字符编码

维基百科，自由的百科全书

字符编码（英语：Character encoding）、**字集碼**是把**字符集**中的字符编码为指定集合中某一对象（例如：比特模式、自然数序列、8位元组或者电脉冲），以便文本在计算机中存储和通过通信网络的传递。常见的例子包括将拉丁字母表编码成摩斯电码和ASCII。其中，ASCII将字母、数字和其它符号編號，並用7位元的二进制來表示这个整数。通常會額外使用一个扩充的位元，以便于以8个字节的方式存储。

在计算机技术发展的早期，如ASCII（1963年）和EBCDIC（1964年）这样的**字符集**逐漸成為標準。但这些字符集的局限很快就变得明显，于是人们开发了許多方法来扩展它们。对于支持包括东亚CJK字符家族在内的写作系统的要求能支持更大量的字符，并且需要一种系统而不是临时的方法实现这些字符的编码。

目录

简单字符集

按照惯例，人们认为字符集和字符编码是同义词，因为使用同样的标准来定义提供什么字符并且这些字符如何编码到一系列的代码单元（通常一个字符一个单元）。由于历史的原因，MIME和使用这种编码的系统使用术语**字符集**来表示用于将一组字符编码成一系列八位字节数据的整个系统。

现代编码模型

由统一碼和通用字符集所構成的现代字符编码模型则没有跟从简单字符集的观点。它们将字符编码的概念分为：有哪些字符、它们的编号、这些编号如何编码成一系列的“码元”（有限大小的数字）以及最后这些单元如何組成八位字节流。區分這些概念的核心思想是建立一个能够用不同方法來编码的一个通用字符集。为了正确地表示这个模型需要更多比“字符集”和“字符编码”更为精确的术语表示。在Unicode Technical Report (UTR) #17中，现代编码模型分为5个层次，所用的术语列在下面：

1. **抽象字符表**（Abstract character repertoire）是一个系统支持的所有抽象字符的集合。字符表可以是封闭的，即除非创建一个新的标准（ASCII和多数ISO/IEC 8859系列都是这样的例子），否则不允许添加新的符号；字符表也可以是开放的，即允许添加新的符号（统一碼和一定程度上代碼頁是这方面的例子）。特定字符表中的字符反映了如何将书写系统分解成线性信息单元的决定。例如拉丁、希腊和斯拉夫字母表分为字母、数字、变音符号、标点和如空格这样的一些少数特殊字符，它们都能按照一种简单的线性序列排列（尽管对它们的处理需要另外的规则，如带有变音符号的字母这样的特定序列如何解释——但这不属于字符表的范畴）。为了方便起见，这样的字符表可以包括预编号的字母和变音符号的组合。其它的书写系统，如阿拉伯语和希伯莱语，由于要适应双向文字和在不同情形下按照不同方式交叉在一起的字形，就使用更为复杂的符号表表示。
2. **编码字符集**（CCS:Coded Character Set）是将字符集*C*中每个字符映射到1个坐标（整数值对：x, y）或者表示为1个非负整数*N*。字符集及码位映射称为编码字符集。例如，在一个给定的字符表中，表示大写拉丁字母“A”的字符被赋予整数65、字符“B”是66，如此继续下去。多个编码字符集可以表示同样的字符表，例如ISO-8859-1和IBM的代码页037和代码页500含蓋同样的字符表但是将字符映射为不同的整数。由此产生了**编码空间**（encoding space）的概念：简单说就是包含所有字符的表的维度。可以用一对整数来描述，例如 GB 2312的汉字编码空间是94 x 94。可以用一个整数来描述，例如：ISO-8859-1的编码空间是256。也可以用字符的存储单元尺寸来描述，例如：ISO-8859-1是一个8比特的编码空间。编码空间还可以用其子集来表述，如行、列、面（plane）等。编码空间中的一个位置（position）称为**码位**（code point）。一个字符所占用的码位称为**码位值**（code point value）。1个编码字符集就是把抽象字符映射为码位值。
3. **字符编码表**（CEF:Character Encoding Form），也称为"storage format"，是将编码字符集的非负整数值（即抽象的码位）转换成有限比特长度的整型值（称为**码元**code units）的序列。这对于定长编码来说是个到自身的映射（null mapping），但对于变长编码来说，该映射比较复杂，把一些码位映射到一个码元，把另外一些码位映射到由多个码元组成的序列。例如，使用16比特长的存储单元保存数字信息，系统每个单元只能够直接表示从0到65,535的数值，但是如果使用多个16位单元就能够表示更大的整数。这就是CEF的作用，它可以把Unicode从0到140万的码空间范围的每个码位映射到单个或多个在0到65,5356范围内的码值。最简单的字符编码表就是單純地选择足够大的单位，以保证编码字符集中的所有数值能够直接编码（一个码位对应一个码值）。这对于能够用使用八位元组來表示的编码字符集（如多数传统的非CJK的字符集编码）是合理的，对于能够使用十六位元來表示的编码字符集（如早期版本的Unicode）来说也足够合理。但是，随着编码字符集的大小增加（例如，现在的Unicode的字符集至少需要21位才能全部表示），这种直接表示法变得越来越没有效率，并且很难让现有计算机系统适应更大的码值。因此，许多使用新近版本Unicode的系统，或者将Unicode码位對應為可变长度的8位字节序列的UTF-8，或者将码位對應为可变长度的16位序列的UTF-16。
4. **字符编码方案**（CES:Character Encoding Scheme），也称作"serialization format"。將定长的整型值（即码元）映射到8位字节序列，以便编码后的数据的文件存储或网络传输。在使用Unicode的场合，使用一个简单的字符来指定字节顺序是大端序或者小端序（但对于UTF-8来说并不需要专门指明字节序）。然而，有些复杂的字符编码机制（如ISO/IEC 2022）使用控制字符转义序列在几种编码字符集或者用于减小每个单元所用字节数的压缩机制（如SCSU、BOCU和Punycode）之间切换。
5. 传输编码语法（transfer encoding syntax），用于处理上一层次的字符编码方案提供的字节序列。一般其功能包括两种：一是把字节序列的值映射到一套更受限制的值域內，以满足传输环境的限制，例如Email传输时Base64或者quoted-printable，都是把8位的字节编码为7位长的数据；另一是压缩字节序列的值，如LZW或者行程长度编码等无损压缩技术。

高层机制（higher level protocol）提供了额外信息，用于选择Unicode字符的特定变种，如XML属性xml:lang

字符映射（character map）在Unicode中保持了其传统意义：从字符序列到编码后的字节序列的映射，包括了上述的CCS, CEF, CES层次。

字符集、代码页，与字符映射

术语字符编码（character encoding），字符映射（character map），字符集（character set）或者代码页，在历史上往往是同义概念，即字符表（repertoire）中的字符如何编码为码元的流（stream of code units）–通常每个字符对应单个码元。

码元（Code Unit，也称「代码单元」）是指一个已编码的文本中具有最短的比特组合的单元。对于UTF-8来说，码元是8比特长；对于UTF-16来说，码元是16比特长；对于UTF-32来说，码元是32比特长[1]。码值（Code Value）是过时的用法。

代码页通常意味着面向字节的编码，但强调是一套用于不能语言的编码方案的集合.著名的如"Windows"代码页系列，"IBM"/"DOS"代码页系列.

IBM的字符数据表示体系（Character Data Representation Architecture - CDRA）与编码字符集标识符（coded character set identifiers - CCSIDs）常常把charset, character set, code page, or CHARMA等类似意义的术语混用.

Unix或Linux不使用代码页概念，它们用charmap，比locales具有更广泛的含义.

与上文的编码字符集（Coded Character Set - CCS）不同，字符编码（character encoding）是从抽象字符到代码字（code word）的映射. HTTP（与MIME）的用法中，字符集（character set）与字符编码同义，但与CCS不是一个意思.

字符编码（不全）

- ASCII
- EBCDIC

西欧标准

- ISO-8859-1
- ISO-8859-5
- ISO-8859-6
- ISO-8859-7
- ISO-8859-11
- ISO-8859-15
- ISO/IEC 646

DOS字符集（又称IBM代码页）

- CP437
- CP737
- CP850
- CP852
- CP855
- CP857
- CP858
- CP860
- CP861
- CP863
- CP865
- CP866
- CP869

Windows字符集

- Windows-1250
- Windows-1251用于西里尔字母表
- Windows-1252
- Windows-1253
- Windows-1254
- Windows-1255用于希伯莱语

- Windows-1256用于阿拉伯语
- Windows-1257
- Windows-1258用于越南语

亚洲字符集

中國大陸

- GB 2312
- EUC
- GBK（规定文件为GB13000）
- GB 18030

港澳臺

- 大五碼
- 香港增補字符集
- 中文資訊交換碼（CCCII）
- 中文標準交換碼（CNS 11643）
- EUC

日本

- ISO/IEC 2022
- Shift JIS
- EUC

朝鮮半岛

- EUC
- KOI8-R
- KOI8-U
- KOI7
- MIK

越南

- 越南資訊交換標準代碼

印度

- 印度文字資訊交換碼

Unicode

- Unicode
- UTF-7
- UTF-8
- UTF-16
- UTF-32

字符转换工具

由于有很多种字符编码方法被使用，从一种字符编码转换到另一种，需要一些工具。

跨平台：

- 网页浏览器–大多数现代的网页浏览器都具有此功能。一般是在菜单"查看"（View）/"字符编码"（Character Encoding）
- iconv –程序与编程API，用于字符编码转换
- convert_encoding.py –基于Python的转换工具.[2]
- decodeh.py –用于启发性猜测编码方案的算法与模块[3]
- International Components for Unicode–一套C语言与Java语言的开源库，由IBM提供，用于Unicode等多语言编码的转换、实现.
- chardet – Mozilla的编码自动检测代码的Python语言实现.
- 新版本的Unix命令File做字符编码的检测.（cygwin与mac都有此命令）

Linux:

- recode – [4]
- utrac – 将整个文件内容从一种字符编码转换到另外一种[5]
- cstocs –
- convmv –转换文件名.[6]
- enca –分析编码模式.[7]

Microsoft Windows:

- Encoding.Convert – .NET API.[8]
- MultiByteToWideChar/WideCharToMultiByte – Windows API.[9]
- cscvt –转换工具[10]
- enca –分析编码方法[11]

参见

- Category:字符编码—关于通用字符编码的文章
- Category:字符集—关于特殊字符编码的文章
- 亂碼—非映射字符集
- 字符集
- 字形
- 位圖
- 像素
- 體素
- 中文軟體
- 中文系統

参考文献

1. Glossary of Unicode Terms (http://unicode.org/glossary/)
2. Homepage of Michael Goerz – convert_encoding.py (http://users.physik.fu-berlin.de/~mgoerz/blog/programs/convert_encoding/)
3. Decodeh – heuristically decode a string or text file (http://gizmojo.org/code/decodeh/)互联网档案馆的存檔 (https://web.archive.org/web/20080108123255/http://gizmojo.org/code/decodeh/), 存档日期2008-01-08.
4. Recode – GNU Project – Free Software Foundation (FSF) (http://www.gnu.org/software/recode/recode.html)

5. Utrac Homepage (http://utrac.sourceforge.net/)
6. Convmv – converts filenames from one encoding to another (http://www.j3e.de/linux/convmv/man/)
7. Extremely Naive Charset Analyser (http://directory.fsf.org/project/enca/)
8. Microsoft .NET Framework Class Library – Encoding.Convert Method (http://msdn.microsoft.com/en-us/library/system.text.encoding.convert(VS.71).aspx)
9. MultiByteToWideChar/WideCharToMultiByte – Convert from ANSI to Unicode & Unicode to ANSI (http://support.microsoft.com/kb/138813)
10. Character Set Converter (http://www.kalytta.com/tools.php)
11. Extremely Naive Charset Analyser (http://www.john.geek.nz/2010/02/enca-binary-compiled-for-32-bit-windows/)

外部链接

- Character sets registered by Internet Assigned Numbers Authority
- Unicode Technical Report #17: Character Encoding Model
- SIL's freeware fonts, editors and documentation. See SIL
- ICU Converter Explorer
- The Cyrillic Charset soup
- Early history of character set standardization
- Character Sets And Code Pages At The Push Of A Button
- A complete introduction to Japanese character encodings
- A tutorial on character code issues
- Online Char (ASCII), HEX, Binary, Base64, etc... Encoder/Decoder with MD2, MD4, MD5, SHA1+2, etc. hashing algorithms
- Universal Cyrillic decoder，一个用来帮助恢复由于错误字符编码产生的不可读的西里尔字母的在线程序（以及其它的一些程序）.
- Introduction to i18n，请参阅Chapter 3 - Important Concepts for Character Coding Systems
- 汉字字符编码查询
- 精确解释Unicode