

昵称：琴酒网络
园龄：2年4个月
粉丝：78
关注：0
+加关注

<

2019年5月

>

日	一	二	三	四	五	六
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

搜索

找找看

谷歌搜

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

更多链接

我的标签

linux(25)

linux命令总结(9)

python(6)

命令总结(4)

apache(2)

迭代器(2)

centos6.6(2)

CentOS6.6安装(1)

crond定时任务(1)

CSS3(1)

更多

随笔分类

HTML5(3)

jquery

linux(54)

python(20)

随笔档案

2017年11月 (4)

2017年6月 (1)

2017年5月 (4)

2017年3月 (7)

2017年2月 (33)

linux命令总结sed命令详解

Sed 简介

sed 是一种新型的，非交互式的编辑器。它能执行与编辑器 vi 和 ex 相同的编辑任务。sed 编辑器没有提供交互式使用方式，使用者只能在命令行输入编辑命令、指定文件名，然后在屏幕上查看输出。sed 编辑器没有破坏性，它不会修改文件，除非使用 shell 重定向来保存输出结果。默认情况下，所有的输出行都被打印到屏幕上。

sed 工作过程

sed 编辑器逐行处理文件（或输入），并将输出结果发送到屏幕。sed 的命令就是在 vi和 ed/ex 编辑器中见到的那些。sed 把当前正在处理的行保存在一个临时缓存区中，这个缓存区称为模式空间或临时缓冲。sed 处理完模式空间中的行后（即在该行上执行 sed 命令后），就把该行发送到屏幕上（除非之前有命令删除这一行或取消打印操作）。sed 每处理完输入文件的最后一行后，sed 便结束运行。sed 把每一行都存在临时缓存区中，对这个副本进行编辑，所以不会修改或破坏源文件。如图 1：sed 处理过程。



图 1

从上图可以看出 sed 不是破坏性的，它不会修改正在编辑的文件。

Sed 命令格式

1 | sed 命令行格式为： sed [选项] ‘ command’ 输入文本

Sed 定位

Sed 命令在没有给定的位置时，默认会处理所有行；
Sed 支持一下几种地址类型：
1、first~step
这两个单词的意思：first 指起始匹配行，step 指步长，例如：sed -n 2~5p 含义：从第二行开始匹配，隔 5 行匹配一次，即 2,7,12.....。
2、\$
这个\$符表示匹配最后一行。
3、/REGEXP/
这个是表示匹配正则那一行，通过//之间的正则来匹配。
4、\cREGEXPc
这个是表示匹配正则那一行，通过\c 和 c 之间的正则来匹配,c 可以是任一字符
5、addr1, add2
定址 addr1, add2 决定用于对哪些行进行编辑。地址的形式可以是数字、正则表达式或二者的结合。如果没有指定地址，sed 将处理输入文件中的所有行。如果定址是一个数字，则这个数字代表行号，如果是逗号分隔的两个行号，那么需要处理的定址就是两行之间的范围（包括两行在内）。范围可以是数字，正则或二者组合。

2017年1月 (34)

最新评论

1. Re:django基础之数据库操作
很强

--芦苇菌
2. Re:Linux三剑客之awk命令
有点意思

--TBHacker
3. Re:真正的inotify+rsync实时同步 彻底告别同步慢
./rsync.sh: line 16: conditional binary operator expected./rsync.sh: line 16: syntax error near `\${IN}.....'

--队长china
4. Re:django基础之数据库操作
查看表用的是什么工具?

--伴燕儿飞
5. Re:linux命令总结之seq命令
\t

--MaoHe

阅读排行榜

1. Linux三剑客之awk命令(105603)
2. linux命令总结之ip命令(46988)
3. linux命令总结sed命令详解(35455)
4. linux常见故障一: linux 文件系统变只读(22477)
5. python之OS模块详解(20785)

评论排行榜

1. Linux三剑客之awk命令(2)
2. linux命令总结之seq命令(2)
3. django基础之数据库操作(2)
4. Python常用内置模块之xml模块(1)
5. 真正的inotify+rsync实时同步 彻底告别同步慢(1)

推荐排行榜

1. Linux三剑客之awk命令(15)
2. linux命令总结sed命令详解(4)
3. python之OS模块详解(2)
4. linux命令总结之dig命令(2)
5. linux命令总结之date命令(1)

6、addr1, +N

从 addr1 这行到往下 N 行匹配，总共匹配 N+1 行

7、addr1, ~N

Will match addr1 and the lines following addr1 until the next line whose input line number is a multiple of N.【没有看懂是什么意思】

Sed 的正则表达式

表 1： sed 的正则表达式元字符

元字符	功 能	示 例	示例的匹配对象
^	行首定位符	/^love/	匹配所有以 love 开头的行
\$	行尾定位符	/love\$/	匹配所有以 love 结尾的行
.	匹配除换行外的单个字符	/l.e/	匹配包含字符 l、后跟两个任意字符、再跟字母 e 的行
*	匹配零个或多个前导字符	/*love/	匹配在零个或多个空格紧跟着模式 love 的行
[]	匹配指定字符组内任一字符	/[Ll]ove/	匹配包含 love 和 Love 的行
[^]	匹配不在指定字符组内任一字符	/[^A-KM-Z]ove/	匹配包含 ove，但 ove 之前的那个字符不在 A 至 K 或 M 至 Z 间的行
\(.\)	保存已匹配的字符		
&	保存查找串以便在替换串中引用	s/love/**&*/	符号&代表查找串。字符串 love 将替换前后各加了两个**的引用，即 love 变成 **love**
\<	词首定位符	/\<love/	匹配包含以 love 开头的单词的行
\>	词尾定位符	/love\>/	匹配包含以 love 结尾的单词的行
x\{m\}	连续 m 个 x	/o\{5\}/	分别匹配出现连续 5 个字母 o、至少 5 个连续的 o、或 5~10 个连续的 o 的行
x\{m,\}	至少 m 个 x	/o\{5,\}/	
x\{m,n\}	至少 m 个 x，但不超过 n 个 x	/o\{5,10\}/	

sed的常用选项

表 2.sed 的常用选项

选项	说明
-n	使用安静模式，在一般情况所有的STDIN 都会输出到屏幕上，加入-n 后只打印被 sed 特殊处理的行
-e	多重编辑，且命令顺序会影响结果
-f	指定一个 sed 脚本文件到命令行执行，
-r	Sed 使用扩展正则
-i	直接修改文档读取的内容，不在屏幕上输出

Sed 操作命令

sed 操作命令告诉 sed 如何处理由地址指定的各输入行。如果没有指定地址， sed 就会处理输入的所有行。表 3.sed 命令

命 令	说 明
a\	在当前行后添加一行或多行
c\	用新文本修改（替换）当前行中的文本
d	删除行
i\	在当前行之前插入文本
h	把模式空间里的内容复制到暂存缓存区
H	把模式空间里的内容追加到暂存缓存区
g	取出暂存缓冲区里的内容，将其复制到模式空间，覆盖该处原有内容
G	取出暂存缓冲区里的内容，将其复制到模式空间，追加在原有内容后面
l	列出非打印字符
p	打印行
n	读入下一输入行，并从下一条命令而不是第一条命令开始处理
q	结束或退出 sed
r	从文件中读取输入行
!	对所选行意外的所有行应用命令
s	用一个字符串替换另一个

表 4.替换标志

g	在行内进行全局替换
p	打印行
w	将行写入文件

x	交换暂存缓冲区与模式空间的内容
y	将字符转换为另一字符（不能对正则表达式使用 y 命令）

报错信息和退出信息

遇到语法错误时，sed 会向标准错误输出发送一条相当简单的报错信息。但是，如果 sed 判断不出错在何处，它会“断章取义”，给出令人迷惑的报错信息。如果没有语法错误，sed 将会返回给 shell 一个退出状态，状态为 0 代表成功，为非 0 整数代表失败。

sed 使用实例

下面给出测试文件作为输入文件：

```
1 [root@Gin scripts]# cat ceshi.txt
2 northwest      NW      Charles Main    3.0      .98      3        34
3 western         WE      Sharon Gray     5.3      .97      5        23
4 southwest       SW      Lewis Dalsass   2.7      .8        2        18
5 southern        SO      Suan Chin       5.1      .95      4        15
6 southeast       SE      Patricia Hemenway 4.0      .7        4        17
7 eastern         EA      TB Savage       4.4      .84      5        20
8 northeast       NE      AM Main Jr.     5.1      .94      3        13
9 north           NO      Margot Weber    4.5      .89      5         9
10 central        CT      Ann Stephens    5.7      .94      5        13
```

打印：p 命令

命令 p 是打印命令，用于显示模式缓存区的内容。默认情况下，sed 把输入行打印在屏幕上，选项 -n 用于取消默认打印操纵。当选项 -n 和命令 p 同时出现时，sed 可打印选定的内容

案例1：

```
1 [root@Gin scripts]# sed '/north/p' ceshi.txt
2 northwest      NW      Charles Main    3.0      .98      3        34
3 northwest      NW      Charles Main    3.0      .98      3        34
4 western         WE      Sharon Gray     5.3      .97      5        23
5 southwest       SW      Lewis Dalsass   2.7      .8        2        18
6 southern        SO      Suan Chin       5.1      .95      4        15
7 southeast       SE      Patricia Hemenway 4.0      .7        4        17
8 eastern         EA      TB Savage       4.4      .84      5        20
9 northeast       NE      AM Main Jr.     5.1      .94      3        13
10 northeast      NE      AM Main Jr.     5.1      .94      3        13
11 north          NO      Margot Weber    4.5      .89      5         9
12 north          NO      Margot Weber    4.5      .89      5         9
13 central        CT      Ann Stephens    5.7      .94      5        13
```

说明：默认情况下，sed 把所有输入行都打印在标准输出上。如果在某一行匹配到 north，sed 就把该行另外打印一遍。

案例2：

```
1 [root@Gin scripts]# sed -n '/north/p' ceshi.txt
2 northwest      NW      Charles Main    3.0      .98      3        34
3 northeast      NE      AM Main Jr.     5.1      .94      3        13
4 north          NO      Margot Weber    4.5      .89      5         9
```

说明：默认情况下，sed 打印当前缓存区中的输入行。命令 p 指示 sed 将再次打印该行。选项-n 取消 sed 取消默认打印操作。选项-n 和命令配合使用，模式缓冲区内的输入行，只被打印一次。如果不指定-n 选项，sed 就会像上例中那样，打印出重复的行。如果指定了-n，则sed 只打印包含模式 north 的行。

删除：d 命令

命令 d 用于删除输入行。sed 先将输入行从文件复制到模式缓存区，然后对该行执行 sed 命令，最后将模式缓存区的内容显示在屏幕上。如果发出的是命令 d，当前模式缓存区的输入行会被删除，不被显示。

案例 3：

```
1 [root@Gin scripts]# sed '3d' ceshi.txt
2 northwest      NW      Charles Main    3.0      .98      3      34
3 western        WE      Sharon Gray     5.3      .97      5      23
4 southern       SO      Suan Chin      5.1      .95      4      15
5 southeast      SE      Patricia Hemenway 4.0      .7       4      17
6 eastern        EA      TB Savage      4.4      .84      5      20
7 northeast      NE      AM Main Jr.    5.1      .94      3      13
8 north          NO      Margot Weber   4.5      .89      5      9
9 central        CT      Ann Stephens   5.7      .94      5      13
```

说明：删除第 3 行。默认情况下，其余的行都被打印到屏幕上。

案例 4：

```
1 [root@Gin scripts]# sed '3,$d' ceshi.txt
2 northwest      NW      Charles Main    3.0      .98      3      34
3 western        WE      Sharon Gray     5.3      .97      5      23
```

说明：删除从第三行到最后一行内容，剩余各行被打印。地址范围是开始第 3 行，结束最后一行。

案例 5：

```
1 [root@Gin scripts]# sed '/north/d' ceshi.txt
2 western        WE      Sharon Gray     5.3      .97      5      23
3 southwest      SW      Lewis Dalsass   2.7      .8       2      18
4 southern       SO      Suan Chin      5.1      .95      4      15
5 southeast      SE      Patricia Hemenway 4.0      .7       4      17
6 eastern        EA      TB Savage      4.4      .84      5      20
7 central        CT      Ann Stephens   5.7      .94      5      13
```

说明：所有包含模式 north 的行都被删除，其余行被打印。

替换：s 命令

命令 s 是替换命令。替换和取代文件中的文本可以通过 sed 中的 s 来实现，s 后包含在斜杠中的文本是正则表达式，后面跟着的是需要替换的文本。可以通过 g 标志对行进行全局替换

案例 6：

```
1 [root@Gin scripts]# sed 's/west/north/g' ceshi.txt
2 northnorth     NW      Charles Main    3.0      .98      3      34
3 northern       WE      Sharon Gray     5.3      .97      5      23
4 southnorth     SW      Lewis Dalsass   2.7      .8       2      18
5 southern       SO      Suan Chin      5.1      .95      4      15
6 southeast      SE      Patricia Hemenway 4.0      .7       4      17
7 eastern        EA      TB Savage      4.4      .84      5      20
8 northeast      NE      AM Main Jr.    5.1      .94      3      13
9 north          NO      Margot Weber   4.5      .89      5      9
```

```

10 | central          CT      Ann Stephens    5.7    .94    5      13

```

说明：s 命令用于替换。命令末端的 g 表示在行内全局替换；也就是说如果每一行里出现多个 west，所有的 west 都会被替换为 north。如果没有 g 命令，则只将每一行的第一 west 替换为 north。

案例 7：

```

1 | [root@Gin scripts]# sed -n 's/^west/north/p' ceshi.txt
2 | northern          WE      Sharon Gray    5.3    .97    5      23

```

说明：s 命令用于替换。选项 -n 与命令行末尾的标志 p 结合，告诉 sed 只打印发生替换的那些行；也就是说，如果只有在行首找到 west 并替换成 north 时才会打印此行。

案例 8：

```

1 | [root@Gin scripts]# sed 's/[0-9][0-9]$/&.5/' ceshi.txt
2 | northwest        NW      Charles Main   3.0    .98    3      34.5
3 | western          WE      Sharon Gray    5.3    .97    5      23.5
4 | southwest        SW      Lewis Dalsass  2.7    .8     2      18.5
5 | southern         SO      Suan Chin     5.1    .95    4      15.5
6 | southeast        SE      Patricia Hemenway 4.0    .7     4      17
7 | eastern          EA      TB Savage     4.4    .84    5      20.5
8 | northeast        NE      AM Main Jr.   5.1    .94    3      13.5
9 | north            NO      Margot Weber  4.5    .89    5      9
10 | central          CT      Ann Stephens   5.7    .94    5      13.5

```

说明：当“与”符号（&）用在替换串中时，它代表在查找串中匹配到的内容。这个示例中所有以 2 位数结尾的行后面都被加上.5。

案例 9：

```

1 | [root@Gin scripts]# sed -n 's/Hemenway/Jones/gp' ceshi.txt
2 | southeast        SE      Patricia Jones 4.0    .7     4      17

```

说明：文件中出现的所有的 Hemenway 都被替换为 Jones，只有发生变化的行才会打印出来。选项 -n 与命令 p 的组合取消了默认的输出。标志 g 的含义是表示在行内全局替换。

案例 10：

```

1 | [root@Gin scripts]# sed 's/(Mar\)got/\1linanne/p' ceshi.txt
2 | northwest        NW      Charles Main   3.0    .98    3      34
3 | western          WE      Sharon Gray    5.3    .97    5      23
4 | southwest        SW      Lewis Dalsass  2.7    .8     2      18
5 | southern         SO      Suan Chin     5.1    .95    4      15
6 | southeast        SE      Patricia Hemenway 4.0    .7     4      17
7 | eastern          EA      TB Savage     4.4    .84    5      20
8 | northeast        NE      AM Main Jr.   5.1    .94    3      13
9 | north            NO      Marlinanne Weber 4.5    .89    5      9
10 | north            NO      Marlinanne Weber 4.5    .89    5      9
11 | central          CT      Ann Stephens   5.7    .94    5      13

```

说明：包含在圆括号里的模式 Mar 作为标签 1 保存在特定的寄存器中。替换串可以通过 \1 来引用它。则 Margot 被替换为 Marlinane。

案例 11：

```

1 | [root@Gin scripts]# sed 's#3#88#g' ceshi.txt
2 | northwest        NW      Charles Main   88.0    .98    88     884
3 | western          WE      Sharon Gray    5.88    .97    5      288
4 | southwest        SW      Lewis Dalsass  2.7    .8     2      18

```

5	southern	SO	Suan Chin	5.1	.95	4	15
6	southeast	SE	Patricia Hemenway	4.0	.7	4	17
7	eastern	EA	TB Savage	4.4	.84	5	20
8	northeast	NE	AM Main Jr.	5.1	.94	88	188
9	north	NO	Margot Weber	4.5	.89	5	9
10	central	CT	Ann Stephens	5.7	.94	5	188

说明：紧跟在 `s` 命令后的字符就是查找串和替换串之间的分隔符。分隔符默认默认为正斜杠，但可以改变。无论什么字符（换行符，反斜线除外），只要紧跟在 `s` 命令，就成了新的串分隔符。这个方法在查找包含正斜杠模式时很管用，例如查找路径名或生日。

指定行的范围：逗号

行的范围从文件中的一个地址开始，在另一个地址结束。地址范围可以是行号（例如5,10），正则表达式（例如/Dick/和/Joe/），或者两者的结合（例如/north/,\$）范围是闭合的——包含开始条件的行，结束条件的行，以及两者之间的行。如果结束条件无法满足，就会一直操作到文件结尾。如果结束条件满足，则继续查找满足开始条件的位置，范围重新开始。

案例 12：

1	[root@Gin scripts]# sed -n '/west/,/east/p' ceshi.txt						
2	northwest	NW	Charles Main	3.0	.98	3	34
3	western	WE	Sharon Gray	5.3	.97	5	23
4	southwest	SW	Lewis Dalsass	2.7	.8	2	18
5	southern	SO	Suan Chin	5.1	.95	4	15
6	southeast	SE	Patricia Hemenway	4.0	.7	4	17

说明：打印模式 `west` 和 `east` 之间所有的行。如果 `west` 出现在 `east` 之后的某一行，则打印的范围从 `west` 所在行开始，到下一个出现 `east` 的行或文件的末尾（如果前者未出现）。图中用箭头表示出了该范围。

[root@yum test]# sed -n '/west/,/east/p' sed.txt							
→	northwest	NW	Charles Main	3.0	.98	3	34
	western	WE	Sharon Gray	5.3	.97	5	23
	southwest	SW	Lewis Dalsass	2.7	.8	2	18
	southern	SO	Suan Chin	5.1	.95	4	15
→	southeast	SE	Patricia Hemenway	4.0	.7	4	17

案例 13：

1	[root@Gin scripts]# sed -n '5,/northeast/p' ceshi.txt						
2	southeast	SE	Patricia Hemenway	4.0	.7	4	17
3	eastern	EA	TB Savage	4.4	.84	5	20
4	northeast	NE	AM Main Jr.	5.1	.94	3	13

说明：打印从第 5 行开始第一个以 `northeast` 开头的行之间的所有行。

案例 14：

1	[root@Gin scripts]# sed '/west/,/east/s/\$/**VACA**/' ceshi.txt						
2	northwest	NW	Charles Main	3.0	.98	3	34**VACA**
3	western	WE	Sharon Gray	5.3	.97	5	23**VACA**
4	southwest	SW	Lewis Dalsass	2.7	.8	2	18**VACA**
5	southern	SO	Suan Chin	5.1	.95	4	15**VACA**
6	southeast	SE	Patricia Hemenway	4.0	.7	4	17
7	eastern	EA	TB Savage	4.4	.84	5	20
8	northeast	NE	AM Main Jr.	5.1	.94	3	13
9	north	NO	Margot Weber	4.5	.89	5	9

10	central	CT	Ann Stephens	5.7	.94	5	13
----	---------	----	--------------	-----	-----	---	----

说明：修改从模式 wast 和 east 之间的所有行，将各行的行尾(\$)替换为字符串**VACA**。换行符被移到新的字符串后面。

多重编辑：e 命令

-e 命令是编辑命令，用于 sed 执行多个编辑任务的情况下。在下一行开始编辑前，所有的编辑动作将应用到模式缓存区的行上。

案例 15：

```
1 [root@Gin scripts]# sed -e '1,3d' -e 's/Hemenway/Jones/' ceshi.txt
2 southern      SO      Suan Chin      5.1      .95      4      15
3 southeast     SE      Patricia Jones 4.0      .7       4      17
4 eastern       EA      TB Savage      4.4      .84      5      20
5 northeast     NE      AM Main Jr.    5.1      .94      3      13
6 north        NO      Margot Weber   4.5      .89      5      9
7 central      CT      Ann Stephens   5.7      .94      5      13
```

说明：选项-e 用于进行多重编辑。第一重编辑编辑删除第 1~3 行。第二重编辑将Hemenway 替换为 Jones。因为是逐行进行这两行编辑（即这两个命令都在模式空间的当前行上执行），所以编辑命令的顺序会影响结果。例如，如果两条命令都执行的是替换，前一次替换会影响后一次替换。

追加：a 命令

a 命令是追加命令，追加将新文本到文件中当前行(即读入模式的缓存区行)的后面。不管是在命令行中，还是在 sed 脚本中，a 命令总是在反斜杠的后面。

案例 16：

```
1 [root@Gin scripts]# sed '/^north/a Hello world!' ceshi.txt
2 northwest     NW      Charles Main   3.0      .98      3      34
3 Hello world!
4 western       WE      Sharon Gray    5.3      .97      5      23
5 southwest     SW      Lewis Dalsass  2.7      .8       2      18
6 southern      SO      Suan Chin      5.1      .95      4      15
7 southeast     SE      Patricia Hemenway 4.0      .7       4      17
8 eastern       EA      TB Savage      4.4      .84      5      20
9 northeast     NE      AM Main Jr.    5.1      .94      3      13
10 Hello world!
11 north        NO      Margot Weber   4.5      .89      5      9
12 Hello world!
13 central      CT      Ann Stephens   5.7      .94      5      13
```

说明：命令 a 用于追加。字符串 Hello, World! 被加在以 north 开头的各行之后。如果要追加的内容超过一行，则除最后一行外，其他各行都必须以反斜杠结尾。

插入：i 命令

i 命令是插入命令，类似于 a 命令，但不是在当前行后增加文本，而是在当前行前面插入新的文本，即刚读入缓存区模式的行。

案例 17：

```
1 [root@Gin scripts]# sed '/eastern/i Hello,world!\
2 > -----' ceshi.txt
3 northwest     NW      Charles Main   3.0      .98      3      34
4 western       WE      Sharon Gray    5.3      .97      5      23
5 southwest     SW      Lewis Dalsass  2.7      .8       2      18
6 southern      SO      Suan Chin      5.1      .95      4      15
7 southeast     SE      Patricia Hemenway 4.0      .7       4      17
8 Hello,world!
```


9	-----						
10	eastern	EA	TB Savage	4.4	.84	5	20
11	northeast	NE	AM Main Jr.	5.1	.94	3	13
12	north	NO	Margot Weber	4.5	.89	5	9
13	central	CT	Ann Stephens	5.7	.94	5	13

说明：命令 i 是插入命令。如果在某一行匹配到模式 eastern,i 命令就在该行的上方插入命令中插入反斜杠后面后的文本。除了最后一行，

修改：c 命令

c 命令是修改命令。sed 使用该命令将已有的文本修改成新的文本。旧文本被覆盖。

案例 18:

1	[root@Gin scripts]# sed '/eastern/c Hello,world! \'						
2	> -----' ceshi.txt						
3	northwest	NW	Charles Main	3.0	.98	3	34
4	western	WE	Sharon Gray	5.3	.97	5	23
5	southwest	SW	Lewis Dalsass	2.7	.8	2	18
6	southern	SO	Suan Chin	5.1	.95	4	15
7	southeast	SE	Patricia Hemenway	4.0	.7	4	17
8	Hello,world!						
9	-----						
10	northeast	NE	AM Main Jr.	5.1	.94	3	13
11	north	NO	Margot Weber	4.5	.89	5	9
12	central	CT	Ann Stephens	5.7	.94	5	13

说明：c 命令是修改命令。该命令将完整地修改在模式缓冲区行的当前行。如果模式 eastern 被匹配，c 命令将其后的文本替换包含 eastern 的行。

获取下一行：n 命令

n 命令表示下一条命令。sed 使用该命令获取输入文件的下一行，并将其读入到模式缓冲区中，任何 sed 命令都将应用到匹配行，紧接着的下一行上。

案例 19:

1	[root@Gin scripts]# sed '/eastern/{n;s/AM/Archie;/}' ceshi.txt						
2	northwest	NW	Charles Main	3.0	.98	3	34
3	western	WE	Sharon Gray	5.3	.97	5	23
4	southwest	SW	Lewis Dalsass	2.7	.8	2	18
5	southern	SO	Suan Chin	5.1	.95	4	15
6	southeast	SE	Patricia Hemenway	4.0	.7	4	17
7	eastern	EA	TB Savage	4.4	.84	5	20
8	northeast	NE	Archie Main Jr.	5.1	.94	3	13
9	north	NO	Margot Weber	4.5	.89	5	9
10	central	CT	Ann Stephens	5.7	.94	5	13

说明：如果在某一行匹配到模式 eastern，n 命令就指示 sed 用下一个输入行（即包含 AM Main Jr 的那行）替换模式空间中的当前行，并用 Archie 替换 AM，然后打印该行，再继续往下处理

转换：y, 命令

y 命令表示转换。该命令与 tr 命令相似，字符按照一对一的方式从左到右进行转换。例如 y/abc/ABC/, 会把小写字母转换成大写字母，a-->A,b-->B,c-->C。

案例 20:

1	[root@Gin scripts]# sed '1,3y/abcdefghijklmnopqrstuvwxyz/ABCDEFGHIJKLMNOPQRSTUVWXYZ'						
2	NORTHWEST	NW	CHARLES MAIN	3.0	.98	3	34
3	WESTERN	WE	SHARON GRAY	5.3	.97	5	23

```

4 | SOUTHWEST      SW      LEWIS DALSASS  2.7      .8      2      18
5 | southern       SO      Suan Chin      5.1      .95     4      15
6 | southeast      SE      Patricia Hemenway  4.0      .7      4      17
7 | eastern        EA      TB Savage      4.4      .84     5      20
8 | northeast      NE      AM Main Jr.    5.1      .94     3      13
9 | north          NO      Margot Weber   4.5      .89     5      9
10| central        CT      Ann Stephens   5.7      .94     5      13

```

说明：y 命令把 1~3 行中的所有的小写命令字母都转换成了大写。正则表达式元字符对 y 命令不起作用。与替分隔符一样，斜杠可以被替换成其他字符。

退出：q 命令

q 命令表示退出命令。该命令将导致 sed 程序退出，且不再进行其他的处理。

案例 21：

```

1 | [root@Gin scripts]# sed '5q' ceshi.txt
2 | northwest      NW      Charles Main    3.0      .98     3      34
3 | western        WE      Sharon Gray     5.3      .97     5      23
4 | southwest      SW      Lewis Dalsass   2.7      .8      2      18
5 | southern       SO      Suan Chin      5.1      .95     4      15
6 | southeast      SE      Patricia Hemenway  4.0      .7      4      17

```

说明：打印完第 5 行之后，q 让 sed 程序退出。

案例 22：

```

1 | [root@Gin scripts]# sed '/Lewis/{ s/Lewis/Joseph;q; }' ceshi.txt
2 | northwest      NW      Charles Main    3.0      .98     3      34
3 | western        WE      Sharon Gray     5.3      .97     5      23
4 | southwest      SW      Joseph Dalsass  2.7      .8      2      18

```

说明：在某行匹配到模式 Lewis 时，s 表示先用 Joseph 替换 Lewis，然后 q 命令让 sed 退出。

生产环境案例

在实际生产中，在修改配置文件的时候，有一些空格、空行、带 “#” 开头的注释都要删除或替换，下面为大家介绍几个实用的例子

案例 23：

```

1 | [root@Gin scripts]# cat sed.txt
2 | today is nice day
3 | you can walk out on the street
4 | it will be import to you
5 | ##每行的前面都有空格

1 | [root@Gin scripts]# sed 's/^[ ]*//' sed.txt
2 | today is nice day
3 | you can walk out on the street
4 | it will be import to you
5 | ## 注：[ ]里面有个空格
6 | 或者：
7 | [root@Gin scripts]# sed 's/^[[:space:]]*//' sed.txt
8 | today is nice day
9 | you can walk out on the street
10| it will be import to you

```

案例24：删除文本中空行和空格组成的行及#号注释的行

```
1 [root@Gin scripts]# grep -Eiv "^#|^$" ssh_config
2 Host *
3     GSSAPIAuthentication yes
4     ForwardX11Trusted yes
5     SendEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_M
6     SendEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
7     SendEnv LC_IDENTIFICATION LC_ALL LANGUAGE
8     SendEnv XMODIFIERS
```

案例 25: 从 Google 上下载下来的配置文件往往都带有数字, 现在需要删除所有行的首数字。

```
1 [root@Gin scripts]# cat sed.txt
2 1today is nice day
3 2you can walk out on the street
4 3it will be import to you
5 [root@Gin scripts]# sed 's/^[0-9][0-9]*//g' sed.txt
6 today is nice day
7 you can walk out on the street
8 it will be import to you
```

总结

本章内容总结了 sed 命令的用法, 前面部分是 sed 命令的语法, 后面部分则主要以实际案例来说明 sed 的用法, 最后面一点介绍了 sed 命令在生产实践中的运用。所谓学为练, 练为战, 希望大家能够将 sed 命令勤加练习, 必将会在工作中有所用途, 尤其是频繁的分析日志文件, Awk+Sed 是比较好的组合。最后希望本文对大家有所帮助, 真正达到熟练的程度这就靠大家在工作中归纳总结了。

分类: [linux](#)

标签: [linux命令总结](#), [sed命令](#)

好文要顶

关注我

收藏该文



琴酒网络

关注 - 0

粉丝 - 78

+加关注

« 上一篇: [linux常见故障一: linux 文件系统变只读](#)

» 下一篇: [Centos6.6系统root用户密码恢复案例](#)

4

0

posted @ 2017-02-07 23:06 [琴酒网络](#) 阅读(35460) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【活动】看雪2019安全开发者峰会, 共话安全领域焦点

【培训】Java程序员年薪40W, 他1年走了别人5年的路

相关博文:

· [Linuxsed命令总结](#)

- sed命令详解
- sed 命令详解
- sed命令详解
- sed命令

最新新闻:

- 装备 “GPS” 导航的神奇微生物
- 铁打的百度，流水的二把手
- 台积电包揽全球所有5G调制解调器代工订单
- 麻疹 “死灰复燃” 疫苗是迄今 “最佳武器”
- 李彦宏：向海龙是私人原因离职 接任者沈抖能力很强
- » 更多新闻...

Copyright ©2019 琴酒网络