

本笔记为北京千锋 V2017.1 《03 Linux Shell 脚本自动化编程实战》，笔记中所涉及到的项目均基于 Centos7u3 x86_64 或 Centos6u8 环境。

千锋所有学员均可自由使用和转载该笔记，为了尊重作者的辛劳，敬请注明出处！

=====

```

graph LR
    A[文本文件  
this is a line] --> B(sed 的模式空间(缓冲区)  
this is a line)
    B --> C(sed 的输出  
this is a line)
  
```

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

Sed 主要用来自动编辑一个或多个文件；简化对文件的反复操作；编写转换程序等。

二、命令格式

注：

只有当命令存在语法错误时，sed 的退出状态才是非 0

与 `grep` 一样，`sed` 在文件中查找模式时也可以使用正则表达式(RE)和各种元字符。正则表达式是

使用基本元字符集 `^, $, ., *, [], [^], \< \>, \(), \{\}`

使用扩展元字符的方式:

```
\+
sed -r
```

四、sed 基本用法

```
# sed -r '' /etc/passwd
# sed -r 'p' /etc/passwd
# sed -r -n 'p' /etc/passwd
# sed -r -n '/root/p' /etc/passwd
# sed -r 's/root/alice/' /etc/passwd
# sed -r 's/root/alice/g' /etc/passwd
# sed -r 's/root/alice/gi' /etc/passwd

# sed -r '/root/d' /etc/passwd
# sed -r '\crootcd' /etc/passwd
=====

[root@tianyun ~] vim a.txt
/etc/abc/456
etc

[root@tianyun ~]# sed -r '//etc/abc/456/d' a.txt
sed: -e 表达式 #1, 字符 0: no previous regular expression

[root@tianyun ~]# sed -r '/^etc\abc\456/d' a.txt
/etc/abc/456

[root@tianyun ~]# sed -r '\#/etc/abc/456#d' a.txt
/etc/abc/456

[root@tianyun ~]# sed -r 's#/etc/abc/456#/dev/sda1#'
=====
```

五、sed 扩展

==地址（定址）

地址用于决定对哪些行进行编辑。地址形式可以是数字、正则表达式或二者的结合。如果没有指定

地址，sed 将处理输入文件中的所有行。

```
# sed -r 'd' /etc/passwd
# sed -r '3d' /etc/passwd
# sed -r '1,3d' /etc/passwd
# sed -r '/root/d' /etc/passwd
# sed -r '/root/,5d' /etc/passwd
# sed -r 's/root/alice/g' /etc/passwd

# sed -r '/^adm/,+20d' /etc/passwd
# sed -r '2,5d' /etc/passwd
```

```
# sed -r '/root/d' /etc/passwd
# sed -r '/root/!d' /etc/passwd
```

```
# sed -r '1~2d' /etc/passwd //删除所有奇数行 odd-numbered
# sed -r '0~2d' /etc/passwd //删除所有偶数行 even-numbered
```

==sed 命令

sed 命令告诉 sed 对指定行进行何种操作，包括打印、删除、修改等。

命令 功能

- a** 在当前行后添加一行或多行
- c** 用新文本修改（替换）当前行中的文本
- d** 删除行
- i** 在当前行之前插入文本
- 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
- !** 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

s 替换标志

- g** 在行内进行全局替换
- i** 忽略大小写
- r** 从文件中读
- w** 将行写入文件
- y** 将字符转换为另一字符（不支持正则表达式）
- h** 把模式空间里的内容复制到暂存缓冲区(覆盖)
- H** 把模式空间里的内容追加到暂存缓冲区
- g** 取出暂存缓冲区的内容，将其复制到模式空间，覆盖该处原有内容
- G** 取出暂存缓冲区的内容，将其复制到模式空间，追加在原有内容后面
- x** 交换暂存缓冲区与模式空间的内容

==选项

选项 功能

- e** 允许多项编辑
- f** 指定 sed 脚本文件名
- n** 取消默认的输出
- i** inplace，就地编辑

-r 支持扩展元字符

六、sed 命令示例

打印命令: **p**

```
# sed -r '/north/p' datafile  
# sed -r -n '/north/p' datafile
```

删除命令: **d**

```
# sed -r '3d' datafile  
# sed -r '3{d;}' datafile  
# sed -r '3{d}' datafile  
  
# sed -r '3,$d' datafile  
# sed -r '$d' datafile  
# sed -r '/north/d' datafile  
# sed -r '/sout/d' datafile
```

替换命令: **s**

```
# sed -r 's/west/north/g' datafile  
# sed -r 's/^west/north/' datafile  
# sed -r 's/[0-9][0-9]$/&.5/' datafile //&代表在查找串中匹配到的内容  
# sed -r 's/Hemenway/Jones/g' datafile  
# sed -r 's/(Mar)got/\1ianne/g' datafile  
# sed -r 's#3#88#g' datafile
```

读文件命令: **r**

```
# sed -r '/Suan/r /etc/newfile' datafile  
# sed -r '2r /etc/hosts' a.txt  
# sed -r '/2/r /etc/hosts' a.txt
```

写文件命令: **w**

```
# sed -r '/north/w newfile' datafile  
# sed -r '3,$w /new1.txt' datafile
```

追加命令: **a**

```
# sed -r '2a\111111111111' /etc/hosts  
# sed -r '2a\111111111111\  
> 222222222222\  
> 333333333333' /etc/hosts
```

插入命令: **i**

```
# sed -r '2i\111111111111' /etc/hosts  
# sed -r '2i1111111111\  
> 222222222222'
```

```
> 2222222222\
> 3333333333' /etc/hosts
```

修改命令: c

```
# sed -r '2c\111111111111' /etc/hosts
# sed -r '2c\11111111111\
> 2222222222\
> 3333333333' /etc/hosts
```

获取下一行命令: n

```
# sed -r '/eastern/{ n; d }' datafile
# sed -r '/eastern/{ n; s/AM/Archile/ }' datafile
```

暂存和取用命令: h H g G

```
# sed -r '1h;$G' /etc/hosts
# sed -r '1{h;d);$G' /etc/hosts
# sed -r '1h; 2,$g' /etc/hosts
# sed -r '1h; 2,3H;$G' /etc/hosts
```

暂存空间和模式空间互换命令: x

```
# sed -r '4h; 5x; 6G' /etc/hosts
```

反向选择: !

```
# sed -r '3d' /etc/hosts
# sed -r '3!d' /etc/hosts
```

多重编辑选项: e

```
# sed -r -e '1,3d' -e 's/Hemenway/Jones/' datafile
# sed -r '1,3d; s/Hemenway/Jones/' datafile

# sed -r '2s/WE/UPLOOKING/g; 2s/Gray/YYY/g' datafile
# sed -r '2{s/WE/UPLOOKING/g; s/Gray/YYY/g}' datafile
```

七、sed 常见操作

删除配置文件中#号注释行

```
sed -ri '/^#/d' file.conf
```

```
sed -ri '/^[ \t]*#/d' file.conf
```

删除配置文件中//号注释行

```
sed -ri '\Y^[ \t]*//Yd' file.conf
```

删除无内容空行

```
sed -ri '/^[ \t]*$/d' file.conf
```

删除注释行及空行：

```
# sed -ri '/^[ \t]*#/d; /^[ \t]*$/d' /etc/vsftpd/vsftpd.conf
```

```
# sed -ri '/^[ \t]*#|^[ \t]*$/d' /etc/vsftpd/vsftpd.conf
```

```
# sed -ri '/^[ \t]*($|#)/d' /etc/vsftpd/vsftpd.conf
```

修改文件：

```
# sed -ri '$a\chroot_local_user=YES' /etc/vsftpd/vsftpd.conf
```

```
# sed -ri '/^SELINUX=cSELINUX=disabled' /etc/selinux/config
```

```
# sed -ri '/UseDNS/cUseDNS no' /etc/ssh/sshd_config
```

```
# sed -ri '/GSSAPIAuthentication/cGSSAPIAuthentication no' /etc/ssh/sshd_config
```

给文件行添加注释：

```
# sed -r '2,6s/^/#/' a.txt
```

```
# sed -r '2,6s/(.*)/#\1/' a.txt
```

```
# sed -r '2,6s/.*/#&/' a.txt &匹配前面查找的内容
```

```
# sed -r '3,$ s/^##*/#/' a.txt 将行首零个或多个#换成一个#
```

```
# sed -r '30,50s/^[ \t]*##*/#' /etc/nginx.conf
```

```
# sed -r '2,8s/^[ \t#]*#/' /etc/nginx.conf
```

sed 中使用外部变量

```
# var1=11111
```

```
# sed -ri "3a$var1" /etc/hosts
```

```
# sed -ri "$a$var1" /etc/hosts
```

```
# sed -ri '$a\'"$var1" /etc/hosts
```

```
# sed -ri 3a$var1 /etc/hosts
```

```
# sed -ri "\$a$var1" /etc/hosts
```

作业：

```
[root@tianyun ~]# vim 12345.txt
```

1
2
3
4
5

```
[root@tianyun ~]# sed -r '1!G; $!h; $!d' 12345.txt
```

5
4
3
2
1