

34 文件传输，潇洒同步

更新时间：2019-07-31 10:13:30



“

青年是学习智慧的时期，中年是付诸实践的时期。

—— 卢梭

”

内容简介

1. 前言
2. wget：下载文件
3. scp：网间拷贝
4. ftp & sftp：传输文件
5. rsync：同步备份
6. 总结
7. 第四部分第五课预告

1. 前言

上两课 [带你玩转Linux和Shell编程](#) | [第四部分第二课：连接远程终端和对称加密，安全快捷](#) 和 [带你玩转Linux和Shell编程](#) | [第四部分第三课：非对称加密和使用SSH建立安全连接](#) 的内容有点“饱满”，大家可能“消化”得比较辛苦。

这一课的内容相对简单，所以我们慢慢“享用”。但是这一课要学的几个命令还是很有用的，特别是 rsync。

经过上一课的学习，我们已经知道如何远程连接到其它电脑了。从今以后，你就可以在自己的电脑上运行命令，来控制远方的另一台电脑了，很酷吧。

我们继续探究 Linux 的奇幻网络世界。这个网络世界比较特别，也许你已经有所体会：

前辈们做了大量努力，靠加密方法来保证网络间传输的数据的安全，以防我们私人的信息泄露，例如密码。

这一课我们来学习文件传输，主要分为以下几方面：

- 如何下载文件
- 如何连接到 FTP，读取、下载文件
- 如何安全地拷贝文件

2. wget：下载文件

我们就从一个简单的命令开始吧，就是：wget。

它可以使我们直接从终端控制台下载文件，只需要给出文件的 HTTP 或 FTP 地址。

命令格式：

```
wget [参数] [URL地址]
```

例如：

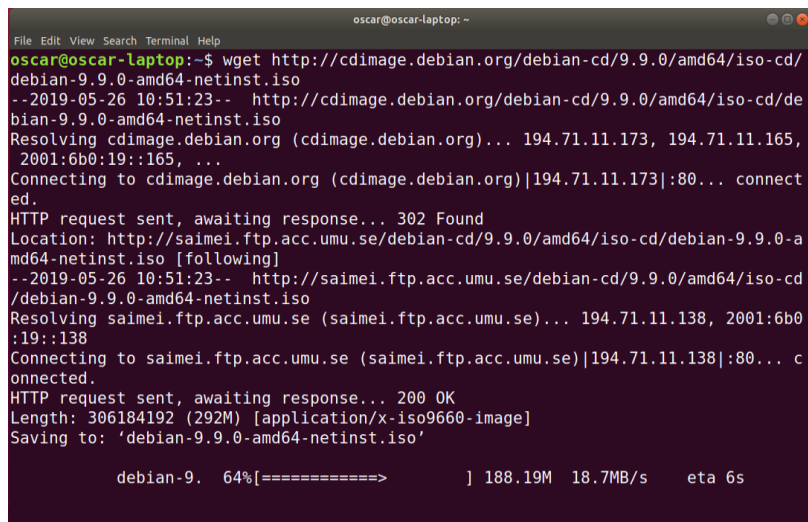
```
wget http://cdimage.debian.org/debian-cd/9.9.0/amd64/iso-cd/debian-9.9.0-amd64-netinst.iso
```

就会开始从 <http://cdimage.debian.org/debian-cd/9.9.0/amd64/iso-cd/>

上下载 debian-9.9.0-amd64-netinst.iso 这个文件。

如果要停止下载，只需要按 Ctrl +C。

可以看到下方会出现一个进度条，显示下载进度：

A terminal window titled 'oscar@oscar-laptop: ~' showing the execution of the 'wget' command. The output displays the file being downloaded, the URL, the file size (306184192 bytes), and the download progress. The progress bar shows 64% completion, with a download speed of 18.7MB/s and an estimated time to completion of 6 seconds.

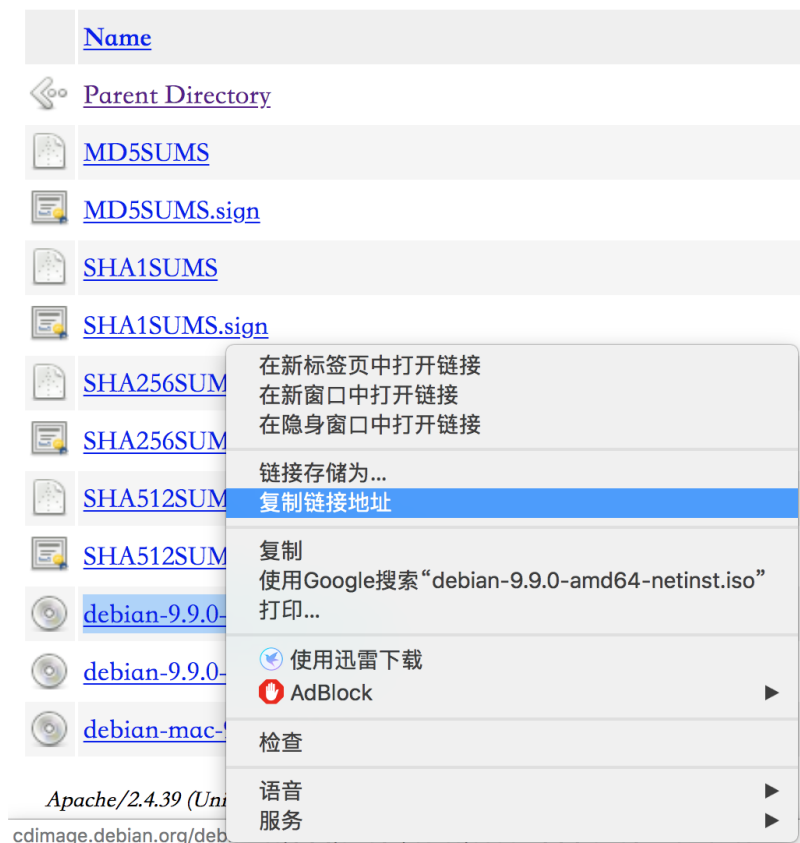
```
oscar@oscar-laptop:~$ wget http://cdimage.debian.org/debian-cd/9.9.0/amd64/iso-cd/debian-9.9.0-amd64-netinst.iso
--2019-05-26 10:51:23-- http://cdimage.debian.org/debian-cd/9.9.0/amd64/iso-cd/debian-9.9.0-amd64-netinst.iso
Resolving cdimage.debian.org (cdimage.debian.org)... 194.71.11.173, 194.71.11.165, 2001:6b0:19::165, ...
Connecting to cdimage.debian.org (cdimage.debian.org)|194.71.11.173|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://saimei.ftp.acc.umu.se/debian-cd/9.9.0/amd64/iso-cd/debian-9.9.0-amd64-netinst.iso [following]
--2019-05-26 10:51:23-- http://saimei.ftp.acc.umu.se/debian-cd/9.9.0/amd64/iso-cd/debian-9.9.0-amd64-netinst.iso
Resolving saimei.ftp.acc.umu.se (saimei.ftp.acc.umu.se)... 194.71.11.138, 2001:6b0:19::138
Connecting to saimei.ftp.acc.umu.se (saimei.ftp.acc.umu.se)|194.71.11.138|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 306184192 (292M) [application/x-iso9660-image]
Saving to: 'debian-9.9.0-amd64-netinst.iso'

debian-9. 64%[=====>] 188.19M 18.7MB/s eta 6s
```

- 64 % 表示已下载百分之六四。
- 18.7MB/s 是下载速度，表示 117 MB 每秒。
- eta 是预计剩余时间，此处是 6 秒。

那么，怎么事先获得供 wget 下载的地址呢？

你可以用浏览器（比如 firefox、chrome），找到要下载的文件，然后在文件上点击鼠标右键，左键点击“复制链接地址”，如下图：



然后黏贴到 `wget` 命令的地址参数中就可以了。

`wget` 非常稳定，它在带宽很窄的情况下和不稳定网络中有很强的适应性。
如果是由于网络的原因下载失败，`wget` 会不断的尝试，直到整个文件下载完毕。
如果是服务器打断下载过程，它会再次联到服务器上从停止的地方继续下载。
这对从那些限定了链接时间的服务器上下载大文件非常有用。

继续中断的下载

要继续一个中断的下载，只要在相同的下载命令中加入 `-c` 参数，例如：

```
wget -c http://cdimage.debian.org/debian-cd/9.9.0/amd64/iso-cd/debian-9.9.0-amd64-netinst.iso
```

`c` 是英语 `continue` 的缩写，表示“继续”。

`wget` 有非常多的参数选项，我们不能逐一列举。可以参看 `wget` 的使用手册：

```
man wget
```

`wget` 的一个优点是会显示下载的进度，稍后我们会学习的 `ftp` 命令则不会显示下载进度。

3. `scp`：网间拷贝

我们以前的课程中学习过 `cp` 命令，它用于在自己的电脑上拷贝文件。`cp` 是英语 `copy` 的缩写，表示“拷贝”。

scp 是Secure CoPy 的缩写，表示“安全拷贝”。这个命令可以使我们通过网络，把文件从一台电脑拷贝到另一台。当然，拷贝的信息是安全的，正如它的名字所示。

也有一个命令 rcp，是 Remote CoPy 的缩写，表示“远程拷贝”。它也可以做同样的事，但是信息没有得到安全保护，不推荐。

scp 使用起来有点类似上两课学过的 SSH（Secure SHell）。这并不是巧合，因为 scp 是基于 SSH 的原理来运作的。SSH 首先会在两台通过网络连接的电脑之间创建一条安全通信的管道（如上两课所示），scp 就利用这条管道安全地拷贝文件。

scp 的基本命令格式如下：

```
scp source_file destination_file
```

其中：

- source_file 表示源文件，就是被拷贝的文件；
- destination_file 表示目标文件，就是拷贝产生的文件。

这两个文件都可以用如下方式来表示：

```
user@ip:file_name
```

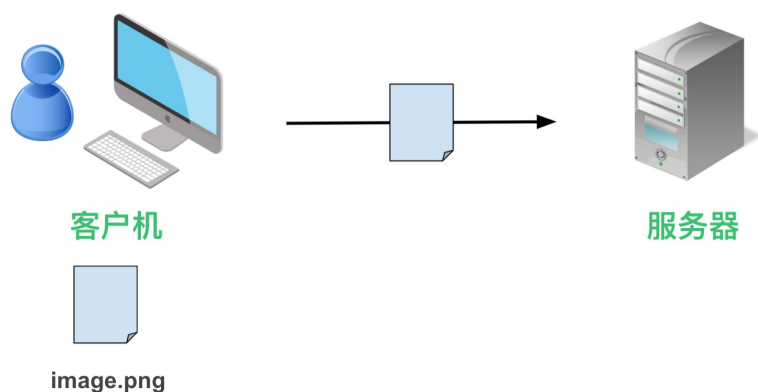
其中 user 是登录名，ip 是域名（例如 [google.com](https://www.google.com)）或 ip 地址（例如 216.58.206.238），file_name 是文件路径，不要忘了中间的 @ 号和冒号（:）。

从自己电脑拷贝文件到另一台电脑

这很简单，例如：

```
scp image.png oscar@89.231.45.67:/home/oscar/images/
```

表示把我的电脑中当前文件夹下的 image.png 文件拷贝到远程电脑（ip 地址是 89.231.45.67。当然了，这是假想的 ip 地址）的用户 oscar 的 /home/oscar/images 目录下，文件名不变（还是 image.png，你也可以改名字）。如下图所示：



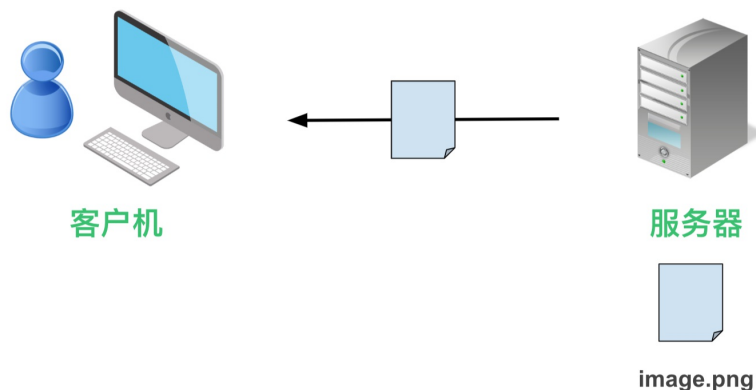
当然了，scp 会请求你输入远程电脑（ip 地址是 89.231.45.67）的用户 oscar 的密码。输入密码，回车，就开始拷贝了。

从另一台电脑拷贝文件到自己电脑

类似的用法，例如：

```
scp oscar@89.231.45.67:/home/oscar/images/image.png file_changed_name.png
```

表示从远程电脑（ip 地址是 89.231.45.67）的用户 oscar 的 /home/oscar/images 目录下把 image.png 拷贝到我的电脑中当前文件夹下，并改名为 file_changed_name.png（不改名也可以）。如下图所示：



修改端口

上述命令中，我们并没有指定用哪个端口，只指定了 ip 地址。默认的端口号是 22，和 SSH 一样。我们也可以修改端口号，用 -P 参数（p 是 port 的缩写，表示“端口”）。例如：

```
scp -P 7821 oscar@89.231.45.67:/home/oscar/images/image.png .
```

表示从远程电脑（ip 地址是89.231.45.67，端口 7821）的用户 oscar 的 /home/oscar/images 目录下把 image.png 拷贝到我的电脑中当前文件夹下，名字不变。此处用点号（.）表示当前目录。

注意：上一课中，SSH 修改端口号使用 -p 参数，p 是小写。而 scp修改端口号使用 -P 参数，P 是大写。

4. ftp & sftp：传输文件

FTP 是 File Transfer Protocol 的缩写，表示“文件传输协议”。顾名思义，就是用于传输文件的。

FTP 协议已经有些年纪了，1985 年诞生，比我还老呢。FTP 现在仍然是传输文件的最常用协议。

正所谓“廉颇老矣，尚能饭否”；人家是“FTP 当道，宝刀未老”。

使用 FTP 主要分为两种情况：

- 从公共的 FTP 服务器下载文件。一般来说，当你点击浏览器上的下载链接时，浏览器就以自动和透明的方式来完成这个操作。这种情况下，连接是匿名的；
- 从私有的 FTP 服务器上传或下载文件。当我们从服务器出租商处租用一台服务器作为个人网站之用时，出租商通常会给我们一个 FTP 的登录名和密码，我们可以连接，以上传及下载文件。这种情况下，连接是需要身份验

证的。

因为并不是每个读者都有自己私人的 FTP 服务器，所以下面演示的时候，我们会连接到公共的 FTP 服务器。当然，如果你想要连接到私人 FTP 服务器，方法是类似的。

我们这里使用纯命令行的形式来操作。当然了，也存在不少优秀的 FTP 软件，可以提供图形操作界面，例如著名的 [FileZilla](#)

连接到 FTP 服务器

我们试着连接到 Debian 的 镜像 FTP 服务器，可以在这里找到各个国家的地址：<https://www.debian.org/mirror/list>。

比如我可以试试 Debian 在法国的 FTP 服务器：<ftp.fr.debian.org>。当然了，你可以选择你想要使用的 FTP 服务器地址。

要连接到 FTP 服务器，方法很简单，运行以下命令：

```
ftp -p ftp.fr.debian.org
```

此处我在 ftp 命令之后用了 -p 这个参数，p 是 passive 的缩写，表示“被动的”。如果不加这个参数，在 <ftp.fr.debian.org> 这个服务器上不能使用 FTP 的命令。

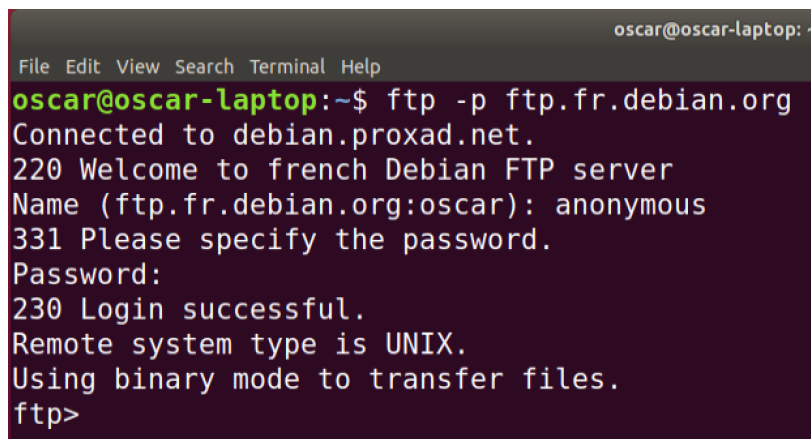
这是 FTP 的一种模式，暂时我们不深入讲解为什么要加 -p 参数，有兴趣可以参看这个关于 passive 模式的 [百度百科](#)。

Debian 的 FTP 服务器应该会有所回应，并向你请求输入用户名和密码。

对于公共的 FTP 服务器，用户名（Name）一般都填写 anonymous（表示“匿名的”）。密码（Password）你随便输入什么都会被接受。

登录成功后，会看到类似以下信息：

```
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```



```
oscar@oscar-laptop: ~
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ ftp -p ftp.fr.debian.org
Connected to debian.proxad.net.
220 Welcome to french Debian FTP server
Name (ftp.fr.debian.org:oscar): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

现在你有了命令提示符了，就是那个：

```
ftp>
```

你就可以输入 FTP 命令了。

在 FTP 服务器中操作

好消息：你在 FTP 服务器上可以使用的命令基本和我们到目前为止学习的 Linux 命令是一样的。

例如：

- ls: 列出当前目录的文件
- pwd: 显示当前目录的路径
- cd: 转换目录

你也可以试试其它命令。








```
oscar@oscar-laptop: ~
File Edit View Search Terminal Help
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (212,27,32,66,178,35)
150 Here comes the directory listing.
drwxr-xr-x  9 1000      1000          4096 May 26 14:50 debian
drwxr-xr-x  8 1000      1000          4096 Mar 01 2015 debian-amd64
drwxr-sr-x  5 1000      1000          102 Mar 13 2016 debian-backports
drwxr-xr-x  6 1000      1000          143 May 26 03:00 debian-non-US
drwxr-xr-x  7 1000      1000          142 May 26 12:12 debian-security
drwxr-sr-x  5 1000      1000          138 Nov 01 2011 debian-volatile
drwxr-xr-x  2 1000      1000           6 May 26 14:00 tmp
226 Directory send OK.
ftp> pwd
257 "/"
ftp> cd debian
250 Directory successfully changed.
ftp> ls
227 Entering Passive Mode (212,27,32,66,125,61)
150 Here comes the directory listing.
-rw-r--r--  1 1000      1000          1181 Apr 27 08:28 README
-rw-r--r--  1 1000      1000          1290 Jun 26 2010 README.CD-manufacture
-rw-r--r--  1 1000      1000          2898 Apr 27 08:28 README.html
-rw-r--r--  1 1000      1000           291 Mar 04 2017 README.mirrors.html
```

可以看到，我用 ls 命令可以列出目录下文件。用 `cd debian` 命令切换到了 debian 这个子目录里面，又用 ls 命令列出包含的文件。

当然了，你在浏览器里面用 HTTP 协议也是可以登录这个 Debian 服务器的：<http://ftp.fr.debian.org>。

← → ↻ ⓘ 不安全 | http://ftp.fr.debian.org

Index of /

	Name	Last modified	Size	Description
	debian-amd64/	2015-03-01 15:28	–	
	debian-backports/	2016-03-13 10:23	–	
	debian-non-US/	2019-05-26 05:00	–	
	debian-security/	2019-05-26 14:12	–	
	debian-volatile/	2011-11-01 12:05	–	
	debian/	2019-05-26 16:50	–	
	tmp/	2019-05-26 16:00	–	

同样可以访问 `debian` 这个子目录里的文件：

← → ↻ ⓘ 不安全 | http://ftp.fr.debian.org/debian/

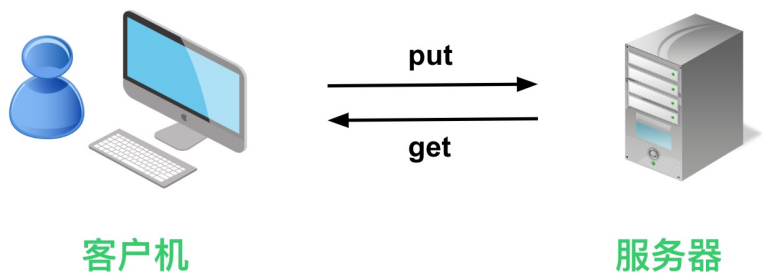
Index of /debian			
Name	Last modified	Size	Description
📁 Parent Directory		-	
👉 README	2019-04-27 10:28	1.2K	
📄 README.CD-manufacture	2010-06-26 11:52	1.3K	
📄 README.html	2019-04-27 10:28	2.8K	
📄 README.mirrors.html	2017-03-04 21:08	291	
📄 README.mirrors.txt	2017-03-04 21:08	86	
📁 dists/	2019-04-27 10:29	-	
📁 doc/	2019-05-26 15:52	-	
📄 extrafiles	2019-05-26 16:37	174K	
📁 indices/	2019-05-26 16:32	-	
📄 ls-IR.gz	2019-05-26 16:32	10M	
📁 pool/	2000-12-19 21:10	-	
📁 project/	2008-11-18 00:05	-	
📁 tools/	2012-10-10 18:29	-	
📁 zzz-dists/	2019-04-07 22:16	-	

文件传输

如果你想上传及下载文件，需要知道两个命令：

- `put`：用于上传文件。`put` 是英语“放入”的意思。
- `get`：用于下载文件。`get` 是英语“获得”的意思。

如下图所示：



我们来下载一个文件试试（`README`），这个文件就位于 `debian` 这个子目录里，所以你事先需要进入 `debian` 这个子目录（`cd debian`）：

```
ftp> get README
local: README remote: README
227 Entering Passive Mode (212,27,32,66,130,207)
150 Opening BINARY mode data connection for README (1181 bytes).
226 Transfer complete.
1181 bytes received in 0.00 secs (9.0103 MB/s)
```



```

-rw-r--r-- 1 1000 1000 1181 Apr 27 08:28 README
-rw-r--r-- 1 1000 1000 1290 Jun 26 2010 README.CD-manufacture
-rw-r--r-- 1 1000 1000 2898 Apr 27 08:28 README.html
-rw-r--r-- 1 1000 1000 291 Mar 04 2017 README.mirrors.html
-rw-r--r-- 1 1000 1000 86 Mar 04 2017 README.mirrors.txt
drwxr-xr-x 14 1000 1000 4096 Apr 27 08:29 dists
drwxr-xr-x 4 1000 1000 4096 May 26 13:52 doc
-rw-r--r-- 1 1000 1000 178616 May 26 14:37 extrafiles
drwxr-xr-x 3 1000 1000 8192 May 26 14:32 indices
-rw-r--r-- 1 1000 1000 10739774 May 26 14:32 ls-lR.gz
drwxr-xr-x 5 1000 1000 62 Dec 19 2000 pool
drwxr-xr-x 4 1000 1000 67 Nov 17 2008 project
drwxr-xr-x 3 1000 1000 77 Oct 10 2012 tools
drwxr-xr-x 14 1000 1000 4096 Apr 07 20:16 zzz-dists
226 Directory send OK.
ftp> get README
local: README remote: README
227 Entering Passive Mode (212,27,32,66,130,207)
150 Opening BINARY mode data connection for README (1181 bytes).
226 Transfer complete.
1181 bytes received in 0.00 secs (9.0103 MB/s)
ftp>

```

下载完毕，README 文件现在位于你的 Ubuntu 系统的当前目录了。

你连接到 FTP 服务器之后，又想要在自己的电脑上运行命令，怎么办呢？

只要在命令前加一个感叹号就可以了，例如：

```
!ls
```

就会在自己电脑上执行 ls 命令，而不是在 FTP 服务器上执行。例如，执行 `!ls` 命令之后，我们可以看到在自己电脑的当前目录下已经有了刚从 FTP 服务器下载下来的 README 文件：

```

ftp> !ls
chinese.txt          name.txt             README
compression          new_file             redirect
date                newly_created_file   renamed_file
debian-9.9.0-amd64-netinst.iso node-v10.15.3-copy.tar.gz repeat.txt
Desktop             node-v10.15.3.tar.gz results.txt
Documents           nohup.out            share
Downloads           number.txt            snap
emacs-26.2-copy.tar.gz one                   sorted_name.txt
emacs-26.2.tar.gz    one_copy              Templates
errors.log           output_find           test
find_log             Pictures              unique.txt
grep_log             Public                Videos
Music                qq
ftp>

```

passive 模式的时候，如果一段时间没有操作，连接可能会断开，只需要重新连接即可。

如果我想用 put 命令给这个公共 FTP 服务器来传输文件，则会出现“Permission denied”的错误，表示“权限被否决”，因为我没有权限给这个 FTP 服务器写入文件，毕竟我是匿名登录，而且公共的 FTP 服务器如果任由写入文件的话，那早就乱套了。

```

ftp> put repeat.txt
local: repeat.txt remote: repeat.txt
227 Entering Passive Mode (212,27,32,66,211,173)
550 Permission denied.

```

其它的 FTP 命令

还有好些其它的 FTP 命令，我们就不逐一列举了。

用 man ftp 来看看其它可用的命令吧。你会发现，并不是所有的命令都与你到目前学过的 Linux 命令是一样的。例如，删除文件不是用 rm 命令，而是 delete 命令。

要从 FTP 服务器断开连接，你可以用 Ctrl + D 组合键。也可以用 bye、exit 或 quit 命令，效果是一样的。

sftp: 安全加密的 ftp

ftp 命令虽然方便，但是有一个致命缺点：不安全，数据不是加密传输的。任何人，只要连接到同一个网络，可以想办法截取到你传输的数据，或者你的密码。

因此，我们需要请出 sftp。sftp 是 Secure FTP 的缩写。表示“安全的FTP”。

sftp 也是基于 SSH 的，所以登录需要用户名和密码，用法如下：

```
sftp user@ip
```

例如：

```
sftp oscar@xxx.com
```

一旦你输入用户名和密码，连接上之后，其它操作和 ftp 是一样的，只不过通信被加密了，更安全。

用 man sftp 来看看其它可用的命令和参数吧。

上述命令中，我们并没有指定用哪个端口，只指定了 ip 地址。默认的端口号是 22，和 SSH 一样。我们也可以修改端口号，用 -oPort 参数。例如：

```
sftp -oPort 3592 oscar@89.231.45.67
```

5. rsync：同步备份

rsync 命令易于使用，功能很强大。

rsync 是一个小程序，需要安装，系统一般默认没有这个命令（但最新版 Ubuntu 系统自带了）：

```
sudo apt install rsync
```

rsync 命令使我们可以同步两个目录，不管这两个目录位于同一台电脑还是不同的电脑（用网络连接）。

rsync 应该是最常用于“增量备份”的命令了吧。什么是“增量备份”呢？

增量备份（incremental backup）是备份的一种类型，指在一次全备份或上一次增量备份后，以后每次的备份只需备份与前一次相比增加或者被修改的文件。

备份有什么好处呢？

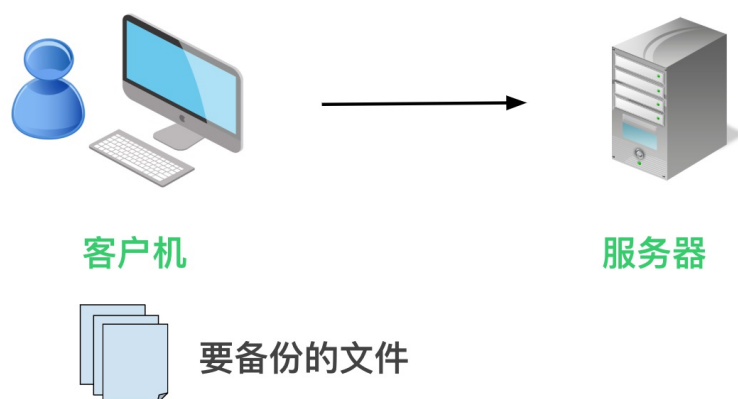
想象一下，假如你不备份文件，那么一旦你的个人电脑“遭遇不测”，例如坏了、被偷等等，那么你的数据就找不回来了。

如果有写了几十页的论文在里面，那“哭倒长城”都有可能啊。

我的几个朋友就有过这样惨痛的经历。

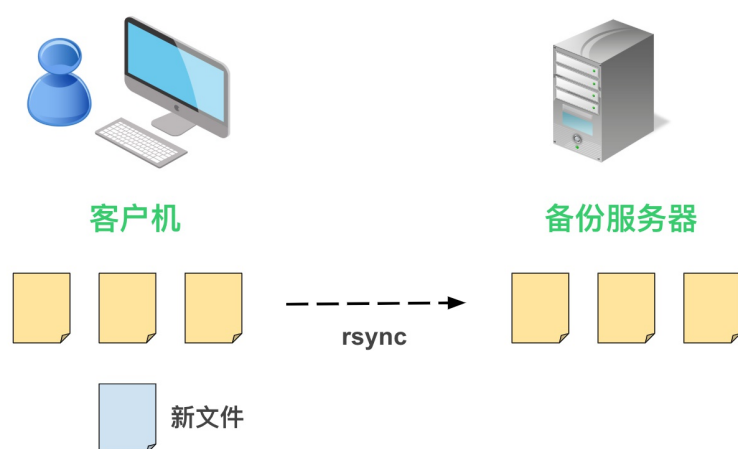
我自己一直有备份的习惯，而且我备份不止会在一个地方。有的重要文件，会备份到移动硬盘，电脑的本地其它硬盘、邮箱等等，危机意识是很重要的。

所以备份很重要。假如你把你电脑上的文件备份到远程服务器上，那么如下图所示：



用 `rsync` 来进行备份，是非常方便的。假如，你把自己的用户家目录都备份到服务器上，也许有十几个 G 的内容。

第一次备份时，需要传输这整整十几个 G 的内容。但是以后呢，只需要传输新增或修改的内容就够了，不需要再传一遍。这就是 `rsync` 的强大之处，所谓“增量备份”的好处。



如上图所示，我用 `rsync` 只传输了新的那个文件，其它的并没有再传输。

`rsync` 就好像更智能的 `scp` 命令。

备份到同一台电脑的其它目录

```
rsync -arv Images/ backups/
```

以上命令，将 `Images` 目录下的所有文件备份到 `backups` 目录下。

`-arv` 参数分别表示：

- `-a`：保留文件的所有信息，包括权限、修改日期等等。`a` 是 `archive` 的缩写，是“归档”的意思；

- -r: 递归调用，表示子目录的所有文件也都包括。r 是 recursive 的缩写，是“递归的”的意思；
- -v: 冗余模式，输出详细操作信息。v 是 verbose 的缩写，是“冗余的”的意思。

删除文件

默认地，rsync 在同步时并不会删除目标目录的文件。例如你的源目录（被同步目录）中删除了一个文件，但是用 rsync 同步时，它并不会删除同步目录中的相同文件。

如果要使 rsync 也同步删除操作。那么可以这么做：

```
rsync -arv --delete Images/ backups/
```

加上 --delete 参数就可以了。delete 是英语“删除”的意思。

备份到另一台电脑的目录

例如：

```
rsync -arv --delete Images/ oscar@89.231.45.67:backups/
```

是不是很简单呢。

至于更多参数，可以用 man rsync 学习。

当然，rsync 的强大之处绝不止于此。

你可以自己配置 rsync，使得它从指定目录（可以是多个目录）备份到指定的 ip 地址的目录下，而且可以指定哪些类型文件是要备份的，哪些类型不要备份，然后把这一长串命令统一用 Shell 脚本来写成一个文件（例如取名叫 backup），使之可执行（用 `chmod +x` 命令），再把这个文件的路径添加到 PATH 中。

这样你以后不论在哪个目录下输入 backup，rsync 就帮你自动同步该目录了，非常帅气。

至于怎么做，就算是留给大家的课后兴趣作业咯。

提示：可以自己百度或 Google，例如“Ubuntu 下 rsync 配置”。

或者你可以等学了第五部分的 Shell 编程之后再尝试。

6. 总结

1. wget 命令可以下载文件；
2. 为了将文件从一台电脑拷贝到另一台电脑，我们可以使用 scp 命令。它使用上一课提到的 SSH，因此传输是加密的，非常安全；
3. 我们可以用 ftp 命令来连接到一个 FTP 服务器，就可以上传及下载文件了；
4. sftp 和 ftp 命令类似，但是它用了 SSH，所以传输的信息是加密的；
5. rsync 命令可以同步同一台电脑或两台不同电脑上的两个文件（夹）的内容，用 rsync 命令来备份文件特别方便。

今天的课就到这里，一起加油吧！

