



链滴

shell sed 命令详解：选取、替换、删除、新增数据

作者：[Leif160519](#)

原文链接：<https://ld246.com/article/1587370777231>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



sed 是一种几乎可以应用在所有 UNIX 平台（包括 Linux）上的轻量级流编辑器。sed 有许多很好的特性。首先，它相当小巧，通常要比你所喜爱的脚本语言小多倍。其次，因为 sed 是一种流编辑器，所以，它可以对从如管道这样的标准输入中接收的数据进行编辑。因此，无须将要编辑的数据存储在磁盘的文件中。因为可以轻易将数据管道输出到 sed，所以，将 sed 用作强大的 Shell 脚本中长而复杂的管道很容易。

sed 主要是用来将数据进行选取、替换、删除、新增的命令。我们看看命令的语法：

sed [选项] [动作] 文件名

选项：

- -n：一般 sed 命令会把所有数据都输出到屏幕上。如果加入此选项，则只会把经过 sed 命令处理行输出到屏幕上；
- -e：允许对输入数据应用多条 sed 命令编辑；
- -f 脚本文件名：从 sed 脚本中读入 sed 操作。和 awk 命令的 -f 选项非常类似；
- -r：在 sed 中支持扩展正则表达式；
- -i：用 sed 的修改结果直接修改读取数据的文件，而不是由屏幕输出动作；
- a \：追加，在当前行后添加一行或多行。当添加多行时，除最后一行外，每行末尾需要用 “\” 代表数据未完结；
- c \：行替换，用c后面的字符串替换原数据行。当替换多行时，除最后一行外，每行末尾需用 “\” 代表数据未完结；
- i \：插入，在当前行前插入一行或多行。当插入多行时，除最后一行外，每行末尾需要用 “\” 代表数据未完结；
- d：删除，删除指定的行；
- P：打印，输出指定的行；
- s：字符串替换，用一个字符串替换另一个字符串。格式为 “行范围s/旧字符串/新字符串/g”（和Vim

的替换格式类似)；

注意: sed 所做的修改并不会直接改变文件的内容 (如果是用管道符接收的命令的输出, 则连文件都有), 而是把修改结果只显示到屏幕上, 除非使用"-i"选项才会直接修改文件。

1.行数据操作

举几个例子来看看 sed 命令到底是干什么的。假设我想查看一下 student.txt 文件的第二行, 就可以用"p"动作了。

```
sed '2p' student.txt
ID Name gender Mark
1 Liming M 86
1 Liming M 86
2 Sc M 90
3 Gao M 83
```

好像看着不怎么顺眼啊! "p"动作确实输出了第二行数据, 但是 sed 命令还会把所有数据都输出一次这时就会看到这个比较奇怪的结果。那如果我想指定输出某行数据, 就需要"-n"选项的帮助了。

```
sed -n '2p' student.txt
1 Liming M 86
```

这样才可以输出指定的行。大家可以这样记忆: 当我们需要输出指定的行时, 需要把"-n"选项和"p"作一起使用。

1.1 删除文件中的数据

再来看看如何删除文件中的数据:

```
sed '2,4d' student.txt
#删除从第二行到第四行的数据
ID Name gender Mark
```

```
cat student.txt
#文件本身并没有被修改
ID Name gender Mark
1 Liming M 86
2 Sc M 90
3 Gao M 83
```

看到这条命令首先需要注意, 所有的动作必须使用"单引号"包含; 其次, 在动作中可以使用数字代表号, 逗号代表连续的行范围。还可以使用"**"代表最后一行, 如果动作是"2,d"**", 则代表从第二行删除最后一行。

1.2 如何追加和插入行数据

再来看看如何追加和插入行数据:

```
sed '2a hello' student.txt
#在第二行后加入hello
ID Name gender Mark
1 Liming M 86
```

```
hello
2 Sc M 90
3 Gao M 83
```

"a"动作会在指定行后追加数据。如果想要在指定行前插入数据，则需要使用"i"动作。

```
sed '2i hello \
world' student.txt
ID Name gender Mark
hello
world
1 Liming M 86
2 Sc M 90
3 Gao M 83
```

如果想追加或插入多行数据，则除最后一行外，每行的末尾都要加入"
表数据未完结。

1.3 "-n"选项的作用

再来看看"-n"选项的作用，命令如下：

```
sed -n '2i hello \
world' student.txt
hello
world
```

看到了吧，"-n"只用于查看 sed 命令操作的数据，而并非查看所有的数据。

1.4 如何实现行数据替换

再来看看如何实现行数据替换：

```
cat student.txt | sed '2c No such person'
ID Name gender Mark
No such person
2 Sc M 90
3 Gao M 83
```

第二行数据变成了"查无此人"。通过这个例子我们看到了，sed 也可以接收和处理管道符传输的数据。

sed 命令在默认情况是不会修改文件内容的。如果我确定需要让 sed 命令直接处理文件的内容，则可使用"-i"选项。不过要小心，这样非常容易误操作，在操作系统文件时请小心谨慎。可以使用这样的命令：

```
sed -i '2c No such person' student.txt
```

2.字符串替换

"c"动作是进行整行替换的，如果仅仅想替换行中的部分数据，就要使用"s"动作了。"s"动作的格式如：

```
sed's/旧字符串/新字符串/g' 文件名
```

替换的格式和 Vim 非常类似，例如：

```
sed '3s/83/99/g' student.txt
#在第三行中，把83换成99
ID Name gender Mark
1 Liming M 86
2 Sc M 90
3 Gao M 83
```

如果想把某行的成绩注释掉，让它不再生效，则可以这样做：

```
sed '4s/^/#/g' student.txt
#在这里使用正则表达式，"^"代表行首
ID Name gender Mark
1 Liming M 86
2 Sc M 90
#3 Gao M 83
```

不仅如此，我们还可以这样做：

```
sed -e 's/Liming//g; s/Gao//g' student.txt
#同时把"Liming"和"Gao"替换为空
ID Name gender Mark
1 M 86
2 Sc M 90
3 M 83
```

"-e"选项可以同时执行多个 sed 动作，当然，如果只执行一个动作，则也可以使用"-e"选项，但是这没有什么意义。还要注意，多个动作之间要用";"或回车分隔，例如，上一条命令也可以这样写：

```
sed -e 's/Liming//g
> s/Gao//g' student.txt
ID Name gender Mark
1 M 86
2 Sc M 90
3 M 83
```

注意：>是敲回车的时候，自动生成的，无需手动输入。

参考：

- [shell sed命令详解：选取、替换、删除、新增数据](#)
- [Linux sed 命令](#)