软工11502班 朱超

目录

[Unicode 1](#_Toc492834875)

[起源 1](#_Toc492834876)

[作用 2](#_Toc492834877)

[UTF-8 3](#_Toc492834878)

[UTF-16 3](#_Toc492834879)

GB2312

# Unicode

Unicode（统一码、万国码、单一码）是计算机科学领域里的一项业界标准,包括字符集、编码方案等。Unicode 是为了解决传统的字符编码方案的局限而产生的，它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求。1990年开始研发，1994年正式公布。

# 起源

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）。

Unicode 编码包含了不同写法的字，如“ɑ/a”、“户/户/戸”。然而在汉字方面引起了一字多形的认定争议。

在文字处理方面，统一码为每一个字符而非字形定义唯一的代码（即一个整数）。换句话说，统一码以一种抽象的方式（即数字）来处理字符，并将视觉上的演绎工作（例如字体大小、外观形状、字体形态、文体等）留给其他软件来处理，例如网页浏览器或是文字处理器。

几乎所有电脑系统都支持基本拉丁字母，并各自支持不同的其他编码方式。Unicode为了和它们相互兼容，其首256字符保留给ISO 8859-1所定义的字符，使既有的西欧语系文字的转换不需特别考量；并且把大量相同的字符重复编到不同的字符码中去，使得旧有纷杂的编码方式得以和Unicode编码间互相直接转换，而不会丢失任何信息。举例来说，全角格式区段包含了主要的拉丁字母的全角格式，在中文、日文、以及韩文字形当中，这些字符以全角的方式来呈现，而不以常见的半角形式显示，这对竖排文字和等宽排列文字有重要作用。

在表示一个Unicode的字符时，通常会用“U+”然后紧接着一组十六进制的数字来表示这一个字符。在基本多文种平面（英文为 Basic Multilingual Plane，简写 BMP。它又简称为“零号平面”, plane 0）里的所有字符，要用四位十六进制数（例如U+4AE0，共支持六万多个字符）；在零号平面以外的字符则需要使用五位或六位十六进制数了。旧版的Unicode标准使用相近的标记方法，但却有些微的差异：在Unicode 3.0里使用“U-”然后紧接着八位数，而“U+”则必须随后紧接着四位数。

# 作用

能够使计算机实现跨语言、跨平台的文本转换及处理

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-8](https://baike.baidu.com/item/UTF-8)、[UTF-16](https://baike.baidu.com/item/UTF-16)、[UTF-32](https://baike.baidu.com/item/UTF-32)都是将数字转换到程序数据的编码方案。

[通用字符集](https://baike.baidu.com/item/%E9%80%9A%E7%94%A8%E5%AD%97%E7%AC%A6%E9%9B%86)（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。UCS-2用两个字节编码，UCS-4用4个字节编码。

历史上存在两个独立的尝试创立单一字符集的组织，即[国际标准化组织](https://baike.baidu.com/item/%E5%9B%BD%E9%99%85%E6%A0%87%E5%87%86%E5%8C%96%E7%BB%84%E7%BB%87)（ISO）和多语言软件制造商组成的统一码联盟。前者开发的 ISO/IEC 10646 项目，后者开发的[统一码](https://baike.baidu.com/item/%E7%BB%9F%E4%B8%80%E7%A0%81)项目。因此最初制定了不同的标准。

1991年前后，两个项目的参与者都认识到，世界不需要两个不兼容的字符集。于是，它们开始合并双方的工作成果，并为创立一个单一编码表而协同工作。从Unicode 2.0开始，Unicode采用了与ISO 10646-1相同的[字库](https://baike.baidu.com/item/%E5%AD%97%E5%BA%93)和字码；ISO也承诺，ISO 10646将不会替超出U+10FFFF的UCS-4编码赋值，以使得两者保持一致。两个项目仍都存在，并独立地公布各自的标准。但[统一码联盟](https://baike.baidu.com/item/%E7%BB%9F%E4%B8%80%E7%A0%81%E8%81%94%E7%9B%9F)和ISO/IEC JTC1/SC2都同意保持两者标准的码表兼容，并紧密地共同调整任何未来的扩展。在发布的时候，Unicode一般都会采用有关字码最常见的字型，但ISO 10646一般都尽可能采用Century字型。

UCS-4根据最高位为0的最高字节分成27=128个group。每个group再根据次高字节分为256个平面（plane）。每个平面根据第3个字节分为256行 （row），每行有256个码位（cell）。group 0的平面0被称作BMP（Basic Multilingual Plane）。如果UCS-4的前两个字节为全零，那么将UCS-4的BMP去掉前面的两个零字节就得到了UCS-2。每个平面有216=65536个码位。Unicode计划使用了17个平面，一共有17×65536=1114112个码位。在Unicode 5.0.0版本中，已定义的码位只有238605个，分布在平面0、平面1、平面2、平面14、平面15、平面16。其中平面15和平面16上只是定义了两个各占65534个码位的专用区（Private Use Area），分别是0xF0000-0xFFFFD和0x100000-0x10FFFD。所谓专用区，就是保留给大家放自定义字符的区域，可以简写为PUA。

平面0也有一个专用区：0xE000-0xF8FF，有6400个码位。平面0的0xD800-0xDFFF，共2048个码位，是一个被称作代理区（Surrogate）的特殊区域。代理区的目的用两个UTF-16字符表示BMP以外的字符。在介绍UTF-16编码时会介绍。

如前所述在Unicode 5.0.0版本中，238605-65534\*2-6400-2048=99089。余下的99089个已定义码位分布在平面0、平面1、平面2和平面14上，它们对应着Unicode定义的99089个字符，其中包括71226个汉字。平面0、平面1、平面2和平面14上分别定义了52080、3419、43253和337个字符。平面2的43253个字符都是汉字。平面0上定义了27973个汉字。

在Unicode中：汉字“字”对应的数字是23383（十进制），十六进制表示为5B57。在Unicode中，我们有很多方式将数字23383表示成程序中的数据，包括：[UTF-8](https://baike.baidu.com/item/UTF-8)、[UTF-16](https://baike.baidu.com/item/UTF-16)、[UTF-32](https://baike.baidu.com/item/UTF-32)。UTF是“Unicode Transformation Format”的缩写，可以翻译成Unicode字符集转换格式，即怎样将Unicode定义的数字转换成程序数据

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-8](https://baike.baidu.com/item/UTF-8)、[UTF-16](https://baike.baidu.com/item/UTF-16)、[UTF-32](https://baike.baidu.com/item/UTF-32)都是将数字转换到程序数据的编码方案。

# UTF-8

UTF-8的特点是对不同范围的字符使用不同长度的编码。对于0x00-0x7F之间的字符，UTF-8编码与[ASCII编码](https://baike.baidu.com/item/ASCII%E7%BC%96%E7%A0%81)完全相同。UTF-8编码的最大长度是6个字节。从上表可以看出，6字节模板有31个x，即可以容纳31位二进制数字。Unicode的最大码位0x7FFFFFFF也只有31位。

例1：“汉”字的Unicode编码是0x6C49。0x6C49在0x0800-0xFFFF之间，使用用3字节模板了：1110xxxx 10xxxxxx 10xxxxxx。将0x6C49写成二进制是：0110 1100 0100 1001， 用这个比特流依次代替模板中的x，得到：11100110 10110001 10001001，即E6 B1 89。

例2：Unicode编码0x20C30在0x010000-0x1FFFFF之间，使用用4字节模板了：11110xxx 10xxxxxx 10xxxxxx 10xxxxxx。将0x20C30写成21位二进制数字（不足21位就在前面补0）：0 0010 0000 1100 0011 0000，用这个比特流依次代替模板中的x，得到：11110000 10100000 10110000 10110000，即F0 A0 B0 B0。

# [UTF-16](https://baike.baidu.com/item/UTF-16)

UTF-16编码以16位无符号整数为单位。我们把Unicodeunicode

编码记作U。编码规则如下：

如果U<0x10000，U的UTF-16编码就是U对应的16位无符号整数（为书写简便，下文将16位无符号整数记作WORD）。

如果U≥0x10000，我们先计算U'=U-0x10000，然后将U'写成二进制形式：yyyy yyyy yyxx xxxx xxxx，U的UTF-16编码（二进制）就是：110110yyyyyyyyyy 110111xxxxxxxxxx。

为什么U'可以被写成20个二进制位？Unicode的最大码位是0x10ffff，减去0x10000后，U'的最大值是0xfffff，所以肯定可以用20个[二进制](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6)位表示。例如：Unicode编码0x20C30，减去0x10000后，得到0x10C30，写成二进制是：0001 0000 1100 0011 0000。用前10位依次替代模板中的y，用后10位依次替代模板中的x，就得到：1101100001000011 1101110000110000，即0xD843 0xDC30。

按照上述规则，Unicode编码0x10000-0x10FFFF的UTF-16编码有两个WORD，第一个WORD的高6位是110110，第二个WORD的高6位是110111。可见，第一个WORD的取值范围（二进制）是11011000 00000000到11011011 11111111，即0xD800-0xDBFF。第二个WORD的取值范围（二进制）是11011100 00000000到11011111 11111111，即0xDC00-0xDFFF。

为了将一个WORD的UTF-16编码与两个WORD的UTF-16编码区分开来，Unicode编码的设计者将0xD800-0xDFFF保留下来，并称为代理区（Surrogate）：

|  |  |  |
| --- | --- | --- |
| D800－DB7F | High Surrogates | 高位替代 |
| DB80－DBFF | High Private Use Surrogates | 高位专用替代 |
| DC00－DFFF | Low Surrogates | 低位替代 |

高位替代就是指这个范围的码位是两个WORD的UTF-16编码的第一个WORD。低位替代就是指这个范围的码位是两个WORD的UTF-16编码的第二个WORD。那么，高位专用替代是什么意思？我们来解答这个问题，顺便看看怎么由UTF-16编码推导Unicode编码。

如果一个字符的UTF-16编码的第一个WORD在0xDB80到0xDBFF之间，那么它的Unicode编码在什么范围内？我们知道第二个WORD的取值范围是0xDC00-0xDFFF，所以这个字符的UTF-16编码范围应该是0xDB80 0xDC00到0xDBFF 0xDFFF。我们将这个范围写成二进制：

1101101110000000 11011100 00000000 - 1101101111111111 1101111111111111

按照编码的相反步骤，取出高低WORD的后10位，并拼在一起，得到

1110 0000 0000 0000 0000 - 1111 1111 1111 1111 1111

即0xe0000-0xfffff，按照编码的相反步骤再加上0x10000，得到0xf0000-0x10ffff。这就是UTF-16编码的第一个WORD在0xdb80到0xdbff之间的Unicode编码范围，即平面15和平面16。因为Unicode标准将平面15和平面16都作为专用区，所以0xDB80到0xDBFF之间的保留码位被称作高位专用替代[1]

Unicode 已经有6.2版本。世界上有一大批计算机、语言学等科学家专门研究Unicode，Unicode标准已经不单是一个编码标准，还是记录人类语言文字资料的一个巨大的数据库，同时从事人类文化遗产的发掘和保护工作。

对于中文而言，Unicode 16编码里面已经包含了[GB18030](https://baike.baidu.com/item/GB18030)里面的所有汉字（27484个字），Unicode标准准备把[康熙字典](https://baike.baidu.com/item/%E5%BA%B7%E7%86%99%E5%AD%97%E5%85%B8)的所有汉字放入到Unicode 32bit编码中。

简单地说，Unicode扩展自ASCII字元集。在严格的ASCII中，每个字元用7位元表示，或者电脑上普遍使用的每字元有8位元宽；而Unicode使用全16位元字元集。这使得Unicode能够表示世界上所有的书写语言中可能用於电脑通讯的字元、象形文字和其他符号。Unicode最初打算作为ASCII的补充，可能的话，最终将代替它。考虑到ASCII是电脑中最具支配地位的标准，所以这的确是一个很高的目标。

Unicode影响到了电脑工业的每个部分，但也许会对作业系统和[程序设计语言](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%E8%AF%AD%E8%A8%80)的影响最大。从这方面来看，我们已经上路了。Windows NT从底层支持Unicode（不幸的是，Windows 98只是小部分支援Unicode）。先天即被ANSI束缚的[C程序设计语言](https://baike.baidu.com/item/C%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%E8%AF%AD%E8%A8%80)通过对宽字元集的支持来支持Unicode。

微软拼音 在输入法启动状态下，单击语言栏上的“功能菜单”按钮，指向“辅助输入法”即可发现“Unicode码输入方式”，利用它可以直接输入Unicode相应十六进制值的方式输入相应文字。例如中文“胥”输入“5066”，朝鲜语文字“셅”输入“c145”（不需要在前面加0x或x）。海峰五笔 此输入法已经直接支持透过五笔码输入方式输入Unicode内的任意中日韩汉字，但无法使用键入Unicode码的方式输入。例如汉字（Unicode部分）“㗎”为“keks”，CJK扩展B区的“㿱”为“iyho”和CJK扩展C区的“뇛”为“muih”。新注音输入法 在输入法启动状态时，打入键盘上的“多功能前导字符键”（及通用键盘上之“`”），第一次使用会弹出说明。输入Unicode字符“胥”则是在键盘上键入“`U5066”。而韩语中的“셅”，则输入“`UC145”。而要输入日语自制汉字“卡”，则是“`U5CE0”。VimIM 在 Vim 环境中，可以直接键入十进制或十六进制 Unicode 码。既不需要启动输入法，也不需要码表。

除了输入法外，操作系统也会提供另外几种方法输入 Unicode。像是Windows 2000之后的 Windows 系统就提供一个可点击的字符映射表。又或者在Microsoft Word下，按下 Alt 键不放，输入 0 和某个字符的 Unicode 编码（十进制），再松开 Alt 键即可得到该字符，如Alt + 033865会得到 Unicode 字符*叶*。另外按Alt + X 组合键，MS Word 也会将光标前面的字符，同其十六进制的四位Unicode 编码进行互相转换。

基本上，计算机只是处理数字。它们指定一个数字，来储存字母或其他字符。在创造Unicode之前，有数百种指定这些数字的编码系统。没有一个编码可以包含足够的字符：例如，单单欧洲共同体就需要好几种不同的编码来包括所有的语言。即使是单一种语言，例如英语，也没有哪一个编码可以适用于所有的字母，标点符号，和常用的技术符号。这些编码系统也会互相冲突。也就是说，两种编码可能使用相同的数字代表两个不同的字符，或使用不同的数字代表相同的字符。任何一台特定的计算机（特别是服务器）都需要支持许多不同的编码，但是，不论什么时候数据通过不同的编码或平台之间，那些数据总会有损坏的危险。

为什么使用Unicode其实原因很简单，因为Unicode比ANSI好用。 自从Windows2K开始，Win的系统内核开始完全支持并完全应用Unicode编写，所有ANSI字符在进入底层前，都会被相应的API转换成Unicode。所以，如果你一开始就使用Unicode，则可以减少转换的用时和RAM开销。 对于JAVA/.NET等这些“新”的语言来说，内置的字符串所使用的字符集已经完全是Unicode。最重要的是，世界上大多数程序用的字符集都是Unicode，因为Unicode有利于程序国际化和标准化。

GB2312[[1]](#footnote-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。

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。

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。

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区的区号加上0xA0)，“低位字节”使用了0xA1-0xFE(把01-94加上0xA0)。

1. 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。

1. [↑](#footnote-ref-1)