

linux sed命令详解

简介

sed 是一种在线编辑器，它一次处理一行内容。处理时，把当前处理的行存储在临时缓冲区中，称为“模式空间”（pattern space），接着用sed命令处理缓冲区中的内容，处理完成后，把缓冲区的内容送往屏幕。接着处理下一行，这样不断重复，直到文件末尾。文件内容并没有 改变，除非你使用重定向存储输出。Sed主要用来自动编辑一个或多个文件；简化对文件的反复操作；编写转换程序等。

sed使用参数



```
[root@www ~]# sed [-nefr] [动作]
```

选项与参数：

- n：使用安静(silent)模式。在一般 sed 的用法中，所有来自 STDIN 的数据一般都会被列出到终端上。但如果加上 -n 参数后，则只有经过sed 特殊处理的那一行(或者动作)才会被列出来。
- e：直接在命令列模式上进行 sed 的动作编辑；
- f：直接将 sed 的动作写在一个文件内，-f filename 则可以运行 filename 内的 sed 动作；
- r：sed 的动作支持的是延伸型正规表示法的语法。(默认是基础正规表示法语法)
- i：直接修改读取的文件内容，而不是输出到终端。

动作说明：[n1[,n2]]function

n1, n2：不见得会存在，一般代表『选择进行动作的行数』，举例来说，如果我的动作是需要在 10 到 20 行之间进行的，则『10,20[动作行为]』

function：

- a：新增，a 的后面可以接字符串，而这些字符串会在新的一行出现(目前的下一行)~
- c：取代，c 的后面可以接字符串，这些字符串可以取代 n1,n2 之间的行！
- d：删除，因为是删除啊，所以 d 后面通常不接任何咚咚；
- i：插入，i 的后面可以接字符串，而这些字符串会在新的一行出现(目前的上一行)；
- p：列印，亦即将某个选择的数据印出。通常 p 会与参数 sed -n 一起运行~
- s：取代，可以直接进行取代的工作哩！通常这个 s 的动作可以搭配正规表示法！例如 1,20s/old/new/g 就是啦！



以行为单位的新增/删除

将 /etc/passwd 的内容列出并且列印行号，同时，请将第 2~5 行删除！

```
[root@www ~]# nl /etc/passwd | sed '2,5d'
1 root:x:0:0:root:/root:/bin/bash
6 sync:x:5:0:sync:/sbin:/bin/sync
7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
.....(后面省略).....
```

sed 的动作为 '2,5d'，那个 d 就是删除！因为 2-5 行给他删除了，所以显示的数据就没有 2-5 行罗~ 另外，注意一下，原本应该是要下达 sed -e 才对，没有 -e 也行啦！同时也要注意的，sed 后面接的动作，请务必以 " 两个单引号括住喔！

只要删除第 2 行

```
nl /etc/passwd | sed '2d'
```

要删除第 3 到最后一行

```
nl /etc/passwd | sed '3,$d'
```

在第二行后(亦即是加在第三行)加上『drink tea?』字样!

```
[root@www ~]# nl /etc/passwd | sed '2a drink tea'
1 root:x:0:0:root:/root:/bin/bash
2 bin:x:1:1:bin:/bin:/sbin/nologin
drink tea
3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
..... (后面省略) .....
```

那如果是要在第二行前

```
nl /etc/passwd | sed '2i drink tea'
```

如果是要增加两行以上, 在第二行后面加入两行字, 例如『Drink tea or』与『drink beer?』



```
[root@www ~]# nl /etc/passwd | sed '2a Drink tea or .....\'
> drink beer ?'
1 root:x:0:0:root:/root:/bin/bash
2 bin:x:1:1:bin:/bin:/sbin/nologin
Drink tea or .....
drink beer ?
3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
..... (后面省略) .....
```



每一行之间都必须要以反斜杠『\』来进行新行的添加喔! 所以, 上面的例子中, 我们可以发现在第一行的最后面就有 \ 存在。

以行为单位的替换与显示

将第2-5行的内容取代成为『No 2-5 number』呢?

```
[root@www ~]# nl /etc/passwd | sed '2,5c No 2-5 number'
1 root:x:0:0:root:/root:/bin/bash
No 2-5 number
6 sync:x:5:0:sync:/sbin:/bin/sync
..... (后面省略) .....
```

透过这个方法我们就能够将数据整行取代了!

仅列出 /etc/passwd 文件内的第 5-7 行

```
[root@www ~]# nl /etc/passwd | sed -n '5,7p'
5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
6 sync:x:5:0:sync:/sbin:/bin/sync
7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
```

可以透过这个 sed 的以行为单位的显示功能， 就能够将某一个文件内的某些行号选择出来显示。

数据的搜寻并显示

搜索 /etc/passwd有root关键字的行



```
nl /etc/passwd | sed '/root/p'
1 root:x:0:0:root:/root:/bin/bash
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/bin/sh
3 bin:x:2:2:bin:/bin:/bin/sh
4 sys:x:3:3:sys:/dev:/bin/sh
5 sync:x:4:65534:sync:/bin:/bin/sync
....下面忽略
```



如果root找到，除了输出所有行，还会输出匹配行。

使用-n的时候将只打印包含模板的行。

```
nl /etc/passwd | sed -n '/root/p'
1 root:x:0:0:root:/root:/bin/bash
```

数据的搜寻并删除

删除/etc/passwd所有包含root的行，其他行输出

```
nl /etc/passwd | sed '/root/d'
2 daemon:x:1:1:daemon:/usr/sbin:/bin/sh
3 bin:x:2:2:bin:/bin:/bin/sh
....下面忽略
#第一行的匹配root已经删除了
```

数据的搜寻并执行命令

找到匹配模式eastern的行后，

搜索/etc/passwd,找到root对应的行，执行后面花括号中的一组命令，每个命令之间用分号分隔，这里把bash替换为blueshell，再输出这行：

```
nl /etc/passwd | sed -n '/root/{s/bash/blueshell/;p}'  
1 root:x:0:0:root:/root:/bin/blueshell
```

如果只替换/etc/passwd的第一个bash关键字为blueshell，就退出

```
nl /etc/passwd | sed -n '/bash/{s/bash/blueshell/;p;q}'  
1 root:x:0:0:root:/root:/bin/blueshell
```

最后的q是退出。

数据的搜寻并替换

除了整行的处理模式之外，sed 还可以用行为单位进行部分数据的搜寻并取代。基本上 sed 的搜寻与替代的与 vi 相当的类似！他有点像这样：

```
sed 's/要被取代的字串/新的字串/g'
```

先观察原始信息，利用 /sbin/ifconfig 查询 IP

```
[root@www ~]# /sbin/ifconfig eth0  
eth0 Link encap:Ethernet HWaddr 00:90:CC:A6:34:84  
inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0  
inet6 addr: fe80::290:ccff:fea6:3484/64 Scope:Link  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
..... (以下省略) .....
```

本机的ip是192.168.1.100。

将 IP 前面的部分予以删除

```
[root@www ~]# /sbin/ifconfig eth0 | grep 'inet addr' | sed 's/^.*addr://g'  
192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0
```

接下来则是删除后续的部分，亦即： 192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0

将 IP 后面的部分予以删除

```
[root@www ~]# /sbin/ifconfig eth0 | grep 'inet addr' | sed 's/^.*addr://g' | sed  
's/Bcast.*$/g'  
192.168.1.100
```

多点编辑

一条sed命令，删除/etc/passwd第三行到末尾的数据，并把bash替换为blueshell

```
nl /etc/passwd | sed -e '3,$d' -e 's/bash/blueshell/'  
1 root:x:0:0:root:/root:/bin/blueshell  
2 daemon:x:1:1:daemon:/usr/sbin:/bin/sh
```

-e表示多点编辑，第一个编辑命令删除/etc/passwd第三行到末尾的数据，第二条命令搜索bash替换为blueshell。

直接修改文件内容(危险动作)

sed 可以直接修改文件的内容，不必使用管道命令或数据流重导向！不过，由於这个动作会直接修改到原始的文件，所以请你千万不要随便拿系统配置来测试！我们还是使用下载的 regular_express.txt 文件来测试看看吧！

利用 sed 将 regular_express.txt 内每一行结尾若为 . 则换成 !

```
[root@www ~]# sed -i 's/\.$/\!/g' regular_express.txt
```

利用 sed 直接在 regular_express.txt 最后一行加入『# This is a test』

```
[root@www ~]# sed -i '$a # This is a test' regular_express.txt
```

由於 \$ 代表的是最后一行，而 a 的动作是新增，因此该文件最后新增『# This is a test』！

sed 的『-i』选项可以直接修改文件内容，这功能非常有帮助！举例来说，如果你有一个 100 万行的文件，你要在第 100 行加某些文字，此时使用 vim 可能会疯掉！因为文件太大了！那怎办？就利用 sed 啊！透过 sed 直接修改/取代的功能，你甚至不需要使用 vim 去修订！

参考 http://vbird.dic.ksu.edu.tw/linux_basic/0330regex_2.php#sed

<http://www.cnblogs.com/stephen-liu74/archive/2011/11/17/2245130.html>

分类: [Linux/Unix](#)

好文要顶

关注我

收藏该文



ggjucheng

关注 - 6

粉丝 - 749

+加关注

6

0

(请您对文章做出评价)

« 上一篇: [linux grep命令详解](#)

» 下一篇: [linux sort,uniq,cut,wc命令详解](#)

posted on 2013-01-13 01:40 [ggjucheng](#) 阅读(110458) 评论(8) [编辑](#) [收藏](#)

站长统计