

awk从放弃到入门（7）：awk动作总结之一

在本博客中，AWK是一个系列文章，本人会尽量以通俗易懂的方式递进的总结awk命令的相关知识点。

awk系列博文直达链接：[AWK命令总结之从放弃到入门（通俗易懂，快进来看）](#)



这篇文章中的知识点是建立在前文的基础上的，如果你还没有掌握前文中的知识，请先参考之前的文章。

注：在阅读这篇文章之前，最好已经了解了一些基本的开发语法，比如，if、if...else、for、while等，否则在阅读时 有可能遇到障碍。

在前文中，我们已经使用过了awk的选项、模式 以及 动作。

这篇文章中，我们再来聊聊动作。

不知从何说起，我们还是从之前的示例开始聊吧，回顾一个小例子，如下图所示。

```
[www.zsythink.net]# cat test3
hey
heey
heey
heey
heey
[www.zsythink.net]#
[www.zsythink.net]# awk '{ print $0 }' test3
hey
heey
heey
heey
heey
[www.zsythink.net]#
```

zsythink.net 朱双印博客

如上图所示，红线标注部分就是awk命令中的"动作"，我想你应该已经非常熟悉了。

其实，我们可以把上述"动作"分解，拆分成两部分去理解，如下图所示。

```
awk '{ print $0 }' test3
```

上图中，我们将动作拆分成了两个部分。

红线标注为第一部分：最外侧的括号，即"{" }"。

蓝线标注为第二部分："print \$0"

在之前的示例中，我们一直把上图中的两个部分当做一个整理去理解，但是现在，我们要把它们分开去理解。

其实，这两个部分都可以被称之为"动作"，只不过它们是不同"类型"的动作而已。

"print"属于"输出语句"类型的动作，顾名思义，"输出语句"类型的动作的作用就是输出、打印信息，没错，"print"与"printf"都属于"输出语句"类型的动作。

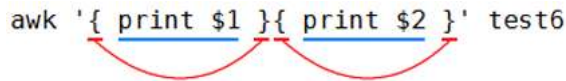
"{" }"其实也可以被称之为"动作"，只不过，"{" }"属于"组合语句"类型的动作，顾名思义，"组合语句"类型的动作的作用就是将多个代码组合成代码块。

这样说可能不容易理解，我们来看个小示例，就容易理解了，示例如下。

```
[www.zsythink.net]# cat test6
f s
1 2
1 2
[www.zsythink.net]# awk '{ print $1 }{ print $2 }' test6
f
s
1
2
1
2
[www.zsythink.net]#
```

zsythink.net 朱双印博客

如上图所示，我们使用了两个大括号"{}", 它们属于"组合语句"类型的动，它们分别将两个print括住，表示这两个print动作分别作为两个独立的个体，如下图所示



也就是说，我们可以这样理解，上图中一共有4个"动作"，两对大括号，两个print，但是上图中，每个大括号中只有一个动作，而我们说过，"组合语句"的作用是将个动作组合成代码块，组合后的代码块被当做一个整体，那么，我们能不能把上图中的两个print动作组合成一个整体呢？

必须能啊，示例如下。

如上图所示，我们只使用了一个大括号，将两个print动作组合成了一个整体，但是细心如你一定发现来了，当我们把多个动作（多段代码）组合成一个代码块的时候（每段代码）之间需要用分号";"隔开，如下图所示。

好了，我想你应该明白了，除了print这种"输出语句"能够被称之为动作以外，像"{}"这种"组合语句"也能被称之为动作，只不过它们的类型不同，功能也不同。

那么，除了"输出语句"与"组合语句"以外，还有其他种类的动作吗？

必须的，我们现在就来认识另一种动作，它就是"控制语句"。

不过，"控制语句"又有很多种，不过不用怕，我们慢慢来，一个一个聊，先来认识一种简单的"控制语句"，它就是"条件判断"。

如果你有过任何一种编程语言的开发经验，你都会非常容易理解"条件判断"，条件判断无非就是条件成立，则执行对应的代码，条件不成立，则不执行对应的命令，语言中，通常使用如下语法结构进行条件判断，也就是编程语法中的 if 条件判断语句。

```
1 if(条件)
2 {
3 语句1;
4 语句2;
5 ...
6 }
```

在awk中，我们同样可以使用if这种语法进行条件判断，只不过，上例中的语法结构是由"多行"组成，而在命令行中使用awk时，我们可以将上例中的"多行"语句写为单行如下。

上图中红线标注的部分即为"条件判断"类型的语法，我们把红线标注的部分单独取出来，来描述一下。

"if(NR == 1)"中的NR为awk的内置变量，NR为行号之意，所以，"if(NR == 1)"表示行号为1时，条件成立。

"if(NR == 1){ print \$0 }"表示行号为1是满足条件，条件满足时，打印整行，换句话说就是只打印第一行。

你可能会纠结，为什么最外侧还需要有一层大括号呢？如下图所示。

告诉你原因，原因就是……

没有为什么，就是要这样写，否则会报错。

如果你非要一个理由，那么我们可以这样理解，所有动作的最外侧必须用"{}"括起。

你可能还是会纠结，if语句的语法结构中也包含大括号啊，那么它属于"组合语句"吗？如下图所示

虽然它的语法结构中也包含大括号，但是我们仍然把if语句称之为"控制语句"，或者我们可以这样理解，"控制语句"中包含"组合语句"。

if语句中的大括号中，也可以执行多个动作，把多个代码段当做一个整体，也就是说，如果if所对应的条件成立，则执行if的大括号中的所有命令，示例如下。

上例表示，如果行号为1，则满足条件，就会执行if对应的大括号中的所有代码，而大括号中，有两个print动作，当条件成立时，这两个print动作都会被执行，当条件不成立时，这两个动作都不会执行。

上例中，"if"对应的大括号中有多条语句，所以"if"语法中的大括号不能省略，但是，如果"if"对应的大括号中只有一条命令，那么"if"对应的大括号则可以省略，示例如下。

如上图所示，当"if"对应的大括号中只有一条命令时，对应的大括号可以省略，但是需要注意，如果条件成立之后，需要执行多条语句，那么"if"对应的大括号则不能省略。

还记得我们在前文中使用到的"模式"吗？示例如下

没错，上图中的用法为awk的"模式"的用法，而我们今天所介绍的用法为awk的"动作"的用法，虽然两者在语法上有所区别，但是达到的目的相同的。

编程语言中，除了"if"之外，还有"if...else..."或者"if...else if...else"这样的语法，awk中也有这样的用法。

"if...else..."的语法如下：

```
1 if(条件)
2 {
3 语句1;
4 语句2;
5 ...
6 }
```

```
7  else
8  {
9  语句1;
10 语句2;
11  ...
12  }
```

"if...else if...else"的语法如下：

```
1  if(条件1)
2  {
3  语句1;
4  语句2;
5  ...
6  }
7  else if(条件2)
8  {
9  语句1;
10 语句2;
11  ...
12  }
13 else
14 {
15 语句1;
16 语句2;
17  ...
18  }
```

其实，这些语法与编程语言中的用法都是相同的，我相信你已经明白了，我们直接来看一些小示例吧。

我们知道，/etc/passwd文件中的第3列存放了用户的ID，在centos6中，用户ID小于500的用户都属于系统用户，用户ID大于500的用户都属于普通用户。所以，我们可以以500为分界线，根据用户ID判断用户是属于系统用户还是普通用户，centos7中以1000为分界线，此处用于示例的系统为centos6，所以以500作。我们可以通过一条awk命令，判断出/etc/passwd文件中的哪些用户属于系统用户，哪些用户属于普通用户，示例如下。

上图中，就用到了"if...else..."语法，如上图所示，\$3对应了passwd文件中的第三列，即用户ID，如果用户ID小于500，则输出\$1，即passwd文件中的第一列，也即用户名，并且输出"系统用户"字样，否则，则执行else中的命令，即打印用户名并输出"普通用户"字样，但是上例中，为了方便演示，我们并没有对输出的文本进行格式化，之前的知识，进行格式化。

好了，再来看一个"if...else if...else"这样的例子，其他它们都差不多，示例如下：

上例中，我们使用了"关系表达式"模式，同时，在动作中，使用了"if...else if...else"这样的"控制语句"，只要前文中的知识都掌握了，那么看懂上述示例，应该是没问题的。

当然了，还有很多种"控制语句"没有介绍，不过今天就到这里吧，希望这篇文章能够帮助到你。



我的微信公众号

关注"实用运维笔记"微信公众号，当博客中有新文章时，可第一时间得知哦~