

# 一、使用 sed 修改配置

## P1 sed 编辑指令

常用 sed 指令：

p(print):打印行  
d(delete):删除行  
c(replace):替换行  
s(substitution):替换关键词  
=: 打印行号

过滤数据（print 指令）

```
[root@svr7 day04]# sed -n "/IPADDR/p" /etc/sysconfig/network-scripts/ifcfg-eth0 #过滤网卡 IP 地址
[root@svr7 day04]# free | sed -n "/Mem/p" #过滤内存信息
[root@svr7 day04]# df -h | sed -n "/\$/p" #过滤磁盘根分区信息
[root@svr7 day04]# sed -n "1p;3p;6p" /etc/passwd #打印不连续的多行
[root@svr7 day04]# sed -n '2!p' /etc/passwd #不打印第二行，其他都打印
```

删除数据（delete 指令，不适用-i 选项，源文件不会被修改）

```
[root@svr7 day04]# sed 'd' /etc/hosts #删除/etc/hosts 全文，没有定位条件匹配所有行
[root@svr7 day04]# sed '1d' /etc/hosts #删除第一行
[root@svr7 day04]# cat /etc/fstab > /tmp/fstab
[root@svr7 day04]# sed '1,3d' /tmp/fstab #删除 1 到 3 行
[root@svr7 day04]# sed '/dev/!d' /tmp/fstab #不包含 dev 的行都删除
[root@svr7 day04]# sed '/^#/d' /tmp/fstab #删除以#开头的行
[root@svr7 day04]# sed '/^$/d' /tmp/fstab #删除空行
```

使用 i 进行永久删除

```
[root@svr7 day04]# sed -i '1,4d' /tmp/fstab #删除 1-4 行
```

替换行（c 以行为单位）

```
[root@svr7 day04]# sed 'c 123456' /tmp/fstab #将所有行替换为 123456
[root@svr7 day04]# sed '/IPADDR/c IPADDR=1.1.1.1' /etc/sysconfig/network-scripts/ifcfg-eth0
#查找 IPADDR 的行替换为 IPADDR=1.1.1.1（替换 IP 地址）
[root@svr7 day04]# sed '/127/c 127.0.0.1 localhost' /etc/hosts #查找 127 的行替换为 127.0.0.1 localhost
[root@svr7 day04]# sed '4c XXXX' /etc/shells #将/etc/shells 文件的第四行替换为 XXXX
```

替换关键词（格式：s/旧的/新的/）

```
[root@svr7 day04]# vim test.txt
2046 2048 2046 2046
1001 2046 2999 1888
2046 2046 2046 2046
```

```
[root@svr7 day04]# sed 's/2046/XXXX/' test.txt      #将每一行的第一个 2046 替换为 XXXX
[root@svr7 day04]# sed 's/2046/XXXX/g' test.txt      #将每一行的所有 2046 替换为 XXXX
[root@svr7 day04]# sed 's/2046/XXXX/2' test.txt      #将每一行的第二个 2046 替换为 XXXX
[root@svr7 day04]# sed 's/2046/(&)/' test.txt        #将每一行的第一个 2046 替换为(2046)
(&指替换的旧的内容,在这里指 2046)
[root@svr7 day04]# sed '2s/2046/XXXX/g' test.txt     #将替换第二行的所有 2046 替换为 XXXX
[root@svr7 day04]# sed '2s/2046//g' test.txt         #替换为空, 等同于删除
[root@svr5 ~]# sed -n '2s/2046/XXXX/p' test.txt      #n 屏蔽输出, p 打印显示。将第二行的
2046 替换为 XXXX
```

将替换符改为其他字符, 也可以达到同样的操作

```
[root@svr5 ~]# sed 's#2046#XXXX#g' test.txt        #以#为 sed 的指令替换符
[root@svr5 ~]# sed 's,2046,XXXX,g' test.txt        #以,为 sed 的指令替换符
正则符号 ( ) 具有保留功能
```

echo "hello the world" | sed -r 's/^(.)(.)(.)\*\$/\3\2\1/'

```
[root@svr7 day04]# echo "hello the world" | sed -r 's/^(.)(.)(.)*$/\3\2\1/'      #扩展正则需
要加 r, .代表任意单个字符 \3\2\1 代表第三个第二个第一个内容, \数字格式固定
```

打印行号 =

```
[root@svr7 day04]# sed -n '1=' /etc/passwd        #打印第一行的行号
[root@svr7 day04]# sed -n '/root/= ' /etc/passwd   #打印包含 root 的行
[root@svr7 day04]# sed -n '/bash$/=' /etc/passwd  #打印以 bash 结尾的行
[root@svr7 day04]# sed -n '$=' /etc/passwd         #统计行数
[root@svr7 day04]# wc -l /etc/passwd               #统计行数 (会显示文件名)
```

## 二、sed 多行文本处理

### P1 文本块指令

常用指令

i (insert) : 插入 (-i 选项才会修改源文件)  
a (append) : 追加  
r (read) : 读取文件 | 导入文件内容  
w (write) : 文件另存为 | 导出文件内容

插入指令: i (insert) 行前写入

```
[root@svr7 day04]# sed '2i ABC_XYZ' test.txt      #在第二行前插入 ABC_XYZ
[root@svr7 day04]# sed '3i ABC_XYZ' test.txt      #在第三行前 ABC_XYZ
[root@svr7 day04]# sed '/2046/i ABC\nXYZ' test.txt #在含有 2046 的行前面插入两行
ABC 和 XYZ (\n 换行回车)
```

```
[root@svr7 day04]# sed '/1888/i ABC\nXYZ' test.txt
```

#在含有 1888 的行前面插入两行 ABC 和 XYZ

追加指令: a (append)行后写入

```
[root@svr7 day04]# sed '2a ABC_XYZ' test.txt
```

#在第二行后插入 ABC\_XYZ

```
[root@svr7 day04]# sed '3a ABC_XYZ' test.txt
```

#在第三行后 ABC\_XYZ

```
[root@localhost ~]# sed '/2046/a ABC\nXYZ' test.txt
```

#在含有 2046 的行后面插入两行 ABC 和 XYZ (\n 换行回车)

```
[root@localhost ~]# sed '/1888/a ABC\nXYZ' test.txt
```

#在含有 1888 的行后面插入两行 ABC 和 XYZ

导入指令: r (read, 将其他文件的内容导入)

```
[root@svr7 day04]# sed '2r /etc/hosts' test.txt
```

#将/etc/hosts 导入到 test.txt 的第二行后面

```
[root@svr7 day04]# sed 'r /etc/hosts' test.txt
```

#将/etc/hosts 导入到 test.txt 的每一行后面

```
[root@svr7 day04]# sed '/1888/r /etc/hosts' test.txt
```

#将/etc/hosts 导入到 test.txt 的含有 1888 的行后面

导出指令: w (write, 将文件内容导出另存到其他文件)

```
[root@svr7 day04]# sed 'w copy_test.txt' test.txt
```

#将 test.txt 文件的所有内容另存为一个新文件 copy\_test.txt

```
[root@svr7 day04]# sed '/1888/w 1888.txt' test.txt
```

#将 test.txt 文件中所有包含 1888 的行另存为新文件 1888.txt

```
[root@svr7 day04]# sed '2,3w line.txt' test.txt
```

#将 test.txt 文件的 2 到 3 行另存为新文件 line.txt

### 三、综合案例

#### P1 sed 编辑指令

随机点名的脚本

#功能描述(Description):随机点名抽奖器. #按 Ctrl+C 结束脚本.

#定义变量:人员列表文件名,从变量里面统计文件的行数 #随机抽取一个人名(随机点名).

```
[root@svr5 ~]# vim /root/shell/day04/name.txt
```

NSDVN2005 班学员名字

```
[root@svr5 ~]# vim /shell/day04/roll.sh
```

```
#!/bin/bash
```

```
name_file="name.txt"
```

```
line_file=$(sed -n '$=' $name_file)
```

```
while :
```

```
do
```

```
clear
```

```
tmp=$(sed -n "${RANDOM%line_file+1}p" $name_file)
```

```
echo -e "\033[32m 随机点名器(按 Ctrl+C 停止): \033[0m"
```

```

echo -e "\033[32m#####\033[0m"
echo -e "\033[32m                                #\033[0m"
echo -e "\033[30m                $tmp                \033[0m"
echo -e "\033[32m                                #\033[0m"
echo -e "\033[32m#####\033[0m"
sleep 0.1
done
[root@svr7 day04]# chmod +x roll.sh
[root@svr7 day04]# ./roll.sh

```

抓取 tmooc 所有图片（此虚拟机需要联网才可以）

#功能描述(Description)编写脚本抓取单个网页中的图片数据.

#将网页源代码保存到文件中.

#对文件进行过滤和清洗,获取需要的种子 URL 链接.

#检测系统如果没有 wget 下载工具则安装该软件.

#wget 为下载工具,其参数选项描述如下:

# -P 指定将数据下载到特定目录(prefix).

# -q 不显示下载过程(quiet).

```

[root@svr5 ~]# vim /root/shell/day04/tmooc.sh
#!/bin/bash
URL="/tmp/spider_$$txt"
curl -s http://www.tmooc.cn/ > $URL
echo -e "\033[32m 正在获取种子 URL,请稍后...\033[0m"
sed -i '/<img/!d' $URL          #删除不包含/dev/null;
then
    yum -y install wget
fi
echo -e "\033[32m 正在批量下载种子数据,请稍后...\033[0m"
for i in $(cat $URL)
do
    wget -P /tmp/ -q $i
done
#删除临时种子列表文件.
rm -rf $URL
[root@svr7 day04]# chmod +x tmooc.sh
[root@svr7 day04]# ./tmooc.sh

```