

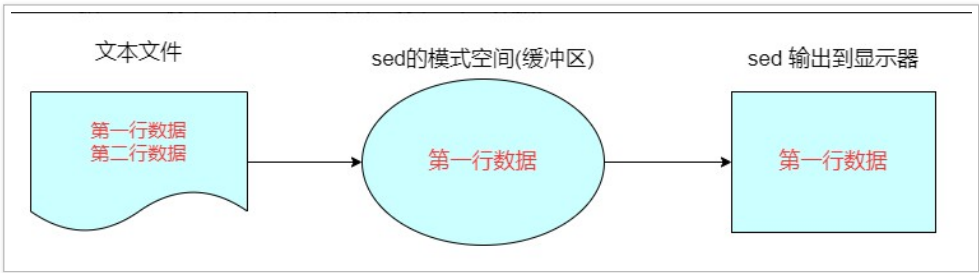
## 2.3 sed 命令详解\_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 2.3 sed 命令详解涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

- 简介

我们之前的学习中, Vim 采用的是交互式文本编辑模式，你可以用键盘命令来交互性地插入、删除或替换数据中的文本。但本节要讲的 sed 命令不同，它采用的是流编辑模式，最明显的特点是，在 sed 处理数据之前，需要预先提供一组规则，sed 会按照此规则来编辑数据。
- 使用场景
  - 超大文件处理；
  - 对文件进行批量增加，替换等。
  - 有规律的文本，例如 以分号，空格等分隔的日志文件等；
- 说明
  - sed 会根据脚本命令来处理文本文件中的数据，这些命令要么从命令行中输入，要么存储在一个文本文件中，此命令执行数据的顺序如下：
  - - 每次仅读取一行内容；
    - 根据提供的规则命令匹配并修改数据。注意，sed 默认不会直接修改源文件数据，而是会将数据复制到缓冲区中，修改也仅限于缓冲区中的数据；
    - 将执行结果输出。

当一行数据匹配完成后，它会继续读取下一行数据，并重复这个过程，直到将文件中所有数据处理完毕。



格式：

```
sed [options] '{command}[flags]' [filename]
```

- e 脚本命令 该选项会将其后面的脚本命令添加到已有的命令中。
- f 脚本文件 该选项会将其文件中的脚本命令添加到已有的命令中。
- n 只显示匹配的行
- i 直接对原文件进行操作，会修改原文件内容。sed命令默认不修改文件

- {command}[flags]
  - sed 内部常用命令
    - i : insert, 在指定匹配到的行前面添加新行内容为 string
    - a : append, 在指定或匹配到的行后面追加新行，内容为 string
    - d : delete, 删除符合地址界定条件的行
    - p : print, 默认 sed 对模式空间内的处理完毕后，将输出的结果输出在标准输出，

添加 `p` 命令，相当于输出了原文，又一次输出了模式匹配处理后的内容。

`s` : 查找并替换，默认只替换每行中第一次被模式匹配到的字符串，如果修饰符为 `g`，则为全部替换。

- flags

- `n` : 1~512之间的数字 表示指定要替换的字符串出现第几次时才进行替换。  
例如，一行中有 3 个 A，但用户只想替换第二个 A，这是就用这个标记；
- `g` : 对数据中所有匹配到的内容进行替换，如果没有 `g`，则只会在第一次匹配成功时做替换操作。  
例如，一行数据中有 3 个 A，则只会替换第一个 A；
- `p` : 会打印与替换命令中指定的模式匹配的行。此标记通常与 `-n` 选项一起使用。

- 场景 1: 显示 `/etc/passwd` 第三行的信息

```
[linux@linux]# sed -n '3p' /etc/passwd
bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

注：这里的 3 代表行数，p 代表输出结果

- 场景 2: 在 `/etc/passwd` 中第一行前添加一行内容为"Good Good Study", 例如:

```
[linux@linux]# sed -e '1iGood Good Study' /etc/passwd
Good Good Study
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
....
```

注：这里的 1 代表第一行，i 代表 insert 插入的意思。表示在第 1 行前插入新的字符串。

- 场景 3: 把 `/etc/passwd` 中所有名字为 root 的字符串改为 class

格式：

```
sed 's/旧字符串/新字符串/g' 文件名
功能：替换所有的字符串
```

```
sed 's/旧字符串/新字符串/数字' 文件名
功能：替换每行中第二次出现该字符串的数据
```

示例代码:

```
[linux@linux]# sed -e 's/root/class/g' /etc/passwd
class:x:0:0:class:/class:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
...
```

```
[linux@linux]# sed 's/root/class/2' /etc/passwd
root:x:0:0:class:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

```
[linux@linux]# sed -n 's/root/class/2p' /etc/passwd
class:x:0:0:class:/class:/bin/bash
```

注：`-n` 选项会禁止 sed 输出，但 `p` 标记会输出修改过的行，将二者匹配使用的效果就是只输出被替换命令修改过的行

- 场景 4: 删除 `/etc/passwd` 中内容并列出行号，并且将第 2~5 行删除！

```
[linux@linux]# cat -n /etc/passwd | sed '2,5d'

1      root:x:0:0:root:/root:/bin/bash
3      bin:x:2:2:bin:/bin:/usr/sbin/nologin
4      sys:x:3:3:sys:/dev:/usr/sbin/nologin
5      sync:x:4:65534:sync:/bin:/bin/sync
6      games:x:5:60:games:/usr/games:/usr/sbin/nologin
...
```

注：cat -n 会显示行号，这里利用管道来显示内容。

- 场景 5：把 / etc/passwd 文件中 root 用户的信息以带行号的形式重定向到 log.txt 文件中，要求把 log.txt。文件中 root 替换为 linux. 并且 log.txt 中保存替换后的文件。

```
[linux@linux]# cat /etc/passwd | grep -n "root" > log.txt
[linux@linux]# sed -i 's/root/linux/g' log.txt
[linux@linux]#cat log.txt
1:linux:x:0:0:linux:/linux:/bin/bash
```

```
HELLO LINUX!
Linux is a free unix-type operating system.
This is a linux testfile!
Linux test
newLine
Google
Taobao
Runoob
Tesetfile
Wiki
```

#### 1. 要求分别实习以下功能

1. 把 2-6 行信息删除，输出到屏幕上
2. 在第 2 行后添加 "drink tea" 字符串，并输出到屏幕上
3. 把 Taobao 替换为 muke

---

全文完

---

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

