

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>





本节主题：

二进制文件应用案例

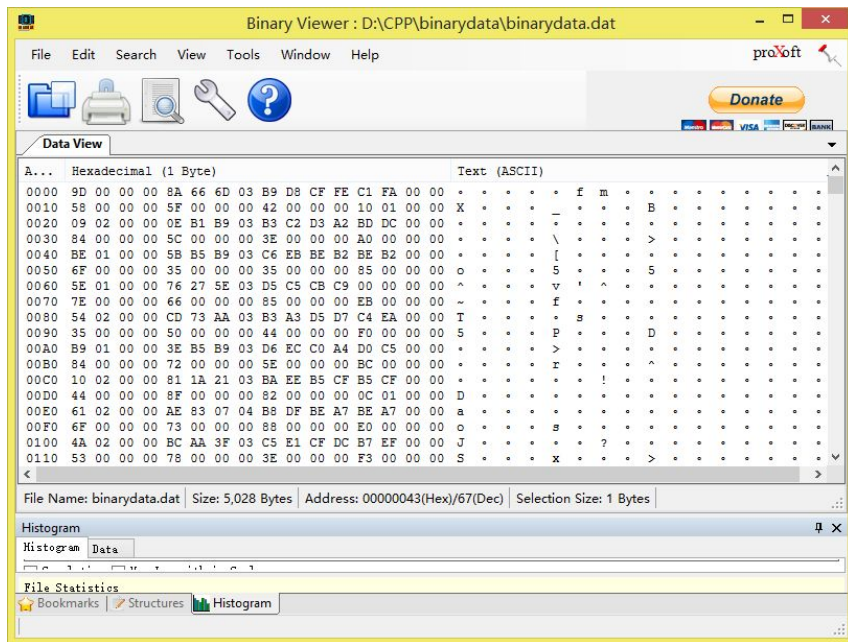
例：高考成绩查询

□ 文本数据和二进制数据的比较：直观、空间、效率、保密

asciidata.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

57501322	关晓龙	88	95	66	272	521
62501134	陈英杰	132	92	62	160	446
62502235	齐静静	111	53	53	133	350
56502134	张松	126	102	133	235	596
61502413	常兆年	53	80	68	240	441
62502206	朱坤信	132	114	94	188	528
52501121	侯迪迪	68	143	130	268	609
67601326	高晶晶	111	115	136	224	586
54504124	裴先凤	83	120	62	243	508
62502209	刁明杰	114	137	65	256	572
62501107	厉彦梅	109	135	74	209	527
67601325	曹爱凤	118	60	87	144	409
67601329	杨丰存	132	113	68	207	520
62502110	姚书东	131	80	131	192	534
69501120	刘玉升	90	63	56	298	507
57501315	赵晨	92	100	90	278	560
62501150	周松松	144	71	99	195	509
53505243	黄甜甜	145	73	85	280	583
52501130	许光明	56	134	101	153	444



系统升级第一步：转换现有数据格式

```
typedef struct
{
    int NO;
    char name[8];
    int chinese;
    int math;
    int english;
    int Comprehensive;
    int total;
} Student; //高考学生信息
const int maxNum = 10000;

int main()
{
    //1. 读入ASCII文件
    //2. 写进二进制文件
    return 0;
}
```

```
//1. 读入ASCII文件
ifstream asciiFile("asciidata.txt", ios::in);
int stuNum = 0;
Student students[maxNum];
while(!asciiFile.eof())
{
    asciiFile>>students[stuNum].NO;
    asciiFile>>students[stuNum].name;
    asciiFile>>students[stuNum].chinese;
    asciiFile>>students[stuNum].math;
    asciiFile>>students[stuNum].english;
    asciiFile>>students[stuNum].Comprehensive;
    asciiFile>>students[stuNum].total;
    stuNum++;
}
asciiFile.close();
```

```
//2. 写入到二进制文件
int i;
ofstream binaryFile("binarydata.dat", ios::out | ios::binary);
binaryFile.write((char*)&stuNum, sizeof(stuNum));
for(i=0; i<stuNum; ++i)
{
    binaryFile.write((char*)&students[i], sizeof(Student));
}
binaryFile.close();
```

二进制文件结构：

- 📁 考生人数;
- 📁 各考生记录

用二进制文件完成业务

```
//2. 各种操作（例排序，还可以查找、修改等）  
sort(students, stuNum);
```

```
typedef struct  
{  
    int NO;  
    char name[8];  
    int chinese;  
    int math;  
    int english;  
    int Comprehensive;  
    int total;  
} Student;  
int main()  
{  
    //1. 读入数据  
    //2. 完成业务  
    //3. 保存数据  
    return 0;  
}
```

```
//1. 读入数据  
int stuNum = 0;  
Student *students;  
fstream binaryFile("binarydata.dat", ios::in|ios::out|ios::binary);  
//读入考生人数  
binaryFile.read((char*)&stuNum, sizeof(stuNum));  
//读入数据  
students = new Student[stuNum];  
int i;  
for(i=0; i<stuNum; ++i)  
{  
    binaryFile.read((char*)&students[i], sizeof(Student));  
}
```

```
//3. 将数据再写入文件  
binaryFile.seekg(0, ios::beg);  
binaryFile.write((char*)&stuNum, sizeof(stuNum));  
for(i=0; i<stuNum; ++i)  
{  
    binaryFile.write((char*)&students[i], sizeof(Student));  
}  
//关闭文件  
binaryFile.close();
```

优点：

- ☐ 省空间，省时间；
- ☐ 一定程度的“数据保密”；
- ☐ 问题：100万考生如何做？数据增删改如何做？

高大上的方案——索引+随机访问

优点：

- 空间、时间的优化；
 - 用有序的索引查无序的数据；
 - 数据更新时数据和索引的同步；
- 大数据安守阵地，小索引密切联系

排序前索引

学号	偏移量
62501134	0
62502235	32
56502134	64
61502413	96
62502206	128
52501121	160
67601326	192

排序后索引

学号	偏移量
52501121	160
56502134	64
61502413	96
62501134	0
62502206	128
62502235	32
67601326	192

数据文件

偏移量	学号	姓名	语	数	外	综	总
0	62501134	陈英杰	132	92	62	160	446
32	62502235	齐静静	111	53	53	133	350
64	56502134	张松	126	102	133	235	596
96	61502413	常兆年	53	80	68	240	441
128	62502206	朱坤信	132	114	94	188	528
160	52501121	侯迪迪	68	143	130	268	609
192	67601326	高晶晶	111	115	136	224	586

索引文件的建立

```
typedef struct
{
    int NO;
    char name[8];
    int chinese;
    int math;
    int english;
    int Comprehensive;
    int total;
} Student;

typedef struct
{
    int NO;
    long offset;
} StudentIndex;
```

```
void writeIndex(StudentIndex *si, int n)
{
    //打开文件
    ofstream indexFile("binarydata.idx",
        ios::out | ios::binary);
    if(!indexFile)
    {
        cerr<<"cannot open index file!"<<endl;
        exit(1);
    }
    int i;
    for(i=0; i<n; ++i)
    {
        indexFile.write((char*)&si[i],
            sizeof(StudentIndex));
    }
    //关闭文件
    indexFile.close();
    return;
}
```

```
int main()
{
    createIndex();
    return 0;
}
```

```
void createIndex() {
    int stuNum;
    StudentIndex *studentsIndex;
    Student student;
    ifstream binaryFile("binarydata.dat", ios::in | ios::binary);
    //建立索引
    binaryFile.read((char*)&stuNum, sizeof(stuNum));
    //读入数据, 建立未排序的索引表
    studentsIndex = new StudentIndex[stuNum];
    int i, j; long mOffset;
    for(i=0; i<stuNum; ++i) {
        mOffset = binaryFile.tellg();
        binaryFile.read((char*)&student, sizeof(Student));
        studentsIndex[i].NO = student.NO;
        studentsIndex[i].offset = mOffset;
    }
    binaryFile.close();
    //为索引表排序
    StudentIndex temp;
    for (i=0; i<stuNum-1; i++)
        for(j=0; j<stuNum-i-1; j++)
            if (studentsIndex[j].NO>studentsIndex[j+1].NO) {
                temp=studentsIndex[j];
                studentsIndex[j]=studentsIndex[j+1];
                studentsIndex[j+1]=temp;
            }
    //将建好的索引表通过文件存储
    writeIndex(studentsIndex, stuNum);
    return;
}
```

索引文件的利用

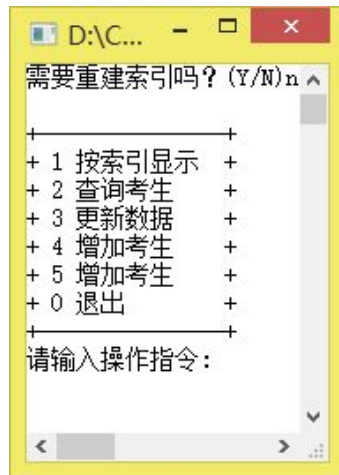
```
typedef struct {...} Student;
typedef struct {...} StudentIndex;
fstream dataFile, indexFile;
int stuNum, maxNum;
StudentIndex *studentsIndex;
```

```
int main(){
    char yn;
    cout<<"重建索引 ? (Y/N)";
    cin>>yn;
    if('Y'==yn || 'y'==yn)
        createIndex();
    init();
    work();
    done();
    return 0;
}
```

```
void done()
{
    dataFile.close();
    writeIndex(studentsIndex, stuNum);
}
```

```
void init(){
    dataFile.open("binarydata.dat", ios::in|ios::out|ios::binary);
    dataFile.read((char*)&stuNum, sizeof(stuNum));
    maxNum = stuNum * 1.1;
    studentsIndex = new StudentIndex[maxNum];
    indexFile.open("binarydata.idx", ios::in|ios::binary);
    int i;
    for(i=0; i<stuNum; ++i)
        indexFile.read((char*)&studentsIndex[i], sizeof(StudentIndex));
    indexFile.close();
    return;
}
```

```
void displayByIndex()
{
    Student stu;
    int i;
    long location;
    cout<<"共"<<stuNum<<"名考生 : "<<endl;
    for(i=0; i<stuNum; ++i)
    {
        location = studentsIndex[i].offset;
        dataFile.seekg(location, ios::beg);
        dataFile.read((char*)&stu, sizeof(Student));
        cout<<i<<": ";
        displayStudent(stu);
    }
}
```



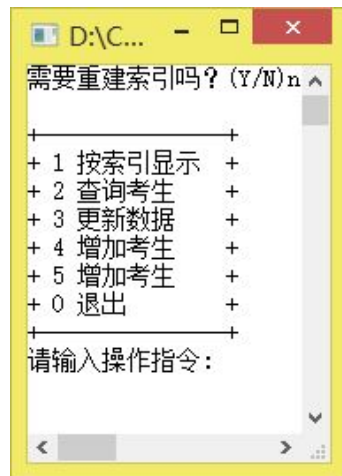
请接着完善程序

排序后索引

学号	偏移量
52501121	160
56502134	64
61502413	96
62501134	0
62502206	128
62502235	32
67601326	192

数据文件

偏移量	学号	姓名	语	数	外	综	总
0	62501134	陈英杰	132	92	62	160	446
32	62502235	齐静静	111	53	53	133	350
64	56502134	张松	126	102	133	235	596
96	61502413	常兆年	53	80	68	240	441
128	62502206	朱坤信	132	114	94	188	528
160	52501121	侯迪迪	68	143	130	268	609
192	67601326	高晶晶	111	115	136	224	586



关于二进制文件

📁 二进制读写文件的优点

- (1) 便于对数据进行随机访问（尤其和索引配合）；
- (2) 存取效率高；
- (3) 可以对文件进行压缩处理；
- (4) 存储密度大；
- (5) 数据文件便于保密；

📁 二进制读写文件需了解

- (1) 二进制读写文件兼容性低；
- (2) 二进制读写文件，不能用字处理程序进行阅读、编辑或修改；

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>

