[**linux环境中英文切换配置以及乱码问题**](http://blog.csdn.net/gatieme/article/details/50991584)

标签： [linux](http://www.csdn.net/tag/linux)[服务器](http://www.csdn.net/tag/%e6%9c%8d%e5%8a%a1%e5%99%a8)[输入法](http://www.csdn.net/tag/%e8%be%93%e5%85%a5%e6%b3%95)[语言](http://www.csdn.net/tag/%e8%af%ad%e8%a8%80)[谷歌](http://www.csdn.net/tag/%e8%b0%b7%e6%ad%8c)

2016-03-27 14:47 2276人阅读 [评论](http://blog.csdn.net/gatieme/article/details/50991584#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/gatieme/article/details/50991584#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

┣〓Ubuntu/CentOS管理〓┫（118） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章 && 转载请著名出处 @ http://blog.csdn.net/gatieme

目录[(?)[+]](http://blog.csdn.net/gatieme/article/details/50991584)

问题

这几天玩实验室的服务器，默认语言是中文的，有时候是挺不方便的，比如遇见一个比较复杂的问题，提示的信息是中文的，但是百度里根本没有好的资料，只能google，但是难道让我google搜中午么。

因此就想把自己的环境还原成英文的。

基础配置

本质就是修改系统的LANG变量

LANG是language的简称，稍微有英语基础的用户一看就看出来这个变量是决定系统的默认语言的，即系统的菜单、程序的工具栏语言、输入法默 认语言等。

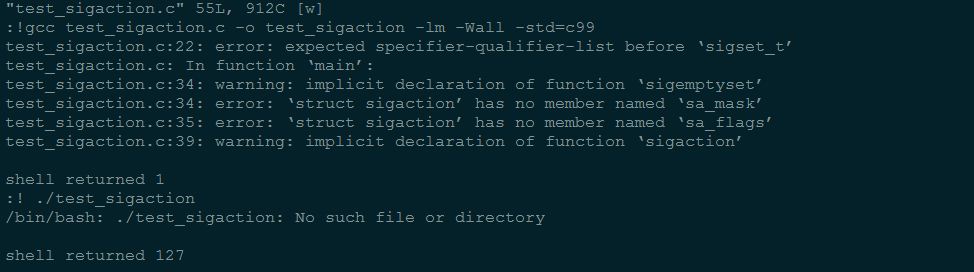
配置自己的LANG

如果你希望修改整个系统的编码和语言信息，可以修改系统的配置文件修改LANG，而如果你不希望影响其他用户直接在你自己的~/.bashrc中配置LANG即可

# use english

export LANG=en\_US.UTF-8

* 1
* 2
* 1
* 2



修改配置文件

CentOS

配置文件/etc/sysconfig/i18n   
中文环境设置：

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"

SYSFONT="lat0-sun16"

* 1
* 2
* 3
* 4
* 1
* 2
* 3
* 4

英文环境设置：

LANG="en\_US"

LANGUAGE="en\_US"

SUPPORTED="zh\_CN.GB18030:zh\_CN:zh:en\_US.UTF-8:en\_US:en"

SYSFONT="lat0-sun16"

SYSFONTACM="8859-15"

* 1
* 2
* 3
* 4
* 5
* 1
* 2
* 3
* 4
* 5

Ubuntu

1、安装中文语言包

sudo apt-get install language-pack-zh

* 1
* 1

2、用vim配置语言环境变量

vim /etc/environment

* 1
* 1

在下面添加如下两行

LANG=”zh\_CN.UTF-8″

LANGUAGE=”zh\_CN:zh:en\_US:en”

* 1
* 2
* 1
* 2

如果你想用英文环境了，改成这两行就OK

LANG=”en\_US.UTF-8″

LANGUAGE=”en\_US:en”

* 1
* 2
* 1
* 2

进阶配置

另外对于国内的[**Linux**](http://lib.csdn.net/base/linux)用户,经常烦恼的一个问题是:   
系统常常在需要显示中文的时候却显示成了乱码,而由于某些原因,需要英文界面的系统的时候,却苦于系统不能正常输入和显示中文.

此外,由于大部分主要Linux发行版都是以英语为主体的,英文界面的系统和应用程序不管在界面的美观程度和稳定程度上 都比中文的略好一些,各种奇怪的BUG也要少一些。

因此，很多稍微有英语基础的Linux用户都宁愿使用英文界面的系统。

但是，矛盾又突现出来：在英文系统下，如何才能正常显示和输入中文呢？有没有两全其美的方案呢？因此，笔者开始探索如何解决这个问题。

我的完美状态是：系统和应用程序全部为英文(系统菜单、应用程序工具栏、默认输入法等），而我需要阅读和撰写中文文档的时候，能正确的显示中文并调出中文输入法。经过尝试，设置成功，现在以Linux为例，讲解一些相关知识和设置过程。

本文主要通过修改系统配置的过程来展现修改linux系统配置的一般思路和过程，如果你不太有耐心看完，请跳过文章的1—4节，直接看第五节快速设置部分。

locale指令

运行locale指令得到当前系统编码设置的详细资料   
locale的五脏六腑

1、 语言符号及其分类(LC\_CTYPE)   
2、 数字(LC\_NUMERIC)   
3、 比较和排序习惯(LC\_COLLATE)   
4、 时间显示格式(LC\_TIME)   
5、 货币单位(LC\_MONETARY)   
6、 信息主要是提示信息,错误信息, 状态信息, 标题, 标签, 按钮和菜单等(LC\_MESSAGES)   
7、 姓名书写方式(LC\_NAME)   
8、 地址书写方式(LC\_ADDRESS)   
9、 电话号码书写方式(LC\_TELEPHONE)   
10、度量衡表达方式(LC\_MEASUREMENT)   
11、默认纸张尺寸大小(LC\_PAPER)   
12、对locale自身包含信息的概述(LC\_IDENTIFICATION)。

相关变量介绍

我 们知道大部分Linux系统是无所谓中文版和英文版的，以Linux为例,系统发行的时候全世界都一样，系统是中文的还是英文的完全取决于你选择 的语言包。

不同国家的人在安装使用的时候选择属于自己国家的语言包，应用程序中的语言也不是写死的，它根据系统的设置来调用相关的语言，所以，一个应用程序写出来不经过修改，全世界不同国家的用户都可以以母语界面使用它，这就事所谓的**internationalization（国际化），简称 i18n**。这也是未来软件的发展趋势。

那么，如果我在系统中安装了不同的语言包和不同的字体，系统是如何判断我所要的语言界面并调用相关的字体的呢？系统中那些文件和变量在控制这些呢？

在redHat系列的Linux系统下，记录系统默认使用语言的文件是/etc/sysconfig/i18n,   
如果默认安装的是中文的系统，i18n的内容如下：

LANG="zh\_CN.UTF-8"

SYSFONT="latarcyrheb-sun16"

SUPPORTED="zh\_CN.UTF-8:zh\_CN:zh"

* 1
* 2
* 3
* 1
* 2
* 3

这里写图片描述

其中LANG变量是language的简称，稍微有英语基础的用户一看就看出来这个变量是决定系统的默认语言的，即系统的菜单、程序的工具栏语言、输入法默 认语言等。

SYSFONT是system font的简称，决定系统默认用哪一种字体。

SUPPORTED变量决定系统支持的语言，即系统能够显示的语言。需要说明的是，由于计算机起源于英语国家，因此，不管你把这些变量设置成什么，英语总是默认支持的，而且不管用什么字体，英文字体总包含在其中。

这些变量中LANG变量是在字符模式 和图形界面下都用到的，在你登录系统后就被读取并生效，相信很多人在字符界面下输入Linux命令的时候经常会遇到显示出来的出错信息是乱码的情况，必需 安装zhcon或者cce等字符模式下的中文软件才能正常显示中文的出错信息。如果我不要他显示中文乱码，我也不要为了看个很简单的出错信息而特意起用 zhcon那我该怎么办呢？一个简单的零时解决的办法就是设置一下LANG变量：

LANG="en\_US.UTF-8"

* 1
* 1

即把系统的语言临时设置成英文，或者更简单一点，可以直接这样：

LANG=""

* 1
* 1

即 把LANG变量清空，由于英语是无论什么情况都支持的，LANG变量被清空后，系统就默认用英语。这样设置后，在字符模式下输出的出错信息等就是全英文的 了。但这种设置是临时的，只是临时改变了LANG这个bash变量而已。当退出重新登录或者切换到其他字符终端后就无效了。到现在，读者应该想到了，只要 把i18n文件中的LANG变量设置成英文的”en\_US.UTF-8”，就可以永久解决这个问题了。

LANG="zh\_CN.UTF-8"

LANG="en\_US.UTF-8"

SYSFONT="latarcyrheb-sun16"

SUPPORTED="zh\_CN.UTF-8:zh\_CN:zh"

* 1
* 2
* 3
* 4
* 1
* 2
* 3
* 4

请不要把LANG变量简单的清空，因为这个变量不仅在字符模式下用到，在图形界面下也用到，简单清空在字符模式下确实不会有问题，但在图形界面下，却会造成 中文无法正常显示的情况，在过去Re d ha t 系列的版本中i18n文件中还有一个叫LANGUAGE的变量，专门控制图形界面下的语言设置，现在的FC系列中已经把这两个变量整合成一个变量了。   
经 过修改这个变量，重新起动图形界面后，就可以看到界面一经完全是英文的了。但是按ctrl+空格却不能调出中文输入法，输入法菜单中也不能添加中文输入 法。我们只简单的修改了LANG变量改变了系统语言设置，当然这一步也可以用图形界面下的工具修改，而不用亲自修改配置文件。

关于运行等级

这个问题似乎和这篇文章的的主题无关，但是介于现在越来越多的linux新手遇到linux图形界面的问题，而且在中文输入法设置过程中也牵涉到这些问题，因此想顺便提一下。   
现在的linux安装完成后，默认就运行在第5个系统运行级别。在SYSTEM V 风格的UNIX系统中，系统被分为不同的运行级别，这和BSD分支的UNIX有所不同，常用的为0~6七个级别:

| **Tables** | **Are** | **Cool** |
| --- | --- | --- |
| 0 | 关机 |  |
| 1 | 单用户 |  |
| 2 | 不带网络的多用户 |  |
| 3 | 带网络的多用户 |  |
| 4 | 保留，用户可以自给定义 |  |
| 5 | 图形界面的多用户 |  |
| 6 | 重起系统 |  |

由于现在的Linux系统安装完后就运行在第5个级别，即系统启动后直接进入图形界面，而不用在字符模式下登录后用startx或者xinit 来起动图形界面。

这样看起来很方便。但是有什么坏处呢？

一旦你改变了某些设置，显示出问题的时候，系统不断在图形和字符间派徊，新手又不知道如何应对， 十分麻烦，而且对于学习研究Linux的人来说，这样不利于了解和学习Linux底层的一些东西。很早就用Linux的老用户都知道，过去的Linux如 redhat6.0，都是默认运行接别为3，即使后来的RedHat9.0也可以在安装时候选默认字符登录还是图形登录。

但现在大多数版本都不管三七二十一直接帮用户选择了直接图形界面登录。虽然对于大多数菜鸟来说，Linux确实越来越简单了，但是很多乐趣，那些新手也体验不到了。

也许你不相信，直接图形登录到系统确实会有很多问题，因此我强烈建议在系统安装完成后把系统的默认运行等级设置在第3级，在字符终端登录后，再手工输入startx 命令起动图形界面。可以用如下的方法修改：   
用文本编辑器修改　/etc/inittab文件，把

id:5:initdefault:

* 1
* 1

这一行，修改成

id:3:initdefault:

* 1
* 1

保存后就重启，系统就默认起动到字符界面。

不同运行级别之间的差别的在于系统默认起动的服务的不同，如运行级别3默认不启动X图形界面服务，而运行级别5 却默认起动。

本质上是没有区别的，更无所谓不同级别间功能强弱的问题。用户完全可自给定义不同级别的默认服务。在任何运行级别，用户都可用init 命令来切换到其他运行级别。

调出中文输入法

我之所以要在上面费那么多笔墨来讲系统运行级别，是因为对系统的认识是从底部向上开始的。

先把默认运行级别修改成3级别，当然，如果你实在不想修改，就临时用init 3命令切换到第3级。

这样你就可以用startx起动图形界面，然后用ctrl+alt+backspace退出图形界面。请注意我说的是“退出”图形界面，而不是按ctrl+alt＋F2切到一个字符终端。

好了，一切由startx开始。当你需要在Linux系统中设置某个东西，或者配置某个服务的时候，最关键的是要知道，这一切是怎么开始的。知其然必需知其 所以然。

如果你有空把/etc/rc.d目录下的系统起动时运行的脚本通读一遍，就完全知道了/etc下的各种配置文件是用来干什么的、如何修改、修改后 有什么效果等等。

玩起系统来也能随心所欲想怎么改就怎么改。这就是我一直强调的,知其然一定要知其所以然。一定要深入系统，读脚本，学会用命令和手工修改 系统配置文件。这样对系统才会有透彻的了解，整天用图形界面的工具是不能帮助你对系统有教为透彻深入的了解的，不同的linux系统提供的图形界面配置程 序会不同，但命令和配置文件都是相同的，越是底层的东西越具有通用性。所以，应当先学会手动配置和修改系统配置文件，等熟悉了以后，再用图形界面的工具修 改，以便减少工作量。

上面提了一下我解决问题的思路。我是顺着这个思路开始的：   
中文输入法是在图形界面下使用的，是图形界面下运行的一个程序。

而图形界面中的一切，都是由startx程序开启运行的。这就是问题的根源。   
找出startx的位置：

which startx /usr/X11R6/bin/startx

* 1
* 1

看startx是一个脚本还是二进制文件：

file /usr/X11R6/bin/startx /usr/X11R6/bin/startx: Bourne shell script text executable

* 1
* 1

发现startx是一个shell 脚本，于是我打开它分析并阅读，看看能不能找到一些关于输入法起动过程和相关变量的线索：

vi /usr/X11R6/bin/startx

* 1
* 1

我找到了该脚本在运行过程中调用的其他脚本和配置文件的信息：

userclientrc=$HOME/.xinitrc

userserverrc=$HOME/.xserverrc

sysclientrc=/etc/X11/xinit/xinitrc

sysserverrc=/etc/X11/xinit/xserverrc

* 1
* 2
* 3
* 4
* 1
* 2
* 3
* 4

并且知道，startx的目的是寻找系统中可用的桌面系统X服务器 系统、以及用户自定义的参数，最终调用xinit来初始化X图形界面。我没有在startx脚本中找到直接和起动输入法相关的代码，于是就可以肯定，输入 法相关的代码在startx调用的脚本中。

于是我来到/etc/X11/xinit/ 目录下，阅读并分析该目录下的脚本，这些脚本有些是startx直接调用的，有些是被startx调用的脚本再调用的，存在着多级嵌套的关系，没有一点耐 心还真是搞不清楚。最终我在/etc/X11/xinit/xinitrc.d目录中的xinput.sh脚本中找到了和输入法相关的代码：

lang\_region=$(echo $tmplang | sed -e 's/"..\*//')

lang\_region="zh\_CN"　 #这一行是修改后加上去的 for f in $HOME/.xinput.d/${lang\_region} " $HOME/.xinput.d/default " /etc/X11/xinit/xinput.d/${lang\_region} "

/etc/X11/xinit/xinput.d/default ;

do [ -r $f ] && source $f && break done

* 1
* 2
* 3
* 4
* 1
* 2
* 3
* 4

通过分析脚本，我知 道，图形界面启动的时候脚本是根据LANG变量来决定是否启用输入法，以及启用哪种语言的输入法等。问题在于：我们还没有把LANG变量改成英语之前，系 统得到的LANG变量是中文的，因此，它知道需要在图形界面启动过程中启用中文输入法，但把LANG变量改成英文后，系统根据LANG 变量知道系统是英文的，它便不再启动中文输入法，也不再设置和导出相关的变量，导致中文输入法不可用。因此，只要在这个脚本中，“骗”过系统，让输入法脚 本“以为”系统是中文的，它不就运行中文输入法，并导出相关变量了吗？　于是，通过分析脚本，我在xinput.sh中的：

lang\_region=$(echo $tmplang | sed -e 's/"..\*//') 后面又添加了 lang\_region="zh\_CN"

* 1
* 1

直接把

lang\_region=$(echo $tmplang | sed -e 's/"..\*//')

* 1
* 1

修改成

lang\_region="zh\_CN"

* 1
* 1

也可以

多添加一行是为了以后改过来方便，直接删除添加的一行就可以了。

当然，把for循环中的/etc/X11/xinit/xinput.d/${lang\_region}该成   
/etc/X11/xinit/xinput.d/zh \_CN也可以。

当 然还有其他的改法，前提是你要懂得shell 脚本的语法，看得懂脚本的意思。这样修改后，即便系统是英文的，xinput.sh脚本也会去读取/etc/X11/xinit/xinput.d/zh \_CN　文件并导出其中的内容、设置好XMODIFERS等输入法变量，并运行iiimx输入法程序。   
那么为什么不在图形界面启动后直接运行iiimx输入法程序呢？实验一下就知道，这根本不行。因为输入法程序是须要和被输入的应用程序配合运行的软件，在运行过程中需要导出很多变量。直接运行iiimx只运行了主程序，而没相关变量，没办法和应用程序配合完成输入工作。

完成修改工作后，保存脚本文件。输入startx命令启动图形界面，就可以用全英文的系统界面和中文输入法了。但需要注意的是：由于系统是全英文的，默认输 入法也是英文，通过GNOME或者KDE菜单起动的应用程序第一次输入中文的时候不能按ctrl+空格来切换到中文，需要用鼠标在任务栏上点击输入法图标 切换，第一次切换后以后就可以用ctrl+空格快捷键来切换中英文输入法了。

一些后续问题

某些软件，比如Open Office，通过GNOME或者KDE菜单启动的话，即使切换到中文输入法也输不进中文，这是因为整个桌面系统的环境是英文的，软件“继承”了英文环境 的相关变量，这些软件就“认死理”，就是不让输入中文，这时候可以打开一个gnome终端，把LANG变量临时设置成zh\_CN.UTF-8　：

LANG="zh\_CN.UTF-8"

* 1
* 1

然后在这个gnome终端中，用命令打开open office:

oowriter &

* 1
* 1

这 样Open Office就“继承”了gnome终端的LANG变量，起动后，工具栏和菜单等都是中文的，而且能输入中文。推而广之，任何软件都可以用这种方法，根据 需要，打开中文界面的软件和英文界面的软件。要以英文界面运行软件时，只要从GNOME或者KDE菜单打开，要用中文界面运行软件时，在终端中修该 LANG变量，从修改过LANG变量的终端中通过命令运行即可。当然，如果你还安装了其他语言的字体，你还可以以其他语言的界面来运行程序。如日语：

LANG="ja\_JP.UTF-8"

gedit &

* 1
* 2
* 1
* 2

我用上面的两条命令打开的gedit 编辑器就是全日语界面的，但是能输入中文和英文，并显示日文。从而达到，一个系统，多种语言和文字共存的目的。   
当然，前提是要安装了日语字体和日语locale，否则所有有文字的地方会全部显示成一连串问号。总之，要先懂得原理，之后想怎么玩就怎么玩，随心所欲，完全不受限制，充分享受用Linux的乐趣。

快速设置步骤

1. 修改/etc/sysconf/i18n文件，把

LANG="zh\_CN.UTF-8"

* 1
* 1

修改成：

LANG="en\_US.UTF-8"

* 1
* 1

1. 修改/etc/X11/xinit/xinitrc.d/xinput.sh文件，把其中一行：

lang\_region=$(echo $tmplang | sed -e 's/"..\*//')

* 1
* 1

修改成：

lang\_region=”zh\_CN”

* 1
* 1

1. 重新启动图形界面，就可以用英文的界面并且正确显示中文和输入中文。

以下为个人补充:   
用 ssh登录LINUX时,ssh终端有时会出现乱码,即使你设置LANG=en\_US.UTF-8也一样.我试过 SecureCRT,OpenSSH,SSH Secure Shell的客户端,都出现过这问题.有时更改一下客户端的设置就好了,但是有时更改客户端的设置后只能运行一次某个命令,之后又变成乱码了.   
一个偶然的机会,在网上看到有人说将LANG变量的值改成”C”就行了.不过还真邪了,就来个LANG=C,什么问题都解决了,也不用设置客户端了,真是莫名其妙!不过到现在还不知道那个”C”是什么意思,它咋就那么厉害.