

第20章 使用网络函数和协议函数

在本章中，我们将详细介绍PHP所提供的面向网络的函数，这些函数能够使脚本与Internet进行交互。Internet是一个资源丰富的世界，我们可以通过各种协议来利用这些资源。在本章中，我们将主要介绍以下内容：

- 了解可供使用的协议
- 发送和读取电子邮件
- 使用其他Web站点的数据
- 使用网络查找函数
- 使用FTP

20.1 了解可供使用的协议

协议是指在给定情况下的通信规则。例如，当遇到另一个人时，我们会采用这种协议：我们会说“你好”，与他握手，交谈一会儿，然后说再见。不同的情况需要不同的协议。此外，来自不同文化的人们会使用不同协议，这样交流就会变得困难。计算机网络协议也与此类似。

与人类协议一样，不同的计算机协议适用于不同的情况和应用程序。例如，当请求和接收一个Web页面时，我们使用超文本传输协议（Hypertext Transfer Protocol，HTTP）——你的计算机向Web服务器请求一个文档（例如，HTML或PHP文件），而服务器将通过向你的计算机发送文档作为响应。我们也可能使用过文件传输协议（File Transfer Protocol，FTP）在网络上的机器之间传输文件。除此之外，还有许多其他用途的协议。

文档征求意见稿（*Requests For Comments*，RFC）描述了各种协议和其他Internet标准。Internet工程任务组（Internet Engineering Task Force，IETF）定义了这些协议。在Internet上，可以找到大量关于不同标准的RFC。相关的基本资源是RFC Editor，其URL如下所示：

<http://www.rfc-editor.org/>

如果在使用给定的协议时出现问题，定义该协议的文档是最权威的，而且经常用于解决代码中的问题。然而，它们非常详细，其内容经常会有好几百页。

一些非常著名的RFC例子是RFC2616和RFC822，它们分别描述了HTTP/1.1协议和Internet电子邮件格式。

在本章中，我们将了解PHP中使用这些协议的函数及其应用。此外，本章还将特别介绍如何通过SMTP发送邮件，通过POP3和IMAP4读取邮件，如何使用HTTP和其他Web服务器连接，以及如何使用FTP传输文件。

20.2 发送和读取电子邮件

在PHP中，发送邮件的主要方法是使用简单的mail()函数。在第4章“字符串操作与正则

表达式”中，我们已经讨论过此函数的使用。因此，本章将不再深入讨论。这个函数使用简单邮件传输协议（Simple Mail Transfer Protocol, SMTP）来发送邮件。

我们可以使用许多免费的类为mail()函数添加新功能。在第30章“创建一个邮件列表管理器”中，我们将使用一个插件模式的邮件类来发送带有HTML附件的邮件。SMTP只能用来发送邮件。而Internet邮件访问协议（Internet Message Access Protocol, IMAP4，在RFC2060中有详细定义）和邮政协议（Post Office Protocol, POP3，在RFC1939或STD0053中有详细定义）则可以用来读取特定邮件服务器上的邮件。这些协议不能发送邮件。

IMAP4用于读取和操作存储在服务器上的邮件，它比POP3更复杂，通常情况下，POP3适用于将邮件下载到客户端，并从服务器上删除它们。

PHP本身带有IMAP4函数库。这个函数库也可以用来实现POP3和NNTP（Network News Transfer Protocol，网络新闻传输协议）的连接，与建立IMAP4连接一样。

在第29章“创建一个基于Web的电子邮件服务系统”所介绍的示例项目中，将详细介绍IMAP库的使用。

20.3 使用其他Web站点的数据

通过网络能实现的最重要的一件事就是可以在用户自己的页面中使用、修改和嵌入已有服务和信息。PHP使这些操作变得非常简单。下面，我们举例说明。

假设你所在的公司希望本公司的股票价格显示在公司的主页上。股票价格信息可以从证券交易所网站的某个地方获得——但是我们如何得到它呢？

首先，必须找到原始信息的URL。当获知该信息后，在每次有人进入你的网页时，可以打开指向该URL的连接，获取该网页，提取需要的信息。

作为一个例子，我们将在一个脚本中实现此功能，它能够从AMEX站点获取和重新格式化一支股票的价格信息。要实现本例的功能，我们必须从以上站点获得Amazon.com的股票价格（你要包含在网页中的信息可能与此不同，但原理是一样的）。

这里的示例代码将通过其他站点提供的Web服务器来获取数据，并显示在我们的站点上。该脚本如程序清单20-1所示。

程序清单20-1 lookup.php——从NASDAQ获得\$symbol列表所给出股票的报价脚本

```
<html>
<head>
  <title>Stock Quote From NASDAQ</title>
</head>
<body>
<?php
//choose stock to look at
$symbol = 'AMZN';
echo '<h1>Stock quote for' . $symbol . '</h1>';

$url = 'http://finance.yahoo.com/d/quotes.csv'.
      '?s= ' . $symbol . '&e=.csv&f=s1ld1t1cl0hg';
```

```

if (!($contents = file_get_contents($url))) {
    die('Failure to open ' . $url);
}

// extract relevant data
list($symbol, $quote, $date, $time) = explode(',', $contents);
$date = trim($date, ' ');
$time = trim($time, ' ');

echo "<p>" . $symbol . " was last sold at: " . $quote . "</p>";
echo "<p>Quote current as of " . $date . " at " . $time . "</p>";

// acknowledge source
echo "<p>This information retrieved from <br /><a href='" . $url .
    "'>" . $url . "</a>.</p>";
?>
</body>
</html>

```

以上程序清单的运行结果如图20-1所示。

此脚本本身是非常简单的。事实上，该脚本并没有使用任何我们没有见到过的新函数，而只是那些函数的新应用。

回忆一下，在第2章“数据的存储与检索”中，当介绍如何从文件读信息时，我们提到过可以使用文件函数从URL中读取信息。这正是我们在这个例子中所实现的。调用函数file_get_contents()：

```
if (!($contents = file_get_contents($url))) {
```

将返回指定URL的Web页面所有内容，并且将其保存在\$contents变量中。

在PHP中，文件函数可以完成许多操作。这里给出的例子只是通过HTTP载入一个Web页面，我们还可以通过HTTPS、FTP或其他协议以几乎相同的方式与其他服务进行交互。

对于某些任务，可能需要采用一些更特殊的方法。在特定的FTP函数中，还提供了一些FTP功能，但却无法通过fopen()或其他函数使用。在本章稍后内容中，将介绍一个使用FTP函数的示例。对于某些HTTP或HTTPS任务，可能需要使用cURL库。通过cURL函数库，可以登录到一个Web站点并且在一些页面中模拟用户的操作。

在通过file_get_contents()函数获得页面文本后，就可以使用list()函数找到我们所需的网页部分：

```
list($symbol, $quote, $date, $time) = explode(',', $contents);
$date = trim($date, " ");
```

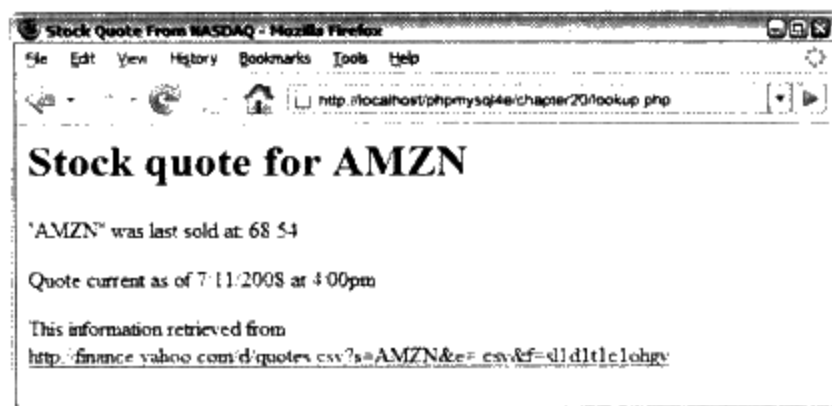


图20-1 该脚本通过一个正则表达式从来自证券交易所网站的信息中摘出股票报价

```
$time = trim($time, ' ');

echo "<p>". $symbol . " was last sold at: " . $quote . "</p>";
echo "<p>Quote current as of " . $date . " at " . $time . "</p>";
```

可以用上述方法实现各种目的。另一个很好的例子就是索取当地的天气信息并把它嵌入网页中。

这种方法的最佳应用是整合来自不同信息源的数据以增加其价值。一个著名的例子是Philip Greenspun的脚本使用了这种方法，此脚本产生了比尔·盖茨的财富时钟：<http://philip.greenspun.com/>。

此网页将分别从两个地方获取信息。首先，它将从U.S.Census Bureau的网址获得美国的当前人口数。然后，再查找Microsoft股票的当前市值，并结合这两条信息，添加一些作者的观点，产生一个新信息——Bill Gates目前财富的估计值。

需要注意的一点是：如果将外面的信息源（像在这个例子中）用于商业目的，最好先查看一下使用这个信息源的合法性。因为在一些情况下，需要考虑一些关于知识产权的问题。

如果要编写一个类似的脚本，可能还希望传输一些其他的数据。例如，如果要连接到一个外部URL，可能需要传输一些用户输入参数。如果打算这样做，使用函数urlencode()是一个好主意。此函数将接受一个字符串并将其转换为适合URL的格式。例如，将空格转换为“+”符号。可以按如下方式调用此函数：

```
$encodedparameter = urlencode($parameter);
```

这种方法的问题是获得信息的源站点可能会改变其数据格式，这样就使脚本无法正常运行。

20.4 使用网络查找函数

PHP提供一组查找函数，这些函数可以用来检查主机名称、IP地址、邮件交换等信息。例如，如果要创建一个目录站点，例如Yahoo!，当用户提交一个新的URL时，我们可能会自动检查URL所指向的主机和此站点的联系信息是否有效。这样，当一个访问者浏览一个网站并发现此网站不存在或电子邮件无效时，可以节省进一步的操作。

程序清单20-2给出了目录提交的HTML表单。

程序清单20-2 directory_submit.html——提交表单的HTML

```
<html>
<head>
  <title>Submit your site</title>
</head>
<body>
<h1>Submit site</h1>
<form method=post action= "directory_submit.php ">
URL: <input type=text name="url"size=30 value="http://"><br />
Email contact: <input type=text name="email"size=23><br />
<input type="submit" name="Submit site">
</form>
```

```
</body>
</html>
```

这是一个非常简单的表单，图20-2所示的就是带有输入样本数据的表单。

当点击提交按钮时，我们首先要检查该URL是否对应一台真实的主机，其次，还要检查电子邮件地址的主机部分是否也对应真实的机器。我们已经编写了这些检查脚本，它的输出结果如图20-3所示。

执行这些检查的脚本使用了两个来自PHP网络函数集中的函数——`gethostbyname()`和`dns_get_mx()`。全部脚本如程序清单20-3所示。

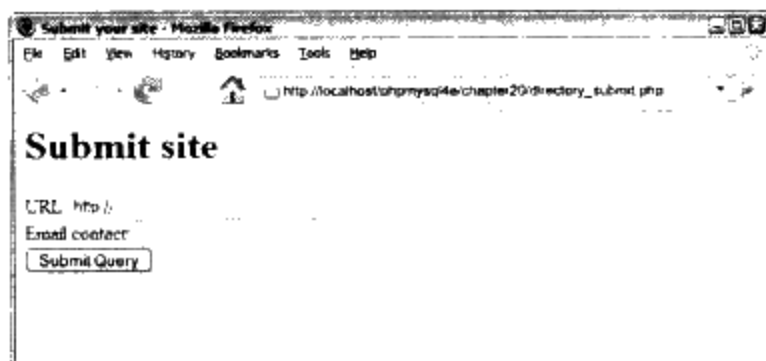


图20-2 目录提交通常要求提交URL和某些详细的联系信息，这样可以方便目录管理员在添加了站点以后通知我们

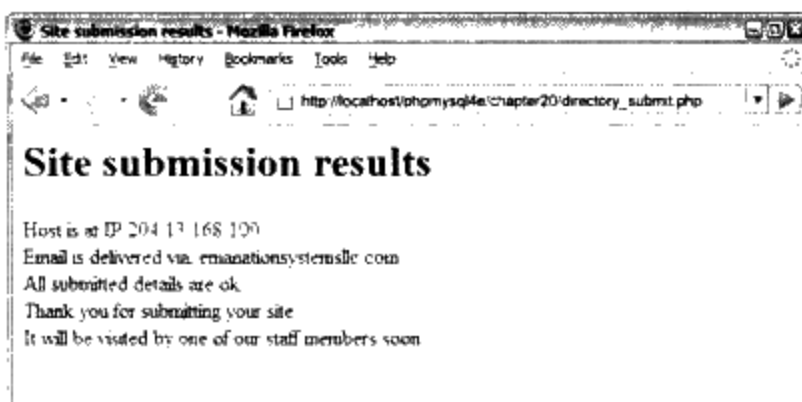


图20-3 此版本的脚本显示对URL主机名和电子邮件地址的检查结果——一个实际的产品版本可能不会显示这些结果，但看看检查返回的信息是非常有趣的

程序清单20-3 directory_submit.php——用于检查URL和电子邮件地址的脚本

```
<html>
<head>
  <title>Site submission results</title>
</head>
<body>
<h1>Site submission results</h1>
<?php

  // Extract form fields

  $url = $_REQUEST[ 'url' ];
  $email = $_REQUEST[ 'email' ];

  // Check the URL

  $url = parse_url($url);
  $host = $url[ 'host' ];
  if ( ! ( $ip = gethostbyname( $host ) ) )
  {
    echo 'Host for URL does not have valid IP';
```

```

        exit;
    }

    echo "Host is at IP $ip <br>";

    // Check the email address

    $email = explode('@', $email);
    $emailhost = $email[1];

    // note that the dns_get_mx() function is *not implemented* in
    // Windows versions of PHP
    if (!dns_get_mx($emailhost, $mxhostsarr))
    {
        echo "Email address is not at valid host";
        exit;
    }

    echo "Email is delivered via: ";
    foreach ($mxhostsarr as $mx)
        echo "$mx ";

    // If reached here, all ok

    echo "<br>All submitted details are ok.<br>";
    echo "Thank you for submitting your site.<br>";
    echo "    , It will be visited by one of our staff members soon."

    // In real case, add to db of waiting sites...
?>
</body>
</html>

```

现在，我们来了解一下这个脚本的有趣部分。

首先，获取URL并将其作为函数`parse_url()`的参数。该函数将返回包含URL不同部分的相关数组。该数组的可用信息部分分别是“模式”、“用户”、“传递”、“主机”、“端口”、“路径”、“查询”和“代码段”(scheme、user、pass、host、port、path、query、fragment)。一般情况下，并不需要所有这些信息。如下所示的例子说明它们是如何组成一个有效的URL。

假设有一个如下所示的URL：`http://nobody:secret@example.com:80/script.php?variable=value#anchor`，数组的每一个元素值分别是：

- scheme: http
- user: nobody
- pass: secret
- host: example.com

- port: 80
- path: /script.php
- query: variable = value
- fragment: anchor

在directory_submit.php脚本中, 我们只需要主机信息, 因此可以按如下操作从数组中取出:

```
$url = parse_url($url);  
$host = $url['host'];
```

在完成以上操作后, 如果主机是在域名服务(DNS)中, 可以获得此主机的IP地址。我们可以通过函数gethostbyname()获得一个主机的IP地址。如果主机存在, 此函数就返回其IP地址, 如果不存在, 函数将返回false:

```
$ip = gethostbyname($host);
```

也可以使用函数gethostbyaddr()来实现, 此函数以IP作为参数并返回主机名。如果顺序地调用这个函数, 最后获得的主机名将与开始时的主机名不同。这可能意味着网站正在使用一个虚拟主机服务。在这个服务中, 一个物理主机和IP地址具有多个域名。

如果URL是有效的, 接下来就需要检查电子邮件地址。首先, 通过调用函数explode()将邮件地址分割成用户名和主机名两部分:

```
$email = explode('@', $email);  
$emailhost = $email[1];
```

当得到主机部分的地址后, 可以通过函数dns_get_mx()检查是否有邮件可以到达的确切地方:

```
dns_get_mx($emailhost, $mxhostsarr);
```

该函数将返回一个邮件地址的一组邮件交换(Mail Exchange, MX)记录, 该地址由数组\$mxhostsarr提供。

MX记录存储在DNS中, 而且其查找方式类似于主机名的查找方式。MX列出的机器不一定是邮件最终到达的机器。相反, 它是一台知道邮件发送路由的机器(可能不止一台, 因此这个函数返回一个数组而不是一个主机名字符串)。如果DNS中没有MX记录, 那么该邮件就没有可发送的目的地。

请注意, 在PHP的Windows版本中, 并没有实现dns_get_mx()函数。如果使用Windows, 必须使用PEAR::Net_DNS包(http://pear.php.net/package/NET_DNS), 该包包含了具有此功能的函数。

如果这些检查都正确, 可以将这个表单的数据输入到数据库, 以便工作人员以后复查。

除了刚才已经使用的函数外, 我们还可以使用更普通的函数checkdnsrr(), 这个函数以主机名作为参数, 如果在DNS中有其记录, 函数将返回true。

20.5 备份或镜像一个文件

文件传输协议（FTP）是用来在网络主机之间传输文件的协议。就像使用HTTP连接一样，使用PHP，可以在FTP中使用`fopen()`函数和其他不同的文件函数来连接一个FTP服务器，并在客户端和FTP服务器之间传送文件。但是，标准的PHP安装也提供了一整套专门适用于FTP的函数。

在默认情况下，这些函数并没有内置在PHP的标准安装中。要在UNIX下使用这些函数，必须运行带有`--enable-ftp`命令选项的`configure`程序，然后再次运行`make`文件。如果使用标准的Windows安装，FTP函数将被自动启用。（关于如何配置PHP的详细信息，请参阅附录A“安装PHP和MySQL”。）

20.5.1 使用FTP备份或镜像一个文件

对于从一台主机到另一台主机之间的文件移动和复制来说，FTP函数是非常有用的。FTP最常见的应用可能就是在另一个位置备份Web站点或者镜像文件。接下来，我们来了解一个使用FTP函数镜像文件的简单例子，如程序清单20-4所示。

程序清单20-4 `ftp_mirror.php`——一个从FTP服务器下载新版本文件的脚本

```
<html>
<head>
  <title>Mirror update</title>
</head>
<body>
<h1>Mirror update</h1>
<?php
// set up variables - change these to suit application
$host = 'ftp.cs.rmit.edu.au';
$user = 'anonymous';
$password = 'me@example.com';
$remoteFile = '/pub/tsg/teraterm/ttssh14.zip';
$localFile = '/tmp/writable/ttssh14.zip';

// connect to host
$conn = ftp_connect($host);
if (!$conn)
{
    echo "Error: Could not connect to ftp server<br />";
    exit;
}
echo "Connected to $host.<br />";

// log in to host
$result = @ftp_login($conn, $user, $pass);
if (!$result)
{
```

```
    echo "Error: Could not log on as $user<br />";
    ftp_quit($conn);
    exit;
}
echo "Logged in as $user<br />";

// check file times to see if an update is required
echo "Checking file time...<br />";
if (file_exists($localfile))
{
    $localtime = filemtime($localfile);
    echo "Local file last updated ";
    echo date('G:i j-M-Y', $localtime);
    echo "<br />";
}
else
    $localtime=0;
$remotetime = ftp_mdtm($conn, $remotefile);
if (($remotetime >= 0))
{
    // This doesn't mean the file's not there, server may not support mod
time
    echo "Can't access remote file time.<br />";
    $remotetime=$localtime+1; // make sure of an update
}
else
{
    echo "Remote file last updated ";
    echo date('G:i j-M-Y', $remotetime);
    echo "<br />";
}
if (($remotetime > $localtime))
{
    echo "Local copy is up to date.<br />";
    exit;
}

// download file
echo "Getting file from server...<br />";
$fsp = fopen ($localfile, 'w');
if (!$success = ftp_fget($conn, $fsp, $remotefile, FTP_BINARY))
{
    echo "Error: Could not download file";
    ftp_quit($conn);
    exit;
}
fclose($fsp);
```

```

echo "File downloaded successfully";

// close connection to host
ftp_quit($conn);

?>
</body>
</html>

```

在特定情况下，运行以上脚本的输出结果如图20-4所示。

这是一个非常普通的脚本。可以看到，在此脚本的开始处设置了一些变量：

```

$host = 'ftp.cs.rmit.edu.au';
$user = 'anonymous';
$password = 'me@example.com';
$remote_file = '/pub/tsg/teraterm/ttssh14.zip';
$local_file = '/tmp/writable/ttssh14.zip';

```

变量\$host包含了即将连接的FTP服务器的名称，\$user和\$password对应于登录的用户名和密码。

许多FTP网站都支持匿名登录(anony-mous)，匿名登录是指任何人都可以使用这个用户名来连接FTP服务器。匿名登录不需要密码，但通常的做法是把电子邮件作为密码，这样管理员能够知道用户来自什么地方。在这里，我们沿袭了这个惯例。

变量\$remote_file包含我们将要下载的文件的路径。在这个例子中，我们要下载并镜像Tera Term SSH的一个本地拷贝，以及一个适用于Windows平台的SSH客户端（SSH表示安全Shell。它是Telnet的加密形式）。

变量\$local_file包含了下载文件存储在机器上的本地路径。在这个例子中，我们创建了/tmp/writable目录，该目录的权限设置允许PHP对其执行写操作，也就是可以写入一个文件。无论操作系统是什么，必须创建这个目录，这样这个脚本才能正确运行。如果操作系统具有严格的权限机制，必须确认它们允许这个脚本执行写操作。我们应该能够根据需要修改这些变量和脚本。

该脚本的基本执行步骤与在命令行中手动使用FTP传输文件一样：

1. 连接远程FTP服务器。
2. 登录（通过用户名或匿名登录）。
3. 检查远程文件是否已经更新。
4. 如果更新过，下载此文件。
5. 关闭FTP连接。

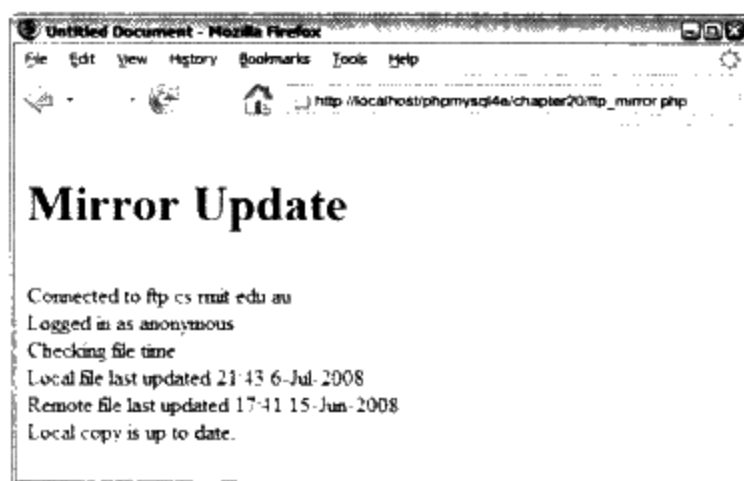


图20-4 该FTP镜像脚本将检查文件的本地版本是否是最新版本，如果不是，就下载一个新的版本

下面，让我们依次讨论这些步骤。

1. 连接远程FTP服务器

这一步与在Windows或者UNIX平台的命令提示符下输入如下命令等价：

```
ftp hostname
```

在PHP中，通过如下所示代码完成此步骤：

```
$conn = ftp_connect($host);  
if (!$conn)  
{  
    echo "Error: Could not connect to ftp server<br />";  
    exit;  
}  
echo "Connected to $host.<br />";
```

在这里，我们调用的函数是`ftp_connect()`。此函数以主机名作为参数并返回连接的句柄或者`false`（如果不能建立连接）。这个函数也能够以要连接的主机端口号作为可选的第二个参数（这里没有用到）。如果不指定一个端口号，此函数将使用默认值21（FTP的默认端口）。

2. 登录到FTP服务器

下一步是使用一个使用特定用户名和密码登录到FTP服务器上。可以通过函数`ftp_login()`来完成此步骤：

```
$result = @ftp_login($conn, $user, $pass);  
if (!$result)  
{  
    echo "Error: Could not log on as $user<br />";  
    ftp_quit($conn);  
    exit;  
}  
echo "Logged in as $user<br />";
```

这个函数需要3个参数：一个FTP连接句柄（通过函数`ftp_connect()`获得）、用户名和密码。如果用户能够登录，此函数将返回`true`，如果不能登录，函数将返回`false`。注意我们在这行的开始处放置一个“@”符号来抑制错误。这是因为如果用户不能登录，他将在浏览器窗口中获得一个PHP警告信息。通过测试`$result`变量的内容，可以捕获这个错误，并可以自己定制更多的、更友好的用户出错信息。

请注意，如果登录尝试失败，实际上将通过`ftp_quit()`函数关闭FTP连接，稍后将详细介绍该函数。

3. 检查文件更新时间

假设我们打算更新文件的本地副本，比较明智的做法是首先检查文件是否需要更新，因为如果文件是最新的，就无须重新下载此文件，特别是当它是一个很大的文件的时候。这将避免不必要的网络通信量。现在，让我们来查看实现检查文件更新时间的代码。

文件时间是我们使用FTP函数而不是其他更简单的文件函数的原因。文件函数可以很容易读，而且在某些情况下，也可以通过网络接口写文件，但是大多数状态函数，例如`filemtime()`，

无法执行远程操作。这一点将在以后发生变化。

要确定是否需要下载一个文件，可以通过函数`file_exists()`确认一个文件的本地副本是否存在。如果没有，显然需要下载此文件。如果此文件存在，通过函数`filemtime()`获得文件的最后修改时间，并把它存储在变量`$localtime`中。如果此文件不存在，设置变量`$localtime`为0，这样，此文件将会比任何可能的远程文件修改时间都要“老”。

```
echo 'Checking file time...<br />';
if (file_exists($localfile))
{
    $localtime = filemtime($localfile);
    echo 'Local file last updated ';
    echo date('G:i j-M-Y', $localtime);
    echo '<br />';
}
else
    $localtime=0;
```

(在第2章“数据的存储与检索”和第19章“与文件系统和服务器的交互”中，我们已经分别介绍了函数`file_exists()`和`filemtime()`的使用)。

解决了本地时间的问题后，我们需要获得远程文件的修改时间。可以通过函数`ftp_mdtm()`获得此信息，如下所示：

```
$remotetime = ftp_mdtm($conn, $remoteFile);
```

这个函数带有两个参数（FTP连接句柄和远程文件的路径），并返回文件最后修改时间的UNIX时间戳或者-1（如果存在某种错误的话）。不是所有的FTP服务器都支持这种特性，因此通过这个函数可能得不到一个有用的结果。在这种情况下，可以选择手动设置，将变量`$localtime`加1，使变量`$remotetime`比`$localtime`“新”。这样，就可以确保能够下载文件，如下代码所示：

```
if (($remotetime >= 0))
{
    // This doesn't mean the file's not there, server may not support mod time
    echo 'Can't access remote file time.<br>';
    $remotetime=$localtime+1; // make sure of an update
}
else
{
    echo 'Remote file last updated ';
    echo date('G:i j-M-Y', $remotetime);
    echo '<br>';
}
```

当我们拥有二者的时间时，就可以对它们进行比较，确认是否需要下载这个文件：

```
if (($remotetime > $localtime))
{
    echo 'Local copy is up to date.<br>';
```

```
    exit;
}
```

4. 下载文件

到这一步，我们就可以从服务器上下载文件。

```
echo 'Getting file from server...<br>';
$fp = fopen ($localfile, 'w');
if (!$success = ftp_fget($conn, $fp, $remotefile, FTP_BINARY))
{
    echo 'Error: Could not download file';
    fclose($fp);
    ftp_quit($conn);
    exit;
}
fclose($fp);
echo 'File downloaded successfully';
```

正如我们在前面代码所见到的，我们通过函数fopen()打开一个本地文件。在成功打开该文件后，调用函数ftp_fget()，该函数将下载这个文件并存储到本地文件中。ftp_fget()函数带有4个参数。前3个非常简单：FTP连接句柄、本地文件句柄和远程文件路径。第4个参数是FTP模式。

对于FTP传输，有两种模式：ASCII和二进制。ASCII模式用于传输文本文件（也就是说，文件全部由ASCII字符组成），二进制模式用于传输所有其他类型的文件。二进制模式在传输一个文件的过程中不会修改这个文件，而ASCII模式将回车换行字符转换成适用于系统的特定字符（在UNIX下为\n，Windows下为\r\n，而Macintosh下为\r）。

PHP的FTP库有两个预定义的常量，FTP_ASCII和FTP_BINARY，它们分别代表这两种模式。我们需要确定哪种模式适合文件，并将相应的常量作为第4个变量传给函数ftp_fget()。在这个例子中，传输的是一个压缩文件，因此我们使用了FTP_BINARY模式。

如果所有的这些操作都是正常执行，ftp_fget()函数将返回true；如果遇到错误，此函数将返回false。函数执行结果存储在变量\$success中，这样用户可以知道此函数的运行情况。

在文件下载完成之后，需要调用函数fclose()关闭本地文件。

除了ftp_fget()函数外，还可以使用函数ftp_get()来实现此目的，此函数具有如下所示的原型：

```
int ftp_get (int ftp_connection, string localfile_path,
             string remotefile_path, int mode)
```

此函数在许多方面与函数ftp_fget()相同，但是它不需要打开本地文件。传递给此函数的参数为本地文件的系统文件名而不是文件句柄。

请注意，PHP中还没有与FTP命令mget等价的函数，此命令能够一次下载多个文件。我们必须多次调用ftp_fget()函数或者ftp_get()函数来代替mget命令。

5. 关闭连接

在完成FTP连接之后，应该通过函数`ftp_quit()`来关闭此连接。

```
ftp_quit($conn);
```

你必须将FTP连接的句柄传递给此函数。

20.5.2 上传文件

如果想执行下载文件的反向操作，也就是要将文件从自己的服务器上复制到远程的机器上，可以使用两个函数，它们的功能基本上与函数`ftp_fget()`和`ftp_get()`相反。这两个函数就是`ftp_fput()`和`ftp_put()`。其函数原型如下所示：

```
int ftp_fput (int ftp_connection, string remotefile_path, int fp, int mode)
```

```
int ftp_put (int ftp_connection, string remotefile_path,  
             string localfile_path, int mode)
```

这些参数与`_get`函数的参数一样。

20.5.3 避免超时

当通过FTP传输文件时，可能会遇到的一个问题就是超过最大可执行时间。这种情况是否会发生是可以预知的，因为PHP将给出一个出错信息。如果服务器运行在一个非常慢或拥挤的网络上，或者正在下载一个很大的文件（如电影剪辑），这种情况就很可能发生。

配置文件`php.ini`定义了所有的PHP脚本最大可执行时间的默认值。此默认值设置为30s。

这样，就可以避免脚本失去控制地运行。然而，当正在通过FTP传输文件时，如果到其他网址的连接速度非常慢，或者文件很大时，文件传输需要的时间可能就会比这个默认值要大。

幸运的是，对于一个特定的脚本，我们可以通过函数`set_time_limit()`来修改这个最大可执行时间。调用这个函数将重新设置脚本所允许的最大可执行时间（以s为单位），计时是从调用此函数的时候开始。例如，如果调用函数：

```
set_time_limit(90);
```

这样，这个脚本将从这个函数的调用开始重新运行90s。

20.5.4 使用其他的FTP函数

在PHP中，还有许多其他非常有用的FTP函数。函数`ftp_size()`能够显示远程服务器上一个文件的大小。此函数原型如下所示：

```
int ftp_size(int ftp_connection, string remotefile_path)
```

此函数返回远程文件的字节数或者-1（如果有错误的话）。但是，该函数并不被所有的服务器所支持。

函数`ftp_size()`的一个实际应用是计算出在特定传输速率一次文件传输所需的最大可执行时间。如果知道文件大小和连接速度，可以估算出传输可能需要的时间，并可以根据实际情

况调用函数`set_time_limit()`。

通过如下所示代码，可以获得并显示远程FTP服务器特定目录上的文件列表：

```
$listing = ftp_nlist($conn, dirname($remoteFile));  
foreach ($listing as $filename)  
    echo "$filename <br>";
```

此代码通过函数`ftp_nlist()`的调用获得特定目录中的文件名的列表。

对于FTP函数，凡是可能通过FTP命令行完成的事情，几乎都可以通过FTP函数来完成。

在PHP在线指南中，可以找到对应于每个FTP命令的特殊函数，其URL如下所示：<http://php.net/manual/en/ref.ftp.php>。

一个例外是命令`mget`（multiple get，也就是获得多个文件），但是可以通过函数`ftp_nlist()`获得文件列表并根据需要选择文件。

20.6 进一步学习

在本章中，我们介绍了许多基础知识。关于这些主题，还有许多内容。有关个别协议的信息和它们的工作原理，可以参考相应的RFC标准，其网址如下所示：

<http://www.rfc-editor.org/>。

也可以在W3C组织找到关于某些协议的信息：

<http://www.w3.org/Protocols/>。

还可以参考有关TCP/IP的书籍，例如，由Andrew Tanenbaum编著的《计算机网络》。

20.7 下一章

我们可以进入第21章的学习了。在第21章“日期和时间的管理”中，我们将讨论PHP日期和日历函数。我们将了解如何把用户输入的格式转换为PHP格式，再转换为MySQL格式，以及进行相互的转换。