sed:

Stream EDitor

**流式**编辑器，**行编辑器**

模式空间: pattern space

模式空间里面有个过滤器

先过滤一下行，再处理。模式空间里面有个过滤器，模式先匹配一下，如果匹配再处理

元字符组合起来就成了模式。

依赖于模式空间，sed把传递的内容一次读取一行 sed指定模式来实现文本过滤，符合模式的文本使用对应的编辑命令处理，然后显示到屏幕上

grep "PATTERN"

**用法：**

sed [options] 'script' input\_file ... 这不能成为模式，里面可能还包含命令等多以叫做脚本

sed [options] -f script\_file input\_file ... 从指定文件中读取脚本

script\_file可以为

sed 'ADDR1,ADDR2commond' input\_file 指定起始结束地址

例如

删除文件第1，2行：

sed ‘1,2d’ file **删除并输出到屏幕但不修改源文件，需要输出重定向。如果想直接操作源文件 –i选项**

还可以!号取反 1,2!d 对命令取反

子命令：p显示, d删除, s, a, i, r, w

替换 追加 读取其他 写出

d: 删除

1,2d

1,2!d

p: 显示 模式空间中的文本在处理之前，每一次都会先显示；p,显示，即**会显示两次，**-n不显示原有内容 sed –n ‘1,3p’ binary.sh 这样就会只显示一次

sed: 基本正则表达式的元字符

比greb强大之处在于既可以过滤还可以处理

sed '/PATTERN/command' input\_file

sed '/PATTERN1/,/PATTERN2/command' input\_file ... 1第一次匹配到的行到2第一次匹配到的行

**sed –r** 选项，扩展的正则表达式

**指定字符**

.: 任意单个字符

[]: 指定范围内的任意单个字符；[a-z]

[^]: 指定范围外的任意单个字符; [^A-Z]

[:lower:] 所有的小写字符

[:upper:] 所有的大写字母

[:digit:] 数字

[:alpha:] 所有的字母

[:alnum:] 数字和字母

[:space:] 空白空间

[:punct:] 所有的标点符号

取反的话直接中括号外边加个取反和中括号

**指定字符出现次数**

\*: 匹配前面的字符任意次

.\*: 匹配任意长度的任意字符

\?: 匹配前面的字符0次或1次； ab\?c: abc, ac, abbc

\{m,n\}

\{m,\} 至少m次

\{0,n\} 至多n次

\{n\} 精确n次

**锚定符：**

\<, \b: 词首，\<root

\>, \b: 词尾, root\>

^: 行首

$: 行尾

分组引用：

\(\), \1, \2

\(abc\(cdef\)\), \1,abcdef \2cdef

查找替换元字符

s/PATTERN/string/

&：

s/l..e/&er/

**例子**：删除#开头的行

#sed ‘/^#/d’ binary.sh

删除出现了大写字母的行：

sed ‘/A-Z]/d’ binary.sh

**扩展的正则表达：grep -E**

?

{m,n} 指定次数

{0,n}

{m,}

() 不需要反斜线了

+: 匹配前面的字符至少1次，相当于{1,}

|: 或者， cat|dog: cat 或 dog 整个字符

下面开始扩展讲一些其他命令：

a \: 在模式匹配到的行后面添加新内容；

i \: 在模式匹配到的行前面添加新内容；

\n: 实现添加多行；换行符

r FILE:

w FILE: 匹配行输出

s 命令：s: s/PATTERN/string/

g: 全局 多次出现多次替换

i: 模式匹配忽略大小写

s@PATTERN@string@

哪些行换为什么字符

**example：**

找包含#的行：

sed –n ‘/#/p’ binary.sh

在包含#的行最后添加一段文字：

sed ‘/#/a \this is a test’ binary.sh

在指定行加入内容:r

sed ‘2r /root/test’ binary.sh

sed ‘/^# / root/test’ binary.sh

大写行输出：w



/dev/vol0/root 改为 /dev/sda2：

转义：

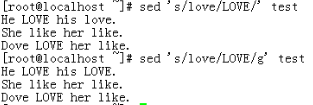
s/\/dev\/vol0\/root/\/dev\/sda2/

或者： s#/dev/vol0/root#/dev/sda2#

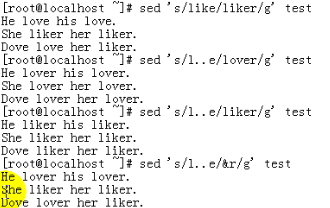
把like love换为liker lover



小写改大写：



前面匹配到什么就改什么一条命令love like都改了



.,$s///g

y/字符集1/字符集2/

选项：

-i: 直接操作原文件

-r: 使用扩展正则表达式

**-e: 多个脚本 联合处理**

n: 下一行，覆盖

N: 下一行，追加

h: 将模式中的内容复制到保留空间，覆盖的方式

H: 将模式中的内容复制到保留空间，追加的方式

g: 将保留空间内容复制到模式空间，覆盖的方式

G: 将保留空间内容复制到模式空间，追加的方式

b: 跳转, 略过符合模式的行

:label

b label

t label:

测试：前面的s命令执行成功就跳转到标记位置；

x: 保留空间和模式空间对调

选择分支：

sed标签最长为7个字符；声明标签时使用 :label的格式，跳转时不用冒号；

/etc/sysconfig/network

/var/log/

取出这个路径的父目录

${para%/\*}

sed [options] 'script' input\_file ...

addr1,addr2: 1,7

addr1,+n: 5,+3

/PATTERN/

/PATTERN1/,/PATTERN2/

m~n: 1~2, 0~8

p

d

a \test

i \test

r FILE

w FILE

s///

g

i

n:

N:

h:

H:

g:

G:

x:

=: 打印行号

b label

t label

选项：

-i

-n

-e

-r

sed:

sed [options] 'script' input\_file ...

-i

-n

-r

-e

定址方式：

addr1,addr2

addr1,+N

/PATTERN/

/PATTERN1/,/PATTERN2/

start~step

编辑命令：

d

p

a \test

i \test

s///

r FILE

w FILE

=

g:

G:

h:

H:

n:

N:

x:

b label

t label