Linux 中高效编写 Bash 脚本的 10 个技巧

内容

Shell 脚本编程 是你在 Linux 下学习或练习编程的最简单的方式。尤其对 系统管理员要处理着自动化任务，且要开发新的简单的实用程序或工具等（这里只是仅举几例）更是必备技能。

本文中，我们将分享 10 个写出高效可靠的 bash 脚本的实用技巧，它们包括：

1、 脚本中多写注释

这是不仅可应用于 shell 脚本程序中，也可用在其他所有类型的编程中的一种推荐做法。在脚本中作注释能帮你或别人翻阅你的脚本时了解脚本的不同部分所做的工作。

对于刚入门的人来说，注释用 # 号来定义。

```
# TecMint 是浏览各类 Linux 文章的最佳站点
```

2、 当运行失败时使脚本退出

有时即使某些命令运行失败，bash 可能继续去执行脚本，这样就影响到脚本的其余部分（会最终导致逻辑错误）。用下面的行的方式在遇到命令失败时来退出脚本执行：

```
# 如果命令运行失败让脚本退出执行
set -o errexit
# 或
set -e
```

3、 当 Bash 用未声明变量时使脚本退出

Bash 也可能会使用能导致起逻辑错误的未声明的变量。因此用下面行的方式去通知 bash 当它尝试去用一个未声明变量时就退出脚本执行：

```
# 若有用未设置的变量即让脚本退出执行
set -o nounset
# 或
set -u
```

4、 使用双引号来引用变量

当引用时（使用一个变量的值）用双引号有助于防止由于空格导致单词分割开和由于识别和扩展了通配符而导致的不必要匹配。

看看下面的例子：

```
#!/bin/bash
# 若命令失败让脚本退出
set -o errexit
# 若未设置的变量被使用让脚本退出
set -o nounset
echo "Names without double quotes"
echo

names="Tecmint FOSSMint Linusay"

for name in $names; do
echo "$name"
done

echo
echo "Names with double quotes"
echo

for name in "$names"; do
echo "$name"
done

exit 0
```

保存文件并退出，接着如下运行一下：

```
$ ./names.sh
```

在脚本中用双引号

5、 在脚本中使用函数

除了非常小的脚本（只有几行代码），总是记得用函数来使代码模块化且使得脚本更可读和可重用。

写函数的语法如下所示：

```
function check_root(){
command1;
command2;
}
# 或
check_root(){
command1;
command2;
}
```

写成单行代码时，每个命令后要用终止符号：

```
check_root(){ command1; command2; }
```

6、 字符串比较时用 = 而不是 ==

注意 == 是 = 的同义词，因此仅用个单 = 来做字符串比较，例如：

```
value1="tecmint.com"
value2="fossmint.com"
if [ "$value1" = "$value2" ]
```

7、 用 $(command) 而不是老旧的 `command` 来做代换

命令代换 是用这个命令的输出结果取代命令本身。用 $(command) 而不是引号 `command` 来做命令代换。

这种做法也是 shellcheck tool （可针对 shell 脚本显示警告和建议）所建议的。例如：

```
user=`echo "$UID"`
user=$(echo "$UID")
```

8、 用 readonly 来声明静态变量

静态变量不会改变；它的值一旦在脚本中定义后不能被修改：

```
readonly passwd_file="/etc/passwd"
readonly group_file="/etc/group"
```

9、 环境变量用大写字母命名，而自定义变量用小写

所有的 bash 环境变量用大写字母去命名，因此用小写字母来命名你的自定义变量以避免变量名冲突：

```
# 定义自定义变量用小写，而环境变量用大写
nikto_file="$HOME/Downloads/nikto-master/program/nikto.pl"
perl "$nikto_file" -h "$1"
```

10、 总是对长脚本进行调试

如果你在写有数千行代码的 bash 脚本，排错可能变成噩梦。为了在脚本执行前易于修正一些错误，要进行一些调试。通过阅读下面给出的指南来掌握此技巧：

如何在 Linux 中启用 Shell 脚本调试模式
如何在 Shell 脚本中执行语法检查调试模式
如何在 Shell 脚本中跟踪调试命令的执行