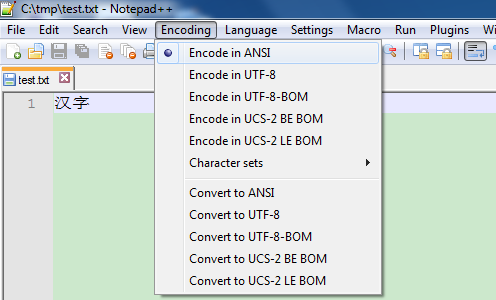
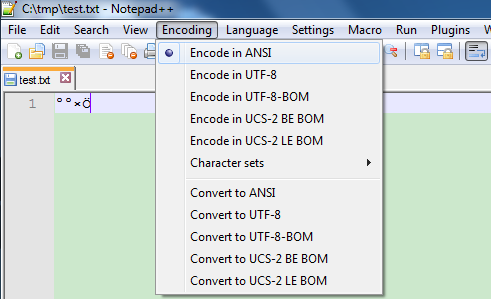
**[ANSI是什么编码？](http://www.cnblogs.com/malecrab/p/5300486.html)**

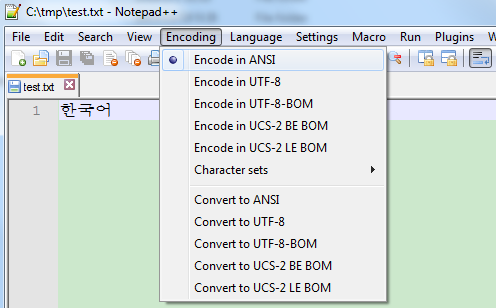
用Notepad++创建一个文本文件text.txt，其默认编码格式为ANSI（乍看之下，还以为是ASCII呢），输入汉字居然不是乱码：  


保存为test.txt，发送给你美国的同事Bob。他也用Notepad++，不幸的是，却发现你的文件内容是这样的：  


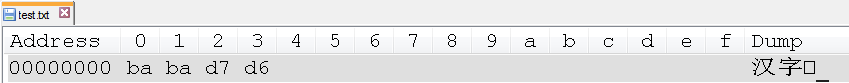
也许你会认为：你用的是中文系统，能正常显示中文；他用的是英文系统，不能显示中文！

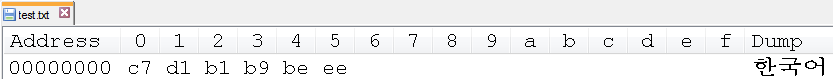
这么想，好像很有道理呢！

但是再细想一下：**一个系统显示乱码，说明它不支持这种编码格式（或者解码方式不对）。难道英文系统不支持ANSI？难道ANSI是一种中文编码？**

如果你身边有一个韩文系统，也装一个Notepad++，默认还是ANSI编码，你可以输入“한국어”，发现也能正常显示：  
  
但是你要输入“汉字”可能就会发现是乱码了...

通过这个反例，就可以说明**ANSI不是一种中文编码。那么，ANSI到底是什么编码？**

用十六进制编辑器打开内容为“汉字”的test.txt文件：  
  
你会发现：其中baba和d7d6正好是“汉”和“字”两个字的[GBK编码](http://www.qqxiuzi.cn/zh/hanzi-gbk-bianma.php)值。

同样，用十六进制编辑器打开内容为“한국어”的test.txt文件：  
  
你会发现：其中c7d1、b1b9和beee正好是“한”、“국”和“어”三个字符的[EUC-KR编码](http://www.chi2ko.com/jingyan/euckr2uni.htm)值。

由此可以看出：其实ANSI并不是某一种特定的字符编码，而是在不同的系统中，ANSI表示不同的编码。你的美国同事Bob的系统中ANSI编码其实是ASCII编码（ASCII编码不能表示汉字，所以汉字为乱码），而你的系统中（“汉字”正常显示）ANSI编码其实是GBK编码，而韩文系统中（“한국어”正常显示）ANSI编码其实是EUC-KR编码。

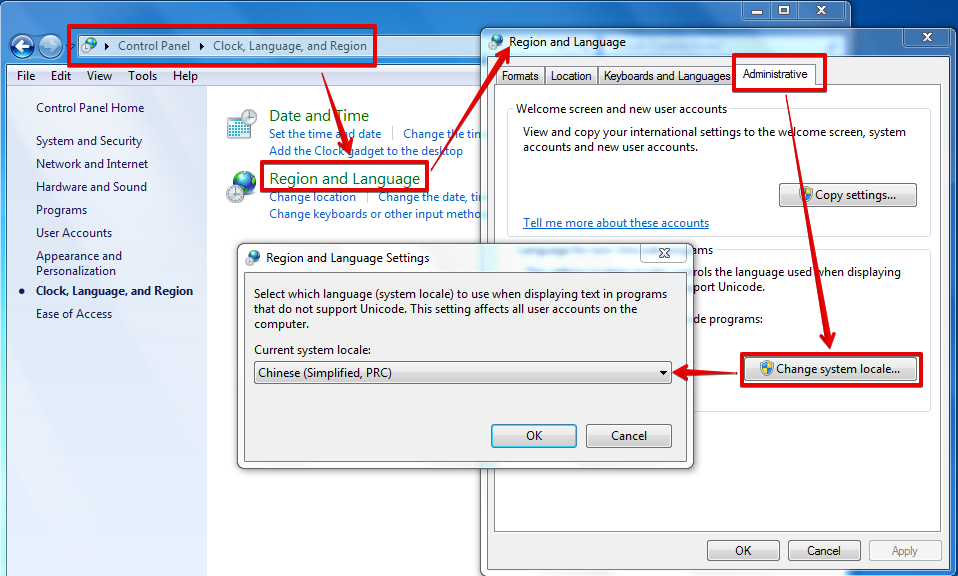
话说计算机是由美国佬搞出来的嘛，他们觉得一个字节（可以表示256个编码）表示英语世界里所有的字母、数字和常用特殊符号已经绰绰有余了（其实ASCII只用了前127个编码）。后来欧洲人不干了，法国人说：我需要在小写字母加上变音符号（如：é），德国人说：我也要加几个字母（Ä ä、Ö ö、Ü ü、ß）。于是，欧洲人就将ASCII没用完的编码（128-255）为自己特有的符号编码（后来称之为“扩展字符集”）。等到我们中国人开始使用计算机的时候，尼玛，256个编码哪够？我泱泱大中华，汉字起码也得N多万吧，就连小学生都得要求掌握两三千字。国标局最后拍板：一个字节不够，那我们就用多个字节来为汉字编码吧，但是，国情那么穷，字节那么贵，三个字节伤不起，那就用俩字节吧，先给常用的几千汉字编个码，等以后国家强盛了人民富裕了，咱再扩展呗---于是GB2312就产生了。台湾同胞一看，尼玛，全是简体字，还让不让我们写繁体字的活了，于是台湾同胞也自己弄了个繁体字编码---大五码（Big-5）。同时，其它国家也在为自己的文字编码。最后，微软苦逼了：顾客就是上帝啊，你们的编码我都得满足啊，这样吧，卖给美国国内的系统默认就用ASCII编码吧，卖给中国人的系统默认就用GBK编码吧，卖给韩国人的系统默认就用EUC-KR编码，...但是为了避免你们误会我卖给你们的系统功能有差异，我就统一把你们的默认编码都显示成ANSI吧。---本故事纯属虚构，但“ANSI编码”确实只存在于Windows系统。

**那么Windows系统是如何区分ANSI背后的真实编码的呢？**

微软用一个叫“[Windows code pages](https://en.wikipedia.org/wiki/Code_page)”（在命令行下执行chcp命令可以查看当前code page的值）的值来判断系统默认编码，比如：简体中文的code page值为936（它表示GBK编码，win95之前表示GB2312，详见：[Microsoft Windows' Code Page 936](https://en.wikipedia.org/wiki/Code_page_1386)），繁体中文的code page值为950（表示Big-5编码）。

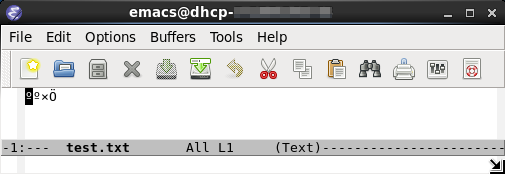
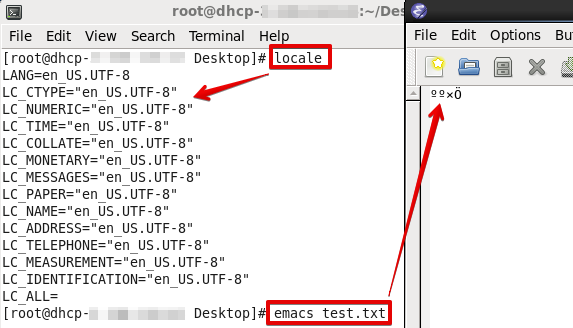
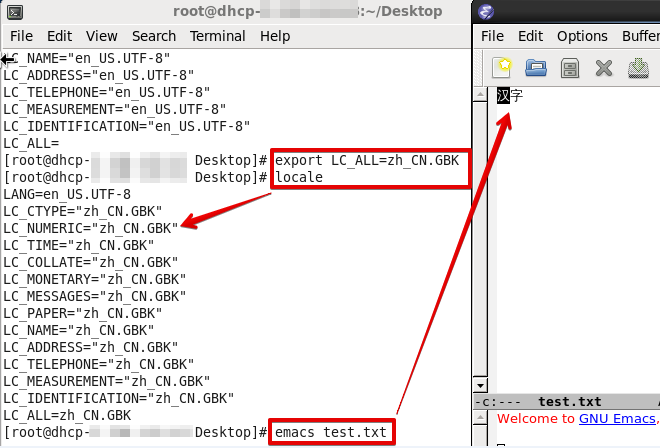
**我们能否通过修改Windows code pages的值来改变“ANSI编码”呢？**

命令提示符下，我们可以通过chcp命令来修改当前终端的active code page，例如：  
(1) 执行：chcp 437，code page改为437，当前终端的默认编码就为ASCII编码了（汉字就成乱码了）；  
(2) 执行：chcp 936，code page改为936，当前终端的默认编码就为GBK编码了（汉字又能正常显示了）。  
上面的操作只在当前终端起作用，并不会影响系统默认的“ANSI编码”。（更改命令行默认codepage参看：[设置cmd的codepage的方法](http://www.cnblogs.com/SunWentao/archive/2010/05/26/1744382.html)）。

Windows下code page是根据当前系统区域（locale）来设置的，要想修改系统默认的“ANSI编码”，我们可以通过修改系统区域来实现（“控制面板” =>“时钟、语言和区域”=>“区域和语言”=>“管理”=>“更改系统区域设置...”）：  
  
图中的系统locale为简体中文，意味着当前“ANSI编码”实际是GBK编码。当你把它改成Korean(Korea)时，“ANSI编码”实际是EUC-KR编码，“한국어”就能正常显示了；当你把它改成English(US)时，“ANSI编码”实际是ASCII编码，“汉字”和“한국어”都成乱码了。（改了之后需要重启系统的。。。）

说明：locale是国际化与本地化中重要的概念，本文不深入讲解该内容。

**你上面说的都是windows的情形吧，Linux呢？**

将前述内容为“汉字”的文件test.txt拷贝至Linux下，用Emacs打开：  
  
也是乱码！原因也是locale的问题：  
  
更改locale后再打开：  
  
正常显示了。。。