**正则表达式与Sed文本编辑器**

# 正则表达式

## 正则表达式概述

使用一组字符串来描述、匹配一系列符合某个句法规则的字符串，由普通字符与特殊字符组成，正则表达式广泛使用在脚本编程、文件编辑器中，例如php、java、python、perl、shell等，正则表达式（regular expression）简写为regex、regexp、RE。用来检索、替换符合模式的文本。具有强大的文本匹配功能，能够在文本中快速高效地处理文本。

在大多数语言中正则表达式都被包括在两个正斜杠(/)之间。例如：/pattern/

正则表达式就是一套为处理大量的字符串而定义的规则和方法，通过正则表达这些特殊符号匹配我们需要的文本，进而我们可以对文本进行增删查改等操作。

为什么要学习正则表达式？

在Linux运维工作中，需要时刻面对大量的日志、程序、命令的输出。迫切的需要过滤我们需要的一部分内容，甚至是一个字符串。例如：只想获取网卡的IP地址信息，从日志中提取我们想需要的信息等。

## 正则表达式分类

* 基础正则表达式
* 扩展正则表达式

## Linux中常用的文本处理工具

* grep 匹配，查询
* sed 编辑（增加，删除，修改）
* awk 文本格式化（字符串提取）

### 注意事项：

* Linux中正则一般以行为单位处理文件。
* alias grep='grep --color=auto'
* 注意字符集，LANG=C

### 通配符：处理文件名较多

正则表达式和我们命令行中使用的通配符是有本质区别的

ls -l \*.txt

\* 任意长度任意字符串，属于通配符

？单个任意字符串，属于通配符

### grep命令的参数：

-v 排除匹配的内容

-E 支持扩展的正则表达式

-i 忽略大小写

-o 只输出匹配的内容

--color=auto 匹配的内容显示颜色

-n 在行首显示行号

-q 匹配内容不显示（静默输出）

-w 将过滤条件当成单词来匹配

-c 只输出匹配行的计数

## 基础正则表达式

\ 转义字符。例如：“\!”将逻辑否的!看做普通字符

^ 匹配字符串开始的位置，以…为开头的

$ 匹配字符串结束的位置，以…结束的

^$ 表示空行，不是空格

. 匹配任意的一个字符

\* 匹配前面子表达式（字符）0次或者多次

.\* 匹配任意字符串

[list] 匹配list列表中的一个字符，例如：[abc]、[a-z]、[a-zA-Z0-9]

[^list] 匹配任意非list表中的一个字符，例如：[^a-z]、[^0-9]、[^A-Z0-9]

\{n\} 匹配前面子表达式n次

\{n,\} 匹配前面子表达式至少n次

\{n,m\} 匹配前面子表达式最少n次，最大m次

\< 固定词首

\> 固定词尾

常见的转义字符

|  |  |
| --- | --- |
| 转义字符 | 意义 |
| \n | 换行将当前位置移到下一行开头 |
| \r | 回车将当前位置移动到本行开头，替换掉本行开头相应位置内容 |
| \t | 水平制表调到下一个TAB位置 |
| \v | 垂直制表，将后面内容（包括空格）移动到下一行下方显示 |
| \\ | 代表一个反斜杠字符“\” |

练习示例：

**[root@localhost ~]# nl test.txt //nl=cat -n**

**1 gd**

**2 god**

**3 good**

**4 goood**

**5 gooood**

**6 gold**

**7 glad**

**8 gaad**

**9 abcDfg**

**10 food**

**11 601151272**

**12 HELLO**

**13 010-6666888**

**14 0666-5666888**

**15 IP 192.168.1.108**

**16 IP 173.16.16.1**

**17 pay $180**

**从windows导入进linux系统时，可能会将window的换行符识别为特殊字符，导致文件末尾有不识别的字符，可以用col命令删除特殊字符**

**[root@localhost ~]# grep 8 test.txt**

**010-6666888**

**0666-5666888**

**IP 192.168.1.108**

**pay $180**

**[root@localhost ~]# grep '8$' test.txt**

**[root@localhost ~]# cat test.txt|col -b >test1.txt**

**[root@localhost ~]# mv test1.txt test.txt**

**mv：是否覆盖"test.txt"？ y**

**[root@localhost ~]# grep '8$' test.txt**

**010-6666888**

**0666-5666888**

**IP 192.168.1.108**

**vim 多行处理：**

**删除文件行号：如果文件中有行号或者注册符号**

**Ctrl+v 把要删除的字符串对应列按方向键选中，按x删除**

**多行注释：需要在多行相同位置添加内容（比如行首加注释）**

**Ctrl+v 选中需要插入内容的多列的首字母，shift+i 插入内容 Esc两次**

**删除多行：**

**按v 选中多行按x删除**

**打印出包含“$”字符的行：含有转义符\的正则表达式一般用单引号引起来，不含转义符的可以用双引号或单引号，$字符是正则中有特殊含义的：表示行尾，加转义符后就只表示$这个字符本身**

**[root@localhost ~]# sed -n '/\$/p' test.txt**

**pay $180**

**[root@localhost ~]# awk '/\$/{print}' test.txt**

**pay $180**

**[root@localhost ~]# awk '/\$/{print $0}' test.txt**

**pay $180**

**[root@localhost ~]# grep '\$' test.txt**

**pay $180**

**过滤出以小写字母开头的行：^代表以…开头，[]代表匹配括号中任意一个**

**[root@localhost ~]# grep "^[a-z]" test.txt**

**[root@localhost ~]# sed -n '/^[a-z]/p' test.txt**

**gd**

**god**

**good**

**goood**

**gooood**

**gold**

**glad**

**gaad**

**abcDfg**

**food**

**pay $180**

**过滤出以数字结尾的行：**

**[root@localhost ~]# grep "[0-9]$" test.txt**

**[root@localhost ~]# sed -n '/[0-9]$/p' test.txt**

**[root@localhost ~]# awk '/[0-9]$/{print}' test.txt**

**010-66668888**

**0666-5666888**

**IP 192.168.1.108**

**IP 173.16.16.1**

**pay $180**

**过滤出go与d之间有任意一个字符的行： .代表任意一个字符**

**[root@localhost ~]# grep "go.d" test.txt**

**[root@localhost ~]# sed -n '/go.d/p' test.txt**

**[root@localhost ~]# awk '/go.d/{print}' test.txt**

**good**

**gold**

**过滤出g于d之间任意两个字符的行：**

**[root@localhost ~]# grep "g..d" test.txt**

**[root@localhost ~]# sed -n '/g..d/p' test.txt**

**[root@localhost ~]# awk '/g..d/{print}' test.txt**

**good**

**gold**

**glad**

**gaad**

**过滤出g和d之间没有o或者有多个o的行：\*代表前面字符或表达式重复任意次（包括0次）**

**[root@localhost ~]# grep "go\*d" test.txt**

**[root@localhost ~]# sed -n '/go\*d/p' test.txt**

**[root@localhost ~]# awk '/go\*d/{print}' test.txt**

**gd**

**god**

**good**

**goood**

**gooood**

**过滤出go和d之间有任意字符串的行：.\*组合代表任意多个任意组合的字符**

**[root@localhost ~]# grep "go.\*d" test.txt**

**[root@localhost ~]# sed -n '/go.\*d/p' test.txt**

**[root@localhost ~]# awk '/go.\*d/{print}' test.txt**

**过滤出g和ad之间有字母l或a的行：**

**[root@localhost ~]# grep "g[la]ad" test.txt**

**[root@localhost ~]# sed -n '/g[la]ad/p' test.txt**

**[root@localhost ~]# awk '/g[la]ad/{print}' test.txt**

**glad**

**gaad**

**过滤出行内有非小写字母的行：**

**[^abc]:[]内的^,代表取反，也就是查找不在括号内的字符的意思**

**[root@localhost ~]# grep [^a-z] test.txt**

**[root@localhost ~]# sed -n '/[^a-z]/p' test.txt**

**abcdEfg**

**60115127Z**

**HELLO**

**010-66668888**

**0666-5666888**

**IP 192.168.1.108**

**IP 173.16.16.1**

**pay $180**

**过滤出电话号码：\{3,4\}代表前面字符或表达式重复的次数范围，不加转义符\,会认为查找大括号{}，这个是将普通字符转义为特殊字符**

**[root@localhost ~]# grep "[0-9]\{3,4\}-[0-9]\{7,8\}" test.txt**

**[root@localhost ~]# sed -n '/[0-9]\{3,4\}-[0-9]\{7,8\}/p' test.txt**

**010-66668888**

**0666-5666888**

**过滤出IP地址：**

**[root@localhost ~]# grep "[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}" test.txt**

**[root@localhost ~]# sed -n '/[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}/p' test.txt**

**IP 192.168.1.108**

**IP 173.16.16.1**

## 扩展正则表达式

扩展正则表达式是基础正则表达式的扩充深化

+ 匹配前面子表达式1次及以上

? 匹配前面子表达式0次或者1次

() 将括号中的字符串作为一个整体

| 以或的方式匹配字条串

grep和sed不能直接支持扩展的正则表达式，grep可以加-E选项，sed可以加-r选项支持

练习示例：

**过滤出字母g与d之间至少有1个字母o的行：+代表前面字符或表达式出现至少一次**

**[root@localhost ~]# egrep 'go+d' test.txt**

**[root@localhost ~]# grep -E 'go+d' test.txt**

**[root@localhost ~]# sed -rn '/go+d/p' test.txt**

**[root@localhost ~]# awk '/go+d/{print}' test.txt**

**god**

**good**

**goood**

**gooood**

**过滤出字母g与d之间没有字母o或只有1个o的行：？代表前面字符或表达式出现至多一次**

**[root@localhost ~]# egrep 'go?d' test.txt**

**[root@localhost ~]# awk '/go?d/{print}' test.txt**

**gd**

**god**

**过滤出字母g与d之间两个字母o一起出现至少1次的行：多个字符组成的表达式用()括起来，代表是一个整体的匹配项（分组），比如oo为一个整体匹配项时，必须是oo,oooo这样的才匹配，ooo都不匹配**

**[root@localhost ~]# egrep "g(oo)+d" test.txt**

**[root@localhost ~]#** **awk '/g(oo)+d/{print}' test.txt**

**good**

**gooood**

**过滤出字母g与d之间是la或者是aa的行：多个表达式（分组）用竖杠|分隔，是或的关系，及匹配其中一项即可**

**[root@localhost ~]# egrep "g(la|aa)d" test.txt**

**[root@localhost ~]# awk '/g(la|aa)d/{print}' test.txt**

**glad**

**gaad**

过滤ifconfig中的IP：

**[root@localhost ~]# ifconfig ens33 | awk '/netmask/{print $2}'**

**192.168.200.111**

如何根据inet字符精确查找本机的ip地址？

# 文本流编辑器-sed

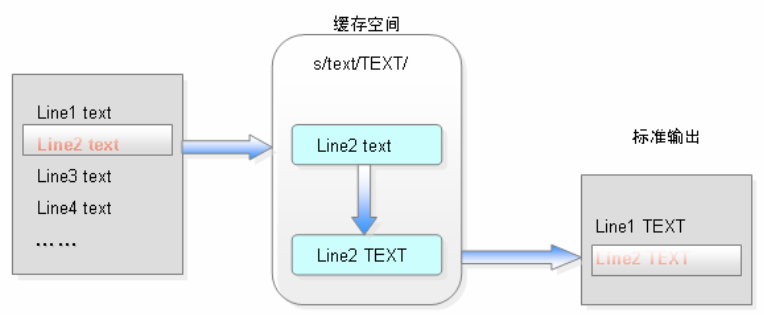
sed是一个非交互式的流文本编辑器（stream editor），常用于读取文本内容或根据指定的条件进行处理，可实现增删改查的功能。被广泛应用于shell脚本，以完成自动化处理任务。sed依赖于正则表达式。

sed在使用时通常会指定行号或正则表达式，否则处理的是全文。

工作原理：

sed每次只从文件或标准输入中读取一行数据，将之拷贝到一个编辑缓冲区，然后对该行数据按指定命令进行处理，并将结果输出到屏幕（标准输出），接着读入下一行。整个文件处理过程像流水线一样逐行处理。

Sed默认并不直接修改文件，他只是操作编辑缓冲区中的信息，编辑完的信息默认在屏幕上显示，如果有需要可将采用重定向的方式导出到指定的文件中（或者结合“-i”选项直接在文件中修改。）



## 格式：

sed [选项] ‘地址 编辑命令’ 输入文件

shell命令 | sed [选项] ‘地址 编辑命令’

sed [选项] –f sed脚本文件 输入文件

## 常用选项：

-e 指定要执行的命令，只有一个编辑命令时可省略

-n 只输出处理后的行，读入时不显示，不对原文件进行修改

-i 直接编辑原文件，不输出结果

-f 从脚本文件中读取编辑命令并执行

-r 支持扩展的正则表达式

## 编辑命令格式：

‘地址 编辑命令’

地址：可为行数、正则表达式，没有地址代表全文

操作：

p 打印（输出）

d 删除（整行）

s 替换（字符串）

c 替换（整行）

r 读取指定文件（到行后）

= 显示文件行号

a append，追加指定内容到行后

i insert，追加指定内容到行前

w 另存为

n 表示读入下一行内容（跳过当前行不处理）

H 复制到剪贴板

g 将剪贴板中的内容覆盖到指定行

G 将剪贴板中的内容追加到指定行后

## 练习案例：

**[root@localhost ~]# nl test.txt**

**1 gd**

**2 god**

**3 good**

**4 goood**

**5 gooood**

**6 gold**

**7 glad**

**8 gaad**

**9 abcDfg**

**10 food**

**11 601151272**

**12 HELLO**

**13 010-6666888**

**14 0666-5666888**

**15 IP 192.168.1.108**

**16 IP 173.16.16.1**

**17 pay $180**

* p 打印

**输出第12行内容**

**[root@localhost ~]# sed -n '12p' test.txt**

**HELLO**

**输出3~5行的内容**

**[root@localhost ~]# sed -n '3,5p' test.txt**

**good**

**goood**

**gooood**

**输出所有奇数行: p;n逐行处理，分号代表两个命令，第一个p命令打印当前行，处理完后第二个命令n代表继续读一行并且跳过这行不做操作，也就是一次操作执行了两行，然后回来继续执行下一个处理（及从第三行开始处理）**

**[root@localhost ~]# nl test.txt| sed -n 'p;n'**

**1 gd**

**3 good**

**5 gooood**

**7 glad**

**9 abcDfg**

**11 601151272**

**13 010-6666888**

**15 IP 192.168.1.108**

**17 pay $180**

**命令中的分号代表执行第一条命令后再执行第二条，等于-e指定两条命令**

**[root@localhost ~]# nl test.txt| sed -n -e 'p' -e 'n'**

**1 gd**

**3 good**

**5 gooood**

**7 glad**

**9 abcDfg**

**11 601151272**

**13 010-6666888**

**15 IP 192.168.1.108**

**17 pay $180**

**输出所有偶数行：先读第一行跳过，再打印输出第二行，继续循环读第三行…**

**[root@localhost ~]# nl test.txt| sed -n 'n;p'**

**2 god**

**4 goood**

**6 gold**

**8 gaad**

**10 food**

**12 HELLO**

**14 0666-5666888**

**16 IP 173.16.16.1**

**输出1~5行之间的奇数行（第1,3,5行）:注意p和n必须在{}内代表在1-5行内操作这两个命令，如果没有大括号，p命令只在1到5行操作，n命令5行之后任然要操作，但是因为n是跳过，所以显示的信息还是只有1,3,5行**

**[root@localhost ~]# nl test.txt| sed -n '1,5{p;n}'**

**1 gd**

**3 good**

**5 gooood**

**[root@localhost ~]# nl test.txt| sed -n '1,5p;n'**

**1 gd**

**3 good**

**5 gooood**

**输出1~5行之间的偶数行：当输出偶数行时，没有{}就会有很大区别了，不在1到5行时不跳过，而是执行后面的p输出命令**

**[root@localhost ~]# nl test.txt| sed -n '1,5{n;p}'**

**2 god**

**4 goood**

**6 gold**

**[root@localhost ~]# nl test.txt| sed -n '1,5n;p'**

**2 god**

**4 goood**

**6 gold**

**7 glad**

**8 gaad**

**9 abcDfg**

**10 food**

**11 601151272**

**12 HELLO**

**13 010-6666888**

**14 0666-5666888**

**15 IP 192.168.1.108**

**16 IP 173.16.16.1**

**17 pay $180**

**输出包含字母H的行：查找内容（指定符合的条件）要用//指定**

**[root@localhost ~]# sed -n '/H/p' test.txt**

**HELLO**

**输出所有包含大写字母的行**

**[root@localhost ~]# sed -n '/[A-Z]/p' test.txt**

**abcDfg**

**HELLO**

**IP 192.168.1.108**

**IP 173.16.16.1**

**输出最后一行：单独的$不是查找的条件，也不是行尾的意思，而是最后一行，注意不能加//,否则就是查找以…结尾的意思了，也不能加转义符\,加了就是查找包含$的行**

**[root@localhost ~]# sed -n '$p' test.txt**

**pay $180**

**[root@localhost ~]# nl test.txt|sed -n '15,$p'**

**15 IP 192.168.1.108**

**16 IP 173.16.16.1**

**17 pay $180**

**[root@localhost ~]# sed -n '/8$/p' test.txt**

**010-6666888**

**0666-5666888**

**IP 192.168.1.108**

**[root@localhost ~]# sed -n '/\$/p' test.txt**

**pay $180**

* d 删除（整行）：d命令不能加-n选项，否则什么都看不到，因为-n是指显示符合条件的，而符合条件的直接从缓存区删除了，也就无法显示了，也就是d选项其实显示的是不符合条件的也就是没有被删除的

**删除第16行**

**[root@localhost ~]# nl test.txt|sed '16d'**

**1 gd**

**2 god**

**3 good**

**4 goood**

**5 gooood**

**6 gold**

**7 glad**

**8 gaad**

**9 abcDfg**

**10 food**

**11 601151272**

**12 HELLO**

**13 010-6666888**

**14 0666-5666888**

**15 IP 192.168.1.108**

**17 pay $180**

**删除原文件中的空行：-i选项直接对原文件操作，不在缓存区操作，可以在文件中添加一个空行，然后用-i命令项直接删除**

**[root@localhost ~]# sed -i '/^$/d' test.txt**

**[root@localhost ~]#cat test.txt**

**gd**

**god**

**good**

**goood**

**gooood**

**gold**

**glad**

**gaad**

**abcdEfg**

**food**

**60115127Z**

**HELLO**

**010-66668888**

**0666-5666888**

**IP 192.168.1.108**

**IP 173.16.16.1**

**pay $180**

**删除第1行和第3行**

**[root@localhost ~]# sed -e '1d' -e '3d' test.txt**

**god**

**goood**

**gooood**

**gold**

**glad**

**同样，删除第1行和第3行**

**[root@localhost ~]# sed '1d;3d' test.txt**

**god**

**goood**

**gooood**

**gold**

**glad**

* s 替换（字符串）：替换要在s后用三个/分隔需要替换的新旧内容，分隔符也可以用其他符号代替。如果有指定条件就在s前再加两个/指定查找的条件内容，指定查找的行直接写数字不要加/

**将所有小写o替换为大写O（g表示若同一行有多个小写o，全部替换，若不加g，则只替换每行的第一个小写o）**

**[root@localhost ~]# sed 's/o/O/g' test.txt**

**gd**

**gOd**

**gOOd**

**gOOOd**

**gOOOOd**

**gOld**

**glad**

**gaad**

**abcdEfg**

**fOOd**

**60115127Z**

**HELLO**

**010-66668888**

**0666-5666888**

**IP 192.168.1.108**

**IP 173.16.16.1**

**pay $180**

**以IP开头的行，行首加上#**

**[root@localhost~]# sed '/^IP/ s/^/#/' test.txt**

**gd**

**god**

**good**

**goood**

**gooood**

**gold**

**glad**

**gaad**

**abcdEfg**

**food**

**60115127Z**

**HELLO**

**010-66668888**

**0666-5666888**

**#IP 192.168.1.108**

**#IP 173.16.16.1**

**pay $180**

**在每行行尾插入字符串EOF**

**[root@localhost ~]# sed 's/$/EOF/' test.txt**

**gd**EOF

godEOF

goodEOF

gooodEOF

goooodEOF

goldEOF

**gladEOF**

**gaadEOF**

**abcdEfgEOF**

**foodEOF**

**60115127ZEOF**

**HELLOEOF**

**010-66668888EOF**

**0666-5666888EOF**

**IP 192.168.1.108EOF**

**IP 173.16.16.1EOF**

**pay $180EOF**

**将/etc/passwd文件20~25行的/sbin/nologin替换为/bin/bash。如果用s///的形式，将会多次使用转义符\，我们可以使用其他符号，如@进行分隔**

**[root@localhost ~]# nl /etc/passwd|sed '20,25s/\/sbin\/nologin/\/bin\/bash/g'|grep bash**

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

**20 gluster:x:996:993:GlusterFS daemons:/var/run/gluster:/bin/bash**

**21 saslauth:x:995:76:Saslauthd user:/run/saslauthd:/bin/bash**

**22 abrt:x:173:173::/etc/abrt:/bin/bash**

**23 setroubleshoot:x:994:991::/var/lib/setroubleshoot:/bin/bash**

**24 rtkit:x:172:172:RealtimeKit:/proc:/bin/bash**

**25 pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/bin/bash**

**43 ycq:x:1000:1000:ycq:/home/ycq:/bin/bash**

**[root@localhost ~]# nl /etc/passwd|sed '20,25s@/sbin/nologin@/bin/bash@g'|grep bash**

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

**20 gluster:x:996:993:GlusterFS daemons:/var/run/gluster:/bin/bash**

**21 saslauth:x:995:76:Saslauthd user:/run/saslauthd:/bin/bash**

**22 abrt:x:173:173::/etc/abrt:/bin/bash**

**23 setroubleshoot:x:994:991::/var/lib/setroubleshoot:/bin/bash**

**24 rtkit:x:172:172:RealtimeKit:/proc:/bin/bash**

**25 pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/bin/bash**

**43 ycq:x:1000:1000:ycq:/home/ycq:/bin/bash**

* c 替换（整行）

**将第2行替换为AAAAAAA**

**[root@localhost ~]# sed '2cAAAAAAAAAAAAAAA' test.txt**

**gd**

**AAAAAAAAAAAAAAA**

**good**

**goood**

**gooood**

**gold**

**glad**

**gaad**

**abcDfg**

**food**

**601151272**

**HELLO**

**010-6666888**

**0666-5666888**

**IP 192.168.1.108**

**IP 173.16.16.1**

**pay $180**

**把第5行至最后一行的内容替换为两行，AAAAAA…和BBBBBBB…（\n为换行）**

**[root@localhost ~]# sed '5,$cAAAAAAAAAAA\nBBBBBBBBBBBB' test.txt**

**gd**

**god**

**good**

**goood**

**AAAAAAAAAAA**

**BBBBBBBBBBBB**

* r 读取指定文件（到行后）

**读取/etc/resolv.conf文件内容在第5行后**

**[root@localhost ~]# sed '5r /etc/resolv.conf' test.txt**

**gd**

**god**

**good**

**goood**

**gooood**

**# Generated by NetworkManager**

**nameserver 202.106.0.20**

**nameserver 8.8.8.8**

**gold**

**glad**

**gaad**

**abcDfg**

**food**

**601151272**

**HELLO**

**010-6666888**

**0666-5666888**

**IP 192.168.1.108**

**IP 173.16.16.1**

**pay $180**

* a 追加指定内容到行后

**在第2行后添加NNNN**

**[root@localhost ~]# sed '2aNNNNNNNNNNNNNNNNNN' test.txt**

**gd**

**god**

**NNNNNNNNNNNNNNNNNN**

**good**

**……**

**在所有带数字的行下追加====…**

**[root@localhost ~]# sed '/[0-9]/a==============' test.txt**

**gd**

**god**

**good**

**goood**

**gooood**

**gold**

**glad**

**gaad**

**abcDfg**

**food**

**601151272**

**==============**

**HELLO**

**010-6666888**

**==============**

**0666-5666888**

**==============**

**IP 192.168.1.108**

**==============**

**IP 173.16.16.1**

**==============**

**pay $180**

**==============**

* i 追加指定内容到行前

**在第2行前追加NNNNN…**

**[root@localhost ~]# sed '2iNNNNNNNNNNNNNNNNNN' test.txt**

**gd**

**NNNNNNNNNNNNNNNNNN**

**god**

**……**

* w 另存为

**将15~16行内容另存到out.txt文件中**

**[root@localhost ~]# sed -n '15,16p' test.txt**

**IP 192.168.1.108**

**IP 173.16.16.1**

**[root@localhost ~]# sed '15,16w out.txt' test.txt**

**[root@localhost ~]# cat out.txt**

**IP 192.168.1.108**

**IP 173.16.16.1**

* H 复制到剪贴板
* G将剪贴板中的内容追加到指定行后

**将包含字母IP的行剪切到最后一行下**

**[root@localhost ~]# sed '/IP/ {H;d};$G' test.txt**

**gd**

**god**

**good**

**goood**

**gooood**

**gold**

**glad**

**gaad**

**abcdEfg**

**food**

**60115127Z**

**HELLO**

**010-66668888**

**0666-5666888**

**pay $180**

**IP 192.168.1.108**

**IP 173.16.16.1**

**将1~5行内容复制到15行、16行下**

**[root@localhost ~]# sed '1,5H;15,16G' test.txt**

**gd**

**god**

**good**

**goood**

**gooood**

**gold**

**glad**

**gaad**

**abcdEfg**

**food**

**60115127Z**

**HELLO**

**010-66668888**

**0666-5666888**

**IP 192.168.1.108**

**gd**

**god**

**good**

**goood**

**gooood**

**IP 173.16.16.1**

**gd**

**god**

**good**

**goood**

**gooood**

**后向引用（分组调用）**

**[root@localhost ~]# grep '#ServerName' /etc/httpd/conf/httpd.conf**

**#ServerName www.example.com:80**

**[root@localhost ~]# sed -i 's/#\(ServerName** [**www.\)example\(.com:80\)/\1baidu\2/gp' /etc/httpd/conf/httpd.conf**](http://www.\)example\(.com:80\)/\1baidu\2/gp'%20/etc/httpd/conf/httpd.conf)

**ServerName www.baidu.com:80**

**sed & 引用方式：查找到指定字符串并在其后添加字符串**

**[root@localhost ~]# sed -n '1,12 s#root#&admin#gp' /etc/passwd**

**rootadmin:x:0:0:rootadmin:/rootadmin:/bin/bash**

**operator:x:11:0:operator:/rootadmin:/sbin/nologin**

**sed从脚本文件中读取命令执行,可以执行相对复杂的命令：-f 命令文件**

[root@localhost ~]# sed -n '3,5p' test.txt

good

goood

gooood

[root@localhost ~]# cat 1.txt

3,5H

2,9d

13,$d

10,11G

[root@localhost ~]# nl test.txt|sed -f 1.txt

1 gd

10 food

3 good

4 goood

5 gooood

11 601151272

3 good

4 goood

5 gooood

12 HELLO

[root@localhost ~]#