

sed命令详解

转载：http://blog.chinaunix.net/u/22677/showart_1076318.html

1.简介

sed是非交互式的编辑器。它不会修改文件，除非使用shell重定向来保存结果。默认情况下，所有的输出行都被打印到屏幕上。

sed编辑器逐行处理文件（或输入），并将结果发送到屏幕。具体过程如下：首先sed把当前正在处理的行保存在一个临时缓存区中（也称为模式空间），然后处理临时缓冲区中的行，完成后将该行发送到屏幕上。sed每处理完一行就将其从临时缓冲区删除，然后将下一行读入，进行处理和显示。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冲区中，对这个副本进行编辑，所以不会修改原文件。

2.定址

定址用于决定对哪些行进行编辑。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将处理输入文件的所有行。

地址是一个数字，则表示行号；是"\$"符号，则表示最后一行。例如：

```
sed -n '3p' datafile
#打印第三行
```

只显示指定行范围的文件内容，例如：

```
# 只查看文件的第100行到第200行
sed -n '100,200p' mysql_slow_query.log
```

地址是逗号分隔的，那么需要处理的地址是这两行之间的范围（包括这两行在内）。范围可以用数字、正则表达式、或二者的组合表示。例如：

```
sed '2,5d' datafile
#删除第二到第五行
sed '/My/,/You/d' datafile
#删除包含"My"的行到包含"You"的行之间的行
sed '/My/,10d' datafile
#删除包含"My"的行到第十行的内容
```

3.命令与选项

sed命令告诉sed如何处理由地址指定的各输入行，如果没有指定地址则处理所有的输入行。

3.1 sed命令

命令	功能
a\	在当前行后添加一行或多行。多行时除最后一行外，每行末尾需用"\\"续行
c\	用此符号后的新文本替换当前行中的文本。多行时除最后一行外，每行末尾需用"\\"续行
\	在当前行之前插入文本。多行时除最后一行外，每行末尾需用"\\"续行
d	删除行
h	把模式空间里的内容复制到暂存缓冲区
H	把模式空间里的内容追加到暂存缓冲区
g	把暂存缓冲区里的内容复制到模式空间，覆盖原有的内容
G	把暂存缓冲区的内容追加到模式空间里，追加在原有内容的后面
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对其的处理
q	结束或退出sed
r	从文件中读取输入行
!	对所选行以外的所有行应用命令
s	用一个字符串替换另一个
g	在行内进行全局替换
w	将所选的行写入文件
x	交换暂存缓冲区与模式空间的内容
y	将字符替换为另一字符（不能对正则表达式使用y命令）

3.2 sed选项

选项	功能
-e	进行多项编辑，即对输入行应用多条sed命令时使用
-n	取消默认的输出
-f	指定sed脚本的文件名

4.退出状态

sed不向grep一样，不管是否找到指定的模式，它的退出状态都是0。只有当命令存在语法错误时，sed的退出状态才不是0。

5.正则表达式元字符

与grep一样，sed也支持特殊元字符，来进行模式查找、替换。不同的是，sed使用的正则表达式是括在斜杠线"/"之间的模式。

如果要把正则表达式分隔符"/"改为另一个字符，比如o，只要在这个字符前加一个反斜线，在字符后跟上正则表达式，再跟上这个字符即可。例如：sed -n '\o^Myop' datafile

元字符	功能	示例
^	行首定位符	/^my/ 匹配所有以my开头的行
\$	行尾定位符	/my\$/ 匹配所有以my结尾的行
.	匹配除换行符以外的单个字符	/m..y/ 匹配包含字母m，后跟两个任意字符，再跟字母y的行
*	匹配零个或多个前导字符	/my*/ 匹配包含字母m,后跟零个或多个y字母的行
[]	匹配指定字符组内的任一字符	/[Mm]y/ 匹配包含My或my的行
[]	匹配不在指定字符组内的任一字符	/[^Mm]y/ 匹配包含y，但y之前的那个字符不是M或m的行
\(...\)	保存已匹配的字符	1,20s/\(you\)self\1r/ 标记元字符之间的模式，并将其保存为标签1，之后可以使用\1来引用它。最多可以定义9个标签，从左边开始编号，最左边的是第一个。此例中，对第1到第20行进行处理，you被保存为标签1，如果发现youself，则替换为your。
&	保存查找串以便在替换串中引用	s/my/**&*/ 符号&代表查找串。my将被替换为**my**
\<	词首定位符	/\<my/ 匹配包含以my开头的单词的行
\>	词尾定位符	/my\>/ 匹配包含以my结尾的单词的行
x\{m\}	连续m个x	/9\{5\}/ 匹配包含连续5个9的行
x\{m,\}	至少m个x	/9\{5,\}/ 匹配包含至少连续5个9的行
x\{m,n\}	至少m个，但不超过n个x	/9\{5,7\}/ 匹配包含连续5到7个9的行

6.范例

6.1 p命令

命令p用于显示模式空间的内容。默认情况下，sed把输入行打印在屏幕上，选项-n用于取消默认的打印操作。当选项-n和命令p同时出现时,sed可打印选定的内容。

```
sed '/my/p' datafile
#默认情况下，sed把所有输入行都打印在标准输出上。如果某行匹配模式my，p命令将该行另外打印一遍。
```

```
sed -n '/my/p' datafile
#选项-n取消sed默认的打印，p命令把匹配模式my的行打印一遍。
```

6.2 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空间里，然后对该行执行sed命令，最后将模式空间里的内容显示在屏幕上。如果发出的是命令d，当前模式空间里的输入行会被删除，不被显示。

```
sed '$d' datafile
#删除最后一行，其余的都被显示
```

```
sed '/my/d' datafile
#删除包含my的行，其余的都被显示
```

6.3 s命令

```
sed 's/^My/You/g' datafile
#命令末端的g表示在行内进行全局替换，也就是说如果某行出现多个My，所有的My都被替换为You。
```

```
sed -n '1,20s/My$/You/gp' datafile
#取消默认输出，处理1到20行里匹配以My结尾的行，把行内所有的My替换为You，并打印到屏幕上。
```

```
sed 's#My#Your#g' datafile
#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s命令，就成了新的串分隔符。
```

6.4 e选项

-e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冲区中的行上。

```
sed -e '1,10d' -e 's/My/Your/g' datafile
#选项-e用于进行多重编辑。第一重编辑删除第1-3行。第二重编辑将出现的所有My替换为Your。因为是逐行进行这两项编辑（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。
```

6.5 r命令

r命令是读命令。sed使用该命令将一个文本文件中的内容加到当前文件的特定位置上。

```
sed '/My/r introduce.txt' datafile
#如果在文件datafile的某一行匹配到模式My，就在该行后读入文件introduce.txt的内容。如果出现My的行不止一行，则在出现My的各行后都读入introduce.txt文件的内容。
```

6.6 w命令

```
sed -n '/hrwang/w me.txt' datafile
```

6.7 a\命令

a\命令是追加命令，追加将添加新文本到文件中当前行（即读入模式缓冲区中的行）的后面。所追加的文本行位于sed命令的下方另起一行。如果要追加的内容超过一行，则每一行都必须以反斜线结束，最后一行除外。最后一行将以引号和文件名结束。

```
sed '/^hrwang/a\
>hrwang and mifan are husband\
>and wife' datafile
#如果在datafile文件中发现匹配以hrwang开头的行，则在该行下面追加hrwang and mifan are husband and wife
```

6.8 i\命令

i\命令是在当前行的前面插入新的文本。

6.9 c\命令

sed使用该命令将已有文本修改成新的文本。

6.10 n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。

```
sed '/hrwang/{n;s/My/Your/}' datafile
```

注：如果需要使用多条命令，或者需要在某个地址范围内嵌套地址，就必须用花括号将命令括起来，每行只写一条命令，或这用分号分割同一行中的多条命令。

6.11 y命令

该命令与UNIX/Linux中的tr命令类似，字符按照一对一的方式从左到右进行转换。例如，y/abc/ABC/将所有小写的a转换成A，小写的b转换成B，小写的c转换成C。

```
sed '1,20y/hrwang12/HRWANG^$/' datafile
#将1到20行内，所有的小写hrwang转换成大写，将1转换成^,将2转换成$。
#正则表达式元字符对y命令不起作用。与s命令的分隔符一样，斜线可以被替换成其它的字符。
```

6.12 q命令

q命令将导致sed程序退出，不再进行其它的处理。

```
sed '/hrwang/{s/hrwang/HRWANG;q;}' datafile
```

6.13 h命令和g命令

```
#cat datafile

My name is hrwang.

Your name is mifan.

hrwang is mifan's husband.

mifan is hrwang's wife.

sed -e '/hrwang/h' -e '$G' datafile

sed -e '/hrwang/H' -e '$G' datafile

#通过上面两条命令，你会发现h会把原来暂存缓冲区的内容清除，只保存最近一次执行h时保存进去的模式空间的内容。而H命令则把每次匹配hrwnag的行都追加保存在暂存缓冲区。

sed -e '/hrwang/H' -e '$g' datafile

sed -e '/hrwang/H' -e '$G' datafile

#通过上面两条命令，你会发现g把暂存缓冲区中的内容替换掉了模式空间中当前行的内容，此处即替换了最后一行。而G命令则把暂存缓冲区的内容追加到了模式空间的当前行后。此处即追加到了末尾。
```

7. sed脚本

sed脚本就是写在文件中的一列sed命令。脚本中，要求命令的末尾不能有任何多余的空格或文本。如果在一行中有多个命令，要用分号分隔。执行脚本时，sed先将输入文件中第一行复制到模式缓冲区，然后对其执行脚本中所有的命令。每一行处理完毕后，sed再复制文件中下一行到模式缓冲区，对其执行脚本中所有命令。使用sed脚本时，不再用引号来确保sed命令不被shell解释。例如sed脚本script：

```
#handle datafile
3\l
~~~~~
3,$s\(^hrwang\) is \(mifan\)\/2 is \1/
$a\
We will love eachother forever! !

#sed -f script datafile
My name is hrwang
Your name is mifan
~~~~~
mifan is hrwang's husband.          # 啦啦～～～
mifan is hrwang's wife.
We will love eachother forever! !
```