

# 【Linux篇】sed命令详解

5月前 作者：傻啦猫@\_@ 分类： 博客文章 阅读(165) 原文 违法举报

## 文章目录

- sed命令详解
- 一、基本语法
- 二、实操案例
  - 内部命令选项说明
  - 命令选项说明
  - 标志
- 三、sed小技巧

## sed命令详解

sed是一种流编辑器，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”，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。



## 一、基本语法

```
sed [options] '{command}[flags]' [filename]
# 中括号内容必有 大括号内容可有可无
sed # 执行命令
[options] # 命令选项
{command}[flags] # sed内部选项和参数
[filename] # 文件
```

### 命令选项

- e script 将脚本中指定的命令添加到处理输入时执行的命令中，多条件，一行中要有多个操作
- f script 将文件中指定的命令添加到处理输入时执行的命令中
- n 抑制自动输出
- i 编辑文件内容
- i.bak 修改时同时创建.bak备份文件。

-r 使用扩展的正则表达式  
! 取反（跟在模式条件后与shell有所区别）

sed常用内部命令

a 在匹配后面添加  
i 在匹配前面添加  
p 打印  
d 删除  
s 查找替换  
c 更改  
y 转换 N D P

flags

数字 表示新文本替换的模式  
g: 表示用新文本替换现有文本的全部实例  
p: 表示打印原始的内容  
w filename: 将替换的结果写入文件

## 二、实操案例

```
sed -i '/<\java-config>/i \ <jvm-options>-Dlog4j2.formatMsgNoLookups=true</jvm-options>' ${BES_HOME}/conf/server.config  
if [ -n "$jvmOptionsXmx" ];then  
    sed -i "$(sed -n '/-Xmx/' ${BES_HOME}/conf/server.config)"c\ $jvmOptionsXmx" ${BES_HOME}/conf/server.config  
fi  
if [ -n "$jvmOptionsXms" ];then  
    sed -i "$(sed -n '/-Xms/' ${BES_HOME}/conf/server.config)"c\ $jvmOptionsXms" ${BES_HOME}/conf/server.config  
fi
```

# 新加一行

```
sed -i '/<\java-config>/i \ <jvm-options>-Dlog4j2.formatMsgNoLookups=true</jvm-options>'  
${BES_HOME}/conf/server.config
```

# 判断环境变量（容器化环境）中是否有参数，有则替换

```
if [ -n "$jvmOptionsXmx" ];then  
    sed -i "$(sed -n '/-Xmx/' ${BES_HOME}/conf/server.config)"c\ $jvmOptionsXmx" ${BES_  
HOME}/conf/server.config  
fi
```

```
if [ -n "$jvmOptionsXms" ];then  
    sed -i "$(sed -n '/-Xms/' ${BES_HOME}/conf/server.config)"c\ $jvmOptionsXms" ${BES_HOME}/conf/server.config  
fi
```

c6wwf

```
<jvm-options>-Djavax.net.ssl.certificateFile=${com.bes.instanceRoot}/conf/security/cert.pem</jvm-options>
<jvm-options>-Djavax.net.ssl.certificateKeyFile=${com.bes.instanceRoot}/conf/security/key.pem</jvm-options>
<jvm-options>-Djavax.net.ssl.certificateChainFile=${com.bes.instanceRoot}/conf/security/chain.pem</jvm-options>
<jvm-options>-Djavax.xml.stream.XMLInputFactory=com.bes.wstx.stax.WstxInputFactory</jvm-options>
<jvm-options>-Xloggc:${com.bes.instanceRoot}/logs/gc.log_%t</jvm-options>
<jvm-options>-XX:+PrintGCDateStamps</jvm-options>
<jvm-options>-XX:+PrintGCDetails</jvm-options>
<jvm-options>-Dcom.sun.management.jmxremote.rmi.port=22341</jvm-options>
<jvm-options>-Djava.rmi.server.hostname=0.0.0.0</jvm-options>
<jvm-options>-Dwebgate.was.instance.home=/bes/BES</jvm-options>
<jvm-options>-Dwebgate.agent.name=sams-app-app-84cccbd97-c6wwf_BES_Con</jvm-options>
<jvm-options>-javaagent:/bes/apm/webgate-agent/lib/webgate-agent.jar</jvm-options>
<jvm-options>-Dshared.loader=/bes/BES/libExt/*.jar</jvm-options>
<jvm-options>-javaagent:/bes/BES/prometheus_monitor/jmx_prometheus_javaagent-0.15.0.jar=19999:/bes/BES/prometheus_monit
<jvm-options>-Duser.timezone=GMT+8</jvm-options>
<jvm-options>-Dfile.encoding=GBK</jvm-options>
<jvm-options>-Dlog4j2.formatMsgNoLookups=true</jvm-options>
<jvm-options>-XX:+UseG1GC</jvm-options>
</java-config>
```

CSDN @傻啦猫@\_@

这个案例是我们项目中实际用到的，效果就是配置JVM参数。具体学习看下面的即可。

## 准备数据

```
[root@jiangnan ~]# vim sed.txt
[root@jiangnan ~]# cat sed.txt
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
3 the quick brown fox jumps over the lazy dog.
4 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
```

## 内部命令选项说明

文件内容增加操作，将数据追加到某个位置之后，使用命令 **a**。

### 1. 在指定行后面追加内容

在第二到四行每行后新开一行追加数据: append data "haha"

```
[root@jiangnan data]# sed '2,4a\append data "haha"' sed.txt
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
append data "haha"
3 the quick brown fox jumps over the lazy dog.
append data "haha"
4 the quick brown fox jumps over the lazy dog.
append data "haha"
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

说明：a表示在匹配的行后面新开一行新增内容。2,4a表示第2到4行，如果不写数字，表示所有行。

## 2. 匹配字符串追加

找到包含"3 the"的行，在其后新开一行追加内容: append data "haha"

```
[root@jiangnan data]# sed '/3 the/a\append data "haha"' sed.txt
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
3 the quick brown fox jumps over the lazy dog.
append data "haha"
4 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

这里注意：/要匹配的字符串/

这种方法比较常用，匹配指定的字符串然后进行增加，比如XML文件等。

**文件内容增加操作，将数据插入到某个位置之前，使用命令 i。**

## 3. 匹配字符串插入

找到包含"3 the"的行，在其前新开一行插入内容: insert data "haha"

```
[root@jiangnan data]# sed '/3 the/i\append data "haha"' sed.txt
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
append data "haha"
3 the quick brown fox jumps over the lazy dog.
4 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

这里注意：/要匹配的字符串/

使用方法同a类似。

**文件内容修改操作—替换,将一行中匹配的内容替换为新的数据，使用命令s。**

## 4. 在指定行后面追加内容

将sed.txt 中第二到第四行的dog替换为cat

```
[root@jiangnan data]# sed '2,4s/dog/cat/' sed.txt
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy cat.
3 the quick brown fox jumps over the lazy cat.
4 the quick brown fox jumps over the lazy cat.
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

## 5. 匹配字符串所在行指定字符的替换

将包含字符串"3 the"的行中的dog替换为cat

```
[root@jiangnan data]# sed '/3 the/s/dog/cat/' sed.txt
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
3 the quick brown fox jumps over the lazy cat.
4 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

这里注意：/匹配字符串所在的行/s/指定匹配字符串/要替换的字符串/

文件内容修改操作—更改,将一行中匹配的内容替换为新的数据,使用命令c。

## 6. 匹配行整行的替换

将sed.txt 文件中的第二、三、四行的内容更改为：change data “haha”

```
[root@jiangnan data]# sed '2,4c\change data haha' sed.txt
1 the quick brown fox jumps over the lazy dog.
change data haha
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

## 7. 匹配字符串所在行整行的替换

将sed.txt文件中包含"3 the"的行内容更改为: change data “haha”

```
[root@jiangnan data]# sed '/3 the/c\change data haha' sed.txt
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
change data haha
4 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

文件内容修改操作—字符转换,将一行中匹配的内容替换为新的数据,使用命令y。

## 8. 将sed.txt中的a b c字符转换为对应的 A B C字符

```
[root@jiangnan data]# sed 'y/abc/ABC/' sed.txt
1 the quiCk Brown fox jumps over the lAzY dog.
2 the quiCk Brown fox jumps over the lAzY dog.
3 the quiCk Brown fox jumps over the lAzY dog.
4 the quiCk Brown fox jumps over the lAzY dog.
5 the quiCk Brown fox jumps over the lAzY dog.
[root@jiangnan data]#
```

文件内容删除，将文件中的指定数据删除，使用命令d。

#### 9. 删除文件sed.txt 第三到第四行的数据

```
[root@jiangnan data]# sed '3,4d' sed.txt
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

文件内容查看，将文件内容输出到屏幕，使用命令p。

#### 10. 打印sed.txt的内容

```
[root@jiangnan data]# sed 'p' sed.txt
1 the quick brown fox jumps over the lazy dog.
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
3 the quick brown fox jumps over the lazy dog.
3 the quick brown fox jumps over the lazy dog.
4 the quick brown fox jumps over the lazy dog.
4 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

#### 11. 打印sed.txt文件包含字符串"3 the"的行

```
[root@jiangnan data]# sed '/3 the/p' sed.txt
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
3 the quick brown fox jumps over the lazy dog.
3 the quick brown fox jumps over the lazy dog.
4 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

现象：可以看得出，打印内容是重复的行，原因是打印了指定文件内容一次，又将读入缓存的所有数据打印了一次，所以会看到这样的效果， 如果不想看到这样的结果，可以加命令选项-n抑制内存输出即可。

```
[root@jiangnan data]# sed -n '/3 the/p' sed.txt
3 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

## 命令选项说明

在命令行中使用多个命令 -e

## 1. 将brown替换为green dog替换为cat

```
[root@jiangnan data]# sed -e 's/brown/green;;s/dog/cat/' sed.txt
1 the quick green fox jumps over the lazy cat.
2 the quick green fox jumps over the lazy cat.
3 the quick green fox jumps over the lazy cat.
4 the quick green fox jumps over the lazy cat.
5 the quick green fox jumps over the lazy cat.
[root@jiangnan data]#
```

## 从文件读取编辑器命令 -f 适用于日常重复执行的场景

```
1) 将命令写入文件
[root@jiangnan data]# vim abc.txt
[root@jiangnan data]# cat abc.txt
s/brown/green/
s/dog/cat/
s/fox/elephant/
[root@jiangnan data]#

2) 使用-f命令选项调用命令文件
[root@jiangnan ~]# sed -f abc.txt sed.txt
1 the quick green elephant jumps over the lazy cat.
2 the quick green elephant jumps over the lazy cat.
3 the quick green elephant jumps over the lazy cat.
4 the quick green elephant jumps over the lazy cat.
5 the quick green elephant jumps over the lazy cat.
[root@jiangnan data]#
```

## 抑制内存输出 -n

### 2. 打印sed.txt文件的第二行到最后一行内容 \$最后的意思

```
[root@jiangnan data]# sed -n '2,$p' sed.txt
2 the quick brown fox jumps over the lazy dog.
3 the quick brown fox jumps over the lazy dog.
4 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

## 使用正则表达式 -r

### 3. 打印sed.txt中以字符串"3 the"开头的行内容

```
[root@jiangnan data]# sed -n -r '/^(3 the)/p' sed.txt
3 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

正则	说明	备注
/key/	查询包含关键字的行	sed -n '/root/p' 1.txt
/key1/,/key2/	匹配包含两个关键字之间的行	sed -n '/^adm/,/^mysql/p' 1.txt
/key/,x	从匹配关键字的行开始到文件第x行之间的行（包含关键字所在行）	sed -n '/^ftp/,7p'
x,/key/	从文件的第x行开始到与关键字的匹配行之间的行	
x,y!	不包含x到y行	
/key/!	不包括关键字的行	sed -n '/bash\$/p' 1.txt

说明：在正则表达式中如果出现特殊字符(^\$.\*/[]),需要以前导“\”号做转义

**注意：**做完了上面的案例，但是我们发现源文件并没有任何改变，原因就是sed操作的是缓存区里的内容，对我们的源文件没有做改变。如果需要修改文件内容可以直接使用-i命令选项，-i命令选项提供了备份功能，比如参数使用-i.bak，那么在修改源文件的同时会先备份一个以.bak结尾的源文件，然后再进行修改操作。

```
[root@jiangnan data]# sed -i.bak 's/brown/green/' sed.txt
[root@jiangnan data]# cat sed.txt
1 the quick green fox jumps over the lazy dog.
2 the quick green fox jumps over the lazy dog.
3 the quick green fox jumps over the lazy dog.
4 the quick green fox jumps over the lazy dog.
5 the quick green fox jumps over the lazy dog.
[root@jiangnan data]# cat sed.txt.bak
1 the quick brown fox jumps over the lazy dog.
2 the quick brown fox jumps over the lazy dog.
3 the quick brown fox jumps over the lazy dog.
4 the quick brown fox jumps over the lazy dog.
5 the quick brown fox jumps over the lazy dog.
[root@jiangnan data]#
```

## 标志

准备数据

```
[root@jiangnan data]# vi sed2.txt
[root@jiangnan data]# cat sed2.txt
1 the quick brown fox jumps over the lazy dog . dog
2 the quick brown fox jumps over the lazy dog . dog
3 the quick brown fox jumps over the lazy dog . dog
4 the quick brown fox jumps over the lazy dog . dog
5 the quick brown fox jumps over the lazy dog . dog
[root@jiangnan data]#
```

**数字标志：**此标志是一个非零正数，默认情况下，执行替换的时候，如果一行中有多个符合的字符串，如果没有标志位定义，那么只会替换第一个字符串，其他的就被忽略掉了，为了能精确替



换，可以使用数字位做定义。

#### 1. 替换一行中的第二处dog为cat

```
[root@jiangnan data]# sed 's/dog/cat/2' sed2.txt
1 the quick brown fox jumps over the lazy dog . cat
2 the quick brown fox jumps over the lazy dog . cat
3 the quick brown fox jumps over the lazy dog . cat
4 the quick brown fox jumps over the lazy dog . cat
5 the quick brown fox jumps over the lazy dog . cat
[root@jiangnan data]# sed 's/dog/cat/' sed2.txt
1 the quick brown fox jumps over the lazy cat . dog
2 the quick brown fox jumps over the lazy cat . dog
3 the quick brown fox jumps over the lazy cat . dog
4 the quick brown fox jumps over the lazy cat . dog
5 the quick brown fox jumps over the lazy cat . dog
[root@jiangnan data]#
```

对比有标志和没有标志的区别

**g标志:**将一行中的所有符合的字符串全部执行替换

#### 2. 将包含3 the所在行中的dog替换为cat

```
[root@jiangnan data]# sed '/3 the/s/dog/cat/g' sed2.txt
1 the quick brown fox jumps over the lazy dog . dog
2 the quick brown fox jumps over the lazy dog . dog
3 the quick brown fox jumps over the lazy cat . cat
4 the quick brown fox jumps over the lazy dog . dog
5 the quick brown fox jumps over the lazy dog . dog
[root@jiangnan data]#
```

‘g’表示global，全部替换

**p标志:**打印文本内容，类似于-p命令选项

```
[root@jiangnan data]# sed '3s/dog/cat/p' sed2.txt
1 the quick brown fox jumps over the lazy dog . dog
2 the quick brown fox jumps over the lazy dog . dog
3 the quick brown fox jumps over the lazy cat . dog
3 the quick brown fox jumps over the lazy cat . dog
4 the quick brown fox jumps over the lazy dog . dog
5 the quick brown fox jumps over the lazy dog . dog
[root@jiangnan data]#
```

**w filename标志:**将修改的内容存入filename文件中

```
[root@jiangnan data]# sed '3s/dog/cat/w sed3.txt' sed2.txt
1 the quick brown fox jumps over the lazy dog . dog
2 the quick brown fox jumps over the lazy dog . dog
```

```
3 the quick brown fox jumps over the lazy cat . dog
4 the quick brown fox jumps over the lazy dog . dog
5 the quick brown fox jumps over the lazy dog . dog
[root@jiangnan data]# cat sed3.txt
3 the quick brown fox jumps over the lazy cat . dog
[root@jiangnan data]#
```

可以看出，将修改的第三行内容存在了sed3.txt文件中

### 三、sed小技巧

统计data2有多少行

```
[root@jiangnan ~]# sed -n '$=' data2
```

5

打印data2内容时加上行号

```
[root@jiangnan ~]# sed '=' data2
```

1

1 the quick brown fox jumps over the lazy dog . dog

2

2 the quick brown fox jumps over the lazy dog . dog

3

3 the quick brown fox jumps over the lazy dog . dog

4

4 the quick brown fox jumps over the lazy dog . dog

5

5 the quick brown fox jumps over the lazy dog . dog

输出包含the的所在行的行号 (= 用来输出行号)

```
[root@jiangnan ~]# sed -n '/the/= ' test.txt
```

输出以PI开头的行

```
[root@jiangnan ~]# sed -n '/^PI/p' test.txt
```

输出以数字结尾的行

```
[root@jiangnan ~]# sed -n '/[0-9]$/p' test.txt
```

输出包含单词wood的行 \< ,\>表示单词边界

```
[root@jiangnan ~]# sed -n '/\<wood\>/p' test.txt
```

a wood cross!

每行开始添加#字符

```
[root@jiangnan ~]# sed 's/^/#/' test.txt
```

在包含the的每行行首添加#字符

```
[root@jiangnan ~]# sed '/the/s/^/#/' test.txt
```

在每行末尾添加EOF字符

```
[root@jiangnan ~]# sed 's/$/EOF/' test.txt
```

将3-5行所有的the替换为THE

```
[root@jiangnan ~]# sed '3,5s/the/THE/g' test.txt
```

将包含the的行中的o替换为0

```
[root@jiangnan ~]# sed '/the/s/o/0/g' test.txt
```

迁移符合条件的文本

H 复制到剪贴板；

g, G 将剪贴板中的数据覆盖/追加到指定行；

w保存为文件；

r读取指定文件；

a 追加指定内容

将包含the的行迁移到行尾（；用于多个操作）

H复制到剪贴板---d删除---\$G追加到行尾

```
[root@jiangnan ~]# sed '/the/{H;d};$G' test.txt
```

将1-5行迁移到17行后

```
[root@jiangnan ~]# sed '1,5{H;d};17G' test.txt
```

将包含the的行另存为新文件

```
[root@jiangnan ~]# sed '/the/w out.file' test.txt
```

```
[root@jiangnan ~]# ls
```

```
anaconda-ks.cfg  out.file  test.txt  :wq
```

```
[root@jiangnan ~]# cat out.file
```

在包含the每行后添加文件hostname内容

```
[root@jiangnan ~]# sed '/the/r /etc/hostname' test.txt
```

在第3行后插入新行，内容为New

```
[root@jiangnan ~]# sed '3aNew' test.txt
```

在包含the的每行后插入新行

```
[root@jiangnan ~]# sed '/the/aNew' test.txt
```

在第3行后插入多行（\n 换行符）

```
[root@jiangnan ~]# sed '3aNew1\nNew2' test.txt
```

```
ngnan ~]# sed '1,5{H;d};17G' test.txt
```

将包含the的行另存为新文件

```
[root@jiangnan ~]# sed '/the/w out.file' test.txt
```

```
[root@jiangnan ~]# ls
```

```
anaconda-ks.cfg  out.file  test.txt  :wq
```

```
[root@jiangnan ~]# cat out.file
```

在包含the每行后添加文件hostname内容

```
[root@jiangnan ~]# sed '/the/r /etc/hostname' test.txt
```

在第3行后插入新行，内容为New

```
[root@jiangnan ~]# sed '3aNew' test.txt
```

在包含the的每行后插入新行

```
[root@jiangnan ~]# sed '/the/aNew' test.txt
```

在第3行后插入多行（\n 换行符）

```
[root@jiangnan ~]# sed '3aNew1\nNew2' test.txt
```

微信公众号先已开通，搜索“江小南和他的小伙伴们”就能找到我哦，各位小伙伴们可以关注一下，文章会进行同步更新，方便查看哦。

**版权声明：**本文内容来源于网络，版权归原作者所有，此博客不拥有其著作权，亦不承担相应法律责任。文本页已经标记具体来源原文地址，请点击[原文](#)查看来源网址,站内文章以及资源内容站长不承诺其正确性，如侵犯了您的权益，请联系站长如有侵权请联系站长，将立刻删除

分享到：



¥ 赞助服务器费用

领支付宝红包

**使用ChatGPT 写代码/写文案/优化产品**

支付宝搜索 **767759638** 领取快餐现金

ChatGPT官网小白访问教程

**【阿里云】活动服务器 4折**

**【阿里云】学生服务器2核2G/10.14元/月**

**【腾讯云】服务器，2核2G内存3M带宽/95元/年**

**购买 ChatGPT 官网账号，独立号，可改密，不封号**

升级Plus/充值API



