

find

MengChunlei

April 11, 2021

1 基本用法

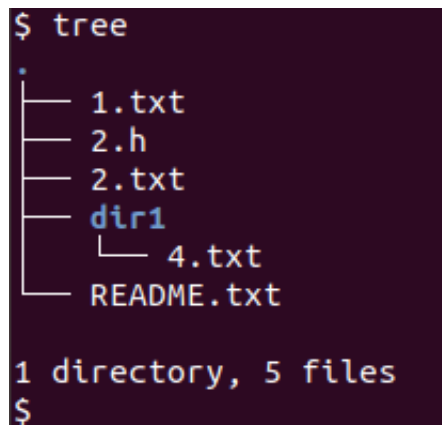
find 是用来在一个目录 (包括子目录) 下进行文件查找的命令, 基本语法规则为:

$$find[paths][expression][actions]$$

- *paths* 定义了查找的目录 (可以是多个目录, 用空格分隔)
- *expression* 定义了查找文件的条件
- *action* 定义了对于找到的文件要进行什么操作 (打印出来, 删除等等)

举个例子: *find ./ -name "*.txt" -print*

- *./* 表示在当前目录及其子目录下进行查找
- *-name "*.txt"* 表示查找名字后缀为 *txt* 的文件
- *-print* 表示对于找到的文件直接打印在屏幕上



```
$ tree
.
├── 1.txt
├── 2.h
├── 2.txt
├── dir1
│   └── 4.txt
└── README.txt

1 directory, 5 files
$
```

Figure 1: 当前目录结构

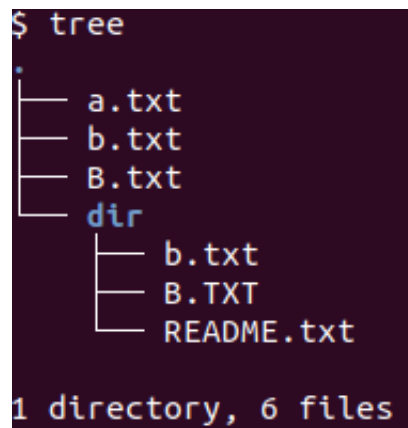
```
1 $ find ./ -name "*.txt" -print
2 ./2.txt
3 ./README.txt
4 ./dir1/4.txt
5 ./1.txt
6 $ find ./ -name "*.txt" /* 默认的 action 为 print */
7 ./2.txt
8 ./README.txt
9 ./dir1/4.txt
10 ./1.txt
11 $ find -name "*.txt" /* 默认的 paths 为当前目录 */
12 ./2.txt
13 ./README.txt
14 ./dir1/4.txt
15 ./1.txt
16 $ find dir1/ dir2/ /* 可以同时多个目录中查找 */
17 dir1/
```

```

18 dir1/4.txt
19 dir2/
20 dir2/B.txt
21 dir2/dir
22 dir2/dir/README.txt
23 dir2/dir/B.TXT
24 dir2/dir/b.txt
25 dir2/a.txt
26 dir2/b.txt
27 $ find /tmp/ -name "*.log"
28 find: '/tmp/systemd-private-6ded1284157744119460 Permission denied
29 /tmp/XN4851.log
30 /tmp/hay4851.log
31 find: '/tmp/systemd-private-6ded128415774411946052051140cff6- Permission denied
32 find: '/tmp/systemd-private-6ded128415774411946052051140cff6-sys Permission denied
33 $
34 $ find /tmp/ -name "*.log" 2>/dev/null /*过滤错误*/
35 /tmp/XN4851.log
36 /tmp/hay4851.log
37 $ find /tmp/ -name ??????.log 2>/dev/null /*只搜索 6 个字符的 log 文件*/
38 /tmp/XN4851.log

```

2 按照文件名字查找



```

$ tree
.
├── a.txt
├── b.txt
├── B.txt
└── dir
    ├── b.txt
    ├── B.TXT
    └── README.txt

1 directory, 6 files

```

Figure 2: 当前目录结构

```

1 $ find . -name b.txt
2 ./dir/b.txt
3 ./b.txt
4 $ find . -iname b.txt /*不区分大小写*/
5 ./B.txt
6 ./dir/B.TXT
7 ./dir/b.txt
8 ./b.txt

```

3 按照文件大小查找

```

1 $ ll
2 total 32
3 drwxrwxr-x 2 nio nio 4096 4月 11 11:08 ./
4 drwxrwxr-x 5 nio nio 4096 4月 11 11:07 ../
5 -rw-rw-r-- 1 nio nio 119 4月 11 11:08 file1
6 -rw-rw-r-- 1 nio nio 35 4月 11 11:08 file2
7 -rw-rw-r-- 1 nio nio 54 4月 11 11:08 file3

```

```

8 -rw-rw-r-- 1 nio nio    31 4月  11 11:08 file4
9 -rw-rw-r-- 1 nio nio    23 4月  11 11:08 file5
10 -rw-rw-r-- 1 nio nio    10 4月  11 11:08 file6
11 $ find . -size -54c /*小于54字节的文件*/
12 ./file6
13 ./file5
14 ./file4
15 ./file2
16 $ find . -size 54c /*等于54字节的文件*/
17 ./file3
18 $ find . -size +54c /*大于54字节的文件*/
19 .
20 ./file1

```

下面是文件单位:

- *b*: 512 字节的 block
- *c*: 字节
- *k*: kb
- *M*: mb
- *G*: GB

4 按照时间查找

Linux 中用到的时间有三种, 访问时间 (*atime*)、数据修改时间 (*mtime*)、状态修改时间 (*ctime*). 平时常用的应该是 *mtime*. 比如:

- `find . -mtime -5`: 5 天以内
- `find . -mtime 5`: 第五天
- `find . -mtime +5`: 6 天前

下面是时间说明图:

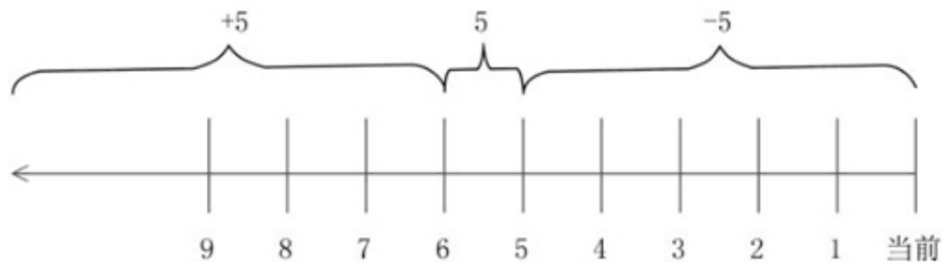


Figure 3: 时间说明图

另外，还有一组类似的以分钟作为单位的命令: *amin*, *mmin*, *cmin*

还可以使用 *-newer*、*-anewer* 和 *-cnewer* 选项查找已修改或访问过的文件与特定的文件比较。

```

1 $ echo aaa > 1.txt
2 $ echo bbb > 2.txt
3 $ echo ccc > 3.txt
4 $ find . -newer 2.txt /*修改时间晚于2.txt的文件*/
5 .
6 ./3.txt

```

5 按照文件类型查找

```
1 $ ll
2 total 12
3 drwxrwxr-x 3 nio nio 4096 4月 11 11:00 ./
4 drwxrwxr-x 5 nio nio 4096 4月 11 11:07 ../
5 -rw-rw-r-- 1 nio nio 0 4月 11 10:59 a.txt
6 -rw-rw-r-- 1 nio nio 0 4月 11 10:59 b.txt
7 -rw-rw-r-- 1 nio nio 0 4月 11 11:00 B.txt
8 drwxrwxr-x 2 nio nio 4096 4月 11 11:00 dir/
9 $ find . -type f /*查找文件*/
10 ./B.txt
11 ./dir/README.txt
12 ./dir/B.TXT
13 ./dir/b.txt
14 ./a.txt
15 ./b.txt
16 $ find . -type d /*查找文件夹*/
17 .
18 ./dir
```

6 逻辑运算

逻辑命令有三个，用来对条件做复合运算：

- -a: 条件 and
- -o: 条件或
- -not: 条件非

```
1 $ find . -name b.txt -a -mmin -50 /*名字为b.txt且最近50分钟修改过的文件*/
2 ./dir/b.txt
3 ./b.txt
4 $ find . '(' -not -name b.txt ')' -o -type d /*名字不是b.txt或者文件夹*/
5 .
6 ./B.txt
7 ./dir
8 ./dir/README.txt
9 ./dir/B.TXT
10 ./a.txt
```

7 其他 action

action 部分除了 *print* 以外还有两个命令：

- -exec: 用来执行后面的命令
- -ok: 用来执行后面的命令，但是每次会询问是否执行

```
1 $ ls
2 a.txt b.txt B.txt dir
3 $ find . -name b.txt
4 ./dir/b.txt
5 ./b.txt
6 $
7 $ find . -name b.txt -exec ls -l {} \; /*对于每个找到的文件执行ls -l*/
8 -rw-rw-r-- 1 nio nio 0 4月 11 10:59 ./dir/b.txt /*其中{}指代找到的文件*/
9 -rw-rw-r-- 1 nio nio 0 4月 11 10:59 ./b.txt
10 $ find . -name b.txt -exec rm {} \; /*删掉找到的文件*/
11 $ ls
```

```

12 a.txt B.txt dir
13 $ find . -path "*" /dir2 / *.txt"
14 ./dir5/dir2/a.txt
15 ./dir5/dir/dir2/4.txt
16 ./dir2/B.txt
17 ./dir2/dir/README.txt
18 ./dir2/dir/b.txt
19 ./dir2/a.txt
20 ./dir2/b.txt
21 $ find . -path "*" /dir2 / *.txt" -exec ls {} \; /*注意每找到一个打印一行*/
22 ./dir5/dir2/a.txt
23 ./dir5/dir/dir2/4.txt
24 ./dir2/B.txt
25 ./dir2/dir/README.txt
26 ./dir2/dir/b.txt
27 ./dir2/a.txt
28 ./dir2/b.txt
29 $ find . -path "*" /dir2 / *.txt" -exec ls {} + /*注意加号，所有的打印一行*/
30 ./dir2/a.txt ./dir2/b.txt ./dir2/B.txt ./dir2/dir/b.txt ./dir2/dir/README.txt
    ./dir5/dir2/a.txt ./dir5/dir/dir2/4.txt
31 $ find . -path "*" /dir2 / *.txt" -exec tar -czvf dir_txt.gz {} + /*打包所有找到的文件*/
32 ./dir5/dir2/a.txt
33 ./dir5/dir/dir2/4.txt
34 ./dir2/B.txt
35 ./dir2/dir/README.txt
36 ./dir2/dir/b.txt
37 ./dir2/a.txt
38 ./dir2/b.txt

```

8 控制查找的深度

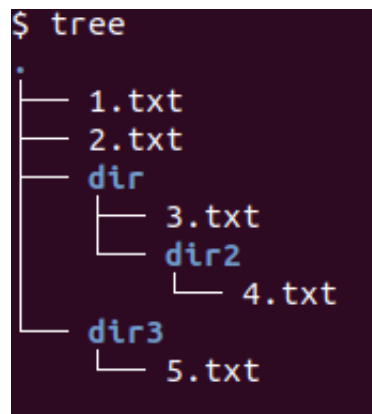


Figure 4: 时间说明图

```

1 $ find .
2 .
3 ./2.txt
4 ./dir3
5 ./dir3/5.txt
6 ./1.txt
7 ./dir
8 ./dir/3.txt
9 ./dir/dir2
10 ./dir/dir2/4.txt
11 $ find . -maxdepth 1 /*只搜索深度为 1 的文件*/
12 .
13 ./2.txt

```

```

14 ./dir3
15 ./1.txt
16 ./dir
17 $ find . -mindepth 2 /*从深度为 2 开始搜索*/
18 ./dir3/5.txt
19 ./dir/3.txt
20 ./dir/dir2
21 ./dir/dir2/4.txt

```

9 带路径搜索

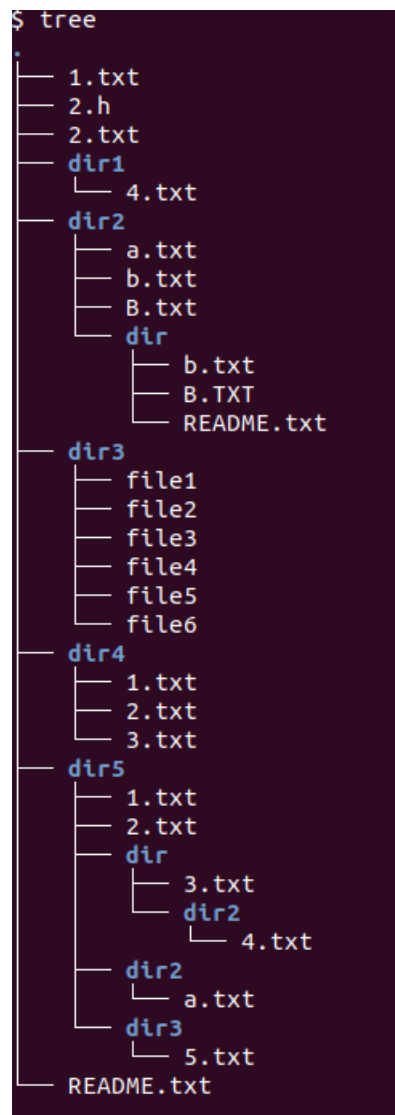


Figure 5: 时间说明图

```

1 $ find . -path "**/dir2/*.txt" /*目标文件所在的文件夹为dir2*/
2 ./dir5/dir2/a.txt
3 ./dir5/dir/dir2/4.txt
4 ./dir2/B.txt
5 ./dir2/dir/README.txt
6 ./dir2/dir/b.txt
7 ./dir2/a.txt
8 ./dir2/b.txt

```
