

11 目录相关的命令 ls, cd 和 du

更新时间：2019-07-04 15:15:07



立志是事业的大门，工作是登门入室的旅途。

——巴斯德

内容简介

1. 前言
2. ls 命令：列出文件和目录
3. cd 命令：切换目录
4. du 命令：显示目录包含的文件大小
5. 总结

1. 前言

上一课文件组织，pwd和which命令。组织不会亏待你中，我们认识了 Linux 的文件组织。

这一课我们接着学习三个和文件与目录有关的命令。

2. ls 命令：列出文件和目录

ls 命令绝对是最常用的 Linux 命令之一。

之前的课程我们也简单地介绍了 ls 命令的用法。现在我们来比较仔细地学习一下 ls 命令的功能和它的众多参数。

ls 是 list 的缩写，是英语“列出”的意思。所以 ls 命令用于列出文件和目录。

首先，我们在自己的家目录中执行 ls 命令，不带任何参数：

```
ls
```

```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
oscar@oscar-laptop:~$ ls  
Desktop    Downloads  Pictures   share      Videos  
Documents  Music      Public     Templates  
oscar@oscar-laptop:~$
```

Ubuntu 的终端默认是有颜色标注的，一般来说：

- 蓝色 --> 目录
- 绿色 --> 可执行文件
- 红色 --> 压缩文件
- 浅蓝色 --> 链接文件
- 灰色 --> 其他文件

如果在你的终端里没有标明颜色，那么你可以加上参数 `--color=auto` 来开启颜色标注。如下：

```
ls --color=auto
```

如果你不需要颜色标注，那么可以用 `--color=none` 来关闭颜色标注。如下：

```
ls --color=none
```

当然，为了避免每次都输入这么一串额外的参数，其实我们可以在一个配置文件里修改，使得永久显示颜色或不显示颜色。我们之后的课程再说。

`ls` 命令接受很多参数，如果要把这些参数全部列出来，也没什么意思，毕竟平时常用的参数也就那几个。所以，我们一起来看一下 `ls` 常用的一些参数的使用方法。

-a：显示所有文件和目录，包括隐藏的

在 Linux 中，我们可以隐藏一些文件。这并不是一种很强的保护机制，因为我们其实还是可以把这些文件显示出来。

平时我们单独使用 `ls`，不带参数时，就没有列出这些隐藏的文件，也是为了简洁起见。

我们的家目录是个很好的演示隐藏文件的目录，因为这个目录里的隐藏文件还不少呢。我们在 `ls` 命令后面添加 `-a` 参数看看：

```
ls -a
```

```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
oscar@oscar-laptop:~$ ls -a  
.  
..  
.bash_history  
.bash_logout  
.bashrc  
.cache  
.config  
.dbus  
Desktop  
Documents  
Downloads  
.gnupg  
.ICEauthority  
.local  
Music  
.pam_environment  
Pictures  
.profile  
Public  
share  
.ssh  
.sudo_as_admin_successful  
Templates  
.vboxclient-clipboard.pid  
.vboxclient-display.pid  
.vboxclient-draganddrop.pid  
.vboxclient-seamless.pid  
Videos  
oscar@oscar-laptop:~$
```

可以看到比单独用 ls 时多了那些以点开头的文件和目录，现在你知道为什么要隐藏这些以点开头的文件了吧。因为这些文件还真不少，会显得很混乱。

在 Linux 中，以点 (.) 开头的文件是隐藏文件，不过这里不包括开头的两个 . 和 ...，这两个的含义如下：

- . (一个点) 表示当前目录。在当前情况下，就是 /home/oscar；
- .. (两个点) 表示上一级目录。因为当前目录是 /home/oscar，所以上一级目录就是 /home。

我们有一个大写的参数 -A，它的作用和小写的 -a 几乎一样，唯一不同就是 -A 不列出 . 和 ... 这两个文件。

```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
oscar@oscar-laptop:~$ ls -a
.          .dbus      Music      .sudo_as_admin_successful
..         Desktop  .pam_environment  Templates
.bash_history Documents  Pictures    .vboxclient-clipboard.pid
.bash_logout Downloads  .profile    .vboxclient-display.pid
.bashrc    .gnupg     Public      .vboxclient-draganddrop.pid
.cache     .ICEauthority share       .vboxclient-seamless.pid
.config    .local     .ssh        Videos
oscar@oscar-laptop:~$ ls -A
.bash_history .gnupg      .ssh
.bash_logout .ICEauthority .sudo_as_admin_successful
.bashrc      .local      Templates
.cache       Music       .vboxclient-clipboard.pid
.config      .pam_environment .vboxclient-display.pid
.dbus        .Pictures   .vboxclient-draganddrop.pid
Desktop      .profile    .vboxclient-seamless.pid
Documents    Public      Videos
Downloads    share
oscar@oscar-laptop:~$
```

-l：详细列表

-l 这个参数算是 ls 命令最常用的了。加上 -l 参数使得 ls 命令列出一个显示文件和目录的详细信息列表：

```
ls -l
```

```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~
oscar@oscar-laptop:~$ ls -l
total 32
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
drwxr-xr-x 2 oscar oscar 4096 May  1 10:48 Music
drwxr-xr-x 2 oscar oscar 4096 May  1 10:48 Pictures
drwxr-xr-x 2 oscar oscar 4096 May  1 10:48 Public
drwxr-xr-x 1 root  root   640 Jul 12  2018 share
drwxr-xr-x 2 oscar oscar 4096 May  1 10:48 Templates
drwxr-xr-x 2 oscar oscar 4096 May  1 10:48 Videos
oscar@oscar-laptop:~$
```

我们可以看到，每一个文件或目录都有对应的一行信息。在这些信息的最前面，有一个 total xxx 的，是表示当前目录所有文件的总大小是 total 后面的那个数字所表示的千字节（Kilo Byte = 1024 Byte）数。

在我们的例子中，是 total 32，所以表示我的家目录中所有文件的总大小加起来是 32 个千字节。

后面学到 du 命令的时候，就知道其实这里的 total 后面的大小并不是统计了当前目录真正所有文件的大小（包括逐层的文件大小相加）。

这里的大小只是统计了当前目录列出的这一层的文件和目录的大小，并没有深入统计总的大小。

一行上的每一列是一个单独的信息，我们从左到右一一说明：

- 文件权限：第一列的 `drwxr-xr-x` 之类的。暂时不需要理解，我们之后会用单独的一课来学习 Linux 下的文件权限；
- 链接的数目：第二列的 `1`, `2` 之类的。暂时不需要理解，之后的课也会解释；
- 文件所有者的名称：第三列的 `oscar` 或 `root`，表示这个文件或目录的所有者是哪个用户。如果此文件是属于另一个用户，比如说 `john`，那么文件所有者就会显示 `john` 了；
- 文件所在的群组：第四列的 `oscar` 或 `root`，表示这个文件或目录的是属于哪个群组。关于群组，我们暂时不需要理解，之后在文件权限的那一课会讲解；
- 文件大小：第五列的 `4096` 或 `640` 等等。单位是 `octet`，也就是 `byte`，是英语“字节”的意思。比如 `share` 那个目录的大小是 `640` 个字节。这里列出的文件都是目录，因此这些大小并没有显示目录中所有文件的总大小；
- 最近一次修改的时间：第六列。比如 `share` 那个目录的最近一次修改时间是 `Jul 12 2018`，也就是 2018 年 7 月 12 日。因为 `share` 目录是宿主机（我的情况是我的 macOS）上的共享文件夹；
- 文件或目录的名称：第七列。

-h: 以 Ko, Mo, Go 的形式显示文件大小

当我们仅用 `ls -l` 时，列出的文件详细信息中，文件的大小是以字节为单位的。有时这不是太直观。

我们可以再加一个参数 `-h`，`h` 是 `human-readable` 的缩写，表示“适合人阅读的”。所以我们的命令变成了：

```
ls -lh
```

```
oscar@oscar-laptop:~$ ls -lh
total 32K
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Desktop
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Documents
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Downloads
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Music
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Pictures
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Public
drwxr-xr-x 1 root  root   640 Jul 12  2018 share
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Templates
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Videos
oscar@oscar-laptop:~$
```

这次的显示就美观多了。可以看到，最开头的 `total` 后面的 `32`，变成了 `32K`，也就是 `32 KB` 的意思，表示当前目录所有文件的总大小是 `32 KB`（`KB` 大家应该不陌生吧，就是 `Ko`，是 `Kilo Octet` 的缩写，表示“千字节”，等于 `1024 B`）。同样，`32 KB` 不是当前目录真正所有文件的总大小。

可以看到几乎每个文件的大小信息那一列，都变成了 `4.0K` 这样的形式，也就是之前的 `4096 B`。用了 `-h` 参数，我们的文件信息更便于阅读了。除了 `share` 目录的大小还是显示 `640`，因为不到 `1 KB`，所以不进行转换。

-t: 按文件最近一次修改时间排序

`t` 是 `time` 的缩写，表示“时间”。

```
ls -lt
```

```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
oscar@oscar-laptop:~$ ls -lt  
total 32  
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  
drwxr-xr-x 2 oscar oscar 4096 May  1 10:48 Music  
drwxr-xr-x 2 oscar oscar 4096 May  1 10:48 Pictures  
drwxr-xr-x 2 oscar oscar 4096 May  1 10:48 Public  
drwxr-xr-x 2 oscar oscar 4096 May  1 10:48 Templates  
drwxr-xr-x 2 oscar oscar 4096 May  1 10:48 Videos  
drwxr-xr-x 1 root  root   640 Jul 12 2018 share  
oscar@oscar-laptop:~$
```

可以看到，这是按照文件或目录最近一次修改时间的降序排列，也就是说排在第一位的是 Desktop（桌面）这个目录，修改时间是最新的，其它的依次降序。

当然，我们了解了这些 ls 的常用参数后，也可以把这几个参数合并在一起使用，效果是叠加的：

```
ls -lath
```

```
oscar@oscar-laptop:~$ ls -lath  
total 104K  
drwxr-xr-x 17 oscar oscar 4.0K May  2 18:33 .  
-rw----- 1 oscar oscar 2.7K May  2 18:33 .ICEauthority  
-rw-r----- 1 oscar oscar  5 May  2 18:33 .vboxclient-draganddrop.pid  
-rw-r----- 1 oscar oscar  5 May  2 18:33 .vboxclient-display.pid  
-rw-r----- 1 oscar oscar  5 May  2 18:33 .vboxclient-seamless.pid  
-rw-r----- 1 oscar oscar  5 May  2 18:33 .vboxclient-clipboard.pid  
-rw----- 1 oscar oscar 305 May  1 23:12 .bash_history  
drwx----- 13 oscar oscar 4.0K May  1 15:43 .config  
drwx----- 14 oscar oscar 4.0K May  1 15:42 .cache  
drwx----- 3 root  root  4.0K May  1 15:42 .dbus  
-rw-r--r-- 1 oscar oscar  0 May  1 15:01 .sudo_as_admin_successful  
-rw-r--r-- 1 oscar oscar 310 May  1 11:06 .pam_environment  
drwx----- 3 oscar oscar 4.0K May  1 10:57 .gnupg  
drwx----- 2 oscar oscar 4.0K May  1 10:57 .ssh  
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Desktop  
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Documents  
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Downloads  
drwxr-xr-x 2 oscar oscar 4.0K May  1 10:48 Music
```

3. cd 命令：切换目录

cd 命令也是最常用的 Linux 命令之一，甚至可以说是最常用的命令。

cd 是英语 change directory 的缩写，表示“切换目录”。顾名思义，cd 命令是用于切换目录的命令。

也许你发现了，目前来说，我们还没跳出我们用户的家目录这个“小圈子”。对于渴望自由，向往远方的我们来说，简直不能忍，不是吗？

好的，那就使用 cd 命令，开动起我们的这艘“破船”吧！我们得去其它海域逛逛。

与 ls 命令不同，cd 命令没有那么多参数，cd 命令后面接的参数只有一个，你要去的那个目录的路径。

比如我想去根目录 / 转转，那么我们就可以使用：

```
cd /
```

```
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ cd /
oscar@oscar-laptop:/$ pwd
/
oscar@oscar-laptop:/$
```

可以看到，使用 `pwd` 命令，显示当前我们已经位于根目录 `/` 中了，我们也可以从命令行提示符中看出来，当前冒号和美元符之间已经从之前的 `~` 变成了 `/`。

怎么样，对 Linux 的“狂拽酷炫”是不是已经有点感觉了？就是这么轻盈，穿梭在各个目录之间。比起 Windows 的鼠标点点点，是不是快捷了很多？

我们再去其他地方转转，输入：

```
cd usr
```

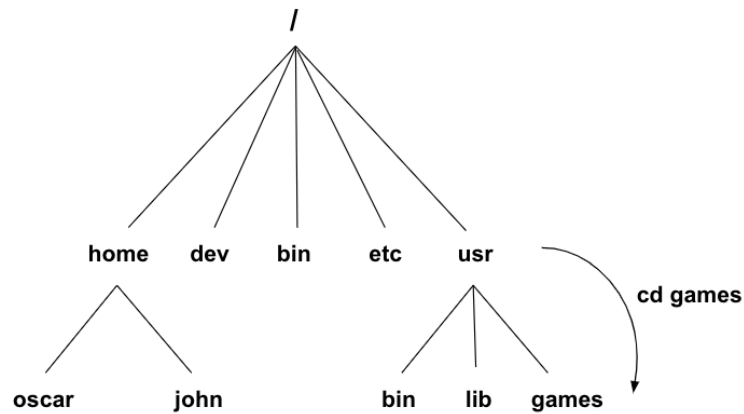
因为我们当前是在根目录 `/` 中，所以用 `cd usr` 我们就切换到了 `/usr` 这个目录：

```
File Edit View Search Terminal Help
oscar@oscar-laptop:/$ cd usr
oscar@oscar-laptop:/usr$ pwd
/usr
oscar@oscar-laptop:/usr$
```

然后我们想看看 `/usr` 这个目录中有哪些子目录，用 `ls` 命令，可以看到有好几个子目录，我们再用 `cd` 命令进入其中的 `games`（game 是英语“游戏”的意思）这个子目录：

```
oscar@oscar-laptop: /usr/games
File Edit View Search Terminal Help
oscar@oscar-laptop:/$ cd usr
oscar@oscar-laptop:/usr$ pwd
/usr
oscar@oscar-laptop:/usr$ ls
bin  games  include  lib  local  sbin  share  src
oscar@oscar-laptop:/usr$ cd games
oscar@oscar-laptop:/usr/games$ ls
espdiff
oscar@oscar-laptop:/usr/games$
```

刚才的一系列操作，我们可以用以下图示来表示：



好了，现在我们位于 `/usr/games` 这个目录下。那么如果我要回到上一级目录（也叫“父目录”），应该怎么做呢？

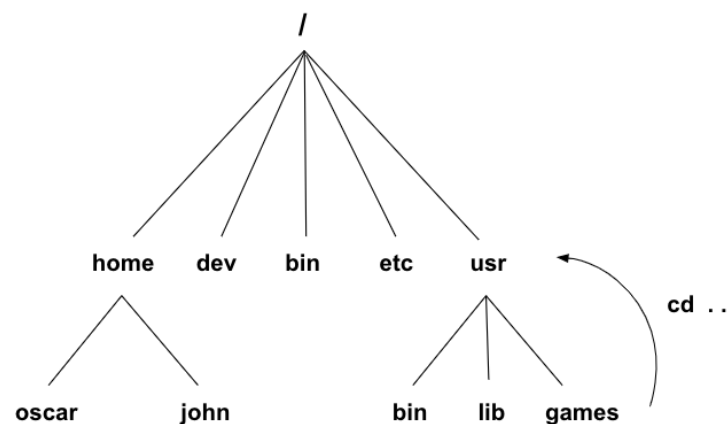
可以用：

```
cd ..
```

就是 `cd` 命令后面接两个点。Linux 中，一个点表示当前目录，而两个点表示上一级目录：

```
oscar@oscar-laptop:~/usr/games$ cd ..
oscar@oscar-laptop:~/usr$ pwd
/usr
oscar@oscar-laptop:~/usr$
```

可以用如下图示表示：



那么，聪明如你，应该不难想到，如何连续后退两个层级。是的，可以用：

```
cd ../../
```

```
oscar@oscar-laptop: /usr/games$ cd ../../
oscar@oscar-laptop: /$ pwd
/
oscar@oscar-laptop: /$
```

可以看到，我们用 `cd ../../` 连续后退了两个层级，回到了根目录 `/`。

实际上，我们在切换目录时，可以有两种路径表示方式，一种是相对路径，一种是绝对路径。

顾名思义，相对路径就是相对于当前目录的一个路径表示法。绝对路径就是绝对的，不因当前目录而改变的一个路径表示法。

相对路径

刚才，我们位于 `/usr` 目录时，我们只用了 `cd games` 就切换到了 `/usr/games`，这里 `cd games` 中的 `games` 就是相对路径。因为这个 `games` 目录是 `/usr` 目录的子目录，所以是相对当前的 `/usr` 目录而言的。

如果我们在根目录里运行命令 `cd games`，那么会出现错误，因为根目录 `/` 里并没有 `games` 这个子目录，它找不到这个相对路径，会显示 `No such file or directory`（表示“不存在此文件或目录”）：

```
oscar@oscar-laptop: /$ cd games
bash: cd: games: No such file or directory
oscar@oscar-laptop: /$
```

所以我们只能这样：

```
cd usr/games
```

才能从根目录用相对路径的方式切换到 `/usr/games` 目录中：

```
oscar@oscar-laptop: /$ cd usr/games/
oscar@oscar-laptop: /usr/games$ pwd
/usr/games
oscar@oscar-laptop: /usr/games$
```

绝对路径

与相对路径不一样，不管我们当前位于哪个目录，绝对路径的使用都是一样的。

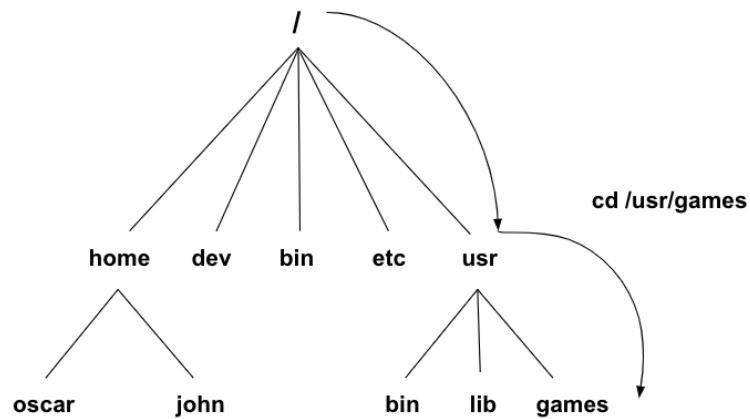
绝对路径很好辨认，因为它总是以 `/` 开头，`/` 表示根目录。假设，我们目前位于家目录中，那么如果我要用绝对路径来切换到 `/usr/games` 中，怎么做呢？很简单，就是使用：

```
cd /usr/games
```



```
oscar@osca
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ cd /usr/games/
oscar@oscar-laptop:/usr/games$ pwd
/usr/games
oscar@oscar-laptop:/usr/games$
```

图示如下：

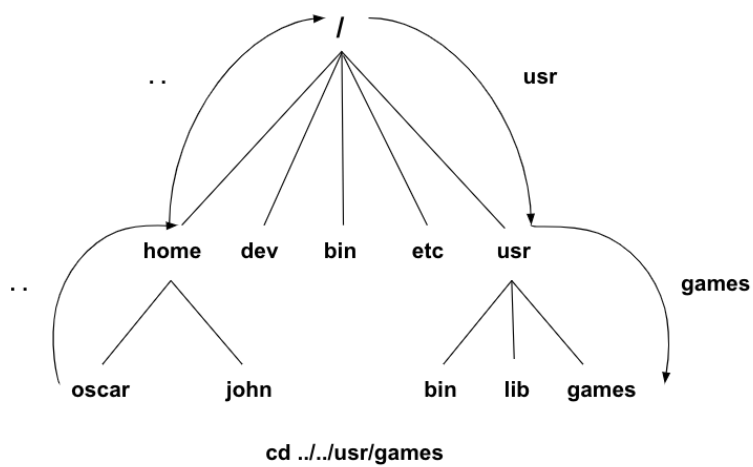


如果我们要用相对路径达到同样的目的，那么需要这样：

```
cd ../../usr/games
```

```
oscar@oscar-lapt
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ cd ../../usr/games
oscar@oscar-laptop:/usr/games$ pwd
/usr/games
oscar@oscar-laptop:/usr/games$
```

图示如下：



重回家目录

家目录毕竟可以说是用户使用最多的目录，所以我们在用 `cd` 命令跳转到其他目录之后，经常渴望快速切换到自己的家目录，怎么做呢？如何“乘风归去”呢？

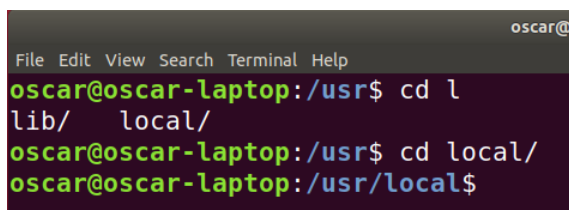
有三种方法：

- 比较“笨”的方法：`cd /home/oscar` 这里的 `oscar` 可以替换为你自己的用户名，这种方法有点繁琐；
- 较好的方法：`cd ~`；
- 最好的方法：`cd`。就是直接使用 `cd` 命令，不加任何参数，就可以切换到用户的家目录了，是不是很棒？

路径自动补全

细心如你可能已经想到了，因为上一课中我们已经提过 `Tab` 键可以进行命令的补全（按两下 `Tab` 键），也提了一下其实 `Tab` 键还可以用于路径的补全。

所以，这样用 `Tab` 键可以大大提高我们的效率，而且防止我们手动输入路径时将名字写错：



```
oscar@
File Edit View Search Terminal Help
oscar@oscar-laptop:/usr$ cd l
lib/  local/
oscar@oscar-laptop:/usr$ cd local/
oscar@oscar-laptop:/usr/local$
```

4. du 命令：显示目录包含的文件大小

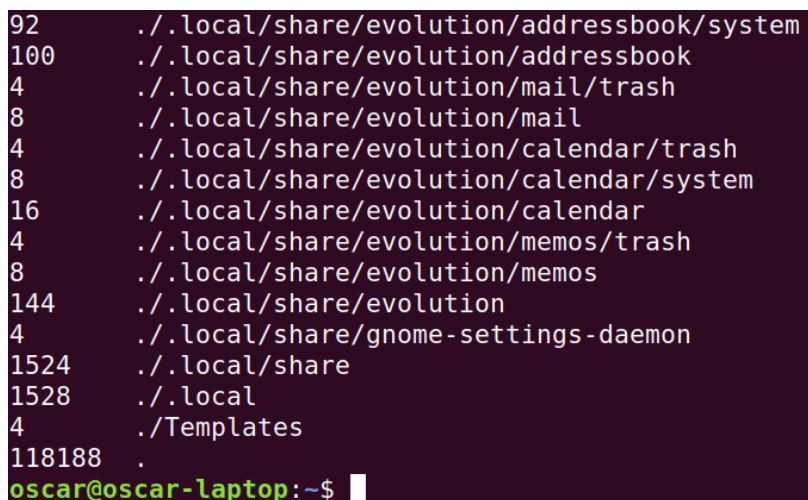
`du` 命令也是很有用的，因为它可以让我们知道文件和目录所占的空间大小。

虽然 `ls -l` 命令可以显示文件和目录的大小，但是 `du` 命令统计的才是真正的文件大小。`du` 命令会深入遍历每个目录的子目录，把所有文件的大小都做一个统计。

`du` 是英语 `disk usage` 的缩写，表示“磁盘使用/占用”。

来对我们的家目录做一个真正的大小的统计吧：

```
du
```



```
92      ../local/share/evolution/addressbook/system
100     ../local/share/evolution/addressbook
4       ../local/share/evolution/mail/trash
8       ../local/share/evolution/mail
4       ../local/share/evolution/calendar/trash
8       ../local/share/evolution/calendar/system
16      ../local/share/evolution/calendar
4       ../local/share/evolution/memos/trash
8       ../local/share/evolution/memos
144     ../local/share/evolution
4       ../local/share/gnome-settings-daemon
1524    ../local/share
1528    ../local
4       ../Templates
118188  .
oscar@oscar-laptop:~$
```

我在自己的家目录运行 `du` 命令后，列出的文件大小的列表太长，所以只能截图了最后的部分。

可以看到，我的家目录的总大小是 118188 个字节（Byte）。是不是觉得换算这么庞大的数字让人抓狂呢？

不要着急，我们的 du 命令也可以使用参数。

-h: 以 Ko, Mo, Go 的形式显示文件大小

-h 参数的用法类似之前在 ls 命令中的 -h 参数（h 是英语 human-readable 的首字母，表示“适合人阅读的”），所以我们可以运行 du -h 来看一下：

```
du -h
```

```
92K    ../local/share/evolution/addressbook/system
100K   ../local/share/evolution/addressbook
4.0K   ../local/share/evolution/mail/trash
8.0K   ../local/share/evolution/mail
4.0K   ../local/share/evolution/calendar/trash
8.0K   ../local/share/evolution/calendar/system
16K    ../local/share/evolution/calendar
4.0K   ../local/share/evolution/memos/trash
8.0K   ../local/share/evolution/memos
144K   ../local/share/evolution
4.0K   ../local/share/gnome-settings-daemon
1.5M   ../local/share
1.5M   ../local
4.0K   ../Templates
116M   .
oscar@oscar-laptop:~$
```

可以看到我的家目录 /home/oscar 的总大小是 116 MB。

-a: 显示文件和目录的大小

默认情况下，du 命令只显示目录的大小。如果加上 -a 参数（a 是英语 all 的首字母，表示“全部”），则会显示目录和文件的大小。我们使用 du -ah 来看一下：

```
du -ah
```

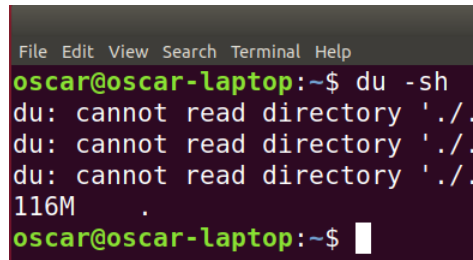
```
28K    ../local/share/gnome-software/ubuntu-reviews.db
32K    ../local/share/gnome-software
4.0K   ../local/share/evolution/tasks/trash
8.0K   ../local/share/evolution/tasks
4.0K   ../local/share/evolution/addressbook/trash
4.0K   ../local/share/evolution/addressbook/system/photos
84K    ../local/share/evolution/addressbook/system/contacts.db
92K    ../local/share/evolution/addressbook/system
100K   ../local/share/evolution/addressbook
4.0K   ../local/share/evolution/mail/trash
8.0K   ../local/share/evolution/mail
4.0K   ../local/share/evolution/calendar/trash
4.0K   ../local/share/evolution/calendar/system/calendar.ics
8.0K   ../local/share/evolution/calendar/system
16K    ../local/share/evolution/calendar
4.0K   ../local/share/evolution/memos/trash
8.0K   ../local/share/evolution/memos
144K   ../local/share/evolution
0      ../local/share/gnome-settings-daemon/input-sources-converted
4.0K   ../local/share/gnome-settings-daemon
1.5M   ../local/share
1.5M   ../local
4.0K   ../Templates
116M   .
oscar@oscar-laptop:~$
```

可以看到总大小还是 116 MB。因为虽然 du 命令默认情况下只显示各个目录的大小，但是统计的总大小已经包括了目录中的各个文件了，只不过默认没有把文件列出来罢了。

-s: 只显示总计大小

如果我们不想看到各个目录和文件的大小统计，而只想知道当前目录的总大小，可以使用 `-s` 参数（`s` 是英语 `summarize` 的首字母，表示“总结，概括”），我们用 `du -sh` 来看一下：

```
du -sh
```



```
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ du -sh
du: cannot read directory './.
du: cannot read directory './.
du: cannot read directory './.
116M  .
oscar@oscar-laptop:~$
```

5. 总结

1. `ls` 命令用于显示当前目录中的文件；
2. `cd` 命令用于切换目录；
3. `du` 命令用于统计目录大小。

今天的课就到这里，一起加油吧！