

正则表达式 Regular expression

原创

GeorgeKai

2017-12-26 20:28:54

评论(0)

318人阅读

1. lsof list open file 显示被打开的文件
例: `lsof | grep delete` 找出进程调用不为0的程序
2. stat 显示文件的状态信息
3. du 统计目录的大小
-sh 带单位显示, 统计目录总大小
`du -sh /*`

第1章 本章正题: 正则表达式

1.1 正则表达式注意事项:

1. 正则表达式按照行为单位处理
2. 正则神坑---一定要注意中文符号
3. grep / egrep 过滤出来的内容加上别名, 将下面的命令写入/etc/bashrc尾部, source /etc/bashrc 生效。
`alias grep='grep --color=auto'`
`alias egrep='egrep --color=auto'`

1.2 基础正则

1.2.1 常用符号: ^ \$ ^\$. * .* [] [^]

1. ^ 以什么什么开头的行

例: `[root@georgekai oldboy]# grep '^m' oldboy.txt`
my blog is http://oldboy.blog.51cto.com
my qq is 49000448
my god , i am not oldbey, but OLDBOY!

2. \$ 以什么什么结尾的行

`[root@georgekai oldboy]# grep 'm$' oldboy.txt`
my blog is http://oldboy.blog.51cto.com
our size is http://blog.oldboyedu.com

注: 1. 注意后面的空格, 可以用 `cat -A oldboy.txt` 会在每一行的后面加上\$标记。
也可以用 `grep ' ' oldboy.txt`, 然后用光标全选, 就可以看的出后面的空格。

2. cat -A 在每一行尾部加上\$符号, 一般用于排除空格结尾

3. ^\$ 显示文件中的空行, 这一行什么符号都没有, 空格也算一个行号

`[root@georgekai oldboy]# grep -n '^$' oldboy.txt`
3:
8:

注: 1. -n 显示行号

2. -v 排除 如 `grep -v '^$'` 表示排除空行

3. 空格也是一个符号

4. . 显示文件中任意“一个”字符, 包含空格

`[root@georgekai oldboy]# grep -o '.' oldboy.txt`
I

a
m

注: -o 显示grep命令每一次找到了什么显示什么, 相当于执行过程
每次找到的一个条件, 都会用一行来显示, 包括空格

5. * 表示前一个字符连续出现0次或0次以上

```
[root@georgekai oldboy]# grep '0*' oldboy.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!
```

注：1. 连续出现0次，表示没有出现，整个文本都会显示出来

6. .* 表示任意、所有，包含空行（贪婪性）

```
[root@georgekai oldboy]# grep '^.*o' oldboy.txt
I am oldboy teacher!
I like badminton ball ,billiard ball and chinese chess!
my blog is http://oldboy.blog.51cto.com
our size is http://blog.oldboyedu.com
not 4900000448.
my god ,i am not oldbey,but OLDBOY!
```

注：1. .* 所有符号，连续出现的字符，有多少匹配多少

2. 正则中，所有符号 或 连续出现 会表现出贪婪性

7. 找出文件中以m开头并且以m结尾的行

```
[root@georgekai oldboy]# grep '^m.*m$' oldboy.txt
my blog is http://oldboy.blog.51cto.com
```

注：注意尾部如果有空格，也是找不出来的

小结：.*

1. 所以符号 任何东西
2. 贪婪性
3. 并且

8. 找出文件中以 . 结尾的行

```
[root@georgekai oldboy]# grep '\.$' oldboy.txt
I teach linux.
not 4900000448.
```

注：1. 表示任何一个字符，所以需要撬棍来使它成为普通字符

2. \ 撬棍表示转义

10. [] [abc] 找出包含a或b或c的行

```
[root@georgekai oldboy]# grep '[abc]' oldboy.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 god ,i am not oldbey,but OLDBOY!
```

注：1. [abc]中括号每次匹配一个符号，可以用-o查看

2. []里面的字符没有特殊含义，就是一个普通字符

例1：以m或n或o开头的行

```
[root@georgekai oldboy]# grep '^[mno]' oldboy.txt
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!
```

例2：找出带有a-z、A-Z、0-9的字符的行

```
[root@georgekai oldboy]# grep '[a-zA-Z0-9]' oldboy.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
```

例3:找出文件中以小写字母开头并且以.或!结尾的行

```
[root@georgekai oldboy]# grep '^[a-z].*[.!]$' oldboy.txt
not 4900000448.
my god , i am not oldbey, but OLDBOY!
```

例4:找出文件中包含a-z和A-Z的行

```
[root@georgekai oldboy]# grep '[a-Z]' oldboy.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!
```

11. [^] [^abc] 找出不包含a或b或c的行，排除的意思

```
[root@georgekai oldboy]# grep '[^abc]' oldboy.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!
```

注：1. [^] 中括号里面带 ^ 表示排除 ^ 后面的每一个字符
2. ^ 号在中括号中只有放在第一位才表示排除

例1：找出文件中不是以m或n开头的行

方法一：[root@georgekai oldboy]# grep '^[^mn]' oldboy.txt

```
I am oldboy teacher!
I teach linux.
I like badminton ball ,billiard ball and chinese chess!
our size is http://blog.oldboyedu.com
```

方法二：[root@georgekai oldboy]# grep -v '^[mn]' oldboy.txt

```
I am oldboy teacher!
I teach linux.
```

```
I like badminton ball ,billiard ball and chinese chess!
our size is http://blog.oldboyedu.com
```

1.3 .tr 替换 （不能改变文件内容）

例1：将文件中回车替换成空格

```
[root@georgekai oldboy]# tr '\n' ' ' < oldboy.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@georgekai oldboy]#
```

注：1. tr 是一对一的替换，tr 要配合 < 使用（指定从那个文件中替换）
2. sed 的阉割版
3. tr 改不了文件内容

例2：将小写字母全部替换成大写字母

```
[root@georgekai oldboy]# tr 'a-z' 'A-Z' <oldboy.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!

注: 1. **tr** 是一对一的替换, 如**a** 替换成了**A**, **b**替换**B**

小结正则--BRE:

1. **^**
2. **\$**
3. **^\$**
4. **.**
5. *****
6. **.***
7. **[] [a-z] [A-Z] [0-9]**

1.4 扩展正则--ERE

1.4.1 常用符号: + | ()

1. **+** 前一个字符连续出现1次或1次以上

```
[root@georgekai oldboy]# egrep '0+' oldboy.txt
```

my qq is 49000448

not 4900000448.

注: 1. 连续出现, 注意它的贪婪性

2. **+** 和 **[]** 一起配合使用

例1: 取出文件中连续出现的小写字母的

```
[root@georgekai oldboy]# egrep '[a-z]+' oldboy.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!

2. **|** 或者, 取出一个字符串或者另一个字符串

```
[root@georgekai oldboy]# egrep '3306|1521' /etc/services
```

```
mysql          3306/tcp          # MySQL
```

```
mysql          3306/udp          # MySQL
```

```
ncube-lm       1521/tcp          # nCube License Manager
```

```
ncube-lm       1521/udp          # nCube License Manager
```

3. **()** 表示一个整体, 反向引用/后向引用, 一般在**sed**中用

例1: 查找出**oldboy**或**oldbey**

```
[root@georgekai oldboy]# egrep 'oldb(o|e)y' oldboy.txt
```

I am oldboy teacher!

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

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

my god , i am not oldbey, but OLDBOY!

例2: 在**123456**中取出**34**

```
[root@georgekai oldboy]# echo 123456 | sed -r 's#(34)#<\1>#g'
```

12<34>56

注: 1. **<\1>** 表示第1个小括号的内容, **<\2>** 表示第二个

2. **-r** **sed**的扩展正则用法

例3：在 **georgekai** 中取出 **kai**

```
[root@georgekai oldboy]# echo georgekai | sed -r 's#(.....)(...)#<\2>#g'  
<kai>
```

注：1 一个 .（点）表示一个前面的一个字符

小伙伴们可以关注我的微信公众号：linux运维菜鸟之旅，更新比51cto慢一些，不过要方便许多



关注“中国电信天津网厅”公众号，首次绑定可免费领2G流量，为你的学习提供流量！



版权声明：原创作品，如需转载，请注明出处。否则将追究法律责任
