18. 正则三剑客 - grep

笔记本： 优秀笔记
创建时间： 2018/12/27 23:08 **更新时间：** 2018/12/28 8:49
作者： 306798658@qq.com

什么是正则

概念：它使用单个字符串来描述或匹配一系列符合某个句法则规则的字符串；
正则表示式通常用来检索和替换那些符合某个模式的文本内容。
无论是查找某个文档，还是查询某个日志文件并分析其内容，都会用正则表示式。
正则就是一串有规则的字符串；
掌握好正则对于编写shell脚本有很大帮助；
各种变成语言中都有正则，原理是一样的。

- grep命令

命令格式：grep [-cinvrABC] 'word' filename
常用选项如下：

```
-c 表示打印符合要求的行数；
-i 表示忽略大小写；
-n 表示输出符合要求的行及其行号；
-v 表示打印不符合要求的号；
-r 遍历所有的子目录；
-A 后面跟一个数字，例如-A2表示打印符合要求的行及下面的两行；
-B 后面跟一个数字，例如-B2表示打印符合要求的行及上面的两行；
-C 后面跟一个数字，例如-C2表示打印符合要求的行及上下各两行。
-E 使其支持扩展正则，等同于egrep
```

扩展正则，指的是表达式中含有：+ ? | ( ) { }等符号，在grep中直接使用不行，需要加\脱义，如下3个命令等同：

```
# grep '^ab\?c' 1.txt
# egrep '^ab?c' 1.txt
# grep -E '^ab?c' 1.txt
```

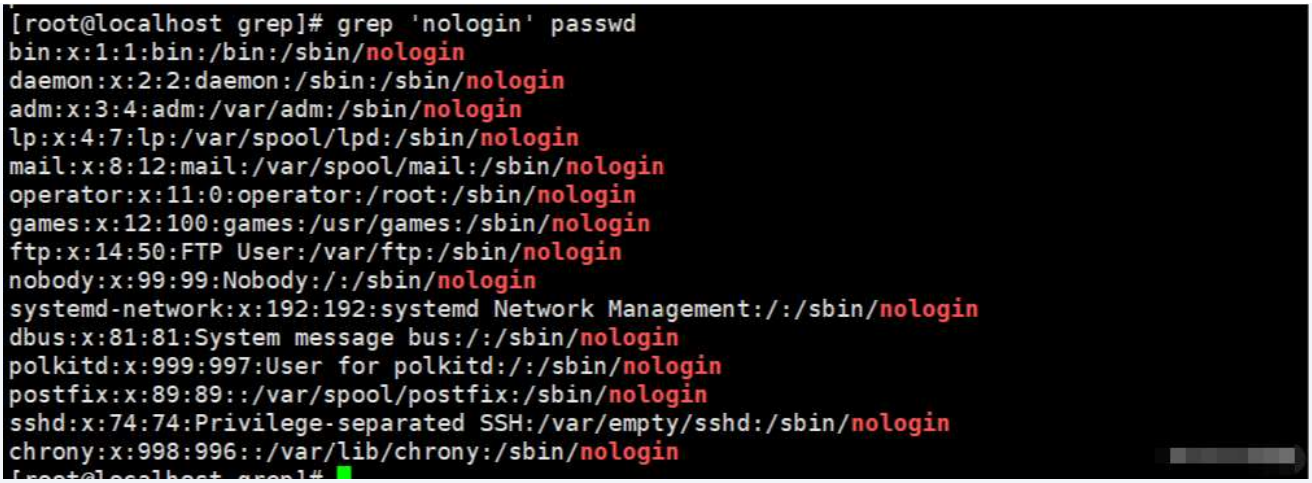
^ 这个符号表示以什么什么开头。如:^# 以#开头；
或者[^#] 取反，除#号开头的行

- 创建一个grep目录，拷贝过来一个文件/etc/passwd

> [root@localhost /]# mkdir grep #创建一个grep目录
> [root@localhost /]# cd /grep/ #进这个目录里面
> [root@localhost grep]# cp /etc/passwd . #拷贝passwd到.（本目录下）
> [root@localhost grep]# ls #查看目录下有什么文件
>
> passwd

- 过滤passwd文件里，nologin的字符

>



这里nologin过滤完后标红，是因为grep有一个--color=auto 添加颜色显示
可以查看下grep命令

```
[root@localhost grep]# which grep #which查看grep
alias grep='grep --color=auto'

/usr/bin/grep
```

- grep查看符合要求的行数 -c

```
[root@localhost grep]# grep -c 'nologin' passwd #-c符合’nologin'的行数

15
```

- grep -n显示这个文件包包含‘nologin'的行数。

```
[root@localhost grep]# grep -n 'nologin' passwd
2:bin:x:1:1:bin:/bin:/sbin/nologin
3:daemon:x:2:2:daemon:/sbin:/sbin/nologin
4:adm:x:3:4:adm:/var/adm:/sbin/nologin
5:lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
9:mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
10:operator:x:11:0:operator:/root:/sbin/nologin
11:games:x:12:100:games:/usr/games:/sbin/nologin
12:ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
13:nobody:x:99:99:Nobody:/:/sbin/nologin
14:systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
15:dbus:x:81:81:System message bus:/:/sbin/nologin
16:polkitd:x:999:997:User for polkitd:/:/sbin/nologin
17:postfix:x:89:89::/var/spool/postfix:/sbin/nologin
18:sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin

19:chrony:x:998:996::/var/lib/chrony:/sbin/nologin
```

- grep不区分大小写-i

```
[root@localhost grep]# grep -ni 'nologin' passwd
2:bin:x:1:1:bin:/bin:/sbin/nologin
3:daemon:x:2:2:daemon:/sbin:/sbin/NOLogin
4:adm:x:3:4:adm:/var/adm:/sbin/nologin
5:lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
9:mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
10:operator:x:11:0:operator:/root:/sbin/nologin
11:games:x:12:100:games:/usr/games:/sbin/nologin
12:ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
13:nobody:x:99:99:Nobody:/:/sbin/nologin
14:systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
15:dbus:x:81:81:System message bus:/:/sbin/nologin
16:polkitd:x:999:997:User for polkitd:/:/sbin/nologin
17:postfix:x:89:89::/var/spool/postfix:/sbin/nologin
18:sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
19:chrony:x:998:996::/var/lib/chrony:/sbin/nologin
```

- grep取反，除了nologin的显示出来 -v

```
[root@localhost grep]# grep -nv 'nologin' passwd
1:root:x:0:0:root:/root:/bin/bash
3:daemon:x:2:2:daemon:/sbin:/sbin/NOLogin
6:sync:x:5:0:sync:/sbin:/bin/sync
7:shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
8:halt:x:7:0:halt:/sbin:/sbin/halt
[root@localhost grep]#
```

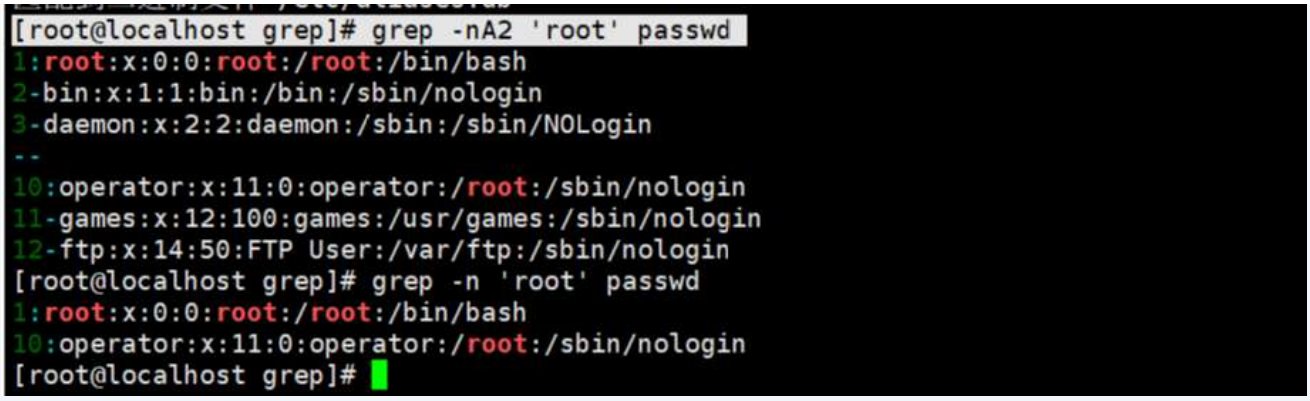
- 遍历所有的子目录 -r

[root@localhost grep]# grep -r 'root' /etc/ (etc目录下所有包含root的文件列出来)

```
/etc/postfix/main.cf:# Exception:  delivery for root is done as $default_user.
/etc/postfix/main.cf:# To attach to the screen sesssion, su root and run "screen -r
/etc/postfix/master.cf:# service type  private unpriv  chroot  wakeup  maxproc command + args
/etc/chrony.keys:# symmetric keys.  It should be readable only by root or the user to which
/etc/sudoers:## the root user, without needing the root password.
/etc/sudoers:## Allow root to run any commands anywhere
/etc/sudoers:root        ALL=(ALL)       ALL
/etc/sudoers:## cdrom as root
匹配到二进制文件 /etc/aliases.db
[root@localhost grep]# grep -r 'root' /etc/
```

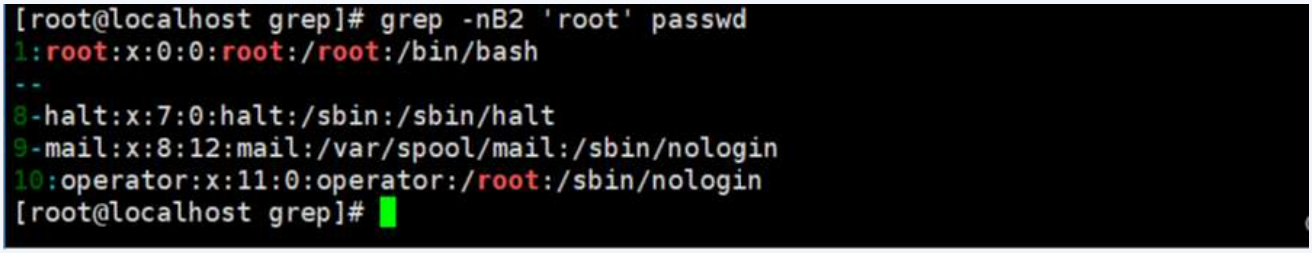
- -A2把包含root的行，以及这行下面的两行都打印出来。

[root@localhost grep]# grep -nA2 'root' passwd



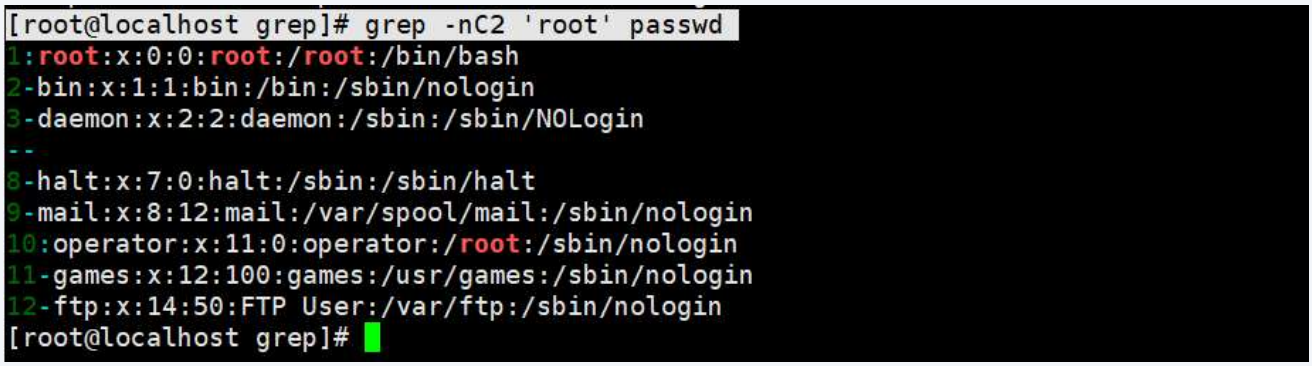
- -B2把包含root的行，以及这行上面的两行都打印出来。

> [root@localhost grep]# grep -nB2 'root' passwd



- -C2把包含root的行以及这上下两行都打印出来。

> [root@localhost grep]# grep -nC2 'root' passwd



grep正则表达式具体用法示例

查看一个文件里0-9所有的数字，[]括号表示里面的任意一个字符

> [root@localhost grep]# grep '[0-9]' passwd

```
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:997:User for polkitd:/:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
chrony:x:998:996::/var/lib/chrony:/sbin/nologin
```

反选-v，把不包含数字的行列出来。

[root@localhost grep]# grep -nv '[0-9]' /etc/inittab

```
[root@localhost grep]# grep -nv '[0-9]' /etc/inittab
1:# inittab is no longer used when using systemd.
2:#
3:# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
4:#
5:# Ctrl-Alt-Delete is handled by /usr/lib/systemd/system/ctrl-alt-del.target
6:#
7:# systemd uses 'targets' instead of runlevels. By default, there are two main targets:
8:#
11:#
12:# To view current default target, run:
13:# systemctl get-default
14:#
15:# To set a default target, run:
16:# systemctl set-default TARGET.target
17:#
```

取反，过滤掉所有以#开头的行

[root@localhost grep]# grep -nv '^#' inittab

把不是以#开头的行，显示出来

- 过滤非0-9开头的数字的行显示出来

```
[root@localhost grep]# grep '^[^0-9]' inittab
# inittab is no longer used when using systemd.
#
# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# Ctrl-Alt-Delete is handled by /usr/lib/systemd/system/ctrl-alt-del.target
#
# systemd uses 'targets' instead of runlevels. By default, there are two main targets:
#
# multi-user.target: analogous to runlevel 3
# graphical.target: analogous to runlevel 5
#
```

过滤出任意一个字符和重复字符，示例如下：

[root@localhost grep]# grep 'r.o' passwd

root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin

```
[root@localhost grep]# grep 'r.o' passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
```

.表示任意一个字符，上例中，r.o表示r与o之间有一个任意字符的行过滤出来。

指定要过滤出的字符出现次数，示例如下：

[root@localhost grep]# grep 'o{2}' passwd
root:x:0:0:root:/root:/bin/bash
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin

```
[root@localhost grep]# grep 'o\{2\}' passwd
root:x:0:0:root:/root:/bin/bash
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
```

这里用到了符号{}表示内部为数字，表示前面的字符要重复的词数。
{}左右都需要加上转义字符\，另外使用{}还可以表示一个范围，具体格式为{n1,n2}

{}花括号表示前一个字符的范围，如果o{2},表示2个连续的oo显示出来。

过滤出一个或多个指定的字符，示例如下：

[root@localhost grep]# egrep 'o+b' passwd

nobody:x:99:99:Nobody:/:/sbin/nologin

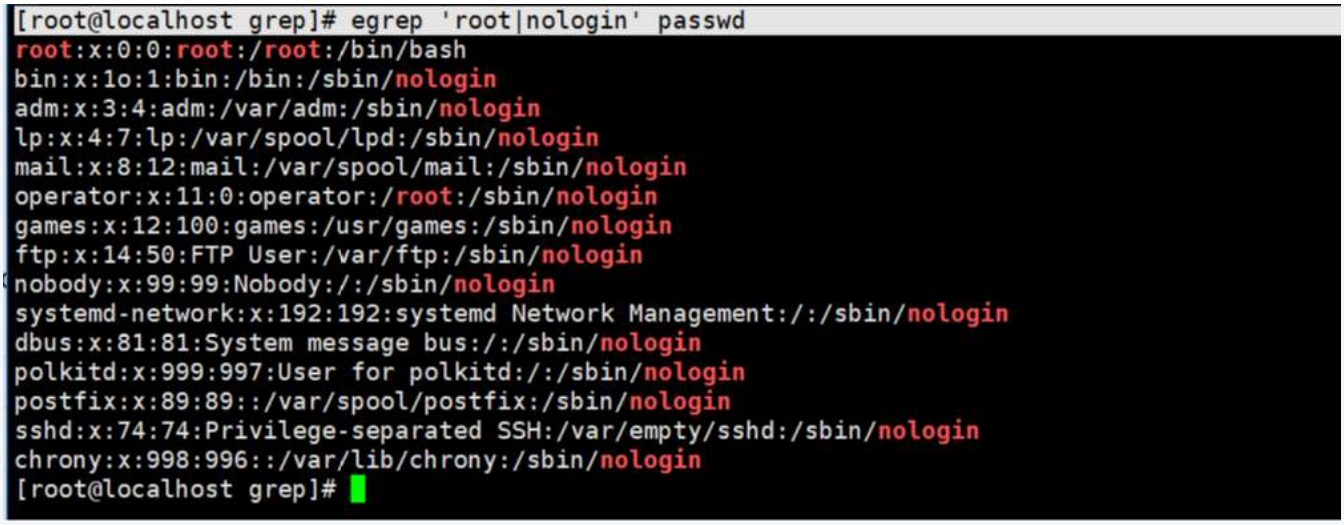
过滤出零个或一个指定的字符，示例如下：

[root@localhost grep]# egrep 'o?1o' passwd
bin:x:1o:1:bin:/bin:/sbin/nologin

o1o

过滤出字符串1或者字符串2，示例如下： |或者

[root@localhost grep]# egrep 'root|nologin' passwd



总结:

. 表示任意一个字符
* 表示*前面那个字符有零个或者多个
{} 表示一个范围
+ 表示匹配一个或多个+前面的字符
? 表示?前面的字符有零个或1个
| 表示或者