



Introduction to C++ (Season 2)

Unit 7: File Input and Output

第7单元:出入虽同趣,所向各有宜—文件输入输出

Section 06 : Binary IO

第06节: 二进制读写



Binary I/O (二进制读写)

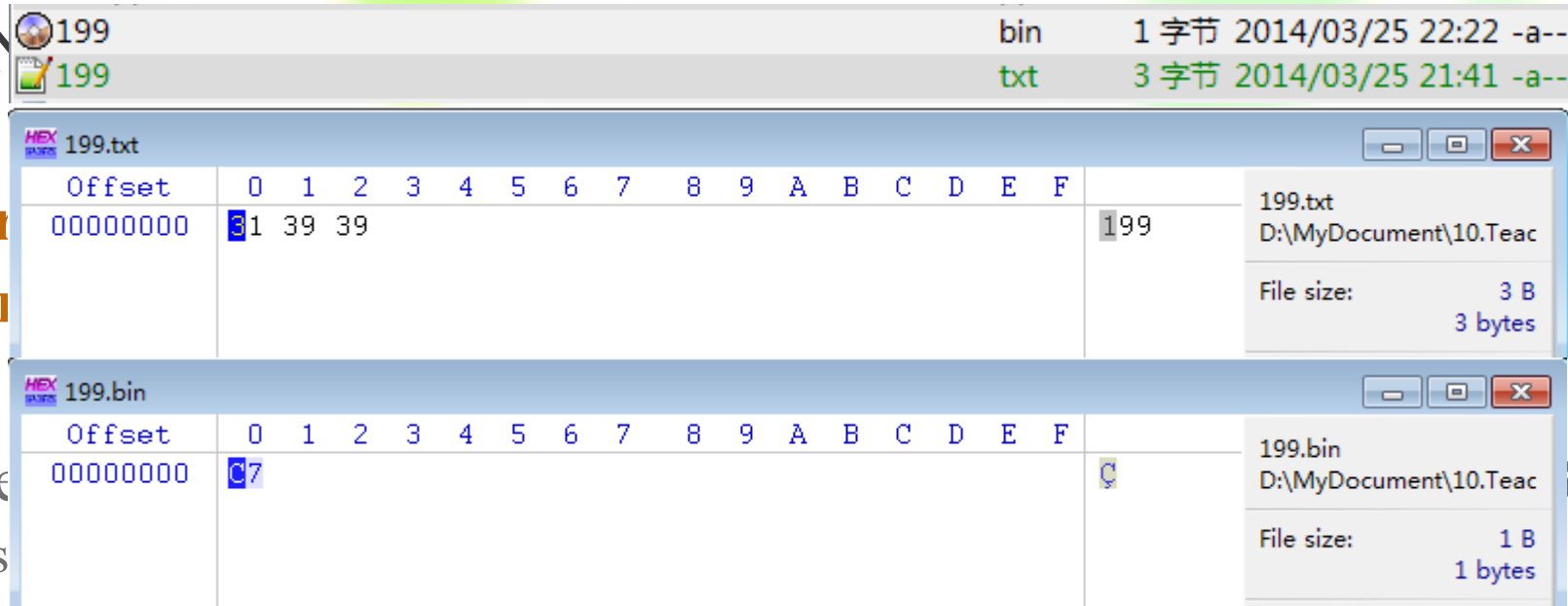
❖ TEXT file vs BIN

- Both **stores** as a
- text file : **inter**
- binary file : **inter**

❖ For example, the

- in text file: as the
 - stores as 3 bytes
- in bin file: as a byte-type value C7 (decimal 199 = hex C7 (在二进制文件中存为C7))
 - stores as 1 bytes: 0xC7

❖ Text I/O is built upon binary I/O to provide a level of abstraction for character encoding and decoding. (文本模式的读写是建立在二进制模式读写的基础上的, 只不过是将二进制信息进行了字符编解码)



ios::binary

❖ Binary I/O does not require conversions. (二进制读写无需信息转换)

- numeric value → write (bin I/O) → file
- value in memory → copy (no conversion) → file

The two
are equal

❖ How to perform binary I/O ? (如何进行二进制读写)

- By default, a file is opened in text mode. (文件默认以文本模式打开)
- open a file using the binary mode ios::binary. (用ios::binary以二进制模式打开文件)

❖ Text I/O function: (文本模式读写函数)

- write: << operator; put()
- read : >> operator; get(); getline()

❖ Binary I/O function: (二进制模式读写函数)

- write: write();
- read : read();

The write Function

The syntax for the write function is (write函数的语法如下)

streamObject.write(char * address, int size)

Write Any Type (将任意类型数据写入文件)

- ❖ How to write data other than characters? (如何将非字符数据写入文件)
 - Convert any data into a sequence of bytes (byte stream) (转换为字节序列, 即字节流)
 - Write the sequence of bytes to file with write() (用write函数将字节序列写入文件)
- ❖ How to convert any data into byte stream? (如何将信息转换为字节流)
 - reinterpret_cast : cast the address of a primitive type value or an object to a character array pointer for binary I/O. (将数据的地址转换为为字符类型指针用于二进制读写)
 - syntax: reinterpret_cast<dataType>(address)
 - address is the starting address of the data (primitive, array, or object)
 - dataType is the data type you are converting to.
 - In this case for binary I/O, dataType is char *.

The read Function

❖ The syntax for the read function is (read函数的语法如下)

`streamObject.read(char * address, int size)`

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

int main() {
    fstream binaryio;
    binaryio.open("city.dat", ios::in |
ios::binary);
    char s[10];
    binaryio.read(s, 5);
    s[5] = '\0';
    cout << s;
    binaryio.close();
    return 0;
}
```

H a n g z \0
0 1 2 3 4 5

Read Any Type (读任意类型数据)

❖ How to read data other than characters?

- use reinterpret_cast operator

❖ Example:

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

int main() {
    fstream binaryio;
    binaryio.open("temp.dat", ios::in | ios::binary);
    int value;
    binaryio.read(reinterpret_cast<char *>(&value),
sizeof(value));
    cout << value;
    binaryio.close();
    return 0;
}
```