

第1章 find文件目录查找--必须掌握

1.find命令的应用场景

- 1 随着Linux系统运行的时间越来越长，安装的服务越来越多，时间久了，我们难免会忘记一些文件或目录的位置。
- 2 或者我们需要在不知道明确文件名的情况下，按照某些特定的需求去查找符合条件的文件。
- 3 比如：找出7天以前的文件，找出大于1G的文件，找出以*.conf结尾的文件等等。
- 4 而find命令就是专门为查找文件而服务的命令，学好find会让你的Linux之路如虎添翼。

2.find命令基本语法

- 1 find [查找路径] [选项参数] [限定条件] [执行动作]

3.find命令关键参数

选项参数

- 1 -maxdepth N #搜索的目录最深层级 N表示层级

限定条件

- 1 -name #根据文件名进行查找
- 2 -mtime N #按照文件修改时间进行查找
- 3 -size N[bkMG] #按照文件大小进行查找
- 4 -type #按照文件类型进行查找
- 5 -perm #按照文件权限进行查找
- 6 -user #按照文件所属用户进行查找

执行动作

- 1 -delete #将find查到的文件进行删除操作
- 2 -exec #将find查到的文件传递给后面的shell命令
- 3 -ok #在执行每个命令之前，都会让用户来选择是否确认执行

逻辑运算符

- 1 ! #取反
- 2 -a #并且 and
- 3 -o #或 or

4.find查找练习题

4.1 按名称查找

```
1 #按关键词查找文件
2 find /etc -name "passwd"
3 find /etc -name "hostname"
4 find /etc -name "*.conf"
5
6 #通过 ! 排除不想查找的文件
7 find /etc -name "ifcfg-*" ! -name "ifcfg-lo"
```

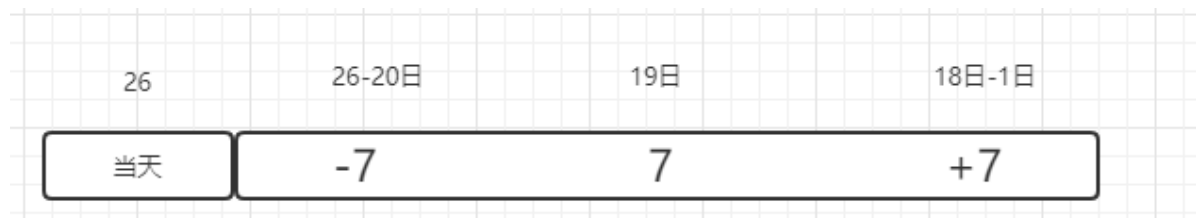
4.2 按大小查找

```
1 #查找大于10M的文件
2 find /etc -size +10M
3
4 #查找小于10M的文件
5 find /etc -size -10M
6
7 #查找等于10M的文件
8 find /etc -size 10M
9
10 #查找范围大于80M小于100M
11 find / -size +80M -size -100M
12
13 #按照文件类型和文件名和文件大小查找
14 find / -type f -size +9M -size -11M -name "*.tar.gz"
```

4.3 按类型查找

```
1 #查找类型为普通文件
2 find /dev -type f
3
4 #查找类型目录
5 find /dev -type d
6
7 #查找类型为软链接
8 find /dev -type l
```

4.4 按时间查找



```

1 #创建测试文件
2 mkdir find && cd find
3 for i in {01..26};do date -s 202103$i && touch file-$i;done
4 date -s "2021-03-26"
5
6 #查找7天以前的文件
7 find . -name "file-*" -mtime +7
8
9 #查找7天以内的文件
10 find . -name "file-*" -mtime -7
11
12 #查找第7天的文件
13 find . -name "file-*" -mtime 7

```

4.5 限制查找目录的层级

```

1 #创建测试文件
2 mkdir data1/1/2 -p
3 touch data1/a.txt
4 touch data1/1/b.txt
5 touch data1/1/2/c.txt
6 tree data1/
7 -----
8 data1/
9 |— 1
10 |   |— 2
11 |   |   |— c.txt
12 |   |— b.txt
13 |— a.txt
14
15 #查找不同目录层次的文件
16 [root@linux find]# find data1/ -maxdepth 1 -type f
17 data1/a.txt
18 [root@linux find]# find data1/ -maxdepth 2 -type f
19 data1/1/b.txt
20 data1/a.txt
21 [root@linux find]# find data1/ -maxdepth 3 -type f
22 data1/1/2/c.txt
23 data1/1/b.txt
24 data1/a.txt

```

5.find查找后执行动作

5.1 -exec执行动作

```

1 #使用-exec按条件查找文件并显示文件详细信息
2 find /etc -name "ifcfg*" -exec ls -l {} \;
3 find /etc -name "ifcfg*" -ls;
4
5 #使用-exec按条件查找文件并删除
6 find . -type f -name "1.txt"
7 find . -type f -name "1.txt" -exec rm -rf {} \;
8 find . -type f -name "1.txt"
9
10 #使用-exec按条件查找文件并安全删除
11 find . -type f -name "2.txt" -ok rm -rf {} \;

```

5.2 xargs执行动作

```

1 #使用xargs查找文件并显示文件详细信息
2 find /etc -name "ifcfg*" | xargs ls -l;
3
4 #使用xargs执行mv命令
5 find . -type f -name "*.txt" | xargs -i mv {} /opt/
6 find . -type f -name "*.txt" | xargs mv -t /opt/
7
8 #使用xargs将查找到的文件打包压缩
9 find . -type f -name "3.txt" | xargs tar zcf 3.tar.gz
10 tar zcf 3.tar.gz $(find . -type f -name "3.txt")
11
12 #删除目录下所有文件，但保留一个指定文件
13 touch file{1..10}.txt
14 find . -type f ! -name "file10.txt" | xargs rm -f
15 find . -type f ! -name "file10.txt" -exec rm -f {} \;
16
17 #找出7天以前的日志并删除
18 find . -type f -mtime +7 | xargs rm -rf
19 rm -rf $(find . -type f -mtime +7)

```

5.3 exec和xargs区别

- 1 **exec**是将查找的结果文件逐个传递给后面的命令来执行，如果文件多时则执行的效率低。
- 2 **xargs**是将查找的结果一次性传输给后面的命令执行，执行效率更高。

exec执行演示：

```

1 [root@linux find]# touch {1..5}.txt
2 [root@linux find]# find . -type f
3 ./1.txt
4 ./2.txt
5 ./3.txt
6 ./4.txt
7 ./5.txt
8
9 #可以发现exec每个结果都被加上了输出内容
10 [root@linux find]# find . -type f -exec echo "oldboyedu" {} \;
11 oldboyedu ./1.txt
12 oldboyedu ./2.txt
13 oldboyedu ./3.txt

```

```
14 oldboyedu ./4.txt
15 oldboyedu ./5.txt
```

xargs执行演示:

```
1 #可以发现xargs只会输出一老oldboyedu
2 [root@linux find]# find . -type f |xargs echo oldboyedu
3 oldboyedu ./1.txt ./2.txt ./3.txt ./4.txt ./5.txt ./6.txt ./7.txt ./8.txt
  ./9.txt ./10.txt
```

6.find逻辑运算符

```
1 #查找目录层级为1L的目录
2 find . -maxdepth 1 -type d
3
4 #查找目录层级为1L的目录，但是排除 .
5 find . -maxdepth 1 -type d ! -name "."
6
7 #查找目录层级为1L的目录，但是排除'.'或oldboy
8 find . -maxdepth 1 -type d ! -name "." -o -name "oldboy"
9
10 #查找目录层级为1L的目录，名字不为'.'并且名字为ext的目录
11 find . -maxdepth 1 -type d ! -name "." -a -name "ext"
```

7.find练习题汇总

```
1 创建测试文件命令:
2 mkdir /opt/dir{1..4} -p
3
4 touch -d "2020-11-26 01:00" /opt/dir1/111.txt
5 touch -d "2020-11-27 01:00" /opt/dir1/111.jpg
6 touch -d "2020-11-28 01:00" /opt/dir2/222.txt
7 touch -d "2020-11-29 01:00" /opt/dir2/222.jpg
8 touch -d "2020-11-30 01:00" /opt/dir3/333.html
9 touch -d "2020-12-01 01:00" /opt/dir4/444.conf
10 touch -d "2020-12-02 01:00" /opt/dir4/555.conf
11
12 dd if=/dev/zero of=/opt/dir1/111.data bs=1K count=100
13 dd if=/dev/zero of=/opt/dir2/222.data bs=1K count=300
14 dd if=/dev/zero of=/opt/dir3/333.data bs=1M count=10
15 dd if=/dev/zero of=/opt/dir4/444.data bs=1M count=100
16
17 echo "DB_NAME=oldboy" >> /opt/dir4/444.conf
18 echo "DB_PASSWD=123456" >> /opt/dir4/444.conf
19 echo "www.baidu.com" >> /opt/dir2/222.html
20 echo "www.oldboyedu.com" >> /opt/dir3/333.html
21
22 useradd www
23 chown www:www /opt/dir2/222.html
24 chown www:www /opt/dir3/333.html
25
26
27 练习题:
28 1. 找出/opt目录下所有以txt结尾的文件
29 2. 找出/opt目录下所有以txt结尾的文件但是排除掉文件名包含222的文件
```

```
30 3.找出/opt/目录下所有的conf文件
31 4.找出/opt目录下大于500K的文件
32 5.找出/opt目录下大于10M的文件
33 6.找出/opt目录下大于200K但是小于50M的文件
34 7.找出/opt目录下前3天的文件
35 8.找出/opt目录下最近2天的文件
36 9.找出/opt/dir2目录下以jpg结尾的文件
37 10.找出所有文件属于www用户的文件
38 11.找出/opt/目录下文件内容包含baidu的文件
39 12.找出/opt目录下文件内容包含oldboy的文件
40 13.找出/opt目录下属于www用户并且文件内容包含oldboy的文件
41 14.用一条命令找出/opt目录下所有的data文件并显示详细信息
42 15.用一条命令找出/opt目录下大于1M的文件并删除
```

第3章 xargs 将标准输入转换成命令行参数

1.xargs命令介绍

```
1 xargs命令是向其他命令传递命令行参数的一个过滤器，它能够将管道或者标准输入传递的数据转换成
  xargs命令后所跟命令的命令行参数。
```

2.xargs应用场景

```
1 结合管道和find对文件进行处理
2 控制每行输出的数量
```

3.xargs关键参数

```
1 -n      #指定每行最大参数量是n，可以将标准输入的文本划分为多行，每行显示n个
2 -d      #自定义分隔符
3 -i      #以{}替代前面的结果
4 -I      #以其他符号替代前面的结果
```

4.练习题

```
1 #创建测试文本
2 cat > test.txt << EOF
3 123456
4 789
5 10 11
6 EOF
7
8 #多行输入变单行
9 [root@linux find]# xargs < test.txt
10 123456 789 10 11
11 [root@linux find]# cat test.txt |xargs
12 123456 789 10 11
13
14 #指定每行输出个数
15 [root@linux find]# xargs -n 3 < test.txt
16 123456 789 10
17 11
```

```
18 [root@linux find]# cat test.txt |xargs -n 3
19 123456 789 10
20 11
21
22 #自定义分隔符
23 [root@linux find]# echo splitxsplitxsplitxsplitx
24 splitxsplitxsplitxsplitx
25
26 [root@linux find]# echo splitxsplitxsplitxsplitx|xargs -d x
27 split split split split
28
29 [root@linux find]# echo splitxsplitxsplitxsplitx|xargs -d x -n 2
30 split split
31 split split
```