

linux下.bashrc文件修改和生效

转载 和光同尘ss 最后发布于2017-04-17 09:47:24 阅读数 48711 ☆ 收藏

1) .bashrc文件

在linux系统普通用户目录（cd /home/xxx）或root用户目录（cd /root）下，用指令ls -al可以看到4个隐藏文件，

.bash_history 记录之前输入的命令

.bash_logout 当你退出时执行的命令

.bash_profile 当你登入shell时执行

.bashrc 当你登入shell时执行

请注意后两个的区别：'.bash_profile'只在会话开始时被读取一次，而'.bashrc'则每次打开新的终端时，都要被读取。

这些文件是每一位用户对终端功能和属性设置，修改.bashrc可以改变环境变量PATH、别名alias和提示符。具体如何修改会在下面做介绍。

除了可以修改用户目录下的.bashrc文件外，还可以修改如“/etc/profile”文件、“/etc/bashrc”文件及目录“/etc /profile.d”下的文件。但是修改/etc路径下的配应用到整个系统，属于系统级的配置，而修改用户目录下的.bashrc则只是限制在用户应用上，属于用户级设置。两者在应用范围上有所区别，建议如需修改用户目录下的.bashrc，即无需root权限，也不会影响其他用户。

2) PATH环境变量修改

PATH变量决定了shell 将到哪些目录中寻找命令或程序。如果要执行的命令的目录在 \$PATH 中，您就不必输入这个命令的完整路径，直接输入命令就可。第三方软件没有将可执行文件放到 Linux 的标准目录中。因此，将这些非标准的安装目录添加到 \$PATH 是一种解决的办法。此外，您也将看到如何处理变量。

首先，作为惯例，所有环境变量名都是大写。由于 Linux 区分大小写，这点您要留意。当然，您可以自己定义一些变量，如'\$path'、'\$pAtH'，但 shell 不识别这些变量。

第二点是变量名有时候以'\$'开头，但有时又不是。当设置一个变量时，直接用名称，而不需要加"\$"，如

```
"PATH=/usr/bin:/usr/local/bin:/bin"
```

假如要获取变量值的话，就要在变量名前加'\$'：

```
"echo $PATH"
```

则会显示当前设置的PATH变量"/usr/bin:/usr/local/bin:/bin"

否则的话，变量名就会被当作普通文本了：

```
"echo PATH"
```

显示"PATH"

处理 \$PATH 变量要注意的第三点是：您不能只替换变量，而是要将新的字符串添加到原来的值中。在大多数情况下，您不能用"PATH=/some /directory"删除 \$PATH 中其他的所有目录，这样您在该终端运行程序时，就不得不给出完整路径。所以，只能作添加："PATH=\$PATH:/some/directory"，假如您要添加/usr/local/arm/3.4.1/bin交叉编译命令，则操作为"PATH=\$PATH:/usr/local/arm/3.4.1/bin"

这样，PATH 被设成当前的值（以 \$PATH 来表示）+ 新添的目录。

到目前为止，你只为当前终端设置了新的 \$PATH 变量。如果您打开一个新的终端，运行 echo \$PATH，将显示旧的 \$PATH 值，而看不到你刚才添加的为你先前定义的是一个局部环境变量（仅限于当前的终端）。

要定义一个全局变量，使在以后打开的终端中生效，您需要将局部变量输出(export)，可以用"export"命令：

```
1 | export PATH=$PATH:/some/directory
```

现在如果打开一个新的终端，输入 echo \$PATH，也能看到新设置的\$PATH了。请注意，命令'export'只能改变当前终端及以后运行的终端里的变量。对以前的终端没有作用。

为了将目录永久添加到 \$PATH，只要将"export"的那行添加到.bashrc或/etc/bashrc文件中。

3) alias别名

一般在.bashrc或/etc/bashrc文件里有几句话

```
alias rm='rm -i'
```



举报

```
alias cp='cp -i'
```

```
alias mv='mv -i'
```

有了这几句话，当在终端中输入“mv test.c led.c”实际上输入的是“mv -i test.c led.c”，所以说alias是一个别名。你可以在该配置文件中添加自己风格的别名“ls -l”，只需要在终端中输入“ll”就实现了“ls -l”的功能。还可以添加其他语句，随自己喜好。

4) 提示符

当打开一个控制台(console) 时，最先看到的就是提示符(prompt)，如：[root@localhost ~]#

在默认设置下，提示符将显示用户名、主机名（默认是'localhost'）、当前所在目录（在 Unix 中，'~'表示您的 home 目录）。

按照传统，最后一个字符可以标识普通用户（\$），还是'root'（#）。

可以通过 \$PS1 变量来设置提示符。

命令“echo \$PS1”，将显示当前的设定。其中可用字符的含义在 man bash 的'PROMPTING'部分有说明。

如何才能完成理想的设置呢？对于健忘的初学者来讲，默认设定有些不友好，因为提示符只显示当前目录的最后一部分。如果你看到象这样的提示符

```
[wsf@localhost bin]$
```

则当前目录可能是'/bin'、'/usr/bin'、'/usr/local/bin'及'/usr/X11R6/bin'。当然，你可以用

```
pwd （输出当前目录，print working directory）
```

能不能叫 shell 自动告诉你当前目录呢？

当然可以。这里我将提到的设定，包括提示符，大都包含在文件'/etc/bashrc'中。您可以通过编辑各自 home 目录下的'.bash_profile'和'.bashrc'来改变设

在 man bash 中的'PROMPTING'部分，对这些参数(parameter)有详细说明。您可以加入一些小玩意，如不同格式的当前时间，命令的历史记录号，甚

一种更适当的设定：

```
PS1="[u: \w]$ "
```

这样，提示符就变成：

```
[wsf: /usr/bin]$
```

您可以通过命令 export 来测试不同的设置（比如，export PS1="[u: \w]\$ "）。如果找到了适合的提示符，就将设置放到您的'.bashrc'中。这样，每次打

(5) 举例，生效方式

这个文件主要保存个人的一些个性化设置，如命令别名、路径等。下面是个例子：

```
1 | # User specific aliases and functions
2 |
3 | PATH="/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin"
4 |
5 | LANG=zh_CN.GBK
6 |
7 | export PATH LANG
8 |
9 | alias rm='rm -i'
10 |
11 | alias ls='/bin/ls -F --color=tty --show-control-chars'
```

例子中定义了路径，语言，命令别名（使用rm删除命令时总是加上-i参数需要

用户确认，使用ls命令列出文件列表时加上颜色显示）。

每次修改.bashrc后，使用source ~/.bashrc（或者 . ~/.bashrc）

就可以立刻加载修改后的设置，使之生效。

一般会在.bash_profile文件中显式调用.bashrc。登陆linux启动bash时首先会

去读取~/.bash_profile文件，这样~/.bashrc也就得到执行了，你的个性化设

置也就生效了。



举报