论文

2015届

编码集的研究

学生姓名：**韩虎伟**

学号：**21503617**

院系：**计算机科学学院**

专业：**软件工程**

指导教师：**向华**

完成日期：**2017年9月10日**

**目录**

前言...........................................1

1. Unicode编码集...............................2
   1. unicode编码集来历.....................2
   2. unicode编码集产生原因.................2
   3. unicode编码集解决问题.................2
   4. unicode编码集字节数...................3
2. Gb2312编码集................................3

2.1、Gb2312编码集来历..........................3

2.2、Gb2312编码集产生原因......................3

2.3、Gb2312编码集解决问题......................3

2.4、Gb2312编码集字节数........................4

3、Big5编码集..................................4

3.1、Big5编码集来历............................4

3.2、Big5编码集产生原因........................4

3.3、Big5编码集解决问题........................4

3.4、Big5编码集字节数..........................5

4、UTF 8/16编码集..............................5

4.1、UTF 8/16编码集来历........................5

4.2、UTF 8/16编码集产生原因....................5

4.3、UTF 8/16编码集解决问题....................6

4.4、UTF 8/16编码集字节数......................6

前言

现实生活中，不同的语系有自己的字符集合，例如藏文有自己的字符集合，汉文有自己的字符集合。到计算机的世界中，也有各种字符集合，例如[ASCII字符集合](http://zh.wikipedia.org/zh/ASCII)，[GB2312字符集合](http://zh.wikipedia.org/zh/GB_2312)，[GBK字符集合](http://zh.wikipedia.org/zh/%E6%B1%89%E5%AD%97%E5%86%85%E7%A0%81%E6%89%A9%E5%B1%95%E8%A7%84%E8%8C%83)。还有一个其他字符集合的超集--[Unicode字符集](http://zh.wikipedia.org/wiki/Unicode)定义了几乎绝大部分现存语言需要的字符，是一种通用的字符集，来支持多语言环境(可以同时处理多种语言混合的情况）。各个国家和地区在制定编码标准的时候，“字符集合”和“字符编码”一般都是同时制定的。所以像ASCII字符集合一样，它也同时代表了一种字符的编码。

1. unicode编码集
   1. unicode编码集的来历

20世纪80年代末，组成Unicode组织的商业机构，和国际合作的[国际标准化组织](https://zh.wikipedia.org/wiki/%E5%9C%8B%E9%9A%9B%E6%A8%99%E6%BA%96%E5%8C%96%E7%B5%84%E7%B9%94" \o "国际标准化组织)因为电脑普及和信息国际化的前提下，分别各自成立了Unicode组织[[3]](https://zh.wikipedia.org/wiki/Unicode#cite_note-3)和ISO-10646工作小组。他们不久便发现对方的存在，大家为着相同的目的而工作。1991年，Unicode Consortium与ISO/IEC JTC1/SC2同意保持Unicode码表与ISO 10646标准保持兼容并密切协调各自标准进一步的扩展。虽然实际上两者的字集编码相同，但实质上两者确实为两个不同的标准。Unicode 1.1对应于ISO 10646-1:1993，Unicode 3.0对应于ISO 10646-1:2000，Unicode 3.2对应于ISO 10646-2:2001，Unicode 4.0对应于ISO 10646:2003，Unicode 5.0对应于ISO 10646:2003及附录1–3。

Unicode自版本2.0开始保持了向后兼容，即新的版本仅仅增加字符，原有字符不会被删除或更名。

统一码联盟在1991年首次发布了**The Unicode Standard**。Unicode的开发结合了[国际标准化组织](https://zh.wikipedia.org/wiki/%E5%9C%8B%E9%9A%9B%E6%A8%99%E6%BA%96%E5%8C%96%E7%B5%84%E7%B9%94" \o "国际标准化组织)所制定的[ISO/IEC 10646](https://zh.wikipedia.org/wiki/ISO/IEC_10646)，即[通用字符集](https://zh.wikipedia.org/w/index.php?title=%E9%80%9A%E7%94%A8%E5%AD%97%E5%85%83%E9%9B%86&action=edit&redlink=1" \o "通用字符集（页面不存在）)。Unicode与ISO/IEC 10646在编码的运作原理相同，但**The Unicode Standard**包含了更详尽的实现信息、涵盖了更细节的主题，诸如比特编码（bitwise encoding）、校对以及呈现等。**The Unicode Standard**也枚举了诸多的字符特性，包含了那些必须支持两种阅读方向的文字（由左至右或由右至左的文字阅读方向，例如阿拉伯文是由右至左）。Unicode与ISO/IEC 10646这两个标准在术语上的使用有些微的不同。

在2005年，Unicode的第十万个字符被引入成为标准之一，该字符被用于[马拉雅拉姆语](https://zh.wikipedia.org/wiki/%E9%A6%AC%E6%8B%89%E9%9B%85%E6%8B%89%E5%A7%86%E8%AA%9E" \o "马拉雅拉姆语)。(来源维基百科)

* 1. unicode编码集产生原因

Unicode是为了解决传统的[字符编码](https://zh.wikipedia.org/wiki/%E5%AD%97%E5%85%83%E7%B7%A8%E7%A2%BC)方案的局限而产生的，例如[ISO 8859-1](https://zh.wikipedia.org/wiki/ISO/IEC_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](https://zh.wikipedia.org/wiki/%E6%8B%89%E4%B8%81%E5%AD%97%E6%AF%8D)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）。(来源维基百科)

* 1. unicode编码集解决问题

Unicode是为了解决传统的[字符编码](https://zh.wikipedia.org/wiki/%E5%AD%97%E5%85%83%E7%B7%A8%E7%A2%BC" \o "字符编码)方案的局限而产生的，例如[ISO 8859-1](https://zh.wikipedia.org/wiki/ISO/IEC_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](https://zh.wikipedia.org/wiki/%E6%8B%89%E4%B8%81%E5%AD%97%E6%AF%8D" \o "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”、“強／强”、“戶／户／戸”。然而在[汉字](https://zh.wikipedia.org/wiki/%E6%B1%89%E5%AD%97" \o "汉字)方面引起了一字多形的认定争议（详见[中日韩统一表意文字](https://zh.wikipedia.org/wiki/%E4%B8%AD%E6%97%A5%E9%9F%93%E7%B5%B1%E4%B8%80%E8%A1%A8%E6%84%8F%E6%96%87%E5%AD%97" \o "中日韩统一表意文字)主题）。

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。

目前，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](https://zh.wikipedia.org/wiki/%E5%85%A8%E5%BD%A2" \o "全角)格式区块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。(来源维基百科)

* 1. unicode字节数

unicode一到四个字节表示一个字符。(来源百度百科)

1. GB2312编码集

2.1、GB2312编码集的来历

《信息交换用汉字编码字符集》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[GB 2312](https://baike.baidu.com/item/GB%202312)—1980。(来源维基百科)

2.2、GB2312编码集产生原因

信息交换用汉字编码字符集和[汉字输入](https://baike.baidu.com/item/%E6%B1%89%E5%AD%97%E8%BE%93%E5%85%A5)编码之间的关系是，根据不同的汉字输入方法，通过必要的设备向计算机输入汉字的编码，计算机接收之后，先转换成信息交换用汉字编码字符，这时计算机就可以识别并进行处理；汉字输出是先把机内码转成汉字编码，再发送到[输出设备](https://baike.baidu.com/item/%E8%BE%93%E5%87%BA%E8%AE%BE%E5%A4%87)。(来源维基百科)

2.3、GB2312编码集解决问题

它适用于一般汉字处理、汉字通信等系统之间的信息交换。随着我国汉字信息处理技术的发展,计算机的应用范围不断扩大,使用汉字字数较多的部门迫切需要在《基本集》的基础上继续制定信息交换用汉字编码字符集各辅助集的国家标准。(来源维基百科)

2.4、GB2312编码集字节数

在使用GB 2312的程序通常采用[EUC](https://zh.wikipedia.org/wiki/EUC" \o "EUC)储存方法，以便兼容于[ASCII](https://zh.wikipedia.org/wiki/ASCII" \o "ASCII)。这种格式称为**EUC-CN**。[浏览器](https://zh.wikipedia.org/wiki/%E7%BD%91%E9%A1%B5%E6%B5%8F%E8%A7%88%E5%99%A8" \o "网页浏览器)编码表上的“GB2312”就是指这种表示法。

每个汉字及符号以两个[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82)来表示。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。（与区位码对比：0xB0=0xA0+16,0xA1=0xA0+1）。(来源维基百科)

1. Big5编码集

3.1、Big5编码集来历

“大五码”（Big5）是由台湾[财团法人信息产业策进会](https://zh.wikipedia.org/wiki/%E8%B3%87%E8%A8%8A%E5%B7%A5%E6%A5%AD%E7%AD%96%E9%80%B2%E6%9C%83)为[五大中文套装软件](https://zh.wikipedia.org/wiki/%E4%BA%94%E5%A4%A7%E4%B8%AD%E6%96%87%E5%A5%97%E8%A3%9D%E8%BB%9F%E9%AB%94)所设计的中文共通内码，在1983年12月完成公告[[2]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-2)[[3]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-3)，隔年3月，信息产业策进会与[台湾](https://zh.wikipedia.org/wiki/%E8%87%BA%E7%81%A3)13家厂商签定“16位个人电脑套装软件合作开发（BIG-5）项目（[五大中文套装软件](https://zh.wikipedia.org/wiki/%E4%BA%94%E5%A4%A7%E4%B8%AD%E6%96%87%E5%A5%97%E8%A3%9D%E8%BB%9F%E9%AB%94)）”[[4]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-4)，因为此中文内码是为台湾自行制作开发之“五大中文套装软件”所设计的，所以就称为Big5中文内码[[5]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-5)[[6]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-6)[[7]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-7)[[8]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-8)。五大中文套装软件虽然并没有如预期的取代国外的套装软件，但随着采用Big5码的[国乔中文系统](https://zh.wikipedia.org/w/index.php?title=%E5%9C%8B%E5%96%AC%E4%B8%AD%E6%96%87%E7%B3%BB%E7%B5%B1&action=edit&redlink=1)及[倚天中文系统](https://zh.wikipedia.org/wiki/%E5%80%9A%E5%A4%A9%E4%B8%AD%E6%96%87%E7%B3%BB%E7%B5%B1)先后在台湾市场获得成功，使得Big5码深远地影响繁体中文电脑[内码](https://zh.wikipedia.org/wiki/%E5%85%A7%E7%A2%BC)，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”两个中文名称。(来源维基百科)

3.2、Big5编码集产生原因

Big5码的产生，是因为当时[个人电脑](https://zh.wikipedia.org/wiki/%E5%80%8B%E4%BA%BA%E9%9B%BB%E8%85%A6)没有共通的内码，导致厂商推出的中文应用软件无法推广，并且与[IBM 5550](https://zh.wikipedia.org/wiki/IBM_5550)、[王安码](https://zh.wikipedia.org/w/index.php?title=%E7%8E%8B%E5%AE%89%E7%A2%BC&action=edit&redlink=1)等内码，彼此不能兼容；另一方面，台湾当时尚未推出中文编码标准。在这样的时空背景下，为了使台湾早日进入信息时代，所采行的一个项目；同时，这个项目对于以台湾为核心的亚洲繁体汉字圈也产生了久远的影响。(来源维基百科)

3.3、Big5编码集解决问题

在Big5码诞生后，大部分台湾的电脑软件都使用了Big5码，加上后来[倚天中文系统](https://zh.wikipedia.org/wiki/%E5%80%9A%E5%A4%A9%E4%B8%AD%E6%96%87%E7%B3%BB%E7%B5%B1)的高度普及，使后来的[微软](https://zh.wikipedia.org/wiki/%E5%BE%AE%E8%BD%AF)[Windows 3.x](https://zh.wikipedia.org/wiki/Windows_3.x)等亦予以采用。虽然后来台湾还有各种想要取代Big5码，像是倚天中文系统所推行的[倚天码](https://zh.wikipedia.org/wiki/%E5%80%9A%E5%A4%A9%E7%A2%BC)、台北市电脑公会所推动的[公会码](https://zh.wikipedia.org/w/index.php?title=%E5%85%AC%E6%9C%83%E7%A2%BC&action=edit&redlink=1)等，但是由于Big5字码已沿用多年，因此在习惯不易改变的情况下，始终无法成为主流字码。而台湾后来发展的[国家标准CNS 11643中文标准交换码](https://zh.wikipedia.org/wiki/CNS11643)由于非一般的内码系统，是以交换使用为目的，受先天所限，必须使用至少三个字节来表示一个汉字，所以普及率远远不及Big5码。(来源维基百科)

3.4、Big5编码集字节数

Big5码是一套[双字节字符集](https://zh.wikipedia.org/wiki/%E5%8F%8C%E5%AD%97%E8%8A%82%E5%AD%97%E7%AC%A6%E9%9B%86" \o "双字节字符集)，使用了双八码存储方法，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。(来源维基百科)

1. UTF 8/16编码集

4.1、UTF 8/16编码集来历

1992年初，为创建良好的[字节串编码系统](https://zh.wikipedia.org/w/index.php?title=%E4%BD%8D%E5%85%83%E7%B5%84%E4%B8%B2%E7%B7%A8%E7%A2%BC%E7%B3%BB%E7%B5%B1&action=edit&redlink=1" \o "字节串编码系统（页面不存在）)以供多[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82" \o "字节)字符集使用，开始了一个正式的研究。[ISO/IEC 10646](https://zh.wikipedia.org/wiki/%E9%80%9A%E7%94%A8%E5%AD%97%E7%AC%A6%E9%9B%86)的初稿中有一个非必须的附录，名为UTF。当中包含了一个供32[比特](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)的[字符](https://zh.wikipedia.org/wiki/UCS)使用的字节串编码系统。这个编码方式的性能并不令人满意，但它提出了将0-127的范围保留给ASCII以兼容旧系统的概念。

1992年7月，[X/Open](https://zh.wikipedia.org/wiki/X/Open" \o "X/Open)委员会XoJIG开始寻求一个较佳的编码系统。[Unix系统实验室](https://zh.wikipedia.org/wiki/Unix%E7%B3%BB%E7%BB%9F%E5%AE%9E%E9%AA%8C%E5%AE%A4)（USL）的Dave Prosser为此提出了一个编码系统的建议。它具备可更快速实现的特性，并引入一项新的改进。其中，7[比特](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)的[ASCII](https://zh.wikipedia.org/wiki/ASCII)符号只代表原来的意思，所有多字节序列则会包含第8[比特](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)的符号，也就是所谓的[最高有效比特](https://zh.wikipedia.org/wiki/%E6%9C%80%E9%AB%98%E6%9C%89%E6%95%88%E4%BD%8D%E5%85%83)。

1992年8月，这个建议由[IBM](https://zh.wikipedia.org/wiki/IBM)[X/Open](https://zh.wikipedia.org/wiki/X/Open)的代表流传到一些感兴趣的团体。与此同时，[贝尔实验室](https://zh.wikipedia.org/wiki/%E8%B2%9D%E7%88%BE%E5%AF%A6%E9%A9%97%E5%AE%A4)[九号项目](https://zh.wikipedia.org/wiki/%E8%B2%9D%E7%88%BE%E5%AF%A6%E9%A9%97%E5%AE%A4%E4%B9%9D%E8%99%9F%E8%A8%88%E7%95%AB)[操作系统](https://zh.wikipedia.org/wiki/%E4%BD%9C%E6%A5%AD%E7%B3%BB%E7%B5%B1)工作小组的[肯·汤普逊](https://zh.wikipedia.org/wiki/%E8%82%AF%C2%B7%E6%B1%A4%E6%99%AE%E9%80%8A)对这编码系统作出重大的修改，让编码可以自我同步，使得不必从字符串的开首读取，也能找出字符间的分界。1992年9月2日，[肯·汤普逊](https://zh.wikipedia.org/wiki/%E8%82%AF%C2%B7%E6%B1%A4%E6%99%AE%E9%80%8A)和[罗勃·派克](https://zh.wikipedia.org/wiki/%E7%BE%85%E5%8B%83%C2%B7%E6%B4%BE%E5%85%8B)一起在[美国](https://zh.wikipedia.org/wiki/%E7%BE%8E%E5%9C%8B)[新泽西州](https://zh.wikipedia.org/wiki/%E6%96%B0%E6%BE%A4%E8%A5%BF%E5%B7%9E)一架餐车的餐桌垫上描绘出此设计的要点。接下来的日子，Pike及汤普逊将它实现，并将这编码系统完全应用在[九号项目](https://zh.wikipedia.org/wiki/%E8%B2%9D%E7%88%BE%E5%AF%A6%E9%A9%97%E5%AE%A4%E4%B9%9D%E8%99%9F%E8%A8%88%E7%95%AB)当中，及后他将有关成果回馈X/Open。

1993年1月25-29日的在[圣地牙哥](https://zh.wikipedia.org/wiki/%E8%81%96%E5%9C%B0%E7%89%99%E5%93%A5)举行的[USENIX](https://zh.wikipedia.org/wiki/USENIX)会议首次正式介绍UTF-8。

自1996年起，[微软](https://zh.wikipedia.org/wiki/%E5%BE%AE%E8%BB%9F)的[CAB](https://zh.wikipedia.org/wiki/CAB)（MS Cabinet）规格在UTF-8标准正式落实前就明确容许在任何地方使用UTF-8编码系统。但有关的编码器实际上从来没有实现这方面的规格。(来源维基百科)

4.2、UTF 8/16编码集产生原因

1992年初，为创建良好的[字节串编码系统](https://zh.wikipedia.org/w/index.php?title=%E4%BD%8D%E5%85%83%E7%B5%84%E4%B8%B2%E7%B7%A8%E7%A2%BC%E7%B3%BB%E7%B5%B1&action=edit&redlink=1)以供多[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82)字符集使用，开始了一个正式的研究。[ISO/IEC 10646](https://zh.wikipedia.org/wiki/%E9%80%9A%E7%94%A8%E5%AD%97%E7%AC%A6%E9%9B%86)的初稿中有一个非必须的附录，名为UTF。当中包含了一个供32[比特](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)的[字符](https://zh.wikipedia.org/wiki/UCS)使用的字节串编码系统。这个编码方式的性能并不令人满意，但它提出了将0-127的范围保留给ASCII以兼容旧系统的概念。(来源维基百科)

4.3、UTF 8/16编码集解决问题

UTF-8编码可以通过屏蔽位和[移位](https://baike.baidu.com/item/%E7%A7%BB%E4%BD%8D)操作快速读写。[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)串比较时strcmp()和[wcscmp](https://baike.baidu.com/item/wcscmp)()的返回结果相同，因此使排序变得更加容易。[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82)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](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82%E9%A1%BA%E5%BA%8F)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。(来源维基百科)

4.4、UTF 8/16编码集字节数

对于UTF-8编码中的任意字节B，如果B的第一位为0，则B独立的表示一个字符(ASCII码)；如果B的第一位为1，第二位为0，则B为一个多字节字符中的一个字节(非ASCII字符)；如果B的前两位为1，第三位为0，则B为两个字节表示的字符中的第一个字节；如果B的前三位为1，第四位为0，则B为三个字节表示的字符中的第一个字节；如果B的前四位为1，第五位为0，则B为四个字节表示的字符中的第一个字节；

因此，对UTF-8编码中的任意字节，根据第一位，可判断是否为ASCII字符；根据前二位，可判断该字节是否为一个字符编码的第一个字节；根据前四位（如果前两位均为1），可确定该字节为字符编码的第一个字节，并且可判断对应的字符由几个字节表示；根据前五位（如果前四位为1），可判断编码是否有错误或数据传输过程中是否有错误(来源维基百科)