

find

内容提要

1. 熟悉 `find` 命令的格式
2. 掌握 `find` 命令中各种表达式的书写方法
3. 学会使用 `find` 命令查找文件

find 命令的格式

`find` [<http://www.proxyservice.net/index.php?q=aHR0cDovL21hbi5jeC9maW5k>] 命令用于在文件系统中查找满足条件的文件。`find` 命令功能强大，提供了相当多的查找条件。`find` 命令还可以对查找到的文件做操作，如执行 Shell 命令等。

`find` 命令的格式是：

```
find [<起始目录> ...] [<选项表达式>] [<条件匹配表达式>] [<动作表达式>]
```

其中：

- **<起始目录>**：对每个指定的 **<起始目录>** 递归搜索目录树
 - 若在整个文件系统范围内查找，则起始目录是`/`
 - 若在当前目录下寻找，则起始目录是`.`，省略**<起始目录>**表示当前目录
- **<选项表达式>**：控制 `find` 命令的行为
- **<条件匹配表达式>**：根据匹配条件查找文件
- **<动作表达式>**：指定对查找结果的操作，默认为显示在标准输出（`-print`）

不带任何参数的 `find` 命令将在屏幕上递归显示当前目录下的文件列表。下面给出一些常用的表达式的解释。

选项表达式

| 表达式 | 说明 |
|-------------------------------|--|
| <code>-follow</code> | 如果遇到符号链接文件，就跟踪链接所指的文件 |
| <code>-regextype TYPE</code> | 指定 <code>-regex</code> 和 <code>-iregex</code> 使用的正则表达式类型，默认为 <code>emacs</code> ，还可选择 <code>posix-awk</code> , <code>posix-basic</code> , <code>posix-egrep</code> 和 <code>posix-extended</code> |
| <code>-depth</code> | 查找进入子目录前先查找完当前目录的文件 |
| <code>-mount</code> | 查找文件时不跨越文件系统 |
| <code>-xdev</code> | 查找文件时不跨越文件系统 |
| <code>-maxdepth LEVELS</code> | 设置最大的查找深度 |
| <code>--help</code> | 显示 <code>find</code> 命令帮助信息 |
| <code>--version</code> | 显示 <code>find</code> 的版本 |

条件匹配表达式

| 表达式 | 说明 |
|---------------------------|---|
| -name PATTERN | 匹配文件名 |
| -iname PATTERN | 匹配文件名（忽略大小写） |
| -lname PATTERN | 匹配符号链接文件名 |
| -ilname PATTERN | 匹配符号链接文件名（忽略大小写） |
| -path PATTERN | 匹配文件的完整路径（不把 '/' 和 '.' 作为特殊字符） |
| | PATTERN 使用 Shell 的匹配模式，可以使用 Shell 的通配符（*、? []），要用"或"括起来 |
| 表达式 | 说明 |
| -regex PATTERN | 以正则表达式匹配文件名 |
| -iregex PATTERN | 以正则表达式匹配文件名（忽略大小写） |
| 表达式 | 说明 |
| -amin N | 查找 N 分钟以前被访问过的所有文件 |
| -atime N | 查找 N 天以前被访问过的所有文件 |
| -cmin N | 查找 N 分钟以前文件状态被修改过的所有文件（比如权限修改） |
| -ctime N | 查找 N 天以前文件状态被修改过的所有文件（比如权限修改） |
| -mmin N | 查找 N 分钟以前文件内容被修改过的所有文件 |
| -mtime N | 查找 N 天以前文件内容被修改过的所有文件 |
| -uid N | 查找属于 ID 号为 N 用户的所有文件 |
| -gid N | 查找属于 ID 号为 N 组的所有文件 |
| -inum N | 查找 i-node 是 N 的文件 |
| -links N | 查找硬链接数为 N 的文件 |
| -size N[bcwkMG] | 查找大小为 N 的文件，b(块)默认单位；c(字节)；w(双字节) |
| | N 可以有三种输入方式，+N 或 -N 或 N。假设 N 为 20，则：(1) +20 ：表示20以上（21，22，23等）；(2) -20 ：表示20以内（19，18，17等）；(3) 20 ：表示正好是20。 |
| 表达式 | 说明 |
| -perm MODE | 精确匹配权限模式为 MODE 的文件。MODE：与 chown 命令的书写方式一致，既可以使用字符模式也可以使用8进制模式 |
| -perm -MODE | 匹配权限模式至少为 MODE 的文件 |
| 表达式 | 说明 |
| -anewer FILE | 查找所有比 FILE 的访问时间新的文件 |
| -cnewer FILE | 查找所有比 FILE 的状态修改时间新的文件（比如权限修改） |
| -newer FILE | 查找所有比 FILE 的内容修改时间新的文件 |
| -samefile FILE | 查找与 FILE 具有相同 i-node 的文件（硬链接） |
| 表达式 | 说明 |
| -fstype TYPE | 只查找指定类型的文件系统 |
| -type [bcdpfls] | 查找指定类型的文件 [块设备, 字符设备, 目录, 管道, 普通文件, 符号链接, socket套接字] |
| -empty | 内容为空的文件 |
| -user NAME | 查找用户名为 NAME 的所有文件 |
| -group NAME | 查找组名为 NAME 的所有文件 |
| -nouser | 文件属于不在 /etc/passwd 文件中的用户 |

| | |
|----------|--------------------------------------|
| -nogroup | 文件属于不在 <code>/etc/group</code> 文件中的组 |
|----------|--------------------------------------|

动作表达式

| 表 达 式 | 说 明 |
|---------------------|--|
| -print | 在标准输出上列出查找结果（每行一个文件） |
| -print0 | 在标准输出上列出查找结果（取消间隔符） 同样与 <code> xargs -0</code> 连用 |
| -fprint FILE | 与 <code>-print</code> 一致，只是输出到文件 <code>FILE</code> |
| -fprint0 FILE | 与 <code>print0</code> 一致，只是输出到文件 <code>FILE</code> |
| -ls | 使用 <code>'ls -dils'</code> 在标准输出上列出查找结果 |
| -fls FILE | 与 <code>-ls</code> 一致，只是输出到文件 <code>FILE</code> |
| -prune | 忽略对某个目录的查找 |
| -exec COMMAND {} \; | 对符合查找条件的文件执行 Linux 命令 |
| -ok COMMAND {} \; | 对符合查找条件的文件执行 Linux 命令；与 <code>-exec</code> 不同的是，它会询问用户是否需要执行 |

组合条件表达式

在书写表达式时，可以使用逻辑运算符与、或、非组成的复合条件，并可以用`()`改变默认的操作符优先级。下面以优先级由高到低列出可用的逻辑操作符。若以空格作为各个表达式的间隔符，则各个表示式之间是与关系。

| 操 作 符 | 说 明 |
|----------------|---|
| (EXPR) | 改变操作符优先次序，一些 UNIX 版的 <code>find</code> 命令要使用 <code>\(EXPR \)</code> 形式 |
| ! EXPR | 表示对表达式取反 |
| EXPR1 EXPR2 | 与逻辑，若 <code>EXPR1</code> 为假，将不再评估 <code>EXPR2</code> |
| EXPR1 -a EXPR2 | 与 <code>EXPR1 EXPR2</code> 功能一致 |
| EXPR1 -o EXPR2 | 逻辑或，若 <code>EXPR1</code> 为真，将不再评估 <code>EXPR2</code> |
| EXPR1 , EXPR2 | 若 <code>EXPR1</code> 为假，继续评估 <code>EXPR2</code> |

find 命令使用举例

| |
|---|
| <pre>##### ### find 的版本和使用帮助信息 ##### \$ find --help # 显示 find 命令帮助信息 \$ find --version # 显示 find 的版本 ##### ### 不指定匹配表达式，显示所有文件 ##### # 递归显示当前目录的文件列表 \$ find # 递归显示 / 目录的文件列表 \$ find / # 递归显示 / 目录的文件列表（仅限于3层目录） \$ find / -maxdepth 3 # 递归显示 / 目录的文件列表（仅限于 / 文件系统） \$ find / -xdev # 递归显示 /home、/www、/srv 目录的文件列表 \$ find /home /www /srv ##### ### 按文件名/路径名查找 ##### # 查找特定的文件名 \$ find -name myfile \$ find -maxdepth 2 -name symfony # 使用通配符查找特定的文件名</pre> |
|---|

```
$ find -name 'd*'
$ find -name '???'
$ find -name '[afd]*'
$ find -iname '[a-z]*'
$ find -name 'ch[0-2][0-9].txt*'
# 匹配文件路径名
$ find -path '*server'
./vbird/server
./server
$ find -path '*server[12]'
./server1
./server2
./server1/server2
./server2/server2
# 以正则表达式匹配文件路径名
$ find -regex '.*'
$ find -regex '.*ch0.*'
./ch01
./ch00
./vbird/server/1000results/ch09-01.jpg
$ find -regex '.*ch[0-9]+'
./ch01
./ch21
./ch00
./ch333
./ch1
./ch41
$ find -regex '.*ch[0-9]+\.txt'
./ch1.txt
./ch24.txt
#####
### 按文件属性查找
#####
# 只查找普通文件
$ find . -type f
# 只查找符号链接文件
$ find . -type l
# 查找硬连接数大于 1 的文件或目录
$ find /home -links +1
# 查找 /tmp 目录下小于 10M 的文件
$ find /tmp -size -10M
# 查找 /home 目录下大于 1G 的文件
$ find /home -size +1G
# 查找系统中为空的文件或者目录
$ find / -empty
# 查找在 /www 中最后10分钟访问过的文件
$ find /www -amin -10
# 查找在 /www 中最后2天访问过的文件
$ find /www -atime -2
# 查找在 /home 下最近2天内改动过的文件
$ find /home -mtime -2
# 列出被改动过后 2 日内被存取过的文件或目录
$ find /home -used -2
# 列出被改动过后 90 日前被存取过的文件或目录
$ find /home -used +90
# 列出 /home 目录中属于用户 osmond 的文件或目录
$ find /home -user osmond
# 列出 /home 目录中 UID 大于 501 的文件或目录
$ find /home -uid +501
# 列出 /home 目录中组为 osmond 的文件或目录
$ find /home -group osmond
# 列出 /home 目录中 GID 为 501 的文件或目录
$ find /home -gid 501
# 列出 /home 目录中不属于本地用户的文件或目录
$ find /home -nouser
# 列出 /home 目录中不属于本地组的文件或目录
$ find /home -nogroup
# 精确查找权限为 664 的文件或目录
$ find . -perm 664
# 查找权限至少为 664 的文件或目录
$ find . -perm -664
```

```
#####  
### 使用逻辑运算构造复杂表达式  
#####  
# 查找 /tmp 目录下21天之前访问过的大于 10G 的文件  
$ find /tmp -size +10M -a -atime +21  
# 查找 / 目录下属主为 jjheng 或 osmond 的文件  
$ find / -user jjheng -o -user osmond  
# 查找 /tmp 目录下的属主不是 osmond 的文件  
$ find /tmp ! -user osmond  
# 在 /mnt 下查找 *.txt 且文件系统类型不为 vfat 的文件  
$ find /mnt -name '*.txt' ! -fstype vfat  
# 在 /tmp 下查找名为 l 开头且类型为符号链接的文件  
$ find /tmp -name 'l*' -type l  
# 查找以 server 开头的目录名  
$ find . -type d -name 'server*'  
# 找出 /var/log 目录下所有的前5天修改过的.log 文件  
$ find /var/log -name '*.log' -mtime +5  
#####  
### 按文件样本查找  
#####  
# 查找所有比 FILE1 的访问时间新的文件  
$ find -anewer FILE1  
# 查找所有比 FILE2 的访问时间旧的文件  
$ find ! -anewer FILE2  
# 查找所有比 FILE1 的访问时间新的  
# 且比 FILE2 的访问时间旧的文件  
$ find -anewer FILE1 ! -anewer FILE2  
# 查找所有比 FILE1 的内容修改时间新的文件  
$ find -newer FILE1  
# 查找所有比 FILE2 的内容修改时间旧的文件  
$ find ! -newer FILE2  
# 查找所有比 FILE1 的内容修改时间新的  
# 且比 FILE2 的内容修改时间旧的文件  
$ find -newer FILE1 ! -newer FILE2  
# 查找与 FILE 具有相同 i-node 的文件（硬链接）  
$ find -samefile FILE -ls  
#####  
### 对查找到的文件实施命令操作  
#####  
# 查找并列出当前目录下不安全的文件（世界可读写执行）  
$ find . -perm -007 -exec ls -l {} \  
# 查找 logs 目录下的所有的 .log 文件并查看它的详细信息  
$ find logs -name "*.log" -type f -exec ls -l {} \  
# 查找当天修改过的普通文件  
$ find . -type f -mtime -1 -exec ls -l {} \  
# 查找当前目录下的.php文件并用grep过滤出包含include的行  
$ find . -name "*.php" -exec grep "include" {} \  
# 查找并删除当前目录及其子目录下所有扩展名为 .tmp 的文件  
$ find . -name '*.tmp' -exec rm {} \  
# 在logs目录中查找7天之内未修改过的文件并在删除前询问  
$ find logs -type f -mtime +7 -exec -ok rm {} \  
# 查询并删除一周以来从未访问过的以 .o 结尾或名为 a.out  
# 且不存在于 nfs 文件系统中的所有文件  
$ find / ( -name a.out -o -name '*.o' ) -atime +7 \  
! -fstype nfs -exec rm {} \  
# 查询并删除当前目录及其子目录下所有的空目录  
$ find . -depth -type d -empty -exec rmdir {} \  
# 将default目录下的文件由GBK编码转换为UTF-8编码  
# 目录结构不变，转码后的文件保存在utf/default目录下  
# From: http://www.xiaojb.com/archives/it/convert-gbk-utf-8.shtml  
$ find default -type d -exec mkdir -p utf/{} \  
$ find default -type f -exec iconv -f GBK -t UTF-8 {} -o utf/{} \  
# 下面 find 命令的书写形式均等价  
$ find -name \*.sh -exec cp {} /tmp \  
$ find -name '*.sh' -exec cp {} /tmp ';'\  
$ find -name "*.sh" -exec cp {} /tmp "\";  
$ find -name \*.sh -exec cp {\} /tmp \  
$ find -name '*.sh' -exec cp '{}' /tmp ';'\  
$ find -name "*.sh" -exec cp "{}" /tmp "\";  
#####  
### 在查找中排除指定的目录
```

```
#####
# 显示当前目录树
$ tree -F -L 2
.
|-- bin/
|   |-- switch-lang.sh*
|   `-- sys2wiki.sh*
|-- book/
|   |-- basic/
|   |-- basic-utf8/
|   |-- basic.zip
|   |-- server/
|   |-- server-utf8/
|   |-- server.zip
|   `-- to-zh-CN-utf8.sh*
`-- bak.sh*
# 显示当前目录下除 book 目录的所有文件
$ find . -name book -prune -o -print
# 查找当前目录下（除了 book 目录）的所有 .sh 文件
$ find . -name book -prune -o -name '*.sh' -print
# 显示当前目录下除 book/server 目录的所有文件
$ find . -path ./book/server -prune -o -print
# 使用绝对路径完成上述任务
$ find /home/osmond -path /home/osmond/book/server -prune -o -print
# 查找当前目录下（除了 book/server 目录）的所有 .sh 文件
$ find . -path ./book/server -prune -o -name '*.sh' -print
# 显示当前目录下除 book/server 和 book/server-utf8 目录的所有文件
$ find . -path './book/server*' -prune -o -print
# 查找当前目录下（除了 book/server 和 book/server-utf8 目录）的所有 .sh 文件
$ find . -path './book/server*' -prune -o -name '*.sh' -print
# 显示当前目录下除 book/server 和 book/basic 目录的所有文件
$ find . \( -path ./book/server -o -path ./book/basic \) -prune -o -print
# 查找当前目录下（除了 book/server 和 book/basic 目录）的所有 .sh 文件
$ find . \( -path ./book/server -o -path ./book/basic \) -prune -o -name '*.sh' -print
```

- 显示源文件
- 登录