XML作业

姓名：胡卓凡

班级：软工11502

目录

[Unicode 1](#_Toc492813247)

[来历以及发明原因： 1](#_Toc492813248)

[实现or解决问题： 1](#_Toc492813249)

[编码方式： 1](#_Toc492813250)

[是否兼容汉字（或者其他语言）： 1](#_Toc492813251)

[编码集相互转换： 1](#_Toc492813252)

[GB2312 2](#_Toc492813253)

[来历以及发明原因： 2](#_Toc492813254)

[编码方式： 2](#_Toc492813255)

[是否兼容其他语言： 2](#_Toc492813256)

[Big5 2](#_Toc492813257)

[来历以及发明原因： 2](#_Toc492813258)

[编码方式： 3](#_Toc492813259)

[存在的问题： 3](#_Toc492813260)

[冲码问题 3](#_Toc492813261)

[私人造字 4](#_Toc492813262)

[UTF-8/16 4](#_Toc492813263)

[Utf8： 4](#_Toc492813264)

[Utf16： 4](#_Toc492813265)

# Unicode

## 来历以及发明原因：

Unicode 是为了解决传统的 字符编码方案的局限而产生的，例如ISO 8859所定义的字符虽然在不同的国家中广泛地使用，可是在不同国家间却经常出现不兼容的情况。很多传统的 [编码方式](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=50014&ss_c=ssc.citiao.link)都有一个共同的问题，即容许电脑处理双语环境（通常使用 拉丁字母以及其本地语言），但却无法同时支持多语言环境（指可同时处理多种语言混合的情况）。

## 实现or解决问题：

能够使计算机实现跨语言、 跨平台的文本 转换及处理。

## 编码方式：

Unicode是国际组织制定的可以容纳世界上所有文字和符号的[字符编码](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=738663&ss_c=ssc.citiao.link" \t "_blank)方案。目前的Unicode字符分为17组编排，0x0000 至 0xFFFF，每组称为平面（Plane），而每平面拥有65536个码位，共1114112个。然而目前只用了少数平面。 [UTF](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=101264050&ss_c=ssc.citiao.link)-8、 [UTF-16](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=11006007&ss_c=ssc.citiao.link)、 [UTF-32](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=68495805&ss_c=ssc.citiao.link)都是将[数字转换](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=69265523&ss_c=ssc.citiao.link" \t "_blank)到程序数据的编码方案。

[通用字符集](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=8804219&ss_c=ssc.citiao.link)（Universal Character Set, UCS）是由ISO制定的ISO 10646（或称ISO/IEC 10646）标准所定义的标准[字符集](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=291629&ss_c=ssc.citiao.link" \t "_blank)。UCS-2用两个字节编码，UCS-4用4个字节编码。

## 是否兼容汉字（或者其他语言）：

兼容汉字（以及其他语言）；“汉字”的UTF-8编码需要6个字节。“汉字”的UTF-16编码需要两个char16\_t，大小是4个字节。“汉字”的UTF-32编码需要两个char32\_t，大小是8个字节。

## 编码集相互转换：

根据[字节序](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=44712328&ss_c=ssc.citiao.link" \t "_blank)的不同，UTF-16可以被实现为UTF-16LE或UTF-16BE，UTF-32可以被实现为UTF-32LE或UTF-32BE。下面介绍UTF-8、UTF-16、UTF-32、字节序和BOM。

# GB2312

## 来历以及发明原因：

《信息交换用汉字编码[字符集](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=291629&ss_c=ssc.citiao.link" \t "_blank)》是由中国国家标准总局1980年发布，1981年5月1日开始实施的一套国家标准，标准号是GB2312—1980。它是计算机可以识别的编码，适用于汉字处理、汉字通信等系统之间的信息交换。

## 编码方式：

在使用GB2312的程序中，通常采用EUC储存方法，以便兼容于ASCII。 浏览器编码表上的“GB2312”，通常都是指“EUC-CN”表示法。

每个汉字及符号以两个字节来表示。第一个字节称为“高位字节”（也称“区字节）”，第二个字节称为“低位字节”（也称“位字节”）。

“高位字节”使用了0xA1-0xF7(把01-87区的区号加上0xA0)，“低位字节”使用了0xA1-0xFE(把01-94加上 0xA0)。 由于一级汉字从16区起始，汉字区的“高位字节”的范围是0xB0-0xF7，“低位字节”的范围是0xA1-0xFE，占用的码位是 72\*94=6768。其中有5个空位是D7FA-D7FE。

例如“啊”字在大多数程序中，会以两个字节，0xB0（第一个字节） 0xA1（第二个字节）储存。区位码=区字节+位字节（与区位码对比：0xB0=0xA0+16,0xA1=0xA0+1）。

## 是否兼容其他语言：

GB2312编码通行于中国大陆；新加坡等地也采用此编码。中国大陆几乎所有的中文系统和国际化的软件都支持GB2312。1995年颁布的《汉字编码扩展规范》（GBK）与GB2312—1980国家标准所对应的内码标准兼容，同时在字汇一级支持ISO/IEC10646—1和GB 13000—1的全部中、日、韩（CJK）汉字，共计20902字。

# Big5

## 来历以及发明原因：

“[五大码](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=828398)”（Big5)是在1984年由台湾13家厂商与台湾地区[财团法人](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=70802117" \t "_blank)[信息工业](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=74320188)策进会为五大中文套装软件（[宏碁](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=95850)、神通、[佳佳](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=64598952)、零壹、大众）所设计的中文[内码](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=8404524&ss_c=ssc.citiao.link" \t "_blank)，所以就称为Big5中文内码，虽然五大套装软件并没有成功，但Big5码却深远地影响中文电脑内码，直至今日。“五大码”的英文名称“Big5”后来被人按英文字序译回中文，以致现在有“五大码”和“[大五码](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=828398)”两个中文名称。

Big5码的产生，是因为当时台湾不同厂商各自推出不同的编码，如IBM 5550、[王安](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=64622189" \t "_blank)码等，彼此不能兼容；另一方面，台湾当时尚未推出官方的[汉字编码](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=3554862)，而中国内地所推行的GB 2312编码，亦未有收录繁体字。同时，这个计划对于以台湾为核心的亚洲汉字圈也产生了久远的影响。

实现or解决问题：

当时其他的汉字编码与现行[英语软件](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=46615122)欠缺相容。

## 编码方式：

ig5码是一套双字节字符集，使用了双八码储存方法，以两个字节来安放一个字。第一个字节称为“高位字节”，第二个字节称为“低位字节”。“高位字节”使用了0x81-0xFE，“低位字节”使用了0x40-0x7E，及0xA1-0xFE。在Big5的分区中：

0x8140-0xA0FE 保留给使用者自定义字符（造字区）

0xA140-0xA3BF 标点符号、[希腊字母](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=437970" \t "_blank)及特殊符号，包括在0xA259-0xA261，安放了[双音节](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7633173" \t "_blank)度量衡单位用字：兙兛兞兝兡兣嗧瓩糎。

0xA3C0-0xA3FE 保留。此区没有开放作造字区用。

0xA440-0xC67E [常用汉字](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7801281)，先按笔划再按部首排序。

## 存在的问题：

由于各厂商及政府推出的Big5延伸，彼此互不兼容，造成[乱码](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=262888&ss_c=ssc.citiao.link" \t "_blank)问题。鉴于Unicode能正确地处理七万多个汉字，近年的操作系统和应用程序（如[苹果电脑](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=135570" \t "_blank)Mac OS X和以CocoaAPI撰写之程序、MicrosoftWindows 2000及之后版本、Microsoft [Office2000](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=6988716)及之后版本、[Mozilla浏览器](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=63417153&ss_c=ssc.citiao.link" \t "_blank)、Internet Explorer浏览器、Java语言等等），已改用Unicode编码。可惜现时仍有一些旧的软件（如VisualBasic6、部分Telnet或BBS软件），未能支持Unicode编码，故相信Big5缺字的问题仍会困扰用户一段时间，直至所有程序都能改用Unicode为止。

### 冲码问题

因为低位[元字符](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=57356587)中包含了编程语言、[shell](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=267368)、script 中，字串或命令常会用到的[特殊字符](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=269089)，例如0x5C “\”、0x7C “|”等。“\” 在许多用途的字串中是当作转义符号又称为跳脱字符，例如 \n（换行）、\r（归位）、\t（tab）、\\（\本身符号）、\"（[引号](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=461966&ss_c=ssc.citiao.link)）等等。而 “|” 在[UNIX操作系统](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=5436069" \t "_blank)中大多当作命令管线的使用，如 "ls -la | more" 等等。如果在字串中有这些特殊的[转义字符](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=262922)，会

BIG-5

被程式或[直译器](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=584648)解释为特殊用途。但是因为是中文的原因，故无法正确解释为上面所述的行为，因此程式可能会忽略此转义符号或是中断执行。若此，就违反了使用者本来要当成中文字符一部份使用的本意。

在[常用字](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7547535" \t "_blank)如“功”（0xA55C)、“许”（0xB35C)、“盖”（0xBB5C)、“育”（0xA87C)中时常出现，造成了许多软件无法正确处理以Big5编码的字串或文件。这个问题被戏谑性地人名化，称为“[许功](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=73899658)盖”或“许盖功”（这三个字都有这种问题）。但是额外的困扰是，有些输出功能并不会把“\”当作特殊字符看待，所以有些程式或网页就会错误地常常出现在“许功盖”这些字后面多了“\”。

### 私人造字

在[倚天中文系统](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=4257268" \t "_blank)，以及后来的Windows 3.1、95及98中，定义了四个私人造字区范围：0xFA40-0xFEFE、0x8E40-0xA0FE、0x8140-0x8DFE、0xC6A1-0xC8FE。私人造字区的原意，是供使用者加入本来在编码表中缺少的字符，但当每个使用者都在不同的地方加上不同的字符后，当交换资料时，对方便难以知道某一个编码究竟想表达什么字。

# UTF-8/16

## Utf8：

UTF-8（8-bit Unicode Transformation Format）是一种针对Unicode的可变长度字符编码，也是一种[前缀码](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=71130909&ss_c=ssc.citiao.link" \t "_blank)，又称万国码。由Ken Thompson于1992年创建。它可以用来表示Unicode标准中的任何字符，且其编码中的第一个字节仍与ASCII兼容，这使得原来处理ASCII字符的软件无须或只须做少部份修改，即可继续使用。因此，它逐渐成为电子邮件、网页及其他存储或传送文字的应用中，优先采用的编码。

优点：UTF-8编码可以通过屏蔽位和移位操作快速读写。字符串比较时strcmp()和wcscmp()的返回结果相同，因此使排序变得更加容易。字节FF和FE在UTF-8编码中永远不会出现，因此他们可以用来表明UTF-16或UTF-32文本（见BOM）UTF-8是[字节顺序](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=56845612&ss_c=ssc.citiao.link)无关的。它的字节顺序在所有系统中都是一样的，因此它实际上并不需要BOM。

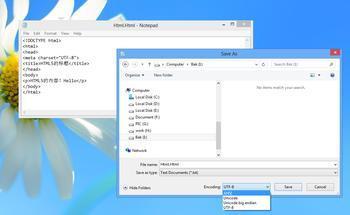
缺点：你无法从UNICODE字符数判断出UTF-8文本的字节数，因为UTF-8是一种变长编码它需要用2个字节编码那些用[扩展ASCII](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=11005095&ss_c=ssc.citiao.link)[字符集](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=291629&ss_c=ssc.citiao.link)只需1个字节的字符ISOLatin-1是UNICODE的子集，但不是UTF-8的子集8位字符的UTF-8编码会被email网关过滤，因为internet信息最初设计为7位[ASCII码](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=87696&ss_c=ssc.citiao.link)。因此产生了UTF-7编码。UTF-8在它的表示中使用值100xxxxx的几率超过50%，而现存的实现如ISO2022，4873，6429，和8859系统，会把它错认为是C1控制码。因此产生了UTF-7.5编码。

## Utf16：

 UTF-16是Unicode的其中一个使用方式。UTF是 Unicode Translation Format，即把Unicode转做某种格式的意思。它定义于ISO/IEC 10646-1的附录Q，而RFC2781也定义了相似的做法。在Unicode[基本多文种平面](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=10945487&ss_c=ssc.citiao.link)定义的字符（无论是[拉丁字母](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=151782&ss_c=ssc.citiao.link)、汉字或其他文字或符号），一律使用2字节储存。而在[辅助平面](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=6193877&ss_c=ssc.citiao.link)定义的字符，会以代理对（[surrogate](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=64476050&ss_c=ssc.citiao.link) pair）的形式，以两个2字节的值来储存。

有关模式：

UTF-16的大尾序和小尾序储存形式都在用。一般来说，以Macintosh制作或储存的文字使用大尾序格式，以Microsoft或Linux制作或储存的文字使用小尾序格式。

[](http://baike.sogou.com/PicBooklet.v?relateImageGroupIds=1417410&lemmaId=11006007&now=http://pic.baike.soso.com/p/20130810/20130810110626-897789517.jpg&type=1)null为了弄清楚UTF-16文件的大小尾序，在UTF-16文件的开首，都会放置一个U+FEFF字符作为Byte Order Mark（UTF-16LE以FF FE代表，UTF-16BE以FE FF代表），以显示这个文字档案是以UTF-16编码，其中U+FEFF字符在UNICODE中代表的意义是ZERO WIDTH NO-BREAK SPACE，顾名思义，它是个没有宽度也没有断字的空白。