

16 流编辑器sed基础

流编辑器sed基础

在shell脚本中如何修改文件内容？使用VI编辑器？

一、sed概述

流编辑器 `sed`（`StreamEditor`）是一种**非交互式**的编辑器，也就是说 `sed` 与常见的编辑器（如 `vim`）不同，`sed` **没有交互式的编辑界面、光标移动以及快捷键功能**。

默认情况下，`sed` **并不会改变**原文件本身，而只是对流经 `sed` 命令的文本进行修改，并将修改后的结果打印到标准输出中（屏幕）。

`sed` 处理文本时是以**行为单位**的，一次处理一行内容。处理时，把**当前处理的行**存储在**临时缓冲区**中，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。

`sed` 编辑器是一个很强大的工具，它有如下特点。

- 非交互，基于**正则表达式模式匹配**的**过滤**及**修改**文本。
- **逐行处理**，大文件使用 `sed` 会显得格外有优势。
- 可实现对文本的**输出、删除、替换、复制、剪切、导入、导出**等各种操作
- 脚本化，在**Shell脚本编程中使用** `sed` 比 `vim` 方便。

二、sed语法

2.1 sed语法格式

`sed` 命令语法格式为：`sed` [选项] 命令 文件1 文件2 ...

命令部分包括：[定址表达式] [操作子命令] [参数(标记)]

`sed` 从文件中读取数据，如果**没有输入文件**，则默认**对标准输入数据**进行处理。

⚠注意！ `sed` 和 `grep` 不一样，**无论是否找到指定的模式，退出状态都是0**；只有当命令存在**语法错误**时，退出状态才是**非0**。

2.2 sed选项

`sed` 命令常见的选项如表所示。

命令选项	描述
<code>-n</code>	使用安静模式。在一般 <code>sed</code> 的用法中，所有来自标准输出的内容一般都会被列出到屏幕，但如果加上 <code>-n</code> 选项后，则只有经过 <code>sed</code> 特殊处理的那一行才会被列出来。
<code>-e</code>	允许在该选项后面加一条新的编辑指令。当有多条编辑指令时，应该使用该选项逐一添加，如果编辑指令只有一条，可以不使用该选项。
<code>-i</code>	直接修改读取的文件内容，而不是由屏幕输出。
<code>-f</code>	直接将 <code>sed</code> 的操作写在一个文件内， <code>-f 文件名</code> 则可以执行文件内的 <code>sed</code> 命令。
<code>-r</code>	支持扩展正则表达式。
<code>-h</code>	输出 <code>sed</code> 的帮助信息。

2.2.1 `-n` 选项：不显示默认标准输出信息

`sed` 默认会把模式空间处理完毕后的内容输出到标准输出，也就是输出到屏幕上，加上 `-n` 选项后被设定为安静模式，也就是不会输出默认打印信息，除非子命令中特别指定打印选项，则只会把匹配修改的行进行打印。

▼ 演示-n选项

Shell | 复制代码

```
1 # 注意sed命令部分不能缺失，''表示输出文件内容
2 [root@shell ~]# sed '' /etc/hosts
3 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdoma
  in4
4 ::1         localhost localhost.localdomain localhost6 localhost6.localdoma
  in6
5 # 加-n选项后，只有经过sed选定的行才会显示
6 [root@shell ~]# sed -n '' /etc/hosts
```

2.2.2 `-e` 选项：执行多个操作子命令

如果需要用 `sed` 对文本内容进行多种操作，则需要执行多条子命令来进行操作。

```
1 # 输出/etc/hosts第1行
2 [root@shell ~]# sed -n '1p' /etc/hosts
3 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdom
ain4
4 # 输出/etc/hosts第2行
5 [root@shell ~]# sed -n '2p' /etc/hosts
6 ::1         localhost localhost.localdomain localhost6 localhost6.localdom
ain6
7 # 使用-e选项在一个命令中执行多个操作子命令
8 [root@shell ~]# sed -n -e '1p' -e '2p' /etc/hosts
9 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdom
ain4
10 ::1         localhost localhost.localdomain localhost6 localhost6.localdom
ain6
11 # 使用;分隔操作子命令与-e选项作用完全相同
12 [root@shell ~]# sed -n '1p;2p' /etc/hosts
13 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdom
ain4
14 ::1         localhost localhost.localdomain localhost6 localhost6.localdom
ain6
```

2.2.3 -i 选项：直接修改输入文件

`sed` 默认会把输入行读取到模式空间，简单理解就是一个内存缓冲区，`sed` 操作子命令处理的内容是模式空间中的内容，而非直接处理文件内容。因此在 `sed` 修改模式空间内容之后，并非直接写入修改输入文件，而是打印输出到标准输出。如果需要修改输入文件，那么就可以指定 `-i` 选项。

⚠️ **-i 选项是危险操作，操作前一定要注意！**

▼ 演示-i选项

Shell | 复制代码

```
1 # 创建测试文件
2 [root@shell ~]# echo 'hello world' > test_sed.txt
3 [root@shell ~]# cat test_sed.txt
4 hello world
5 # 将hello替换为A
6 [root@shell ~]# sed 's/hello/A/' test_sed.txt
7 A world
8 [root@shell ~]# cat test_sed.txt
9 hello world
10 # 使用-i修改输入文件
11 [root@shell ~]# sed -i 's/hello/A/' test_sed.txt
12 [root@shell ~]# cat test_sed.txt
13 A world
```

▼ 演示备份文件

Shell | 复制代码

```
1 # 创建测试文件
2 [root@shell ~]# echo 'hello world' > test_sed.txt
3 [root@shell ~]# cat test_sed.txt
4 hello world
5 # 创建备份文件，以免误操作
6 [root@shell ~]# sed -i.bak 's/hello/A/' test_sed.txt
7 [root@shell ~]# cat test_sed.txt
8 A world
9 [root@shell ~]# cat test_sed.txt.bak
10 hello world
```

2.2.4 -f 选项：执行多个操作子命令

`-e` 选项可以执行多个操作子命令操作，用分号分隔多个操作子命令也是可以，如果命令操作比较多的时候就会比较麻烦，这时可以多个子命令操作写入脚本文件，然后使用 `-f` 选项来运行指定该脚本中的命令。

使用 `-f` 选项的 `sed` 命令格式为： `sed [选项] -f 命令脚本文件 文件1 文件2 ...`

2.2.5 -r 选项：支持扩展正则表达式

`sed` 命令的匹配模式支持正则表达式的，默认只能支持基本正则表达式，如果需要支持扩展正则表达式，那么需要添加 `-r` 选项。

以下是 `sed` 支持的元字符。

支持的基本元字符集为： `^, $, ., *, [], [^], \< \>, \(\), \{\}`

支持的扩展元字符集为： `?, +, { }, |, ()`

演示-r选项

Shell | 复制代码

```
1 [root@shell ~]# echo "hello world" | sed -r 's/(hello)|(world)/A/g'
2 A A
```

2.3 sed定址表达式

默认情况下 `sed` 会对**每一行内容**进行匹配、处理、输出，某些情况不需要对处理的文本全部编辑，只需要其中的一部分，比如1-10行，偶数行，或者是包含 `hello` 的行，这种情况下就需要我们去定位特定的行来处理，而不是全部内容，这里把这个定位指定的行叫做**定址**。定制表达式可以是**数字、正则表达式或二者的结合**。

2.3.1 数字定址表达式

数字定址其实就是通过**数字**去指定**具体**要操作编辑的**行**，数字定址有如下几种形式。

格式	意义
<code>n</code>	选择第 <code>n</code> 行。
<code>\$</code>	选择最后 <code>1</code> 行。
<code>m,n</code>	选择 <code>m</code> 到 <code>n</code> 行，共 <code>n+1-m</code> 行。
<code>n,\$</code>	从第 <code>n</code> 行到文件结尾。
<code>m~n</code>	从 <code>m</code> 行开始间隔 <code>n</code> 行选择1行。
<code>m,+n</code>	从第 <code>m</code> 行开始，再选择 <code>n</code> 行。
<code>m,~n</code>	从第 <code>m</code> 行开始，到 <code>n</code> 的倍数行为止。
<code>定址表达式!</code>	取反，表示定址表达式之外的行。

案例：数字定址表达式

```
1 # 显示样例文件内容及行号
2 [root@shell ~]# cat -n /etc/passwd
3     1 root:x:0:0:root:/root:/bin/bash
4     2 bin:x:1:1:bin:/bin:/sbin/nologin
5     3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
6     4 adm:x:3:4:adm:/var/adm:/sbin/nologin
7     5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
8     6 sync:x:5:0:sync:/sbin:/bin/sync
9     7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
10    8 halt:x:7:0:halt:/sbin:/sbin/halt
11    9 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
12   10 operator:x:11:0:operator:/root:/sbin/nologin
13   11 games:x:12:100:games:/usr/games:/sbin/nologin
14   12 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
15   13 nobody:x:99:99:Nobody:/:/sbin/nologin
16   14 systemd-network:x:192:192:systemd Network Management:/:/sbin/nolog
    in
17   15 dbus:x:81:81:System message bus:/:/sbin/nologin
18   16 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
19   17 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
20   18 postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
```

```
1 # 输出样例文件第1行, 1即指定第1行
2 [root@shell ~]# sed -n '1p' /etc/passwd
3 root:x:0:0:root:/root:/bin/bash
4 # 输出样例文件最后1行, $即指定最后1行
5 [root@shell ~]# cat -n /etc/passwd | sed -n '$p'
6     18 postfix:x:89:89::/var/spool/postfix:/sbin/nologin
7 # 输出样例文件第2行至第4行, 2,4即指定第2行至第4行
8 [root@shell ~]# cat -n /etc/passwd | sed -n '2,4p'
9     2 bin:x:1:1:bin:/bin:/sbin/nologin
10    3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
11    4 adm:x:3:4:adm:/var/adm:/sbin/nologin
12 # 输出样例文件第2行至第4行, 2,+2即指定第2行及后面的2行
13 [root@shell ~]# cat -n /etc/passwd | sed -n '2,+2p'
14    2 bin:x:1:1:bin:/bin:/sbin/nologin
15    3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
16    4 adm:x:3:4:adm:/var/adm:/sbin/nologin
17 # 输出样例文件偶数行
18 [root@shell ~]# cat -n /etc/passwd | sed -n '2~2p'
19    2 bin:x:1:1:bin:/bin:/sbin/nologin
20    4 adm:x:3:4:adm:/var/adm:/sbin/nologin
21    6 sync:x:5:0:sync:/sbin:/bin/sync
22    8 halt:x:7:0:halt:/sbin:/sbin/halt
23   10 operator:x:11:0:operator:/root:/sbin/nologin
24   12 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
25   14 systemd-network:x:192:192:systemd Network Management:/:/sbin/nolog
in
26   16 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
27   18 postfix:x:89:89::/var/spool/postfix:/sbin/nologin
28 # 输出样例文件奇数行
29 [root@shell ~]# cat -n /etc/passwd | sed -n '1~2p'
30    1 root:x:0:0:root:/root:/bin/bash
31    3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
32    5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
33    7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
34    9 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
35   11 games:x:12:100:games:/usr/games:/sbin/nologin
36   13 nobody:x:99:99:Nobody:/:/sbin/nologin
37   15 dbus:x:81:81:System message bus:/:/sbin/nologin
38   17 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
39 # 从第1行开始, 每隔3行输出样例文件
40 [root@shell ~]# cat -n /etc/passwd | sed -n '1~3p'
41    1 root:x:0:0:root:/root:/bin/bash
42    4 adm:x:3:4:adm:/var/adm:/sbin/nologin
43    7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
44   10 operator:x:11:0:operator:/root:/sbin/nologin
```

```

45      13 nobody:x:99:99:Nobody:/:/sbin/nologin
46      16 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
47 # 从第2行开始输出样例文件，到3的倍数行为止
48 [root@shell ~]# cat -n /etc/passwd | sed -n '2,~3p'
49      2 bin:x:1:1:bin:/bin:/sbin/nologin
50      3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
51 # 输出样例文件第2行至最后1行之外的行，即第1行
52 [root@shell ~]# cat -n /etc/passwd | sed -n '2,$!p'
53      1 root:x:0:0:root:/root:/bin/bash

```

2.3.2 正则定址表达式

正则定址使用目的和数字定址完全一样，使用方式上有所不同，是通过**正则表达式的匹配**来确定需要处理编辑哪些行，其它行就不需要额外处理。

Shell | 复制代码

```

1 # 输出包含root的行
2 [root@shell ~]# cat -n /etc/passwd | sed -n '/root/p'
3      1 root:x:0:0:root:/root:/bin/bash
4      10 operator:x:11:0:operator:/root:/sbin/nologin
5 # 输出root开头的行
6 [root@shell ~]# cat /etc/passwd | sed -n '/^root/p'
7 root:x:0:0:root:/root:/bin/bash

```

2.3.3 数字定址和正则定址混用

数字定址和正则定址可以配合使用。


```

1  # 显示样例文件内容及行号
2  [root@shell ~]# cat -n /etc/passwd
3      1 root:x:0:0:root:/root:/bin/bash
4      2 bin:x:1:1:bin:/bin:/sbin/nologin
5      3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
6      4 adm:x:3:4:adm:/var/adm:/sbin/nologin
7      5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
8      6 sync:x:5:0:sync:/sbin:/bin/sync
9      7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
10     8 halt:x:7:0:halt:/sbin:/sbin/halt
11     9 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
12    10 operator:x:11:0:operator:/root:/sbin/nologin
13    11 games:x:12:100:games:/usr/games:/sbin/nologin
14    12 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
15    13 nobody:x:99:99:Nobody:/:/sbin/nologin
16    14 systemd-network:x:192:192:systemd Network Management:/:/sbin/nolog
    in
17    15 dbus:x:81:81:System message bus:/:/sbin/nologin
18    16 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
19    17 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
20    18 postfix:x:89:89::/var/spool/postfix:/sbin/nologin
21  [root@shell ~]# cat /etc/passwd | sed -n '/^root/p'
22  root:x:0:0:root:/root:/bin/bash
23  [root@shell ~]# cat /etc/passwd | sed -n '/^root/,2p'
24  root:x:0:0:root:/root:/bin/bash
25  bin:x:1:1:bin:/bin:/sbin/nologin
26  [root@shell ~]# cat /etc/passwd | sed -n '/^root/,+2p'
27  root:x:0:0:root:/root:/bin/bash
28  bin:x:1:1:bin:/bin:/sbin/nologin
29  daemon:x:2:2:daemon:/sbin:/sbin/nologin

```

2.4 sed操作子命令

操作子命令指示 `sed` 对指定行进行何种操作，包括打印、删除、修改等。

命令	描述
<code>p</code>	打印 输出 被选中的行。
<code>d</code>	删除 被选中的行。
<code>a 字符串</code>	将 字符串 作为单独一行 追加 到被选中的 行之后 。

c 字符串	将选中的行 替换 为字符串。
i 字符串	在被选中的 行之前插入 字符串。
s/字符串1/字符串2/标志	查找替换，通常 s 可以搭配 正则表达式 ，将 字符串1替换为字符串2 。
r 文件名	在选择的行之后 追加文件的内容 。
w 文件名	将选择的行内容 写入文件 。
=	打印当前 行号 。

2.4.1 子命令d：删除行

子命令 **d** 表示删除指定的行内容。

```
1 # 删除所有行
2 [root@shell ~]# cat -n /etc/passwd | sed 'd'
3 # 删除第1行
4 [root@shell ~]# cat -n /etc/passwd | sed '1d'
5     2 bin:x:1:1:bin:/bin:/sbin/nologin
6     3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
7     4 adm:x:3:4:adm:/var/adm:/sbin/nologin
8     5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
9     6 sync:x:5:0:sync:/sbin:/bin/sync
10    7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
11    8 halt:x:7:0:halt:/sbin:/sbin/halt
12    9 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
13   10 operator:x:11:0:operator:/root:/sbin/nologin
14   11 games:x:12:100:games:/usr/games:/sbin/nologin
15   12 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
16   13 nobody:x:99:99:Nobody:/:/sbin/nologin
17   14 systemd-network:x:192:192:systemd Network Management:/:/sbin/nolog
    in
18   15 dbus:x:81:81:System message bus:/:/sbin/nologin
19   16 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
20   17 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
21   18 postfix:x:89:89:/:var/spool/postfix:/sbin/nologin
22 # 删除第2行至最后1行
23 [root@shell ~]# cat -n /etc/passwd | sed '2,$d'
24     1 root:x:0:0:root:/root:/bin/bash
```

2.4.2 子命令a: 追加内容

子命令 **a** 表示在指定行**下边**插入指定行的内容。

```

1  # 在/etc/hosts每行之下都插入1行, 内容为A
2  [root@shell ~]# cat /etc/hosts | sed 'a A'
3  127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdom
ain4
4  A
5  :::1        localhost localhost.localdomain localhost6 localhost6.localdom
ain6
6  A
7  # 在/etc/hosts第1行之下插入1行, 内容为A
8  [root@shell ~]# cat /etc/hosts | sed '1a A'
9  127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdom
ain4
10 A
11 :::1        localhost localhost.localdomain localhost6 localhost6.localdom
ain6
12 # 在/etc/hosts第1行之下插入多行, 内容用\n换行
13 [root@shell ~]# cat /etc/hosts | sed '1a A\nB\nC'
14 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdom
ain4
15 A
16 B
17 C
18 :::1        localhost localhost.localdomain localhost6 localhost6.localdom
ain6

```

2.4.3 子命令i: 插入内容

子命令 **i** 和 **a** 功能类似, 只不过是在指定行**上边**插入指定行的内容。

```

1  # 在/etc/hosts第1行之上插入1行, 内容为A
2  [root@shell ~]# cat /etc/hosts | sed '1i A'
3  A
4  127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdoma
in4
5  :::1        localhost localhost.localdomain localhost6 localhost6.localdoma
in6

```

2.4.4 子命令c: 内容替换

子命令 **c** 是表示把指定的行内容**替换**为自己需要的行内容。

```
1 # 将/etc/hosts中每行的内容都替换为A
2 [root@shell ~]# cat /etc/hosts | sed 'c A'
3 A
4 A
5 # 将/etc/hosts中第一行的内容都替换为A
6 [root@shell ~]# cat /etc/hosts | sed '1c A'
7 A
8 ::1          localhost localhost.localdomain localhost6 localhost6.localdom
ain6
9 # 将/etc/hosts中1-2行的内容都替换为A, 注意1-2行2行变1行
10 [root@shell ~]# cat /etc/hosts | sed '1,2c A'
11 A
```

2.4.5 子命令s：替换内容

子命令 **s** 为替换子命令，是 **sed** 使用频率最高的子命令，没有之一。因为**支持正则表达式**，所以功能变得强大无比。

基本语法： **s/模式/替换字符串/标志**

/ 可以换成其它的符号。

子命令 **s** 支持的标志如下。

- **g**：全局更改。
- **n**：可以是1-512，表示第n次出现的情况进行替换。
- **w file**：写入到一个文件 **file** 中。

```

1 # 将每行第1个localhost替换为controller
2 [root@shell ~]# cat /etc/hosts | sed 's/localhost/controller/'
3 127.0.0.1 controller localhost.localdomain localhost4 localhost4.localdo
  main4
4 ::1 controller localhost.localdomain localhost6 localhost6.localdo
  main6
5 # 将localhost替换为controller
6 [root@shell ~]# cat /etc/hosts | sed 's/localhost/controller/g'
7 127.0.0.1 controller controller.localdomain controller4 controller4.loca
  ldomain4
8 ::1 controller controller.localdomain controller6 controller6.loca
  ldomain6
9 # 将127.0.0.1替换为controller
10 [root@shell ~]# cat /etc/hosts | sed 's/127.0.0.1/controller/'
11 controller localhost localhost.localdomain localhost4 localhost4.localdo
  main4
12 ::1 localhost localhost.localdomain localhost6 localhost6.localdom
  ain6
13 # 将每行第1个匹配的模式替换为A
14 [root@shell ~]# cat /etc/hosts | sed -r 's/[a-z]+[0-9]+*/A/'
15 127.0.0.1 A localhost.localdomain localhost4 localhost4.localdomain4
16 ::1 A localhost.localdomain localhost6 localhost6.localdomain6
17 # 将行中所有匹配的模式都替换为A, , 默认不加g标记时只替换每行的第一个字符。
18 [root@shell ~]# cat /etc/hosts | sed -r 's/[a-z]+[0-9]+*/A/g'
19 127.0.0.1 A A.A A A.A
20 ::1 A A.A A A.A
21 # 将每行第2个匹配的模式替换为A
22 [root@shell ~]# cat /etc/hosts | sed -r 's/[a-z]+[0-9]+*/A/2'
23 127.0.0.1 localhost A.localdomain localhost4 localhost4.localdomain4
24 ::1 localhost A.localdomain localhost6 localhost6.localdomain6
25 # 将修改后的内容保存到sed_result.txt中
26 [root@shell ~]# cat /etc/hosts | sed -r 's/[a-z]+[0-9]+*/A/w sed_result.t
  xt'
27 127.0.0.1 A localhost.localdomain localhost4 localhost4.localdomain4
28 ::1 A localhost.localdomain localhost6 localhost6.localdomain6
29 [root@shell ~]# cat sed_result.txt
30 127.0.0.1 A localhost.localdomain localhost4 localhost4.localdomain4
31 ::1 A localhost.localdomain localhost6 localhost6.localdomain6
32 # 在文件中每行的首尾分别加上111、222
33 # 注意: &表示原内容
34 [root@shell ~]# cat /etc/hosts | sed -r 's/.*/111&222/'
35 111127.0.0.1 localhost localhost.localdomain localhost4 localhost4.local
  domain4222
36 111::1 localhost localhost.localdomain localhost6 localhost6.local
  domain6222

```

```

37 # 将分隔符改为#, 方便处理路径
38 [root@shell ~]# cat /etc/passwd | sed 's#/sbin/nologin#/bin/login/#'
39 root:x:0:0:root:/root:/bin/bash
40 bin:x:1:1:bin:/bin:/bin/login/
41 daemon:x:2:2:daemon:/sbin:/bin/login/
42 adm:x:3:4:adm:/var/adm:/bin/login/
43 lp:x:4:7:lp:/var/spool/lpd:/bin/login/
44 sync:x:5:0:sync:/sbin:/bin/sync
45 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
46 halt:x:7:0:halt:/sbin:/sbin/halt
47 mail:x:8:12:mail:/var/spool/mail:/bin/login/
48 operator:x:11:0:operator:/root:/bin/login/
49 games:x:12:100:games:/usr/games:/bin/login/
50 ftp:x:14:50:FTP User:/var/ftp:/bin/login/
51 nobody:x:99:99:Nobody:./bin/login/
52 systemd-network:x:192:192:systemd Network Management:./bin/login/
53 dbus:x:81:81:System message bus:./bin/login/
54 polkitd:x:999:998:User for polkitd:./bin/login/
55 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/bin/login/
56 postfix:x:89:89:./var/spool/postfix:/bin/login/
57 [root@shell ~]# cat /etc/passwd
58 root:x:0:0:root:/root:/bin/bash
59 bin:x:1:1:bin:/bin:/sbin/nologin
60 daemon:x:2:2:daemon:/sbin:/sbin/nologin
61 adm:x:3:4:adm:/var/adm:/sbin/nologin
62 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
63 sync:x:5:0:sync:/sbin:/bin/sync
64 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
65 halt:x:7:0:halt:/sbin:/sbin/halt
66 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
67 operator:x:11:0:operator:/root:/sbin/nologin
68 games:x:12:100:games:/usr/games:/sbin/nologin
69 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
70 nobody:x:99:99:Nobody:./sbin/nologin
71 systemd-network:x:192:192:systemd Network Management:./sbin/nologin
72 dbus:x:81:81:System message bus:./sbin/nologin
73 polkitd:x:999:998:User for polkitd:./sbin/nologin
74 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
75 postfix:x:89:89:./var/spool/postfix:/sbin/nologin

```

2.4.6 子命令r: 读入文件

子命令 **r** 类似于 **a**，也是将内容追加到指定行的后边，只不过 **r** 是将指定文件内容读取并追加到指定行下边。

```

1 # 在/etc/passwd第1行后插入/etc/hosts所有内容
2 [root@shell ~]# cat /etc/hosts
3 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdom
ain4
4 ::1         localhost localhost.localdomain localhost6 localhost6.localdom
ain6
5 [root@shell ~]# cat -n /etc/passwd | sed '1r /etc/hosts'
6     1 root:x:0:0:root:/root:/bin/bash
7 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdom
ain4
8 ::1         localhost localhost.localdomain localhost6 localhost6.localdom
ain6
9     2 bin:x:1:1:bin:/bin:/sbin/nologin
10    3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
11    4 adm:x:3:4:adm:/var/adm:/sbin/nologin
12    5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
13    6 sync:x:5:0:sync:/sbin:/bin/sync
14    7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
15    8 halt:x:7:0:halt:/sbin:/sbin/halt
16    9 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
17   10 operator:x:11:0:operator:/root:/sbin/nologin
18   11 games:x:12:100:games:/usr/games:/sbin/nologin
19   12 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
20   13 nobody:x:99:99:Nobody:/:/sbin/nologin
21   14 systemd-network:x:192:192:systemd Network Management:/:/sbin/nolog
in
22   15 dbus:x:81:81:System message bus:/:/sbin/nologin
23   16 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
24   17 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
25   18 postfix:x:89:89:/:var/spool/postfix:/sbin/nologin

```

2.4.7 子命令w：写文件

`sed` 默认并不改写原文件，而只是对缓冲区的文本做了修改并输出到屏幕。所以想保存文件可以使用 `w` 命令将结果保存到外部指定文件。

```

1 #将/etc/passwd的第1行内容写入sed_output.txt
2 [root@shell ~]# cat -n /etc/passwd | sed -n '1w sed_output.txt'
3 [root@shell ~]# cat sed_output.txt
4     1 root:x:0:0:root:/root:/bin/bash

```


除了 `w` 命令之外还可以用**重定向**保存文件。

▼ Shell 复制代码

```
1 [root@shell ~]# cat -n /etc/passwd | sed -n '1p' > sed_redirect.txt
2 [root@shell ~]# cat sed_redirect.txt
3      1 root:x:0:0:root:/root:/bin/bash
```

2.4.8 子命令=: 打印行号

子命令 `=`，可以将行号打印出来。

▼ Shell 复制代码

```
1  # 输出第2至最后1行的行号
2 [root@shell ~]# cat /etc/passwd | sed -n '2,$='
3 2
4 3
5 4
6 5
7 6
8 7
9 8
10 9
11 10
12 11
13 12
14 13
15 14
16 15
17 16
18 17
19 18
```

小结

- `sed` 语法格式：`sed [选项] [定址表达式] 操作子命令 [输入文件]`
- `sed` 选项
- `sed` 定址表达式：数字、正则表达式
- `sed` 操作子命令

课程目标

- 知识目标：熟练掌握 `sed` 命令的基本语法。
- 技能目标：能够利用 `sed` 命令完成实际需求。

课外拓展

- 进一步了解 `sed` 命令的应用场景。

参考资料

- `sed --help` 或 `man sed`
- 《Linux Shell核心编程指南》，丁明一，电子工业出版社