Linux解决vim中文乱码

2012年8月2日　　10:14

解决vim中文乱码问题

由于在windows下默认是gb编码，而我的vim默认是utf-8（gedit默认也是utf-8），所以打开会成乱码。修改了一下配置文件，使vim支持gb编码就好了。

$vim ~/.vimrc

let &termencoding=&encoding
set fileencodings=utf-8,gbk

$:wq

再次打开vi，显示就正常了。如果不正常，重新开一个终端，再次打开vi。

更详细的资料：
vim中编辑不同编码的文件时需要注意的一些地方

此文讲解的是vim编辑多字节编码文档（中文）所要了解的一些基础知识，注意其没有涉及gvim，纯指字符终端下的vim。
vim编码方面的基础知识：
1，存在3个变量：
encoding—-该选项使用于缓冲的文本(你正在编辑的文件)，寄存器，Vim 脚本文件等等。你可以把 ‘encoding’ 选项当作是对 Vim 内部运行机制的设定。
fileencoding—-该选项是vim写入文件时采用的编码类型。
termencoding—-该选项代表输出到客户终端（Term）采用的编码类型。
2，此3个变量的默认值：
encoding—-与系统当前locale相同，所以编辑文件的时候要考虑当前locale，否则要设置的东西就比较多了。
fileencoding—-vim打开文件时自动辨认其编码，fileencoding就为辨认的值。为空则保存文件时采用encoding的编码，如果没有修改encoding，那值就是系统当前locale了。
termencoding—-默认空值，也就是输出到终端不进行编码转换。
由此可见，编辑不同编码文件需要注意的地方不仅仅是这3个变量，还有**系统当前locale**和、**文件本身编码以及自动编码识别**、**客户运行vim的终端所使用的编码类型**3个关键点，这3个关键点影响着3个变量的设定。
如果有人问：为什么我用vim打开中文文档的时候出现乱码？
答案是不确定的，原因上面已经讲了，不搞清楚这3个关键点和这3个变量的设定值，出现乱码是正常的，倒是不出现乱码那反倒是凑巧的。
再来看一下常见情况下这三个关键点的值以及在这种情况下这3个变量的值：
1，locale—-目前大部分Linux系统已经将utf-8作为默认locale了，不过也有可能不是，例如有些系统使用中文locale zh_CN.GB18030。在locale为utf-8的情况下，启动vim后encoding将会设置为utf-8，这是兼容性最好的方式，因为内部 处理使用utf-8的话，无论外部存储编码为何都可以进行无缺损转换。locale决定了vim内部处理数据的编码，也就是encoding。
2，文件的编码以及自动编码识别—-这方面牵扯到各种编码的规则，就不一一细讲了。但需要明白的是，文件编码类型并不是保存在文件内的，也就是说没有任何 描述性的字段来记录文档是何种编码类型的。因此我们在编辑文档的时候，要么必须知道这文档保存时是以什么编码保存的，要么通过另外的一些手段来断定编码类 型，这另外的手段，就是通过某些编码的码表特征来断定，例如每个字符占用的字节数，每个字符的ascii值是否都大于某个字段来断定这个文件属于何种编 码。这种方式vim也使用了，这就是vim的自动编码识别机制了。但这种机制由于编码各式各样，不可能每种编码都有显著的特征来辨别，所以是不可能 100%准确的。对于我们GB2312编码，由于其中文是使用了2个acsii值高于127的字符组成汉字字符的，因此不可能把gb2312编码的文件与 latin1编码区分开来，因此自动识别编码的机制对于gb2312是不成功的，它只会将文件辨识为latin1编码。此问题同样出现在gbk，big5 上等。因此我们在编辑此类文档时，需要手工设定encoding和fileencoding。如果文档编码为utf-8时，一般vim都能自动识别正确的 编码。
3，客户运行vim的终端所使用的编码类型—-同第二条一样，这也是一个比较难以断定的关键点。第二个关键点决定着从文件读取内容和写入内容到文件 时使用的编码，而此关键点则决定vim输出内容到终端时使用的编码，如果此编码类型和终端认为它收到的数据的编码类型不同，则又会产生乱码问题。在 linux本地X环境下，一般终端都认为其接收的数据的编码类型和系统locale类型相符，因此不需关心此方面是否存在问题。但如果牵涉到远程终端，例 如ssh登录服务器，则问题就有可能出现了。例如从1台locale为GB2310的系统（称作客户机）ssh到locale为utf-8的系统（称作服 务器）并开启vim编辑文档，在不加任何改动的情况下，服务器返回的数据为utf-8的，但客户机认为服务器返回的数据是gb2312的，按照 gb2312来解释数据，则肯定就是乱码了，这时就需要设置termencoding为gb2312来解决这个问题。此问题更多出现在我们的 windows desktop机远程ssh登录服务器的情况下，这里牵扯到不同系统的编码转换问题。所以又与windows本身以及ssh客户端有很大相关性。在 windows下存在两种编码类型的软件，一种是本身就为unicode编码方式编写的软件，一种是ansi软件，也就是程序处理数据直接采用字节流，不 关心编码。前一种程序可以在任何语言的windows上正确显示多国语言，而后一种则编写在何种语言的系统上则只能在何种语言的系统上显示正确的文字。对 于这两种类型的程序，我们需要区别对待。以ssh客户端为例，我们使用的putty是unicode软件，而secure CRT则是ansi 软件。对于前者，我们要正确处理中文，只要保证vim输出到终端的编码为utf-8即可，就是termencoding=utf-8。但对于后者，一方面 我们要确认我们的windows系统默认代码页为cp936（中文windows默认值），另一方面要确认vim设置的termencoding=cp936。
最后来看看处理中文文档最典型的几种情况和设置方式：
1，系统locale是utf-8（很多linux系统默认的locale形式），编辑的文档是GB2312或GBK形式的（Windows记事本 默认保存形式，大部分编辑器也默认保存为这个形式，所以最常见），终端类型utf-8（也就是假定客户端是putty类的unicode软件）
则vim打开文档后，encoding=utf-8（locale决定的），fileencoding=latin1（自动编码判断机制不准导致的），termencoding=空（默认无需转换term编码），显示文件为乱码。

解决方案1：首先要修正fileencoding为cp936或者euc-cn（二者一样的，只不过叫法不同），注意修正的方法不是:set fileencoding=cp936，这只是将文件保存为cp936，正确的方法是重新以cp936的编码方式加载文件为:edit ++enc=cp936，可以简写为:e ++enc=cp936。
解决方案2：临时改变vim运行的locale环境，方法是以LANG=zh_CN vim abc.txt的方式来启动vim，则此时encoding=euc-cn（locale决定的），fileencoding=空（此locale下文件 编码自动判别功能不启用，所以fileencoding为文件本身编码方式不变，也就是euc-cn），termencoding=空（默认值，为空则等 于encoding）此时还是乱码的，因为我们的ssh终端认为接受的数据为utf-8，但vim发送数据为euc-cn，所以还是不对。此时再用命令: set termencoding=utf-8将终端数据输出为utf-8，则显示正常。
2，情况与1基本相同，只是使用的ssh软件为secure CRT类ansi类软件。
vim打开文档后，encoding=utf-8（locale决定的），fileencoding=latin1（自动编码判断机制不准导致的），termencoding=空（默认无需转换term编码），显示文件为乱码。
解决方案1：首先要保证运行secure CRT的windows机器的默认代码页为CP936，这一点中文windows已经是默认设置了。其他的与上面方案1相同，只是要增加一步，:set termencoding=cp936
解决方案2：与上面方案2类似，不过最后一步修改termencoding省略即可，在此情况下需要的修改最少，只要以locale为zh_CN开 启vim，则encoding=euc-cn，fileencoding和termencoding都为空即为encoding的值，是最理想的一种情 况。
可见理解这3个关键点和3个参数的意义，对于编码问题有很大助力，以后就可以随心所欲的处理文档了，同时不仅仅是应用于vim，在其他需要编码转换的环境里，都可以应用类似的思路来处理问题解决问题。
最后推荐一款功能强大的windows下的ssh客户端—-xshell，它具有类似secure CRT一样的多tab 的ssh窗口的能力，但最为方便的是这款工具还有改变Term编码的功能，这样我们就可以不用频繁调整termencoding，只需在ssh软件里切换 编码即可，这是我用过的最为方便的ssh工具。它是商业软件，但非注册用户使用没有任何限制，只是30天试用期超出后会每次启动都提示注册，对于功能没有 丝毫影响。