linux正则表达式详解（一） -通配符与基本正则表达式

运维朱工

我们在很多地方都会用到通配符和正则表达式来实现我们的日常操作，提高我们的工作效率。但是很多新伙伴，往往容易将他们弄混。

首先我们需要知道通配符和正则表达式的**使用场景**：

通配符也叫文件名替换，它主要是作用于匹配文件名，常用命令是ls、find、cp、mv；

正则表达式主要是作用于匹配文件中的字符串,常用命令命令是grep、awk、sed。

通配符日常使用：

通配符	含义	实例
*	匹配0个或者多个字符	f* 以f开头的任意文件。 b*.txt 以b开头的，中间有任意字符，并以.txt结尾的任意文件。
[ ]	匹配括号中的单个字符	[abc]* 以a、b、c任意一个字符开头的文件
?	匹配任意单个字符	f?.txt 以f开头，后面跟一个字符的，以.txt结尾的任意文件
[!]	匹配不在括号中的任意单个字符	[!abc]* 不以abc开头的任意文件
[a-z]	匹配a到z中的任意单个字符,表示范围,只能用于查找文件，不能用于创建文件	[a-z]* 以a到z开头的任意文件
{a,b,z}/{a..z}	以逗号分隔时，表示单独字符，以两个点号分隔时，表示连续字符。可用于查找和创建文件。	{a,b,z}* 以abz任意一个字符开头的文件; {a..z}* 以a到z任意一个字符开头的文件

*** 匹配0或者多个字符**

实例：

```
#匹配前：
[root@localhost test]# ls
f  f-1.txt  f-2.txt  f-3.txt  ff

#匹配后：
[root@localhost test]# ls f*
f  f-1.txt  f-2.txt  f-3.txt  ff
```

[] 匹配括号中的单个字符

实例：

```
#匹配前：
[root@localhost test]# ls
f  f-1.txt  f-2.txt  f-3.txt  ff

#匹配后：
[root@localhost test]# ls f-[12].txt
f-1.txt  f-2.txt
```

这里要注意，因为匹配的单个字符，所以后面.txt也要补全，否则会报文件找不到。

? 匹配任意单个字符

实例：

```
#匹配前：
[root@localhost test]# ls
f  f-1.txt  f-2.txt  f-3.txt  ff

#匹配后：
[root@localhost test]# ls f-?.txt
f-1.txt  f-2.txt  f-3.txt
```

[!] 匹配不在括号中的任意单个字符

实例：

```
#匹配前：
[root@localhost test]# ls
f  f-1.txt  f-2.txt  f-3.txt  ff

#匹配后：
[root@localhost test]# ls f-[!12].txt
f-3.txt
```

[a-z] 匹配a到z中的任意单个字符,表示范围,只能用于查找文件，不能创建

实例：

```
#匹配前：
[root@localhost test]# ls
f  f-12.txt  f-1.txt  f-2.txt  f-3.txt

#匹配后：
[root@localhost test]# ls f-[1-2].txt
f-1.txt  f-2.txt
```

{a,b,c}/{a..z} 以逗号分隔时，表示单独字符，以两个点号分隔时，表示连续字符。可用于查找和创建文件。

实例：

```
a.txt  d.txt  f-12.txt  f-3.txt  g.txt  j.txt  m.txt  p.txt  s.txt        u.txt  x.txt
b.txt  e.txt  f-1.txt   ff       h.txt  k.txt  n.txt  q.txt  t-[1-9].txt  v.txt  y.txt
c.txt  f      f-2.txt   f.txt    i.txt  l.txt  o.txt  r.txt  t.txt        w.txt  z.txt
```

匹配后：

```
[root@localhost test]# ls {a..z}.txt
a.txt  c.txt  e.txt  g.txt  i.txt  k.txt  m.txt  o.txt  q.txt  s.txt  u.txt  w.txt  y.
b.txt  d.txt  f.txt  h.txt  j.txt  l.txt  n.txt  p.txt  r.txt  t.txt  v.txt  x.txt  z.
```

匹配后：

```
[root@localhost test]# ls {a,b,c}.txt
a.txt  b.txt  c.txt
```

ps:当用逗号分隔时，查找效果和[abc]是一样的。

正则表达式日常使用：

元字符	含义
*	匹配前一个字符0次或者多次
.	匹配除了换行符以外的任意单个字符
^	锚定行首
$	锚定行尾
[abz]	匹配括号中指定的任意单个字符
[^abz]	匹配括号中以外的任意单个字符
\	转义符，取消元字符的特殊含义
\{n\}	表示前面的字符出现几次,如果不想使用\，可以使用egrep
\{n,\}	至少匹配前面的字符n次
\{n,m\}	至少匹配前面的字符n次，但是不得多于m次
-	连字符，[a-z]表示a到z；[a-Z] 表示a到z, A到Z的连续范围字符
特别需要注意，因为有些元字符在shell中有特殊含义（通配符），所以要用引号将匹配项括起来，避免不必要的shell扩展。	

ps:做个纠正，**"{}"是扩展正则**，所以grep不能直接使用，需要加-E或者使用egrep。

*　　匹配前一个字符0次或者多次

实例：

```
[root@localhost test]# grep "a*" f
abc
aabc
ab
abbb
babb
aaab
```

匹配a 0次或者多次

.　　匹配除了换行符以外的任意单个字符

```
[root@localhost test]# cat f
abc
aabc
ab
abbb
babb
aaab
[root@localhost test]# grep "ab." f
abc
aabc
abbb
babb
```

ab. 是一个整体，必须是连续的3个字符，否则不符合

^ 锚定行首

实例：

```
[root@localhost test]# cat f
abc
aabc
ab
abbb
babb
aaab
[root@localhost test]# grep "^ab." f
abc
abbb
```

必须是要以ab. 开头的字符串，才符合

$ 锚定行尾

实例：

```
[root@localhost test]# cat f
abc
aabc
ab
abbb
babb
aaab
[root@localhost test]# grep "c$" f
abc
aabc
```

匹配以c结尾的行

[] 匹配括号中指定的任意单个字符

实例：

```
aabc
ab
abbb
babb
aaab
[root@localhost test]# grep "[cdg]" f
abc
aabc
```

匹配c、d、g单个字符

[^] 匹配括号中以外的任意单个字符

实例：

```
[root@localhost test]# cat f
abc
aabc
ab
abbb
babb
aaab
[root@localhost test]# grep "[^cdg]" f
abc
aabc
ab
abbb
babb
aaab
```

匹配除了括号中的字符，记得双引号，否则后果很严重

\ 转义符，取消元字符的特殊含义

实例：

```
[root@localhost test]# grep "a*" f
abc
aabc
ab
abbb
babb
[root@localhost test]# grep "a\*" f
[root@localhost test]#
```

使用\，*变成了自身的字符，不在有特殊含义

\{n\} 表示前面的字符出现几次,如果不想使用\，可以使用egrep

实例：

```
[root@localhost test]# cat f
abc
aabc
ab
abbb
babb
[root@localhost test]# egrep "a{2}" f
aabc
[root@localhost test]# grep "a\{2\}" f
aabc
```

匹配a两次，这里可以使用扩展匹配-E，或者使用egrep

实例：

```
[root@localhost test]# egrep "a{1,}" f
abc
aabc
ab
abbb
babb
```

a至少匹配一次

\{n,m\}	至少匹配前面的字符n次，但是不得多于m次

实例：

```
[root@localhost test]# cat f
abc
aabc
ab
abbb
babb
aaab
[root@localhost test]# egrep "a{1,2}b" f
abc
aabc
ab
abbb
babb
aaab
```

ab为一个整体，但是a可以匹配1次或者2次

ps:一定要注意，在**通配符**中，{}可以用来表示连续的字符，但是在**正则表达式中**，是表示前面字符出现的次数。

今天就先说到这，后面我们继续探讨拓展正则表达式。

获取文章更新，以及常用软件，可以关注公众号： **运维朱工**

#	name
1001	PDF全能工具箱
1002	everything/飞速查找文件
1003	Topaz Gigapixel/AI 智能无损放大图片
1004	天若OCR开源版5.0/截图识字，翻译
1005	eclipse编辑器/开发专用
1006	office 2019专业版
1007	visio网络拓扑工具
1008	Packet tracer 7/网络仿真学习工具
1009	win 10 永久激活工具
1010	win 10 专业版镜像
1011	win7 镜像
1012	TrafficMonitor/显示网速，ip
1013	Quicker/快捷面板

编辑于 2021-11-26 03:14

Linux 运维 Linux 正则表达式