

## 向架构师的方向，缓慢前行

博客园

首页

新随笔

联系

订阅

管理

### 公告

昵称: [goodbay说拜拜](#)

园龄: 1年2个月

粉丝: 2

关注: 1

关注成功

< 2023年6月 >						
日	一	二	三	四	五	六
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

### 常用链接

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

[我的标签](#)

### 随笔分类 (176)

[实战案例\(24\)](#)

[Linux-系统应用-命令\(29\)](#)

[Linux练习\(9\)](#)

[shell脚本\(3\)](#)

[shell脚本编写\(7\)](#)

[DNS 服务模块\(3\)](#)

[防火墙模块\(3\)](#)

[VPN配置流程\(3\)](#)

[MySQL\(24\)](#)

[运维自动化之ANSIBLE\(4\)](#)

[LAMP\(3\)](#)

[日志服务管理\(2\)](#)

[网络文件共享\(6\)](#)

[高可用集群\(4\)](#)

[NGINX\(13\)](#)

[tomcat\(10\)](#)

随笔 - 152 文章 - 0 评论 - 0 阅读 - 23477

## Linux正则表达式

### 正则表达式

REGEXP: Regular Expressions, 由一类特殊字符及文本字符所编写的模式, 其中有些字符 (元字符) 不表示字符字面意义, 而表示控制或通配的功能, 类似于增强版的通配符功能, 但与通配符不同, 通配符功能是用来处理文件名, 而正则表达式是处理文本内容中字符

### 正则表达式分为两类

#### 基本正则表达式

#### 扩展正则表达式

#### 1、基本正则表达式元字符

##### 1.1 字符匹配



#### 常用

- `.` 匹配任意单个字符, 可以是一个汉字
- `[]` 匹配指定范围内的任意单个字符, 示例: `[abc]`
- `[^]` 匹配指定范围外的任意单个字符, 示例: `[^abc]`
- `[:alnum:]` 字母和数字
- `[:alpha:]` 代表任何英文大小写字符, 亦即 A-Z, a-z
- `[:lower:]` 小写字母, 示例: `[[:lower:]]`, 相当于 `[a-z]`
- `[:upper:]` 大写字母
- `[:space:]` 包括空格、制表符(水平和垂直)、换行符、回车

#### 拓展

- `[:cntrl:]` 不可打印的控制字符 (退格、删除、警铃...)
- `[:digit:]` 十进制数字
- `[:xdigit:]` 十六进制数字
- `[:graph:]` 可打印的非空白字符
- `[:print:]` 可打印字符
- `[:punct:]` 标点符号
- `[:blank:]` 空白字符 (空格和制表符)

反向代理HAproxy(5)  
redis(11)  
docker(10)  
kubernetes(1)  
zabbix监控(2)

```
\w      #匹配单词构成部分，等价于[_:alnum:]
\W      #匹配非单词构成部分，等价于[^:alnum:]
\S      #匹配任何非空白字符。等价于  [^\f\n\r\t\v]
\s      #匹配任何空白字符，包括空格、制表符、换页符；
Unicode 正则表达式会匹配全角空格符
```

### 1.2 匹配次数

用在要指定次数的字符后面，用于指定前面的字符要出现的次数

```
*          #匹配前面的字符任意次，包括0次，即：
.*        #任意长度的任意字符
\?        #匹配其前面的字符出现0次或1次，即：
\+        #匹配其前面的字符出现最少1次，即：
{n\}      #匹配前面的字符n次
\{m,n\}   #匹配前面的字符至少m次，至多n次
\{,n\}    #匹配前面的字符至多n次，<=n
\{n,\}    #匹配前面的字符至少n次
```

### 1.3 位置锚定

位置锚定可以用于定位出现的位置

```
^          #行首锚定，用于模式的最左侧
$          #行尾锚定，用于模式的最右侧
^PATTERN$ #用于模式匹配整行
^$         #空行
^[[:space:]]*$ #空白行
\< 或 \b  #词首锚定，用于单词模式的左
\> 或 \b  #词尾锚定，用于单词模式的右
\<PATTERN\> #匹配整个单词
```

#注意：单词是由字母，数字，下划线组成

### 1.4 分组其他

```
\(string1\\(string2\)\\)
```

```
\1 : string1\(string2\)
\2 : string2

a\b      #a或b
C\|cat   #C或cat
\(C\|c\)at  #Cat或cat
```

2、扩展正则表达式原子符

2.1 字符匹配

.	任意单个字符
[abcd]	指定范围的字符
[^abcd]	不在指定范围的字符
[:alnum:]	字母和数字
[:alpha:]	代表任何英文大小
[:lower:]	小写字母, 示例: [
[:upper:]	大写字母
[:blank:]	空白字符 (空格和
[:space:]	水平和垂直的空白
[:cntrl:]	不可打印的控制字
[:digit:]	十进制数字
[:xdigit:]	十六进制数字
[:graph:]	可打印的非空白字
[:print:]	可打印字符
[:punct:]	标点符号

2.2 次数匹配

*	匹配前面字符任意次
?	0或1次
+	1次或多次
{n}	匹配n次
{m, n}	至少m, 至多n次

2.3 位置锚定

^	行首
\$	行尾
\<, \b	语首
\>, \b	语尾

2.4

分组其他

( )

后向引用: \1, \2, ...

|

a|b

C|cat

(C|c)at

分组

#a或b

#C或cat

#Cat或cat

grep -E 配合扩展正则表达式使用，对文本内容过滤筛选。

<div>

>

[15:19:09 root@centos8 ~]#ifconfig ens160 |

grep netmask | grep -Eo '[0-9]{1,3}\.[0-9]

{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | head -n1

192.168.1.85

[15:19:20 root@centos8 ~]#

分类: [Linux-系统应用-命令](#)

标签: [通配符与正则表达式](#)

好文要顶

关注成功

收藏该文

goodbay说拜拜

粉丝 - 2 关注 - 1

关注成功

1

0

推荐

反对

支持成功

« 上一篇: [Linux 实现静态路由实验](#)

» 下一篇: [Linux文本处理三剑客](#)

posted @ 2022-04-21 15:19 [goodbay说拜拜](#) 阅读

(454) 评论(0) [编辑](#) [收藏](#) [举报](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

发表评论

编辑

预览

B



</>

“ ”



支持 Markdown

//

自动补全

提交评论

退出

订阅评论

我的博客

[Ctrl+Enter快捷键提交]

【推荐】中国云计算领导者：阿里云轻量应用服务器2核2G低至108元/年

【推荐】第五届金蝶云苍穹低代码开发大赛正式启动，百万奖金等你拿！

#### 编辑推荐：

- 如何将一个实例的内存二进制内容读出来？
- K8S | 核心原理分析
- 如何获取 C#程序 内核态线程栈
- 如何计算一个实例占用多少内存？
- [MAUI 程序设计] 用 Handler 实现自定义跨平台控件

#### 即构专区：

- 端午“沉浸式云旅游”怎么玩？即构助力“直播+”新场景落地
- 即构✖️叮咚课堂：行业第一套AI课堂解决方案是怎么被实现的？
- 活动回顾：Flutter实时音视频应用场景实践
- ZEGO全新语音聊天解决方案，4步搭建爆火的语音聊天室
- 【活动回顾】后疫情时代，教育行业的增量和变量