# **[Linux自定义shell终端提示符](https://www.cnblogs.com/lienhua34/p/5018119.html)**

# 1.基本转义字符

PS1的值由一系列静态文本或\和转义字符序列组成，如:

$ PS1=”\u@\H \w$ ”

 表1: Shell 提示符中用到的转义字符

|  |  |
| --- | --- |
| 序列 | 显示值 |
| \a | 以 ASCII 格式编码的铃声 . 当遇到这个转义序列时，计算机会发出嗡嗡的响声。 |
| \d | 以日，月，天格式来表示当前日期。例如，“Mon May 26” |
| \h | 本地机的主机名，但不带末尾的域名。 |
| \H | 完整的主机名。 |
| \j | 运行在当前 shell 会话中的工作数。 |
| \l | 当前终端设备名。 |
| \n | 一个换行符。 |
| \r | 一个回车符。 |
| \s | shell 程序名。 |
| \t | 以24小时制，hours:minutes:seconds 的格式表示当前时间. |
| \T | 以12小时制表示当前时间。 |
| \@ | 以12小时制，AM/PM 格式来表示当前时间，例如“10:51 PM”。 |
| \A | 以24小时制，hours:minutes 格式表示当前时间。 |
| \u | 当前用户名。 |
| \v | shell 程序的版本号，例如4.3。 |
| \V | shell 程序的版本号，例如4.3.11。 |
| \w | 当前工作目录名。 |
| \W | 当前工作目录名的最后部分。 |
| \! | 当前命令的历史号。 |
| \# | 当前 shell 会话中的命令数。 |
| \$ | 这会显示一个"$"字符，除非你拥有超级用户权限。在那种情况下， 它会显示一个"#"字符。 |
| \[ | 标志着一系列一个或多个非打印字符的开始。这被用来嵌入非打印的控制字符，这些字符以某种方式来操作终端仿真器，比方说移动光标或者是更改文本颜色。 |
| \] | 标志着非打印字符序列结束。 |

例如我现在想让提示符显示格式为：“<用户名>@<当前目录名> $ ”，则

$ PS1=”\u@\W\$ ”

效果如下所示，

qwer@poiu#

上面能够满足我们的效果了，但是相对于LinuxMint原始的提示符，缺少了颜色，不太美观。下面我们来学习如何添加颜色。大多数终端仿真器程序支持一定的非打印字符序列来控制，比方说字符属性（像颜色，黑体和可怕的闪烁） 和光标位置。

# 2.字体颜色

字体颜色是由一个ANSI 转义编码来控制的。该控制编码会嵌入字符流中并发送给终端仿真器。但是，该控制编码不会被“打印”到屏幕上，而是会被终端解释为一个指令。正如我们在上表看到的字符序列， 这个 \[ 和 \] 序列被用来封装这些非打印字符。一个 ANSI 转义编码以一个八进制033（这个编码是由 退出按键产生的）开头，其后跟着一个可选的字符属性（0：正常、1：黑体、4：下划线、5：闪烁、7：反向（前景色和背景色反转）），在之后是一个指令。

表14-2: 用转义序列来设置文本颜色

|  |  |  |  |
| --- | --- | --- | --- |
| 序列 | 文本颜色 | 序列 | 文本颜色 |
| \033[0;30m | 黑色 | \033[1;30m | **深灰色** |
| \033[0;31m | 红色 | \033[1;31m | **浅红色** |
| \033[0;32m | 绿色 | \033[1;32m | **浅绿色** |
| \033[0;33m | 棕色 | \033[1;33m | **黄色** |
| \033[0;34m | 蓝色 | \033[1;34m | **浅蓝色** |
| \033[0;35m | 粉红 | \033[1;35m | **浅粉色** |
| \033[0;36m | 青色 | \033[1;36m | **浅青色** |
| \033[0;37m | 浅灰色 | \033[1;37m | **白色** |

 例如我们来设置一个同LinuxMint默认的绿色提示符，

$ PS1="\[\033[01;32m\]\u@\W\$\[\033[00m\] "

于是，我们便有了下面的效果。

qwer@poiu#

上面跟设置的提示符格式中的最后那个”\[\033[00m\]”是用于将后续的字符颜色还原回原来的颜色。如果没有没有加最后这个转义码，则会出现下面结果（我们自己手工输入的命令也都将是绿色的）。

# 3.背景颜色

除了字体颜色，我们也可以设置字体的背景颜色。同样是通过转义的控制编码来实现，下表是背景颜色的控制编码。

表3: 用转义序列来设置背景颜色

|  |  |  |  |
| --- | --- | --- | --- |
| \033[0;40m | 黑色 | \033[1;44m | 蓝色 |
| \033[0;41m | 红色 | \033[1;45m | 粉红 |
| \033[0;42m | 绿色 | \033[1;46m | 青色 |
| \033[0;43m | 棕色 | \033[1;47m | 浅灰色 |

# 4.移动光标：

转义编码也可以用来定位光标。这些编码通常被用来，每次当提示符出现的时候，会在屏幕的不同位置，比如说上面一个角落，显示一个时钟或者其它一些信息。下表是一系列用来定位光标的转义编码：

表14-4: 光标移动转义序列

|  |  |
| --- | --- |
| 转义编码 | 行动 |
| \033[l;cH | 把光标移到第 l 行，第 c 列。 |
| \033[nA | 把光标向上移动 n 行。 |
| \033[nB | 把光标向下移动 n 行。 |
| \033[nC | 把光标向前移动 n 个字符。 |
| \033[nD | 把光标向后移动 n 个字符。 |
| \033[2J | 清空屏幕，把光标移到左上角（第零行，第零列）。 |
| \033[K | 清空从光标位置到当前行末的内容。 |
| \033[s | 存储当前光标位置。 |
| \033[u | 唤醒之前存储的光标位置。 |

# 5.保存提示符

我们在当前打开的Terminal中设置环境变量PS1的值来自定义的终端提示符，只在当前Terminal中有效，打开新的Terminal就会采用系统默认的提示符了。我们当让不希望每次打开Terminal的时候都要重新设置一次SP1环境变量。

~/.bashrc文件会在每次打开一个Terminal之前会被执行一次。于是，我们可以将设置环境变量SP1的命令放到该文件中，便不需要每次都手工再设置一次SP1的值。于是，我们在~/.bashrc配置文件末尾加入下面两行内容，

PS1="\[\033[01;32m\]\u@\W\$\[\033[00m\] "

export PS1

于是，每次打开Terminal便都会有如下提示符效果，

qwer@poiu~#

## 6.示例

（1）格式：“<用户名>@<当前目录名>$”

$ PS1="\u@\W\$ "

效果如下：

qwer@poiu#

（2）添加下划线：

$ PS1="\[\033[4;37m\]\u@\W\$\[\033[00m\] "

效果如下：

qwer@poiu~#

（3）前景色为绿色：

$ PS1="\[\033[0;32m\]\u@\W\$\[\033[00m\] "

效果如下：

qwer@poiu~#

（4）前景色为绿色，背景色为红色：

$ PS1="\[\033[0;41m\033[1;32m\]\u@\W\$\[\033[00m\] "

效果如下：

 qwer@poiu~#

（5）设置一个超级复杂的，通过移动光标来设置一个每次显示提示符的时候都在当前Terminal的第一行显示当前时间。

$ PS1='\[\033[s\033[0;0H\033[0;41m\033[K\033[1;33m\t\033[0m\033[u\033[1;32m\]\u@\W\$ \[\033[00m\] '