文本处理工具之一grep命令详解

2014-06-16 我来说两句 来源：文本处理工具之一grep命令详解 收藏 我要投稿

grep(Globel Search Regular Expression and Printing out the line)全面搜索正则表达式并把行打印出来）是一种强大的文本搜索工具，是一个对行进行操作的搜索工作，它能使用正则表达式搜索文本，并把匹配的行打印出来。Unix的grep家族包括grep、egrep和fgrep。 egrep表示扩展的grep，相比grep支持更多的元字符，"grep -E"相当于egrep。fgrep是fast grep，不支持元字符，但是搜索速度更快。grep搜索的结果被送到屏幕，不影响原文件内容。

1、grep的语法[] (man grep查看grep的帮助文档)

grep [options] 'pattern' FILE

命令 选项 模式 文件

grep不加引号直接过滤字符串;grep在进行模式匹配的时候必须加引号，单引和双引号都可以;grep在引用变量的时候必须加双引号

2、grep的选项[option]

-r：递归的搜索

-v:反向选取,只显示不符合模式的行

-o:只显示被模式匹配到的字符串，而不是整个行

-i:匹配时不区分大小写

-A #:显示匹配到的行时，顺便显示后面的#行(#表示数值)

-B #:前面的#行

-C #:前后的#行

-E:使用扩展的正则表达式

eg:grep选项的例子

```
cat > eg1.text << EOF
This is first
how are you
How old are you
fine,thanks
what,so what
What is your name
EOF
grep "you" eg1.text
grep -o "you" eg1.text
grep -v "you" eg1.text
grep -i "what" eg1.text
grep -A 1 "fine" eg1.text
grep -B 1 "fine" eg1.text
grep -C 1 "fine" eg1.text
```

以上代码直接粘贴复制在linux上可直接运行，代码解释运行效果，如下

```
[root@localhost ~]# cat > eg1.text << EOF
> This is first
> how are you
> How old are you
> fine,thanks
> what,so what
> What is your name
> EOF
[root@localhost ~]# grep "you" eg1.text
how are you
How old are you
What is your name
[root@localhost ~]# grep -o "you" eg1.text
you
you
you
[root@localhost ~]# grep -v "you" eg1.text
This is first
fine,thanks
what,so what
[root@localhost ~]# grep -i "what" eg1.text
what,so what
What is your name
[root@localhost ~]# grep -A 1 "fine" eg1.text
fine,thanks
what,so what
[root@localhost ~]# grep -B 1 "fine" eg1.text
How old are you
fine,thanks
[root@localhost ~]# grep -C 1 "fine" eg1.text
How old are you
fine,thanks
what,so what
```

用cat生成一个测试的文本文件

匹配文件中包含you的字符,且默认会输出匹配字符的一行

匹配文件中包含you的字符，-o参数只输出匹配的字符

-v参数是匹配包含you字符之外的行

-i参数，表示不区分大小写

-A参数后面跟一个数字，表示匹配字符的行，然后还显示后面1行

-B参数后面跟一个数字，表示匹配字符的行，然后还显示前面1行

-C参数后面跟一个数字，表示匹配字符的行，然后还显示前面1行和后面1行

3、

正则表达式(man regex)是指一个用来描述或者匹配一系列符合某个句法规则的字符串的单个字符串，通常被用来检索或替换那些符合某个

模式的文本内容。正则表达式分为：基本正则表达式和扩展正则表达式。

元字符就是指那些在正则表达式中具有特殊意义的专用字符。

grep支持基本正则表达式的元字符：

^:锚点行首的符合条件的内容，用法格式"^pattern"

$:锚点行首的符合条件的内容，用法格式"pattern$"

^$:匹配空白行

.:匹配任意单个字符

*:匹配紧挨在前面的字符任意次(0,1,多次)

.*:匹配任意长度的任意字符

\?:匹配紧挨在前面的字符0次或1次

\{m,n\}:匹配其前面的字符至少m次，至多n次

\{m,\}:匹配其前面的字符至少m次

\{m\}:精确匹配前面的m次

\{0,n\}:0到n次

\<:锚点词首----相当于\b,用法格式：\<pattern

\>:锚点词尾,用法格式:\>pattern

\<pattern\>：单词锚点

\(\):分组，用法格式：\(pattern\),引用第一个小括号的分组\1,第二个是\2,以此类推

[]：匹配指定范围内的任意单个字符

[^]：匹配指定范围外的任意单个字符

eg:基本正则表达的例子

(1)、显示/proc/meminfo文件中以不区分大小的s开头的行;

```
grep "^[sS]" /proc/meminfo
```

(2)、显示/etc/passwd中以nologin结尾的行;

```
grep "nologin$" /etc/passwd
```

(3)、显示/etc/inittab中空格开头的行;

```
grep "^$" /etc/inittab
```

(4)、显示/etc/passwd中，以r开头的字符而后跟了任意单个字符的行;

```
grep --color "^r." /etc/passwd
```

(5)、显示/etc/passwd中，以r开头后跟了o，o出现任意次的行;

```
grep --color "^ro*" /etc/passwd
```

(6)、显示/etc/passwd文件中，r后跟了任意长度任意字符后跟了h的行;

```
grep --color "r.*h" /etc/passwd
```

(7)、显示/etc/passwd中，r后跟了o，o出现0次或者1次的行;

```
grep --color "ro\?" /etc/passwd
```

(8)、显示/etc/passwd中，r后跟了o，o出现至少1次至多2次的行;

```
grep --color "ro\{1,2\}" /etc/passwd
```

(9)、显示/etc/passwd中，r后跟了o，o只出现2次的行;

```
grep --color "ro\{2\}" /etc/passwd
```

(10)、显示/etc/passwd中，匹配root这个单词的行;

```
grep --color "\<root\>" /etc/passwd
```

grep支持扩展表达式的元字符：支持所有基本正则表达式的元字符，有些和基本元字符在用法上不一样，扩展正则表达式的命令egrep或者grep -E

?:匹配紧挨在前面的字符0次或1次

{m,n}:至少m次，至多n次

():分组

+:至少匹配前面的字符一次

alb:匹配a或者b

eg:扩展正则表达式的例子

(1)、显示/etc/passwd中，r开头后跟了o，o出现0次或者1次的行;

```
egrep --color "ro?" /etc/passwd
```

(2)、显示/etc/passwd中，r开头后跟了o，o出现至少1次至多2次的行;

```
egrep --color "ro{1,2}" /etc/passwd
```

(3)、显示/etc/inittab文件中以一个数字开头并以一个与开头数字相同的数字结尾的;

```
egrep --color "^([0-9]).*\1$"  /etc/inittab
```

```
[root@localhost ~]# egrep --color "^([0-9]).*\1$"  /etc/inittab
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

(4)、显示/etc/passwd中，r开头后跟了o，o出现至少1次；

```
egrep --color "ro+" /etc/passwd
grep -E --color  "ro{1,}" /etc/passwd
```

(5)、显示/etc/passwd中，匹配root或者halt的行；

```
egrep --color "root|halt" /etc/passwd
```

(6)、显示/var/log/secure文件中包含"LOGIN ON"或者"Failed passwd"的行；

```
egrep --color "(LOGIN ON|Failed passwd)" /var/log/secure
```

grep支持字符和字符集合

\d:数字字符匹配。等效于 [0-9]。

\s:匹配任何空白字符，包括空格、制表符、换页符等。与 [\f\n\r\t\v] 等效。

\S:匹配任何非空白字符。与 [^ \f\n\r\t\v] 等效

\w:匹配任何字类字符，包括下划线。与“[A-Za-z0-9_]”等效。

\W:与任何非单词字符匹配。与“[^A-Za-z0-9_]”等效。

[:digit:]:所有数字, 相当于0-9 或者\d

[:lower:]:所有的小写字母

[:upper:]:所有的大写字母

[:alpha:]:所有的字母

[:alnum:]:相当于[0-9a-zA-Z]

[:space:]:空白字符 相当于\s

[:punct:]:所有标点符号

eg：支持字符集合的例子

(1)、显示/etc/rc.d/rc.sysinit中以#开头，且后面跟一个或多个空白字符，而后又跟了任意非空白字符的行；

```
grep "^#[[:space:]]\{1,\}[^[:space:]]"  /etc/rc.d/rc.sysinit
grep -E "^#\s{1,}\S" /etc/rc.d/rc.sysinit
```

(2)、显示/etc/inittab中包含了:一个数字:(即两个冒号中间一个数字)的行；

```
grep --color ":[[:digit:]]:" /etc/inittab
grep --color ":\d:" /etc/inittab
```

经典的例子

(1)、分组的例子

```
cat > test.txt <<EOF
He like his liker
He love his lover
She love her lover
```

```
She like her lover

EOF

grep "l..e.*l..er" test.txt

grep "\(l..e\).*\1r" test.txt
```

```
[root@localhost ~]# cat > test.txt <<EOF
> He like his liker
> He love his liker
> She love her lover
> She like her lover
> EOF
[root@localhost ~]# grep "l..e.*l..er" test.txt
He like his liker
He love his liker
She love her lover
She like her lover
```

```
[root@localhost ~]# grep "\(l..e\).*\1r" test.txt
He like his liker
She love her lover
```

(2)、匹配1-255的数字

```
cat > num.txt << EOF

12

234

255

256

EOF

grep --color -E "\<([1-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\>" num.txt
```

```
[root@localhost ~]# cat >num.txt <<EOF
> 12
> 234
> 255
> 256
> EOF
[root@localhost ~]# grep --color -E "\<([1-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\>" num.tx
12
234
255
```

(3)、匹配ABC类IP地址即 1.0.0.1---223.255.255.254

```
cat > ip.txt <<EOF

1.0.0.254

1.0.0.255

1.2.3.4

223.255.255.254

224.255.255.252

2.255.255.255

EOF

grep -E --color "\<([1-9]|[1-9][0-9]|1[0-9][0-9]|2[0-1][0-9]|22[0-3])\.([0-9]|[1-9][0-9]
```

```
[root@localhost ~]# cat > ip.txt <<EOF
> 1.0.0.254
> 1.0.0.255
> 1.2.3.4
> 223.255.255.254
> 224.255.255.252
> 2.255.255.255
> EOF
[root@localhost ~]# grep -E --color "\<([1-9]|[1-9][0-9]|1[0-9][0-9]|2[0-1][0-9]|22[0-3])\.([0-9
|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])
.([1-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-4])\>" ip.txt
1.0.0.254
1.2.3.4
223.255.255.254
```

(4)、匹配Email地址:任意长度数字字母@任意长度数字字母.（com"orglnet等等）

```
cat > email.txt << EOF

5678967@qq.com

jie231@sina.cn
```

```
ken_tom@netcom.org

jerry#li@baidu.net

li@souhu.net

EOF

grep -E --color "^\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$" email.txt
```

```
[root@localhost ~]# cat > email.txt << EOF
> 5678967@qq.com
> jie231@sina.cn
> ken_tom@netcom.org
> jerry#li@baidu.net
> li@souhu.net
> EOF
[root@localhost ~]# grep -E --color "^\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$" email.txt
5678967@qq.com
jie231@sina.cn
ken_tom@netcom.org
li@souhu.net
[root@localhost ~]#
```

(5)、匹配手机号码：手机号码是1[3"4|5|8]后面接9位数字的

```
cat > tel.txt << EOF

13690876890

12589098379

15608764083

15820974619

138074082711

18618203761

19209783900

1329873909

EOF

grep --color -E "\<1[3|4|5|8][0-9]{9}\>" tel.txt
```

```
[root@localhost ~]# cat > tel.txt << EOF
> 13690876890
> 12589098379
> 15608764083
> 15820974619
> 138074082711
> 18618203761
> 19209783900
> 1329873909
> EOF
[root@localhost ~]# grep --color -E "\<1[3|4|5|8][0-9]{9}\>" tel.txt
13690876890
15608764083
15820974619
18618203761
[root@localhost ~]#
```