# 三剑客-基础部分题目

1. <https://blog.csdn.net/qq_37242520/article/details/89376488>

涉及到三剑客过滤,替换,取列操作

1. 选择

在给定文件中查找与条件相符字符串的命令及查找某个目录下相应文件的命令为：（多选）A C

A:grep 筛选

B:gzip

C:find 查找目录下文件

1

2

3

sort在给定文件中查找(筛选)与 设定条件相符字符串的命令为（A）

A：grep

B：gzip

C：find

D：sort

1

2

3

4

\_\_\_D\_\_\_\_命令可以从文本文件的每一行中截取指定内容的数据。

A：cp

B：dd

C：fmt

D：cut (剪切)

1

2

3

4

下面表述错误的是D

^表示开头，如^abc表示以abc开头的

$表示结尾如abc$表示以abc为结尾的

？表示前一个字符出现0或1次，如ab？c表示a与c之间可有0或1个b存在

+ 表示前一个字符出现0或多次。如ab+c表示a与c之间可有0或多个b存在

1

2

3

4

填空

Unix中在当前目录下所有.cc 的文件中找到含有“oldboy”内容的文件，命令为\_\_\_

find -type f -name '\*.cc'|xargs grep 'asiainfo'

find -type f -name '\*.cc' -exec grep 'asiainfo' {} \;

grep 'asiainfo' \`find . -type f -name '\*.cc'`

如果没有指定扩展名则直接使用

grep -rl 'oldboy' ./\*

-r 递归查询

-l (小写字母L)只显示grep查找出来的文件

1

2

3

4

5

6

7

8

使用AWK指定分隔符的参数是\_\_\_-F或-vFS=\_\_\_\_

找出当前目录下包含127.0.0.1关键字的文件？

1.

grep '127.0.0.1' `find . -type f `

2.三剑客+反引号

sed -n '/127.0.0.1/p' `find . -type f`

awk '/127.0.0.1/' `find . -type f`

3.find+|xargs+三剑客

find . -type f|xargs grep '127.0.0.1'

find . -type f|xargs sed -n '/127.0.0.1/p'

find . -type f|xargs awk '/127.0.0.1/'

4.find -exec +三剑客

find .-type f -exec grep '127.0.0.1' {} \;

find .-type f -exec sed -n '/127.0.0.1/p' {} \;

find .-type f -exec awk '/127.0.0.1/' {} \;

1

2

3

4

5

6

7

8

9

10

11

12

13

删除/显示1.txt文件第3行到第10行的内容？

删除:

sed -n '3,10p' 1.txt

sed -i '3,10d' 1.txt

awk '(NR>=3&&NR<=10)' 1.txt \\把第三行到第十行排除

awk '(NR<3||NR>10)' oldboy.txt \\ 小于第三行或者大于十行

显示:

sed -n '3,10p' 1.txt

awk 'NR==3,NR==7' 1.txt

awk 'NR>=3 && NR<=10' 1.txt

head -10 num.txt|tail -8

1

2

3

4

5

6

7

8

9

10

11

使用awk取出/etc/passwd第一列数据也就是取出用户名。

基础必备：

awk -F: '{print $1}' /etc/passwd

sed 's#:.\*$##g' /etc/passwd

egrep -o '^[0-Z\_-]+' /etc/passwd \\[0-Z]==[a-zA-Z0-9]

提高：

egrep -o '^[^:]+' /etc/passwd

sed -r 's#(^[^:]+).\*$#\1#g' /etc/passwd

awk '{gsub(/:.\*$/,"");print}' /etc/passwd

\\print 在 awk 中后面什么都不加表示 print $0

awk '{gsub(/:.\*$/,"")}1' /etc/passwd

1

2

3

4

5

6

7

8

9

10

11

12

用命令行更改config.txt文件，把里面所有的“name”更改为“address”

sed 's#name#address#g' config.txt

awk '{gsub(/name/,"address")}1' config.txt

利用vi/vim也可以替换：

:%s#name#address#g

1

2

3

4

5

写出查询file.txt以abc结尾的行

grep 'abc$' file.txt

sed -n ‘/abc$/p’ file.txt

awk '/abc$/' file.txt

1

2

3

删除file.txt文件中的空行

模拟环境:

echo -e 'oldboy\noldboy\n\nalex\n \nalex\lidao lidao alex\n

oldboy\n\noldboyedu group 598972270' >file.txt

仅删除文件中的空行：

grep -v '^$' file.txt

sed '/^$/d' file.txt

awk '!/^$/' file.txt

文件中的空行可能是只有空格的行

egrep -v '$| +$' file.txt

grep -v '^ \*$' file.txt

sed -n '/^ \*$/p' file.txt

awk '/^ \*$/' file.txt

^ \*$ 表示两种情况:

\*表示前1个字符出现0次的时候:^$

\*表示前1个字符出现1次及1次以上的时候:^ +$

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

打印file.txt文件中的第10行

sed -n '10p' 4.txt

awk 'NR==10' 4.txt

head -10 file.txt|tail -1

1

2

3

删除/etc/fstab文件中所有以#开头的行，行首的#号及#后面的所有空白字符

空白字符：空格,多个空格,tab键

sed -r '/^#/s#\# \*##g' /etc/fstab|cat -A \\查看一下行尾是否还有空白行

awk 'gsub (/^\# \*/,"");{print $1}' /etc/fstab|cat -A

此题可以分为两个题目:

1. 删除/etc/fstab 文件中所有以#开头的行的行首的# 号及#后面的所有空白字符

2. 删除/etc/fstab 文件中所有以#开头的行的行首的#号及#后面的连续的空白字符

可以分为两步:哪个是条件，哪个是动作

❑第 1 步找出以#开头的行

awk '/^#/{gsub(/^#|[ \t]+/,"")}1' /etc/fstab

sed -r '/^#/s@^#|[ \t]+@@g' /etc/fstab

❑第 2 步在删除这一行中的空白字符和行首的井号

删除/etc/fstab 文件中所有以#开头的行的行首的# 号及#后面的连续的空白字符

awk '{gsub(/^#[ \t]+/,"")}1' /etc/fstab

sed -r 's@^#[ \t]+@@g' /etc/fstab

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

查找最后修改时间是3天前，后缀是\*.log的文件并删除

find /tmp/ -type f -mtime +3 -name '\*.log' |xargs rm -f

find /tmp/ -type f -mtime +3 -name '\*.log' -exec rm -f {} \;

\rm -rf `find /tmp/ -type f -mtime +3 -name '\*.log' `

find /tmp/ -type f -mtime +3 -name '\*.log' -exec rm {} +

find /oldboy -mtime +3 -type f -name "\*.log" -delete \\ -delete删除

\\ \; -exec find 命令找出 1 个文件就把文件放在{}的位置

\\ +;如果-exec后面使用+ find会把前面找到的所有文件1次性的放在{}所在位置

1

2

3

4

5

6

7

8

9

10

11

12

查找文件中1到20行之间，同时将"aaa"替换为"AAA",“ddd"替换"DDD”

sed -ne 's#a#A#g;s#d#D#g;1,20p' 2.txt

sed '1,20y/ad/AD/' 2.txt \\ y类似于tr的 1对1 替换 不能使用 a-z A-Z 范围

sed 's#[ad]#\U&#g' 2.txt \\

tr 'ad' 'AD' <2.txt

awk '{gsub(/a/,"A");gsub(/d/,"D");print}' 2.txt

awk 'NR>=1&&NR<=20{print toupper($0)}' 2.txt

awk 'NR==1,NR==20{print toupper($0)}' 2.txt

1

2

3

4

5

6

7

8

9

10

11

12

13

使用sed命令打印出文件的第一行到第三行

head -3 lidao.txt

sed -n '1,3p' lidao.txt

awk 'NR>=1&&NR<=3' lidao.txt

awk 'NR==1;NR==2;NR==3' lidao.txt

1

2

3

4

5

6

7

在/var/sync/log/cef\_watchd-20080424.1og文件中查找大小写不敏感“mysql"字符串中的命令是

grep -iw 'mysql' /var/sync/log/cef\_watchd-20080424.1og

sed -nr '/(mysql|MYSQL)/p' /var/sync/log/cef\_watchd-20080424.1og

sed -nr '/(mysql|MYSQL)/Ip' /var/sync/log/cef\_watchd-20080424.1og \\sed中 I表示不区分大小写,极少用

awk -vIGNORECASE=1 '/[a-z]/' /var/sync/log/cefwatchd-20080424.log

1

2

3

4

5

6

7

为了在/var/sync/log 目录中查找更改时间在5日以前的文件井删除它们，linux命令是

find /var/sync/log -type f -mtime +5 |xargs rm -rf

find /var/sync/log -type f -mtime +5 -exec rm -rf {} \;

rm -rf `find /var/sync/log -type f -mtime +5`

1

2

3

4

5

把data目录及其子目录下所有以扩展名.txt结尾的文件中包含oldgirl的字符串全部替换为oldboy.

sed -i 's#oldgirl#oldboy#g' `find /data/ -type f -name '\*.txt'`

find /data/ -type f -name '\*.txt'|xargs sed -ni 's#oldboy#oldgirl#gp'

1

2

3

4

5

写出删除abc.txt文件中的空行

sed '/^$/d' abc.txt

sed -n '/^$/!p' 1.txt

grep -v '^$' 1.txt

1

2

3

查看linux服务器ip的命令，同时只显示包含ip所在的行打印出来

ip a s eth0|sed -n '3p'|sed -r 's#^.\*t ([0-9].\*)/.\*$#\1#g' #sed多管道

ip a s eth0|sed -nr '3s#^.\*t (.\*)/.\*$#\1#gp' #用sed

ip a s eth0|awk -F'[/ ]+' 'NR==3{print $3}' #用awk

ip a s eth0|sed -n '3p'|awk -F'[ /]+' '{print$3}' #sed+awk

ip a s eth0|awk 'NR==3'|sed -r 's#^.\*t (.\*)/.\*$#\1#g' #awk+sed

ip a s eth0|grep 'inet '|sed -r 's#^.\*t (.\*)/.\*#\1#g' #grep+sed

ip a s eth0|grep 'inet '|awk -F'[ /]+' '{print $3}' #grep+awk

1

2

3

4

5

6

7

用awk获取文件中第三行的倒数第二列字段

awk 'NR==3{print $(NF-1)}' 2.txt

1

有文件file1

请用shell查询file1里面空行的所在行号

grep -n '^$' file1.txt

awk '/^$/{print NR,$0}' file1.txt

sed -n '/^$/{=;p}' file.txt|sed 'N;s#\n# #g'

1

2

3

4

5

查询file1以abc结尾的行

grep 'abc$' file.txt

sed -n ‘/abc$/p’ file.txt

awk '/abc$/' file.txt

1

2

3

打印file文件第一行到第三行

head -3 lidao.txt

sed -n '1,3p' lidao.txt

awk 'NR>=1&&NR<=3' lidao.txt

awk 'NR==1;NR==2;NR==3' lidao.txt

1

2

3

4

显示file文件里匹配foo字串那行以及上下5行、显示foo及前5行、显示foo及后5行

grep -A5 'foo' file.txt

grep -B5 'foo' file.txt

grep -C5 'foo' file.txt

1

2

3

echo “ABCDEF abtdefg” |sed -i s/c/t/g 这条命令是否正确？如若正确可能会显示什么样的结果？

不正确

-i 修改文件内容

no input files

sed 无法找到要修改的文件

1

2

3

4

当前系统中没有任何文本编辑器（vi,emacs,vim,edit等），如何过滤掉注释行和空行查看/etc/ssh/sshd\_config文件

sed -r '/^#|^$/d' /etc/ssh/sshd\_config

sed -nr '/^$|^#/!p' /etc/ssh/sshd\_config

awk '!/^$|^#/' /etc/ssh/sshd\_config

grep '^[a-Z]' /etc/ssh/sshd\_config

egrep -v '^$|^#' /etc/ssh/sshd\_config

1

2

3

4

5

查找file.log文件中的包含关键字“helloworld”的内容，及其上下两行的内容重定向保存到1.txt。请写出命令

grep -C2 'helloworld' file.log >> 5.txt

1

2

二. 进阶题目:

需要用到awk数组和函数知识点

利用sed命令将test.txt中所有的回车替换成空格？

模拟环境：

echo -e 'oldboy\noldboy\n\nalex\n \nalex\lidao lidao alex\n oldboy\n\noldboyedu group 598972270' >file.txt

仅删除文件中的空行

grep -v '^$' file.txt

sed '/^$/d' file.txt

awk '!/^$/' file.txt

文件中的空行可能是只有空格的行

egrep -v '$| +$' file.txt

grep -v '^ \*$' file.txt

sed -n '/^ \*$/p' file.txt

awk '/^ \*$/' file.txt

^ \*$ 表示两种情况:

表示前1个字符出现0次的时候:^$

表示前1个字符出现1次及1次以上的时候:^ +$

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

从a.log文件中提取包含“WARNING”或“FATAL”，同时不包含“IGNOR”行，然后，提取以“：”分割的第五个字段

awk -F: '/WARNING|FATAL/&&!/IGNOR/{print $5}'

1

找出ifconfig命令结果中的1-255之间的数值；

net-tools

ifconfig| egrep -o '[0-9]+'|awk '$0>=1 && $0<=255'

1

2

文件内容排序、文件行取唯一值的命令分别是？

sort

uniq

1

2

用awk获取文件中第三行的倒数第二列字段

awk 'NR==3{print $(NF-1)}' file.txt

1

awk是一个很强大的文本处理工具，请使用awk统计当前主机的并发访问量

netstat -ant | awk '/^tcp/ {++state[$NF]} END {for(key in state) print key,"\t",state[key]}'

1

三. 老牛逼题目 15k+

有一个文件nginx.log，请用awk提取时间（HH:MM）,文件内容如下：

183.250.220.178|-l[20/jul/2017:10:35:14 +0800]|POST /audiosearch/search

HTTP/1.1|200|54|-lDalvik/1.6.0（linux;U;Android 4,4,4;Konka Android TV 638

Build/KTU84P）|-l-[5.069|5.001,0.005|www.kuyun.com|8771|172.21.19.67:8084,172.21.19.66:8084]

183.250.220.178|-l[20/jul/2017:10:35:14 +0800]|POST /audiosearch/search

HTTP/1.1|200|54|-lDalvik/1.6.0（linux;U;Android 4,4,4;Konka Android TV 638

Build/KTU84P）|-l-[5.069|5.001,0.005|www.kuyun.com|8771|172.21.19.67:8084,172.21.19.66:8084]

统计apache访问日志流量排名前10个IP

统计出apache的access.log 中访问最多的5个ip

awk ‘{h[$1]++}END{for(po in h) print po,h[po]}’ access.log |awk ‘{print $2,$1}’ |sort -rn|head -5 sort -rn 以相反的顺序依照数值大小排序

这条linux命令的含义：netstat -n |awk ‘/^tcp/ {++} print a,${a} }’

该命令执行不了

使用awk分析日志（最后一个字段是IP，统计访问量前十的IP，并进行排序）

awk -F: '/WARNING|FATAL/&& !/IGNOR/{print $5}' a.log

# 第二个版本

# [linux基础学习-10.4-正则表达式练习题](https://www.cnblogs.com/Klanti/p/9143253.html)

<https://www.cnblogs.com/Klanti/p/9143253.html>

### 一、基础正则

##### 环境准备

[root@oldboyedu42-lnb oldboy]# cat /oldboy/re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is

http://oldboy.blog.51cto.com

our size is

http://blog.oldboyedu.com

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

#### 1、^ ^m 表示以....开头的行

[root@oldboyedu42-lnb oldboy]# grep '^m' re.txt

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

my qq is 49000448

my god ,i am not oldbey,but OLDBOY!

#### 2、 $ m$ 表示以....结尾的行

[root@oldboyedu42-lnb oldboy]# cat -A re.txt

I am oldboy teacher!$

I teach linux.$

$

I like badminton ball ,billiard ball and chinese chess!$

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/) $ #注意此处结尾是空格不是m

$

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/) $ #注意此处结尾是空格不是m

$

my qq is 49000448$

$

not 4900000448.$

my god ,i am not oldbey,but OLDBOY!$

#删除结尾的空格

[root@oldboyedu42-lnb oldboy]# vim re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

#检查是否修改成功

[root@oldboyedu42-lnb oldboy]# cat -A re.txt

I am oldboy teacher!$

I teach linux.$

$

I like badminton ball ,billiard ball and chinese chess!$

my blog is http://oldboy.blog.51cto.com$

$

our size is http://blog.oldboyedu.com$

$

my qq is 49000448$

$

not 4900000448.$

my god ,i am not oldbey,but OLDBOY!$

[root@oldboyedu42-lnb oldboy]#

[root@oldboyedu42-lnb oldboy]# grep 'm$' re.txt

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

cat -A #显示出文件中所有的符号 $这一行的结尾

[root@oldboyedu42-lnb oldboy]# grep ' ' re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

#### 3、 ^$ 空行 这一行里面什么都没有

[root@oldboyedu42-lnb oldboy]# grep -n '^$' re.txt

3:

6:

8:

10:

练习题:排查文件中的空行

[root@oldboyedu42-lnb oldboy]# grep -v '^$' re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

#### 4、 . 任意一个字符 不会匹配空行

#grep -o '.' re.txt

#-o 显示grep的执行过程，grep每一次找出什么

grep -o '.' re.txt

#-o 显示grep的执行过程，grep每一次找出什么

#-o 的结果中 每一行表示 grep每次找出什么

[root@oldboyedu42-lnb oldboy]# grep '0' re.txt

my qq is 49000448

not 4900000448.

[root@oldboyedu42-lnb oldboy]# grep -o '0' re.txt

0

0

0

0

0

0

0

0

[root@oldboyedu42-lnb oldboy]# grep '00' re.txt

my qq is 49000448

not 4900000448.

[root@oldboyedu42-lnb oldboy]# grep -o '00' re.txt

00

00

00

说明：如果匹配的内容在一行中有多处，grep会从左到右匹配到最后一个，多多益善

提示：点（.）的特殊含义小结：

1、当前目录

2、使得文件生效相当于source

3、隐藏文件的开头

4、任意一个字符

#### 5、 撬棍 转义字符 去掉符号特殊含义 脱掉马甲，打回原形

显示出文件中 以.结尾的行？

[root@oldboyedu42-lnb oldboy]# grep '.$' re.txt

I teach linux.

not 4900000448.

撬棍系列 转义字符系列

\n ====== 回车

#### 6、 \* 前一个字符连续出现0次或0次以上

[root@oldboyedu42-lnb oldboy]# grep '0\*' re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

[root@oldboyedu42-lnb oldboy]#

[root@oldboyedu42-lnb oldboy]# grep '0\*' re.txt -o

000

00000

#连续出现

#正则表示连续出现的时候，会尽可能的匹配（吃）更多的符号 贪婪性

[root@oldboyedu42-lnb oldboy]# #0\* 表示0连续出现0次

[root@oldboyedu42-lnb oldboy]# #0\* 表示0连续出现1次及1次以上

[root@oldboyedu42-lnb oldboy]# #0

[root@oldboyedu42-lnb oldboy]# #000

[root@oldboyedu42-lnb oldboy]# #0000000

[root@oldboyedu42-lnb oldboy]# #0\* 表示0连续出现0次

[root@oldboyedu42-lnb oldboy]# #'0\*' 只出现0次的时候 ====== ''

[root@oldboyedu42-lnb oldboy]# #会把整个文件的内容都显示出来

[root@oldboyedu42-lnb oldboy]# grep '' re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

小结正则之\*：

1]] 贪婪性

2]] 连续出现

#### 7、 .\* 所有 任何符号

#在正则中表示连续出现 表示所有 贪婪性

找出文件中以字母m开头的行 并且 以m结尾的行

[root@oldboyedu42-lnb oldboy]# grep '^m' re.txt

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

my qq is 49000448

my god ,i am not oldbey,but OLDBOY!

[root@oldboyedu42-lnb oldboy]# grep '^m' re.txt |grep 'm$'

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

[root@oldboyedu42-lnb oldboy]# grep '^m.\*m$' re.txt

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

[root@oldboyedu42-lnb oldboy]# grep '^.\*m$' re.txt

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

[root@oldboyedu42-lnb oldboy]# grep 'm$' re.txt

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

[root@oldboyedu-42mvp ~]# grep '.\*m' re.txt

#### 8、 [] [abc] 表示一个整体，a或b或c任意一个字符

[root@oldboyedu42-lnb oldboy]# grep '[abc]' re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my god ,i am not oldbey,but OLDBOY!

grep '[abc]' re.txt -o

===>从哪里来，到哪里去用的是-

[root@oldboyedu42-lnb oldboy]# grep '[a-z]' re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

[root@oldboyedu42-lnb oldboy]# grep '[A-Z]' re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my god ,i am not oldbey,but OLDBOY!

[root@oldboyedu42-lnb oldboy]# grep '[a-zA-Z]' re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

#找出文件中以m或n开头的行

[root@oldboyedu42-lnb oldboy]# #第1个里程碑-m或n

[root@oldboyedu42-lnb oldboy]# grep '[mn]' re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

[root@oldboyedu42-lnb oldboy]# #第2个里程碑-以m或n开头的行

[root@oldboyedu42-lnb oldboy]# grep '^[mn]' re.txt

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

grep '[a-z]' re.txt

grep '[A-Z]' re.txt

grep '[0-9]' re.txt

#### 9、 [^] 排除 [^abc]

[root@oldboyedu42-lnb ~]# grep '[^abc]' /oldboy/re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

||||||||||||||||||000000000

### 二、扩展正则

#### 1、 + 前一个字符连续出现1次或多次

[root@oldboyedu42-lnb oldboy]# egrep '0+' re.txt

my qq is 49000448

not 4900000448.

[root@oldboyedu42-lnb oldboy]# egrep '0' re.txt

my qq is 49000448

not 4900000448.

[root@oldboyedu42-lnb oldboy]# egrep '0+' re.txt -o

000

00000

#+可以把连续的字符变为一个整体

[root@oldboyedu42-lnb oldboy]# egrep '0' re.txt -o

0

0

0

0

0

0

0

0

[root@oldboyedu42-lnb oldboy]# #连续出现

取出文件中连续出现的小写字母

[root@oldboyedu42-lnb oldboy]# #取出文件中连续出现的小写字母

[root@oldboyedu42-lnb oldboy]# egrep '小写字母+' re.txt

[root@oldboyedu42-lnb oldboy]# egrep '[a-z]+' re.txt

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my qq is 49000448

not 4900000448.

my god ,i am not oldbey,but OLDBOY!

[root@oldboyedu42-lnb oldboy]# egrep '[a-z]+' re.txt -o

am

oldboy

teacher

teach

linux

like

badminton

ball

billiard

ball

and

chinese

chess

my

blog

is

http

oldboy

blog

cto

com

our

size

is

http

blog

oldboyedu

com

my

qq

is

not

my

god

i

am

not

oldbey

but

小结:

1.+连续出现 1次及多次

1. 可以把连续的字符变为一个整体，一次取出来(-o)
2. +一般与[]配合

#1.找出规律

#2.中英文符号

#3.区分大小写

[root@oldboyedu42-lnb oldboy]# egrep '^[0-9]+$' id.txt

440304199604012792

130528197108126121

342923198310042132

330900199806382320

654126197703092303

131127197105115662

[root@oldboyedu42-lnb oldboy]# egrep '^[0-9]+X$' id.txt

61242619860416291X

[root@oldboyedu42-lnb oldboy]# egrep '^[0-9]+[0-9X]$' id.txt

440304199604012792

130528197108126121

342923198310042132

61242619860416291X

330900199806382320

654126197703092303

131127197105115662

#### 2、 | 或者

[root@oldboyedu42-lnb oldboy]# egrep 'linux或oldboy' re.txt

[root@oldboyedu42-lnb oldboy]# egrep 'linux|oldboy' re.txt

I am oldboy teacher!

I teach linux.

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

#### 3、 () 变成一个整体 反向引用 后向引用

#先乘除后加减，有括号的先算括号里面的

[root@oldboyedu42-lnb oldboy]# egrep 'oldbo|ey' re.txt

I am oldboy teacher!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my god ,i am not oldbey,but OLDBOY!

[root@oldboyedu42-lnb oldboy]# egrep 'oldb(o|e)y' re.txt

I am oldboy teacher!

my blog is [http://oldboy.blog.51cto.com](http://oldboy.blog.51cto.com/)

our size is [http://blog.oldboyedu.com](http://blog.oldboyedu.com/)

my god ,i am not oldbey,but OLDBOY!

#反向引用 后向引用 sed

##先通过()把你想要的内容保护起来，然后再使用

echo 123456|xxxxxxx

<123456>

[root@oldboyedu42-lnb oldboy]# echo 123456

123456

[root@oldboyedu42-lnb oldboy]# echo 123456|sed -r 's#(.\*)#\1#g'

123456

[root@oldboyedu42-lnb oldboy]# echo 123456|sed -r 's#(.\*)#<\1>#g'

<123456>

echo 123456

34

[root@oldboyedu42-lnb oldboy]# echo 123456

123456

[root@oldboyedu42-lnb oldboy]# echo 123456|sed -r 's#12(34)56#\1#g'

34

[root@oldboyedu42-lnb oldboy]# echo 123456|sed -r 's#(1)2(34)5(6)#\2#g'

34

[root@oldboyedu42-lnb oldboy]# echo 123456|sed -r 's#(.).(..)..#\2#g'

34

小结:

1.()小括号 []中括号 {}大括号 花括号

2.后向引用（sed）

#### 4、 {} 花括号

0{n,m} 前一个字符连续出现至少n次，最多m次 >=n && <=m

0{n} 前一个字符连续出现n次 ==n

0{n,} 前一个字符连续出现至少n次 >=n

0{,m} 前一个字符连续出现最多m次 <=m

[root@oldboyedu42-lnb oldboy]# egrep '0{1,4}' re.txt

my qq is 49000448

not 4900000448.

[root@oldboyedu42-lnb oldboy]# egrep '0{1,4}' re.txt -o

000

0000

0

[root@oldboyedu42-lnb oldboy]# egrep '0{2,4}' re.txt

my qq is 49000448

not 4900000448.

[root@oldboyedu42-lnb oldboy]# egrep '0{2,4}' re.txt -o

000

0000

#取出文件中连续出现8次到10次的数字

[root@oldboyedu42-lnb oldboy]# #取出文件中连续出现8次到10次的数字

[root@oldboyedu42-lnb oldboy]# egrep '[0-9]{8,10}' re.txt

my qq is 49000448

not 4900000448.

[root@oldboyedu42-lnb oldboy]# egrep '^[0-9]{17}[0-9X]$' id.txt

440304199604012792

130528197108126121

342923198310042132

61242619860416291X

330900199806382320

330900199806382320

654126197703092303

### 三、取出ip地址(ifconfig ip）

#### 方法1-sed

[root@oldboyedu42-lnb oldboy]# ifconfig eth0|sed -n '2p'

inet addr:10.0.0.200 Bcast:10.0.0.255 Mask:255.255.255.0

[root@oldboyedu42-lnb oldboy]# ifconfig eth0|sed -n '2p'|sed 's#^.\*:##g'

255.255.255.0

[root@oldboyedu42-lnb oldboy]# ifconfig eth0|sed -n '2p'|sed 's#^.\*addr:##g'

10.0.0.200 Bcast:10.0.0.255 Mask:255.255.255.0

[root@oldboyedu42-lnb oldboy]# ifconfig eth0|sed -n '2p'|sed 's#^.\*addr:##g'|sed 's# Bc.\*$##g'

10.0.0.200

#### #方法2-sed-后向引用

[root@oldboyedu42-lnb oldboy]# ifconfig eth0|sed -n '2p'

inet addr:10.0.0.200 Bcast:10.0.0.255 Mask:255.255.255.0

[root@oldboyedu42-lnb oldboy]# ifconfig eth0|sed -n '2p'|sed -r 's#^.\*dr:(.\*) Bc.\*$#\1#g'

10.0.0.200

#### #方法3-awk-指定多个分隔符

[root@oldboyedu42-lnb oldboy]# ifconfig eth0|awk 'NR==2'

inet addr:10.0.0.200 Bcast:10.0.0.255 Mask:255.255.255.0

[root@oldboyedu42-lnb oldboy]# ifconfig eth0|awk 'NR==2'|awk -F "[ :]+" '{print $4}'

10.0.0.200

[root@oldboyedu42-lnb oldboy]# ifconfig eth0|awk 'NR==2'|egrep '[ :]'

inet addr:10.0.0.200 Bcast:10.0.0.255 Mask:255.255.255.0

[root@oldboyedu42-lnb oldboy]# ifconfig eth0|awk 'NR==2'|egrep '[ :]+'

inet addr:10.0.0.200 Bcast:10.0.0.255 Mask:255.255.255.0

#### #方法4-awk-'条件{命令}'

[root@oldboyedu42-lnb ~]# ifconfig eth0|awk 'NR==2'

inet addr:10.0.0.200 Bcast:10.0.0.255 Mask:255.255.255.0

[root@oldboyedu42-lnb ~]# ifconfig eth0|awk 'NR==2'|awk '{print $2}'

addr:10.0.0.200

[root@oldboyedu42-lnb ~]# #awk '找谁{干啥}'

[root@oldboyedu42-lnb ~]# #awk '条件{命令}'

[root@oldboyedu42-lnb ~]# ifconfig eth0|awk 'NR==2{print $2}'

addr:10.0.0.200

[root@oldboyedu42-lnb ~]# #错误 ifconfig eth0|awk -F"[ :]+" 'NR==2{print $2}'

[root@oldboyedu42-lnb ~]# ifconfig eth0|awk -F "[ :]+" 'NR==2{print $4}'

10.0.0.200

#### #方法5-sed

sed -nr '2s#^.\*dr:(.\*) Bc.\*$#\1#gp'

sed -n '2s#inet#oldboy#gp'

[root@oldboyedu42-lnb ~]# ifconfig eth0|sed -n '2s#inet#oldboy#gp'

oldboy addr:10.0.0.200 Bcast:10.0.0.255 Mask:255.255.255.0

老男孩IT教育出品-sed命令反向引用取出网卡ip地址详解

<https://www.processon.com/view/link/59fa9baae4b0f84f8975eefe>

#### #方法6

[root@oldboyedu42-lnb ~]# ip a s eth0 |awk 'NR==3'

inet 10.0.0.200/24 brd 10.0.0.255 scope global eth0

[root@oldboyedu42-lnb ~]# ip a s eth0 |awk 'NR==3'|awk -F"[ /]+" '{print $3}'

10.0.0.200

#### #方法7

[root@oldboyedu42-lnb ~]# ip a s eth0 |sed -n '3p'

inet 10.0.0.200/24 brd 10.0.0.255 scope global eth0

[root@oldboyedu42-lnb ~]# ip a s eth0 |sed -n '3p'|sed -r 's#^.\* (.\*)/##g'

24 brd 10.0.0.255 scope global eth0

[root@oldboyedu42-lnb ~]# ip a s eth0 |sed -n '3p'|sed -r 's#^.\* ([0-9.]+)/.\*$#\1#g'

10.0.0.200

### 四、取出文件权限

#### #方法1

[root@oldboyedu42-lnb ~]# stat /etc/hosts |awk -F "[(/]" 'NR==4'

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

[root@oldboyedu42-lnb ~]# stat /etc/hosts |awk -F "[(/]" 'NR==4{print $2}'

0644

#### #方法2-sed

[root@oldboyedu42-lnb ~]# stat /etc/hosts |sed -n '4p'

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

[root@oldboyedu42-lnb ~]# stat /etc/hosts |sed -n '4p'|sed -r 's#^.\*\(0.\*/##g'

root\)

[root@oldboyedu42-lnb ~]# stat /etc/hosts |sed -n '4p'|sed -r 's#^.\*\(0(.\*)/-##g'

rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

[root@oldboyedu42-lnb ~]# stat /etc/hosts |sed -n '4p'|sed -r 's#^.\*\(0(.\*)/-.\*$##g'

[root@oldboyedu42-lnb ~]# stat /etc/hosts |sed -n '4p'|sed -r 's#^.\*\(0(.\*)/-.\*$#\1#g'

644

<https://www.processon.com/view/link/59fbc9c0e4b0f84f89765231>

#### #方法3

[root@oldboyedu42-lnb ~]# stat /etc/hosts |awk 'NR==4'|sed -r 's#^.\*\(0|/.\*$##g'

644

#### #方法4-stat

[root@oldboyedu42-lnb ~]# stat -c%a /etc/hosts

644

[root@oldboyedu42-lnb ~]# #命令的结果中 有你想要的