[**每天一个linux命令（39）：grep 命令**](http://www.cnblogs.com/peida/archive/2012/12/17/2821195.html)

Linux系统中grep命令是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹 配的行打印出来。grep全称是Global Regular Expression Print，表示全局正则表达式版本，它的使用权限是所有用户。

grep的工作方式是这样的，它在一个或多个文件中搜索字符串模板。如果模板包括空格，则必须被引用，模板后的所有字符串被看作文件名。搜索的结果被送到标准输出，不影响原文件内容。

grep可用于shell脚本，因为grep通过返回一个状态值来说明搜索的状态，如果模板搜索成功，则返回0，如果搜索不成功，则返回1，如果搜索的文件不存在，则返回2。我们利用这些返回值就可进行一些自动化的文本处理工作。

**1．命令格式：**

grep [option] pattern file

**2．命令功能：**

用于过滤/搜索的特定字符。可使用正则表达式能多种命令配合使用，使用上十分灵活。

**3．命令参数：**

-a   --text   #不要忽略二进制的数据。

-A<显示行数>   --after-context=<显示行数>   #除了显示符合范本样式的那一列之外，并显示该行之后的内容。

-b   --byte-offset   #在显示符合样式的那一行之前，标示出该行第一个字符的编号。

-B<显示行数>   --before-context=<显示行数>   #除了显示符合样式的那一行之外，并显示该行之前的内容。

-c    --count   #计算符合样式的列数。

-C<显示行数>    --context=<显示行数>或-<显示行数>   #除了显示符合样式的那一行之外，并显示该行之前后的内容。

-d <动作>      --directories=<动作>   #当指定要查找的是目录而非文件时，必须使用这项参数，否则grep指令将回报信息并停止动作。

-e<范本样式>  --regexp=<范本样式>   #指定字符串做为查找文件内容的样式。

-E      --extended-regexp   #将样式为延伸的普通表示法来使用。

-f<规则文件>  --file=<规则文件>   #指定规则文件，其内容含有一个或多个规则样式，让grep查找符合规则条件的文件内容，格式为每行一个规则样式。

-F   --fixed-regexp   #将样式视为固定字符串的列表。

-G   --basic-regexp   #将样式视为普通的表示法来使用。

-h   --no-filename   #在显示符合样式的那一行之前，不标示该行所属的文件名称。

-H   --with-filename   #在显示符合样式的那一行之前，表示该行所属的文件名称。

-i    --ignore-case   #忽略字符大小写的差别。

-l    --file-with-matches   #列出文件内容符合指定的样式的文件名称。

-L   --files-without-match   #列出文件内容不符合指定的样式的文件名称。

-n   --line-number   #在显示符合样式的那一行之前，标示出该行的列数编号。

-q   --quiet或--silent   #不显示任何信息。

-r   --recursive   #此参数的效果和指定“-d recurse”参数相同。

-s   --no-messages   #不显示错误信息。

-v   --revert-match   #显示不包含匹配文本的所有行。

-V   --version   #显示版本信息。

-w   --word-regexp   #只显示全字符合的列。

-x    --line-regexp   #只显示全列符合的列。

-y   #此参数的效果和指定“-i”参数相同。

**4．规则表达式：**

**grep的规则表达式:**

^  #锚定行的开始 如：'^grep'匹配所有以grep开头的行。

$  #锚定行的结束 如：'grep$'匹配所有以grep结尾的行。

.  #匹配一个非换行符的字符 如：'gr.p'匹配gr后接一个任意字符，然后是p。

\*  #匹配零个或多个先前字符 如：'\*grep'匹配所有一个或多个空格后紧跟grep的行。

.\*   #一起用代表任意字符。

[]   #匹配一个指定范围内的字符，如'[Gg]rep'匹配Grep和grep。

[^]  #匹配一个不在指定范围内的字符，如：'[^A-FH-Z]rep'匹配不包含A-R和T-Z的一个字母开头，紧跟rep的行。

\(..\)  #标记匹配字符，如'\(love\)'，love被标记为1。

\<      #锚定单词的开始，如:'\<grep'匹配包含以grep开头的单词的行。

\>      #锚定单词的结束，如'grep\>'匹配包含以grep结尾的单词的行。

x\{m\}  #重复字符x，m次，如：'0\{5\}'匹配包含5个o的行。

x\{m,\}  #重复字符x,至少m次，如：'o\{5,\}'匹配至少有5个o的行。

x\{m,n\}  #重复字符x，至少m次，不多于n次，如：'o\{5,10\}'匹配5--10个o的行。

\w    #匹配文字和数字字符，也就是[A-Za-z0-9]，如：'G\w\*p'匹配以G后跟零个或多个文字或数字字符，然后是p。

\W    #\w的反置形式，匹配一个或多个非单词字符，如点号句号等。

\b    #单词锁定符，如: '\bgrep\b'只匹配grep。

**POSIX字符:**

为了在不同国家的字符编码中保持一至，POSIX(The Portable Operating System Interface)增加了特殊的字符类，如[:alnum:]是[A-Za-z0-9]的另一个写法。要把它们放到[]号内才能成为正则表达式，如[A- Za-z0-9]或[[:alnum:]]。在linux下的grep除fgrep外，都支持POSIX的字符类。

[:alnum:]    #文字数字字符

[:alpha:]    #文字字符

[:digit:]    #数字字符

[:graph:]    #非空字符（非空格、控制字符）

[:lower:]    #小写字符

[:cntrl:]    #控制字符

[:print:]    #非空字符（包括空格）

[:punct:]    #标点符号

[:space:]    #所有空白字符（新行，空格，制表符）

[:upper:]    #大写字符

[:xdigit:]   #十六进制数字（0-9，a-f，A-F）

**5．使用实例：**

**实例1：查找指定进程**

**命令：**

ps -ef|grep svn

**输出：**

[root@localhost ~]# ps -ef|grep svn

root 4943   1      0  Dec05 ?   00:00:00 svnserve -d -r /opt/svndata/grape/

root 16867 16838  0 19:53 pts/0    00:00:00 grep svn

[root@localhost ~]#

**说明：**

第一条记录是查找出的进程；第二条结果是grep进程本身，并非真正要找的进程。

**实例2：查找指定进程个数**

**命令：**

ps -ef|grep svn -c

ps -ef|grep -c svn

**输出：**

[root@localhost ~]# ps -ef|grep svn -c

2

[root@localhost ~]# ps -ef|grep -c svn

2

[root@localhost ~]#

**说明：**

**实例3：从文件中读取关键词进行搜索**

**命令：**

cat test.txt | grep -f test2.txt

**输出：**

[root@localhost test]# cat test.txt

hnlinux

peida.cnblogs.com

ubuntu

ubuntu linux

redhat

Redhat

linuxmint

[root@localhost test]# cat test2.txt

linux

Redhat

[root@localhost test]# cat test.txt | grep -f test2.txt

hnlinux

ubuntu linux

Redhat

linuxmint

[root@localhost test]#

**说明：**

输出test.txt文件中含有从test2.txt文件中读取出的关键词的内容行

**实例3：从文件中读取关键词进行搜索 且显示行号**

**命令：**

cat test.txt | grep -nf test2.txt

**输出：**

[root@localhost test]# cat test.txt

hnlinux

peida.cnblogs.com

ubuntu

ubuntu linux

redhat

Redhat

linuxmint

[root@localhost test]# cat test2.txt

linux

Redhat

[root@localhost test]# cat test.txt | grep -nf test2.txt

1:hnlinux

4:ubuntu linux

6:Redhat

7:linuxmint

[root@localhost test]#

**说明：**

输出test.txt文件中含有从test2.txt文件中读取出的关键词的内容行，并显示每一行的行号

**实例5：从文件中查找关键词**

**命令：**

grep 'linux' test.txt

**输出：**

[root@localhost test]# grep 'linux' test.txt

hnlinux

ubuntu linux

linuxmint

[root@localhost test]# grep -n 'linux' test.txt

1:hnlinux

4:ubuntu linux

7:linuxmint

[root@localhost test]#

**说明：**

**实例6：从多个文件中查找关键词**

**命令：**

grep 'linux' test.txt test2.txt

**输出：**

[root@localhost test]# grep -n 'linux' test.txt test2.txt

test.txt:1:hnlinux

test.txt:4:ubuntu linux

test.txt:7:linuxmint

test2.txt:1:linux

[root@localhost test]# grep 'linux' test.txt test2.txt

test.txt:hnlinux

test.txt:ubuntu linux

test.txt:linuxmint

test2.txt:linux

[root@localhost test]#

**说明：**

多文件时，输出查询到的信息内容行时，会把文件的命名在行最前面输出并且加上":"作为标示符

**实例7：grep不显示本身进程**

**命令：**

ps aux|grep \[s]sh

ps aux | grep ssh | grep -v "grep"

**输出：**

[root@localhost test]# ps aux|grep ssh

root   2720  0.0  0.0  62656  1212 ?      Ss   Nov02   0:00 /usr/sbin/sshd

root  16834  0.0  0.0  88088  3288 ?      Ss   19:53   0:00 sshd: root@pts/0

root  16901  0.0  0.0  61180   764 pts/0  S+   20:31   0:00 grep ssh

[root@localhost test]# ps aux|grep \[s]sh]

[root@localhost test]# ps aux|grep \[s]sh

root   2720  0.0  0.0  62656  1212 ?      Ss   Nov02   0:00 /usr/sbin/sshd

root  16834  0.0  0.0  88088  3288 ?      Ss   19:53   0:00 sshd: root@pts/0

[root@localhost test]# ps aux | grep ssh | grep -v "grep"

root   2720  0.0  0.0  62656  1212 ?      Ss   Nov02   0:00 /usr/sbin/sshd

root  16834  0.0  0.0  88088  3288 ?      Ss   19:53   0:00 sshd: root@pts/0

**说明：**

**实例8：找出已u开头的行内容**

**命令：**

cat test.txt |grep ^u

**输出：**

[root@localhost test]# cat test.txt |grep ^u

ubuntu

ubuntu linux

[root@localhost test]#

**说明：**

**实例9：输出非u开头的行内容**

**命令：**

cat test.txt |grep ^[^u]

**输出：**

[root@localhost test]# cat test.txt |grep ^[^u]

hnlinux

peida.cnblogs.com

redhat

Redhat

linuxmint

[root@localhost test]#

**说明：**

**实例10：输出以hat结尾的行内容**

**命令：**

cat test.txt |grep hat$

**输出：**

[root@localhost test]# cat test.txt |grep hat$

redhat

Redhat

[root@localhost test]#

**说明：**

**实例11：**

**命令：**

**输出：**

[root@localhost test]# ifconfig eth0|grep "[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}"

          inet addr:192.168.120.204  Bcast:192.168.120.255  Mask:255.255.255.0

[root@localhost test]# ifconfig eth0|grep -E "([0-9]{1,3}\.){3}[0-9]"

          inet addr:192.168.120.204  Bcast:192.168.120.255  Mask:255.255.255.0

[root@localhost test]#

**说明：**

**实例12：显示包含ed或者at字符的内容行**

**命令：**

cat test.txt |grep -E "ed|at"

**输出：**

[root@localhost test]# cat test.txt |grep -E "peida|com"

peida.cnblogs.com

[root@localhost test]# cat test.txt |grep -E "ed|at"

redhat

Redhat

[root@localhost test]#

**说明：**

**实例13：显示当前目录下面以.txt 结尾的文件中的所有包含每个字符串至少有7个连续小写字符的字符串的行**

**命令：**

grep '[a-z]\{7\}' \*.txt

**输出：**

[root@localhost test]# grep '[a-z]\{7\}' \*.txt

test.txt:hnlinux

test.txt:peida.cnblogs.com

test.txt:linuxmint

[root@localhost test]#

**说明：**