

awk 系列：如何使用 awk 按模式筛选文本或字符串

2016-7-21 08:02

编译自：<http://www.tecmint.com/awk-filter-text-or-string-using-patterns/>
原创：LCTT <https://linux.cn/article-7599-1.html>

作者：Aaron Kili
译者：FSSlc



Linux Awk Command Series - Part 3

```
tecmint@tecmint ~ $ awk '/ *\[2-9]\.[0-9][0-9] */ { print $0 "*" ; } / *\[0-1]\.[0-9][0-9] */ { print ; }' food_prices.list
1      Mangoes      10      $2.45*
2      Apples       20      $1.50
3      Bananas      5       $0.90
4      Pineapples   10      $3.46*
5      Oranges      10      $0.78
6      Tomatoes     5       $0.55
7      Onions       5       $0.45
tecmint@tecmint ~ $ |
```

Filter Text or Strings Using Patterns

作为 awk 命令系列的第三部分，这次我们将看一看如何基于用户定义的特定模式来筛选文本或字符串。

在筛选文本时，有时你可能想根据某个给定的条件或使用一个可被匹配的特定模式，去标记某个文件或数行字符串中的某几行。使用 awk 来完成这个任务是很容易的，这也正是 awk 中可能对你有所帮助的几个功能之一。

让我们看一看下面这个例子，比方说你有一个写有你想要购买的食物购物清单，其名称为 food_prices.list，它所含有的食物名称及相应的价格如下所示：

```
$ cat food_prices.list
```

No	Item_Name	Quantity	Price
1	Mangoes	10	\$2.45
2	Apples	20	\$1.50
3	Bananas	5	\$0.90
4	Pineapples	10	\$3.46
5	Oranges	10	\$0.78
6	Tomatoes	5	\$0.55
7	Onions	5	\$0.45

的：

```
$ awk '/ *\[2-9]\.[0-9][0-9] */ { print $1, $2, $3, $4, "*" ; } /
*\[0-1]\.[0-9][0-9] */ { print ; }' food_prices.list
```

```
tecmin@tecmin ~ $ awk '/ *\[2-9]\.[0-9][0-9] */ { print $1, $2, $3, $4, "*"
; } / *\[0-1]\.[0-9][0-9] */ { print ; }' food_prices.list
```

打印出单价大于 \$2 的项目

从上面的输出你可以看到在含有 *mangoes* 和 *pineapples* 的那行末尾都已经有了一个 **(*)** 标记。假如你检查它们的单价，你可以看到它们的单价的确超过了 \$2。

在这个例子中，我们已经使用了两个模式：

- 第一个模式： `/ *\[2-9]\.[0-9][0-9] */` 将会得到那些含有食物单价大于 \$2 的行，
- 第二个模式： `/*\[0-1]\.[0-9][0-9] */` 将查找那些食物单价小于 \$2 的那些行。

上面的命令具体做了什么呢？这个文件有四个字段，当模式一匹配到含有食物单价大于 \$2 的行时，它便会输出所有的四个字段并在该行末尾加上一个 **(*)** 符号来作为标记。

第二个模式只是简单地输出其他含有食物单价小于 \$2 的行，按照它们出现在输入文件 `food_prices.list` 中的样子。

这样你就可以使用模式来筛选出那些价格超过 \$2 的食物项目，尽管上面的输出还有些问题，带有 **(*)** 符号的那些行并没有像其他行那样被格式化输出，这使得输出显得不够清晰。

我们在 awk 系列的第二部分中也看到了同样的问题，但我们可以使用下面的两种方式来解决：

1、可以像下面这样使用 `printf` 命令，但这样使用又长又无聊：

```
$ awk '/ *\[2-9]\.[0-9][0-9] */ { printf "%-10s %-10s %-10s %-10s\n",
$1, $2, $3, $4 "*" ; } / *\[0-1]\.[0-9][0-9] */ { printf "%-10s %-10s
%-10s %-10s\n", $1, $2, $3, $4; }' food_prices.list
```

```
tecmin@tecmin ~ $ awk '/ *\[2-9]\.[0-9][0-9] */ { printf "%-10s %-10s %-10s %-10s\n", $1, $2, $3, $4 "*" ; } / *\[0-1]\.[0-9][0-9] */ { printf "%-10s %-10s %-10s %-10s\n", $1, $2, $3, $4; }' food_prices.list |
```

使用 Awk 和 Printf 来筛选和输出项目

2、使用 `$0` 字段。Awk 使用变量 `0` 来存储整个输入行。对于上面的问题，这种方式非常方便，并且它还简单、快速：

```
$ awk '/ *\[2-9]\.[0-9][0-9] */ { print $0 "*" ; } / *\[0-1]\.[0-9][0-9] */ { print ; }' food_prices.list
```

```
tecmin@tecmin ~ $ awk '/ *\[2-9]\.[0-9][0-9] */ { print $0 "*" ; } / *\[0-1]\.[0-9][0-9] */ { print ; }' food_prices.list |
```

使用 Awk 和变量来筛选和输出项目

结论

这就是全部内容了，使用 awk 命令你便可以通过几种简单的方法去利用模式匹配来筛选文本，帮助你在一个文件对文本或字符串的某些行做标记。

希望这篇文章对你有所帮助。记得阅读这个系列的下一部分，我们将关注在 awk 工具中使用比较运算符。

via: <http://www.tecmint.com/awk-filter-text-or-string-using-patterns/>

作者：Aaron Kili 译者：FSSlc 校对：wxy

本文由 LCTT 原创编译，Linux中国 荣誉推出