

## 20 查找文件，无所遁形

更新时间：2019-07-10 14:04:21



“

与有肝胆人共事，从无字句处读书。

——周恩来

”

### 内容简介

1. 前言
2. locate 命令，快速查找
3. find 命令，深入查找
4. 总结

### 1. 前言

上一课RTFM 阅读那该死的手册，我们学习了很重要的技能：如何阅读使用手册。

这一课也不难，不过内容不少。

之前的课程中，我们已经见识过，Linux 下文件的组织形式是很特别的，跟 Windows 不一样。

我们也用 `ls /` 这个命令来列出根目录下的所有目录，有 `/bin/`、`/etc/`、`/var/`、`/home`等等，而这些目录下又有子目录和文件，错综复杂：

```
ls /
```

这些目录中有一部分是历史遗留的，从 Unix 时代就有了。问题是：我们如何在这“茫茫文海”中查找我们需要的文件呢？

不要担心，这一课我们就来学习 Linux 中用于文件查找的工具。

有的工具查找起来快，但是不全面；有的比较慢，但是全面。

一起来学习吧。

## 2. locate 命令，快速查找

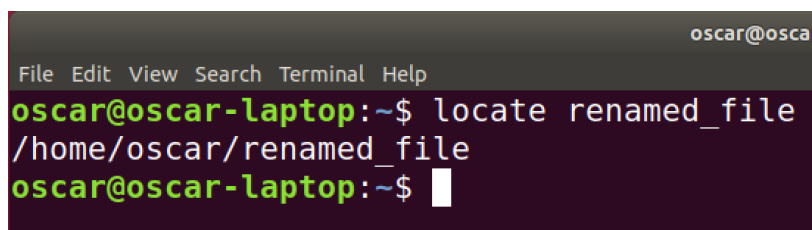
第一种查找文件的方法可以说是很简单的，用到的命令是 locate。

locate 是英语“定位”的意思。这个命令用于定位要查找的文件，而且此命令很快。

locate 命令的用法也很直观，后接需要查找的文件名（当然也可以用正则表达式）。

例如我们来查找一个叫做 renamed\_file 的文件：

```
locate renamed_file
```

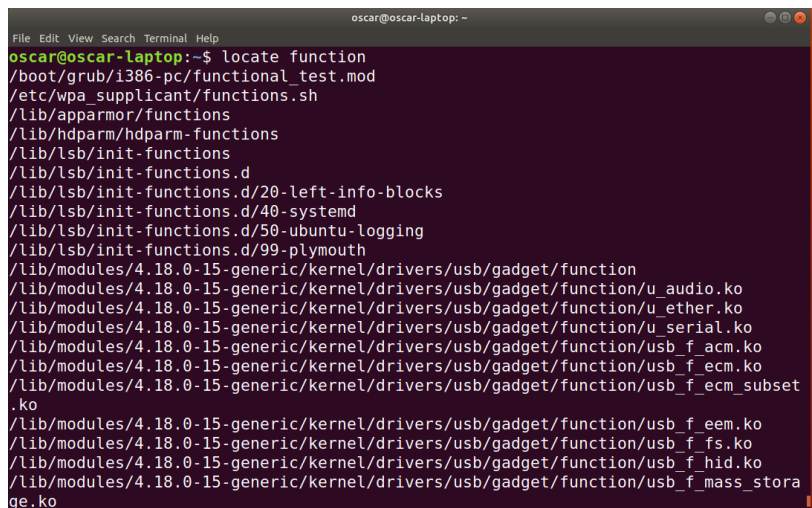


```
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ locate renamed_file
/home/oscar/renamed_file
oscar@oscar-laptop:~$
```

可以看到 locate 命令帮我们找到了 renamed\_file 文件，位于 /home/oscar/ 目录下。

我们再来用 locate 查找 function（function 是“函数、功能”的意思）：

```
locate function
```



```
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ locate function
/boot/grub/i386-pc/functional_test.mod
/etc/wpa_supplicant/functions.sh
/lib/apparmor/functions
/lib/hdparm/hdparm-functions
/lib/lsb/init-functions
/lib/lsb/init-functions.d
/lib/lsb/init-functions.d/20-left-info-blocks
/lib/lsb/init-functions.d/40-systemd
/lib/lsb/init-functions.d/50-ubuntu-logging
/lib/lsb/init-functions.d/99-plymouth
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function/u_audio.ko
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function/u_ether.ko
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function/u_serial.ko
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function/usb_f_acm.ko
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function/usb_f_ecm.ko
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function/usb_f_ecm_subset.ko
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function/usb_f_eem.ko
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function/usb_f_fs.ko
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function/usb_f_hid.ko
/lib/modules/4.18.0-15-generic/kernel/drivers/usb/gadget/function/usb_f_mass_storage.ko
```

我们运行 locate function 后，终端列出了所有包含 function 的文件和目录。

所以 locate 命令是搜索包含关键字的所有文件和目录。

### 文件的数据库

在使用 locate 命令查找文件时，大家可能会遇到这样的问题：我刚创建的文件，为什么用 locate 命令查找不到呢？

比如，我用 touch 命令新建一个文件，取名叫 newly\_created\_file（意思是“刚创建的文件”）：

```
touch newly_created_file
```

然后，我用 locate 命令去查找这个刚创建的文件：

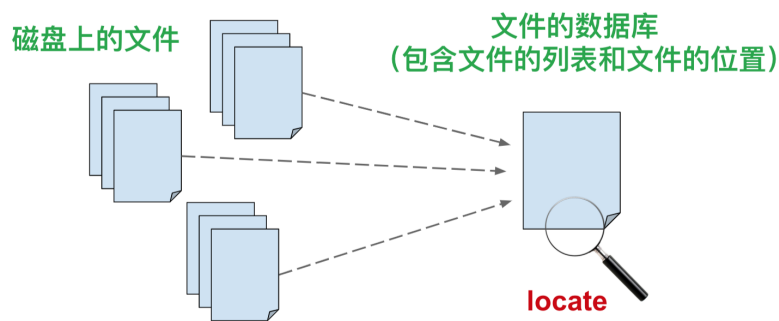
```
locate newly_created_file
```

```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
oscar@oscar-laptop:~$ touch newly_created_file  
oscar@oscar-laptop:~$ ls  
Desktop  file.txt          one      Public      Templates  
Documents Music             one_copy renamed_file test  
Downloads newly_created_file Pictures share    Videos  
oscar@oscar-laptop:~$ locate newly_created_file  
oscar@oscar-laptop:~$
```

可以看到，locate 命令并没有列出这个文件。

这正好是 locate 命令的缺陷，我正要说：locate 命令不会对你实际的整个硬盘进行查找，而是在文件的数据库里查找记录。

locate 命令的原理如下图所示：



对于刚创建不久的文件，由于它们还没被收录进文件数据库，因此 locate 命令就找不到其索引，自然就不会返回任何结果。

Linux 系统一般每天会更新一次文件数据库。因此，只要你隔 24 小时再用 locate 查找，应该就能找到你刚创建的文件了。

但是有人会说：“臣妾不想等这么久啊。皇上，难道臣妾要在这深宫之中等候多时么？”当然不是，甄嬛不会一直赢的，皇后总有出头之日。

我们可以用 updatedb 命令强制系统立即更新文件数据库，但是 updatedb 命令只能由 root 用户执行。

update 是英语“更新”的意思，db 是英语 database 的缩写，表示“数据库”。所以 updatedb 命令用于“更新数据库”。

因此我们可以运行：

```
sudo updatedb
```

一旦执行成功，你再用 locate 查找刚才创建的文件，就可以找到了：

```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
oscar@oscar-laptop:~$ touch newly_created_file
oscar@oscar-laptop:~$ ls
Desktop  file.txt      one      Public      Templates
Documents Music         one_copy renamed_file test
Downloads newly_created_file Pictures share  Videos
oscar@oscar-laptop:~$ locate newly_created_file
oscar@oscar-laptop:~$ sudo updatedb
[sudo] password for oscar:
oscar@oscar-laptop:~$ locate newly_created_file
/home/oscar/newly_created_file
oscar@oscar-laptop:~$
```

总结一下：locate 命令方便快捷，易于使用，但缺点也很明显：

1. locate 命令会列出所有在文件数据库中找到内容，有时候结果太多了，太繁杂；。
2. locate 命令不能找到一天之内刚创建的文件，除非你先用 root 身份运行 updatedb 命令来更新文件数据库。

当 locate 命令不够用时，我们需要一个更强大的命令，它就是 find 命令。

### 3. find 命令，深入查找

find 是英语“找到”的意思，find 命令也用于查找文件。

find 命令是查找文件的利器，而且它允许我们对每个找到的文件做后续操作。find 命令非常强大，可以做很多事情，因此也比较复杂。

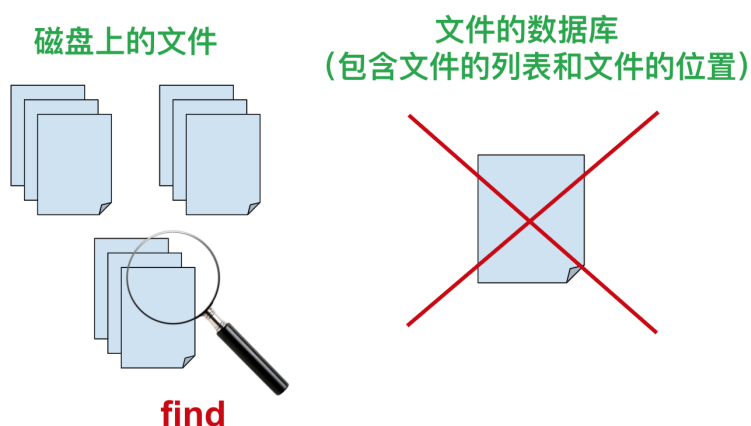
find 命令可以说是 Linux 中最常用的命令之一了，所以，一起来探秘吧。

#### find 命令查找当前实际的文件

与 locate 命令不同，find 命令不会在文件数据库中查找文件的记录，而是遍历你的实际硬盘。

所以，如果你的硬盘容量很大的话，那 find 命令会查找比较久。

find 命令的原理如下图所示：



find 命令的这种“耿直”的查找方式保证了我们不会遗漏一天之内创建的文件。但这绝对不是 find 的唯一强大之处，远远不是。

#### find 命令的功能

find 命令的用法是这样的：

```
find "何处""何物""做什么"
```

这几个参数中，只有“何物”是必须指定的，也就是要查找什么。

- 何处：指定在哪个目录中查找。此目录的所有子目录也会被查找，与 `locate` 命令的查找所有文件数据库的所有记录不同，`find` 命令可以限定查找目录，比如我们可以只让 `find` 查找 `/home` 目录。默认地，假如我们没有给出“何处”这个参数，那么 `find` 命令会在当前目录及其子目录中查找；
- 何物：也就是要查找什么。我们可以根据文件的名称来查找，也可以根据其大小来查找，也可以根据其最近访问时间查找等等，这个参数是必须的；
- 做什么：用 `find` 命令找到文件后，可以对每个文件做一定的操作，称为“后续处理”。默认地，假如不指定这个参数，那么 `find` 命令只会显示找到的文件，不会做其它事情。

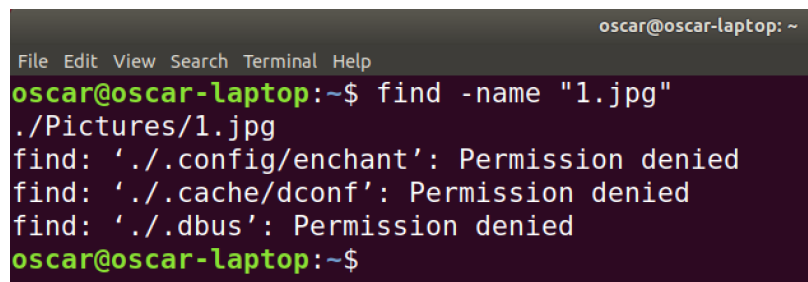
## find 命令的基础用法

### 根据文件名查找

我们首先来学习最基本的查找：用文件名来查找。

首先，我们定位到用户的家目录中（对于我来说是 `/home/oscar`），然后我要在当前目录及其子目录中查找 `1.jpg` 这个文件：

```
find -name "1.jpg"
```

A terminal window titled "oscar@oscar-laptop: ~" with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is "oscar@oscar-laptop:~\$". The command "find -name '1.jpg'" is entered. The output shows the file found at "Pictures/1.jpg", followed by three "Permission denied" messages for ".config/enchant", ".cache/dconf", and ".dbus". The prompt returns to "oscar@oscar-laptop:~\$".

```
oscar@oscar-laptop:~$ find -name "1.jpg"
./Pictures/1.jpg
find: './.config/enchant': Permission denied
find: './.cache/dconf': Permission denied
find: './.dbus': Permission denied
oscar@oscar-laptop:~$
```

这里，我们用 `-name` 参数指定了文件名字，是 `1.jpg`，用双引号括起来。不加双引号或者用单引号也是可以的。

`name` 是英语“名字”的意思。最终，`find` 命令找到了 `1.jpg` 这个文件，位于 `/home/oscar/Pictures` 目录下。

如果 `find` 命令没有结果显示，那么表示此文件不存在。

因为我们没有指定“何处”这个参数，所以 `find` 命令就会在当前目录（此处是 `/home/oscar`）及其子目录中查找。

上面的图中，有几行以“Permission denied”结尾的输出：

```
find: './.config/enchant': Permission denied
find: './.cache/dconf': Permission denied
find: './.dbus': Permission denied
```

`permission` 是“许可”的意思，`denied` 是“否决”的意思，就是说“权限被否决”，没有权限做这事。

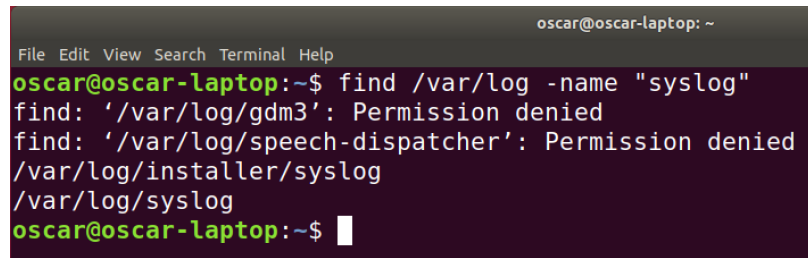
这是因为我们没有权限在 `.config/enchant`、`.cache/dconf` 和 `.dbus` 这几个目录里用 `find` 命令来查找，你可以用 `sudo ls -l` 来看一下它们的访问权限。暂时我们不深究。

那么，假如现在我位于家目录中，我却想在其它目录中进行查找，怎么办呢？

那就须要指定"何物"这个参数了。

例如，我要在 /var/log 目录下查找名为 syslog 的文件，我应该这么输入：

```
find /var/log -name "syslog"
```



```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
oscar@oscar-laptop:~$ find /var/log -name "syslog"  
find: '/var/log/gdm3': Permission denied  
find: '/var/log/speech-dispatcher': Permission denied  
/var/log/installer/syslog  
/var/log/syslog  
oscar@oscar-laptop:~$
```

可以看到，虽然 find 命令找到了我们想要的结果：

```
/var/log/installer/syslog  
/var/log/syslog
```

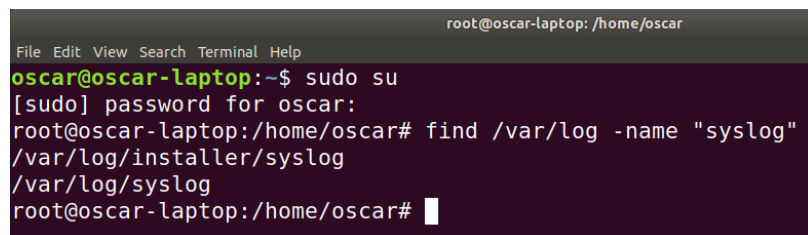
但又有几行以“Permission denied”结尾的输出：

```
find: '/var/log/gdm3': Permission denied  
find: '/var/log/speech-dispatcher': Permission denied
```

这也是因为我们没有权限在 /var/log/gdm3 和 /var/log/speech-dispatcher 目录中用 find 命令来查找。

“几次三番被否决，是可忍孰不可忍”。于是，我决定切换成 root 身份，再来执行查找：

```
sudo su  
  
find /var/log -name "syslog"
```



```
root@oscar-laptop: /home/oscar  
File Edit View Search Terminal Help  
oscar@oscar-laptop:~$ sudo su  
[sudo] password for oscar:  
root@oscar-laptop:/home/oscar# find /var/log -name "syslog"  
/var/log/installer/syslog  
/var/log/syslog  
root@oscar-laptop:/home/oscar#
```

可以看到，切换到 root 后，就可以查找了，并且不再显示“Permission denied”了。有些时候我们用 find 命令查找，需要在前面加上 sudo 命令，以便访问所有目录。

在上面的查找中，我们的参数如下：

- 何处：/var/log（及它的子目录）；
- 何物：syslog；
- 没有指定“做什么”参数，因此 find 命令默认显示了查找到的文件。

我们注意到了：与 locate 命令不同的是，find 命令只会查找完全符合“何物”的字符串表示的文件。locate 则会查找所有包含关键字的文件。



比如，如果要用 `find` 来找 `thing` 这个文件，那么只会找到名字就是 `thing` 的文件；而 `locate` 命令会查找到比如 `thing`、`thing1`、`onething`、`twothings` 这样的文件。

因此，我们用 `find` 查找 `syslog` 文件时，就不会查找到 `syslog2` 这样的文件。

不过我们可以用我们之前学过的通配符：`*`（星号）来实现匹配多个名称。

例如，我要查找所有以 `syslog` 这个关键字开头的文件，可以这样来实现：

```
find /var/log -name "syslog"
```

```
root@oscar-laptop:/home/oscar# find /var/log -name "syslog*"
/var/log/syslog.3.gz
/var/log/syslog.1
/var/log/syslog.5.gz
/var/log/syslog.4.gz
/var/log/syslog.6.gz
/var/log/installer/syslog
/var/log/syslog
/var/log/syslog.2.gz
root@oscar-laptop:/home/oscar#
```

可以看到，在 `syslog` 之后加了 `*` 之后，就会匹配凡是以 `syslog` 开头的文件了，因为 `*` 号是匹配一个或多个任意字符。

聪明如你肯定想到了，如果要查找以 `syslog` 结尾的文件，可以用：`*syslog`。

那你肯定也知道，如何查找包含 `syslog` 这个关键字的文件，可以用 `*syslog*`。

那如果我要在整个硬盘里查找怎么做呢？很简单：

```
find / -name "syslog"
```

在 `Linux` 的根目录查找很花时间。而且，如果你不是以 `root` 身份查找的话，就会有很多目录提示你“`Permission denied`”，所以最好在命令开始处加上 `sudo`。

## 根据文件大小查找

如果你不知道你要找的文件名称，那怎么办呢？

不要担心，还有其它方式来查找文件。

我们先来看如何根据文件大小查找。

例如，我们可以查找 `/var` 中大小超过 10 MB 的文件（当然你要以 `root` 身份）：

```
find /var -size +10M
```

```
root@oscar-laptop: /home/oscar
File Edit View Search Terminal Help
root@oscar-laptop:/home/oscar# find /var -size +10M
/var/cache/apt/srcpkgcache.bin
/var/cache/apt/archives/linux-headers-4.18.0-18_4.18.0-18.19-18.04.1_all.deb
/var/cache/apt/archives/linux-firmware_1.173.5_all.deb
/var/cache/apt/archives/linux-modules-extra-4.18.0-18-generic_4.18.0-18.19-18.04.1_amd64.deb
/var/cache/apt/archives/firefox_66.0.3+build1-0ubuntu0.18.04.1_amd64.deb
/var/cache/apt/archives/libwebkit2gtk-4.0-37_2.24.1-0ubuntu0.18.04.1_amd64.deb
/var/cache/apt/archives/libreoffice-common_1%3a6.0.7-0ubuntu0.18.04.5_all.deb
/var/cache/apt/archives/thunderbird_1%3a60.6.1+build2-0ubuntu0.18.04.1_amd64.deb
/var/cache/apt/archives/firefox_66.0.4+build3-0ubuntu0.18.04.1_amd64.deb
/var/cache/apt/archives/snapd_2.38+18.04_amd64.deb
/var/cache/apt/archives/linux-modules-4.18.0-18-generic_4.18.0-18.19-18.04.1_amd64.deb
/var/cache/apt/archives/libreoffice-core_1%3a6.0.7-0ubuntu0.18.04.5_amd64.deb
/var/cache/apt/pkgcache.bin
/var/lib/apt/lists/fr.archive.ubuntu.com_ubuntu_dists_bionic_universe_binary-amd64_Packages
/var/lib/apt/lists/fr.archive.ubuntu.com_ubuntu_dists_bionic_universe_binary-1386_Packages
/var/lib/apt/lists/fr.archive.ubuntu.com_ubuntu_dists_bionic_universe_i18n_Translation-en
/var/lib/snapd/seed/snaps/gtk-common-themes_818.snap
/var/lib/snapd/seed/snaps/core_6350.snap
```

这次，我们使用了 `-size` 参数，来指定查找文件的大小。`size` 是英语“尺寸，大小”的意思。

后面紧跟的 `+10M` 表示大于 10 MB（或 10 Mo），也就是 10 兆字节。

M 是兆，也就是 10 的 6 次方；平时我们所说的 Ko（或 KB），Mo（或 MB），Go（或 GB）其实分别是千字节，兆字节，千兆字节的意思。

一个字节（Byte 或 Octet）是 8 位二进制数。我们说硬盘 512 GB，就是硬盘容量有 512 千兆字节（虽然实际容量是不到的，暂时不深究）。

如果我们要查找小于指定大小的文件，可以用减号，例如：

```
find /var -size -50k
```

表示查找小于 50 Ko 的文件：

```
find /var -size +20G
```

表示查找大于 20 Go 的文件。

如果没有加减号，则查找大小等于指定数值的文件。

## 根据文件的最近访问时间查找

如果你记得你近 7 天里在家目录中访问过 JPG 格式的图片，但是你忘记它们的名字了，如何查找呢？

可以使用 `-atime` 参数。`atime` 是 `access` 和 `time` 的缩写，`access` 是英语“访问，进入”的意思，`time` 是英语“时间”的意思。

可以这样查找：

```
find -name "*.jpg" -atime -7
```



```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
oscar@oscar-laptop:~$ find -name "*.jpg" -atime -7
./Pictures/1.jpg
find: './.config/enchant': Permission denied
find: './.cache/dconf': Permission denied
find: './.dbus': Permission denied
oscar@oscar-laptop:~$
```

-atime 参数后面紧跟的 -7 表示 7 天之内，减号表示小于。

## 仅查找目录或文件

我们可以指定查找的文件类型，我们知道 Linux 中文件的类型大致分为两种：目录和文件（虽说 Linux 中“一切都是文件”，目录也是文件，但还是可以区分的）。

因此，我们可以用 -type 参数来指定查找的文件类型。type 是英语“类型”的意思。

- -type d：只查找目录类型。d 是 directory 的首字母，表示“目录”；
- -type f：只查找文件类型。f 是 file 的首字母，表示“文件”。

如果不用 -type 参数指定类型，那么 find 命令默认是同时查找目录和文件的。比如说，有 syslog 这个文件，和 syslog 这个目录，那么 find 会把它们都查找出来。

例如，我的家目录中存在名为 new\_file 的目录和文件，在家目录里运行如下命令：

```
sudo find . -name "new_file"
```

```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
oscar@oscar-laptop:~$ sudo find . -name "new_file"
./new_file
./one/new_file
./one_copy/new_file
oscar@oscar-laptop:~$
```

可以看到列出来三个结果，分别是家目录下的 new\_file 目录、以及 one 和 one\_copy 目录下的两个名叫 new\_file 的文件。

此时，我在上述命令后面加上类型参数：

```
sudo find . -name "new_file" -type d
```

```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
oscar@oscar-laptop:~$ sudo find . -name "new_file"
./new_file
./one/new_file
./one_copy/new_file
oscar@oscar-laptop:~$ sudo find . -name "new_file" -type d
./new_file
oscar@oscar-laptop:~$ ls -l
total 48
drwxr-xr-x 2 oscar oscar 4096 May 1 10:48 Desktop
drwxr-xr-x 2 oscar oscar 4096 May 1 10:48 Documents
drwxr-xr-x 2 oscar oscar 4096 May 1 10:48 Downloads
-rw-r--r-- 1 thomas friends 0 May 5 21:40 file.txt
drwxr-xr-x 2 oscar oscar 4096 May 1 10:48 Music
drwxr-xr-x 2 oscar oscar 4096 May 8 22:16 new_file
-rw-r--r-- 1 oscar oscar 0 May 8 21:06 newly_created_file
drwxr-xr-x 4 oscar oscar 4096 May 8 22:13 one
drwxr-xr-x 3 oscar oscar 4096 May 3 21:44 one_copy
drwxr-xr-x 2 oscar oscar 4096 May 8 21:35 Pictures
drwxr-xr-x 2 oscar oscar 4096 May 1 10:48 Public
-rw----- 1 oscar oscar 0 May 3 21:14 renamed_file
drwxr-xr-x 1 root root 640 May 8 21:34 share
drwxr-xr-x 2 oscar oscar 4096 May 1 10:48 Templates
drwxr-xr-x 2 oscar oscar 4096 May 8 22:10 test
drwxr-xr-x 2 oscar oscar 4096 May 8 22:13 Videos
```

可以看到，find 这时只找到了 new\_file 这个目录，而没有列出另外两个名为 new\_file 的文件。

用 ls -l 来查看一下，可以看到 new\_file 确实是一个目录。

## find 命令的高级用法：操作查找结果

到目前为止，我们只是使用了 find 的基本功能。我们并没有指定“做什么”这个参数，也就是还没对查找到的结果做任何操作。

当然了，默认地，find 命令会显示每个查找到的文件。

事实上：

```
find -name "*.jpg"
```

等价于：

```
find -name "*.jpg" -print
```

-print 参数用于打印结果。print 是英语“打印”的意思。

## 格式化打印查找结果

默认地，find 命令只列出查找到的文件，但是我们可以将其用格式化的方式打印出来。

我们可以使用 -printf 参数。是不是想起了 C 语言中的 printf 函数？是的，printf 是 print formatted 的缩写，表示“格式化打印”。

例如，我们可以这样打印查找到的内容：

```
sudo find . -name "*.jpg" -printf "%p - %u\n"
```

```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~$ sudo find . -name "*.jpg" -printf "%p - %u\n"
./Pictures/1.jpg - oscar
oscar@oscar-laptop:~$
```

我们在 `-printf` 参数后面写了 `"%p - %u\n"`，这个格式字符串的意思如下：

- `%p`：文件名；
- `:` 就是一个短横；
- `%u`：文件的所有者，这里是我的用户名，所以是 `oscar`；
- `\n`：用于换行。

可以看到，用法和 C 语言的 `printf` 函数很类似。

## 删除查找到的文件

假如我要删除查找到的文件，我可以用 `-delete` 参数。用法很简单：

```
find -name "*.jpg" -delete
```

将会删除当前目录及其子目录下所有以 `.jpg` 为后缀的文件，而且不会有确认提示，所以请慎用 `-delete` 参数。

## 调用命令

使用 `-exec` 参数，可以后接一个命令，对每个查找到的文件进行操作。

`exec` 是英语 `execute` 的缩写，表示“执行”。

假设我想要将当前目录下所有查找到的 `JPG` 文件的访问权限都改为 `600`，那么我们可以这样做：

```
find -name "*.jpg" -exec chmod 600 {} \;
```

就是说对于每个找到的 `.jpg` 结尾的文件，都进行 `-exec` 参数指定的操作：

- 这个操作不必用双引号括起来；
- `{}` 会用查找到的每个文件来替换；
- `;` 是必须的结尾。

虽然 `-exec` 参数一开始有点看不懂其用法，但是慢慢地你会发现，这个参数太强大了，你可以对查找到的文件做任何你想要的操作。

如果你对于没有确认提示不太放心，你可以将 `-exec` 参数换成 `-ok` 参数。

用法一样，只不过 `-ok` 参数会对每一个查找到的文件都做确认提示，输入 `y` 加回车表示对此文件进行此操作；输入 `n` 加回车表示对此文件不进行此操作。

## 小结

1. 为了在 Linux 硬盘中查找一个文件，我们可以用 `locate` 命令。这个命令很快，因为它不会实际遍历硬盘，而是在被称为“文件数据库”的记录里查找，所以它找不到刚添加的文件。为了更新文件数据库，我们需要用 `root` 用户身

份运行 updatedb 命令；

2. find 命令比 locate 命令更强大，用法更多样。它会遍历实际的硬盘来查找文件，所以更慢，但是更精确。find 命令可以接三个参数：“何处”，“何物”，“做什么”；
3. find 命令可以根据文件的名字来查找，用 -name 参数；根据大小来查找，用 -size 参数；根据最后访问时间来查找，用 -atime 参数等等；
4. 用 find 命令查找到了文件，我们可以选择不显示，而是删除文件，用 -delete 参数；或者对文件执行命令，用 -exec 参数。

今天的课就到这里，一起加油吧！