**中文字符编码和通用编码**



[百炼成钢百忍成佛](https://blog.csdn.net/beyondqd) 2011-11-08 13:39:14  14851  收藏 3

文章标签： [扩展](https://www.csdn.net/gather_20/MtTaEg0sMjM2MTktYmxvZwO0O0OO0O0O.html) [语言](https://www.csdn.net/gather_2e/MtTaEg0sMDU5NTUtYmxvZwO0O0OO0O0O.html) [character](https://www.csdn.net/gather_2f/MtTaEg0sNTcxNTAtYmxvZwO0O0OO0O0O.html) [存储](https://www.csdn.net/gather_2d/MtTaEg0sNDg1MTYtYmxvZwO0O0OO0O0O.html) [图形](https://so.csdn.net/so/search/s.do?q=%E5%9B%BE%E5%BD%A2&t=blog&o=vip&s=&l=&f=&viparticle=) [windows](https://www.csdn.net/gather_28/OtDaQgysNTkwNy1ibG9n.html)

  字符是各种文字和符号的总称，包括各国家文字、标点符号、图形符号、数字等。字符集是多个字符的集合，字符集种类较多，每个字符集包含的字符个数不同，常见字符集名称：ASCII字符集、GB2312字符集、BIG5字符集、 GB 18030字符集、Unicode字符集等。计算机要准确的处理各种字符集文字，需要进行字符编码，以便计算机能够识别和存储各种文字。

中文文字数目大，而且还分为简体中文和繁体中文两种不同书写规则的文字，而计算机最初是按英语单字节字符设计的，因此，对中文字符进行编码，是中文信息交流的技术基础。本文将按照字符集的时间顺序讨论几种典型的字符集，选取几种代表性的中文字符集，研究历史由来、特点、技术特征。

汉字编码范围

名称 第一字节 第二字节

GB2312 0xB0-0xF7(176-247)    0xA0-0xFE（160-254）

GBK 0x81-0xFE（129-254）     0x40-0xFE（64-254）

Big5 0x81-0xFE（129-255）     0x40-0x7E（64-126）

0xA1－0xFE（161-254）

ASCII 字符集

1．名称的由来

ASCII（American Standard Code for Information Interchange，美国信息互换标准代码）是基于罗马字母表的一套电脑编码系统。

2．特点

它主要用于显示现代英语和其他西欧语言。它是现今最通用的单字节编码系统，并等同于国际标准ISO 646。

3．包含内容

控制字符：回车键、退格、换行键等。

可显示字符：英文大小写字符、阿拉伯数字和西文符号

4．技术特征

7位（bits）表示一个字符，共128字符

5．ASCII扩展字符集

7位编码的字符集只能支持128个字符，为了表示更多的欧洲常用字符对ASCII进行了扩展，ASCII扩展字符集使用8位（bits）表示一个字符，共256字符。

ASCII扩展字符集比ASCII字符集扩充出来的符号包括表格符号、计算符号、希腊字母和特殊的拉丁符号。

GB2312 字符集

 1．名称的由来

GB2312又称为GB2312-80字符集，全称为《信息交换用汉字编码字符集·基本集》，由原中国国家标准总局发布，1981年5月1日实施。

2．特点

GB2312是中国国家标准的简体中文字符集。它所收录的汉字已经覆盖99.75%的使用频率，基本满足了汉字的计算机处理需要。在中国大陆和新加坡获广泛使用。

3．包含内容

GB2312收录简化汉字及一般符号、序号、数字、拉丁字母、日文假名、希腊字母、俄文字母、汉语拼音符号、汉语注音字母，共 7445 个图形字符。其中包括6763个汉字，其中一级汉字3755个，二级汉字3008个；包括拉丁字母、希腊字母、日文平假名及片假名字母、俄语西里尔字母在内的682个全角字符。

4．技术特征

（1）分区表示：

GB2312中对所收汉字进行了“分区”处理，每区含有94个汉字/符号。这种表示方式也称为区位码。

各区包含的字符如下：01-09区为特殊符号；16-55区为一级汉字，按拼音排序；56-87区为二级汉字，按部首/笔画排序；10-15区及88-94区则未有编码。

（2）双字节表示

两个字节中前面的字节为第一字节，后面的字节为第二字节。习惯上称第一字节为“高字节” ，而称第二字节为“低字节”。

“高位字节”使用了0xA1-0xF7 (把01-87区(88-94区未有编码)的区号加上0xA0)，“低位字节”使用了0xA1-0xFE (把01-94加上0xA0)。

5．编码举例

以GB2312字符集的第一个汉字“啊”字为例，它的区号16，位号01，则区位码是1601，在大多数计算机程序中，高字节和低字节分别加0xA0得到程序的汉字处理编码0xB0A1。计算公式是：0xB0=0xA0+16, 0xA1=0xA0+1。

GBK 字符集

1．名称的由来

GBK是GB2312的扩展，是向上兼容的，因此GB2312中的汉字的编码与GBK中汉字的相同。另外，GBK中还包含繁体字的编码，它与Big5编码之间的关系我还没有弄明白，好像是不一致的。

2. 特点

GBK中

**一 预备知识**1，字符：字符是抽象的最小文本单位。它没有固定的形状（可能是一个字形），而且没有值。“A”是一个字符，“ ”（德国、法国和许多其他欧洲国家通用货币的标志）也是一个字符。“中”“国”这是两个汉字字符。字符仅仅代表一个符号，没有任何实际值的意义。  
2，字符集：字符集是字符的集合。例如，汉字字符是中国人最先发明的字符，在中文、日文、韩文和越南文的书写中使用。这也说明了字符和字符集之间的关系，字符组成字符集（iso8859-1，GB2312/GBK，unicode）。  
3，代码点：字符集中的每个字符都被分配到一个“代码点”。每个代码点都有一个特定的唯一数值，称为标值。该标量值通常用十六进制表示。  
4，代码单元： 在每种编码形式中，代码点被映射到一个或多个代码单元。“代码单元”是各个编码方式中的单个单元。代码单元的大小等效于特定编码方式的位数：  
UTF-8 ：UTF-8 中的代码单元由 8 位组成；在 UTF-8 中，因为代码单元较小的缘故，每个代码点常常被映射到多个代码单元。代码点将被映射到一个、两个、三个或四个代码单元；  
UTF-16 ：UTF-16 中的代码单元由 16 位组成；UTF-16 的代码单元大小是 8 位代码单元的两倍。所以，标量值小于 U+10000 的代码点被编码到单个代码单元中；  
UTF-32：UTF-32  中的代码单元由 32 位组成； UTF-32 中使用的 32 位代码单元足够大，每个代码点都可编码为单个代码单元；  
GB18030：GB18030  中的代码单元由 8 位组成；在 GB18030 中，因为代码单元较小的缘故，每个代码点常常被映射到多个代码单元。代码点将被映射到一个、两个或四个代码单元。  
5，举例：

“中国北京香蕉是个大笨蛋”这是我定义的aka字符集；各字符对应代码点为：  
北 00000001  
京 00000010  
香 10000001  
蕉 10000010  
是 10000100  
个 10001000  
大 10010000  
笨 10100000  
蛋 11000000  
中 00000100  
国 00001000  
下面是我定义的 zixia 编码方案（8位），可以看到它的编码中表示了aka字符集的所有字符对应的 代码单元；  
北 10000001  
京 10000010  
香 00000001  
蕉 00000010  
是 00000100  
个 00001000  
大 00010000  
笨 00100000  
蛋 01000000  
中 10000100  
国 10001000  
所谓文本文件 就是我们按一定编码方式将二进制数据表示为对应的文本如 00000001000000100000010000001000000100000010000001000000这样的文件。我用一个支持 zixia编码和aka字符集的记事本打开，它就按照编码方案显示为  “香蕉是个大笨蛋 ”  
如果我把这些字符按照GBK另存一个文件，那么则肯定不是这个，而是  
1100111111100011 1011110110110110 1100101011000111 1011100011110110 1011010011110011 1011000110111111 1011010110110000 110100001010

**二，字符集**1， 常用字符集分类  
ASCII及其扩展字符集  
作用：表语英语及西欧语言。  
位数：ASCII是用7位表示的，能表示128个字符；其扩展使用8位表示，表示256个字符。  
范围：ASCII从00到7F，扩展从00到FF。  
ISO-8859-1字符集  
作用：扩展ASCII，表示西欧、希腊语等。  
位数：8位，  
范围：从00到FF，兼容ASCII字符集。  
GB2312字符集  
作用：国家简体中文字符集，兼容ASCII。  
位数：使用2个字节表示，能表示7445个符号，包括6763个汉字，几乎覆盖所有高频率汉字。  
范围：高字节从A1到F7, 低字节从A1到FE。将高字节和低字节分别加上0XA0即可得到编码。  
BIG5字符集  
作用：统一繁体字编码。  
位数：使用2个字节表示，表示13053个汉字。  
范围：高字节从A1到F9，低字节从40到7E，A1到FE。  
GBK字符集  
作用：它是GB2312的扩展，加入对繁体字的支持，兼容GB2312。  
位数：使用2个字节表示，可表示21886个字符。  
范围：高字节从81到FE，低字节从40到FE。  
GB18030字符集  
作用：它解决了中文、日文、朝鲜语等的编码，兼容GBK。  
位数：它采用变字节表示(1 ASCII，2，4字节)。可表示27484个文字。  
范围：1字节从00到7F; 2字节高字节从81到FE，低字节从40到7E和80到FE；4字节第一三字节从81到FE，第二四字节从30到39。  
UCS字符集  
作用：国际标准 ISO 10646 定义了通用字符集 (Universal Character Set)。它是与UNICODE同类的组织，UCS-2和UNICODE兼容。  
位数：它有UCS-2和UCS-4两种格式，分别是2字节和4字节。  
范围：目前，UCS-4只是在UCS-2前面加了0×0000。  
UNICODE字符集  
作用：为世界650种语言进行统一编码，兼容ISO-8859-1。  
位数：UNICODE字符集有多个编码方式，分别是UTF-8，UTF-16和UTF-32。  
2 ，按所表示的文字分类  
语言                                 字符集                                     正式名称  
英语、西欧语                     ASCII，ISO-8859-1                MBCS 多字节  
简体中文                             GB2312                                    MBCS 多字节  
繁体中文                             BIG5                                         MBCS 多字节  
简繁中文                             GBK                                         MBCS 多字节  
中文、日文及朝鲜语         GB18030                                  MBCS 多字节  
各国语言                             UNICODE，UCS                    DBCS 宽字节

**三，编码**UTF-8：采用变长字节 (1 ASCII, 2 希腊字母, 3 汉字, 4 平面符号) 表示，网络传输, 即使错了一个字节，不影响其他字节，而双字节只要一个错了，其他也错了，具体如下：  
如果只有一个字节则其最高二进制位为0；如果是多字节，其第一个字节从最高位开始，连续的二进制位值为1的个数决定了其编码的字节数，其余各字节均以10开头。UTF-8最多可用到6个字节。

UTF-16：采用2字节，Unicode中不同部分的字符都同样基于现有的标准。这是为了便于转换。从 0×0000到0×007F是ASCII字符，从0×0080到0×00FF是ISO-8859-1对ASCII的扩展。希腊字母表使用从0×0370到 0×03FF 的代码，斯拉夫语使用从0×0400到0×04FF的代码，美国使用从0×0530到0×058F的代码，希伯来语使用从0×0590到0×05FF的代 码。中国、日本和韩国的象形文字（总称为CJK）占用了从0×3000到0×9FFF的代码；由于0×00在c语言及操作系统文件名等中有特殊意义，故很 多情况下需要UTF-8编码保存文本，去掉这个0×00。举例如下：  
UTF-16: 0×0080  = 0000 0000 1000 0000  
UTF-8:   0xC280 = 1100 0010 1000 0000  
UTF-32：采用4字节。  
优缺点  
UTF-8、UTF-16和UTF-32都可以表示有效编码空间 (U+000000-U+10FFFF) 内的所有Unicode字符。  
使用UTF-8编码时ASCII字符只占1个字节，存储效率比较高，适用于拉丁字符较多的场合以节省空间。  
对于大多数非拉丁字符（如中文和日文）来说，UTF-16所需存储空间最小，每个字符只占2个字节。  
Windows NT内核是Unicode（UTF-16），采用UTF-16编码在调用系统API时无需转换，处理速度也比较快。  
采用UTF-16和UTF-32会有Big Endian和Little Endian之分，而UTF-8则没有字节顺序问题，所以UTF-8适合传输和通信。  
UTF-32采用4字节编码，一方面处理速度比较快，但另一方面也浪费了大量空间，影响传输速度，因而很少使用。

**四，如何判断字符集**1，字节序  
首先说一下字节序对编码的影响，字节序分为Big Endian字节序和Little Endian字节序。不同的处理器可能不一样。所以，传输时需要告诉处理器当时的编码字节序。对于前者而言，高位字节存在低地址，低字节存于高地址；后者相反。例如，0X03AB,  
Big Endian字节序  
0000: 0 3  
0001: AB  
Little Endian字节序是  
0000: AB  
0001: 0 3  
2，编码识别  
UNICODE，根据前几个字节可以判断UNICODE字符集的各种编码，叫做Byte Order Mask方法BOM：  
UTF-8: EFBBBF (符合UTF-8格式，请看上面。但没有含义在UCS即UNICODE中)  
UTF-16 Big Endian：FEFF (没有含义在UCS-2中)  
UTF-16 Little Endian：FFFE (没有含义在UCS-2中)  
UTF-32 Big Endian：0000FEFF (没有含义在UCS-4中)  
UTF-32 Little Endian：FFFE0000 (没有含义在UCS-4中)

GB2312：高字节和低字节的第1位都是1。

BIG5，GBK&GB18030：高字节的第1位为1。操作系统有默认的编码，常为GBK，可以下载别的并升级。通过判断高字节的第1位从而知道是ASCII或者汉字编码。