

文件存储在磁盘都有其编码

例如：文件本身是**UTF-8**的编码

通过**UTF-8**解码文件内容，不会乱码

在通过**UTF-8**编码数据，不会乱码

ASCII(American Standard Code for Information Interchange, 美国信息交换标准代码)

128个

标准ASCII 码也叫基础ASCII码，使用7 位**二进制数**（剩下的1位二进制为0）来表示所有的大写和小写字母，数字0 到9、标点符号，以及在美式英语中使用的特殊**控制字符**。

iso-8859-1

256个字符

8个bit表示一个字符，用**一个字节 (byte) (8bit)** 表示一个字符，共计可以表示256字符

unicode

采用2个字节表示一个字符，全球通用，存储太大

UTF(Unicode Translation Format)

unicode是一种编码方式，utf是存储方式；utf-8是unicode的一种实现方式

UTF-16LE (little endian) 小端,UTF-16BE (big endian) 大端

也是采用 2个字节表示一个字符，**定长字节**

开头会加上不存在的字符：

zero width no-break space;

0xFEFF (BE)

0xFFFF (LE)

UTF-8

变长字节表示

一般来说，UTF-8用3个字符表示一个中文

BOM (Byte Order Mark)

开头大小端

gb2312

2个字节表示一个汉字

gbk (是gb2312的超集)

gbk加入了一些生僻字，兼容gb2312

gb18030 (是gbk的超级)

兼容gbk

big5

台湾使用的繁体字编码

对于汉字：你好

使用iso-8859-1解码，出现乱码

```
1 Charset charset = Charset.forName("iso-8859-1");
2 CharsetDecoder charsetDecoder = charset.newDecoder();//字节数组转字符串,解码
3
4 CharBuffer charBuffer = charsetDecoder.decode(inputData);
5 System.out.println(new String(charBuffer.array()));
```

结果：ä½ ä½

iso-8859-1是一个字节表示一个字符，文件是UTF-8的三个字节表示一个汉字，汉字“你好”的你被分成三个字符“ä½”，好被分成三个字符“ä½”。

但如果你解码的时候还是使用iso-8859-1；

生成的文件以UTF-8编码，

这六个字符又被组装成你好

```
1 public class NioTest13 {
2     public static void main(String[] args) throws Exception{
3         String inputFile = "src/main/java/com/zhangtianyi/nio/resource/NioTest13_In.txt";
4         String outputFile = "src/main/java/com/zhangtianyi/nio/resource/NioTest13_Out.txt";
```

```
5 RandomAccessFile inputRandomAccessFile = new RandomAccessFile(inputFile,
"r");
6 RandomAccessFile outputRandomAccessFile = new RandomAccessFile(outputFile,
"rw");
7
8 long fileLength = new File(inputFile).length();
9
10 FileChannel inputChannel = inputRandomAccessFile.getChannel();
11 FileChannel outputChannel = outputRandomAccessFile.getChannel();
12 //内存映射文件
13 MappedByteBuffer inputData = inputChannel.map(FileChannel.MapMode.READ_
ONLY, 0, fileLength);
14
15 System.out.println("*****");
16 Charset.availableCharsets().entrySet().forEach(entry -> {
17 System.out.println(entry.getKey() + ":" + entry.getValue());
18 });
19 System.out.println("*****");
20
21
22 // Charset charset = Charset.forName("Utf-8");
23 Charset charset = Charset.forName("iso-8859-1");
24 CharsetDecoder charsetDecoder = charset.newDecoder();//字节数组转字符串,
解码
25 CharsetEncoder charsetEncoder = charset.newEncoder();//字符串转字节数组,
编码
26
27 CharBuffer charBuffer = charsetDecoder.decode(inputData);
28 System.out.println(new String(charBuffer.array()));
29 for (int i = 0; i < charBuffer.array().length; i++) {
30 System.out.println(charBuffer.array()[i]);
31 }
32
33 ByteBuffer outputData = charsetEncoder.encode(charBuffer);
34
35 outputChannel.write(outputData);
36
37 inputChannel.close();
38 outputChannel.close();
39
40 }
41 }
```

