

## Linux 命令行工具使用小技巧（二）

### pushd 和 popd 命令基础

为了更好地理解 pushd 和 popd 命令的作用，让我们先讨论堆栈的概念。想象你厨房案板上有一个空白区域，你想在上面放一套盘子。你会怎么做？很简单，一个接一个地放在上面。所以在整个过程的最后，案板上的第一个盘子是盘子中的最后一个，你手中最后一个盘子是盘子堆中的第一个。现在当你需要一个盘子时，你选择在堆的顶部的那个盘子并使用它，然后需要时选择下一个。

pushd 和 popd 命令是类似的概念。在 Linux 系统上有一个目录堆栈，你可以堆叠目录路径以供将来使用。你可以使用 dirs 命令来在任何时间点快速查看堆栈的内容。

下面的例子显示了在命令行终端启动后立即在我的系统上使用 dirs 命令的输出：

```
$ dirs
```

```
~
```

输出中的波浪号（~）表示目录堆栈当前仅包含用户的主目录。

继续下去，使用 pushd 和 popd 命令来执行存储目录路径并删除它的操作。使用 pushd 非常容易 - 只需将要存储在目录堆栈中的路径作为此命令的参数传递。这里有一个例子：

```
pushd /home/himanshu/Downloads/
```

上述命令的作用是，将当前工作目录更改为你作为参数传递的目录，并且还将路径添加到目录堆栈中。为了方便用户，pushd 命令在其输出中产生目录堆栈的内容。因此，当运行上面的命令时，产生了以下输出：

```
~/Downloads ~
```

输出显示现在堆栈中有两个目录路径：一个是用户的主目录，还有用户的下载目录。它们的保存顺序是：主目录位于底部，新添加的 Downloads 目录位于其上。

要验证 pushd 的输出是正确的，你还可以使用 dirs 命令：

```
$ dirs
```

```
~/Downloads ~
```

因此你可以看到 dirs 命令同样产生相同的输出。

让我们再使用下 pushd 命令：

```
$ pushd /usr/lib/; pushd /home/himanshu/Desktop/
```

```
/usr/lib ~/Downloads ~
```

```
~/Desktop /usr/lib ~/Downloads ~
```

所以目录堆栈现在包含总共四个目录路径，其中主目录（~）在底部，并且用户的桌面目录在顶部。

一定要记住的是堆栈的头是你当前的目录。这意味着现在我们当前的工作目录是

```
~/Desktop。
```

现在，假设你想回到 /usr/lib 目录，所以你所要做的就是执行 popd 命令：

```
$ popd
```

```
/usr/lib ~/Downloads ~
```

popd 命令不仅会将当前目录切换到 /usr/lib，它还会从目录堆栈中删除 ~/Desktop，这一点可以从命令输出中看出。这样，popd 命令将允许你以相反的顺序浏览这些目录。

### 一些高级用法

现在我们已经讨论了 `pushd` 和 `popd` 命令的基础知识，让我们继续讨论与这些命令相关的一些其它细节。首先，这些命令还允许你操作目录堆栈。例如，假设你的目录堆栈看起来像这样：

```
$ dirs
```

```
~/Desktop /usr/lib ~ ~/Downloads
```

现在，我们的要求是改变堆栈中目录路径的顺序，最上面的元素（`~/Desktop`）放到底部，剩下的每个都向上移动一个位置。这可以使用以下命令实现：

```
pushd +1
```

上面的命令对目录堆栈做的结果：

```
$ dirs
```

```
/usr/lib ~ ~/Downloads ~/Desktop
```

因此，我们看到目录堆栈中的元素顺序已经改变，并且现在和我们想要的一样。当然，你可以让目录堆栈元素移动任何次数。例如，以下命令会将它们向上移动两次：

```
$ pushd +2
```

```
~/Downloads ~/Desktop /usr/lib ~
```

你也可以使用负的索引值：

```
$ pushd -1
```

```
/usr/lib ~ ~/Downloads ~/Desktop
```

相似地，你可以在 `popd` 命令中使用此技术来从目录堆栈删除任何条目，而不用离开当前工作目录。例如，如果要使用 `popd` 从顶部（目前是 `~/Downloads`）删除第三个条目，你可以运行以下命令：

```
popd +2
```

记住堆栈索引的初始值是 0，因此我们使用 2 来访问第三个条目。

因此目录堆栈现在包含：

```
$ dirs
```

```
/usr/lib ~ ~/Desktop
```

确认条目已经被移除了。

如果由于某些原因，你发现你很难记住元素在目录堆栈中的位置以及它们的索引，你则可以对在 `dirs` 命令中使用 `-v` 选项。这里有一个例子：

```
$ dirs -v
```

```
0 /usr/lib
```

```
1 ~
```

```
2 ~/Desktop
```

你可能已经猜到了，左边的数字是索引，接下来跟的是这个索引对应的目录路径。

**注意：** 在 `dir` 中使用 `-c` 选项清除目录堆栈。

现在让我们简要地讨论一下 `popd` 和 `pushd` 命令的实际用法。虽然它们第一眼看起来可能有点复杂，但是这些命令在编写 **shell** 脚本时会派上用场 - 你不需要记住你从哪里来；只要执行一下 `popd`，你就能回到你来的目录。

经验丰富的脚本编写者通常以以下方式使用这些命令：

```
popd >/dev/null 2>&1
```

上述命令确保 `popd` 保持静默（不产生任何输出）。同样，你也可以静默 `pushd`。  
`pushd` 和 `popd` 命令也被 **Linux** 服务器管理员使用，他们通常在几个相同的目录之间移动。在[这里](#)介绍了一些其他有用的使用场景。

## 总结

我同意 `pushd` 和 `popd` 的概念不是很直接。但是，它需要的只是一点练习 - 是的，你需要多实践。花一些时间在这些命令上，你就会开始喜欢它们，特别是当它们提供了方便时。

---

via: <https://www.howtoforge.com/tutorial/linux-command-line-tips-tricks-part-2/>