编码标准

gaccob

2012 年 9 月 7 日

ASCII

最早的编码标准, 1个字节, 128个字符集, 0×00-0x7F.

EASCII

Extended ASCII, 延伸美国标准信息交换码, EASCII码比ASCII码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. 相比于ASCII码, 增加了128－255的这一段.

GB2312

GB2312开始支持了汉字, 1981年5月开始在国内实行.

GB2312基于分区处理, 编码时采用2字节, 兼容ASCII码, 高字节(0xA1-0xF7, =区号+0xA0),低字节(0xA1-0xFE), 高位都是1, 所以能兼容ASCII码.

BIG5

BIG5开始支持了繁体汉字. 与GB2312一样, 用两个字节来为每个字符编码, 第一个字节称为"高位字节", 第二个字节称为"低位字节".

Unicode

为了解决传统的字符编码方案的局限而产生的, 由国际标准化组织（ISO）发布的编码标准.

Unicode是一种编码标准, 它的实现有很多种: UTF(Unicode Translation Format)的UTF8和UTF16, GB系列的GBK和GB18030等.

- GBK

  在unicode推出之后, 中国大陆制定了GB13000标准, 几乎等同于unicode1.1.

  1995年微软利用了GB2312中未使用的编码空间, 收录了GB13000中的所有字符制定了汉字内码扩展规范GBK.

  GBK编码最多使用2个字节：字节为00－7F的表示与ASCII完全一致；最高位为1的字节表示2个字节编码, 高字节范围为81－FE, 低字节范围为40－FE, 或者80－FE.

- GB18030

  在2000年, 电子工业标准化研究所起草了GB18030标准, 项目代号"GB 18030-2000", 全称《信息技术-信息交换用汉字编码字符集-基本集的扩充》.

  GB18030收录了GBK中的所有字符, 并将Unicode中其他中文字符（少数民族文字, 偏僻字）也一并收录进来重新编码.

  采用多字节编码, 每个字符由1或2或4个字节进行编码.

- UTF8

  UTF8文件头"EF BB BF"（BOM-byte order mark）, 但不是所有的UTF8文件都有.

  UTF-8 是以8位为单元对原始Unicode码进行编码, 并规定：多字节码以转换后第1个字节起头的连续"1"的数目, 表示转换成几个字节："110"连续两个"1", 表示转换结果为2个字节, "1110"表示3个字节, 跟随在标记位之后的"0", 其作用是分隔标记位和字符码位.

  第2～第4个字节的起头两个位固定设置为"10", 也作为标记, 剩下的6个位才做为字符码位使用. 2字节UTF-8码剩下11个字符码位, 可用以转换0080～07FF的原始字符码；3字节剩下16个字符码位, 可用以转换0800～FFFF的原始字符码, 由此类推

```
原始码（16进制）  UTF－8编码（二进制）
0000 – 007F 0xxxxxxx
0080 – 07FF 110xxxxx 10xxxxxx
0800 – FFFF 1110xxxx 10xxxxxx  10xxxxxx
```

  ASCII码<007F, 编为1个字节的UTF8码.

  汉字的 Unicode编码范围为0800－FFFF, 所以被编为3个字节的UTF8码.

  下面这一段是根据utf8字符获得unicode的解析代码:

```
int _get_unicode(const char* str, int n)
{
    int i;
    int unicode = str[0] & ((1 << (8-n)) - 1);
    for (i=1; i<n; i++) {
        unicode = unicode << 6 | ((uint8_t)str[i] & 0x3f
            );
    }
    return unicode;
}

int get_unicode(char** utf8, int* unicode)
{
    uint8_t c;
    if (!utf8 || !(*utf8) || !unicode) {
        return -1;
    }
    if (**utf8 == 0) {
        return -1;
    }
    c = (*utf8)[0];
    if ((c & 0x80) == 0) {
        *unicode = _get_unicode(*utf8, 1);
        *utf8 += 1;
    } else if ((c & 0xe0) == 0xc0) {
        *unicode = _get_unicode(*utf8, 2);
        *utf8 += 2;
    } else if ((c & 0xf0) == 0xe0) {
        *unicode = _get_unicode(*utf8, 3);
        *utf8 += 3;
    } else if ((c & 0xf8) == 0xf0) {
        *unicode = _get_unicode(*utf8, 4);
        *utf8 += 4;
    } else if ((c & 0xfc) == 0xf8) {
        *unicode = _get_unicode(*utf8, 5);
        *utf8 += 5;
    } else if ((c & 0xfe) == 0xfc) {
        *unicode = _get_unicode(*utf8, 6);
        *utf8 += 6;
    } else {
        return -1;
    }
    return 0;
}
```

参考文章

1. 字符编码的前世今生

2. http://baike.baidu.com/view/42488.htm
3. http://zh.wikipedia.org/wiki/Unicode