sed是一种非交互式的流编辑器，通过多种转换修改流经它的文本，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可做的编辑工作包括删除、查找替换、添加、插入、从其他文件中读入数据等。

注意：要想保存修改后的文件，必须使用重定向生成新的文件。如果想直接修改源文件本身则需要使用“-i”参数。

sed命令使用的场景包括以下一些：

- 常规编辑器编辑困难的文本；
- 太过于庞大的文本，使用常规编辑器难以胜任（比如说vi一个几百兆的文件）；
- 有规律的文本修改，加快文本处理速度（比如说全文替换）；

准备文件Sed.txt来做实验：

```
[root@sunday-test ~]# cat Sed.txt
this is line 1,this is First line
this is line 2,the Second line,Empty line followed

this is line 4,this is Third line
this is line 5,this is Fifth line
```

使用sed修改文件流的方式如下：

```
sed [options] 'command' file
#options是sed可以接受的参数
#command是sed的命令集（一共有25个）
#使用-e参数和分号连接多编辑命令
```

-e参数本身只是sed的一个简单参数，表示将下一个字符串解析成sed编辑命令，一般情况下可以忽略，但是当sed需要传递多个编辑命令时该参数就不能少了。

下面的例子就是演示了在将this改为That的同时，还要将line改成LINE,两个编辑命令前都要使用-e参数，如果有更多的编辑需求，以此类推。

```
[root@sunday-test ~]# sed -e 's/this/That/g' -e 's/line/LINE/g' Sed.txt
That is LINE 1,That is First LINE
That is LINE 2,the Second LINE,Empty LINE followed

That is LINE 4,That is Third LINE
```

That is LINE 5,That is Fifth LINE

使用分号连接两个编辑命令也可以

[root@sunday-test ~]# sed 's/this/That/g ;s/line/LINE/g' Sed.txt

删除

使用d命令可删除指定的行

将文件的第一行删除后输出到屏幕

[root@sunday-test ~]# sed '1d' Sed.txt

this is line 2,the Second line,Empty line followed

this is line 4,this is Third line

this is line 5,this is Fifth line

sed默认不修改原文件，如果希望保存修改后的文件则需要用重定向

[root@sunday-test ~]# sed '1d' Sed.txt > saved_file

如果想直接修改文件，使用 ‘-i’ 参数

[root@sunday-test ~]# sed -i '1d' saved_file

[root@sunday-test ~]# cat saved_file

this is line 4,this is Third line

this is line 5,this is Fifth line

删除指定范围的行（第1行到第3行）

[root@sunday-test ~]# sed '1,3d' Sed.txt

this is line 4,this is Third line

this is line 5,this is Fifth line

删除第3行到最后行

[root@sunday-test ~]# sed '3,$d' Sed.txt

this is line 1,this is First line

this is line 2,the Second line,Empty line followed

删除最后一行：

[root@sunday-test ~]# sed '$d' Sed.txt

this is line 1,this is First line

this is line 2,the Second line,Empty line followed

this is line 4,this is Third line

删除所有包含 ‘Empty’ 的行

[root@sunday-test ~]# sed '/Empty/d' Sed.txt

this is line 1,this is First line

this is line 4,this is Third line
this is line 5,this is Fifth line
删除空行
[root@sunday-test ~]# sed '/^$/d' Sed.txt
this is line 1,this is First line
this is line 2,the Second line,Empty line followed
this is line 4,this is Third line
this is line 5,this is Fifth line
查找替换
使用s命令可将查找到的匹配文本内容替换为新的文本，默认情况只替换第一次匹配到的内容
[root@sunday-test ~]# sed 's/line/LINE/' Sed.txt
this is LINE 1,this is First line
this is LINE 2,the Second line,Empty line followed

this is LINE 4,this is Third line
this is LINE 5,this is Fifth line
如果想只替换第二个匹配到的line为LINE
[root@sunday-test ~]# sed 's/line/LINE/2' Sed.txt
this is line 1,this is First LINE
this is line 2,the Second LINE,Empty line followed

this is line 4,this is Third LINE
this is line 5,this is Fifth LINE
使用g选项，可以完成所有匹配值的替换
[root@sunday-test ~]# sed 's/line/LINE/g' Sed.txt
this is LINE 1,this is First LINE
this is LINE 2,the Second LINE,Empty LINE followed

this is LINE 4,this is Third LINE
this is LINE 5,this is Fifth LINE
将以this开头的this替换为that
[root@sunday-test ~]# sed 's/^this/that/' Sed.txt

that is line 1,this is First line
that is line 2,the Second line,Empty line followed

that is line 4,this is Third line
that is line 5,this is Fifth line
过滤出ens33网卡的IP地址和掩码信息
[root@sunday-test ~]# ifconfig ens33 | grep '\<inet\>' | sed 's/^.*inet //' | sed 's/broadcast.*$//'
192.168.5.146 netmask 255.255.255.0

字符转换
使用y命令可以进行字符转换，其作用为将一系列字符逐个地变换为另外一系列字符，基本用法如下：
以下命令会将file中的O转换为N、L转换为E、D转换为W
注意转换字符和被转换字符的长度要相等，否则sed无法执行
sed 'y/OLD/NEW' file
例如：
[root@sunday-test ~]# sed 'y/1245/ABCD/' Sed.txt
this is line A,this is First line
this is line B,the Second line,Empty line followed

this is line C,this is Third line
this is line D,this is Fifth line

插入文本
使用i或a命令插入文本，其中i代表在匹配行之前插入，而a代表在匹配行之后插入，示例如下：
使用i在第二行前插入文本
[root@sunday-test ~]# sed '2 i Insert' Sed.txt
this is line 1,this is First line
Insert
this is line 2,the Second line,Empty line followed

this is line 4,this is Third line
this is line 5,this is Fifth line

使用a在第二行后插入文本
[root@sunday-test ~]# sed '2 a Insert' Sed.txt
this is line 1,this is First line
this is line 2,the Second line,Empty line followed
Insert

this is line 4,this is Third line
this is line 5,this is Fifth line
在匹配行的上一行插入文本
[root@sunday-test ~]# sed '/Second/i\Insert' Sed.txt
this is line 1,this is First line
Insert
this is line 2,the Second line,Empty line followed

this is line 4,this is Third line
this is line 5,this is Fifth line
如果要同时新增多行，则每行之间要用反斜杠\n来进行新行的添加
[root@sunday-test ~]# sed '2 a\insert\ninsert2\ninsert3' Sed.txt
this is line 1,this is First line
this is line 2,the Second line,Empty line followed
insert
insert2
insert3

this is line 4,this is Third line
this is line 5,this is Fifth line
取代行
c命令,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
[root@sunday-test ~]# sed '2,4c this is 2-4 line' Sed.txt
this is line 1,this is First line
this is 2-4 line
this is line 5,this is Fifth line
读入文本
使用r命令可从其他文件中读取文本，并插入匹配行之后，例如：
将/etc/passwd中的内容读出放到Sed.txt空行之后

[root@sunday-test ~]# sed '/^$/r /etc/passwd' Sed.txt

打印

使用p命令可进行打印，这里使用sed命令时一般都加-n参数，表示不打印没关系的行。

不加-n参数，会输出所有行，找到的行会重复显示

[root@sunday-test ~]# sed '/the/p' Sed.txt

this is line 1,this is First line

this is line 2,the Second line,Empty line followed

this is line 2,the Second line,Empty line followed

this is line 4,this is Third line

this is line 5,this is Fifth line

加-n参数，只显示找到的行

[root@sunday-test ~]# sed -n '/the/p' Sed.txt

this is line 2,the Second line,Empty line followed

sed脚本

使用sed脚本可以加快工作效率，调用sed命令并使用-f参数指定文件

写如下脚本，作用是将全文的this改成THAT，并删除所有空行

[root@sunday-test ~]# cat sed01.rules

s/this/THAT/g

/^$/d

执行sed脚本

[root@sunday-test ~]# sed -f sed01.rules Sed.txt

THAT is line 1,THAT is First line

THAT is line 2,the Second line,Empty line followed

THAT is line 4,THAT is Third line

THAT is line 5,THAT is Fifth line

也可以这样写脚本：

[root@sunday-test ~]# cat sed02.rules

#!/usr/bin/sed -f　　//第一行是sed命令解释行。脚本在这一行查找sed以运行命令

s/this/THAT/g

/^$/d

[root@sunday-test ~]# chmod +x sed02.rules

[root@sunday-test ~]# ./sed02.rules Sed.txt

THAT is line 1,THAT is First line

THAT is line 2,the Second line,Empty line followed

THAT is line 4,THAT is Third line

THAT is line 5,THAT is Fifth line

表 10-1　sed 常用的命令

sed 命令	作　用
a	在匹配行后面加入文本
c	字符转换
d	删除行
D	删除第一行
i	在匹配行前面加入文本
h	复制模板块的内容到存储空间
H	追加模板块的内容到存储空间
g	将存储空间的内容复制到模式空间
G	将存储空间的内容追加到模式空间
n	读取下一个输入行，用下一个命令处理新的行
N	追加下一个输入行到模板块后并在二者间插入新行

sed 命令	作　用
p	打印匹配的行
P	打印匹配的第一行
q	退出 sed
r	从外部文件中读取文本
w	追加写文件
!	匹配的逆
s/old/new	用 new 替换正则表达式 old
=	打印当前行号

表 10-2　sed 常用的参数

sed 参数	作　用
-e	多条件编辑
-h	帮助信息
-n	不输出不匹配的行
-f	指定 sed 脚本
-V	版本信息
-i	直接修改原文件

表 10-3　sed 常用的正则表达式匹配

元字符	作　　用
^	匹配行的开始。如：/^cat/ 匹配所有以 cat 开头的行
$	匹配行的结束。如：/cat$/ 匹配所有以 cat 结尾的行
.	匹配任一非换行字符。如：/c.t/ 匹配 c 后接一个任意字符，然后是 t
*	匹配零个或任意多个字符。如：/*cat/ 匹配一串字符后紧跟 cat 的所有行
[]	匹配指定范围内的字符。如：/[Cc]at/ 匹配 cat 和 Cat
[^]	匹配不在指定范围内的字符。如：/[^A-Z]/ 匹配不是以大写字母开头的行
\(..\)	保存匹配的字符。如：s/\(love\)able/\1rs/，loveable 被替换成 lovers
&	保存搜索字符用来替换其他字符。如 s/love/**&**/，love 变成 **love**
\<	锚定单词的开始。如：/\<cat/ 匹配包含以 cat 开头的单词的行
\>	锚定单词的结束。如：/cat\>/ 匹配包含以 cat 结尾的单词的行
x\{n\}	重复字符 x，m 次。如：/o\{5\}/ 匹配包含 5 个 o 的行
x\{m,\}	重复字符 x，至少 m 次。如：/o\{5,\}/ 匹配至少有 5 个 o 的行
x\{n,m\}	重复字符 x，至少 m 次，不多于 n 次。如：/o\{5,10\}/ 匹配 5 到 10 个 o 的行