

C++语言基础

迂者 - 贺利坚

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

<http://edu.csdn.net>





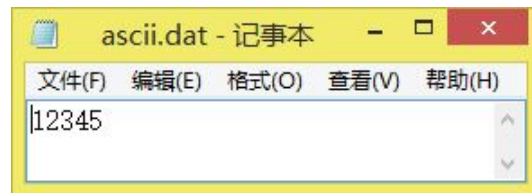
本节主题：

二进制文件及其顺序读写

理解二进制存储

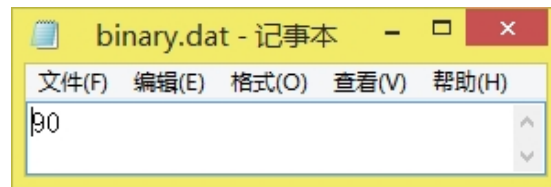
对于数值数据，ASCII形式与二进制形式不同

例：short int x=12345;



ASCII形式存储				
'1' (49)	'2' (50)	'3' (51)	'4' (52)	'5' (53)
00110001	00110010	00110011	00110100	00110101

二进制形式存储	
12345	
00110000	00111001



对比

ASCII文件直观，便于人阅读，但一般占存储空间较多，且需要花费时间转换；

二进制文件是计算机的内部形式，节省空间，且不需要转换，但不能直观显示。

对于字符信息，在内存中是以ASCII代码形式存放，无论用ASCII文件输出还是用二进制文件输出，形式一样的。

对比ASCII文件和二进制文件

//将short int x=12345写入文本文件

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <cstdlib>
```

```
using namespace std;
```

```
int main( )
```

```
{
```

```
    short int x=12345;
```

```
    ofstream outfile("binary.dat");
```

```
    //失败处理
```

```
    outfile<<x<<endl;
```

```
    outfile.close( );
```

```
    return 0;
```

```
}
```



//将short int x=12345写入二进制文件

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <cstdlib>
```

```
using namespace std;
```

```
int main( )
```

```
{
```

```
    short int x=12345;
```

```
    ofstream outfile("binary.dat",ios::binary);
```

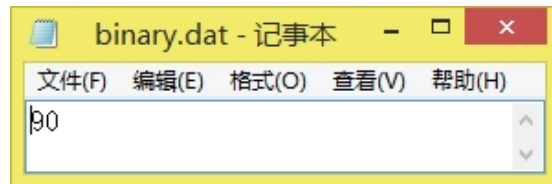
```
    //失败处理
```

```
    outfile.write((char*)&x,2);
```

```
    outfile.close( );
```

```
    return 0;
```

```
}
```



用成员函数read和write读写二进制文件

打开方式

```
ofstream a("file1.dat", ios::out | ios::binary);
```

```
ifstream b("file2.dat", ios::in | ios::binary);
```

文件读写方式

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

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

char *buffer 指向内存中一段存储空间

int len 是读写的字节数

例

a.write(p1, 50) 将p1指向的空间中50个字节存入文件对象a

b.read(p2, 30) 从文件对象b读出30个字节，存之指向空间



将数据以二进制形式存放在磁盘文件中

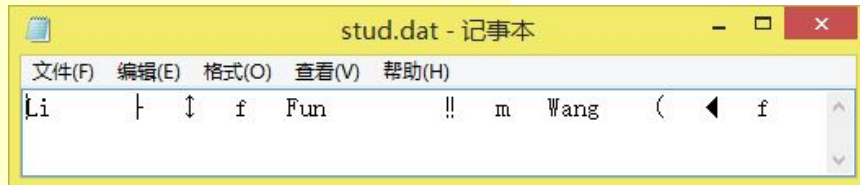
```
#include<iostream>
#include <fstream>
#include<cstdlib>
using namespace std;
struct student{
    char name[5];
    int num;
    int age;
    char sex;
};
student stud[3]= {
    {"Li",25,18,'f'},
    {"Fun",32,19,'m'},
    {"Wang",40,17,'f'}
};
```

```
int main( )
{
    ofstream outfile("stud.dat",ios::binary);
    if(!outfile)
    {
        cerr<<"open error!"<<endl;
        exit(1);//退出程序
    }
    for(int i=0; i<3; i++)
    {
        outfile.write((char*)&stud[i],sizeof(stud[i]));
    }
    cout<<"任务完成，请查看文件。"<<endl;
    outfile.close( );
    return 0;
}
```

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

`outfile.write((char*)&stud[0],sizeof(stud));`

`outfile.write((char*)&stud[i],sizeof(stud[i]));`



查看二进制文件



二进制文件查看器

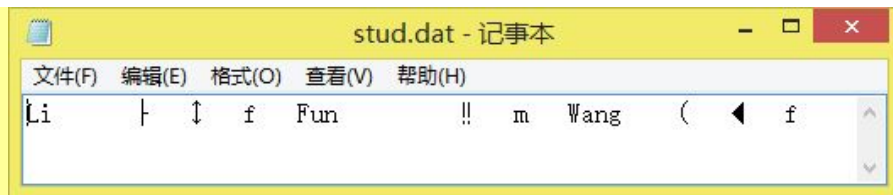


百度一下

```
struct student{
    char name[5];
    int num;
    int age;
    char sex;
};

student stud[3]= {
    {"Li",25,18,'f'},
    {"Fun",32,19,'m'},
    {"Wang",40,17,'f'}
};
```

```
int main( )
{
    ... ..
    for(int i=0;i<3;i++)
        outfile.write((char*)&stud[i], sizeof(stud[i]));
    ... ..
}
```



Hexadecimal	Text (ASCII)
4C 69 00 00 00 00 00 00 19 00 00 00 12 00 00 00	L i
66 00 00 00 46 75 6E 00 00 00 00 00 20 00 00 00	f . . . F u n
13 00 00 00 6D 00 00 00 57 61 6E 67 00 00 00 00 m . . . W a n g
28 00 00 00 11 00 00 00 66 00 00 00	(. f . . .

低地址----->高地址

补充常识：内存对齐、高位优先

将二进制文件中的数据读入内存

```
#include<iostream>
#include<fstream>
#include<cstdlib>
using namespace std;
struct student
{
    char name[5];
    int num;
    int age;
    char sex;
};
```

- ❏ 磁盘文件中的数据都按照同一结构体的格式进行读写，能够获得匹配。
- ❏ 常常定义某一领域的标准文件格式，各生产厂商依此开发各种产品，例音、视频。

```
for(i=0; i<3; i++)
{
    cout<<"NO."<<i+1<<endl;
    cout<<"name:"<<stud[i].name<<endl;
    cout<<"num:"<<stud[i].num<<endl;;
    cout<<"age:"<<stud[i].age<<endl;
    cout<<"sex:"<<stud[i].sex<<endl<<endl;
}
```

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

```
int main( )
{
    student stud[3];
    int i;
    ifstream infile("stud.dat",ios::binary);
    if(!infile)
    {
        cerr<<"open error!"<<endl;
        exit( );
    }
    infile.read((char*)&stud[0],sizeof(stud));
    for(i=0; i<3; i++)
        infile.read((char*)&stud[i],sizeof(stud[i]));
    infile.close( );
    // 显示文件中的数据
    return 0;
}
```


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/>