上章回顾

- awk中的内部变量都有哪些
- awk的基本语法是什么
- awk的工作原理是什么

本章内容

- 掌握sed的基本语法结构
- 熟悉sed常用的命令
- 掌握cut工具的使用

一、sed介绍

1、sed的工作流程

1）sed 在处理文件时，遵循特定的工作流程。当 sed 开始运行，它会逐行读取文件内容。每读取一行，就将这一行存入一个**临时缓存区，这个缓存区被称为模式空间。**

2）在模式空间中，sed 会根据用户指定的编辑命令对该行内容进行处理。

- **例如，如果用户指定了替换命令，sed 会在模式空间内完成替换操作；若指定删除命令，则会在模式空间中删除对应的行。**

3）处理完模式空间中的内容后，sed 会把处理后的行发送到屏幕上显示（除非使用了特定选项，如-n取消默认输出）。由于 sed 是对模式空间中的副本进行操作，所以不会直接修改原始文件，这为用户提供了安全可靠的文件处理方式。在完成对一行的处理后，sed 会清空模式空间，然后读取文件的下一行，重复上述过程，直至处理完整个文件。

4）这种工作方式使得 sed 在文件处理方面具有很高的灵活性和安全性，用户可以在不影响原始文件的前提下，对文件内容进行各种复杂的编辑操作。

- 例如，在批量处理大量文本文件时，用户可以使用 sed 快速地修改文件中的特定内容，而不用担心破坏原始数据。

5）sed 主要应用场景包括自动编辑一个或多个文件，在处理系统配置文件时，通过 sed 可以快速修改大量配置项；简化对文件的反复操作，避免手动重复编辑；还可用于编写转换程序，将一种格式的数据转换为另一种格式 。

2、sed的使用方法

1）命令行模式：

- 在命令行中直接输入 sed 命令及相关选项、编辑命令和文件位置，适用于简单、临时的文件处理需求。例如，当需要快速删除文件中某一行时，直接在命令行输入相关 sed 命令即可完成操作 。

2）脚本模式

- 将多个 sed 命令组合成一个脚本文件，通过执行脚本文件来处理文件。这种方式适合处理复杂的文件编辑任务，多个命令可以按照特定的逻辑顺序执行，提高处理效率和代码的可维护性。比如，在进行一系列复杂的文本替换、删除和插入操作时，将这些命令写入脚本文件，方便多次使用和修改 。

3、命令行模式：

1）语法：sed [option] 'sed的命令|地址定位' 文件位置（如果需要引用变量，则需要使用双引号）

- **-e：多项编辑**
  - 功能：当需要对输入行应用多条 sed 命令时使用该选项。它允许你在一次 sed 操作中组合多个不同的命令。

```
sed -e 's/hello/hi/' -e '/sed/d' test.txt          #此命令先执行 s/hello/hi/ 将 hello 替换为 hi，再执行 /sed/d 删除包含 sed 的行。
```

- **-n：取消默认的输出**
  - 功能：通常情况下，sed 会输出处理后的每一行。而使用 -n 选项能取消默认输出，只有在使用 p 命令时才会输出指定的行。

```
sed -n '/world/p' test.txt          #由于使用了 -n 选项，只有匹配到 world 的行才会被打印出来。
```

- **-f：指定sed脚本的文件名称**
  - 功能：若有多个 sed 命令，可将它们写入一个脚本文件，然后用 -f 选项指定该文件，这样可以使复杂的 sed 操作更易于管理和维护。

```
sed -f script.sed test.txt
```

- **-r：使用扩展正则表达式**
  - 默认情况下，sed 使用基本正则表达式。使用 -r 选项可以使用扩展正则表达式，扩展正则表达式提供了更强大和灵活的匹配能力。

```
sed -r '/^[a-zA-Z]/d' test.txt
```

- **-p：打印行**
  - -p 选项通常和 -n 一起使用，用于打印匹配的行。单独使用时，由于 sed 默认会输出处理后的行，所以 -p 效果不明显，但结合 -n 可以精确控制输出内容。

```
sed -n '/example/p' test.txt
```

- **-d：删除行**
  - 功能：该选项用于删除匹配的行。

```
sed '/world/d' test.txt
```

二、sed使用示例

1、新建参考文件

p：打印行

d：删除行

> 如果使用d输出数据就不要使用-n选项

```
[root@server shell06] cat a.txt
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
298374837483
172.16.0.254
10.1.1.1

[root@server ~] sed 'p' a.txt    #打印全部内容
[root@server ~] sed -n 'p' a.txt
    #仅打印副本数据
[root@server ~] sed -n '1p' a.txt
    #仅查看第一行
[root@server ~] sed -n '2p' a.txt
    #仅查看第二行
[root@server ~] sed -n '1,5p' a.txt
    #查看1~5行
[root@server ~] sed -n '5,10p' a.txt
    #查看5~10行
[root@server ~] sed -n '$p' a.txt
    #查看最后一行
[root@server ~] sed '5p' a.txt
        #查看第五行的原和副本
[root@server ~] sed -n '5p' a.txt
    #查看第五行副本
[root@server ~] sed '1d' a.txt
        #不显示第一行
[root@server ~] sed -n '1d' a.txt        #不显示所有航
[root@server ~] sed '1d' a.txt
[root@server ~] sed '1,5d' a.txt        #不显示1~5行
[root@server ~] sed '$d' a.txt
            #不显示最后一行
```

2、使用i、a、c命令

- **i\ 在当前行之前插入文本。多行时除最后一行外，每行末尾需用"\"续行**
- **a\ 在当前行后添加一行或多行。多行时除最后一行外，每行末尾需用“\”续行**
- **c\ 用此符号后的新文本替换当前行中的文本。多行时除最后一行外，每行末尾需用"\"续行 整行替换**

```
[root@server ~] sed '2ixxx' a.txt          #第二行上面添加数据
[root@server ~] sed '$a99999' a.txt
      #最后一行下面添加数据
[root@server ~] sed 'a99999' a.txt
       #每一行下面添加数据
[root@server ~] sed '5chello world' a.txt     #将第五行改为hello world
[root@server ~] sed 'chello world' a.txt      #所有内容全部换为hello world
[root@server ~] sed '1,5chello world' a.txt  #1~5行换为hello world
[root@server ~] sed 'a\aaaaa\nbbbbb\n88888' 1.txt   #在每一行后面添加
[root@server ~] sed '$a yyyyy\n8888' 1.txt #从最后一行开始添加

[root@server ~] sed '/user01/c888888' 1.txt    #将user01替换为888888
[root@server ~] sed '/^user01/c888888' 1.txt    #将行起始字符为user01替换为888888
[root@server ~] sed '18chello world' 1.txt       #将18行替换为hello world
```

3、使用r、w命令

- **r 从文件中读取到的内容再输出输入行**
- **w 将所选的行写入文件（另存为）**

```
[root@server ~] sed '3r /etc/hosts' 2.txt    #将/etc/hosts文件中的数据输入到2.txt文件的中，从第三行开始
[root@server ~] sed '$r /etc/hosts' 2.txt
    #将/etc/hosts文件中的数据输入到2.txt文件的中，从最后一行开始
[root@server ~] sed '/root/w a.txt' 2.txt
    #将2.txt文件中包含root字样的行写入到a.txt文件中
[root@server ~] sed '/[0-9]\{4\}/w a.txt' 2.txt
    #将2.txt中查找包含4个数字的行，并将其写入到文件a.txt中。
[root@server ~] sed -r '/([0-9]{1,3}\.){3}[0-9]{1,3}/w b.txt' 2.txt
-r 使用扩展表达式
/([0-9]{1,3}\.){3}[0-9]{1,3}/ 正则表达式，用于匹配文件中的IP地址
    [0-9]{1,3} 匹配1到3位的数字，
    \. 转义
    {3} 前面的模式重复三次
    [0-9]{1,3} 匹配最后一个数字
w 1.txt 将输出的ip写入的文件
a.txt     要处理的文件
```

4、添加 “!”符号，对所选行以外的所有行(取反)应用命令

```
[root@server ~] sed -n '1!p' 1.txt    #不显示第一行
[root@server ~] sed -n '4p' 1.txt
     #显示第四行
[root@server ~] sed -n '4!p' 1.txt
    #仅不显示第四行
[root@server ~] sed -n '1,17p' 1.txt
    #显示1-17行
[root@server ~] sed -n '1,17!p' 1.txt    #仅不显示1-17行
```

5、使用s、g、&、r命令

1）s：用一个字符串替换另一个,默认每行第一个匹配字符串替换

2）g：进行全局替换

3）&：保护被替换内容

4）r：扩展正则

> 注意：搜索替换中的分隔符可以自己指定

```
[root@server ~] sed -n 's/root/ROOT/p' 1.txt    #将带有root字段的行第一个root改为ROOT
[root@server ~] sed -n 's/root/ROOT/gp' 1.txt
    #将带有root字段的行所有root改为ROOT
[root@server ~] sed -n 's/^root/#root/p' a.txt    #将行开头为root字段的行的开头改为#root
[root@server ~] sed 's/[/:\.]//gp' a.txt
        #删除文件中的“/:.”符号
[root@server ~] sed -n 's@/sbin/nologin@qingmei@gp' a.txt    #将/sbin/nologin改为qingmei，用@符号进行替换
[root@server ~] sed -n 's/\/sbin\/nologin/qingmei/gp' a.txt    #使用转义符查找要的字符串，将/sbin/nologin改为qingmei
[root@server ~] sed -n '10s#/sbin/nologin#qingmei#p' a.txt    #将第10行的数据进行替换，使用#进行替换，@符号也可以

[root@server ~] sed -n '1,5s/^/#/p' a.txt 注释掉文件的1-5行内容
#root:x:0:0:root:/root:/bin/bash
#bin:x:1:1:bin:/bin:/sbin/nologin
#daemon:x:2:2:daemon:/sbin:/sbin/nologin
#adm:x:3:4:adm:/var/adm:/sbin/nologin
#lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin


[root@server ~] sed -n 's/^root/#&/p' a.txt    #使用“&”符号保留root，在root之前添加注释号
#root:x:0:0:root:/root:/bin/bash
```

```
[root@server ~] sed -n -r 's/^root|^stu/#&/p' /etc/passwd     #注释掉以root开头或者以stu开头的行
[root@server ~] sed -n '1,5s/^[a-z].*/#&/p' passwd     #注释掉1~5行中以任意小写字母开头的行
[root@server ~] sed -n '1,5s/^/#/p' /etc/passwd      #注释1~5行
或者
sed -n '1,5s/^/#/p' passwd 以空开头的加上#
sed -n '1,5s/^#//p' passwd 以#开头的替换成空
[root@server ~]# sed -n '/^root/p' 1.txt     #显示以root开头的行
[root@server ~]# sed -n 's/^root/#&/p' 1.txt
    #使用“&”符号保留root，在root之前添加注释号
[root@server ~]# sed -n 's/\(^root\)/#\1/p' 1.txt     #\(^root\)是一个正则表达式，表示以root开头。表示在行前添加#
[root@server ~]# sed -nr '/^root|^stu/p' 1.txt     #匹配root和stu开头的行
[root@server ~]# sed -nr 's/^root|^stu/#&/p' 1.txt     #保留起始位并且添加注释
```

6、**使用“=”打印行号；e 多项编辑**

```
[root@server ~] sed -n '/bash$/=' passwd      #打印以bash结尾的行的行号
[root@server ~] sed -ne '/root/=' -ne '/root/p' passwd     #同时显示行号以及数据
[root@server ~] sed -n '/nologin$/=;/nologin$/p' 1.txt     #同时显示行号以及数据
[root@server ~] sed -ne '/nologin$/=' -ne '/nologin$/p' 1.txt #同时显示行号以及数据
[root@server ~] sed -e '5ihello world' -e '8ahome' 1.txt      #第五、八行分别向上、向下添加数据
```

7、综合运用

```
[root@server ~] sed -n '1,5s/^/#&/p' 1.txt
#root:x:0:0:root:/root:/bin/bash
#bin:x:1:1:bin:/bin:/sbin/nologin
#daemon:x:2:2:daemon:/sbin:/sbin/nologin
#adm:x:3:4:adm:/var/adm:/sbin/nologin
#lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

[root@server ~] sed -n '1,5s/\(^\)/#\1/p' 1.txt
#root:x:0:0:root:/root:/bin/bash
#bin:x:1:1:bin:/bin:/sbin/nologin
#daemon:x:2:2:daemon:/sbin:/sbin/nologin
#adm:x:3:4:adm:/var/adm:/sbin/nologin
```

```
#lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

8、使用 i 直接修改源文件

> 切记：-ni命令不可以一起使用
> -n只打印输出的结果，而不会修改源文件
> -i直接修改源文件
> 使用-i之后p也无法被使用，原因是因为-i后直接将数据写入到文件中，而不会再次输出了

```
[root@server ~] sed -i 's/root/ROOT/gp' a.txt     #将root改为ROOT
[root@server ~] sed -i 's/root/ROOT/g' a.txt     #
[root@server ~] sed -i '2d' a.txt     #删除第二行
[root@server ~] sed -i '1,3s/^#//' a.txt     #1~3行将#删除
[root@server ~] sed -i '1,3s/^/#/' a.txt     #1~3行添加#
```

9、综合练习

```
1.过滤vsftpd.conf文件中以#开头和空行：
[root@server ~] grep -Ev '^#|^$' /etc/vsftpd/vsftpd.conf
[root@server ~] sed -e '/^#/d' -e '/^$/d' /etc/vsftpd/vsftpd.conf
[root@server ~] sed '/^#/d;/^$/d' /etc/vsftpd/vsftpd.conf
[root@server ~] sed -r '/^#|^$/d' /etc/vsftpd/vsftpd.conf
2.过滤smb.conf文件中生效的行：
[root@server ~] sed -e '/^#/d' -e '/^;/d' -e '/^$/d' -e '/^\t$/d' -e '/^\t#/d'
 smb.conf
[root@server ~] sed -r '/^(#|$|;|\t#|\t$)/d' smb.conf
3.获取ifcfg-ens32配置文件中的IP，子网掩码
[root@server ~] sed -rn '/(^IPADDR|^NETMASK)/p' /etc/sysconfig/networkscripts/ifcfg-
ens33 | awk -F = '{print $2}'
#grep -o只显示匹配到的内容
[root@server ~] grep -o -E '([0-9]{1,3}\.){3}[0-9]{1,3}'
/etc/sysconfig/network-scripts/ifcfg-ens33
```

10、总结

1）sed：--sed命令---选项---或者正则表达式--或者地址定位-----文件名

2）sed主要就是针对文件内容进行过滤、替换，定址就是决定哪些用于编辑；地址的形式可以是数字、正则或者二者的结合，如果没有指定sed默认处理所有行

```
x 指定x行号 sed -n '5p' 1.txt
```

```
x,y 指定x到y行号 sed -n '1,5p' 1.txt
/key/ 查询包含关键字的行 sed -n '/root/p' 1.txt
/key1/,/key2/ 匹配包含两个关键字之间的行 sed -n '/^adm/,/^mysql/p' 1.txt
/key/,x 从匹配关键字的行开始到文件第x行之间的行（包含关键字所在行）sed -n '/^lp/,7p'
x,/key/ 从第x行开始到与关键字的匹配行之间的行 sed -n '7p,/^lp/'
x,y! 不包含x到y行  sed -n '!3,5p'
```

三、sed脚本格式

1、用法

```
[root@server ~] sed -f scripts.sed -i file //使用脚本处理文件
建议使用 ./sed.sh file
-i 替换源文件
脚本的第一行写上
---------------------
#!/bin/sed -f
1,5d
s/root/hello/g
```

2、注意事项

1） 脚本文件是一个sed的命令行清单。'commands'

2） 在每行的末尾不能有任何空格、制表符（tab）或其它文本。

3） 如果在一行中有多个命令，应该用分号分隔。

4） 不需要且不可用引号保护命令

5） #号开头的行为注释

3、实例（一）

- 在第2行之后插入一行内容，内容为"******************"。
- 将第2行到文件末尾的内容中的"stu"替换为"user"。
- 在文件末尾插入一行内容，内容为"we inster new line"。
- 将文件开头的以小写字母开头的行替换为以"#"开头的行

```
[root@server ~]# cat passwd
stu3:x:509:512::/home/user3:/bin/bash
stu4:x:510:513::/home/user4:/bin/bash
stu5:x:511:514::/home/user5:/bin/bash
[root@server ~]# cat sed.sh
#!/bin/sed -f
2a******************
```

```
2,$s/stu/user/
$a we inster new line
s/^[a-z].*/#&/
```

4、实例（二）

- 在第3行之后插入一行内容，内容为"**********"。
- 将第1行到第3行的内容替换为以"#"开头的行。
- 在末尾插入一行内容，内容为"hello world"。

```
[root@server ~]# cat 1.sed
#!/bin/sed -f
3a**********
$chello world
1,3s/^/#&/
[root@server ~]# sed -f 1.sed -i 11.txt
[root@server ~]# cat 11.txt
#root:x:0:0:root:/root:/bin/bash
#bin:x:1:1:bin:/bin:/sbin/nologin
#daemon:x:2:2:daemon:/sbin:/sbin/nologin
**********
adm:x:3:4:adm:/var/adm:/sbin/nologin
helloworld
```

四、sed结合正则使用

1、正则表达式必须以”/“前后规范间隔

- 例如：sed '/root/d' file
- 例如：sed '/^root/d' file

2、如果匹配的是扩展正则表达式，需要使用-r选项扩展sed

- grep -E
- sed -r
- + ? () {n,m} | \d

> 如果在正则中存在特殊字符，如：（^$. * / []等）均需要使用“\”号做转义，转成普通符号

3、逗号分隔符

- 例如：sed '5,7d' file　　#删除5到7行
- 例如：sed '/root/,/ftp/d' file　　#删除第一个匹配字符串"root"到第一个匹配字符串"ftp"的所有行本行不找 循环执行

4、组合方式

- 例如：sed '1,/foo/d' file　#删除第一行到第一个匹配字符串"foo"的所有行

- 例如：sed '/foo/,+4d' file #删除从匹配字符串”foo“开始到其后四行为止的行
- 例如：sed '1~5d' file　　#从第一行开始删每五行删除一行
- 例如：sed -nr '/foo|bar/p' file　#显示配置字符串"foo"或"bar"的行
- 例如：sed -n '/foo/,/bar/p' file　#显示匹配从foo到bar的行
- 例如：sed '1~2d' file　　#删除奇数行
- 例如：sed '0~2d' file　　#删除偶数行

5、特殊情况.

- 例如：sed '$d' file　#删除最后一行
- 例如：sed '1d' file　#删除第一行

6、其他:

- sed 's/.//' a.txt　　#删除每一行中的第一个字符
- sed 's/.//2' a.txt　　#删除每一行中的第二个字符
- sed 's/.//N' a.txt　　#从文件中第N行开始，删除每行中第N个字符（N>2）
- sed 's/.$//' a.txt　　#删除每一行中的最后一个字符

五、cut命令

1、cut工具是一种在计算机中用于处理文本和数据的命令行工具。它可以从文件的每一行剪切字节、字符和字段，并将这些字节、字符和字段输出。

```
# cut 选项 文件名
常用选项：
-c: 以字符为单位进行分割,截取
-d: 自定义分隔符，默认为制表符\t
-f: 与-d一起使用，指定截取哪个区域
举例：
[root@server ~]# cut -d: -f1 1.txt 以:冒号分割，截取第1列内容
[root@server ~]# cut -d: -f1,6,7 1.txt 以:冒号分割，截取第1,6,7列内容
[root@server ~]# cut -c4 1.txt 截取文件中每行第4个字符
[root@server ~]# cut -c1-4 1.txt 截取文件中每行的1-4个字符
[root@server ~]# cut -c4-10 1.txt 截取文件中每行的4-10个字符
[root@server ~]# cut -c5- 1.txt 从第5个字符开始截取后面所有字符
```

作业：

1. 将任意数字替换成空或者空格
2. 去掉文件1-5行中的数字、冒号、斜杠
3. 匹配root关键字替换成hello，并把改后的文件保存到test.txt文件中
4. 注释掉文件的2-3行和匹配到以root开头或者以ftp开头的行

5. 优化apache安装脚本，从yum安装到最终软件启动进行编写
6. 编写mysql安装脚本（切记保留下临时密码）