

目录堆栈

网道 (WangDoc.com) , 互联网文档计划

为了方便用户在不同目录之间切换, Bash 提供了目录堆栈功能。

目录 [隐藏]

1. `cd -`
2. `pushd`, `popd`
3. `dirs` 命令

1. `cd -`

Bash 可以记忆用户进入过的目录。默认情况下, 只记忆前一次所在的目录, `cd -` 命令可以返回前一次的目录。

```
# 当前目录是 /path/to/foo
$ cd bar
```

```
# 重新回到 /path/to/foo
$ cd -
```

上面例子中, 用户原来所在的目录是 `/path/to/foo`, 进入子目录 `bar` 以后, 使用 `cd -` 可以回到原来的目录。

📄 Bash 脚本教程

- 📄 1. 简介
- 📄 2. 基本语法
- 📄 3. 模式扩展
- 📄 4. 引号和转义
- 📄 5. 变量
- 📄 6. 字符串操作
- 📄 7. 算术运算
- 📄 8. 操作历史
- 📄 9. 行操作
- 📄 10. 目录堆栈
- 📄 11. 脚本入门
- 📄 12. `read` 命令
- 📄 13. 条件判断
- 📄 14. 循环
- 📄 15. 函数
- 📄 16. 数组
- 📄 17. `set` 命令, `shopt` 命令

2. pushd, popd

如果希望记忆多重目录，可以使用 `pushd` 命令和 `popd` 命令。它们用来操作目录堆栈。

`pushd` 命令的用法类似 `cd` 命令，可以进入指定的目录。

```
$ pushd dirname
```

上面命令会进入目录 `dirname`，并将该目录放入堆栈。

第一次使用 `pushd` 命令时，会将当前目录先放入堆栈，然后将所要进入的目录也放入堆栈，位置在前一个记录的上方。以后每次使用 `pushd` 命令，都会将所要进入的目录，放在堆栈的顶部。

`popd` 命令不带有参数时，会移除堆栈的顶部记录，并进入新的栈顶目录（即原来的第二条目录）。

下面是一个例子。

```
# 当前处在主目录，堆栈为空
$ pwd
/home/me

# 进入 /home/me/foo
# 当前堆栈为 /home/me/foo /home/me
$ pushd ~/foo

# 进入 /etc
# 当前堆栈为 /etc /home/me/foo /home/me
$ pushd /etc

# 进入 /home/me/foo
# 当前堆栈为 /home/me/foo /home/me
$ popd

# 进入 /home/me
# 当前堆栈为 /home/me
$ popd
```

📄 18. 脚本除错

📄 19. mktemp 命令, trap 命令

📄 20. 启动环境

📄 21. 命令提示符

🔗 链接

📄 本文源码

📄 代码仓库

📄 反馈

```
# 目录不变，当前堆栈为空
$ popd
```

这两个命令的参数如下。

(1) -n 参数

`-n` 的参数表示仅操作堆栈，不改变目录。

```
$ popd -n
```

上面的命令仅删除堆栈顶部的记录，不改变目录，执行完成后还停留在当前目录。

(2) 整数参数

这两个命令还可以接受一个整数作为参数，该整数表示堆栈中指定位置的记录（从0开始）。`pushd` 命令会把这条记录移动到栈顶，同时切换到该目录；`popd` 则从堆栈中删除这条记录，不会切换目录。

```
# 将从栈顶算起的3号目录（从0开始）移动到栈顶，同时切换到i
$ pushd +3
```

```
# 将从栈底算起的3号目录（从0开始）移动到栈顶，同时切换到i
$ pushd -3
```

```
# 删除从栈顶算起的3号目录（从0开始），不改变当前目录
$ popd +3
```

```
# 删除从栈底算起的3号目录（从0开始），不改变当前目录
$ popd -3
```



上面例子的整数编号都是从0开始计算，`popd +0` 是删除第一个目录，`popd +1` 是删除第二个，`popd -0` 是删除最后一个目录，`popd -1` 是删除倒数第二个。

(3) 目录参数

`pushd` 可以接受一个目录作为参数，表示将该目录放到堆栈顶部，并进入该目录。

```
$ pushd dir
```

`popd` 没有这个参数。

3. dirs 命令

`dirs` 命令可以显示目录堆栈的内容，一般用来查看 `pushd` 和 `popd` 操作后的结果。

```
$ dirs
~/foo/bar ~/foo ~
```

该命令会输出一行文本，列出目录堆栈，目录之间使用空格分隔。栈顶（最晚入栈的目录）在最左边，栈底（最早入栈的目录）在最右边。

它有如下参数。

- `-c`：清空目录栈。
- `-l`：用户主目录不显示波浪号前缀，而打印完整的目录。
- `-p`：每行一个条目打印目录栈，默认是打印在一行。
- `-v`：每行一个条目，每个条目之前显示位置编号（从0开始）。
- `+N`：`N` 为整数，表示显示堆顶算起的第 `N` 个目录，从零开始。
- `-N`：`N` 为整数，表示显示堆底算起的第 `N` 个目录，从零开始。

[📄 行操作](#)

[脚本入门 📄](#)

本教程采用[知识共享 署名-相同方式共享 3.0协议](#)。

分享本文



1条评论

 登录 ▼

加入讨论...

通过以下方式登录

或注册一个 **DISQUS** 帐号 

姓名

♡ 1

分享

最佳

最新

最早



YourLi

1 年前



...试了一下，应该说 栈顶始终是当前目录吧！

比如 `popd -n`

```
yourli@LAPTOP-13BPD4ET:/home$ dirs -l
/home /etc /sys /home/yourli
yourli@LAPTOP-13BPD4ET:/home$ cd /bin
yourli@LAPTOP-13BPD4ET:/bin$ dirs -l
/bin /etc /sys /home/yourli
yourli@LAPTOP-13BPD4ET:/bin$ popd -n
/bin /sys ~
yourli@LAPTOP-13BPD4ET:/bin$ dirs -l
/bin /sys /home/yourli
```

实际流程应该是 将 `/bin` 移除，此时 `/etc` 为栈顶，但是栈顶始终指向当前目录，所以修改成 `/bin`

结果就好像把栈顶后一个目录删除一样

o o 回复 

订阅

隐私

不要出售我的数据

联系: contact@wangdoc.com