

Linux中文乱码问题终极解决方法 - h2z - 博客园

方法一： 修改/root/.bash_profile文件，增加export LANG=zh_CN.GB18030

该文件在用户目录下，对于其他用户，也必须相应修改该文件。

使用该方法时putty能显示中文，但桌面系统是英文，而且所有的网页中文显示还是乱码

方法二：

修改/etc/sysconfig/i18n文件

```
#LANG="en_US.UTF-8"
```

```
#SUPPORTED="en_US.UTF-8:en_US:en"
```

```
#SYSFONT="latarcyrheb-sun16"
```

修改为：

```
LANG="zh_CN.GB18030"
```

```
LANGUAGE="zh_CN.GB18030:zh_CN.GB2312:zh_CN"
```

```
SUPPORTED="zh_CN.GB18030:zh_CN:zh"
```

```
SYSFONT="lat0-sun16"
```

```
SYSFONTACM="8859-15"
```

参考：

Linux中文乱码问题

最近，公司在XP系统于LINUX之间传数据时出现了中文乱码问题！

首先，字符集：

汉字编码：

* GB2312字集是简体字集，全称为GB2312(80)字集，共包括国标简体汉字6763个。 * BIG5字集是台湾繁体字集，共包括国标繁体汉字13053个。 * GBK字集是简繁体字集，包括了GB字集、BIG5字集和一些符号，共包括21003个字符。 * GB18030是国家制定的一个强制性大字符集标准，全称为GB18030-2000，它的推出使汉字集有了一个“大一统”的标准。

ASCII：

American Standard Code for Information Interchange，美国信息交换标准码。 目前计算机中用得最广泛的字符集及其编码，由美国国家标准局(ANSI)制定。 它已被国际标准化组织(ISO)定为国际标准，称为ISO 646标准。 ASCII字符集由控制字符和图形字符组成。 在计算机的存储单元中，一个ASCII码值占一个字节(8个二进制位)，其最高位(b7)用作奇偶校验位。 所谓奇偶校验，是指在代码传送过程中用来检验是否出现错误的一种方法，一般分奇校验和偶校验两种。 奇校验规定：正确的代码一个字节中1的个数必须是奇数，若非奇数，则在最高位b7添1。

偶校验规定：正确的代码一个字节中1的个数必须是偶数，若非偶数，则在最高位b7添1。

UTF：

Unicode 的实现方式不同于编码方式。 一个字符的Unicode编码是确定的，但是在实际传输过程中，由于不同系统平台的设计不一定一致，以及出于节省空间的目的，对Unicode编码的实现 方式有所不同。 Unicode的实现方式称为Unicode转换格式(Unicode Translation Format，简称为 UTF)。 * UTF-8： 8bit变长编码，对于大多数常用字符集(ASCII中0~127字符)它只使用单字节，而对其它常用字符(特别是朝鲜和汉语会意文字)，它使用3字节。 * UTF-16： 16bit编码，是变长码，大致相当于20位编码，值在0到0x10FFFF之间，基本上就是unicode编码的实现，与CPU字序有关。

注意：ASCII char (2) ;UTF-8 宽字符 wchar 4倍 。兼容性最好的编码就是UTF-8！ 毕竟GBK/GB2312是国内的标准，当大量使用国外的开源软件时，UTF-8才是编码界最通用的语言。

在Linux中通过locale来设置程序运行的不同语言环境，locale由ANSI C提供支持。locale的命名规则为<语言>_<地区>.<字符集编码>，如zh_CN.UTF-8，zh代表中文，CN代表大陆地区，UTF-8表示字符集。

在locale环境中，有一组变量，代表国际化环境中的不同设置：

1. LC_COLLATE

定义该环境的排序和比较规则

2. LC_CTYPE

用于字符分类和字符串处理，控制所有字符的处理方式，包括字符编码，字符是单字节还是多字节，如何打印等。是最最重要的一个环境变量。

3. LC_MONETARY

货币格式

4. LC_NUMERIC

非货币的数字显示格式

5. LC_TIME

时间和日期格式

6. LC_MESSAGES

提示信息语言。另外还有一个LANGUAGE参数，它与LC_MESSAGES相似，但如果该参数一旦设置，则LC_MESSAGES参数就会失效。

LANGUAGE参数可同时设置多种语言信息，如LANGUAGE="zh_CN.GB18030:zh_CN.GB2312:zh_CN"。

7. LANG

LC_*的默认值，是最低级别的设置，如果LC_*没有设置，则使用该值。类似于 LC_ALL。

8. LC_ALL

它是一个宏，如果该值设置了，则该值会覆盖所有LC_*的设置值。注意，LANG的值不受该宏影响。

例子：

设置前，使用默认locale：

代码示例：

```
[root@ahlinux ~]# locale
```

```
LANG="POSIX"
```

```
LC_CTYPE="POSIX"
```

```
LC_NUMERIC="POSIX"
```

```
LC_TIME="POSIX"
```

```
LC_COLLATE="POSIX"
```

```
LC_MONETARY="POSIX"
```

```
LC_MESSAGES="POSIX"
```

```
LC_PAPER="POSIX"
```

```
LC_NAME="POSIX"
```

```
LC_ADDRESS="POSIX"
```

```
LC_TELEPHONE="POSIX"
```

```
LC_MEASUREMENT="POSIX"
```

```
LC_IDENTIFICATION="POSIX"
```

```
LC_ALL=
```

设置后，使用zh_CN.GBK中文locale：

代码示例：

```
[root@ahlinux ~]# export LC_ALL=zh_CN.GBK
```

```
[root@ahlinux ~]# locale
```

```
LANG=zh_CN.UTF-8
```

```
LC_CTYPE="zh_CN.GBK"
```

```
LC_NUMERIC="zh_CN.GBK"
```

```
LC_TIME="zh_CN.GBK"
```

```
LC_COLLATE="zh_CN.GBK"
```

```
LC_MONETARY="zh_CN.GBK"
```

```
LC_MESSAGES="zh_CN.GBK"
```

```
LC_PAPER="zh_CN.GBK"
```

```
LC_NAME="zh_CN.GBK"
```

```
LC_ADDRESS="zh_CN.GBK"
```

```
LC_TELEPHONE="zh_CN.GBK"
```

```
LC_MEASUREMENT="zh_CN.GBK"
```

```
LC_IDENTIFICATION="zh_CN.GBK"
```

```
LC_ALL=zh_CN.GBK
```

"C"是系统默认的locale，"POSIX"是"C"的别名。所以当我们新安装完一个系统时，默认的locale就是C或POSIX。

在Debian中安装locales的方法如下：

- 通过apt-get install locales命令安装locales包
- 安装完成locales包后，系统会自动进行locale配置，你只要选择所需的locale，可以多选。最后指定一个系统默认的locale。这样系统就会帮你自动生成相应的locale和配置好系统的locale。
- 增加新的locale也很简单，用dpkg-reconfigure locales重新配置locale即可。
- 我们也可手动增加locale，只要把新的locale增加到/etc/locale.gen文件中，再运行locale-gen命令即可生成新的 locale。再通过设置上面介绍的LC_*变量就可设置系统的locale了。下一个locale.gen文件的样例。

代码示例：

- # This file lists locales that you wish to have built. You can find a list
- # of valid supported locales at /usr/share/i18n/SUPPORTED. Other
- # combinations are possible, but may not be well tested. If you change
- # this file, you need to rerun locale-gen.
- #
- zh_CN.GBK GBK
- zh_CN.UTF-8 UTF-8

在我看来只要搞清楚LANG和SUPPORTED就OK了，其他可能平时也用不太多。

这里再介绍下如何设置环境变量的。

修改 /etc/sysconfig/i18n 文件，如

代码示例：

LANG="en_US.UTF-8"，xwindow会显示英文界面，

LANG="zh_CN.GB18030"，xwindow会显示中文界面。

还有一种方法 cp /etc/sysconfig/i18n \$HOME/.i18n

修改 \$HOME/.i18n 文件，如

代码示例：

LANG="en_US.UTF-8"，xwindow会显示英文界面，

LANG="zh_CN.GB18030"，xwindow会显示中文界面。

这样就可以改变个人的界面语言，而不影响别的用户

修改后的/etc/sysconfig/i18n 文件为：

代码示例：

LANG="en_US.UTF-8"

SUPPORTED="zh_CN.GB18030:zh_CN:zh:en_US.UTF-8:en_US:en"

SYSFONT="latarcyrheb-sun16"

LC_ALL="en_US.UTF-8"

export LC_ALL

设置完毕后重启或者用rc.local使生效

或修改登录用户的.bash_profile文件加入

代码示例：

export LANG=zh_CN.GB18030

export LANGUAGE=zh_CN.GB18030:zh_CN.GB2312:zh_CN

一定要知道Windows XP 是GB2312的编码，如果你的服务器字符集不是这个，估计就会是乱码，所以要调整。

有些人在调整的时候，说我改了系统环境变量，结果造成用户内容显示乱码，无非解决的方法就是两个：

1. 用iconv转化为目前的编码

2. 用你原来使用的编码

看了这两条，你肯定必须要明确，你原来的字符编码是如何的。说来说去，无非就是LANG SUPPORTED 和你原文件字符集的编码：)

当然了locale -a你可以看看目前系统里支持的字符集，如果不支持，还要安装奥。

前两个方法很实用，我试验过。其他方法都是在网上搜到的，呵呵。。。

就是从数据库中取出来时，在存入linux的文件里时，在字符流时制定编码格式。代码如下：

代码示例：

```
FileOutputStream fos=new FileOutputStream(new File(filePath),true);
```

```
Writer out=new OutputStreamWriter(fos,"UTF-8");
```

```
out.write(s);
```

```
out.write("\n");
```

```
out.flush();
```

```
fos.close();
```

```
out.close();
```

```
vi .bash_profile
```

```
export lang=zh_CN
```

```
vi /etc/sysconfig/i18n
```

```
LANG="en_US.UTF-8"
```

```
SUPPORTED="en_US.UTF-8:en_US:en:zh_CN.GB18030:zh_CN:zh:zh_TW.big5:zh_TW:zh:ja_JP.UTF-8:ja_JP:ja:ko_KR.eucKR:ko_KR:ko"
```

```
SYSFONT="latarcyrheb-sun16"
```

只改第一个不管用，好像第二个尤其重要，必须改。

1、console终端乱码

在/etc/profile文件的最后一行添加如下内容：

代码示例：

```
export LC_ALL="zh_CN.GB18030"
```

2、xwindow终端乱码

在/etc/sysconfig/i18n文件的最后一行添加如下内容：

代码示例：

```
export LC_ALL="zh_CN.GB18030"
```

乱码分两种情况:

1. 终端(纯shell界面)的乱码

代码示例:

```
vi /etc/profile
```

```
export LC_ALL="zh_CN.GB18030:zh_CN.GB2312:zh_CN.GBK:zh_CN:en_US.UTF-8:en_US:en:zh:zh_TW:zh_CN.BIG5"
```

保存退出, reboot系统即可..

2. X-window(图形界面)的乱码

代码示例:

```
vi /etc/sysconfig/i18n
```

```
LANG="zh_CN.GB18030:zh_CN.GB2312:zh_CN.GBK:zh_CN:en_US.UTF-8:en_US:en:zh:zh_TW:zh_CN.BIG5"
```

```
LANGUAGE="zh_CN.GB18030:zh_CN.GB2312:zh_CN.GBK:zh_CN:en_US.UTF-8:en_US:en:zh:zh_TW:zh_CN.BIG5"
```

保存reboot即可...

新发了台linux虚拟机, 用VIM出现了中文乱码问题, 查找了资料, 解决方法:

```
vi /etc/sysconfig/i18n
```

将内容改为

代码示例:

```
LANG="zh_CN.GB18030"
```

```
LANGUAGE="zh_CN.GB18030:zh_CN.GB2312:zh_CN"
```

```
SUPPORTED="zh_CN.GB18030:zh_CN:zh:en_US.UTF-8:en_US:en"
```

```
SYSEFONT="lat0-sun16"
```

这样中文在SSH, telnet终端就可以正常显示了

其中, 主要修改的内容主要为zh_CN.GB18030, 其中要注意VI个性的内容为根目录下, 要注意权限。

每次装完linux, 用SSH连接, 中文总是显示乱码。

解决方法: 编辑/etc/sysconfig/i18n, 将其中的 LANG="zh_CN.UTF-8" 改为 LANG="zh_CN.GB2312" 。

断开重新连接。

附1, linux下中文乱 码问题解决方案。

从windows传过来的文件拷到linux下是乱码, 俺们又想在linux下显示中文, 怎么办呢?我们首先进行测试, linux下的中文是否可以正常显示?答:yes. 所以问题就比较明显了, windows下拷过来的不能显示, 说明windows下和linux下支持的格式是不相同的。

linux下一般是采用utf-8的编码, 而我们在windows上编辑文件时是gb2312的编码。所以导致中文编码会乱码。要更正这个问题实际上很简单只要把文件转换成utf-8编码格式然后再导入就ok啦。

然后使用如下命令进行转换:

```
iconv -f gb2312 -t utf-8 test.txt> testutf8.tzt
```

(-f 就是源编码, -t 转换目标编码, test.txt源文件, testutf8.txt生成目标编码文件)

备注: 使用 iconv -l 可以查看系统支持编码格式。当然也可以添加其中的编码格式:

默认为utf8, 如果想用其他编码如GBK

手动更改配置文件的命令:

```
shell> vi /etc/sysconfig/i18n
```

将LANG="zh_CN.UTF-8"修改为:

```
LANG="zh_CN.GBK"
```

保存并关闭, 运行下面的命令使配置生效:

```
shell> source /etc/sysconfig/i18n
```

将终端字符编码显示为简体中文:

```
shell> vi /etc/profile.d/chinese.sh
```

添加下面这行:

代码示例:

```
export LC_ALL=zh_CN.GBK
```

```
shell> source /etc/profile.d/Chinese.sh
```

附2, 解决Linux操作系统下Java的中文乱码问题。

在jdk15以后, 只需在~/jre/lib/fonts/下建一个fallback目录, 把你想在java中使用的字体拷贝到这个目录中即可

以下方法在fc6下测试通过, 假设用户的jre路径为 /usr/java/jdk1.6.0_03/jre/

代码示例:

```
cd /usr/java/jdk1.6.0_03/jre/lib/fonts
```

```
sudo mkdir fallback
```

将C:\WINDOWS\Fonts\simsun.ttc拷贝到 /usr/java/jdk1.6.0_03/jre/lib/fonts/fallback文件夹内

`export LC_ALL=zh_CN.GB2312;export LANG=zh_CN.GB2312`是最有效的。

1. 不管用那种ssh客户端，字体设定一定要设为可以显示中文的字体。

2. 远程的locale一定要设置为LANG=zh_CN.UTF-8

修改/etc/profile

增加这一行

`export LC_ALL=zh_CN.GBK`

附3，SSH显示中文乱码问题

1)、打开/etc/sysconfig/i18n

设置为：

代码示例：

`LANG="zh_CN.GB2312"`

`LANGUAGE="zh_CN.GB18030:zh_CN.GB2312:zh_CN"`

`SUPPORTED="zh_CN.GB18030:zh_CN.GB2312:zh_CN.UTF-8:zh:en_US.UTF-8:en_US:en:ja_JP.UTF-8:ja_JP:ja"`

`SYSFONT="lat0-sun16"`

`SYSFONTACM="8859-15"`

其中LANG="zh_CN.GB2312" 是必须的(如果你不想让中文乱码的话!!!)

其它的可以按照自己的需求来改变。

2)、打开smb.conf

添加：

代码示例：

`display charset=cp936`

`unix charset=cp936`

`doc charset=cp936`