# Linux Sed 使用介绍

简介：

Sed 是一款流编辑工具，用来对文本进行过滤与替换操作。Sed通过一次进读取一行内容来对某些指令进行处理后输出，所以Sed更适合与处理大数据文件。

## Sed基本语格式法

Sed 从文件中读取数据，如果没有输入文件，则默认对标准输入进程数据进行处理，脚本指令是第一个非“-”开头的参数，具体语法格式如下：

用法：sed [选项] ... {脚本指令} [输入文件]

常用选项：

--version 显示sed版本

--help 显示帮助文档

-n,--quiet 静默输出，默认情况下，sed程序在所有脚本指令执行 完毕后，将自动打印模式控件中的内容。

-e script 允许多个脚本指令被执行

-f script file 从文件中读取脚本指令，对编写自动脚本程序很实用

-i 慎用，该选项将直接修改源文件

-lN 该选项指定l命令可以输出的行长度，l指定为输出非 打印字符

--posix 禁用GNU sed 扩展功能

-s 默认情况下，sed将把输入的多个文件名作为一个厂的 连续的输入流。而GNU sed则允许把他们当作单独的 文件

-u 最低限度的缓存输入与输出

1，基本格式范例

Sed通过特定的脚本指令对文件处理。

Sed脚本指令的基本格式是：[地址]命令（有些命令仅可以对一行操作，有些可以对多行操作），命令也可以用花括号进行组合，使用命令序列可以作用于同一个地址。

简单的几个指令：

a,append 表示追加指令

i,inster 表示插入指令

d,delete 表示删除指令

s,substitution 表示替换指令

格式：

address{

Command1

Command5

Command3

}

注意：第一个命令可以和左花括号在同一行，但右花括号必须单独处于一行。此外命令后添加空格会产生错误。

演示：

test.txt样本源文件内容（数字代表行号）：

1 root:x:0:0:root:/root:/bin/bash

2 bin:x:1:1:bin:/bin:/sbin/nologin

3

4 adm:x:3:4:adm:/var/adm:/sbin/nologin

5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

6

7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown

8 halt:x:7:0:halt:/sbin:/sbin/halt

[root@dong test]# sed ‘2a TYPE=abc’ test.txt

# 第二行后追加TYPE=abc

[root@dong test]# sed ‘3i TYPE=abc’ test.txt

# 第三行前追加 TYPE=abc

[root@dong test]# sed ‘s/root/admin’ test.txt

# 将样本文件中的所有root替换为admin

[root@dong test]# sed ‘4,5d’ test.txt

# 删除4,5行的内容

[root@dong test]# sed ‘/adm/a TYPE=abc’ test.txt

# 匹配到包含adm的行，并在其后面添加TYPE=abc

[root@dong test]# sed ‘/^halt/d’ test.txt

# 匹配以halt开始的行，并删除该行

另外，操作指令可以写到脚本文件中，并通过sed的-f选项读取，脚本文件中的注释行是以#开始的行，如果#后面的字符为n，则屏蔽Sed程序的自动输出功能，等同于命令选项-n。

创建脚本一个sed脚本，内容如下：

[root@dong test]# cat sed.sh # 脚本内容为，匹配到空白行则删除该行

# test

/^$/d

[root@dong test]# sed -f sed.sh test.txt

# 对test.txt文件执行sed.sh脚本指令

执行多个指令：

[root@dong test]# sed ‘s/yes/no/;s/static/dhcp/’ test.txt # 使用分号隔开指令

[root@dong test]# sed -e ‘s/yes/no/’ -e ‘s/static/dhcp/’ test.txt # 使用-e选项

[root@dong test]# sed ‘ # 利用分行

>s/yes/no/

>s/static/dhcp’ test.txt

2，操作地址匹配范例

在sed中操作文件，若不指定一个地址，则默认对文件的所有行操作。例如：

Sed ‘d’ test.txt 将删除test.txt的所有行。而’2d’仅删除第二行。

Sed提供了一些方式来确定操作地址的范围：

Number 指定输入文件的唯一行号

Sed -n ‘2d’ test.txt # 删除test.txt文件的第二行

First~step 指定已first开始，并指定操作步长为step。

Sed -n ‘1~2p’ test.txt # 打印文件的奇数行

$ 匹配文件的而最后一行

/regexp/ 匹配正则表达式的行

Addr1,addr2 匹配从操作地址1到操作地址2的所有行

Sed ‘2,8d’ test.txt # 删除2~8之间的所有行

Addr,+N 匹配地址1以及后面的N行内容

3，常用指令汇总

S 替换 格式:s/匹配需要替换的内容/替换的新内容/标记

标记有 n 和 g

n: 1~512之间的数字，指定模式中第n次出现 进行替换。

g: 对模式空间的所有匹配进行替换。若没有g, 则默认仅替换第一个

a 追加

c 更改

y 按字符转换

p 打印 打印模式空间的内容

w 保存至文件 将模式空间的内容写到文件file中。w file

d 删除

i 插入

l 打印（显示非打印字符）

L 打印（不显示非打印字符）

r 读入文件内容

q 退出

& 用正则表达式匹配的内容进行替换

\n 匹配第n个子串，该子串之前在pattern中用\(\)指定

\ 转义特殊字符

以下列出10个案例供参考：

**案例一：**

**需求：仅替换第二个“body”为“/body”**

**样本Test.txt内容如下：**

**<html>**

**<body>abc<body>**

**</html>**

**脚本sed.sh内容如下：**

**/body/{**

**s//\/body/2**

**}**

**或者**

**s/body/\/body/2**

**执行sed -f sed.sh test.txt**

**执行结果：**

**<html>**

**<body>abc</body>**

**</html>**

**案例二：**

**需求：给所有的第一个h1,h2等添加<>,第二个h1,h2添加</>**

**样本Test.txt内容如下：**

**<html>**

**h1helloh1**

**h2helloh2**

**h3helloh3**

**</html>**

**脚本sed.sh内容如下：**

**/h[0-9]/{**

**s//\<&\>/1**

**s//\<\/&\>/2**

**}**

**或者**

**s/h[0-9]/\<&\>/1**

**s/h[0-9]/\<\/&>/2**

**执行sed -f sed.sh test.txt**

**执行结果：**

**<html>**

**<h1>hello</h1>**

**<h2>hello</h2>**

**<h3>hello</h3>**

**</html>**

**案例三：**

**需求：删除文件中的所有空白行**

**样本Test.txt内容如下：**

**<html>**

**h1helloh1**

**h2helloh2**

**h3helloh3**

**</html>**

**脚本sed.sh内容如下：**

**/.\*/{**

**/^$/d**

**}**

**或者**

**/^$/d**

**执行sed -f sed.sh test.txt**

**执行结果：**

**<html>**

**h1helloh1**

**h2helloh2**

**h3helloh3**

**</html>**

**案例四：**

**案例四~案例十所使用的样本文件内容如下（test.txt）：**

**<html>**

**h1helloh1**

**h2helloh2**

**h3helloh3**

**</html>**

**需求：在h2之后添加一行，内容为：h33abch33**

**脚本sed.sh内容如下：**

**/h2/{**

**a h33abch33**

**}**

**或者**

**/h2 /a h33abch33**

**需求：在匹配/html行前插入内容h4helloh4**

**脚本send.sh内容如下：**

**/\/html/{**

**i h4helloh4**

**}**

**或者**

**/\/html/i h4helloh4**

**需求：将包含h33行的内容更改为h5helloh5**

**脚本send.sh内容如下：**

**/h33/{**

**c h5helloh5**

**}**

**或者**

**/h33/c h5helloh5**

**需求：列印（l）显示模式空间中的内容，显示非打印字符，一般与-n一起使用，否则会输出两次。**

**sed -n ‘3,4l’ test.txt**

**需求：将h5helloh5转换为大写**

**脚本send.sh内容如下：**

**/h5helloh5/{**

**y/abcdefghijklmnopqrstuvwxyz/ABCDEFGHIJKLMNOPQRSTUVWXYZ/**

**}**

**或者**

**/h5helloh5/y/helo/HELO/**

**需求：显示1,2行内容，但不显示非显示字符，一般结合-n使用。**

**Sed -n ‘1,2p’ test.txt**

**注意：以上所有的案例都是为修改源文件的，若要修改源文件请使用-i参数。**

**例如：sed -i /^$/d test.txt**

关于’s///’命令的另一个妙处是：’/’分隔符有许多替换方案，如果规则表达式或替换字符中有许多斜杠，则可以通过在’s’之后指定一个不同的字符来更改分隔符。

例如：

Sed -e ‘s:/usr/local:/usr:g’ mylist.txt

此时是替换分隔符，sed会将/usr/local替换为/usr