[**每天一个linux命令（22）：find 命令的参数详解**](http://www.cnblogs.com/peida/archive/2012/11/16/2773289.html)

find一些常用参数的一些常用实例和一些具体用法和注意事项。

**1．使用name选项：**

文件名选项是find命令最常用的选项，要么单独使用该选项，要么和其他选项一起使用。  可以使用某种文件名模式来匹配文件，记住要用引号将文件名模式引起来。  不管当前路径是什么，如果想要在自己的根目录$HOME中查找文件名符合\*.log的文件，使用~作为 'pathname'参数，波浪号~代表了你的$HOME目录。

find ~ -name "\*.log" -print

想要在当前目录及子目录中查找所有的‘ \*.log‘文件，可以用：

find . -name "\*.log" -print

想要的当前目录及子目录中查找文件名以一个大写字母开头的文件，可以用：

find . -name "[A-Z]\*" -print

想要在/etc目录中查找文件名以host开头的文件，可以用：

find /etc -name "host\*" -print

想要查找$HOME目录中的文件，可以用：

find ~ -name "\*" -print 或find . -print

要想让系统高负荷运行，就从根目录开始查找所有的文件。

find / -name "\*" -print

如果想在当前目录查找文件名以一个个小写字母开头，最后是4到9加上.log结束的文件：

命令：

find . -name "[a-z]\*[4-9].log" -print

输出：

[root@localhost test]# ll

总计 316

-rw-r--r-- 1 root root 302108 11-13 06:03 log2012.log

-rw-r--r-- 1 root root     61 11-13 06:03 log2013.log

-rw-r--r-- 1 root root      0 11-13 06:03 log2014.log

-rw-r--r-- 1 root root      0 11-13 06:06 log2015.log

drwxr-xr-x 6 root root   4096 10-27 01:58 scf

drwxrwxr-x 2 root root   4096 11-13 06:08 test3

drwxrwxr-x 2 root root   4096 11-13 05:50 test4

[root@localhost test]# find . -name "[a-z]\*[4-9].log" -print

./log2014.log

./log2015.log

./test4/log2014.log

[root@localhost test]#

**2．用perm选项：**

按照文件权限模式用-perm选项,按文件权限模式来查找文件的话。最好使用八进制的权限表示法。

如在当前目录下查找文件权限位为755的文件，即文件属主可以读、写、执行，其他用户可以读、执行的文件，可以用：

[root@localhost test]# find . -perm 755 -print

.

./scf

./scf/lib

./scf/service

./scf/service/deploy

./scf/service/deploy/product

./scf/service/deploy/info

./scf/doc

./scf/bin

[root@localhost test]#

还有一种表达方法：在八进制数字前面要加一个横杠-，表示都匹配，如-007就相当于777，-005相当于555,

**命令：**

find . -perm -005

**输出：**

[root@localhost test]# ll

总计 316

-rw-r--r-- 1 root root 302108 11-13 06:03 log2012.log

-rw-r--r-- 1 root root     61 11-13 06:03 log2013.log

-rw-r--r-- 1 root root      0 11-13 06:03 log2014.log

-rw-r--r-- 1 root root      0 11-13 06:06 log2015.log

drwxr-xr-x 6 root root   4096 10-27 01:58 scf

drwxrwxr-x 2 root root   4096 11-13 06:08 test3

drwxrwxr-x 2 root root   4096 11-13 05:50 test4

[root@localhost test]# find . -perm -005

.

./test4

./scf

./scf/lib

./scf/service

./scf/service/deploy

./scf/service/deploy/product

./scf/service/deploy/info

./scf/doc

./scf/bin

./test3

[root@localhost test]#

**3．忽略某个目录：**

如果在查找文件时希望忽略某个目录，因为你知道那个目录中没有你所要查找的文件，那么可以使用-prune选项来指出需要忽略的目录。在使用-prune选项时要当心，因为如果你同时使用了-depth选项，那么-prune选项就会被find命令忽略。如果希望在test目录下查找文件，但不希望在test/test3目录下查找，可以用：

**命令：**

find test -path "test/test3" -prune -o -print

**输出：**

[root@localhost soft]# find test -path "test/test3" -prune -o -print

test

test/log2014.log

test/log2015.log

test/test4

test/test4/log2014.log

test/test4/log2013.log

test/test4/log2012.log

test/scf

test/scf/lib

test/scf/service

test/scf/service/deploy

test/scf/service/deploy/product

test/scf/service/deploy/info

test/scf/doc

test/scf/bin

test/log2013.log

test/log2012.log

[root@localhost soft]#

**4．使用find查找文件的时候怎么避开某个文件目录：**

**实例1：**在test 目录下查找不在test4子目录之内的所有文件

**命令：**

find test -path "test/test4" -prune -o -print

**输出：**

[root@localhost soft]# find test

test

test/log2014.log

test/log2015.log

test/test4

test/test4/log2014.log

test/test4/log2013.log

test/test4/log2012.log

test/scf

test/scf/lib

test/scf/service

test/scf/service/deploy

test/scf/service/deploy/product

test/scf/service/deploy/info

test/scf/doc

test/scf/bin

test/log2013.log

test/log2012.log

test/test3

[root@localhost soft]# find test -path "test/test4" -prune -o -print

test

test/log2014.log

test/log2015.log

test/scf

test/scf/lib

test/scf/service

test/scf/service/deploy

test/scf/service/deploy/product

test/scf/service/deploy/info

test/scf/doc

test/scf/bin

test/log2013.log

test/log2012.log

test/test3

[root@localhost soft]#

**说明：**

find [-path ..] [expression]

在路径列表的后面的是表达式

-path "test" -prune -o -print 是 -path "test" -a -prune -o -print 的简写表达式按顺序求值, -a 和 -o 都是短路求值，与 shell 的 && 和 || 类似如果

-path "test" 为真，则求值 -prune , -prune 返回真，与逻辑表达式为真；否则不求值 -prune，与逻辑表达式为假。如果 -path "test" -a -prune 为假，则求值 -print ，-print返回真，或逻辑表达式为真；否则不求值 -print，或逻辑表达式为真。

这个表达式组合特例可以用伪码写为:

if -path "test" then

-prune

else

-print

**实例2：避开多个文件夹:**

**命令：**

find test \( -path test/test4 -o -path test/test3 \) -prune -o -print

**输出：**

[root@localhost soft]# find test \( -path test/test4 -o -path test/test3 \) -prune -o -print

test

test/log2014.log

test/log2015.log

test/scf

test/scf/lib

test/scf/service

test/scf/service/deploy

test/scf/service/deploy/product

test/scf/service/deploy/info

test/scf/doc

test/scf/bin

test/log2013.log

test/log2012.log

[root@localhost soft]#

**说明：**

圆括号表示表达式的结合。  \ 表示引用，即指示 shell 不对后面的字符作特殊解释，而留给 find 命令去解释其意义。

**实例3：查找某一确定文件，-name等选项加在-o 之后**

**命令：**

find test \(-path test/test4 -o -path test/test3 \) -prune -o -name "\*.log" -print

输出：

[root@localhost soft]# find test \( -path test/test4 -o -path test/test3 \) -prune -o -name "\*.log" -print

test/log2014.log

test/log2015.log

test/log2013.log

test/log2012.log

[root@localhost soft]#

**5．使用user和nouser选项：**

按文件属主查找文件：

**实例1：在$HOME目录中查找文件属主为peida的文件**

**命令：**

find ~ -user peida -print

**实例2：在/etc目录下查找文件属主为peida的文件:**

命令：

find /etc -user peida -print

**说明：**

实例3：为了查找属主帐户已经被删除的文件，可以使用-nouser选项。在/home目录下查找所有的这类文件

**命令：**

find /home -nouser -print

**说明：**

这样就能够找到那些属主在/etc/passwd文件中没有有效帐户的文件。在使用-nouser选项时，不必给出用户名； find命令能够为你完成相应的工作。

**6．使用group和nogroup选项：**

就像user和nouser选项一样，针对文件所属于的用户组， find命令也具有同样的选项，为了在/apps目录下查找属于gem用户组的文件，可以用：

find /apps -group gem -print

要查找没有有效所属用户组的所有文件，可以使用nogroup选项。下面的find命令从文件系统的根目录处查找这样的文件:

find / -nogroup-print

**7．按照更改时间或访问时间等查找文件：**

如果希望按照更改时间来查找文件，可以使用mtime,atime或ctime选项。如果系统突然没有可用空间了，很有可能某一个文件的长度在此期间增长迅速，这时就可以用mtime选项来查找这样的文件。

用减号-来限定更改时间在距今n日以内的文件，而用加号+来限定更改时间在距今n日以前的文件。

希望在系统根目录下查找更改时间在5日以内的文件，可以用：

find / -mtime -5 -print

为了在/var/adm目录下查找更改时间在3日以前的文件，可以用:

find /var/adm -mtime +3 -print

**8．查找比某个文件新或旧的文件：**

如果希望查找更改时间比某个文件新但比另一个文件旧的所有文件，可以使用-newer选项。

它的一般形式为：

newest\_file\_name ! oldest\_file\_name

其中，！是逻辑非符号。

**实例1：查找更改时间比文件log2012.log新但比文件log2017.log旧的文件**

**命令：**

find -newer log2012.log ! -newer log2017.log

**输出：**

[root@localhost test]# ll

总计 316

-rw-r--r-- 1 root root 302108 11-13 06:03 log2012.log

-rw-r--r-- 1 root root     61 11-13 06:03 log2013.log

-rw-r--r-- 1 root root      0 11-13 06:03 log2014.log

-rw-r--r-- 1 root root      0 11-13 06:06 log2015.log

-rw-r--r-- 1 root root      0 11-16 14:41 log2016.log

-rw-r--r-- 1 root root      0 11-16 14:43 log2017.log

drwxr-xr-x 6 root root   4096 10-27 01:58 scf

drwxrwxr-x 2 root root   4096 11-13 06:08 test3

drwxrwxr-x 2 root root   4096 11-13 05:50 test4

[root@localhost test]# find -newer log2012.log ! -newer log2017.log

.

./log2015.log

./log2017.log

./log2016.log

./test3

[root@localhost test]#

**实例2：查找更改时间在比log2012.log文件新的文件**

**命令：**

find . -newer **log2012.log** -print

**输出：**

[root@localhost test]# find -newer log2012.log

.

./log2015.log

./log2017.log

./log2016.log

./test3

[root@localhost test]#

**9．使用type选项：**

**实例1：在/etc目录下查找所有的目录**

命令：

find /etc -type d -print

**实例2：在当前目录下查找除目录以外的所有类型的文件**

**命令：**

find . ! -type d -print

**实例3：在/etc目录下查找所有的符号链接文件**

**命令：**

find /etc -type l -print

**10．使用size选项：**

可以按照文件长度来查找文件，这里所指的文件长度既可以用块（block）来计量，也可以用字节来计量。以字节计量文件长度的表达形式为N c；以块计量文件长度只用数字表示即可。

在按照文件长度查找文件时，一般使用这种以字节表示的文件长度，在查看文件系统的大小，因为这时使用块来计量更容易转换。

**实例1：在当前目录下查找文件长度大于1 M字节的文件**

**命令：**

find . -size +1000000c -print

**实例2：在/home/apache目录下查找文件长度恰好为100字节的文件:**

**命令：**

find /home/apache -size 100c -print

**实例3：在当前目录下查找长度超过10块的文件（一块等于512字节）**

**命令：**

find . -size +10 -print

**11．使用depth选项：**

在使用find命令时，可能希望先匹配所有的文件，再在子目录中查找。使用depth选项就可以使find命令这样做。这样做的一个原因就是，当在使用find命令向磁带上备份文件系统时，希望首先备份所有的文件，其次再备份子目录中的文件。

**实例1：find命令从文件系统的根目录开始，查找一个名为CON.FILE的文件。**

**命令：**

find / -name "CON.FILE" -depth -print

说明：

它将首先匹配所有的文件然后再进入子目录中查找

**12．使用mount选项：**

  在当前的文件系统中查找文件（不进入其他文件系统），可以使用find命令的mount选项。

**实例1：从当前目录开始查找位于本文件系统中文件名以XC结尾的文件**

**命令：**

find . -name "\*.XC" -mount -print