

find命令的使用

❑ **find**: 搜索文件，并执行指定的操作。

find [起始目录] 寻找条件 [操作]

◆ 常用选项：以名称和文件属性查找

- name** 字符串 查找文件名匹配所给字符串的所有文件，
字符串内可用通配符 *、?、[]
- lname** 字符串 查找文件名匹配所给字符串的所有符号链接文件，
字符串内可用通配符 *、?、[]
- gid** n 查找属于ID号为 n 的用户组的所有文件。
- uid** n 查找属于ID号为 n 的用户的所有文件。
- group** 字符串 查找属于用户组名为所给字符串的所有的文件。
- user** 字符串 查找属于用户名为所给字符串的所有的文件。

以名称和文件属性查找

- empty** 查找大小为 0 的目录或文件
- size n[bckw]** 查找指定文件大小的文件，**n** 后面的字符表示单位，缺省为 **b**，代表 **512** 字节的块
- type x** 查找类型为 **x** 的文件，**x** 为下列字符之一：
 - b** 块设备文件；**c** 字符设备文件；**d** 目录文件；
 - p** 命名管道；**f** 普通文件；**l** 符号链接；**s** socket 文件

注：以上选项中的 **n** 可以有三种输入方式：**n**, **+n**, **-n**

- **20** 表示等于 20
- **+20** 表示大于 20（21，22，23 等）
- **-20** 表示小于 20（19，18，17 等）

以时间为条件查找

◆ 常用选项：以时间为条件查找

- amin n 查找 n 分钟前被访问过的文件或目录
- atime n 查找 n 天前被访问过的文件或目录
- anewer <参考文件或目录> 查找其访问时间比指定文件或目录的访问时间更接近现在的文件或目录
- cmin n 查找 n 分钟前文件状态被修改过的文件或目录
- ctime n 查找 n 天前文件状态被修改过的文件或目录
- cnewer <参考文件或目录> 查找其文件或目录的文件状态修改时间比指定文件或目录的文件状态修改时间更接近现在的文件或目录
- mmin n 查找 n 分钟前文件内容被修改过的文件或目录
- mtime n 查找 n 天前文件内容被修改过的文件或目录
- newer <参考文件或目录> 查找其更改时间较指定文件或目录的更改时间更接近现在的文件或目录

选项中可以使用的
n, +n, -n

可以执行的操作

◆ 可以执行的操作

- exec** 命令名称 **{}** 对符合条件的文件执行所给的 Linux 命令，
{} 表示所找到的文件，作为命令的参数；
命令的末尾必须以 **\;** 结束
- ok** 命令名称 **{}** 对符合条件的文件执行所给的 Linux 命令，
与 **exec** 的区别： 它会询问用户是否要执行该命令
- ls** 详细列出所找到的所有文件
- fprint** 文件名 将找到的文件名写入指定文件
- print** 在标准输出设备上显示查找出的文件名
- printf** 格式 按照指定的输出，格式的具体写法请参考 C 语言

find 简单用法举例

◆ find 的简单用法举例

```
find . -name 'mem*'
```

- 查找当前目录 (含所有子目录)中以 **mem** 开头的所有文件和目录
- 使用通配符时需要加单引号或双引号

```
find . -name '[Mm]em*'
```

```
find . -name '[Mm]*[0-9]'
```

```
find log -empty
```

- 查找 **log** 子目录中所有的空文件或空目录

find 简单用法举例

```
find log -mtime 3
```

- 查找 **log** 子目录中，被修改时间距离当前 超过 **3*24** 小时，但不超过 **4*24** 小时 的所有文件和目录

```
find log -mtime +3
```

- 查找 **log** 子目录中，被修改时间距离当前 超过 **4*24** 小时 的所有文件和目录

```
find log -mmin 30
```

```
find log -mmin +30
```

```
find log -mmin -30
```

find 多个条件举例

◆ 利用多个条件查询时，可以使用逻辑运算

- 逻辑与：在命令中用 **-a** 表示，是系统缺省的选项

```
find . -name 'mem*' -a -type f
```

```
find . -name 'mem*' -type f
```

- 逻辑或：在命令中用 **-o** 表示

```
find . -name 'mem*' -o -name 'Mem*'
```

- 逻辑非：在命令中用 **!** 表示

```
find . ! -name 'mem*'
```

find 多个条件举例

```
find . ! -name 'mem*' -a -name 'Mem*'
```

- 当使用多个逻辑选项时，可以用括号把这些选项括起来。为了避免 **Shell** 本身对括号引起误解，在括号前需要加转义字符 **** 来去除括号的特殊意义

```
find . ! \( -name 'mem*' -a -name 'Mem*' \)
```

```
find . ! \( -name 'mem*' -o -name 'Mem*' \)
```

find 带操作用法举例

◆ 找出符合条件的对象，并对它们执行指定的操作

```
find . -name 'mem*' -exec ls -l {} \;
```

- 查找当前目录 (含所有子目录)中以 **mem** 开头的文件和目录，并以长格式显示
- **{}** 两个大括号之间不能有空格，表示查找到的对象
- **\;** 表示命令结束，反斜杠与前面的大括号之间必须要留空格

```
find log -mtime 3 -ls
```

```
find . -name 'mem*' -exec ls -F {} \;
```

find 带操作用法举例

```
find log -mtime 3 -exec cp {} ~/tmp \;
```

- 将符合要求的对象复制到 `~/tmp` 目录中

```
find log -mtime 3 -fprint output.txt
```

- 将结果输出到文件 `output.txt` 中

```
find . -empty -ok rm {} \;
```

- 执行删除操作前要求用户确认