

find 快速参考

按名称、类型、大小、时间、权限搜索并执行操作

基础搜索

运行 find

```
find . # list all files recursively
find /var/log # search from specific path
find . -name "*.txt" # find by name
find / -name "config" 2>/dev/null # suppress permission errors
find dir1 dir2 -name "*.py" # search multiple directories
```

语法概览

find [path...] [tests] [actions]	通用格式——路径、条件、动作
path	起始目录（默认：当前目录）
test	过滤文件的条件（ -name 、 -type 等）
action	对匹配项执行的操作（ -print 、 -exec 等）
Default action	未指定动作时默认 -print

名称模式

名称匹配

```
find . -name "*.log" # case-sensitive glob
find . -iname "readme*" # case-insensitive glob
find . -name "*.py" -o -name "*.js" # OR: Python or JS files
find . -path "*/src/*.ts" # match against full path
find . -regex '.*\.(py|js\)' # POSIX regex on full path
```

名称条件

-name pattern	文件名匹配 shell glob（区分大小写）
-iname pattern	文件名匹配 glob（不区分大小写）
-path pattern	完整路径匹配 shell glob
-ipath pattern	完整路径匹配 glob（不区分大小写）
-regex pattern	完整路径匹配正则表达式
-iregex pattern	完整路径匹配正则（不区分大小写）

类型过滤

按类型过滤

```
find . -type f # regular files only
find . -type d # directories only
find . -type l # symbolic links
find . -type f -name "*.sh" # combine type + name
```

文件类型

-type f	普通文件
-type d	目录
-type l	符号链接
-type b	块设备
-type c	字符设备
-type p	命名管道 (FIFO)
-type s	Socket
-empty	空文件或空目录

大小与时间

大小与时间示例

```
find . -size +100M # larger than 100 MB
find . -size -1k # smaller than 1 KB
find . -mtime -7 # modified in last 7 days
find . -mmin -30 # modified in last 30 minutes
find . -newer reference.txt # newer than reference file
```

大小与时间条件

-size +/-Nc	字节(c)、KB(k)、MB(M)、GB(G)
-mtime +/-N	N×24 小时前修改 (+ 更早, - 更新)
-atime +/-N	N×24 小时前访问
-ctime +/-N	N×24 小时前状态变更
-mmin +/-N	N 分钟前修改
-newer file	比指定文件更新
-newermt date	在指定日期后修改 (GNU)

权限

权限示例

```
find . -perm 644 # exact permissions: rw-r--r--
find . -perm -u+x # user has execute bit set
find . -perm /o+w # others have write (any match)
find . -user root # owned by root
find . -group www-data -type f # owned by group
```

权限条件

-perm mode	精确权限匹配
-perm -mode	所有指定位均已设置
-perm /mode	任一指定位已设置
-user name	属于用户 (名称或 UID)
-group name	属于用户组 (名称或 GID)
-nouser	/etc/passwd 中无对应应用户
-nogroup	/etc/group 中无对应组

动作

动作示例

```
find . -name "*.log" -print # print paths (default)
find . -name "*.tmp" -delete # delete matching files
find . -type f -ls # detailed listing
find . -name "*.txt" -print0 # null-delimited output
find . -type f -printf "%p %s\n" # custom format (GNU)
```

动作参考

-print	打印路径（换行分隔）
-print0	打印路径（null 分隔，适合 xargs）
-ls	打印文件详情（类似 ls -dils ）
-delete	删除匹配文件（隐含 -depth ）
-printf format	自定义输出 (GNU)：%p 路径, %s 大小, %t 时间
-fprint file	将路径写入文件
-quit	第一次匹配后退出

组合条件

逻辑运算符

```
find . -name "*.py" -type f # implicit AND
find . -name "*.py" -a -size +10k # explicit AND
find . -name "*.py" -o -name "*.js" # OR
find . ! -name "*.log" # NOT
find . \( -name "*.py" -o -name "*.js" \) -type f
```

运算符参考

expr1 expr2 / expr1 -a expr2	AND——两者为真（默认）
expr1 -o expr2	OR——其中一个为真
! expr / -not expr	NOT——取反
\(expr \)	分组表达式（shell 需转义括号）
Evaluation order	从左到右； -a 优先级高于 -o

Exec 与删除

Exec 示例

```
find . -name "*.sh" -exec chmod +x {} \;
find . -name "*.log" -exec rm {} +
find . -type f -exec grep -l "TODO" {} +
find . -name "*.bak" -ok rm {} \; # prompt before each
find . -name "*.tmp" -print0 | xargs -0 rm
```

Exec 参考

-exec cmd {} \;	每个文件执行一次（{} = 文件路径）
-exec cmd {} +	批量传入多个文件执行（更快）
-ok cmd {} \;	类似 -exec ，但执行前提示确认
-execdir cmd {} \;	在文件所在目录执行命令
xargs -0	与 -print0 配合，安全批量处理
-delete	删除文件；先处理最深层

深度与裁剪

深度与裁剪示例

```
find . -maxdepth 1 -type f # current dir only
find . -mindepth 2 -name "*.py" # skip top-level
find . -name ".git" -prune -o -print # skip .git dirs
find . -depth -name "*.tmp" -delete # process children first
```

深度选项

-maxdepth N	最多下降 N 层（0 = 仅起始路径）
-mindepth N	不对 N 层以上应用条件
-depth	先处理目录内容，再处理目录本身
-prune	不进入匹配的目录
-mount / -xdev	不跨文件系统边界
-follow / -L	跟随符号链接

常用模式

一行命令

```
find . -name "*.pyc" -delete # clean Python bytecode
find . -type f -size 0 -delete # remove empty files
find . -mtime +30 -name "*.log" -delete # purge old logs
find . -type f -name "*.md" | wc -l # count Markdown files
find . -type d -empty -delete # remove empty dirs
```

实用示例

找最大文件	find . -type f -printf '%s %p\n' sort -rn head
按名找重复	find . -type f awk -F/ '{print \$NF}' sort uniq -d
批量改扩展名	find . -name "*.txt" -exec rename 's/.txt/.md/' {} +
找断裂符号链接	find . -xtype l
打包近期文件	find . -mtime -7 -print0 tar czf recent.tar.gz --null -T -
搜索代码文件	find . -name '*.py' -exec grep -l 'pattern' {} +