NSD SHELL DAY05

1 案例1：sed基本用法

1.1 问题

本案例要求熟悉sed命令的p、d、s等常见操作，并结合正则表达式，完成以下任务：

- 删除文件中每行的第二个、最后一个字符
- 将文件中每行的第一个、第二个字符互换
- 删除文件中所有的数字、行首的空格
- 为文件中每个大写字母添加括号

1.2 方案

sed文本处理工具的用法：

```
用法1：前置命令 | sed [选项] '编辑指令'
用法2：sed [选项] '编辑指令' 文件...
```

相关说明如下：

- “编辑指令”可以为增删改查等指令
- “定址符”用来定义需要操作的文本，由“[地址1 [,地址2]]组成
- 未指定“定址符”时，默认处理所有文本

Top

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：认识sed工具的基本选项

1）sed命令的 -n 选项

执行p打印等过滤操作时，希望看到的是符合条件的文本。但不使用任何选项时，默认会将原始文本一并输出，从而干扰过滤效果。比如，尝试用sed输出/etc/rc.local的第1行：

```
[root@svr5 ~]# sed '1p' /etc/rc.local
#! /bin/sh
#! /bin/sh
#
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.
```

可以发现所有的行都被显示出来了。—— 正确的用法应该添加 -n 选项，这样就可以只显示第1行了：

```
[root@svr5 ~]# sed -n '1p' /etc/rc.local
#! /bin/s
```

而在执行d删除等过滤操作时，希望看到的是删除符合条件的文本之后还能够被保留下来的文本，所以这时候就不应该使用 -n 选项了。比如，删除/etc/rc.local文件的第1-4行文本：

Top

```
[root@svr5 ~]# sed '1,4d' /etc/rc.local
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
```

2）sed命令的 -i 选项

正常情况下，sed命令所做的处理只是把操作结果（包括打印、删除等）输出到当前终端屏幕，而并不会对原始文件做任何更改：

```
[root@svr5 ~]# sed '1,4d' rclocal.txt          //删除第1~4行，输出结果
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
[root@svr5 ~]# cat rclocal.txt              //查看原始文本，并未改动
```

若希望直接修改文件内容，应添加选项 -i。

比如，直接删除rcloal.txt文件的第1~4行，不输出结果：

```
[root@svr5 ~]# sed -i '1,4d' rclocal.txt        //删除操作
[root@svr5 ~]# cat rclocal.txt                //确认删除结果
```

下文中关于使用sed修改文件的示例中，为了避免大家在练习过程中因误操作导致系统故障，部分命令省略 –i 选项，不再逐一说明。需要时，大家可自行加上此选项。

Top

3）多个指令可以使用分号隔离

用分号来隔离多个操作（如果有定址条件，则应该使用{ }括起来），比如：

```
[root@svr5 ~]# sed -n '1p;4p' /etc/rc.local
#! /bin/sh
# You can put your own initialization stuff in here if you don't
```

或者：

```
[root@svr5 ~]# sed -n '{1p;4p}' /etc/rc.local
#! /bin/sh
# You can put your own initialization stuff in here if you don't
```

步骤二：认识sed工具的p输出操作

先创建一个练习用的测试文件，每一行之前添加行号，方便练习时查看效果：

```
[root@svr5 ~]# cat -n /etc/rc.local > rclocal.txt
[root@svr5 ~]# cat rclocal.txt
     1  #! /bin/sh
     2  #
     3  # This script will be executed *after* all the other init scripts.
     4  # You can put your own initialization stuff in here if you don't
     5  # want to do the full Sys V style init stuff.
     6
     7  touch /var/lock/subsys/local
```

Top

1）输出所有行，相当于cat命令。

```
[root@svr5 ~]# sed -n 'p' rclocal.txt
#! /bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
```

2）输出第4行。

```
[root@svr5 ~]# sed -n '4p' rclocal.txt
# You can put your own initialization stuff in here if you don't
```

3）输出第4~7行。

```
[root@svr5 ~]# sed -n '4,7p' rclocal.txt
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
```

Top

4）输出第4行和第7行。

```
[root@svr5 ~]# sed -n '4p;7p' rclocal.txt
4 # You can put your own initialization stuff in here if you don't
7 touch /var/lock/subsys/local
```

5)输出第2行及之后的3行。

```
[root@svr5 ~]# sed -n '2,+3p' rclocal.txt
2 #
3 # This script will be executed *after* all the other init scripts.
4 # You can put your own initialization stuff in here if you don't
5 # want to do the full Sys V style init stuff.
```

6)输出以local结尾的行。

```
[root@svr5 ~]# sed -n '/local$/p' rclocal.txt
7 touch /var/lock/subsys/local
```

7)输出奇数行。

```
[root@svr5 ~]# sed -n 'p;n' rclocal.txt
1 #! /bin/sh
3 # This script will be executed *after* all the other init scripts.
5 # want to do the full Sys V style init stuff.
7 touch /var/lock/subsys/local
```

Top

8)输出偶数行。

```
[root@svr5 ~]# sed -n 'n;p' rclocal.txt
    2 #
    4 # You can put your own initialization stuff in here if you don't
    6
```

9）从第5行输出到最后一行。

```
[root@svr5 ~]# sed -n '5,$p' rclocal.txt
    5 # want to do the full Sys V style init stuff.
    6
    7 touch /var/lock/subsys/local
```

10）输出文本的行数。

```
[root@svr5 ~]# sed -n '$=' rclocal.txt
7
```

步骤三：认识sed工具的d输出操作

还以rclocal.txt文件为例，文件内容如下所示：

Top

```
[root@svr5 ~]# cat rclocal.txt
#! /bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
```

1）删除第3~5行文本

```
[root@svr5 ~]# sed '3,5d' rclocal.txt
#! /bin/sh
#

touch /var/lock/subsys/local
```

2）删除所有包含“init”的行

```
[root@svr5 ~]# sed '/init/d' rclocal.txt
#! /bin/sh
#

touch /var/lock/subsys/local
```

Top

3）删除所有包含“init”的行、所有包含“bin”的行

```
[root@svr5 ~]# sed '/init/d;/bin/d' rclocal.txt
2 #
6
7 touch /var/lock/subsys/local
```

4）删除不包括“init”的行

```
[root@svr5 ~]# sed '/init/!d' rclocal.txt
3 # This script will be executed *after* all the other init scripts.
4 # You can put your own initialization stuff in here if you don't
5 # want to do the full Sys V style init stuff.
```

这个实际效果相当于只显示包含“init”的行：

```
[root@svr5 ~]# sed -n '/init/p' rclocal.txt
3 # This script will be executed *after* all the other init scripts.
4 # You can put your own initialization stuff in here if you don't
5 # want to do the full Sys V style init stuff.
```

5）删除文件的最后一行

Top

```
[root@svr5 ~]# sed '$d' rclocal.txt
#! /bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

```

6）删除文件中的空行

手动新建一个测试文件：

```
[root@svr5 ~]# vim blankline.txt
abc

def
hijklmn



hello world
I am here

end
```

Top

删除所有空行：

```
[root@svr5 ~]# sed '/^$/d' blankline.txt
abc
def
hijklmn
hello world
I am here
end
```

步骤四：认识sed工具的s替换操作

还以rclocal.txt文件为例，文件内容如下所示：

```
[root@svr5 ~]# cat rclocal.txt
  1 #! /bin/sh
  2 #
  3 # This script will be executed *after* all the other init scripts.
  4 # You can put your own initialization stuff in here if you don't
  5 # want to do the full Sys V style init stuff.
  6
  7 touch /var/lock/subsys/local
```

1）将所有行中的第一个“ll”（如果有的话）替换为“TARENA”。

Top

```
[root@svr5 ~]# sed 's/ll/TARENA/' rclocal.txt
  1 #! /bin/sh
```

```
03.      2 #
04.      3 # This script wiTARENA be executed *after* all the other init scripts.
05.      4 # You can put your own initialization stuff in here if you don't
06.      5 # want to do the fuTARENA Sys V style init stuff.
07.      6
08.      7 touch /var/lock/subsys/local
```

2）将所有的“ll”（如果有的话）替换为“TARENA”。

```
01.  [root@svr5 ~]# sed 's/ll/TARENA/g' rclocal.txt
02.      1 #! /bin/sh
03.      2 #
04.      3 # This script wiTARENA be executed *after* aTARENA the other init scripts.
05.      4 # You can put your own initialization stuff in here if you don't
06.      5 # want to do the fuTARENA Sys V style init stuff.
07.      6
08.      7 touch /var/lock/subsys/local
```

3）将第3行内的第2个“script”替换为“SCRIPT”。

```
01.  [root@svr5 ~]# sed '3s/script/SCRIPT/2' rclocal.txt
02.      1 #! /bin/sh
03.      2 #
04.      3 # This script will be executed *after* all the other init SCRIPTs.
05.      4 # You can put your own initialization stuff in here if you don't
06.      5 # want to do the full Sys V style init stuff.
```

Top

```
6
7 touch /var/lock/subsys/local
```

4）删除文件内指定的字符串（替换为空）。

删除所有的“init”字符串：

```
[root@svr5 ~]# sed 's/init//g' rclocal.txt
1 #! /bin/sh
2 #
3 # This script will be executed *after* all the other  scripts.
4 # You can put your own ialization stuff in here if you don't
5 # want to do the full Sys V style  stuff.
6
7 touch /var/lock/subsys/local
```

删除所有的“script”、所有的“stuff”、所有的字母e，或者的关系用转义方式 \| 来表示：

```
[root@svr5 ~]# sed 's/script\|stuff\|e//g' rclocal.txt
1 #! /bin/sh
2 #
3 # This  will b xcutd *aftr* all th othr init s.
4 # You can put your own initialization  in hr if you don't
5 # want to do th full Sys V styl init .
6
7 touch /var/lock/subsys/local
```

Top

5）配置行的注释、解除注释。

以真实文件/etc/rc.local为例，文件内容如下：

```
[root@svr5 ~]# cat /etc/rc.local
#! /bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
```

解除/etc/rc.local文件第3~5行的注释（去掉开头的＃）：

```
[root@svr5 ~]# sed '3,5s/^#//' /etc/rc.local
#! /bin/sh
#
 This script will be executed *after* all the other init scripts.
 You can put your own initialization stuff in here if you don't
 want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
```

Top

将/etc/rc.local文件的第6~7行注释掉（行首添加＃）：

```
[root@svr5 ~]# sed '6,7s/^/#/' /etc/rc.local
#! /bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.
#
#touch /var/lock/subsys/local
```

步骤五：利用sed完成本例要求的任务

参考数据文件内容如下：

```
[root@svr5 ~]# cat nssw.txt
An example Name Service Switch config file. This file should be
sorted with the most- used services at the beginning.
#
The entry '[NOTFOUND=return]' means that the search for an
entry should stop if the search in the previous entry turned
up nothing. Note that if the search failed due to some other reason
(like no NIS server responding) then the search continues with the
```

本小节的操作使用nssw.txt作为测试文件。

1）删除文件中每行的第二个、最后一个字符

分两次替换操作，第一次替换掉第2个字符，第二次替换掉最后一个字符：

Top

```
[root@svr5 ~]# sed 's/.//2;s/.$//' nssw.txt
A example Name Service Switch config file. This file should b
srted with the most- used services at the beginning
#
Te entry '[NOTFOUND=return]' means that the search for a
etry should stop if the search in the previous entry turne
u nothing. Note that if the search failed due to some other reaso
(ike no NIS server responding) then the search continues with th
```

2）将文件中每行的第一个、第二个字符互换

每行文本拆分为“第1个字符”、“第2个字符”、“剩下的所有字符”三个部分，然后通过替换操作重排顺序为“2-1-3”：

```
[root@svr5 ~]# sed -r 's/^(.)(.)(.*)/\2\1\3/' nssw.txt
nA example Name Service Switch config file. This file should be
osrted with the most- used services at the beginning.
#
hTe entry '[NOTFOUND=return]' means that the search for an
netry should stop if the search in the previous entry turned
pu nothing. Note that if the search failed due to some other reason
l(ike n up . Note that if the search failed due to some other
(like NIS server responding) then the search continues with
```

3）删除文件中所有的数字、行首的空格

Top

因原文件内没有数字，行首也没有空格，这里稍作做一点处理，生成一个新测试文件：

```
[root@svr5 ~]# sed 's/o/o7/;s/l/l4/;3,5s/^/  /' nssw.txt > nssw2.txt
[root@svr5 ~]# cat nssw2.txt
An exampl4e Name Service Switch co7nfig file. This file should be
so7rted with the most- used services at the beginning.
  #
  The entry '[NOTFOUND=return]' means that the search fo7r an
  entry sho7ul4d stop if the search in the previous entry turned
up no7thing. Note that if the search fail4ed due to some other reason
(l4ike no7 NIS server responding) then the search continues with the
```

以nssw2.txt文件为例，删除所有数字、行首空格的操作如下：

```
[root@svr5 ~]# sed -r 's/[0-9]//g;s/^( )+//' nssw2.txt
```

4）为文件中每个大写字母添加括号

使用“&”可调用s替换操作中的整个查找串，所以可参考下列操作解决：

```
[root@svr5 ~]# sed 's/[A-Z]/(&)/g' nssw.txt
(A)n example (N)ame (S)ervice (S)witch config file. (T)his file should be
sorted with the most- used services at the beginning.
#
(T)he entry '[(N)(O)(T)(F)(O)(U)(N)(D)=return]' means that the search for an
entry should stop if the search in the previous entry turned
up nothing. (N)ote that if the search failed due to some other reason
(like no (N)(I)(S) server responding) then the search continues with the
```

Top

2 案例2：使用sed修改系统配置

2.1 问题

本案例要求熟悉课上的sed应用案例，并编写脚本anonftp.sh，实现以下功能：

- 通过yum安装vsftpd软件包
- 修改vsftpd服务配置，开启匿名上传
- 调整/var/ftp/pub目录权限，允许ftp写入
- 启动vsftpd服务，并设置开机自运行

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：认识课堂上的sed练习

1）修改默认运行级别

将默认运行级别修改为5，确认修改结果：

```
[root@svr5 ~]# sed -i '/^id:/s/3/5/' /etc/inittab
[root@svr5 ~]# grep "^id:" /etc/inittab
id:5:initdefault:
```

再改回去：

```
[root@svr5 ~]# sed -i '/^id:/s/5/3/' /etc/inittab
[root@svr5 ~]# grep "^id:" /etc/inittab
id:3:initdefault:
```

Top

2）修改IP地址的网段部分，主机地址不变。

直接修改网卡eth0的配置文件，检查原有的配置内容：

```
[root@svr5 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=none
HWADDR=00:0c:29:82:09:e9
ONBOOT=yes
NETMASK=255.255.255.0
IPADDR=192.168.4.4
TYPE=Ethernet
```

若希望将IP地址192.168.4.4修改为172.16.16.4，则应该定位到“IPADDR”所在的行，执行相应的替换（仅测试，尚未修改）：

```
[root@svr5 ~]# sed '/^IPADDR/s/192.168.4.4/172.16.16.4/' \
  /etc/sysconfig/network-scripts/ifcfg-eth0 | grep "^IPADDR"
IPADDR=172.16.16.4
```

要求只修改网段地址时，可以利用扩展正则表达式的 \1、\2、……等调用，分别对应此前第1个、第2个、…… 以 ()包围的表达式所匹配的内容。

所以上述操作可以改为如下（启用扩展匹配应添加 -r 选项）：

```
[root@svr5 ~]# sed -r -i '/^IPADDR/s/192.168.4.(.*)/172.16.16.\1/' \
/etc/sysconfig/network-scripts/ifcfg-eth0
```

Top

确认修改结果：

```
[root@svr5 ~]# grep "^IPADDR" /etc/sysconfig/network-scripts/ifcfg-eth0
IPADDR=172.16.16.4
```

再改回去：

```
[root@svr5 ~]# sed -r -i '/^IPADDR/s/172.16.16.(.*)/192.168.4.\1/' \
/etc/sysconfig/network-scripts/ifcfg-eth0
[root@svr5 ~]# grep "^IPADDR" /etc/sysconfig/network-scripts/ifcfg-eth0
IPADDR=192.168.4.4
```

3）调整httpd服务配置，更改网站根目录

由于需要替换的字符串中有/，为了避免与sed替换操作的分隔混淆，可以使用其他字符作为替换分隔，比如可改用“s#old#new#”的方式实现替换：

```
[root@svr5 ~]# sed -i 's#/var/www/html#/opt/wwwroot#' \
/etc/httpd/conf/httpd.conf
[root@svr5 ~]# grep "^DocumentRoot" /etc/httpd/conf/httpd.conf
DocumentRoot "/opt/wwwroot"
```

若要恢复，可再改回去：

Top

```
[root@svr5 ~]# sed -i 's#/opt/wwwroot#/var/www/html#' \
```

```
/etc/httpd/conf/httpd.conf
[root@svr5 ~]# grep "^DocumentRoot" /etc/httpd/conf/httpd.conf
DocumentRoot "/var/www/html"
```

步骤二：编写anonftp.sh脚本，用来装配匿名FTP服务

1）任务需求及思路分析

vsftpd服务的安装、改目录权限、起服务等操作可以直接写在脚本中。

修改vsftpd.conf配置的工作可以使用sed命令，根据默认配置，只需要定位到以#anon开头的行，去掉开头的注释即可。

2）根据实现思路编写脚本文件

```
[root@svr5 ~]# vim anonftp.sh
#! /bin/bash
yum -y install vsftpd                          //安装vsftpd软件
cp /etc/vsftpd/vsftpd.conf{,.bak}              //备份默认的配置文件
sed -i "/^#anon/s/^#//" /etc/vsftpd/vsftpd.conf     //修改服务配置
chown ftp /var/ftp/pub                         //调整目录权限
/etc/init.d/vsftpd restart                     //启动服务
chkconfig vsftpd on                            //设为自动运行

[root@svr5 ~]# chmod +x anonftp.sh
```

3）验证、测试脚本

运行脚本anonftp.sh：

Top

```
[root@svr5 ~]# ./anonftp.sh
```

```
02.    .. ..
03.    Installed:
04.      vsftpd.x86_64 0:2.0.5-28.el5
05.
06.    Complete!
07.    关闭 vsftpd：                                        [失败]
08.    为 vsftpd 启动 vsftpd：                               [确定]
```

使用ftp登录服务，测试是否可以上传：

```
01.    [root@svr5 ~]# ftp localhost                    //本机访问测试
02.    Connected to localhost.localdomain.
03.    220 (vsFTPd 2.0.5)
04.    530 Please login with USER and PASS.
05.    530 Please login with USER and PASS.
06.    KERBEROS_V4 rejected as an authentication type
07.    Name (localhost:root): ftp                      //匿名登录
08.    331 Please specify the password.
09.    Password:
10.    230 Login successful.
11.    Remote system type is UNIX.
12.    Using binary mode to transfer files.
13.    ftp> cd pub                                     //切换到 pub/ 目录
14.    250 Directory successfully changed.
15.    ftp> put install.log                  //上传当前目录下的install.log 文件
16.    local: install.log remote: install.log
17.    227 Entering Passive Mode (127,0,0,1,192,127)
```

Top

```
150 Ok to send data.
226 File receive OK.
33139 bytes sent in 0.0065 seconds ( 5e+03 Kbytes/s)
ftp> quit                                   //断开FTP连接
221 Goodbye.
```

查看/var/ftp/pub新上传的文件：

```
[ root@svr5 ~] # ls - lh /var/ftp/pub/
总计 36K
- rw- - - - - - -  1 ftp ftp 33K 12- 13 18: 25 install. log
```

3 案例3：sed多行文本处理

3.1 问题

本案例要求使用sed工具来完成下列任务操作：

- 修改主机名配置文件
- 修改hosts文件，添加两条映射记录：192.168.4.5 与 svr5.tarena.com、svr5，还有119.75.217.56与www.baidu.com

3.2 方案

sed工具的多行文本处理操作：

- i：在指定的行之前插入文本
- a：在指定的行之后追加文本
- c：替换指定的行

Top

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：修改主机名配置文件

1）确认修改前的配置

```
[root@svr5 ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=svr5.tarena.com
```

2）使用sed修改主机名配置所在行的内容（c整行替换）

```
[root@svr5 ~]# sed '/^HOSTNAME/cHOSTNAME=mysvr.tarena.com' /etc/sysconfig/network
```

步骤二：修改hosts文件，添加新的记录

1）确认修改前的配置

```
[root@svr5 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

2）使用sed修改hosts文件，添加两行新纪录（a追加）

Top

```
[root@svr5 ~]# sed -i '$a192.168.4.5 svr5.tarena.com svr5\
> 119.75.217.56 www.baidu.com' /etc/hosts
```

```
03.  127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
04.  ::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
05.  192.168.4.5  svr5.tarena.com svr5
06.  119.75.217.56  www.baidu.com
```

4 案例4：sed综合脚本应用

4.1 问题

本案例要求编写脚本getupwd.sh，实现以下需求：

- 找到使用bash作登录Shell的本地用户
- 列出这些用户的shadow密码记录
- 按每行“用户名 --> 密码记录”保存到getupwd.log，如图-1所示

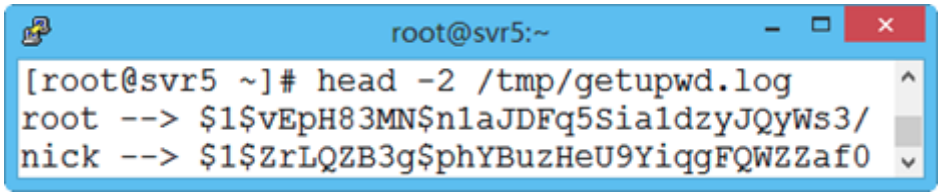

图 - 1

4.2 方案

基本思路如下：

1. 先用sed工具取出登录Shell为/bin/bash的用户记录，保存为临时文件/tmp/urec.tmp，并计算记录数量
2. 再结合while循环遍历取得的账号记录，逐行进行处理
3. 针对每一行用户记录，采用掐头去尾的方式获得用户名、密码字串
4. 按照指定格式追加到/tmp/getuupwd.log文件
5. 结束循环后删除临时文件，报告分析结果

Top

4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：编写getupwd.sh脚本

```
[root@svr5 ~]#vim ./getupwd.sh
#/bin/bash
>/tmp/getupwd.log                                    ##创建空文件
sed -n '/:\/bin\/bash$/w /tmp/urec.tmp' /etc/passwd      ##提取符合条件的账号记录
UNUM=$(egrep -c ':' /tmp/urec.tmp)                       ##取得记录个数
while [ ${i:=1} -le $UNUM ]                              ##从第1行开始，遍历账号记录
do
    UREC=$(sed -n "${i}p" /tmp/urec.tmp)                 ##取指定行数的记录
    NAME=${UREC%%:*}                                     ##截取用户名（记录去尾）
    PREC=$(sed -n "/^$NAME:/p" /etc/shadow)              ##查找与用户名对应的密码记录
    PASS=${PREC#*:}                                      ##掐头
    PASS=${PASS%%:*}                                     ##去尾，只留下密码记录
    echo "$NAME-->$PASS" >> /tmp/getupwd.log             ##保存结果
    let i++                                              ##自增1，转下一次循环
done
/bin/rm -rf /tmp/urec.tmp                                ##删除临时文件
echo "用户分析完毕，请查阅文件 /tmp/getupwd.log"             ##完成后提示

[root@svr5 ~]#chmod +x ./getupwd.sh
```

步骤二：测试、验证执行结果

Top

```
[root@svr5 ~]#./getupwd.sh
```

```
02.     用户分析完毕，请查阅文件 /tmp/getupwd.log
03.
04.     [root@svr5 ~]# less /tmp/getupwd.log
05.     root -->$6$lWgMYmRACwdbfwBo$dr8Yn983nswiJVw0dTMjzbDvSLeCd1GMYjbvsDiFEkL8jnXOLcocBQypOCr4C6BRxNowIxjh6U2qeFU0u1LST/
06.     zengye -->$6$Qb37LOdzRI5995PI$L0zT0gnhGz8ihWkW81J.5XhPp/I7x2./Me2ag0S8tRndCBL9nIjHIKkUKulHxJ6TXyHYmffbVgUT6pbSwf8O71
07.     clamav -->!!
08.     mysql -->!!
09.     abc -->!!
10.     .. ..
```

从上述参考脚本可以发现，使用sed来实现字段提取会比较复杂。下一章课程将会学到awk命令，届时可以通过更简单的方法来改进此脚本内容。

Top