# sed

## 【功能说明】：

sed - stream editor for filtering and transforming text 流编辑器用于过滤和转换文本

一种流编辑器，它是文本处理中非常中的工具，能够完美的配合正则表达式使用。可删除(delete)、改变(change)、添加(append)、插入(insert)、合并、交换文件中的数据行，或读入其它文件的数据到文件中，也可替换(substuite)它们其中的字符串、或转换(tranfer)其中的字母等。它处理时，把当前处理的行存储在临时缓冲区中，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，把缓冲区的内容送往屏幕。接着处理下一行，这样不断重复，直到文件末尾。文件内容并没有 改变，除非你使用重定向存储输出。

## 【语法格式】：

**sed [OPTION]... {script-only-if-no-other-script} [input-file]...**

调用sed命令有两种形式：

sed [options] 'command' file(s)

sed [options] -f scriptfile file(s)

## 【选项参数】：

**选项：**

**-e script, --expression=script**

**add the script to the commands to be executed**

将脚本添加到命令行执行，用于多项编辑

**-f script-file, --file=script-file**

**add the contents of script-file to the commands to be executed**

添加脚本文件中内容到命令被执行，用于一次性执行多项任务

**-n, --quiet, --silent**

**suppress automatic printing of pattern space**

抑制自动打印pattern space，sed默认输出全部，-n用于取消默认输出

**-i[SUFFIX], --in-place[=SUFFIX]**

**edit files in place (makes backup if extension supplied). The default operation mode is to break**

**symbolic and hard links. This can be changed with --follow-symlinks and --copy.**

就地编辑文件(如果提供扩展可备份源文件)。默认的操作模式就是改变符号链接和硬链接文件。这可以改变--follow-symlinks和--copy。

**-r, --regexp-extended**

**use extended regular expressions in the script.sed**

使sed支持扩展正则表达式语法（预设是基础正则表达式语法）

**-s, --separate**

**consider files as separate rather than as a single continuous long stream.**

把文件视为单独的，而不是作为一个单一的连续流

**-u, --unbuffered**

**load minimal amounts of data from the input files and flush the output buffers more often**

从输入文件中加载最小量的数据，并经常刷新输出缓冲区

**函数参数（命令）：**

|  |  |
| --- | --- |
| {} | 集合有相同位址参数的指令 |
| ! | 不执行函数参数 |
| = | Print the current line number 打印输出内容的行号 |
| a\  text | 将户输入的数据添加件，具有嵌入式换行符前面加一个反斜线 |
| i\  text | 插入，i 的后面可以接字串， 使用户输入的数据插入到文件，这些数据会在新的一行出现(目前的上一行)；具有嵌入式换行符前面加一个反斜线。 |
| q [exit-code] | 退出编辑 |
| r filename | Append text read from filename. 读取其它文件内容到当下 |
| w filename | Write the current pattern space to filename.  将当前模式空间的数据写到其它文件 |
| W filename | Write the first line of the current pattern space to filename. This is a GNU extension.  将当前模式空间的第一行数据写到其它文件，这是一个GNU扩展 |
| c\  text | 取代，使用户输入的数据替换所选数据，具有嵌入式换行符前面加一个反斜线。 |
| d | Delete pattern space. Start next cycle. 删除当前模式空间内数据 |
| D | 删除 pattern space 内第一个换行符（字母 \ ）前的数据，即删除第一行 |
| g | 与s 联合使用时，表示对当前行全局匹配替换（与下一个g意义不同） |
| p | Print the current pattern space. 打印当前匹配数据  （pattern space 模式空间） |
| s/regexp/replacement/ | 常说的查找并替换，用一个字符串替换成另一个字符串 |
| h H | Copy/append pattern space to hold space.  h为copy拷贝，H为append追加  拷贝/添加数据从pattern space到hold space （hold space保留空间） |
| g G | Copy/append hold space to pattern space. h为copy拷贝，H为append追加  拷贝/添加数据从hold space到pattern space |
| x | Exchange the contents of the hold and pattern spaces.  交换hold space和pattern space的内容 |
| l | 列出pattern space中的内容，将不可打印字符表示为ASCII码。  长的行将被折行 |
| n N | Read/append the next line of input into the pattern space.  n表示将模式空间中内容的下一行数据读取进来，这个时候模式空间中原有的内容将会被覆盖；  N表示将模式空间中内容的下一行数据读取，然后追加到模式空间中原有的内容后，并用反斜杠隔开。 |
| y/source/dest/ | 表示把一个字符翻译为另外的字符（但是不用于正则表达式） |
| : label | Label for b and t commands. 为b和t命令建立标签 |
| b label | 将执行的指令跳至由“: label”建立的参考位置（标签），如果不存在，则跳至末尾 |
| t label | 自上次输入行读取并自最后一个命令后，再进行一次成功的替代，然后对标签进行标记；如果将标签被省略，则分支到脚本末尾。 |
| T label | 自上次输入行读取并自最后一个命令后，如果不进行成功的替换，则分支到标签；如果将标签被省略，则分支到脚本的结尾。这是一个GNU扩展。 |

支持正则表达式：

正则表达式（REGULAR EXPRESSIONS）实际就是一些特殊字符，被赋予了特定的含义。

基础正则：

1、^word 搜索以word开头的。vi/vim中 ^ 代表一行的开头。

2、word$ 搜索以word结尾的。vi/vim中 $ 代表一行的结尾。

3、^$ 以结尾开头，以开头结尾，表示空行

4、. 代表且只能代表任意一个字符

5、\ 转义符号，让有特殊身份意义的字符，脱掉马甲，还原原型。例：\. 只代表点本身。

6、\* 重复0个或多个前面的字符。

7、.\* 匹配所有字符。例：^.\* 以任意多个字符开头，.\*$以任意多个字符结尾。

8、[abc] 匹配字符集内的任意一个字符。

9、[^abc] 匹配不包含 ^ 后的任意字符的内容。中括号里的 ^ 为取反。

10、a\{n,m\} 重复n到m次前一个重复的字符。若用egrep、sed -r可以去掉斜线。

\{n,\} 重复至少n 次前一个重复的字符。若用egrep、sed -r可以去掉斜线。

\{n\} 重复n 次前一个重复的字符。若用egrep、sed -r可以去掉斜线。

\{,m\} 重复最多m次

扩展正则：

1、+ 重复一个或一个以上前面的字符。

2、? 重复0个或一个前面的字符。

3、| （或者的意思）用或的方式查找多个符合的字符串。

4、() 找出括号内的字符串。

## 【实践操作】：

**例子所用文本：**

[root@nfs01 data]# cat -n test.txt

1 Not every morning wake up your alarm clock, but dream!

2 To me, you are perfect.

3 Life is too short to not experience the madness of love.

4 Life is too short to waste time hating anyone.

5 Sometimes goodbye is the only way.

6 Shall I hang on or shall I let you go?

7 In order to be irreplaceable one must always be different

8 my host name's nfs01

9 my name lzy

10 my qq 623913

11 my motto:

12 Remember these simple guidelines for happiness:

13 1. Free your heart from hate;

14 2. Free your mind from worry;

15 3. Live simply;

16 4. Give more;

17 5. Expect less.

[root@nfs01 data]#

### 1、-e的用法：执行命令行上的编辑指令

[root@nfs01 data]# sed -e '1,7d' -e 's/my/lzy/g' test.txt

lzy host name's nfs01

lzy name lzy

lzy qq 623913

lzy motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

[root@nfs01 data]#

**命令解析：**

**’1,7d’为删除文件中1至7行内容，‘s/my/lzy/g’是将文件中my字符串替换为lzy字符串**

### 2、-f的用法：执行脚本内的指令

[root@nfs01 data]# vim sed.sh

1,7d

s/my/lzy/g

[root@nfs01 data]# sed -f sed.sh test.txt

lzy host name's nfs01

lzy name lzy

lzy qq 623913

lzy motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

[root@nfs01 data]#

### 3、执行多个文件的编辑

在 sed命令列上,一次可执行编辑多个文件,它们跟在编辑指令之后。例如, 替换test.txt、html.log文

件内的 "my" 字符串成 "lzy" , 其命令如下:

**sed -e 's/my/lzy/g' test.txt html.log**

上述命令执行时, sed依test.txt、html.log顺序,执行编辑指令s/my/lzy/g进行字符串的替换。

### 4、替换文件中的数据，s与g结合使用

**一：将文件中含 "my" 字符串的数据行中的 "name" 字符串,替换成为 "mingzi" 字符串**

[root@nfs01 data]# sed -e '/my/s/name/mingzi/g' test.txt

Not every morning wake up your alarm clock, but dream!

To me, you are perfect.

Life is too short to not experience the madness of love.

Life is too short to waste time hating anyone.

Sometimes goodbye is the only way.

Shall I hang on or shall I let you go?

In order to be irreplaceable one must always be different

my host mingzi's nfs01

my mingzi lzy

my qq 623913

my motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

**二：将文件中第5行数据,替换成句子"Those who in quarrels interpose, must often wipe a bloody nose."**

[root@nfs01 data]# sed -e '5c\Those who in quarrels interpose, must often wipe a bloody nose.' test.txt

Not every morning wake up your alarm clock, but dream!

To me, you are perfect.

Life is too short to not experience the madness of love.

Life is too short to waste time hating anyone.

Those who in quarrels interpose, must often wipe a bloody nose.

Shall I hang on or shall I let you go?

In order to be irreplaceable one must always be different

my host name's nfs01

my name lzy

my qq 623913

my motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

**三：将文件中 1 至 6行的数据 , 替换成如下两行资料：**

**How are you?**

**I am OK!**

[root@nfs01 data]# sed -e '1, 6c\

> How are you?\

> I am OK!

> ' test.txt

How are you?

I am OK!

In order to be irreplaceable one must always be different

my host name's nfs01

my name lzy

my qq 623913

my motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

[root@nfs01 data]#

### 5、搬动文件内数据

**一：将文件中的前7行数据 , 搬到文件中第11后输出**

[root@nfs01 data]# vim sed.sh

1,7{

H

d

}

11G

它表示将文件中的前7行数据,先储存在hold space之后删除；指令11G表示将hold space内的资料,添加在文件中的第11行数据后输出。

[root@nfs01 data]# sed -f sed.sh test.txt

my host name's nfs01

my name lzy

my qq 623913

my motto:

Not every morning wake up your alarm clock, but dream!

To me, you are perfect.

Life is too short to not experience the madness of love.

Life is too short to waste time hating anyone.

Sometimes goodbye is the only way.

Shall I hang on or shall I let you go?

In order to be irreplaceable one must always be different

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

[root@nfs01 data]#

**二、将文件中含 "my" 字符串的数据行,搬至html.log文件中储存**

[root@nfs01 data]# sed -e '/my/w html.log' test.txt

Not every morning wake up your alarm clock, but dream!

To me, you are perfect.

Life is too short to not experience the madness of love.

Life is too short to waste time hating anyone.

Sometimes goodbye is the only way.

Shall I hang on or shall I let you go?

In order to be irreplaceable one must always be different

my host name's nfs01

my name lzy

my qq 623913

my motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

[root@nfs01 data]# ll

total 12

-rw-r--r-- 1 root root 56 Nov 16 01:50 html.log

-rw-r--r-- 1 root root 15 Nov 16 01:44 sed.sh

-rw-r--r-- 1 root root 533 Nov 16 01:05 test.txt

[root@nfs01 data]# cat html.log

my host name's nfs01

my name lzy

my qq 623913

my motto:

[root@nfs01 data]#

**三、将html.log文件中内容,搬至文件中含 "motto" 字符串的数据行**

[root@nfs01 data]# sed -e '/motto/r html.log' test.txt

Not every morning wake up your alarm clock, but dream!

To me, you are perfect.

Life is too short to not experience the madness of love.

Life is too short to waste time hating anyone.

Sometimes goodbye is the only way.

Shall I hang on or shall I let you go?

In order to be irreplaceable one must always be different

my host name's nfs01

my name lzy

my qq 623913

my motto:

my host name's nfs01

my name lzy

my qq 623913

my motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

### 6、删除文件中的数据

[root@nfs01 data]# vim test.sh

Not every morning wake up your alarm clock, but dream!

To me, you are perfect.

Sometimes goodbye is the only way.

Shall I hang on or shall I let you go?

In order to be irreplaceable one must always be different

my host name's nfs01

my name lzy

my qq 623913

my motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

**将文件内所有空白行全部删除**

[root@nfs01 data]# sed -e '/^$/d' test.sh

Not every morning wake up your alarm clock, but dream!

To me, you are perfect.

Sometimes goodbye is the only way.

Shall I hang on or shall I let you go?

In order to be irreplaceable one must always be different

my host name's nfs01

my name lzy

my qq 623913

my motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

**删除指定行数据：**

[root@nfs01 data]# sed -e '1,7d' test.txt

my host name's nfs01

my name lzy

my qq 623913

my motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

[root@nfs01 data]#

### 7、s，函数参数 s 表示替换(substitute)文件内字符串

**函数参数 s 表示替换(substitute)文件内字符串**

格式：**[address1[ ,address2]] s/pattern/replacemen/[flag]**

函数参数 s 最多与两个地址参数配合。

关于 "s/pattern/replacement/[flag]" 的说明:

pattern : 它为reguler expression（正则表达式）字符串，它表示文件中要被替换的字符串。

replacement : 它为一般字符串。但其内出现下列字符有特别意义 :

**&** : 代表前 pattern 字符串。

例如：**sed -e 's/test/& my car/' 文件名**

指令中, & 代表 pattern 字符串 "test"；故执行后, 数据文件的 "test" 被替换成 "test my car"。

**\n** : 代表 pattern 中被第 n 个 \( 、\)所括起来的字符串。

例如：**sed -e 's/\(test\) \(my\) \(car\)/[\2 \3 \1]/' 文件名**

指令中, \1 表示 "test"、\2 表示 "my"、\3 表示 "car" 字符串。故执行后,数据文件的 "test my car" 被替换成 "[my car test]"。

**\** : 转义字符，可用它来还原一些特殊符号(如上述的 & 与 \ )本身字面上的意义 ,或用它来代表换行。

**flag** : 主要用它来控制一些替换情况 :

1、当 flag 为 g 时, 代表替换所有符合(match)的字符串。

2、当 flag 为十进制数 m 时,代表替换行内第 m 个符合的字符串。

3、当 flag 为 p 时 , 代表替换第一个符合 pattern 的字符串后, 将数据输出标准输出文件。

4、当 flag 为 w wfile 时,代表替换第一个符合 pattern 的字符串后 ,输出到 wfile 文件内(如果 wfile 不存在,则会重新创建名为 wfile 的文件)。

5、当没有 flag 时,则将资料行内第一个符合 pattern 的字符串以 replacement 字符串来替换 。

6、delimiter（定界符）: 在 "/pattern/replace/[flag] " 中 "/" 被当成一种delimiter。除了空白(blank)、换行(newline) 之外,使用者可用任何字符作为 delimiter。例如下述编辑指令：

**s#/usr#/usr1#g**

上述命令中“#”为 delimiter。如果用 "/" 做 delimiter，则sed会将 pattern 与 replacement 中的 "/" 当成delimiter 而发生错误。

范例: 替换test.txt文件内 "my" 字符串成 "your",同时将这些数据行存入test.log文件内。

说明: 用函数参数 s 指示 sed将 "my" 字符串替换成 "your",另外用 s argument 中的 flag w 指示 sed将替换过的资料行存入 test.log文件内。

命令: **sed -e 's/my/your/w test.log' test.txt**

### 8、d，函数参数 d 表示删除数据行

**函数参数 d 表示删除数据行**

格式: **[address1[ ,address2]] d**

函数参数 d 最多与两个地址参数配合。

sed 执行删除动作情况如下 :

将 pattern space 内符合地址参数的数据删除。

将下一笔资料读进 pattern space 。

重新执行 sed script。

[root@nfs01 data]# sed '1,11d' test.txt

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

### 9、a，函数参数 a 表示将资料添加到文件中

**函数参数 a 表示将资料添加到文件中**

格式：**[address1] a\ 用户所输入的数据**

函数参数 a 最多与一个地址参数配合。

函数参数 a 紧接着 "\" 字符用来表示此行结束,用户所输入的数据必须从下一行输入。如果数据超过一行 , 则须在每行的结尾加入"\"。

sed 执行添加动作情况如下: 当 pattern space 内数据输出后, sed 跟着输出用户所输入的数据。

范例: 添加 "多任务操作系统" 在含 "UNIX" 字符串的数据行后。假设test.log的内容如下:

UNIX

说明: 用函数参数 a 将所输入的数据添加在含 "UNIX" 字符串的数据行后。

sed 命令:

**sed -e '/UNIX/a\**

**多任务操作系统**

**' test.log**

[root@nfs01 data]# sed -e '/UNIX/a\

> 多任务操作系统

> ' test.log

UNIX

多任务操作系统

### 10、i，函数参数 i 表示将资料插入文件中

**函数参数 i 表示将资料插入文件中**

格式: **[address1] i\ 用户所输入的数据**

函数参数 i 最多与一个地址参数配合。

函数参数 i 紧接着 "\" 字符用来表示此行结束,用户所输入的数据必须从下一行输入。如果数据超过一行 , 则须在每行的结尾加入"\"。

sed 执行插入动作的情况如下: 在 pattern space 内数据输出前,sed先输出用户所输入的数据。

范例：将“Linux not is”插在test.log文件中含“UNIX”字符的数据行之前

说明: 用函数参数 i 将数据行 " Linux not is " 插在含 " UNIX " 的数据行之前

命令：

**sed -e '/UNIX/i\**

**Linux not is**

**' test.log**

[root@nfs01 data]# sed -e '/UNIX/i\

> Linux not is

> ' test.log

Linux not is

UNIX

### 11、c，函数参数 c 表示改变文件中的数据

**函数参数 c 表示改变文件中的数据**

格式: **[address1[ ,address2]]c\ 用户所输入的数据**

函数参数 c 最多与两个地址参数配合。

函数参数 c 紧接着 "\" 字符用来表示此行结束 , 使用者所输入的数据必须从下一行输入。如果数据超过一行 , 则须在每行的结尾加入"\"。

sed 执行改变动作的情况: 在 pattern space 内数据输出时, sed改变它成为使用者所输入的数据

**范例：将文件中 1 至 6行的数据 , 替换成如下两行资料：**

**How are you?**

**I am OK!**

**[root@nfs01 data]# sed -e '1, 6c\**

**> How are you?\**

**> I am OK!**

**> ' test.txt**

**How are you?**

**I am OK!**

**In order to be irreplaceable one must always be different**

**my host name's nfs01**

**my name lzy**

**my qq 623913**

**my motto:**

**Remember these simple guidelines for happiness:**

**1. Free your heart from hate;**

**2. Free your mind from worry;**

**3. Live simply;**

**4. Give more;**

**5. Expect less.**

**[root@nfs01 data]#**

### 12、p，函数参数 p 表示打印出数据

**函数参数 p 表示打印出数据**

格式: **[address1[,address2]] p**

函数参数 p 最多与两个地址参数配合。

sed 执行印出动作的情况如下: sed 拷备一份 pattern space 内容至标准输出文件

范例：将文件中8至11行数据打印出来。

[root@nfs01 data]# sed -n '8,11p' test.txt

my host name's nfs01

my name lzy

my qq 623913

my motto:

[root@nfs01 data]#

### 13、l，函数参数 l , 除可将资料中的 nonprinting character以 ASCII码列出外, 其于均与函数参数 p相同

例如,将input.dat 文件中的“^[ ”以 ASCII 码印出：

The Great ^[ is a movie starring Steve McQueen.

执行命令 sed -e 'l' input.dat:

### 14、r，函数参数 r 表示读入其它文件内容到当前输出中

**函数参数 r 表示读入其它文件内容到当前输出中**

格式: [address1] r 其它文件名

函数参数 r 最多与一个地址参数配合。

在指令中,函数参数 r 与其它文件名之间,只能有一空格。

sed 执行读入动作的情况如下: 在 pattern space内数据输出后, sed读出其他文件的内容跟着输出。当其他文件不存在时,sed照样执行其它指令而不会有任何错误讯息产生。

范例：将test1.txt文件中的内容读取到当前文件含“my”字符的数据行后输出

[root@nfs01 data]# sed -e '/my/r test1.txt' test.txt

Not every morning wake up your alarm clock, but dream!

To me, you are perfect.

Life is too short to not experience the madness of love.

Life is too short to waste time hating anyone.

Sometimes goodbye is the only way.

Shall I hang on or shall I let you go?

In order to be irreplaceable one must always be different

my host name's nfs01

Those who in quarrels interpose, must often wipe a bloody nose.

my name lzy

Those who in quarrels interpose, must often wipe a bloody nose.

my qq 623913

Those who in quarrels interpose, must often wipe a bloody nose.

my motto:

Those who in quarrels interpose, must often wipe a bloody nose.

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

### 15、w，函数参数 w 表示将文件中的数据写到其他文件中

**函数参数 w 表示将文件中的数据写到其他文件中**

格式: [address1[,address2]] w 其它文件名

函数参数 w 最多与两个地址参数配合。

在指令中, 函数参数 w 与其他文件名,只能有一空格。

sed 执行写出动作的情况如: 将 pattern space 内资料写到它文件内。数据写入时,会取代(overwrite)原来档案内的数据。另外, 当其他文件不存在时, sed会重新产生(creat)它。

范例：将文件中含“my”的数据行写入test.sh文件中

[root@nfs01 data]# sed -e '/my/w test.sh' test.txt

Not every morning wake up your alarm clock, but dream!

To me, you are perfect.

Life is too short to not experience the madness of love.

Life is too short to waste time hating anyone.

Sometimes goodbye is the only way.

Shall I hang on or shall I let you go?

In order to be irreplaceable one must always be different

my host name's nfs01

my name lzy

my qq 623913

my motto:

Remember these simple guidelines for happiness:

1. Free your heart from hate;

2. Free your mind from worry;

3. Live simply;

4. Give more;

5. Expect less.

[root@nfs01 data]# cat test.sh

my host name's nfs01

my name lzy

my qq 623913

my motto:

[root@nfs01 data]#

### 16、y，函数参数 y 表示转换数据中的字符

**函数参数 y 表示转换数据中的字符**

格式: **[address1[ ,address2]]y /xyz.../abc.../**

函数参数最多配合两个地址参数。

指令中, /abc.../xyz.../(x 、 y 、 z 、 a 、 b 、 c 代表某些字符)为 y 的argument其中 abc... 与 xyz... 的字符个数必须相同。

sed 执行转换时,将 pattern space内数据内的 a 字符转换成 x 字符、b 字符转换成 y 字符、c 字符转换成 z 字符 ...。

范例:

题目: 将test1.txt文件中的小写字母改成大写，test1.txt文件的内容如下：

[root@nfs01 data]# cat test1.txt

Those who in quarrels interpose, must often wipe a bloody nose.

说明:利用函数参数 y 指示 sed做字母大小的转换。

sed 命令如下:

sed -e 'y/abcdefghijklmnopqrstuvwxyz/ABCDEFGHIJKLMNOPQRSTUVWXYZ/' test1.txt

[root@nfs01 data]# sed -e 'y/abcdefghijklmnopqrstuvwxyz/ABCDEFGHIJKLMNOPQRSTUVWXYZ/' test1.txt

THOSE WHO IN QUARRELS INTERPOSE, MUST OFTEN WIPE A BLOODY NOSE.

### 17、!，函数参数 ! 表示不执行函数参数

**函数参数 ! 表示不执行函数参数**

格式：[address1[ , address2]] ! 函数参数

表示,对符合地址参数之数据不执行函数参数。

例如：删除除了含 "my" 字符串的所有数据行,则执行如下命令

sed -e '/my/!d' test.txt

[root@nfs01 data]# sed -e '/my/!d' test.txt

my host name's nfs01

my name lzy

my qq 623913

my motto:

### 18、n，函数参数 n 表示读入下一行资料

**函数参数 n 表示读入下一行资料**

格式: [address1[ ,address2]] n

函数参数 n 最多配合两个地址参数。

sed 执行读入下一行动作的情况如: 输出在 pattern space 的数据，将下一笔资料读到 pattern space，执行下一个编辑指令。

范例: 输出test.txt文件内偶数行资料

说明: 在命令列上以选项 -n ,将数据输出的控制权转给指令。利用函数参数 n 将下一行数据(偶数行)取代 pattern space 内的资料行(奇数行)。利用函数参数 p 将 pattern space 内的数据(偶数行)输出。

最后 ,整个输出只有原先文件内的偶数行数据。

sed命令: sed -n -e 'n' -e 'p' test.txt

[root@nfs01 data]# sed -n -e 'n' -e 'p' test.txt

To me, you are perfect.

Life is too short to waste time hating anyone.

Shall I hang on or shall I let you go?

my host name's nfs01

my qq 623913

Remember these simple guidelines for happiness:

2. Free your mind from worry;

4. Give more;

### 19、q，函数参数 q 表示跳离 sed编辑

函数参数 q 表示跳离 sed

格式:[address1] q

函数参数 q 最多配合一个地址参数。

sed 执行跳离动作时,它停止输入 pattern space 数据,同时停止数据送到标准输出文件。

范例:对文件文件执行 script\_file 内的编辑指令, 除非遇到 " Life " 字符串。

说明: 无论 script\_file 内是何种指令,使用者只要在命令列上用指令/Life/q, 函数参数 q 会强迫 sed遇到 " Life " 时做跳离动作。

sed命令 :sed -e '/Life/q' -f script\_file test.txt

[root@nfs01 data]# cat sed.sh

1,7d

s/my/lzy/g

[root@nfs01 data]# sed -e '/Life/q' -f sed.sh test.txt

Life is too short to not experience the madness of love.

### 20、=，函数参数 = 表示打印出文件内容的行数

**函数参数 = 表示打印出文件内容的行数**

格式:[address1 ,[address2]] =

函数参数 = 最多配合两个地址参数。

执行时,行数将在数据输出前先输出。

范例: 打印出 test.txt文件内容行数

说明 : 用函数参数 = 来印出资料的行数。

sed命令: sed -e '=' test.txt

[root@nfs01 data]# sed -e '=' test.txt

1

Not every morning wake up your alarm clock, but dream!

2

To me, you are perfect.

3

Life is too short to not experience the madness of love.

4

Life is too short to waste time hating anyone.

5

Sometimes goodbye is the only way.

6

Shall I hang on or shall I let you go?

7

In order to be irreplaceable one must always be different

8

my host name's nfs01

9

my name lzy

10

my qq 623913

11

my motto:

12

Remember these simple guidelines for happiness:

13

1. Free your heart from hate;

14

2. Free your mind from worry;

15

3. Live simply;

16

4. Give more;

17

5. Expect less.

18

### 21、#，在script file 内,函数参数 # 后的文字为注解。当注解文字超过多行时,其行间须以 "\" 换行字符相隔

### 22、N，函数参数 N 表示添加下一笔资料在 pattern space 内

**函数参数 N 表示添加下一笔资料在 pattern space 内**

格式:[address1 ,[address2]] N

函数参数 N 最多配合两个地址参数。

sed执行时,将下一行数据读入并添加在 pattern space内,数据行间以换行字符(embedded newline character)分隔。此外,在替换时,换行字符可用 \n 来 match。

范例: 将下述两行数据合并。假设 input.dat 的内容如下 :

The UNIX

Operating System

说明: 先利用函数参数 N 将两行数据置于 pattern space 内,在利用函数参数 s/\n/ / 将两行数据间的分隔号 \n 以空白替代 ,如此两行数据变成一行输出。

sed命令: sed -e 'N' -e 's/\n/ /' input.dat

[root@nfs01 data]# sed -e 'N' -e 's/\n/ /' input.dat

The UNIX Operating System

### 23、D，函数参数 D 表示删除 pattern space 内的第一行资料

**函数参数 D 表示删除 pattern space 内的第一行资料**

格式: [address1,address2]D

函数参数 D 最多配合两个地址参数。

函数参数 D 与 d 的比较如下 :

当 pattern space 内只有一数据行时,D 与 d 作用相同。

当 pattern space 内有多行资料行时

D 表示只删除 pattern space 内第一行资料; d 则全删除。

D 表示执行删除后, pattern space 内不添加下一笔数据,而将剩下的数据重新执行 sed script; d 则读入下一行后执行sed script。

## 【技巧拓展】：

例：过滤出已知当前目录下oldboy中的所有一级目录

[root@oldboy oldboy]# tree

.

├── hello

│   └── abc.log

├── hi

└── test.txt

2 directories, 2 files

[root@oldboy oldboy]# ls -lF|sed -n '/^d/p'

drwxr-xr-x. 2 root root 4096 Oct 4 15:08 hello/

drwxr-xr-x. 2 root root 4096 Oct 4 15:07 hi/

[root@oldboy oldboy]#

**例：sed 后向引用 sed –r ‘s#(.\*)#\1#g’**

**[root@oldboy ~]# echo "oldboy oldgirl">test.txt**

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

**oldboy oldgirl**

**[root@oldboy ~]# sed -r 's#(.\*)#\1#g' test.txt**

**oldboy oldgirl**

**[root@oldboy ~]# sed -r 's#(.\*) (.\*)#\1#g' test.txt**

**oldboy**

**[root@oldboy ~]# sed -r 's#(.\*) (.\*)#\2#g' test.txt**

**oldgirl**

**例：取出/etc/hosts中的第四行的0644**

*[root@oldboy ~]# stat /etc/hosts|sed -n '4p'*

*Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)*

**[root@oldboy ~]# stat /etc/hosts|sed -n '4p'|sed 's#^.\*ss: (##g'|sed 's#/-.\*$##g'**

**0644**

**[root@oldboy ~]# stat /etc/hosts|sed -n '4p'|sed -r 's#^.\*s: \((.\*)/-.\*$#\1#g'**

**0644**

**[root@oldboy ~]# stat /etc/hosts|sed -nr '4s#^.\*s: \((.\*)/-.\*$#\1#gp'**

**0644**

**[root@oldboy ~]# stat /etc/hosts|sed -nr '4s#(^.\*s: \()(.\*)(/-.\*$)#\2#gp'**

**0644**

**扩展：'Ms###Ng'的使用**

s：单独使用->将每一行中第一处匹配的样式进行替换

s、g合用表示全部替换

Ms->对第M行处理，无g，只处理第一处匹配，有g第M行全部替换

Ng->对每一行，从第N处开始替换

Ms、Ng合用表示只对第M行从第N处匹配开始替换

[root@oldboy ~]# cat test.txt

a a a a a

b b b b b

a a a a a

a a a a a

[root@oldboy ~]# sed '2s#b#c#g' test.txt

a a a a a

c c c c c

a a a a a

a a a a a

[root@oldboy ~]# sed 's#a#c#g' test.txt

c c c c c

b b b b b

c c c c c

c c c c c

[root@oldboy ~]# sed 's#a#c#3g' test.txt

a a c c c

b b b b b

a a c c c

a a c c c

[root@oldboy ~]# sed '3s#a#c#3g' test.txt

a a a a a

b b b b b

a a c c c

a a a a a

**例：将找到的文件都插入一段字符。**

[root@oldboy ~]# find / -type f -name "test.txt"|xargs sed ir '1 iPort 52113\nPermitRootLogin no\n

PermitEmptyPasswords no\nUseDNS no\nGSSAPIAuthentication no\ListenAddress 10.0.0.8:52113'