

Linux 命令行工具使用小技巧（四）

Linux 中的命令行别名

按照外行人的定义，别名可以被认为是一个复杂命令或者一组命令（包括它们的参数和选项）的简称或缩写。所以基本上，使用别名，你可以为那些不那么容易书写/记忆的命令创建易于记忆的名称。

例如，下面的命令为 `cd ~` 命令创建别名 `home`：

```
1. alias home="cd ~"
```

这意味着现在在你的系统中无论何地，无论何时你想要回到你的主目录时，你可以很快地输入 `home` 然后按回车键实现。

关于 `alias` 命令，`man` 手册是这么描述的：

`alias` 工具可以创建或者重定义别名定义，或者把现有别名定义输出到标准输出。别名定义提供了输入一个命令时应该被替换的字符串值

一个别名定义会影响当前 `shell` 的执行环境以及当前 `shell` 的所有子 `shell` 的执行环境。按照 **IEEE Std 1003.1-2001** 规定，别名定义不应该影响当前 `shell` 的父进程以及任何 `shell` 调用的程序环境。

那么，别名到底如何帮助命令行导航呢？这是一个简单的例子：

假设你正在 `/home/himanshu/projects/howtoforge` 目录工作，它包括很多子目录以及子子目录。例如下面就是一个完整的目录分支：

```
1. /home/himanshu/projects/howtoforge/command-  
   line/navigation/tips-tricks/part4/final
```

现在想象你在 `final` 目录，然后你想回到 `tips-tricks` 目录，然后再从那里，回到 `howtoforge` 目录。你会怎么做呢？

是的，一般情况下，你会运行下面的一组命令：

```
1. cd ../..  
2.  
3. cd ../../..
```

虽然这种方法并没有错误，但它绝对不方便，尤其是当你在一个很长的路径中想往回走例如说 5 个目录时。那么，有什么解决办法吗？答案就是：别名。

你可以做的是，为每个 `cd ../..` 命令创建容易记忆（和书写）的别名。例如：

```
1. alias bkl="cd ../.."
```

```
2. alias bk2="cd ../../"
3. alias bk3="cd ../../.."
4. alias bk4="cd ../../../../"
5. alias bk5="cd ../../../../../../"
```

现在无论你在什么时候想从当前工作目录往回走，例如说 5 个目录，你只需要运行下面的命令：

```
1. bk5
```

现在这不是很简单吗？

相关细节

尽管当前我们在 **shell** 中用于定义别名的技术（通过使用 **alias** 命令）实现了效果，别名只存在于当前终端会话。很有可能你会希望你定义的别名能保存下来，使得此后你可以在任何新启动的命令行窗口/标签页中使用它们。

为此，你需要在 `~/.bash_aliases` 文件中定义你的别名，你的 `~/.bashrc` 文件默认会加载该文件（如果你使用更早版本的 **Ubuntu**，我没有验证过是否有效）。

下面是我的 `.bashrc` 文件中关于 `.bash_aliases` 文件的部分：

```
1. # Alias definitions.
2. # You may want to put all your additions into a separate file
   like
3. # ~/.bash_aliases, instead of adding them here directly.
4. # See /usr/share/doc/bash-doc/examples in the bash-doc package.
5.
6. if [ -f ~/.bash_aliases ]; then
7. . ~/.bash_aliases
8. fi
```

一旦你把别名定义添加到你的 `.bash_aliases` 文件，该别名在任何新终端中都可用。但是，在任何其它你定义别名时已经启动的终端中，你还不能使用它们 - 解决办法是在这些终端中重新加载 `.bashrc`。下面就是你需要执行的具体命令：

```
1. source ~/.bashrc
```

如果你觉得这要做的也太多了（是的，我期待你有更懒惰的办法），那么这里有一个快捷方式来做到这一切：

```
1. "alias [the-alias]" >> ~/.bash_aliases && source
   ~/.bash_aliases
```

毫无疑问，你需要用实际的命令替换 `[the-alias]`。例如：

```
1. "alias bk5='cd ../../../../..'" >> ~/.bash_aliases && source
   ~/.bash_aliases
```

接下来，假设你已经创建了一些别名，并时不时使用它们有一段时间了。突然有一天，你发现它们其中的一个并不像期望的那样。因此你觉得需要查看被赋予该别名的真正命令。你会怎么做呢？

当然，你可以打开你的 `.bash_aliases` 文件在那里看看，但这种方式可能有点费时，尤其是当文件中包括很多别名的时候。因此，如果你正在查找一种更简单的方式，这就有一个：你需要做的只是运行 `alias` 命令并把别名名称作为参数。

这里有个例子：

```
1. $ alias bk6
2. alias bk6='cd ../../../../..'
```

你可以看到，上面提到的命令显示了被赋值给别名 `bk6` 的实际命令。这里还有另一种办法：使用 `type` 命令。下面是一个例子：

```
1. $ type bk6
2. bk6 is aliased to `cd ../../../../..'
```

`type` 命令产生了一个易于人类理解的输出。

另一个值得分享的是你可以将别名用于常见的输入错误。例如：

```
1. alias mroe='more'
```

最后，还值得注意的是并非每个人都喜欢使用别名。他们中的大部分人认为一旦你习惯了你为了简便而定义的别名，当你在其它相同而不存在别名（而且不允许你创建）的系统中工作时就会变得非常困难。更多（也是更准确的）为什么一些专家不推荐使用别名的原因，你到[这里](#)查看。

总结

就像我们之前文章讨论过的 `CDPATH` 环境变量，别名也是一把应该谨慎使用的双刃剑。尽管如此也别太丧气，因为每个东西都有它自己的好处和劣势。遇到类似别名的概念时，更多的练习和完备的知识才是重点。

那么这就是该系列指南的最后章节。希望你喜欢它并能从中学到新的东西/概念。如果你有任何疑问或者问题，请在下面的评论框中和我们（以及其他的人）分享。

via: <https://www.howtoforge.com/tutorial/command-line-aliases-in-linux/>

作者: [Ansh](#) 译者: [ictlyh](#) 校对: [wxy](#)

