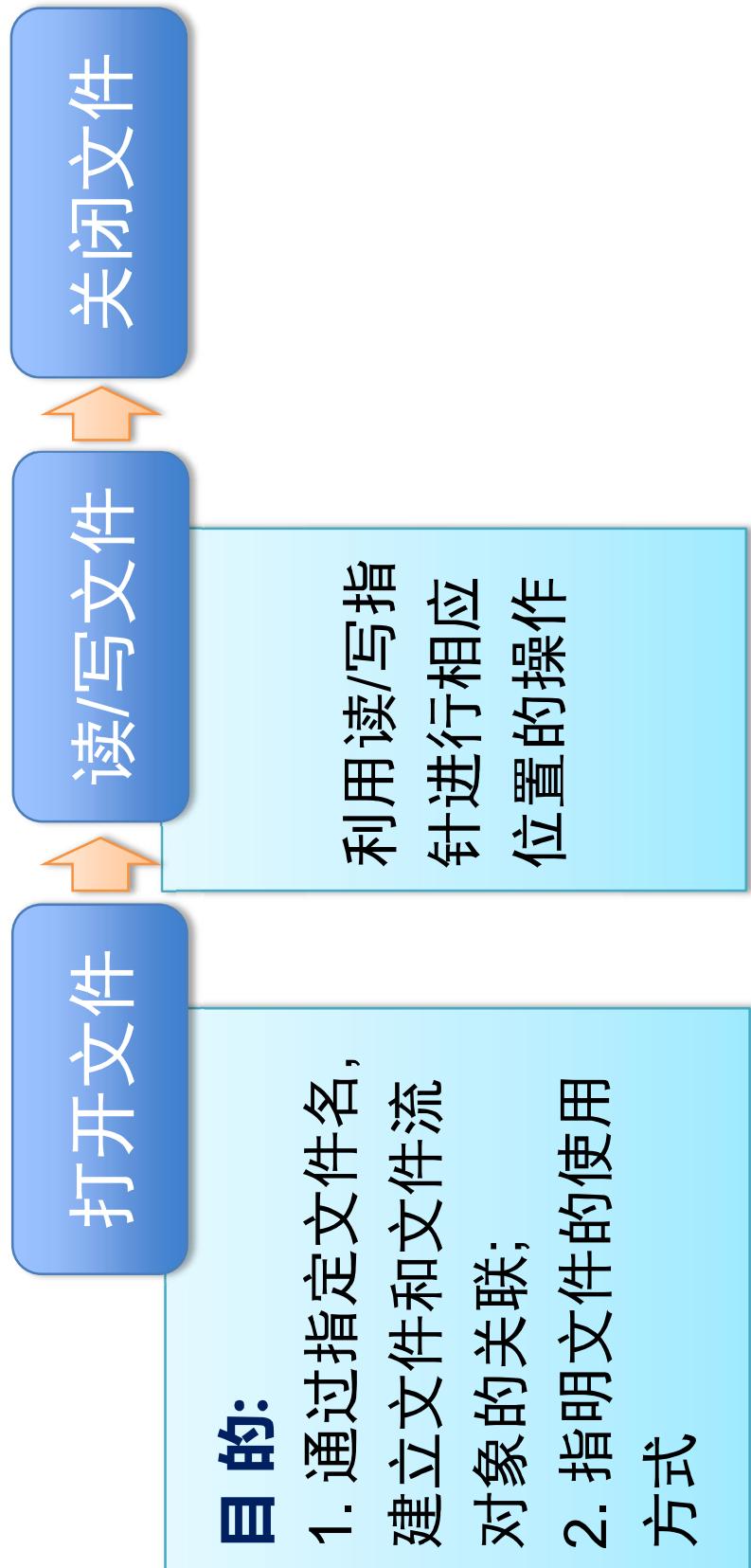
- ▶ 顺序文件 —— 一个有限字符串构成的顺序字符串流
- ▶ C++标准库中: **ifstream**, **ofstream** 和 **fstream** 共3个类  
→ 用于文件操作 —— 统称为**文件流类**



# 文件操作



## ■ 使用/创建文件的基本流程



# 建立顺序文件



fstream  
中  
定义的  
类

将要建立的文  
件的文件名

```
#include <fstream> // 包含头文件  
ofstream outFile("clients.dat", ios::out|ios::binary); //打开文件
```

自定义的  
ofstream类的  
对象

打开并建立文件的选项

- ios::out 输出到文件，删除原有内容
- ios::app 输出到文件，保留原有内容，  
    总是在尾部添加
- ios::binary 以二进制文件格式打开文件

# 建立|||页|||字文件

- 也可以先创建 `ofstream` 对象，再用 `open函数` 打开

```
ofstream fout;
```

```
fout.open(“test.out”, ios::out|ios::binary );
```

- 判断打开是否成功：

```
if(!fout) { cerr << “File open error!”<<endl; }
```

- 文件名可以给出绝对路径，也可以给相对路径
- 没有交代路径信息，就是在当前文件夹下找文件

# 文件的读写指针

- 对于输入文件，有一个**读指针**
- 对于输出文件，有一个**写指针**
- 对于输入输出文件，有一个**读写指针**
- 标识文件操作的当前位置，  
该指针在哪里 → 读写操作就在哪里进行



# 文件的读写指针

```
ofstream fout("a1.out", ios::app);  
long location = fout.tellp(); //取得写指针的位置  
location = 10L;  
fout.seekp(location); // 将写指针移动到第10个字节处  
fout.seekp(location, ios::beg); //从头数location  
fout.seekp(location, ios::cur); //从当前位置数location  
fout.seekp(location, ios::end); //从尾部数location  
► location 可以为负值
```

# 文件的读写指针



```
ifstream fin("a1.in",ios::in);  
long location = fin.tell(); //取得读指针的位置
```

```
location = 10L;  
  
fin.seekg(location); //将读指针移到第10个字节处  
fin.seekg(location,ios::beg); //从头数location  
fin.seekg(location,ios::cur); //从当前位置数location  
fin.seekg(location,ios::end); //从尾部数location
```

- location 可以为负值

# 二进制文件读写

```
int x=10;  
fout.seekp(20, ios::beg);  
fout.write( (const char *)(&x), sizeof(int) );  
  
fin.seekg(0, ios::beg);  
fin.read( (char *)(&x), sizeof(int) );
```

- ▶ 二进制文件读写，直接写二进制数据，记事本看未必正确

## 二进制文件读写

//下面的程序从键盘输入几个学生的姓名的成绩，  
//并以二进制，文件形式存起来

```
#include <iostream>
#include <fstream>
#include <cstring>
using namespace std;
class CStudent {
public:
    char szName[20];
    int nScore;
};
```



```
int main()
{
    CStudent s;
    ofstream OutFile( "c:\\tmp\\students.dat", ios::out|ios::binary );
    while( cin >> s.szName >> s.nScore ) {
        if( strcmp(s.szName, "exit" ) == 0) //名字为exit则结束
            break;
        OutFile.write( (char *) & s, sizeof(s) );
    }
    OutFile.close();
    return 0;
}
```



Note -- 文本文件/二进制文件打开文件的区别:

- 在Unix/Linux下,二者一致,没有区别;
- 在Windows下,文本文件是以“\r\n”作为换行符  
→读出时,系统会将0xd0a只读入0x0a  
→写入时,对于0xa系统会自动写入0x0d

输入:

Tom 60  
Jack 80  
Jane 40  
exit 0

则形成的 students.dat 为 72字节

用记事本打开,呈现:

Tom 烫烫烫烫烫烫烫烫烫烫烫烫烫  
Jane 烫烫烫烫烫烫烫烫?



## 二进制文件读写

//下面的程序将 students.dat 文件的内容读出并显示

```
#include <iostream>
#include <fstream>
using namespace std;
class CStudent
{
public:
    char szName[20];
    int nScore;
};
```



```
int main(){
    CStudent s;
    ifstream inFile("students.dat", ios::in | ios::binary);
    if(!inFile) {
        cout << "error" << endl;
        return 0;
    }
    while( inFile.read( (char*) & s, sizeof(s) ) ){
        int nReadedBytes = inFile.gcount(); //看刚才读了多少字节
        cout << s.szName << " " << s.score << endl;
    }
    inFile.close();
    return 0;
}
```

输出：  
Tom 60  
Jack 80  
Jane 40

## 二进制文件读写

//下面的程序将 students.dat 文件的Jane的名字改成Mike

```
#include <iostream>
#include <fstream>
using namespace std;

class CStudent
{
public:
    char szName[20];
    int nScore;
};
```



```
int main()
{
    CStudent s;
    fstream iofile( "c:\\tmp\\\\students.dat" , ios::in|ios::out|ios::binary);
    if(!iofile) {
        cout << "error" ;
        return 0;
    }
    iofile.seekp( 2 * sizeof(s), ios::beg); //定位写指针到第三个记录
    iofile.write( "Mike", strlen("Mike") + 1);
    iofile.seekg(0, ios::beg); //定位读指针到开头
    while( iofile.read( (char* ) & s, sizeof(s)) )
        cout << s.szName << " " << s.nScore << endl;
    iofile.close();
    return 0;
}
```

输出：  
Tom 60  
Jack 80  
Mike 40

# 显式关闭文件

- ifstream fin("test.dat", ios::in);  
**fin.close();**
- ofstream fout("test.dat", ios::out);  
**fout.close();**

# 例子：mycopy 程序，文件拷贝

//用法示例：

```
//mycopy src.dat dest.dat  
//即将 src.dat 拷贝到 dest.dat  
//如果 dest.dat 原来就有，则原来的文件会被覆盖  
#include <iostream>  
#include <fstream>  
using namespace std;  
  
int main(int argc, char * argv[]){  
    if(argc != 3) {  
        cout << "File name missing!" << endl;  
        return 0;  
    }  
    ifstream inFile(argv[1], ios::binary|ios::in); //打开文件用于读
```

```
if(! inFile) {  
    cout << "Source file open error." << endl;  
    return 0;  
}  
ofstream outFile(argv[2], ios::binary|ios::out); //打开文件用于写  
if(!outFile) {  
    cout << "New file open error." << endl;  
    inFile.close(); //打开的文件一定要关闭  
    return 0;  
}  
char c;  
while(inFile.get(c)) //每次读取一个字符  
    outFile.put(c); //每次写入一个字符  
outFile.close();  
inFile.close();  
return 0;  
}
```