Linux 正则表达式范例

创建日期：
2017-05-29

正则表达式是一种字符模式，用于在查找过程中匹配制定的字符。

元字符通常在 Linux 中分为两类：

1. Shell 元字符，由 Linux Shell 进行解析；
2. 正则表达式元字符，由 vi/grep/sed/awk 等文本处理工具进行解析；

正则表达式一般以文本行进行处理，在进行下面实例之前，先为 grep 命令设置--color 参数：

```
$ alias grep='grep --color=auto'
```

这样每次过滤出来的字符串都会带色彩了。

在开始之前还需要做一件事情，就是创建一个测试用的 re-file 文件，内容如下：

```
$ cat re-file
I had a lovely time on our little picnic.
Lovers were all around us. It is springtime. Oh
love, how much I adore you. Do you know
the extent of my love? Oh, by the way, I think
I lost my gloves somewhere out in that field of
clover. Did you see them?  I can only hope love.
is forever. I live for you. It's hard to get back in the
groove.
```

文件内容摘录自<<UNIX/SHELL 范例精解第四版>>

```
$ cat linux.txt
Linux is a good
god assdxw bcvnbvbjk
greatttttt  wexcvxc
operaaaating  dhfghfvx
gooodfs awrerdxxhkl
gdsystem awxxxx
glad
good
```

正则表达式元字符

元字符	功能
^	以什么开头
$	以什么结尾
.	匹配一个字符
*	匹配 0 个或多个
[]	匹配集合中的

元字符	功能	
[x-y]	匹配集合范围内的	
[^]	匹配不在集合中的	
\	转义	'love\.'

- 特殊的元字符

元字符	功能	实例	怎么匹配
\<	以什么开头	'\<love'	匹配以 love 开头的所有行
\>	以什么结尾	'love\>'	匹配 love 结尾的所有行
\(..\)	标签匹配以后使用的字符	'\(love\)able \1er'	用位置 \1\2 引导前面做好的标签，最大支持 9 个
x\{m\} or x\{m,\} or x\{m,n\}	重复字符 x，m 次，至少 m 次，至少 m 且不超过 n 次	o\{5,10\}	o 字符重复 5 到 10 次的行

- 扩展的正则表达式

元字符	说明	
+	重复前一个字符一个或一个以上	
?	0 个或者一个字符	
、	、	表示或，查找多个字符串
()	分组过滤匹配	

实操

- 匹配以 love 开头的所有行

```
$ grep '^love' re-file
love, how much I adore you. Do you know
```

- 匹配 love 结尾的所有行

```
$ grep 'love$' re-file
```

```
clover. Did you see them?  I can only hope love.
```

- 匹配以 l 开头，中间包含两个字符，结尾是 e 的所有行

```
$ grep 'l..e' re-file
I had a lovely time on our little picnic.
love, how much I adore you. Do you know
the extent of my love? Oh, by the way, I think
I lost my gloves somewhere out in that field of
clover. Did you see them?  I can only hope love.
is forever. I live for you. It's hard to get back in the
```

- 匹配 0 个或多个空行，后面是 love 的字符

```
$ grep ' *love' re-file
I had a lovely time on our little picnic.
love, how much I adore you. Do you know
the extent of my love? Oh, by the way, I think
I lost my gloves somewhere out in that field of
clover. Did you see them?  I can only hope love.
```

- 匹配 love 或 Love

```
$ grep '[Ll]ove' re-file  # 对 l 不区分大小写
I had a lovely time on our little picnic.
Lovers were all around us. It is springtime. Oh
love, how much I adore you. Do you know
the extent of my love? Oh, by the way, I think
I lost my gloves somewhere out in that field of
clover. Did you see them?  I can only hope love.
```

- 匹配 A-Z 的字母，其次是 ove

```
$ grep '[A-Z]ove' re-file
Lovers were all around us. It is springtime. Oh
```

- 匹配不在 A-Z 范围内的任何字符行，所有的小写字符

```
$ grep '[^A-Z]' re-file
I had a lovely time on our little picnic.
Lovers were all around us. It is springtime. Oh
love, how much I adore you. Do you know
the extent of my love? Oh, by the way, I think
I lost my gloves somewhere out in that field of
clover. Did you see them?  I can only hope love.
is forever. I live for you. It's hard to get back in the
groove.
```

- 匹配 love.

```
$ grep 'love\.' re-file
clover. Did you see them?  I can only hope love.
```

- 匹配空格

```
$ grep '^$' re-file
```

- 匹配任意字符

```
$ grep '.*' re-file
I had a lovely time on our little picnic.
Lovers were all around us. It is springtime. Oh
love, how much I adore you. Do you know
the extent of my love? Oh, by the way, I think
I lost my gloves somewhere out in that field of
clover. Did you see them?  I can only hope love.
is forever. I live for you. It's hard to get back in the
groove.
```

- 前面 o 字符重复 2 到 4 次

```
$ grep 'o\{2,4\}' re-file
groove.
```

- 重复 o 字符至少 2 次

```
$ grep 'o\{2,\}' re-file
groove.
```

- 重复 0 字符最多 2 次

```
$ grep 'o\{,2\}' re-file
I had a lovely time on our little picnic.
Lovers were all around us. It is springtime. Oh
love, how much I adore you. Do you know
the extent of my love? Oh, by the way, I think
I lost my gloves somewhere out in that field of
clover. Did you see them?  I can only hope love.
is forever. I live for you. It's hard to get back in the
groove.
```

- 重复前一个字符一个或一个以

```
$ egrep "go+d" linux.txt
Linux is a good
god assdxw bcvnbvbjk
gooodfs awrerdxxhkl
good
```

- 0 个或者一个字符

```
ansheng@Ubuntu:/tmp$ egrep "go?d" linux.txt
god assdxw bcvnbvbjk
gdsystem awxxxx
```

- 或，查找多个字符串

```
$ egrep "gd|good" linux.txt
Linux is a good
gdsystem awxxxx
good
```

- 分组过滤匹配

```
$ egrep "g(la|oo)d" linux.txt
Linux is a good
glad
good
```

Python 大陆

微信扫描二维码，关注我的公