

Bash 简介

网道 (WangDoc.com) , 互联网文档计划

Bash 是 Unix 系统和 Linux 系统的一种 Shell (命令行环境) , 是目前绝大多数 Linux 发行版的默认 Shell。

目录 [隐藏]

- 1. Shell 的含义
- 2. Shell 的种类
- 3. 命令行环境
 - 3.1 终端模拟器
 - 3.2 命令行提示符
 - 3.3 进入和退出方法
- 4. Shell 和 Bash 的历史

1. Shell 的含义

学习 Bash, 首先需要理解 Shell 是什么。Shell 这个单词的原意是“外壳”, 跟 kernel (内核) 相对应, 比喻内核外面的一层, 即用户跟内核交互的对话界面。

具体来说, Shell 这个词有多种含义。

首先, Shell 是一个程序, 提供一个与用户对话的环境。这个环境只有一个命令提示符, 让用户从键盘输入命令, 所以又称为命令行环境 (command line interface, 简称为 CLI) 。Shell 接收到用户输入的命令, 将命令送入操作系统

📄 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

执行，并将结果返回给用户。本书中，除非特别指明，Shell 指的就是命令行环境。

其次，Shell 是一个命令解释器，解释用户输入的命令。它支持变量、条件判断、循环操作等语法，所以用户可以用 Shell 命令写出各种小程序，又称为脚本（script）。这些脚本都通过 Shell 的解释执行，而不通过编译。

最后，Shell 是一个工具箱，提供了各种小工具，供用户方便地使用操作系统的功能。

2. Shell 的种类

Shell 有很多种，只要能给用户提供了命令行环境的程序，都可以看作是 Shell。

历史上，主要的 Shell 有下面这些。

- Bourne Shell (sh)
- Bourne Again shell (bash)
- C Shell (csh)
- TENEX C Shell (tcsh)
- Korn shell (ksh)
- Z Shell (zsh)
- Friendly Interactive Shell (fish)

Bash 是目前最常用的 Shell，除非特别指明，下文的 Shell 和 Bash 当作同义词使用，可以互换。

下面的命令可以查看当前设备的默认 Shell。

```
$ echo $SHELL
/bin/bash
```

当前正在使用的 Shell 不一定是默认 Shell，一般来说，`ps` 命令结果的倒数第二行是当前 Shell。

```
$ ps
PID TTY          TIME CMD
```

📖 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

```
4467 pts/0    00:00:00 bash
5379 pts/0    00:00:00 ps
```

上面示例中，`ps` 命令结果的倒数第二行显示，运行的命令（`cmd`）是 `bash`，表明当前正在使用的 Shell 是 Bash。

下面的命令可以查看当前的 Linux 系统安装的所有 Shell。

```
$ cat /etc/shells
```

上面三个命令中，`$` 是命令行环境的提示符，用户只需要输入提示符后面的内容。

Linux 允许每个用户使用不同的 Shell，用户的默认 Shell 一般都是 Bash，或者与 Bash 兼容。

3. 命令行环境

3.1 终端模拟器

如果是不带有图形环境的 Linux 系统（比如专用于服务器的系统），启动后就直接是命令行环境。

不过，现在大部分的 Linux 发行版，尤其是针对普通用户的发行版，都是图形环境。用户登录系统后，自动进入图形环境，需要自己启动终端模拟器，才能进入命令行环境。

所谓“终端模拟器”（terminal emulator）就是一个模拟命令行窗口的程序，让用户在一个窗口中使用命令行环境，并且提供各种附加功能，比如调整颜色、字体大小、行距等等。

不同 Linux 发行版（准确地说是不同的桌面环境）带有的终端程序是不一样的，比如 KDE 桌面环境的终端程序是 `konsole`，Gnome 桌面环境的终端程序是 `gnome-terminal`，用户也可以安装第三方的终端程序。所有终端程序，尽管名字不同，基本功能都是一样的，就是让用户可以进入命令行环境，使用 Shell。

3.2 命令行提示符

进入命令行环境以后，用户会看到 Shell 的提示符。提示符往往是一串前缀，最后以一个美元符号 `$` 结尾，用户可以在这个符号后面输入各种命令。

```
[user@hostname] $
```

上面例子中，完整的提示符是 `[user@hostname] $`，其中前缀是用户名（`user`）加上 `@`，再加主机名（`hostname`）。比如，用户名是 `bill`，主机名是 `home-machine`，前缀就是 `bill@home-machine`。

注意，根用户（`root`）的提示符，不以美元符号（`$`）结尾，而以井号（`#`）结尾，用来提醒用户，现在具有根权限，可以执行各种操作，务必小心，不要出现误操作。这个符号是可以自己定义的，详见《命令提示符》一章。

为了简洁，后文的命令行提示符都只使用 `$` 表示。

3.3 进入和退出方法

进入命令行环境以后，一般就已经打开 Bash 了。如果你的 Shell 不是 Bash，可以输入 `bash` 命令启动 Bash。

```
$ bash
```

退出 Bash 环境，可以使用 `exit` 命令，也可以同时按下 `Ctrl + d`。

```
$ exit
```

Bash 的基本用法就是在命令行输入各种命令，非常直观。作为练习，可以试着输入 `pwd` 命令。按下回车键，就会显示当前所在的目录。

```
$ pwd  
/home/me
```

如果不小心输入了 `pwe`，会返回一个提示，表示输入出错，没有对应的可执行程序。

```
$ pwe
bash: pwe: 未找到命令
```

4. Shell 和 Bash 的历史

Shell 伴随着 Unix 系统的诞生而诞生。

1969年，Ken Thompson 和 Dennis Ritchie 开发了第一版的 Unix。

1971年，Ken Thompson 编写了最初的 Shell，称为 Thompson shell，程序名是 `sh`，方便用户使用 Unix。

1973年至1975年间，John R. Mashey 扩展了最初的 Thompson shell，添加了编程功能，使得 Shell 成为一种编程语言。这个版本的 Shell 称为 Mashey shell。

1976年，Stephen Bourne 结合 Mashey shell 的功能，重写一个新的 Shell，称为 Bourne shell。

1978年，加州大学伯克利分校的 Bill Joy 开发了 C shell，为 Shell 提供 C 语言的语法，程序名是 `cs`。它是第一个真正替代 `sh` 的 UNIX shell，被合并到 Berkeley UNIX 的 2BSD 版本中。

1979年，UNIX 第七版发布，内置了 Bourne Shell，导致它成为 Unix 的默认 Shell。注意，Thompson shell、Mashey shell 和 Bourne shell 都是贝尔实验室的产品，程序名都是 `sh`。对于用户来说，它们是同一个东西，只是底层代码不同而已。

1983年，David Korn 开发了 Korn shell，程序名是 `ksh`。

1985年，Richard Stallman 成立了自由软件基金会（FSF），由于 Shell 的版权属于贝尔公司，所以他决定写一个自由版权的、使用 GNU 许可证的 Shell 程序，避免 Unix 的版权争议。

1988年，自由软件基金会的第一个付薪程序员 Brian Fox 写了一个 Shell，功能基本上是 Bourne shell 的克隆，叫做 Bourne-Again SHell，简称 Bash，程序名为 `bash`，任何人都可以免费使用。后来，它逐渐成为 Linux 系统的标准 Shell。

1989年，Bash 发布1.0版。

1996年，Bash 发布2.0版。

2004年，Bash 发布3.0版。

2009年，Bash 发布4.0版。

2019年，Bash 发布5.0版。

用户可以通过 `bash` 命令的 `--version` 参数或者环境变量 `$BASH_VERSION`，查看本机的 Bash 版本。

```
$ bash --version
GNU bash, 版本 5.0.3(1)-release (x86_64-pc-linux-gn

# 或者
$ echo $BASH_VERSION
5.0.3(1)-release
```



[🏠 Home](#)

[基本语法 ▶](#)

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



4

分享

最佳

最新

最早

陳

陳先森

6个月前

太赞了!!!

o

o

回复



订阅

隐私

不要出售我的数据

联系: contact@wangdoc.com