

linux find 命令格式及 find 命令详解

1、find 命令的一般形式为；

find pathname -options [-print -exec -ok ...]

2、find 命令的参数；

pathname: find 命令所查找的目录路径。例如用.来表示当前目录，用/来表示系统根目录。

-print: find 命令将匹配的文件输出到标准输出。

-exec: find 命令对匹配的文件执行该参数所给出的 shell 命令。相应命令的形式为'command' { } ;，注意{ } 和 ; 之间的空格。

-ok: 和-exec 的作用相同，只不过以一种更为安全的模式来执行该参数所给出的 shell 命令，在执行每一个命令之前，都会给出提示，让用户来确定是否执行。

3、find 命令选项

-name

按照文件名查找文件。

-perm

按照文件权限来查找文件。

-prune

使用这一选项可以使 find 命令不在当前指定的目录中查找，如果同时使用-depth 选项，那么-prune 将被 find 命令忽略。

-user

按照文件属主来查找文件。

-group

按照文件所属的组来查找文件。

-mtime -n +n

按照文件的更改时间来查找文件，-n 表示文件更改时间距现在 n 天以内，+n 表示文件更改时间距现在 n 天以前。find 命令还有-atime 和-ctime 选项，但它们都和-m time 选项。

-nogroup

查找无有效所属组的文件，即该文件所属的组在/etc/groups 中不存在。

-nouser

查找无有效属主的文件，即该文件的属主在/etc/passwd 中不存在。

-newer file1 ! file2

查找更改时间比文件 file1 新但比文件 file2 旧的文件。

-type

查找某一类型的文件，诸如：

b - 块设备文件。

d - 目录。

c - 字符设备文件。

p - 管道文件。

l - 符号链接文件。

f - 普通文件。

-size n: [c] 查找文件长度为 **n** 块的文件，带有 **c** 时表示文件长度以字节计。**-depth**：在查找文件时，首先查找当前目录中的文件，然后再在其子目录中查找。

-fstype：查找位于某一类型文件系统中的文件，这些文件系统类型通常可以在配置文件 **/etc/fstab** 中找到，该配置文件中包含了本系统中有关文件系统的信息。

-mount：在查找文件时不跨越文件系统 **mount** 点。

-follow：如果 **find** 命令遇到符号链接文件，就跟踪至链接所指向的文件。

-cpio：对匹配的文件使用 **cpio** 命令，将这些文件备份到磁带设备中。

另外,下面三个的区别:

-amin n

查找系统中最后 **N** 分钟访问的文件

-atime n

查找系统中最后 **n*24** 小时访问的文件

-cmin n

查找系统中最后 **N** 分钟被改变文件状态的文件

-ctime n

查找系统中最后 **n*24** 小时被改变文件状态的文件

-mmin n

查找系统中最后 **N** 分钟被改变文件数据的文件

-mtime n

查找系统中最后 **n*24** 小时被改变文件数据的文件

4、使用 **exec** 或 **ok** 来执行 **shell** 命令

使用 **find** 命令时，只要把想要的操作写在一个文件里，就可以用 **exec** 来配合 **find** 命令查找，很方便的

在有些操作系统中只允许 **-exec** 选项执行诸如 **ls** 或 **ls -l** 这样的命令。大多数用户使用这一选项是为了查找旧文件并删除它们。建议在真正执行 **rm** 命令删除文件之前，最好先用 **ls** 命令看一下，确认它们是所要删除的文件。

exec 选项后面跟随着所要执行的命令或脚本，然后是一对儿 **{ }**，一个空格和一个，最后是一个分号。为了使用 **exec** 选项，必须要同时使用 **print** 选项。如果验证一下 **find** 命令，会发现该命令只输出从当前路径起的

相对路径及文件名。

例如：为了用 `ls -l` 命令列出所匹配到的文件，可以把 `ls -l` 命令放在 `find` 命令的 `-exec` 选项中

```
# find . -type f -exec ls -l { } ;  
-rw-r--r-- 1 root root 34928 2003-02-25 ./conf/httpd.conf  
-rw-r--r-- 1 root root 12959 2003-02-25 ./conf/magic  
-rw-r--r-- 1 root root 180 2003-02-25 ./conf.d/README
```

上面的例子中，`find` 命令匹配到了当前目录下的所有普通文件，并在 `-exec` 选项中使用 `ls -l` 命令将它们列出。

在 `/logs` 目录中查找更改时间在 5 日以前的文件并删除它们：

```
$ find logs -type f -mtime +5 -exec rm { } ;
```

记住：在 `shell` 中用任何方式删除文件之前，应当先查看相应的文件，一定要小心！当使用诸如 `mv` 或 `rm` 命令时，可以使用 `-exec` 选项的安全模式。它将在对每个匹配到的文件进行操作之前提示你。

在下面的例子中，`find` 命令在当前目录中查找所有文件名以 `.LOG` 结尾、更改时间在 5 日以上的文件，并删除它们，只不过在删除之前先给出提示。

```
$ find . -name "*.conf" -mtime +5 -ok rm { } ;  
< rm ... ./conf/httpd.conf > ? n
```

按 `y` 键删除文件，按 `n` 键不删除。

任何形式的命令都可以在 `-exec` 选项中使用。

在下面的例子中我们使用 `grep` 命令。`find` 命令首先匹配所有文件名为 “`passwd*`” 的文件，例如 `passwd`、`passwd.old`、`passwd.bak`，然后执行 `grep` 命令看看在这些文件中是否存在一个 `sam` 用户。

```
# find /etc -name "passwd*" -exec grep "sam" { } ;  
sam:x:501:501::/usr/sam:/bin/bash
```

`find` 命令的例子；

1、查找当前用户主目录下的所有文件：

下面两种方法都可以使用

```
$ find $HOME -print  
$ find ~ -print
```

2、让当前目录中文件属主具有读、写权限，并且文件所属组的用户和其他用户具有读权限的文件；

```
$ find . -type f -perm 644 -exec ls -l { } ;
```

3、为了查找系统中所有文件长度为 0 的普通文件，并列出它们的完整路径；

```
$ find / -type f -size 0 -exec ls -l { } ;
```

4、查找 `/var/logs` 目录中更改时间在 7 日以前的普通文件，并在删除之前询问它们；

```
$ find /var/logs -type f -mtime +7 -ok rm { } ;
```

5、为了查找系统中所有属于 `root` 组的文件；

```
$find . -group root -exec ls -l { } ;
```

```
-rw-r--r-- 1 root root 595 10 月 31 01:09 ./fie1
```

6、find 命令将删除当目录中访问时间在 7 日以来、含有数字后缀的 admin.log 文件。

该命令只检查三位数字，所以相应文件的后缀不要超过 999。先建几个 admin.log*的文件，才能使用下面这个命令

```
$ find . -name "admin.log[0-9][0-9][0-9]" -atime -7 -ok
```

```
rm { } ;
```

```
< rm ... ./admin.log001 > ? n
```

```
< rm ... ./admin.log002 > ? n
```

```
< rm ... ./admin.log042 > ? n
```

```
< rm ... ./admin.log942 > ? n
```

7、为了查找当前文件系统中的所有目录并排序；

```
$ find . -type d | sort
```

8、为了查找系统中所有的 rmt 磁带设备；

```
$ find /dev/rmt -print
```

```
xargs
```

xargs - build and execute command lines from standard input

在使用 find 命令的-exec 选项处理匹配到的文件时，find 命令将所有匹配到的文件一起传递给 exec 执行。但有些系统对能够传递给 exec 的命令长度有限制，这样在 find 命令运行几分钟之后，就会出现 溢出错误。错误信息通常是“参数列太长”或“参数列溢出”。这就是 xargs 命令的用处所在，特别是与 find 命令一起使用。

find 命令把匹配到的文件传递给 xargs 命令，而 xargs 命令每次只获取一部分文件而不是全部，不像-exec 选项那样。这样它可以先处理最先获取的一部分文件，然后是下一批，并如此继续下去。

在有些系统中，使用-exec 选项会为处理每一个匹配到的文件而发起一个相应的进程，并非将匹配到的文件全部作为参数一次执行；这样在有些情况下就会出现进程过多，系统性能下降的问题，因而效率不高；

而使用 xargs 命令则只有一个进程。另外，在使用 xargs 命令时，究竟是一次获取所有的参数，还是分批取得参数，以及每一次获取参数的数目都会根据该命令的选项及系统内核中相应的可调参数来确定。

来看看 xargs 命令是如何同 find 命令一起使用的，并给出一些例子。

下面的例子查找系统中的每一个普通文件，然后使用 xargs 命令来测试它们分别属于哪类文件

```
#find . -type f -print | xargs file
```

```
./kde/Autostart/Autorun.desktop: UTF-8 Unicode English text
```

```
./kde/Autostart/.directory: ISO-8859 text
```

```
.....
```

在整个系统中查找内存信息转储文件(core dump)，然后把结果保存到/tmp/core.log 文件中：

```
$ find / -name "core" -print | xargs echo "" >/tmp/core.log
```

上面这个执行太慢，我改成在当前目录下查找

```
#find . -name "file*" -print | xargs echo "" > /temp/core.log
```

```
# cat /temp/core.log
```

```
./file6
```

在当前目录下查找所有用户具有读、写和执行权限的文件，并收回相应的写权限：

```
# ls -l
```

```
drwxrwxrwx 2 sam adm 4096 10 月 30 20:14 file6
```

```
-rwxrwxrwx 2 sam adm 0 10 月 31 01:01 http3.conf
```

```
-rwxrwxrwx 2 sam adm 0 10 月 31 01:01 httpd.conf
```

```
# find . -perm -7 -print | xargs chmod o-w
```

```
# ls -l
```

```
drwxrwxr-x 2 sam adm 4096 10 月 30 20:14 file6
```

```
-rwxrwxr-x 2 sam adm 0 10 月 31 01:01 http3.conf
```

```
-rwxrwxr-x 2 sam adm 0 10 月 31 01:01 httpd.conf
```

用 **grep** 命令在所有的普通文件中搜索 **hostname** 这个词：

```
# find . -type f -print | xargs grep "hostname"
```

```
./httpd1.conf:# different IP addresses or hostnames and have them handled by the
```

```
./httpd1.conf:# VirtualHost: If you want to maintain multiple domains/hostnames  
on your
```

用 **grep** 命令在当前目录下的所有普通文件中搜索 **hostnames** 这个词：

```
# find . -name * -type f -print | xargs grep "hostnames"
```

```
./httpd1.conf:# different IP addresses or hostnames and have them handled by the
```

```
./httpd1.conf:# VirtualHost: If you want to maintain multiple domains/hostnames  
on your
```

注意，在上面的例子中，`\` 用来取消 **find** 命令中的 ***** 在 **shell** 中的特殊含义。

find 命令配合使用 **exec** 和 **xargs** 可以使用户对所匹配到的文件执行几乎所有的命令。

find 命令的参数

下面是 **find** 命令一些常用参数的例子，有用到的时候查查就行了，像上面几个贴子，都用到了其中的一些参数，也可以用 **man** 或查看论坛里其它贴子有 **find** 命令手册

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 ***.txt** 的文件，使用 **~** 作为

'pathname'参数，波浪号~代表了你的\$HOME 目录。

```
$ find ~ -name "*.txt" -print
```

想要在当前目录及子目录中查找所有的 '*.txt'文件，可以用：

```
$ find . -name "*.txt" -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两个小写字母开头，跟着是两个数字，最后是.txt 的文件，下面的命令就能够返回名为 ax37.txt 的文件：

```
$ find . -name "[a-z][a-z][0-9][0-9].txt" -print
```

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：

```
$ find . -perm 755 -print
```

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，-006 相当于 666

```
# ls -l
```

```
-rwxrwxr-x 2 sam adm 0 10 月 31 01:01 http3.conf
```

```
-rw-rw-rw- 1 sam adm 34890 10 月 31 00:57 httpd1.conf
```

```
-rwxrwxr-x 2 sam adm 0 10 月 31 01:01 httpd.conf
```

```
drw-rw-rw- 2 gem group 4096 10 月 26 19:48 sam
```

```
-rw-rw-rw- 1 root root 2792 10 月 31 20:19 temp
```

```
# find . -perm 006
```

```
# find . -perm -006
```

```
./sam
```

```
./httpd1.conf
```

```
./temp
```

-perm mode:文件许可正好符合 mode

-perm +mode:文件许可部分符合 mode

-perm -mode: 文件许可完全符合 mode

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/apps目录下查找文件，但不希望在/apps/bin目录下查找，可以用：

```
$ find /apps -path "/apps/bin" -prune -o -print
```

4、使用 find 查找文件的时候怎么避开某个文件目录

比如要在/usr/sam目录下查找不在dir1子目录之内的所有文件

```
find /usr/sam -path "/usr/sam/dir1" -prune -o -print
```

find [-path ..] [expression] 在路径列表的后面的是表达式

-path "/usr/sam" -prune -o -print 是 **-path "/usr/sam" -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 "/usr/sam"** 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 "/usr/sam" -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 "/usr/sam" then
```

```
-prune
```

```
else
```

```
-print
```

避开多个文件夹

```
find /usr/sam ( -path /usr/sam/dir1 -o -path /usr/sam/file1 ) -prune -o -print
```

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。

查找某一确定文件，**-name** 等选项加在**-o**之后

```
#find /usr/sam (-path /usr/sam/dir1 -o -path /usr/sam/file1 ) -prune -o -name "temp" -print
```

5、使用 user 和 nouser 选项

按文件属主查找文件，如在\$HOME目录中查找文件属主为sam的文件，可以用：

```
$ find ~ -user sam -print
```

在/etc目录下查找文件属主为uucp的文件：

```
$ find /etc -user uucp -print
```

为了查找属主帐户已经被删除的文件，可以使用**-nouser**选项。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。

例如，希望在 **/home** 目录下查找所有的这类文件，可以用：

```
$ find /home -nouser -print
```

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。

查找更改时间比文件 **sam** 新但比文件 **temp** 旧的文件：

例：有两个文件

```
-rw-r--r-- 1 sam adm 0 10 月 31 01:07 fiel
```

```
-rw-rw-rw- 1 sam adm 34890 10 月 31 00:57 httpd1.conf
```

```
-rwxrwxr-x 2 sam adm 0 10 月 31 01:01 httpd.conf
```

```
drw-rw-rw- 2 gem group 4096 10 月 26 19:48 sam
```

```
-rw-rw-rw- 1 root root 2792 10 月 31 20:19 temp
```

```
# find -newer httpd1.conf ! -newer temp -ls
```

```
1077669 0 -rwxrwxr-x 2 sam adm 0 10 月 31 01:01 ./httpd.conf
```

```
1077671 4 -rw-rw-rw- 1 root root 2792 10 月 31 20:19 ./temp
```



```
1077673 0 -rw-r--r-- 1 sam adm 0 10 月 31 01:07 ./fiel
```

查找更改时间在比 **temp** 文件新的文件：

```
$ find . -newer temp -print
```

9、使用 **type** 选项

在/etc 目录下查找所有的目录，可以用：

```
$ find /etc -type d -print
```

在当前目录下查找除目录以外的所有类型的文件，可以用：

```
$ find . ! -type d -print
```

在/etc 目录下查找所有的符号链接文件，可以用

```
$ 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。

在当前目录下查找文件长度大于 **1 M** 字节的文件：

```
$ find . -size +1000000c -print
```

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

```
$ find /home/apache -size 100c -print
```

在当前目录下查找长度超过 **10** 块的文件（一块等于 **512** 字节）：

```
$ 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。

在下面的例子中，**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。

它将首先匹配所有的文件然后再进入子目录中查找。

```
$ find / -name "CON.FILE" -depth -print
```

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。

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：

```
$ find . -name "*.XC" -mount -print
```