

awk从放弃到入门（5）：awk模式（Pattern）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。

我们之前一直在提，awk的使用语法如下（我想你已经很熟悉了）：

```
awk [options] 'Pattern {Action}' file1 file2 ...
```

对于options（选项）而言，我们使用过-F选项，也使用过-v选项。

对于Action（动作）而言，我们使用过print与printf，之后的文章中，我们还会对Action进行总结。

对于Pattern（模式）而言，我们在刚开始学习awk时，就介绍了两种特殊模式，BEGIN模式和END模式，但是，我们并没有详细的介绍“模式”是什么，怎么用，而此细的介绍一下awk中的模式。

“模式”这个词听上去文绉绉的，不是特别容易理解，那么我们换一种说法，我们把“模式”换成“条件”，可能更容易理解，那么“条件”是什么意思呢？我们知道，awk本的，也就是说，awk会先处理完当前行，再处理下一行，如果我们不指定任何“条件”，awk会一行一行的处理文本中的每一行，如果我们指定了“条件”，只有满足会被处理，不满足“条件”的行就不会被处理。这样说是不是比刚才好理解一点了呢？这其实就是awk中的“模式”。

再啰嗦一遍，当awk进行逐行处理的时候，会把pattern（模式）作为条件，判断将要被处理的行是否满足条件，是否能跟“模式”进行匹配，如果匹配，则处理，如不进行匹配，则跳过。

看个小例子，就能秒懂，前提是建立在之前知识的基础之上。

如下图所示，test2文件中有3行文本，第一行有4列，第二行有5列，第三行只有2列。而下图的awk命令中，就使用到了一个简单的模式。

```
[www.zsythink.net]# cat test2
abc 123 iuy ddd
8ua 456 auv ppp 7y7
123 666
[www.zsythink.net]#
[www.zsythink.net]# awk 'NF==5 {print $0}' test2
8ua 456 auv ppp 7y7
[www.zsythink.net]#
```

zsythink.net 朱双印博客

上图中，我们使用了一个简单的“模式”，换句话说，我们使用了一个简单的“条件”，这个条件就是，如果被处理的行正好有5列字段，那么被处理的行则满足“条件”会执行相应的动作，而动作就是{print \$0}，即打印当前行，换句话说，就是只打印满足条件的行，条件就是这一行文本有5列（NF是内置变量，表示当前行的字段数，那么请你重新看一遍之前的文章），而上例中，只有第二行有5列，所以，只有第二行能与我们指定的“模式”相匹配，最终也就只输出了第二行。

这就是所谓的“模式”，其实很简单，对吧。聪明如你，应该已经能够举一反三了，举例如下。

```
[www.zsythink.net]# cat test2
abc 123 iuy ddd
8ua 456 auv ppp 7y7
123 666
[www.zsythink.net]# awk 'NF>2 {print $0}' test2
abc 123 iuy ddd
8ua 456 auv ppp 7y7
[www.zsythink.net]# awk 'NF<=4 {print $0}' test2
abc 123 iuy ddd
123 666
[www.zsythink.net]# awk '$1==123 {print $0}' test2
123 666
[www.zsythink.net]#
```

zsythink.net 朱双印博客

没错，"模式"怎样写，取决于我们想要给出什么样的限制条件。

细心如你一定发现了，上图中使用的"模式"都有一个共同点，就是上述"模式"中，都使用到了关系表达式（关系操作符），比如 ==，比如<=，比如>，当经过关系结果为"真"时，则满足条件（表示与指定的模式匹配），满足条件，就会执行相应的动作，而上例中使用到的运算符都是常见的关系运算符，我们就不解释了，那么awk关系运算符呢？我们来总结一下。

关系运算符	含义	用法示例
<	小于	x < y
<=	小于等于	x <= y
==	等于	x == y
!=	不等于	x != y
>=	大于等于	x >= y
>	大于	x > y
~	与对应的正则匹配则为真	x ~ /正则/
!~	与对应的正则不匹配则为真	x !~ /正则/

我们把这种用到了"关系运算符"的"模式"称之为："关系表达式模式"或者"关系运算符模式"。

其实，在学习"模式"之前，我们一直都在使用"模式"，只是我们没有感觉到而已，为什么这么说呢？听完下面的解释，你就会明白。

上图中的命令1指定了"模式"，而且这种"模式"是"关系表达式模式"，如果当前行的字段数量等于5，模式被匹配，对应的行被打印。

上图中的命令2貌似没有使用任何"模式"，所以，每一行都执行了指定的动作，即每一行都被输出了，其实，这种没有被指定任何"模式"的情况，也是一种"模式"，即为"空模式"，"空模式"会匹配文本中的每一行，所以，每一行都满足"条件"，所以，每一行都会执行相应的动作。

现在，我们不仅懂得了什么是awk的"模式（Pattern）"，而且还掌握了两种"模式"，空模式和关系运算模式。

不对，我们似乎遗忘了什么，我们还用过BEGIN模式和END模式，我们来回顾一下吧。

BEGIN模式，表示在开始处理文本之前，需要执行的操作。

END模式，表示将所有行都处理完毕后，需要执行的操作。

还记得我们在第一篇awk博文中使用到的例子吗，温故知新，回过头看，会有新发现


上图中的示例用到了BEGIN模式，空模式，END模式。

其实，"模式"还有其他的"样子"，只是我们今天还没有见识到而已，不过，今天的时光也没有白费，起码我们知道了"模式"的概念，以及如下三类"模式"

- 1、空模式
- 2、关系运算模式
- 3、BEGIN/END模式

笔者为了偷懒，今天就先到这里吧，等过两天，我们再看看模式的其他用法，再了解一下除了上述三类的其他类型的"模式"。

老规矩，求评论，求点赞，求收藏，求推荐~~~~各位客官，么么哒~~~~



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~