

Bash 启动环境

网道 (WangDoc.com) , 互联网文档计划

目录 [隐藏]

- 1. Session
 - 1.1 登录 Session
 - 1.2 非登录 Session
 - 1.3 .bash_logout
- 2. 启动选项
- 3. 键盘绑定

1. Session

用户每次使用 Shell, 都会开启一个与 Shell 的 Session (对话)。

Session 有两种类型: 登录 Session 和非登录 Session, 也可以叫做 login shell 和 non-login shell。

1.1 登录 Session

登录 Session 是用户登录系统以后, 系统为用户开启的原始 Session, 通常需要用户输入用户名和密码进行登录。

登录 Session 一般进行整个系统环境的初始化, 启动的初始化脚本依次如下。

📖 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

- `/etc/profile` : 所有用户的全局配置脚本。
- `/etc/profile.d` 目录里面所有 `.sh` 文件
- `~/.bash_profile` : 用户的个人配置脚本。如果该脚本存在, 则执行完就不再往下执行。
- `~/.bash_login` : 如果 `~/.bash_profile` 没找到, 则尝试执行这个脚本 (C shell 的初始化脚本)。如果该脚本存在, 则执行完就不再往下执行。
- `~/.profile` : 如果 `~/.bash_profile` 和 `~/.bash_login` 都没找到, 则尝试读取这个脚本 (Bourne shell 和 Korn shell 的初始化脚本)。

Linux 发行版更新的时候, 会更新 `/etc` 里面的文件, 比如 `/etc/profile`, 因此不要直接修改这个文件。如果想修改所有用户的登陆环境, 就在 `/etc/profile.d` 目录里面新建 `.sh` 脚本。

如果想修改你个人的登录环境, 一般是写在

`~/.bash_profile` 里面。下面是一个典型的 `.bash_profile` 文件。

```
# .bash_profile
PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/bin
PATH=$PATH:$HOME/bin

SHELL=/bin/bash
MANPATH=/usr/man:/usr/X11/man
EDITOR=/usr/bin/vi
PS1='\h:\w\$ '
PS2='> '

if [ -f ~/.bashrc ]; then
. ~/.bashrc
fi

export PATH
export EDITOR
```

可以看到, 这个脚本定义了一些最基本的环境变量, 然后执行了 `~/.bashrc`。

📖 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

`bash` 命令的 `--login` 参数，会强制执行登录 Session 会执行的脚本。

```
$ bash --login
```

`bash` 命令的 `--noprofile` 参数，会跳过上面这些 Profile 脚本。

```
$ bash --noprofile
```

1.2 非登录 Session

非登录 Session 是用户进入系统以后，手动新建的 Session，这时不会进行环境初始化。比如，在命令行执行 `bash` 命令，就会新建一个非登录 Session。

非登录 Session 的初始化脚本依次如下。

- `/etc/bash.bashrc`：对全体用户有效。
- `~/.bashrc`：仅对当前用户有效。

对用户来说，`~/.bashrc` 通常是最重要的脚本。非登录 Session 默认会执行它，而登录 Session 一般也会通过调用执行它。每次新建一个 Bash 窗口，就相当于新建一个非登录 Session，所以 `~/.bashrc` 每次都会执行。注意，执行脚本相当于新建一个非互动的 Bash 环境，但是这种情况不会调用 `~/.bashrc`。

`bash` 命令的 `--norc` 参数，可以禁止在非登录 Session 执行 `~/.bashrc` 脚本。

```
$ bash --norc
```

`bash` 命令的 `--rcfile` 参数，指定另一个脚本代替 `.bashrc`。

```
$ bash --rcfile testrc
```

1.3 .bash_logout

`~/.bash_logout` 脚本在每次退出 Session 时执行，通常用来做一些清理工作和记录工作，比如删除临时文件，记录用户在本次 Session 花费的时间。

如果没有退出时要执行的命令，这个文件也可以不存在。

2. 启动选项

为了方便 Debug，有时在启动 Bash 的时候，可以加上启动参数。

- `-n`：不运行脚本，只检查是否有语法错误。
- `-v`：输出每一行语句运行结果前，会先输出该行语句。
- `-x`：每一个命令处理之前，先输出该命令，再执行该命令。

```
$ bash -n scriptname
```

```
$ bash -v scriptname
```

```
$ bash -x scriptname
```

3. 键盘绑定

Bash 允许用户定义自己的快捷键。全局的键盘绑定文件默认为 `/etc/inputrc`，你可以在主目录创建自己的键盘绑定文件 `.inputrc` 文件。如果定义了这个文件，需要在其中加入下面这行，保证全局绑定不会被遗漏。

```
$include /etc/inputrc
```

`.inputrc` 文件里面的快捷键，可以像这样定义，`"\C-t":"pwd\n"` 表示将 `Ctrl + t` 绑定为运行 `pwd` 命令。

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      

联系: contact@wangdoc.com