



Python入门+爬虫+数据分析+办公自动化+永久学习

免费赠：大厂导师答疑+50GPython学习大礼包+5G面试礼包

立即领取 >

## Linux awk命令详解

< 上一节

下一节 >

除了使用 sed 命令，Linux 系统中还有一个功能更加强大的文本数据处理工具，就是 **awk**。它诞生于 20 世纪 70 年代末期，这也许是它影响了众多 Linux 用户的原因之一。

曾有人推测 awk 命令的名字来源于 awkward 这个单词。其实不然，此命令的设计者有 3 位，他们的姓分别是 Aho、Weingberger 和 Kernighan，awk 就取自这 3 位大师姓的首字母。

和 sed 命令类似，awk 命令也是逐行扫描文件（从第 1 行到最后一行），寻找含有目标文本的行，如果匹配成功，则会在该行上执行用户想要的操作；反之，则不对行做任何处理。

awk 命令的基本格式为：

```
[root@localhost ~]# awk [选项] '脚本命令' 文件名
```

此命令常用的选项以及各自的含义，如表 1 所示。

表 1 awk 命令选项以及含义	
选项	含义
-F fs	指定以 fs 作为输入行的分隔符，awk 命令默认分隔符为空格或制表符。
-f file	从脚本文件中读取 awk 脚本指令，以取代直接在命令行中输入指令。
-v var=val	在执行处理过程之前，设置一个变量 var，并给其设备初始值为 val。

awk 的强大之处在于脚本命令，它由 2 部分组成，分别为匹配规则和执行命令，如下所示：

```
'匹配规则{执行命令}'
```

这里的匹配规则，和 sed 命令中的 address 部分作用相同，用来指定脚本命令可以作用到文本内容中的具体行，可以使用字符串（比如 /demo/，表示查看含有 demo 字符串的行）或者正则表达式指定。另外需要注意的是，整个脚本命令是用单引号 (") 括起，而其中的执行命令部分需要用大括号 ({} ) 括起来。

在 awk 程序执行时，如果没有指定执行命令，则默认会把匹配的行输出；如果不指定匹配规则，则默认匹配文本中所有的行。

举个简单的例子：

```
[root@localhost ~]# awk '/^$/ {print "Blank line"}' test.txt
```

在此命令中，`/^$/` 是一个正则表达式，功能是匹配文本中的空白行，同时可以看到，执行命令使用的是 print 命令，此命令经常会使用，它的作用很简单，就是将指定的文本进行输出。因此，整个命令的功能是，如果 test.txt 有 N 个空白行，那么执行此命令会输出 N 个 Blank line。

## awk 使用数据字段变量

awk 的主要特性之一是其处理文本文件中数据的能力，它会自动给一行中的每个数据元素分配一个变量。

默认情况下，awk 会将如下变量分配给它在文本行中发现的数据字段：

- \$0 代表整个文本行；
- \$1 代表文本行中的第 1 个数据字段；
- \$2 代表文本行中的第 2 个数据字段；

- \$n 代表文本行中的第 n 个数据字段。

前面说过，在 awk 中，默认的字段分隔符是任意的空白字符（例如空格或制表符）。在文本行中，每个数据字段都是通过字段分隔符划分的。awk 在读取一行文本时，会用预定义的字段分隔符划分每个数据字段。

所以在下面的例子中，awk 程序读取文本文件，只显示第 1 个数据字段的值：

```
[root@localhost ~]# cat data2.txt
One line of test text.
Two lines of test text.
Three lines of test text.
[root@localhost ~]# awk '{print $1}' data2.txt
One
Two
Three
```

该程序用 \$1 字段变量来表示“仅显示每行文本的第 1 个数据字段”。当然，如果你要读取采用了其他字段分隔符的文件，可以用 -F 选项手动指定。

## awk 脚本命令使用多个命令

awk 允许将多条命令组合成一个正常的程序。要在命令行上的程序脚本中使用多条命令，只要在命令之间放个分号即可，例如：

```
[root@localhost ~]# echo "My name is Rich" | awk '{$4="Christine"; print $0}'
My name is Christine
```

第一条命令会给字段变量 \$4 赋值。第二条命令会打印整个数据字段。可以看到，awk 程序在输出中已经将原文本中的第四个数据字段替换成了新值。

除此之外，也可以一次一行地输入程序脚本命令，比如说：

```
[root@localhost ~]# awk '{
> $4="Christine"
> print $0}'
My name is Rich
My name is Christine
```

在你用了表示起始的单引号后，bash shell 会使用 > 来提示输入更多数据，我们可以每次在每行加一条命令，直到输入了结尾的单引号。

注意，此例中因为没有在命令行中指定文件名，awk 程序需要用户输入获得数据，因此当运行这个程序的时候，它会一直等着用户输入文本，此时如果要退出程序，只需按下 Ctrl+D 组合键即可。

## awk从文件中读取程序

跟 sed 一样，awk 允许将脚本命令存储到文件中，然后再在命令行中引用，比如：

```
[root@localhost ~]# cat awk.sh
{print $1 "'s home directory is " $6}
[root@localhost ~]# awk -F: -f awk.sh /etc/passwd
root's home directory is /root
bin's home directory is /bin
daemon's home directory is /sbin
adm's home directory is /var/adm
lp's home directory is /var/spool/lpd
...
Christine's home directory is /home/Christine
Samantha's home directory is /home/Samantha
Timothy's home directory is /home/Timothy
```

awk.sh 脚本文件会使用 print 命令打印 /etc/passwd 文件的主目录数据字段（字段变量 \$6），以及 userid 数据字段（字段变量 \$1）。注意，在程序文件中，也可以指定多条命令，只要一条命令放一行即可，之间不需要用分号。

## awk BEGIN关键字

awk 中还可以指定脚本命令的运行时机。默认情况下，awk 会从输入中读取一行文本，然后针对该行的数据执行程序脚本，但有时可能需要在处理数据前运行一些脚本命令，这就需要使用 BEGIN 关键字。

BEGIN 会强制 awk 在读取数据前执行该关键字后指定的脚本命令，例如：

```
[root@localhost ~]# cat data3.txt
Line 1
Line 2
Line 3
[root@localhost ~]# awk 'BEGIN {print "The data3 File Contents:"}
> {print $0}' data3.txt
The data3 File Contents:
Line 1
Line 2
Line 3
```

可以看到，这里的脚本命令中分为 2 部分，BEGIN 部分的脚本指令会在 awk 命令处理数据前运行，而真正用来处理数据的是第二段脚本命令。

## awk END关键字

和 BEGIN 关键字相对应，END 关键字允许我们指定一些脚本命令，awk 会在读完数据后执行它们，例如：

```
[root@localhost ~]# awk 'BEGIN {print "The data3 File Contents:"}
> {print $0}
> END {print "End of File"}' data3.txt
The data3 File Contents:
Line 1
Line 2
Line 3
End of File
```

可以看到，当 awk 程序打印完文件内容后，才会执行 END 中的脚本命令。

本节仅介绍了 awk 程序的基本用法，更高级的用法会在下节继续介绍。

关注公众号「[站长严长生](#)」，在手机上阅读所有教程，随时随地都能学习。本公众号由[C语言中文网站长](#)亲自运营，长期更新，坚持原创。



微信扫码关注公众号

优秀文章
Go语言简介
Java类的定义及定义类时可用的关键字
VS调试的总结以及技巧
C语言strcpy()函数：字符串复制（拷贝）函数
Java二维数组详解
SEO网页标题（title）该怎么写？
C++ STL list删除元素详解



ORM是什么

初学者如何找到适合自己的编程书籍？

C++ char16\_t和char32\_t字符类型详解

精美而实用的网站，分享优质编程教程，帮助有志青年。千锤百炼，只为大作；精益求精，处处斟酌；这种教程，看一眼就倾心。

[关于网站](#) | [关于站长](#) | [如何完成一部教程](#) | [公众号](#) | [联系我们](#) | [网站地图](#)

Copyright ©2012-2022 biancheng.net, 冀ICP备2022013920号, 冀公网安备13110202001352号

