

形式为 protocol://DNS-name/file-name。http协议（超文本传输协议，HyperText Transfer Protocol）是最常用的，不过ftp和其他协议也在使用。协议名后面是拥有该文件的主机的DNS名称。最后是一个本地文件名，用来说明需要使用哪个文件。

整个系统按如下方式结合在一起：Web根本上是一个客户机-服务器系统，用户是客户端，而Web站点则是服务器。当用户给浏览器提供一个URL时（或者键入URL，或者点击当前页面上的某个超链接），浏览器则按照一定的步骤调取所请求的Web页面。作为一个例子，假设提供的URL是 `http://www.minix3.org/doc/faq.html`。浏览器按照下面的步骤取得所需的页面。

- 1) 浏览器向DNS询问www.minix3.org的IP地址。
- 2) DNS回答，是130.37.20.20。
- 3) 浏览器建立一个到130.37.20.20上端口80的TCP连接。
- 4) 接着浏览器发送对文件doc/faq.html的请求。
- 5) www.acm.org服务器发送文件doc/faq.html。
- 6) 释放TCP连接。
- 7) 浏览器显示doc/faq.html文件中的所有文本。
- 8) 浏览器获取并显示doc/faq.html中的所有图像。

大体上，这就是Web的基础以及它是如何工作的。许多其他的功能已经添加在了上述基本Web功能之上了，包括样式表、可以在运行中生成的动态网页、带有可在客户机上执行的小程序或脚本的页面等，不过对它们的讨论超出了本书的范围。

8.4.4 基于文件系统的中间件

隐藏在Web背后的基本思想是，使一个分布式系统看起来像一个巨大的、超链接的集合。另一种处理方式则是使一个分布式系统看起来像一个大型文件系统。在这一节中，我们将考察一些与设计一个广域文件系统有关的问题。

分布式系统采用一个文件系统模型意味着只存在一个全局文件系统，全世界的用户都能够读写他们各自具有授权的文件。通过一个进程将数据写入文件而另一个进程把数据读出的办法可以实现通信。由此产生了标准文件系统许多问题，但是也有一些与分布性相关的新问题。

1. 传输模式

第一个问题是，在上传/下载模式（upload/download model）和远程访问模式之间的选择问题。在前一种模式中，如图8-36a所示，通过把远程服务器上的文件复制到本地的方法，实现进程对远程文件的访问。如果只是需要读该文件，考虑到高性能的需要，就在本地读出该文件。如果需要写入该文件，就在本地写入。进程完成工作之后，把更新后的文件送回原来的服务器。在远程访问模式中，文件停留在服务器上，而客户机向服务器发出命令并在服务器上完成工作，如图8-36b所示。

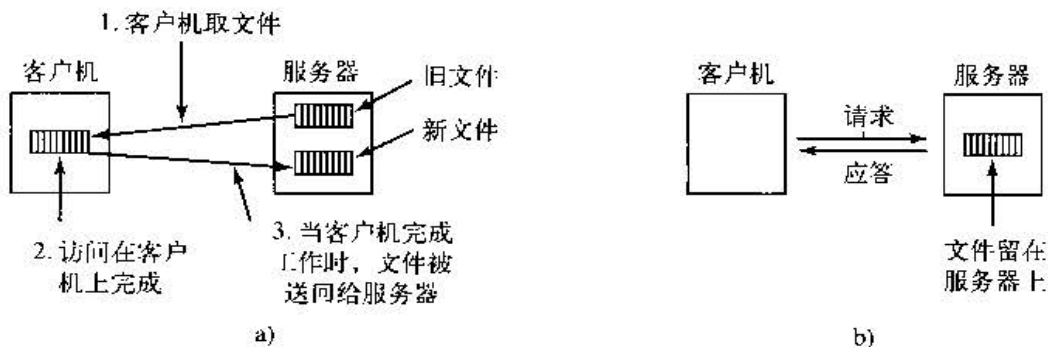


图8-36 a) 上传/下载模式；b) 远程访问模式

上传/下载模式的优点是简单，而且一次性传送整个文件的方法比用小块传送文件的方法效率更高。其缺点是为了在本地存放整个文件，必须拥有足够的空间，即使只需要文件的一部分也要移动整个文件，这样做显然是一种浪费，而且如果有多个并发用户则会产生一致性问题。