

本笔记为北京千锋 V2017.1 《03 Linux Shell 脚本自动化编程实战》，笔记中所涉及到的项目均基于 Centos7u3 x86_64 或 Centos6u8 环境。

千锋所有学员均可自由使用和转载该笔记，为了尊重作者的辛劳，敬请注明出处！

正则表达式 RE

=====

重要的文本处理工具：vim、sed、awk、grep

mysql、oracle、php、python ...

一、什么是正则表达式？

正则表达式（regular expression, RE）是一种字符模式，用于在查找过程中匹配指定的字符。在大多数程序里，正则表达式都被置于两个正斜杠之间；例如 `/l[oe]ve/` 就是由正斜杠界定的正则表达式，

它将匹配被查找的行中任何位置出现的相同模式。在正则表达式中，元字符是最重要的概念。

匹配数字: `^[0-9]+$` 123 456 5y7

匹配 Mail: `[a-z0-9_]+@[a-z0-9]+\.[a-z]+` yangsheng131420@126.com

匹配 IP: `[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}`

或

`[:digit:]{1,3}\.[[:digit:]]{1,3}\.[[:digit:]]{1,3}\.[[:digit:]]{1,3}`

```
[root@tianyun scripts]# egrep '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'  
/etc/sysconfig/network-scripts/ifcfg-eth0
```

IPADDR=172.16.100.1

NETMASK=255.255.255.0

GATEWAY=172.16.100.254

```
[root@tianyun scripts]# egrep '[:digit:]{1,3}\.[[:digit:]]{1,3}\.[[:digit:]]{1,3}\.[[:digit:]]{1,3}'  
/etc/sysconfig/network-scripts/ifcfg-eth0
```

IPADDR=172.16.100.1

NETMASK=255.255.255.0

GATEWAY=172.16.100.254

二、元字符

定义：元字符是这样一类字符，它们表达的是不同于字面本身的含义

shell 元字符(也称为通配符) 由 shell 来解析，如 `rm -rf *.pdf`，元字符 `*` Shell 将其解析为任意多个字符

正则表达式元字符 由各种执行模式匹配操作的程序来解析，比如 vi、grep、sed、awk、python

```
[root@tianyun ~]# rm -rf *.pdf
```

```
[root@tianyun ~]# grep 'abc*' /etc/passwd
```

abrt:x:173:173::/etc/abrt:/sbin/nologin

vim 示例:

:1,\$ s/tom/David/g //如 tom、anatomy、tomatoes 及 tomorrow 中的“tom”被替换了, 而 Tom 确没被替换

:1,\$ s/\<[Tt]om\>/David/g

1. 正则表达式元字符:

===基本正则表达式元字符

元字符 功能 示例

=====

^ 行首定位符 ^love

\$ 行尾定位符 love\$

. 匹配单个字符 l.e

* 匹配前导符 0 到多次 ab*love

. * 任意多个字符

[] 匹配指定范围内的一个字符 [IL]ove

[-] 匹配指定范围内的一个字符 [a-z0-9]ove

[^] 匹配不在指定组内的字符 [^a-z0-9]ove

\ 用来转义元字符 love\.

\< 词首定位符 \<love

\> 词尾定位符 love\>

\(.\) 匹配稍后使用的字符的标签 :% s/172.16.130.1/172.16.130.5/

:% s/\(172.16.130.\)1/\15/

:% s/\(172.\)\(16.\)\(130.\)1/\1\2\35/

:3,9 s/\(.*/\)/#\1/

x\{m\} 字符 x 重复出现 m 次 o\{5\}

x\{m,\} 字符 x 重复出现 m 次以上 o\{5,\}

x\{m,n\} 字符 x 重复出现 m 到 n 次 o\{5,10\}

===扩展正则表达式元字符

+ 匹配一个或多个前导字符 [a-z]+ove

? 匹配零个或一个前导字符 lo?ve

a|b 匹配 a 或 b love|hate

() 组字符 loveable|rs love(able|rs) ov+ (ov)+

(..)(..)\1\2 标签匹配字符 (love)able\1er

x{m} 字符 x 重复 m 次 o{5}

x{m,} 字符 x 重复至少 m 次 o{5,}

x{m,n} 字符 x 重复 m 到 n 次 o{5,10}

2. POSIX 字符类:

表达式 功能 示例

[[:alnum:]] 字母与数字字符 [[:alnum:]]+
[[:alpha:]] 字母字符(包括大小写字母) [[:alpha:]]{4}
[[:blank:]] 空格与制表符 [[:blank:]]*
[[:digit:]] 数字字母 [[:digit:]]?
[[:lower:]] 小写字母 [[:lower:]]{5,}
[[:upper:]] 大写字母 [[:upper:]]+
[[:punct:]] 标点符号 [[:punct:]]
[[:space:]] 包括换行符, 回车等在内的所有空白[[:space:]]+

三、正则匹配示例: vim

```
/love/  
/^love/  
/love$/  
/l.ve/  
/lo*ve/  
/[Ll]ove/  
/love[a-z]/  
/love[^a-zA-Z0-9]/  
  
/.*/  
/^$/  
/^[A-Z]..$/  
/^[A-Z][a-z ]*3[0-5]/  
/[a-z]*\./  
/^[|][A-Z][a-z][a-z]$/  
/^[A-Za-z]*[^\,][A-Za-z]*$/  
/\<fourth\>/  
/\<f.*th\>/  
/5{2}2{3}\./
```

空行

```
/^$/  
/^[|]\t]*$/
```

注释行

```
/^#/  
/^[| \t]*#/
```

```
:1,$ s/\([Oo]ccur\)ence/\1rence/  
:1,$ s/\(square\) and \(fair\) /\2 and \1/  
=====
```