

find命令用来在指定目录下查找文件。任何位于参数之前的字符串都将被视为欲查找的目录名。如果使用该命令时，不设置任何参数，则find命令将在当前目录下查找子目录与文件。并且将查找到的子目录和文件全部进行显示。

语法

find(选项)(参数)

选项

- amin<分钟>：查找在指定时间曾被存取过的文件或目录，单位以分钟计算；
- anewer<参考文件或目录>：查找其存取时间较指定文件或目录的存取时间更接近现在的文件或目录；
- atime<24小时数>：查找在指定时间曾被存取过的文件或目录，单位以24小时计算；
- cmin<分钟>：查找在指定时间之时被更改过的文件或目录；
- cnewer<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；
- ctime<24小时数>：查找在指定时间之时被更改的文件或目录，单位以24小时计算；
- daystart：从本日开始计算时间；
- depth：从指定目录下最深层的子目录开始查找；
- empty：寻找文件大小为0 Byte的文件，或目录下没有任何子目录或文件的空目录；
- exec<执行指令>：假设find指令的回传值为True，就执行该指令；
- false：将find指令的回传值皆设为False；
- fls<列表文件>：此参数的效果和指定“-ls”参数类似，但会把结果保存为指定的列表文件；
- follow：排除符号连接；
- fprint<列表文件>：此参数的效果和指定“-print”参数类似，但会把结果保存成指定的列表文件；
- fprint0<列表文件>：此参数的效果和指定“-print0”参数类似，但会把结果保存成指定的列表文件；
- fprintf<列表文件><输出格式>：此参数的效果和指定“-printf”参数类似，但会把结果保存成指定的列表文件；
- fstype<文件系统类型>：只寻找该文件系统类型下的文件或目录；
- gid<群组识别码>：查找符合指定之群组识别码的文件或目录；
- group<群组名称>：查找符合指定之群组名称的文件或目录；
- help或—help：在线帮助；
- ilname<范本样式>：此参数的效果和指定“-lname”参数类似，但忽略字符大小写的差别；
- iname<范本样式>：此参数的效果和指定“-name”参数类似，但忽略字符大小写的差别；
- inum<inode编号>：查找符合指定的inode编号的文件或目录；
- ipath<范本样式>：此参数的效果和指定“-path”参数类似，但忽略字符大小写的差别；
- iregex<范本样式>：此参数的效果和指定“-regexe”参数类似，但忽略字符大小写的差别；
- links<连接数目>：查找符合指定的硬连接数目的文件或目录；
- lname<范本样式>：指定字符串作为寻找符号连接的范本样式；
- ls：假设find指令的回传值为True，就将文件或目录名称列出到标准输出；
- maxdepth<目录层级>：设置最大目录层级；
- mindepth<目录层级>：设置最小目录层级；
- mmin<分钟>：查找在指定时间曾被更改过的文件或目录，单位以分钟计算；
- mount：此参数的效果和指定“-xdev”相同；
- mtime<24小时数>：查找在指定时间曾被更改过的文件或目录，单位以24小时计算；
- name<范本样式>：指定字符串作为寻找文件或目录的范本样式；
- 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；
- nogroup：找出不属于本地主机群组识别码的文件或目录；
- noleaf：不去考虑目录至少需拥有两个硬连接存在；

- nouser：找出不属于本地主机用户识别码的文件或目录；
- ok<执行指令>：此参数的效果和指定“-exec”类似，但在执行指令之前会先询问用户，若回答“y”或“Y”，则放弃执行命令；
- path<范本样式>：指定字符串作为寻找目录的范本样式；
- perm<权限数值>：查找符合指定的权限数值的文件或目录；
- print：假设find指令的回传值为True，就将文件或目录名称列出到标准输出。格式为每列一个名称，每个名称前皆有“./”字符串；
- print0：假设find指令的回传值为True，就将文件或目录名称列出到标准输出。格式为全部的名称皆在同一行；
- printf<输出格式>：假设find指令的回传值为True，就将文件或目录名称列出到标准输出。格式可以自行指定；
- prune：不寻找字符串作为寻找文件或目录的范本样式；
- regex<范本样式>：指定字符串作为寻找文件或目录的范本样式；
- size<文件大小>：查找符合指定的文件大小的文件；
- true：将find指令的回传值皆设为True；
- typ<文件类型>：只寻找符合指定的文件类型的文件；
- uid<用户识别码>：查找符合指定的用户识别码的文件或目录；
- used<日数>：查找文件或目录被更改之后在指定时间曾被存取过的文件或目录，单位以日计算；
- user<拥有者名称>：查找符合指定的拥有者名称的文件或目录；
- version或—version：显示版本信息；
- xdev：将范围局限在先行的文件系统中；
- xtype<文件类型>：此参数的效果和指定“-type”参数类似，差别在于它针对符号连接检查。

参数

起始目录：查找文件的起始目录。

实例

根据文件或者正则表达式进行匹配

列出当前目录及子目录下所有文件和文件夹

```
find .
```

在 /home 目录下查找以.txt结尾的文件名

```
find /home -name "*.txt"
```

同上，但忽略大小写

```
find /home -iname "*.txt"
```

当前目录及子目录下查找所有以.txt和.pdf结尾的文件

```
find . \( -name "*.txt" -o -name "*.pdf" \)
```

或

```
find . -name "*.txt" -o -name "*.pdf"
```

匹配文件路径或者文件

```
find /usr/ -path "*local*"
```

基于正则表达式匹配文件路径

```
find . -regex ".*\\(\\.txt\\|\\.pdf\\)$"
```

同上，但忽略大小写

```
find . -iregex ".*\\(\\.txt\\|\\.pdf\\)$"
```

否定参数

找出/home下不是以.txt结尾的文件

```
find /home ! -name "*.txt"
```

根据文件类型进行搜索

```
find . -type 类型参数
```

类型参数列表：

- **f** 普通文件
- **l** 符号连接
- **d** 目录
- **c** 字符设备
- **b** 块设备
- **s** 套接字

- p Fifo

基于目录深度搜索

向下最大深度限制为3

```
find . -maxdepth 3 -type f
```

搜索出深度距离当前目录至少2个子目录的所有文件

```
find . -mindepth 2 -type f
```

根据文件时间戳进行搜索

```
find . -type f 时间戳
```

UNIX/Linux文件系统每个文件都有三种时间戳：

- 访问时间（-atime/天，-amin/分钟）：用户最近一次访问时间。
- 修改时间（-mtime/天，-mmin/分钟）：文件最后一次修改时间。
- 变化时间（-ctime/天，-cmin/分钟）：文件数据元（例如权限等）最后一次修改时间。

搜索最近七天内被访问过的所有文件

```
find . -type f -atime -7
```

搜索恰好在七天前被访问过的所有文件

```
find . -type f -atime 7
```

搜索超过七天内被访问过的所有文件

```
find . -type f -atime +7
```

搜索访问时间超过10分钟的所有文件

```
find . -type f -amin +10
```

找出比file.log修改时间更长的所有文件

```
find . -type f -newer file.log
```

根据文件大小进行匹配

```
find . -type f -size 文件大小单元
```

文件大小单元：

- **b** —— 块（512字节）
- **c** —— 字节
- **w** —— 字（2字节）
- **k** —— 千字节
- **M** —— 兆字节
- **G** —— 吉字节

搜索大于10KB的文件

```
find . -type f -size +10k
```

搜索小于10KB的文件

```
find . -type f -size -10k
```

搜索等于10KB的文件

```
find . -type f -size 10k
```

删除匹配文件

删除当前目录下所有.txt文件

```
find . -type f -name "*.txt" -delete
```

根据文件权限/所有权进行匹配

当前目录下搜索出权限为777的文件

```
find . -type f -perm 777
```

找出当前目录下权限不是644的php文件

```
find . -type f -name "*.php" ! -perm 644
```

找出当前目录用户tom拥有的所有文件

```
find . -type f -user tom
```

找出当前目录用户组sunk拥有的所有文件

```
find . -type f -group sunk
```

借助 `-exec` 选项与其他命令结合使用

找出当前目录下所有root的文件，并把所有权更改为用户tom

```
find . -type f -user root -exec chown tom {} \;
```

上例中，`{}` 用于与`-exec`选项结合使用来匹配所有文件，然后会被替换为相应的文件名。

找出自己家目录下所有的.txt文件并删除

```
find $HOME/. -name "*.txt" -ok rm {} \;
```

上例中，`-ok`和`-exec`行为一样，不过它会给出提示，是否执行相应的操作。

查找当前目录下所有.txt文件并把他们拼接起来写入到all.txt文件中

```
find . -type f -name "*.txt" -exec cat {} \;> all.txt
```

将30天前的.log文件移动到old目录中

```
find . -type f -mtime +30 -name "*.log" -exec cp {} old \;
```

找出当前目录下所有.txt文件并以“File:文件名”的形式打印出来

```
find . -type f -name "*.txt" -exec printf "File: %s\n" {} \;
```

因为单行命令中-exec参数中无法使用多个命令，以下方法可以实现在-exec之后接受多条命令

```
-exec ./text.sh {} \;
```

搜索但跳出指定的目录

查找当前目录或者子目录下所有.txt文件，但是跳过子目录sk

```
find . -path "./sk" -prune -o -name "*.txt" -print
```

find其他技巧收集

要列出所有长度为零的文件

```
find . -empty
```