

酷壳 – CoolShell.cn [1]

如何调试bash 脚本

2009年8月31日 陈皓 [2] 发表评论 阅读评论 8,356 人阅读

是Linux操作系统的默认Shell脚本。Shell是用来处理操作系统和用户交互的一个程序。Shell的脚本可以帮助用户自动化地和操作系统进行交互。你也可以理解为一种脚本式的编程。即然有编程，那么，程序的编译器，解释器，调试器就必不可少，Bash也一样，但在调试方面可能会有一些和编程语言不一样的东西和技术，所以，下面这篇文章主要是说明调试bash脚本的各种技术。

跟踪脚本的执行

你可以让bash打印出你脚本执行的过程中的所有语句。这很简单，只需要使用bash的-x选项就可以做到，下面让我们来看一下。

下面的这段脚本，先是输出一个问候语句，然后输出当前的时间：

```
#!/bin/bash

echo "Hello $USER,"

echo "Today is $(date +%Y-%m-%d)"
```

下面让我们使用-x选项来运行这段脚本：

```
$ bash -x example_script.sh

+ echo 'Hello chenhao,'

Hello chenhao,
```

```
++ date +%Y-%m-%d
```

```
+ echo 'Today is 2009-08-31'
```

```
Today is 2009-08-31
```

这时，我们可以看到，bash在运行前打印出了每一行命令。而且每行前面的+号表明了嵌套。这样的输出可以让你看到命令执行的顺序并可以让你知道整个脚本的行为。

在跟踪里输出行号

在一个很大的脚本中，你会看到很多很多的执行跟踪的输出，阅读起来非常费劲，所以，你可以在每一行前加上文件的行号，这会非常有用。要做到这样，你只需要设置下面的环境变量：

```
export PS4='+${BASH_SOURCE}:${LINENO}:${FUNCNAME[0]}: '
```

让我们看看设置上了PS4这个环境变量后会是什么样的输出。

```
$ bash -x example_script.sh
```

```
+example_script.sh:2:: echo 'Hello chenhao,'
```

```
Hello chenhao,
```

```
++example_script.sh:3:: date +%Y-%m-%d
```

```
+example_script.sh:3:: echo 'Today is 2009-08-31'
```

```
Today is 2009-08-31
```

调试部份的脚本

有些时候，你并不想调试整个脚本，你只要调试其中的一部份，那么，你可以在你想要调试的脚本之前，调用“set -x”，结束的时候调用“set +x”就可以了。如下面的脚本所示：

```
#!/bin/bash
```

```
echo "Hello $USER,"
```

```
set -x
```

```
echo "Today is $(date %Y-%m-%d)"
```

```
set +x
```

让我们看看运行起来是啥样？

```
$ ./example_script.sh
```

```
Hello chenhao,
```

```
++example_script.sh:4:: date +%Y-%m-%d
```

```
+example_script.sh:4:: echo 'Today is 2009-08-31'
```

```
Today is 2009-08-31
```

```
+example_script.sh:5:: set +x
```

注意：我们在运行脚本的时候，不需要使用bash -x了。

跟踪日志有时候太多了，多得都受不了，而且，输出的内容很难阅读。一般来说，我们很多时候只关心于条件表达式，变量值，或是函数调用，或是循环等。。在这种情况下，log一些感兴趣的特定的信息，可能会更好。

使用log前，我们先写一个函数：

```
_log() {
```

```
if [ "$_DEBUG" == "true" ]; then
```

```
echo 1>&2 "$@"
```

```
fi
```

```
}
```

于是，你就可以在你的脚本中如下使用：

```
_log "Copying files..."
```

```
cp src/* dst/
```

我们可以看到，上面那个_log函数，需要检查一个_DEBUG 变量，只有这个变量是真，才会真正开发输出日志。这样，你就只需要控制这个开关，而不需要删除你的debug信息。

```
$ _DEBUG=true ./example_script.sh
```

使用Bash 专用调试器

如果你在写一个相当复杂的脚本，并且，你需要一个完整的像调试别的语言一样的调试器，那么你可以试着用这个开源软件—— bashdb ^[3]，一个Bash的专用调试器。这个调试器很强大，你想得到的功能，他都有，比如，设置断点，单步跟踪，跳出函数，等等。它的用户接口很想GDB，这是他的文档 ^[4]。

(全文完)

(转载本站文章请注明作者和出处 酷壳 - CoolShell.cn ^[5]，请勿用于任何商业用途)

——=== 访问 酷壳404页面 ^[6] 寻找遗失儿童。 ===——

1. <http://coolshell.cn/>
2. <http://coolshell.cn/articles/author/haoel>
3. <http://bashdb.sourceforge.net/>
4. <http://bashdb.sourceforge.net/bashdb.html>
5. <http://coolshell.cn/>

6. <http://coolshell.cn/404/>