正则表达式

2019年5月30日　　莫宇剑♡符菁 16:22

正则表达式简介：

1. 正则表达式是为了处理大量的文本|字符串而定义的一套规则和方法
2. 通过正则表达式，系统管理员就可以快速过滤，替换或输出需要的字符串。
3. Linux正则表达式一般以行为单位处理。（以行为单位出来，一次处理一行）

正则表达式的用途：linux运维工作，大量过滤日志工作，化繁为简。

注意事项：正则表达式应用非常广泛，存在于各种语言中，但现在学的是Linux中的正则表达式，最常应用正则表达式的命令是grep（egrep），sed，awk。

正则表达式使用注意事项

1. linux正则表达式以行为单位处理字符串
2. 便于区别过滤出来的字符串，一定配合grep / egrep命令学习。

正则表达式与通配符的对比：

正则表达式用来在文件中匹配符合条件的字符串，如果文件中的语句中包含了查找的字符串，那么就显示整行语句。grep、awk、sed等命令可以支持正则表达式。

通配符用来匹配符合条件的文件名，通配符是完全匹配。如find . -name “abc？” 查找到4个字符的文件名其中前三个字符是abc，列出的abcd、abce等文件。

作用：文本搜索工具，根据用户指定的“模式”对目标文本逐行进行匹配检查；打印匹配到的行

模式：由正则表达式字符及文本字符所编写的过滤条件

grep 命令

语法：grep [选项] (参数)

选项：

```
grep --color=auto ：对匹配到的文本着色显示（CentOS7里默认显示）
grep -v     #  反向搜索
grep -i     #  忽略大小写
grep -n     #  显示匹配的行号
grep -c     #  统计匹配的行数，不显示搜索结果
grep -o     #  仅显示匹配到的字符串
grep -A     #  显示搜索行及其向下相邻的#行
grep -B     #  显示搜索行及其向上相邻的#行
grep -C     #  显示搜索行及其向上下相邻的#行
grep -E     #  egrep 支持扩展正则表达式
grep -e     #  支持多个模式
```

Demo1:

1、显示/proc/meminfo文件中以大小s开头的行(三种方法)

```
[root@test ~]# grep -i "^s" /proc/meminfo
# ^ 行首锚定，用于模式的最左侧
[root@test ~]# grep -E "^[Ss]" /proc/meminfo
# []  匹配指定范围内的任意单个字符，其中可用连字符（-）指的连续字符的范围
[root@test ~]# grep -E "^S|^s" /proc/meminfo
# | 或者（ 匹配|符号前或后的正则表达式）
```

2、基本的正则表达式与扩展到正则表达式之间的区别

BRE和ERE的区别仅仅是元字符的不同

基础正则BRE	扩展正则ERE
\?	?
\+	+
\{\}	{}
\(\)	()
\	

3、显示/etc/passwd文件中不以/bin/bash结尾的行

```
[root@test ~]# grep -vE "/bin/bash$" /etc/passwd  # -v 反向搜索 "\(bin/bash\)$"以bin/bash结尾的行
[root@test ~]# grep -E "/bin/bash$" /etc/passwd
```

4、显示用户rpc默认的shell程序

```
# ( ) : 匹配括号括起来的正则表达式群
[root@test ~]# cat /etc/passwd | grep -E "^(rpc)" |cut -d: -f 7
[root@test ~]# cat /etc/passwd | grep "^\(rpc\)" |cut -d: -f 1,2,3,4,5,6,7
# 取出以用户rpc开头的行。cut -d: -f 7 以：做分隔符取第七列
[root@test ~]# cat /etc/passwd | cut -d: -f 7
```

5、找出/etc/passwd中的两位或三位数

```
# []  匹配指定范围内的任意单个字符，其中可用连字符（-）指的连续字符的范围
[root@test ~]# grep  -E "[0-9]{2,3}" /etc/passwd       # 扩展的正则表达式
[root@test ~]# grep  "[0-9]\{2,3\}" /etc/passwd        # 基本的正则表达式

[root@test ~]# grep  "[0-9]\{1,\}" /etc/passwd
[root@test ~]# grep -E "[0-9]{1,}" /etc/passwd
```

6、显示CentOS7的/etc/grub2.cfg文件中，至少以一个空白字符开头的且后面存在非空白字符的行

```
[root@test ~]# grep -E "^[[:space:]]+[^[:space:]]" /etc/grub2.cfg
[root@test ~]# grep "^[[:space:]]\+[^[:space:]]" /etc/grub2.cfg
[:space:] 水平和垂直的空白字符
基本正则表达式\+：匹配其前面的字符至少1次
扩展正则表达式 +：1次或多次
```

7、找出“netstat -tan”命令的结果中以‘LISTEN’后跟任意多个空白字符结尾的行

```
[root@test ~]# netstat -tan |grep  "LISTEN[[:space:]]*$"
*   匹配前面的字符任意次，包括0次 （贪婪模式：尽可能长的匹配 ）
.   匹配任意单个字符
.*  任意长度的任意字符
```

8、显示CentOS7上所有系统用户的用户名和UID #系统用户UID小于1000

```
[root@test ~]# cat /etc/passwd |cut -d: -f1,3 |grep "[0-9]\{1,3\}"
[root@test ~]# cat /etc/passwd |cut -d: -f1,3 |grep -E "[0-9]{1,3}"
```

9、查看/etc/grub2.cfg文件中以小写字母开头的行

```
[root@test ~]# cat /etc/grub2.cfg |grep -o "^[[:lower:]]"
```

10、添加用户bash、testbash、basher、sh、nologin(其shell为/sbin/nologin),找出"/etc/passwd"用户名同"shell"名的行

```
[root@test ~]# useradd sh
[root@test ~]# useradd bash
[root@test ~]# useradd testbash
[root@test ~]# useradd -s /sbin/nologin nologin
[root@test ~]# cat /etc/passwd |grep  -E "^(.*).*/\1$"
[root@test ~]# cat /etc/passwd |grep  -E "(.*).*/\1$"
[root@test ~]# cat /etc/passwd |grep "^\(.*\).*/\1$"
```

利用分组^段落开头(.)用小括号引起来为后项引用里边的.为任意字符\b词尾锚定.*中间任意字符。发生作用的是：**/\1$中/目录分割符\1$段落结尾引用前项用小括号引起的部分（分组引用的是命令结果而非命令本身）**

11、利用df和grep，取出磁盘各分区利用率，并从大到小排序

```
# 诡异情况
[root@test ~]# df |grep -E "[0-9]{1,3}" #
```

```
[root@test ~]# df |grep -E "[0-9]{1,3}"
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/vda1       51473888 2098472  46737668   5% /
devtmpfs          930564       0    930564   0% /dev
tmpfs             941176      24    941152   1% /dev/shm
tmpfs             941176     444    940732   1% /run
tmpfs             941176       0    941176   0% /sys/fs/cgroup
tmpfs             188236       0    188236   0% /run/user/0
```

```
[root@test ~]# df |grep -E "[0-9]{1,3}%" -o |sort
[root@test ~]# df |grep -E "[0-9]{1,3}%" -o |sort -rn
[root@test ~]# df |grep -o "[0-9]\{1,3\}%"  |sort -rn
```

12、显示三个用户root、shutdown、basher的UID和默认shell

```
[root@test ~]# grep -e^"shutdown" -e^"basher" -e"^root" /etc/passwd | cut -d: -f 3,7
[root@test ~]# grep "^shutdown\|^root\|^basher" /etc/passwd        | cut -d: -f 3,7
[root@test ~]# grep -E "^shutdown|^root|^basher" /etc/passwd     | cut -d: -f 3,7
```

13、找出/etc/rc.d/init.d/functions文件中行首为某单词(包括下划线)后面跟一个小括号的行

```
[root@test ~]# grep "^[_[:alnum:]]\+()" /etc/rc.d/init.d/functions
[root@test ~]# egrep "^[_[:alnum:]]+\(\)" /etc/rc.d/init.d/functions
```

14、统计last命令中以root登录的每个IP地址登录次数

```
[root@test ~]#  last |egrep "^root" |egrep -o "([0-9]{1,3}.){3}[0-9]{1,3}"
```

{n} 匹配前面的字符n次

15、将此字符串：welcome to magedu linux 中的每个字符去重并排序，重复次数多的排到前面

```
[root@centos7 ~]# echo welcome to magedu linux |grep -o . |sort |uniq -c |sort -rn #先用"grep -o ."点表示任意单个字符，-o把匹配到的单个字符成竖列排列出来，排序，去重，再排序。
```

参考资料：

https://blog.51cto.com/12105235/2063196 #linux正则表达式

https://www.cnblogs.com/chensiqiqi/p/6285060.html #正则表达式