sed

MengChunlei

April 11, 2021

1 基本用法

sed 全称是 stream editor. 简单说, 它是按照行来处理文本的一个工具. 本文将按照以下 12 个部分来总结它的用法:

- 替换

- 地址使用

- 删除

- 插入

- 整行替换和转换

- 打印

- 读写文件

- 多行用法

- 空间保持和排除命令

- 流改变

- 模式替代

- 用例示范

2 替换

替换的命令: s/origin/replacement/. 多个替换命令可以串联, 用分号隔开.

Listing 1: Substitution

```
1  [chunleimeng]$ echo "hello world" | sed 's/hello/Hello/'
2  Hello world
3  [chunleimeng]$ cat file1
4  The quick brown fox jumps over the lazy dog.
5  The quick brown fox jumps over the lazy dog.
6  The quick brown fox jumps over the lazy dog.
7  The quick brown fox jumps over the lazy dog.
8  [chunleimeng]$ sed -e 's/brown/green/; s/dog/cat/' file1
9  The quick green fox jumps over the lazy cat.
10 The quick green fox jumps over the lazy cat.
11 The quick green fox jumps over the lazy cat.
12 The quick green fox jumps over the lazy cat.
13 [chunleimeng]$
```

命令可以放在一个文件里面. 也可以用 −e 以写在多行上.

Listing 2: Substitution

```
1  [chunleimeng]$ cat script1.sed
2  s/brown/green/
3  s/dog/cat/
4  [chunleimeng]$ sed −f script1.sed file1
5  The quick green fox jumps over the lazy cat.
6  The quick green fox jumps over the lazy cat.
7  The quick green fox jumps over the lazy cat.
8  The quick green fox jumps over the lazy cat.
9  [chunleimeng]$ sed −e '
10 > s/brown/green/
11 > s/dog/cat/' file1
12 The quick green fox jumps over the lazy cat.
13 The quick green fox jumps over the lazy cat.
14 The quick green fox jumps over the lazy cat.
15 The quick green fox jumps over the lazy cat.
```

替换还有四种控制:

- 数字, 替换第几处
- g, 全部替换
- p, 打印替换的行
- w file, 将替换的结果写到文件

Listing 3: Substitution

```
1  [chunleimeng]$ cat file2
2  this is test a test
3  this is test
4  no key
5  [chunleimeng]$ sed 's/test/new/' file2   /*默认替换第一次出现的地方*/
6  this is new a test
7  this is new
8  no key
9  [chunleimeng]$ sed 's/test/new/2' file2 /*替换第二次出现的地方*/
10 this is test a new
11 this is test
12 no key
13 [chunleimeng]$ sed 's/test/new/g' file2
14 this is new a new
15 this is new
16 no key
17 [chunleimeng]$ sed −n 's/test/new/p' file2   /*−n表示不产生输出*/
18 this is new a test
19 this is new
20 [chunleimeng]$ sed −n 's/test/new/2p' file2
21 this is test a new
22 [chunleimeng]$ sed −n 's/test/new/w result.txt' file2
23 [chunleimeng]$ cat result.txt
24 this is new a test
25 this is new
```

3 地址使用

如果只想把命令作用于某些行或者不作用于某些行, 需要使用地址. 有两种方式:

- 以数字表示区间

- 用本文模式找到特定的行

使用数字表示区间的样例:

Listing 4: Address

```
1  [chunleimeng]$ sed '2s/dog/cat/' file1   /*只替换第二行*/
2  The quick brown fox jumps over the lazy dog.
3  The quick brown fox jumps over the lazy cat.
4  The quick brown fox jumps over the lazy dog.
5  The quick brown fox jumps over the lazy dog.
6  [chunleimeng]$ sed '2,3s/dog/cat/' file1  /*替换第二行到第三行*/
7  The quick brown fox jumps over the lazy dog.
8  The quick brown fox jumps over the lazy cat.
9  The quick brown fox jumps over the lazy cat.
10 The quick brown fox jumps over the lazy dog.
11 [chunleimeng]$ sed '2,$s/dog/cat/' file1  /*替换第二行到最后一行*/
12 The quick brown fox jumps over the lazy dog.
13 The quick brown fox jumps over the lazy cat.
14 The quick brown fox jumps over the lazy cat.
15 The quick brown fox jumps over the lazy cat.
16 [chunleimeng]$ sed '2{
17 > s/fox/elephant/
18 > s/dog/cat/}' file1
19 The quick brown fox jumps over the lazy dog.
20 The quick brown elephant jumps over the lazy cat.
21 The quick brown fox jumps over the lazy dog.
22 The quick brown fox jumps over the lazy dog.
23 [chunleimeng]$
```

使用文本模式示例:

Listing 5: Address

```
1  [chunleimeng]$ cat file2
2  this is test a test
3  this is test
4  no key
5  [chunleimeng]$ sed '/test a/s/test/abc/g' file2
6  this is abc a abc
7  this is test
8  no key
9  [chunleimeng]$
```

4 删除

删除的命令是 d.

Listing 6: Delete

```
1  [chunleimeng]$ cat file3
2  this is line number 1
3  this is line number 2
4  this is line number 3
5  this is line number 4
6  this is line number 1 again
7  line six
8  line seven
9  [chunleimeng]$ sed 'd' file3   /*删除所有行*/
10 [chunleimeng]$ sed '3d' file3  /*删除第三行*/
11 this is line number 1
12 this is line number 2
13 this is line number 4
14 this is line number 1 again
15 line six
16 line seven
17 [chunleimeng]$ sed '3,$d' file3  /*删除第三行到结尾*/
18 this is line number 1
19 this is line number 2
20 [chunleimeng]$ sed '/number 1/d' file3  /*删除包含 'number 1' 的行*/
21 this is line number 2
22 this is line number 3
23 this is line number 4
24 line six
25 line seven
26 [chunleimeng]$ sed '/1/,/3/d' file3  /*删除包含1的行到包含3的行. 数字1第二次出现的时候没有*
27 this is line number 4                 /*对应的3的行,所以后面的都删除了*/
```

5 插入

插入有两种, i 表示在指定行前面插入, a 表示在指定行后面插入.

Listing 7: Insert

```
1  [chunleimeng]$ cat file4
2  this is line number 1
3  this is line number 2
4  [chunleimeng]$ sed '2i\insert line' file4  /*在第二行前面插入*/
5  this is line number 1
6  insert line
7  this is line number 2
8  [chunleimeng]$ sed '$a\insert line' file4  /*在最后一行后面插入*/
9  this is line number 1
10 this is line number 2
11 insert line
12 [chunleimeng]$ sed '2i\   /*插入多行*/
13 > insert line1\
14 > insert line2' file4
15 this is line number 1
16 insert line1
17 insert line2
18 this is line number 2
```

6 整行替换和转换

整行替换的命令是 c.

```
1  [chunleimeng]$ cat file3
2  this is line number 1
3  this is line number 2
4  this is line number 3
5  this is line number 4
6  this is line number 1 again
7  line six
8  line seven
9  [chunleimeng]$ sed '3c\changed line' file3   /*替换第三行*/
10 this is line number 1
11 this is line number 2
12 changed line
13 this is line number 4
14 this is line number 1 again
15 line six
16 line seven
17 [chunleimeng]$ sed '/number 1/c\changed line' file3 /*替换包含'number 1'的行*/
18 changed line
19 this is line number 2
20 this is line number 3
21 this is line number 4
22 changed line
23 line six
24 line seven
25 [chunleimeng]$ sed '2,3c\changed line' file3  /*替换第2行到第3行*/
26 this is line number 1
27 changed line
28 this is line number 4
29 this is line number 1 again
30 line six
31 line seven
```

转换的命令是: y/inchars/outchars, 其中 inchars 和 outchars 要一样长. 它会把每一个 inchars 替换为对应的
outchars.

Listing 9: Transform

```
1  [chunleimeng]$ sed 'y/123/789/' file3
2  this is line number 7
3  this is line number 8
4  this is line number 9
5  this is line number 4
6  this is line number 7 again
7  line six
8  line seven
9  [chunleimeng]$ echo '111222333' | sed 'y/123/789/'   /*会替换所有的字符*/
10 777888999
```

7 打印

控制打印的有三个命令:

- p, 打印本文行
- =, 打印行号
- l, 列出行

Listing 10: Print

```
 1  [chunleimeng]$ cat file5
 2  this is line number 1
 3  this is line number 2
 4  this is line number 3
 5  this is line number 4
 6  [chunleimeng]$ sed -n '/number 3/p' file5
 7  this is line number 3
 8  [chunleimeng]$ sed -n '2,3p' file5
 9  this is line number 2
10  this is line number 3
11  [chunleimeng]$ sed -n '/3/{
12  > p
13  > s/line/test/p
14  > }' file5
15  this is line number 3
16  this is test number 3
17  [chunleimeng]$ sed '=' file5
18  1
19  this is line number 1
20  2
21  this is line number 2
22  3
23  this is line number 3
24  4
25  this is line number 4
26  [chunleimeng]$ sed -n '/number 4/{
27  > =
28  > p
29  > }' file5
30  4
31  this is line number 4
32  [chunleimeng]$ cat file6
33  tag     test    end
34  [chunleimeng]$ sed -n 'l' file6  /*tab会显示为\t*/
35  tag\ttest\tend$
36  [chunleimeng]$
```

8 读写文件

- w, 写文件
- r, 读文件

Listing 11: File

```
 1  [chunleimeng]$ cat file5
 2  this is line number 1
 3  this is line number 2
 4  this is line number 3
 5  this is line number 4
 6  [chunleimeng]$ sed '1,2w result.txt' file5
 7  this is line number 1
 8  this is line number 2
 9  this is line number 3
10  this is line number 4
11  [chunleimeng]$ cat result.txt
12  this is line number 1
13  this is line number 2
14  [chunleimeng]$ cat file7
15  insert line1
16  insert line2
17  [chunleimeng]$ sed '3r file7' file5 /*将 file7 中的内容插入第三行后*/
18  this is line number 1
19  this is line number 2
20  this is line number 3
21  insert line1
22  insert line2
23  this is line number 4
24  [chunleimeng]$ sed '$r file7' file5
25  this is line number 1
26  this is line number 2
27  this is line number 3
28  this is line number 4
29  insert line1
30  insert line2
```

9　多行用法

- next 命令

- 多行删除

- 多行打印

next 命令有两种, 小写 n 表示移动到下一行, 大写 N 表示将下一行的内容附加到当前行的后面.

Listing 12: next-n

```
 1  [chunleimeng]$ cat file8
 2  first line
 3  second line
 4  third line
 5  forth line
 6  [chunleimeng]$ sed '/first/{n ; d}' file8 /*删除 first 行的后面一行*/
 7  first line
 8  third line
 9  forth line
10  [chunleimeng]$
```

```
1  [chunleimeng]$ cat file9
2  first line hello
3  world second line
4  third line
5  abc hello world ef
6  we hello world
7  [chunleimeng]$ sed 'N; s/hello world/HELLO WORLD/' file9  /*可以发现，第一行第二行换行的*/
8  first line hello                                          /*hello world没有替换成功*/
9  world second line
10 third line
11 abc HELLO WORLD ef
12 we hello world
13 [chunleimeng]$ sed 'N; s/hello.world/HELLO WORLD/' file9  /*上面的问题解决了，但是第一行*/
14 first line HELLO WORLD second line                        /*第二行合并了*/
15 third line
16 abc HELLO WORLD ef
17 we hello world
18 [chunleimeng]$ sed '                  /*分两种情况处理.问题是最后一行没有成功.因为没有*/
19 > N                                   /*下一行，所以N命令失败了，从而结束. 后面一节会*/
20 > s/hello\nworld/HELLO\nWORLD/        /*解决这个问题*/
21 > s/hello world/HELLO WORLD/' file9
22 first line HELLO
23 WORLD second line
24 third line
25 abc HELLO WORLD ef
26 we hello world
```

接下来是多行删除. 命令是 D.

```
1  [chunleimeng]$ cat file9
2  first line hello
3  world second line
4  third line
5  abc hello world ef
6  we hello world
7  [chunleimeng]$ sed 'N; /hello.world/d' file9  /*文件的第1-2行删除了一次,第3-4行删除了一次*/
8  we hello world
9  [chunleimeng]$ sed 'N; /hello.world/D' file9  /*删除目标串所在行的前一行*/
10 world second line
11 third line
12 we hello world
```

接下来是多行打印.

```
1  [chunleimeng]$ sed -n 'N; /hello\nworld/p' file9  /*打印合并后的所有*/
2  first line hello
3  world second line
4  [chunleimeng]$ sed -n 'N; /hello\nworld/P' file9  /*打印前面一行*/
5  first line hello
6  [chunleimeng]$
```

10 空间保持和排除命令

sed 命令工作的空间称作模式空间, 还有一块缓冲区叫做保持空间. 有一些命令可以在这两个之间交换数据:

- h: 从模式空间复制到保持空间

- H: 将模式空间附加到保持空间

- g: 从保持空间复制到模式空间

- G: 将保持空间附加到模式空间

- x: 交换模式空间和保持空间

Listing 16: Hold Space

```
1  [chunleimeng]$ cat file9
2  first line hello
3  world second line
4  third line
5  abc hello world ef
6  we hello world
7  [chunleimeng]$ sed -n '/first/{h; n; p; g; p}' file9  /*先打印first行的下一行*/
8  world second line                                      /*再打印first行*/
9  first line hello
```

排除命令是感叹号!, 它表示相反的操作.

Listing 17: Negate

```
1   [chunleimeng]$ cat file9
2   first line hello
3   world second line
4   third line
5   abc hello world ef
6   we hello world
7   [chunleimeng]$ sed -n '/hello/!p' file9  /*包含hello的行不输出*/
8   world second line
9   third line
10  [chunleimeng]$ sed '$!N                  /*最后一行不执行N操作*/
11  > s/hello\nworld/HELLO\nWORLD/
12  > s/hello world/HELLO WORLD/' file9
13  first line HELLO
14  WORLD second line
15  third line
16  abc HELLO WORLD ef
17  we HELLO WORLD
18  [chunleimeng]$ sed -n '{1!G; h; $p}' file9  /*倒序输出文件:*/
19  we hello world                              /*   1: 第一行不执行G操作*/
20  abc hello world ef                          /*   2: 到最后一行的时候才执行p操作*/
21  third line
22  world second line
23  first line hello
```

11 流改变

正常的执行规则对于文本的每一行, 顺序执行脚本中的每个命令. 流改变的意思是可以改变这个规则, 包括两个命令, 一个命令类似 C 语言的 goto, 即满足某个条件的话会跳转到某个地方去执行之后的命令; 另一个也是类似 goto, 不过不是某个条件成立, 而是之前的命令执行成功时跳转. 第一个命令是 b, 第二个命令是 t.

Listing 18: Stream Change-b

```
 1  [chunleimeng]$ cat file9
 2  first line hello
 3  world second line
 4  third line
 5  abc hello world ef
 6  we hello world
 7  [chunleimeng]$
 8  [chunleimeng]$ sed '2,3b; s/world/WORLD/' file9    /*第2-3行直接跳转到脚本的最后*/
 9  first line hello                                   /*其他行不跳转， 去执行后面的替换命令*/
10  world second line
11  third line
12  abc hello WORLD ef
13  we hello WORLD
14  [chunleimeng]$ sed '/hello/b label; s/world/WORLD/; :label s/hello/HELLO/' file9
15  first line HELLO                    /*包含 hello 的行跳转到 label 去替换 hello*/
16  WORLD second line                   /*不包含 hello 的会顺序执行所有的命令*/
17  third line
18  abc HELLO world ef
19  we HELLO world
20  [chunleimeng]$ echo "this, is, a, test, to, remove, commas" | sed -n '
21  :start
22  s/,//1p              /*去掉逗号并打印*/
23  /,/b start'          /*替换后还有逗号就跳转, 否则结束*/
24  this is, a, test, to, remove, commas
25  this is a, test, to, remove, commas
26  this is a test, to, remove, commas
27  this is a test to, remove, commas
28  this is a test to remove, commas
29  this is a test to remove commas
30  [chunleimeng]$ sed '
31  > s/hello/HELLO/                     /*替换 hello 成功， 则跳转到脚本最后*/
32  > t                                  /*否则不跳转， 替换 world*/
33  > s/world/WORLD/' file9
34  first line HELLO
35  WORLD second line
36  third line
37  abc HELLO world ef
38  we HELLO world
39  [chunleimeng]$ echo "this, is, a, test, to, remove, commas" | sed -n '
40  :start
41  s/,//1p
42  t start'            /*替换逗号成功则跳转到 start*/
43  this is, a, test, to, remove, commas
44  this is a, test, to, remove, commas
45  this is a test, to, remove, commas
46  this is a test to, remove, commas
```

```
47  this is a test to remove, commas
48  this is a test to remove commas
```

12 模式替代

这个跟正则表达式中的反向引用类似, 就是使用匹配的模式.

Listing 19: Stream Change-b

```
1  [chunleimeng]$ echo 'The cat sleeps in his hat' | sed 's/\(.at\)/"\1"/g'
2  The "cat" sleeps in his "hat"    /*将cat和hat加上引号*/
3  [chunleimeng]$ echo 'abc def ghij afc' | sed 's/a\(.\)c/XX\1XX/g'
4  XXbXX def ghij XXfXX  /*将abc和afc两边替换成XX*/
5  [chunleimeng]$ echo '1234567' | sed '    /*三个一组加上逗号*/
6  > :start
7  > s/\(.*[0-9]\)\([0-9]\{3\}\)/\1,\2/
8  > t start'
9  1,234,567
```

13 用例示范

Listing 20: Example

```
1  [chunleimeng]$ cat file1
2  The quick brown fox jumps over the lazy dog.
3  The quick brown fox jumps over the lazy dog.
4  The quick brown fox jumps over the lazy dog.
5  The quick brown fox jumps over the lazy dog.
6  [chunleimeng]$ sed '$!G' file1  /*每一行后面加入空行，除了最后一行外*/
7  The quick brown fox jumps over the lazy dog.
8
9  The quick brown fox jumps over the lazy dog.
10
11  The quick brown fox jumps over the lazy dog.
12
13  The quick brown fox jumps over the lazy dog.
14  [chunleimeng]$ sed '=' file1 | sed 'N; s/\n/ /' /*给每一行编号*/
15  1 The quick brown fox jumps over the lazy dog.
16  2 The quick brown fox jumps over the lazy dog.
17  3 The quick brown fox jumps over the lazy dog.
18  4 The quick brown fox jumps over the lazy dog.
19  [chunleimeng]$ cat file10
20  this is line 1
21  this is line 2
22  this is line 3
23  this is line 4
24  this is line 5
25  this is line 6
26  this is line 7
27  this is line 8
28  this is line 9
29  this is line 10
```

```
30  this is line 11
31  this is line 12
32  this is line 13
33  this is line 14
34  this is line 15
35  [chunleimeng]$ sed ':start      /*打印最后6行*/
36  > $q; N; 7,$D                   /*$q是说到最后一行退出*/
37  > b start' file10               /*7,$D是说当前在第6行之后就删掉模式空间的第一行*/
38  this is line 10
39  this is line 11
40  this is line 12
41  this is line 13
42  this is line 14
43  this is line 15
44  [chunleimeng]$ cat file11
45  first line
46
47  second line
48
49
50  third line
51
52  last line
53
54  [chunleimeng]$ sed '/./,/^$/!d' file11 /*删掉连续的空白行*/
55  first line
56
57  second line
58
59  third line
60
61  last line
62
63  [chunleimeng]$ sed '/./,$!d' file11   /*删掉开头的空白行*/
64  first line
65
66  second line
67
68
69  third line
70
71  last line
72
73  [chunleimeng]$ sed '{    /*删掉结尾的空白行*/
74  > :start
75  > /^\n*$/{$d; N; b start}
76  > }' file11
77  first line
78
79  second line
80
81
82  third line
```

```
83
84  last line
```