

第四部分 基础shell编程

第16章 shell脚本介绍

一个shell脚本可以包含一个或多个命令。当然可以不必只为了两个命令就编写一个 shell 脚本，一切由用户自己决定。

本章内容有：

- 使用shell脚本的原因。
- shell脚本基本元素。
- shell脚本运行方式。

16.1 使用shell脚本的原因

shell脚本在处理自动循环或大的任务方面可节省大量的时间，且功能强大。如果你有处理一个任务的命令清单，不得不一个一个敲进去，然后观察输出结果，再决定它是否正确，如果正确，再继续下一个任务，否则再回到清单一步步观察。一个任务可能是将文件分类、向文件插入文本、迁移文件、从文件中删除行、清除系统过期文件、以及系统一般的管理维护工作等等。创建一个脚本，在使用一系列系统命令的同时，可以使用变量、条件、算术和循环快速创建脚本以完成相应工作。这比在命令行下一个个敲入要节省大量的工作时间。

shell脚本可以在行命令中接收信息，并使用它作为另一个命令的输入。

对于不同的UNIX和LINUX，使用一段shell脚本将需要一些小小的改动才能运行通过。实际上shell的可迁移性不成问题，但是系统间命令的可迁移性存在差别。

试试新思路

如果写一段脚本，其执行结果与预想的不同，不必着急。无论多不可思议的结果，记住先把它保存起来，这是修改的基础。这里要说的意思是不要害怕对待新事物，否则将不能树立信心，学起来会更加困难。

16.2 脚本内容

本章不讲怎样设计精巧的脚本，而是怎样使脚本重复利用率高。当通过一些易理解的脚本就可实现同样功能时，没有必要使脚本复杂化。如果作者要写这样一本书，可能会给你留下深刻印象，但这要花费更多的时间和精力去研读和体会脚本。这不是本书的目标。本书脚本流程仅使用基本的脚本技术，十分容易学，然后使用者就可以着手实践了。

脚本不是复杂的程序，它是按行解释的。脚本第一行总是以 `#!/bin/sh` 开始，这段脚本通知shell使用系统上的Bourne shell解释器。

任何脚本都可能有关注，加注释需要此行的第一个字符为 `#`，解释器对此行不予解释。在

第二行注释中写入脚本名是一个好习惯。

脚本从上到下执行，运行脚本前需要增加其执行权限。确保正确建立脚本路径，这样只用文件名就可以运行它了。

16.3 运行一段脚本

下面是一个已经讨论过的例子，此文件为 cleanup。

```
$ pg cleanup
#!/bin/sh
# name: cleanup
# this is a general cleanup script
echo "starting cleanup...wait"
rm /usr/local/apps/log/*.log
tail -40 /var/adm/messages >/tmp/messages
rm /var/adm/messages
mv /tmp/messages /var/adm/messages
echo "finished cleanup"
```

上述脚本通过将目录下文件名截断，清除 /usr/adm/下信息，并删除 /usr/local/apps/log下所有注册信息。

可以使用 chmod 命令增加脚本执行权限。

```
$ chmod u+x cleanup
```

现在运行脚本，只敲入文件名即可。

```
$ cleanup
```

如果返回错误信息：

```
$ cleanup
sh:cleanup:command not found
```

再试：

```
$. /cleanup
```

如果脚本运行前必须键入路径名，或者 shell 结果通知无法找到命令，就需要在 .profile PATH 下加入用户可执行程序目录。要确保用户在自己的 \$HOME 可执行程序目录下，应键入：

```
$ pwd
$ /home/dave/bin
```

如果 pwd 命令最后一部分是 bin，那么需要在路径中加入此信息。编辑用户 .profile 文件，加入可执行程序目录 \$HOME/bin 如下：

```
PATH=$PATH:$HOME/bin
```

如果没有 bin 目录，就创建它。首先确保在用户根目录下。

```
$ cd $HOME
$ mkdir bin
```

现在可以在 .profile 文件中将 bin 目录加入 PATH 变量了，然后重新初始化 .profile。

```
$. ./profile
```

脚本将会正常运行。

如果还有问题，见第 2 章和第 13 章，那里详细介绍了如何解决这一问题。

全书有许多脚本清单，这些脚本都是完整的。将这些脚本输入文件，保存并退出，再使

用chmod命令增加其执行权限，这些脚本就可以实际操作了。

16.4 小结

本章介绍了shell脚本的基本原理，相信关于脚本的功能原理这些已经足够了，读本章时可加快速度。本章目标只是要用户知道运行shell脚本需要做些什么。