[**每天一个linux命令（20）：find命令之exec**](http://www.cnblogs.com/peida/archive/2012/11/14/2769248.html)

find是我们很常用的一个Linux命令，但是我们一般查找出来的并不仅仅是看看而已，还会有进一步的操作，这个时候exec的作用就显现出来了。

**exec解释：**

-exec  参数后面跟的是command命令，它的终止是以;为结束标志的，所以这句命令后面的分号是不可缺少的，考虑到各个系统中分号会有不同的意义，所以前面加反斜杠。

{}   花括号代表前面find查找出来的文件名。

使用find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查找，很方便的。在有些操作系统中只允许-exec选项执行诸如l s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。

**实例1：ls -l命令放在find命令的-exec选项中**

**命令：**

find . -type f -exec ls -l {} \;

**输出：**

[root@localhost test]# find . -type f -exec ls -l {} \;

-rw-r--r-- 1 root root 127 10-28 16:51 ./log2014.log

-rw-r--r-- 1 root root 0 10-28 14:47 ./test4/log3-2.log

-rw-r--r-- 1 root root 0 10-28 14:47 ./test4/log3-3.log

-rw-r--r-- 1 root root 0 10-28 14:47 ./test4/log3-1.log

-rw-r--r-- 1 root root 33 10-28 16:54 ./log2013.log

-rw-r--r-- 1 root root 302108 11-03 06:19 ./log2012.log

-rw-r--r-- 1 root root 25 10-28 17:02 ./log.log

-rw-r--r-- 1 root root 37 10-28 17:07 ./log.txt

-rw-r--r-- 1 root root 0 10-28 14:47 ./test3/log3-2.log

-rw-r--r-- 1 root root 0 10-28 14:47 ./test3/log3-3.log

-rw-r--r-- 1 root root 0 10-28 14:47 ./test3/log3-1.log

[root@localhost test]#

**说明：**

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。

**实例2：在目录中查找更改时间在n日以前的文件并删除它们**

**命令：**

find . -type f -mtime +14 -exec rm {} \;

**输出：**

[root@localhost test]# ll

总计 328

-rw-r--r-- 1 root root 302108 11-03 06:19 log2012.log

-rw-r--r-- 1 root root     33 10-28 16:54 log2013.log

-rw-r--r-- 1 root root    127 10-28 16:51 log2014.log

lrwxrwxrwx 1 root root      7 10-28 15:18 log\_link.log -> log.log

-rw-r--r-- 1 root root     25 10-28 17:02 log.log

-rw-r--r-- 1 root root     37 10-28 17:07 log.txt

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

drwxrwxrwx 2 root root   4096 10-28 14:47 test3

drwxrwxrwx 2 root root   4096 10-28 14:47 test4

[root@localhost test]# find . -type f -mtime +14 -exec rm {} \;

[root@localhost test]# ll

总计 312

-rw-r--r-- 1 root root 302108 11-03 06:19 log2012.log

lrwxrwxrwx 1 root root      7 10-28 15:18 log\_link.log -> log.log

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

drwxrwxrwx 2 root root   4096 11-12 19:32 test3

drwxrwxrwx 2 root root   4096 11-12 19:32 test4

[root@localhost test]#

**说明：**

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。

**实例3：在目录中查找更改时间在n日以前的文件并删除它们，在删除之前先给出提示**

**命令：**

find . -name "\*.log" -mtime +5 -ok rm {} \;

**输出：**

[root@localhost test]# ll

总计 312

-rw-r--r-- 1 root root 302108 11-03 06:19 log2012.log

lrwxrwxrwx 1 root root      7 10-28 15:18 log\_link.log -> log.log

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

drwxrwxrwx 2 root root   4096 11-12 19:32 test3

drwxrwxrwx 2 root root   4096 11-12 19:32 test4

[root@localhost test]# find . -name "\*.log" -mtime +5 -ok rm {} \;

< rm ... ./log\_link.log > ? y

< rm ... ./log2012.log > ? n

[root@localhost test]# ll

总计 312

-rw-r--r-- 1 root root 302108 11-03 06:19 log2012.log

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

drwxrwxrwx 2 root root   4096 11-12 19:32 test3

drwxrwxrwx 2 root root   4096 11-12 19:32 test4

[root@localhost test]#

**说明**：

在上面的例子中， 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。 按y键删除文件，按n键不删除。

**实例4：-exec中使用grep命令**

**命令：**

find /etc -name "passwd\*" -exec grep "root" {} \;

**输出：**

[root@localhost test]# find /etc -name "passwd\*" -exec grep "root" {} \;

root:x:0:0:root:/root:/bin/bash

root:x:0:0:root:/root:/bin/bash

[root@localhost test]#

**说明：**

任何形式的命令都可以在-exec选项中使用。  在上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root用户。

**实例5：查找文件移动到指定目录**

**命令：**

find . -name "\*.log" -exec mv {} .. \;

**输出：**

[root@localhost test]# ll

总计 12drwxr-xr-x 6 root root 4096 10-27 01:58 scf

drwxrwxr-x 2 root root 4096 11-12 22:49 test3

drwxrwxr-x 2 root root 4096 11-12 19:32 test4

[root@localhost test]# cd test3/

[root@localhost test3]# ll

总计 304

-rw-r--r-- 1 root root 302108 11-03 06:19 log2012.log

-rw-r--r-- 1 root root     61 11-12 22:44 log2013.log

-rw-r--r-- 1 root root      0 11-12 22:25 log2014.log

[root@localhost test3]# find . -name "\*.log" -exec mv {} .. \;

[root@localhost test3]# ll

总计 0[root@localhost test3]# cd ..

[root@localhost test]# ll

总计 316

-rw-r--r-- 1 root root 302108 11-03 06:19 log2012.log

-rw-r--r-- 1 root root     61 11-12 22:44 log2013.log

-rw-r--r-- 1 root root      0 11-12 22:25 log2014.log

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

drwxrwxr-x 2 root root   4096 11-12 22:50 test3

drwxrwxr-x 2 root root   4096 11-12 19:32 test4

[root@localhost test]#

**实例6：用exec选项执行cp命令**

**命令：**

find . -name "\*.log" -exec cp {} test3 \;

**输出：**

[root@localhost test3]# ll

总计 0[root@localhost test3]# cd ..

[root@localhost test]# ll

总计 316

-rw-r--r-- 1 root root 302108 11-03 06:19 log2012.log

-rw-r--r-- 1 root root     61 11-12 22:44 log2013.log

-rw-r--r-- 1 root root      0 11-12 22:25 log2014.log

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

drwxrwxr-x 2 root root   4096 11-12 22:50 test3

drwxrwxr-x 2 root root   4096 11-12 19:32 test4

[root@localhost test]# find . -name "\*.log" -exec cp {} test3 \;

cp: “./test3/log2014.log” 及 “test3/log2014.log” 为同一文件

cp: “./test3/log2013.log” 及 “test3/log2013.log” 为同一文件

cp: “./test3/log2012.log” 及 “test3/log2012.log” 为同一文件

[root@localhost test]# cd test3

[root@localhost test3]# ll

总计 304

-rw-r--r-- 1 root root 302108 11-12 22:54 log2012.log

-rw-r--r-- 1 root root     61 11-12 22:54 log2013.log

-rw-r--r-- 1 root root      0 11-12 22:54 log2014.log

[root@localhost test3]#