Programming Spiders, Bots, and Aggregators in Java

网络机器人Java编程指南

用Java实现网络机器人、自动执行复杂的Web交互

[美] Jeff Heaton 著

童兆丰　李 纯　刘润杰 译

SYBEX

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Programming Spiders, Bots, and Aggregators in Java

网络机器人Java编程指南

〔美〕 Jeff Heaton 著

童兆丰 李 纯 刘润杰 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 提 要

这是一本研究如何实现具有Web访问能力的网络机器人的书。该书从Internet编程的基本原理出发，深入浅出，循序渐进地阐述了网络机器人程序Spider、Bot、Aggregator的实现技术，并分析了每种程序的优点及适用场合。本书提供了大量的有效源代码，并对这些代码进行了详细的分析。通过本书的介绍，你可以很方便地利用这些技术，设计并实现网络蜘蛛或网络信息搜索器等机器人程序。

本书通俗易懂，适合于具有一定Java编程基础的软件开发人员阅读，也可供Web开发人员作为技术参考资料使用。

版权贸易合同登记号：01-2002-0900

图书在版编目（CIP）数据

网络机器人Java编程指南/（美）希顿（Heaton, J.）著；童兆丰，李纯，刘润杰译. - 北京：电子工业出版社，2002.7

书名原文：Programming Spiders, Bots, and Aggregators in Java

ISBN 7-5053-7740-X

Ⅰ. 网… Ⅱ. ①希… ②童… ③李… ④刘… Ⅲ. Java语言 - 程序设计 Ⅳ. TP312

中国版本图书馆CIP数据核字（2002）第043562号

责任编辑：徐云鹏

印　　刷：北京天竺颖华印刷厂

出版发行：电子工业出版社　http://www.phei.com.cn

北京市海淀区万寿路173信箱　邮编：100036

北京市海淀区翠微东里甲2号　邮编：100036

经　　销：各地新华书店

开　　本：787×1092　1/16　印张：27.25　字数：700千字

版　　次：2002年7月第1版　2002年7月第1次印刷

定　　价：44.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换，若书店售缺，请与本社发行部联系。联系电话：（010）68279077

谨以此书献给我的祖父母：Agnes Heaton和Roscoe Heaton，以及Emil A. Stricker和Esther Stricker!

致　谢

本书的出版得益于很多人的帮助，在这不可能都一一提及，但我还是要特别感谢为本书做出主要贡献的人。

与Sybex公司合作完成这部书是一个愉快的过程，每一个参与其中的人都很专业，而且随和。我首先要感谢技术编辑Marc Goldford先生，他提出了很多有用的建议，并且测试了书中所有的示例。另外，我要感谢编辑Rebecca Rider女士，由于她出色的工作，使得全书内容清晰，可读性强。Diane Lowery女士是我的acquisitions editor，同样非常感谢她在出书的初期提供的帮助。我还要感谢整个制作组的成员，包括：制作编辑Dennis Fitzgerald，电子出版专家Jill Niles和Judy Fung，以及校对Laurie O'Connell、Nancy Riddiough和Emily Hsuan。

与美国RGA公司（the Reinsurance Group of America, Inc.）Global Software部门的工作人员合作同样让人感到愉快。和我一起工作的是一群颇具天赋的IT专业人员，所以我能不断地从他们身上学到东西。我尤其要感谢我的主管、也是行政总监Kam Chan先生，得益于他的帮助，我在编程之外学会了如何设计大型复杂的系统。另外我要感谢副经理Rick Nolle先生，是他花时间帮我在RGA找到了合适位置。最后，我要感谢Jym Barnes主任，我们经常在一起讨论最新的技术。

我还要谢谢Neil J. Salkind博士，是他提出了本书的构想，并帮我实现了这个计划。我也要感谢我的朋友Lisa Oliver先生帮我审阅了本书很多内容，提出了不少建议和想法。同样我要感谢我的朋友Jeffrey Noedel先生和我一起讨论了很多Bot技术在现实世界的实际应用，以及感谢在圣路易斯的华盛顿大学的Bill Darte先生，我在他指导下做出的一些研究成果都编进了本书。

译 者 序

目前，Internet已经成为人们工作、生活和娱乐的一部分，它就像浩瀚的知识海洋，任由人们自由地遨游。当你在网上冲浪时，你是否知道还有一类特殊的网络用户也在Internet上默默地工作着，它们就是网络机器人。这些机器人按照设计者预定的方式，在网络中穿梭，同时收集关心的信息。热门的搜索引擎站点就是很好的实例，很多搜索引擎的后台工作方式就是使用若干个网络机器人自动收集各站点信息，然后进行分类和整理，将整理结果提供给用户，以方便用户查找他们感兴趣的内容。由于网络机器人的实用性，引起了很多程序员，特别是Web程序员的兴趣。

为此，我们翻译了这本介绍如何实现具有Web访问能力的网络机器人的书，目的是希望帮助读者了解网络机器人，并能利用本书介绍的技术，设计和实现自己的网络机器人。本书深入浅出、循序渐进地阐述了网络机器人程序的实现技术。它从Internet编程的基本原理出发，首先介绍了Java套接字编程技术，然后详细地分析了网络机器人如何解析HTTP协议和HTML语言，以及从中提取有用信息的过程，并通过实际例程，深入剖析了访问Web网站的网络机器人程序Spider（网络蜘蛛程序）、Bot（网络机器人程序）和Aggregator（网络信息搜集器）的实现，最后本书提醒程序员要负责任地使用网络机器人程序，同时向读者描述了网络机器人程序未来的发展趋势。本书提供了大量的有效源代码，并对这些代码进行了详尽的分析。另外，本书的选配光盘中还提供了书中所有源程序和完整的Bot程序包。利用这些技术和资源，读者可以方便地设计并实现网络蜘蛛或网络信息搜索器等机器人程序。

本书由童兆丰、李纯、刘润杰和张文耀四人合译。同时，本书的翻译出版工作还得到许多人士的大力支持，在此表示感谢。

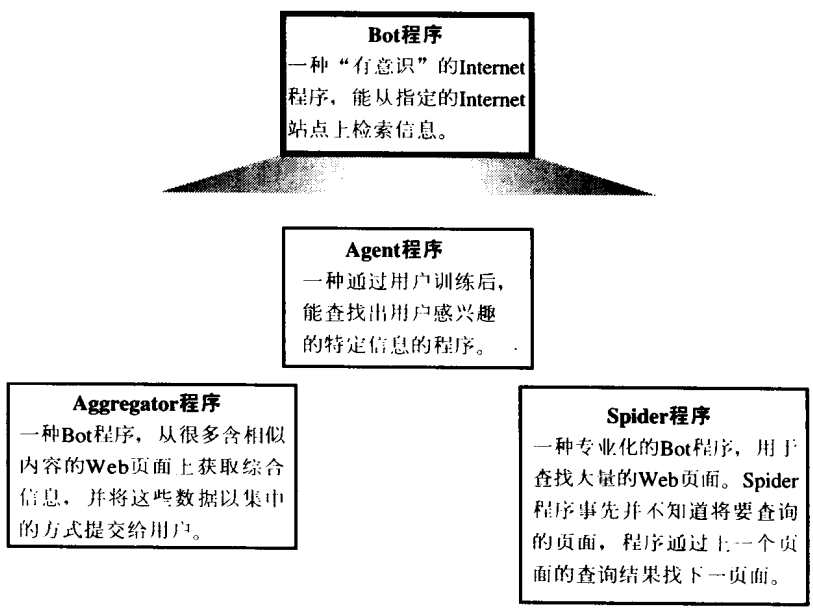
最后，希望读者阅读本书后能有所收益。由于IT技术发展的日新月异，新的技术术语层出不穷，且译者水平有限以及时间仓促，因此，本书许多译名仅供读者参考。译者自知专业水平及驾驭中英文的能力有限，译文中定有欠妥或错误之处，恳请读者给以指正。

简　介

通过Internet，我们可以得到海量的信息，包括今天的新闻，期待的邮包目前所在的位置，昨晚比赛的结果，或者你的公司当前的股票市值。打开你熟悉的浏览器，所有这些信息只需轻轻点击鼠标就能获得。几乎所有当前的信息都能在网上找到，你要做的就是如何去发现它。

Internet上的大多数信息都由网络用户生成，又被其他网络用户使用，因此，Web页面通常设计成吸引用户访问的结构。但这是Web的惟一用途吗？网络用户是一个Web站点惟一的服务对象吗？

实际上，一类全新的Web用户正在逐渐壮大，这些用户是计算机程序，它们和使用浏览器的人类一样具备访问Web的能力。这类程序有很多名称，不同的名称反映了它们各自所能完成的不同的特殊功能。Spider（网络蜘蛛程序）、Bot（网络机器人程序）、Aggregator（网络信息搜集器）、Agent（网络代理程序）、Intelligent Agent（智能网络代理程序）都是普遍使用的术语，代表一些具备Web智能的计算机程序。阅读本书以后，我们可以学会如何创建这些Internet程序。本书还分析了它们之间的不同点，以及说明了每种程序的优点。图I.1描述了这些程序的层次结构。



图I.1　Bot、Spider、Aggregator和Agent程序

什么是Bot程序

Bot是Internet智能程序最简单的形式，它的名字来源于机器人这一术语。机器人是指能承担反复性工作的设备。基于软件的机器人，或网络蠕虫，其工作方式也和在流水线上工作、

不断焊接同样装置的机器人一样，它们被程序控制反复执行同样的任务。

任何一种能访问Internet，并取回数据的程序都可以被称为Bot程序，其中Spider、Agent、Aggregator和Intelligent Agent程序都是专项Bot程序。在某些方面，Bot程序类似于具有宏功能的计算机程序，如Microsoft Word，给用户提供了录制功能。这些宏允许用户重放一系列的命令，以完成重复性的任务。一个Bot程序实质上就是一个宏，设计用来寻找一个或多个Web页面，并从中摘录相关信息。

Internet上使用了很多Bot程序实例，例如，搜索引擎经常就是利用Bot程序检查它们的站点列表，并清除那些已不存在的站点；金融软件会发布检索的收支差额和股票报价，桌面应用程序会检查Hotmail或Yahoo的邮箱账号，用户有邮件时，它会显示一个图标。

在2001年2月发行的Windows开发者杂志（Windows Developer's Journal）上，我发表了一个很简单的程序库，用来创建Bot程序。之后我收到了很多读者的来信，告诉我他们利用我的基础库开发的有趣的应用。我的眼睛捕捉到这样一个应用：一位父亲想为他儿子的生日买一架非常受欢迎、并且是刚发布的视频游戏操纵台。作为宣传的一部分，厂家在Internet站点上设了几台这种游戏操纵台，作为一个拍卖条目，第一个看见这条标注的人将得到这个游戏操纵台。这位父亲在我出版的代码的基础上，编写了一个Bot程序，程序会在拍卖站点上轮询，等待新的操纵台出现。当Bot程序看到待售的新游戏操纵台出现的瞬间，它立刻触发一系列动作，保证了竞标成功。这个计划顺利实施，他儿子也得到了游戏操纵台。这位父亲非常高兴，他把关于我的Bot程序独一无二的应用写信告诉了我，如果我在马里兰州的话，他甚至会邀请我来一局游戏。

这个故事引出了一个应用Bot程序的重要话题，使用Bot程序是否合法？你可以发现一些站点采用了特殊的措施来限制Bot程序的用途。例如，有些股票的开盘站点如果检测到Bot程序时，它们将不显示任何数据。其他一些站点可能会在它们的服务条款或许可协议中明令禁止使用Bot程序。为了防止Bot程序员不理睬这些服务条款，有的站点甚至两种方法都使用了。但是，不允许使用Bot程序的站点还是少数。第12章“负责任地使用Bot程序”将从道德和法律角度出发，详细讨论关于Bot程序的用法。

警告：作为一个Spider、Bot或Aggregator程序的作者，你必须确保你的Bot程序搜索获得的数据是合法的，如果你执行程序的时候还有疑问，应该询问一下站点的主人或律师。

什么是Spider程序

网络蜘蛛程序的名字起源于昆虫界的蜘蛛：蛛蛛吐完丝后，就在又大又复杂的蜘蛛网上旅行，从这根丝爬到那根丝。和昆虫界的蜘蛛类似，一个用计算机处理的蜘蛛程序从互连网的这端移动到那端。

Spider程序是一种专业化的机器人程序，专门设计用来做基于内容的站点查找。Spider程序从一个或几个简单的Web页面上开始执行，然后这些页面被扫描，索引到其他页面，Spider程序再访问这些Web页面，无休止地继续重复上面的过程，直到没有新页面的索引出现了，这个程序才停止。这个过程不会无休止地进行，这是因为Spider程序被强迫查找指定站点，没有这种强迫，Spider程序不太可能完成任务。如果一个Spider程序没有受制于一个站点的话，除非它访问了互联网上的每一个站点，否则它不会停止执行。

Internet搜索引擎代表了Spider程序的最早应用。搜索引擎使用户能通过输入几个关键字来指定Web站点的搜索。为了完成站点的查找，搜索引擎必须从一个站点查到另一个站点，以匹配这些关键字。最初搜索引擎可以在用户等待结果的时候，它在站点间来回穿梭，但是很快这种做法就变得不切实际，很简单，因为有太多的Web站点需要访问。由于这个原因，大型的数据库被保存下来，用于做Web站点与关键字之间的交叉对照。搜索引擎公司，如Google公司，它们就利用Spider程序来遍历Web站点，以创建并维持这些大型的数据库。

Spider程序的另一个常见用法是Web站点的映射。一个Spider程序可以扫描Web站点的主页，通过这些页面，它能扫描整个站点，得到这个站点使用的所有文件的清单。有个Spider程序遍历你自己的站点很有用，因为这种探察能揭示与站点结构相关的信息，例如，Spider能扫描出中断链接，甚至查出拼写错误。

什么是代理或智能代理程序

Merriam-Webster的学院字典是这么定义代理的：一个人代表另一个人行动或从事商业活动。例如，一个文学代理就是代表作者处理很多与出版商交易事务的人。简单地说，一个用计算机处理的代理程序能访问Web站点，并能为特殊的用户从事交易活动，例如代理程序能出售一块投资位置，以应对其他事件。代理程序的常见用法还包括“计算机研究助理”。这种代理程序知道主人对哪种类型的事情感兴趣，当线路上有这类事情出现时，代理程序会为主人抓住这些事情。

代理程序有巨大的潜力，但是它们还没有得到大规模的应用，因为要编写真正功能强大而且通用的代理程序，需要达到人工智能级别的编程水平，现在还达不到这个程度。

在智能代理程序和常规代理程序之间有一个区别，非智能代理程序就是一个Bot程序，它的程序中事先编制的信息对其主控用户来说是惟一的。大部分新闻剪辑代理就是非智能代理，它们的工作方式是这样的：它们的主控用户编制好一系列关键字和新闻信息源，程序只需扫描这些内容就可以了。

智能代理程序是使用了AI（Artificial Intelligence，人工智能）技术编写的Bot程序，它能更容易地适应主控用户的需求。如果这个代理程序用来剪辑文章，那么主控用户可以通过让它知道哪些文章有用，哪些没用，来训练代理程序。使用AI模式识别算法，代理程序会尝试去识别以后遇到的与主控用户的要求近似的文章。

说明： 本书明确是关于Spider、Bot和Aggregator程序的内容，它们都是直接处理Web页面的Bot程序。智能代理程序能基于用户的训练做选择，这不仅是一个Web编程的主题，更是一个AI的主题。由于本书主要安排的是直接与Web浏览相关联的Bot类型，将不包含智能代理的内容。

什么是Aggregator程序

聚集是用几个小对象创建一个组合对象的过程。用计算机处理的聚集过程做同样的事情。Internet用户经常有几个类似的账号，例如，一般用户都有几个银行账号，所有这些账号可能会在不同的环境中使用，并且使用各自的用户ID和口令保护每一个账号。

Aggregator程序允许用户用一条简明的语句浏览所有这些信息。Aggregator程序就是一种Bot程序，设计用来登录几个用户账号，并检索出相类似的信息。通常Bot程序和Aggregator程序的区别可以理解成下面这个实例中的情况：如果一个程序被设计用来取得某一特定的银

行账号，那么它可以看成是Bot程序；而同样的程序如果功能延伸到取得几个银行账号的信息，那这个程序可以看成是Aggregator程序。

今天已经有很多Aggregator程序实例，金融软件（例如Intuit的Quicken和Microsoft Money）能够集中查看用户的金融和信用账目信息。某些电子邮件扫描软件能告诉你几个在线邮箱里是否有消息在等你收阅。

说明： Yodlee（http://www.yodlee.com）是一个专门研究聚集的Web站点。使用Yodlee的用户可以简单地浏览到他们所有的账户信息。Yodlee独一无二的地方在于它能聚集如此之多的不同账户类型。

Java编程语言

由于Java语言能比较理想地实现Internet编程，所以本书着重选择Java程序设计语言作为计算机语言讲解实例。许多其他语言需要由第三方扩展的编程技术，对于Java编程语言都是本身的一部分。Java提供了一整套供Internet程序员使用的类。

Java不是实现本书所能用的惟一语言，因为本书所展现的网络机器人技术是通用的技术，它超越了Java编程语言的范围，这里揭示的技术也能应用于C++、Visual Basic、Delphi，或其他面向对象的程序设计语言。另外，一些程序设计语言具备使用Java类的功能。本书提供的Bot包可以很容易地应用于这类语言环境中。

本书在内容安排上假设你已经对Java程序设计语言比较熟悉，但不要求有Java语言专家级的知识，书中的所有内容不会超出Java基本编程的范围。例如，你不需要懂套接字或HTTP，不过你需要熟练掌握如何在你的计算机平台上编译和执行Java程序。要达到这种程度，推荐一本较好的Java参考书——《Java 2 Complete》（Sybex公司1999出版），可以作为本书理想的配套书籍。

本书是采用Sun公司的JDK 1.3（J2SE版本）编写的，每一个例程以及核心程序包都包含基于Windows和Unix平台创建的脚本文件，不过JDK不是编译这些文件的惟一方法。很多公司都开发出了叫IDE（Integrated Development Environment，集成开发环境）的产品，它提供了能创建和执行Java代码的图形化开发环境。

读者使用本书时，不需要IDE环境，但是本书提供了所有在WebGain的VisualCafé中使用的必需的工程文件。源代码和任何支持JDK 1.3的IDE兼容。一旦建立起工程文件，其他的IDE（如Forte、JBuilder和CodeWarrior）也都能支持。Microsoft Visual J++只支持Java的1.1版，所以在其平台上运行本书的代码会遇到很多问题。不太清楚写完这本书以后，Microsoft公司是否打算继续支持扩展的J++。

本书特色

作为一个读者，我一直觉得对我最有用的书是那些既讲授新技术、又提供全套程序库来演示这项新技术的书。这种方式让我带着工具箱迅速掌握这项正在讨论的技术，然后，随着我对新技术应用的深入，逐渐领会了书中试图传授给读者的潜在技术。本书的结构就是如此，它为读者提供了两个关键内容：

- 一个可重用的Bot、Spider和Aggregator程序包（以后统称为Bot包），它可以应用在任何Java或JSP程序设计中。这个程序包附在选配的光盘中。

• 每一章都含关于如何使用Bot包的实例，这些例程也包含在选配的光盘中。

选配光盘中还包含Bot包的全部源代码，除此之外，每一章都提供了Bot包工作原理的深入讲解。

选配光盘中的内容

选配光盘包含本书介绍的所有例程的Java源代码，也包含Bot包的二进制代码和全部的源代码，大多数例程都使用了该包。另外，选配光盘还包含Bot包中每个类完整的JavaDoc文档。

除了Java代码和Bot包外，还提供了下列两个第三方的应用软件。请阅读Licenses目录下的文件，这些文件解释了使用这些软件的条件：

• 来自Sun Microsystems公司的JDK 1.3版本

• 来自Jakarta Project的Tomcat 4.0版本

JDK 1.3用于编译和测试本书包含的所有例程代码。在本书出版时，JDK 1.4的正式版本还没有，所有的代码也用最新的JDK 1.4 beta版测试过。

Tomcat 4.0是Jakarta Project提供的最新JSP外壳。Tomcat系统用于执行Java服务器网页（Java Server Pages，简称JSP）。本书包含的一些例程利用了JSP技术。

目　录

第1章 Java套接字编程技术

- 套接字家族一览
- 学习网络编程
- Java数据流和过滤器编程
- 理解客户端套接字
- 服务器端套接字揭密

Internet构筑在很多相关的协议基础之上，而更复杂的协议又建立在系统层协议之上。协议是用于两个或多个系统之间协作通信的方式。很多用户在想到Internet时，他们会联想到Web，事实上Web就是一种建立在HTTP（Hypertext Transfer Protocol，超文本传输协议）之上的协议，而HTTP又是建立在TCP/IP（Transmission Control Protocol/Internet Protocol，传输控制协议和互连网协议）之上的协议，它同时也是一种套接字协议。

本书的大量篇幅会安排关于如何处理Web及其相关协议的内容，但在此之前我们首先从TCP/IP套接字编程开始，讨论一下HTTP协议。

在本章中，套接字和TCP/IP编程这两条术语会频繁交替出现，在现实世界中也是如此。从技术角度来说，基于套接字的编程允许使用的协议不止TCP/IP协议，由于近几年来TCP/IP系统的繁荣发展，TCP/IP协议成为套接字编程的惟一通用的协议。

套接字家族

Spider、Bot和Aggregator都是浏览Internet的程序。如果你希望学会如何编写这类程序，实现本书的首要目的，那你必须先学会如何浏览Internet。这并非像普通用户那样浏览，而是理解计算机应用程序实现浏览功能的方法，像Internet Explorer，或其他浏览器程序。

浏览器通过请求使用HTTP协议的文档来实现浏览功能，利用这种协议可以使浏览器之间的通信更方便。在本章中，HTTP只是作为与套接字相关联的内容来讲，关于该协议更多的内容会在第2章“分析超文本传输协议”中做详细描述。本章介绍如何处理套接字这种构筑HTTP的基础协议。

隐蔽的套接字

当套接字用于连接TCP/IP网络时，它即成为了Internet的基础。但是由于套接字的功能就像房屋的地基一样，比较隐蔽，它们经常处于大部分Internet程序开发人员处理的网络最低层。实际上，很多开发Inetrnet应用的程序员对套接字了解甚少，这是因为程序员处理的经常是更高级的组件，这些组件在程序员和具体套接字命令之间起到中间件的作用。正因为如此，程序员不会考虑正在使用的是什么协议，或者是套接字是如何实现这种协议的。另外，编程人员也不会意识到存在于套接字下面的网络层——与硬件更相关的路由器、交换机和集线器

层面。

套接字不关心数据的格式，它和低层的TCP/IP协议都只需确保数据到达正确的目的地。套接字的工作很像邮政服务，它们所做的就是将信息分递到世界各地的计算机系统。更高层的协议，如HTTP协议，就会给正在传送的数据赋予一定意义。如果系统接收HTTP类型的信息，它就知道哪些信息支持HTTP协议，而不支持其他协议，如支持电子邮件信息的SMTP（Simple Mail Transfer Protocol，简单邮件传送协议）等。

就像网络对于中间介质隐含套接字命令的方式一样，本书所带的Bot包（参见选配光盘）对读者隐含了一些内容，这个程序包允许程序员创建高级的Bot应用程序，而无需知道什么是套接字。但这章的内容覆盖了关于如何在最低层的“套接字层”进行实际通信所涉及的几个方面内容。这些细节包括HTTP请求如何利用套接字传送，以及服务器如何响应。如果你只是对创建Bot程序感兴趣，而不关心Internet协议是如何构造的，那么你完全可以跳过这一章。

TCP/IP网络

在使用套接字时，经常涉及TCP/IP网络，建立套接字的目的就是它能抽象TCP/IP和其他低层网络协议的区别，IPX（Internetwork Packet Exchange，Internet包交换）协议就是一个实例。IPX协议是由Novell开发，用于构建第一个局域网（local area network，简称LAN）的协议。使用套接字后，创建的程序既能利用TCP/IP通信，也能利用IPX通信。套接字协议将IPX和TCP/IP的不同与程序隔离开来，这样就使一个简单的程序能在不同的协议上运行。

说明：虽然其他协议也能与套接字一起使用，但它们的Internet浏览功能非常有限，因此本书不讨论这些协议。

当TCP/IP协议第一次引进时，它对当时已经存在的网络结构是一种根本的改变，因为它没有遵从原来已有的典型分层模式。不像其他的网络结构，例如SNA（Systems Network Architecture，系统网络体系结构），TCP/IP协议使客户端和服务器在机器层没有区别，也就是说只有一种简单的计算机，它的功能可以作为客户端，或者是服务器，也可以二者兼有。网络上的每一台计算机都配了一个地址，并且地址没有大小差别。正因为如此，一台运行于政府部门研究机构的超级计算机有一个IP地址，而一台设在儿童睡房的个人电脑同样有一个IP地址，在这二者之间没有区别。

这种类型的网络叫对等网络。TCP/IP网络上的所有计算机都被看成平等的，并且在这个网络上的机器既有客户端功能、又有服务器功能的现象很常见。在对等网络中，一个客户端指的是首先发送网络数据包的程序，而服务器是指收到第一个数据包的程序。一个数据包是一次网络传输，很多数据包在客户端和服务器之间的传输就形成了请求和响应。

网络编程

现在我们来学习如何进行实际的套接字编程，以及处理套接字协议，它有一种普遍的叫法是网络编程。在学习与通信相关的套接字命令之前，需要先分析一下协议。事先知道你传输的是什么内容，这样再学习如何传输它就更清楚了。

这个过程从如何确定一个服务器所使用的协议类型开始，它是通过使用众所周知的网络端口和服务实现的。

众所周知网络端口和服务

网络上的每一台计算机都有很多套接字辅助计算机程序生效。这些套接字叫端口，它们都编了号码。端口号很重要，其中一个非常重要的端口是80端口，这个是HTTP套接字使用的，将贯穿本书的始终，几乎书中的每一个例程都涉及Web访问，都会用到80端口。在任一台计算机上，服务器程序必须指定端口号用于“倾听”每个连接，而客户端程序必须指定端口号用于请求连接。

你可能对这些端口是否能共享会有疑惑。例如，当一个Web用户与Web服务器的80端口建立了连接后，另一个用户能否同时与其80端口也建立连接呢？答案是可以，多个客户端能够连接到同一台服务器端口。但是，每一时刻只有一个程序能侦听同样的服务器端口。这些端口可以看成是电视台，很多电视机（客户端）都可以调到一个特定的频道（服务器）接收信号，但几个台（服务器群）不可能在同一个频道上广播。

表1.1列出了众所周知端口的分配和它们相应的RFC文档编号，这些文档描述了其对应协议的规则。我们会在本章稍后的内容更详细地分析这些RFC文档。

表1.1 众所周知端口分配及其相应的RFC文档编号

端口	通用名称	RFC编号	作用
7	Echo	862	回送数据，多用于测试
9	Discard	863	丢弃发送过来的所有数据
13	Daytime	867	取日期和时间
17	Quotd	865	日期的引用
19	Chargen	864	产生字符，多用于测试
20	ftp-data	959	传输文件，FTP支持文件传输协议
21	ftp	959	传输文件和命令
23	telnet	854	登录远程系统
25	SMTP	821	传输Internet邮件，支持简单邮件传输协议
37	Time	868	测定计算机上的系统时间
43	whois	954	测定远程系统上的用户名
70	gopher	1436	查寻文档，但基本上已经被HTTP取代
79	finger	1288	测定在其他系统上的用户信息
80	http	1945	传输文档，是构成Web的基础
110	pop3	1939	访问存储在服务器上的信息，支持POP3协议（Post Office Protocol, version 3）
443	https	n/a	允许安全的HTTP通信，支持加密套接字层（Secure Sockets Layer，简称SSL）上的超文本传输协议

什么是IP地址

TCP/IP协议实际上是两个协议的组合：传输控制协议（TCP）和Internet协议（IP）。TCP/IP协议的IP组件负责将数据包从一个节点传送到另一个节点，而TCP负责校验从客户端

传送到服务器端的数据正确性。

一个IP地址看上去像一个含四个数的串，数字之间用小数点分隔开。因为实际地址会和协议的IP部分一起传输，所以这个地址叫IP地址。例如，我们自己的站点IP地址是216.122.248.53。四个数字的每一个都是一个字节，因此它的数值范围为0到255，整个IP地址是一个4字节，即32位的数字，这个长度和Java基本数据类型中的整数类型一样。

为什么用四个分隔开的数字来代表一个IP地址呢？如果它仅仅是一个无符号的32位整数，为什么不能作为一个数字识别标志表示成IP地址呢？事实上，你可以把IP地址216.122.248.53表示成3631937589，如果你的浏览器指向http://216.122.248.53，它会把你带到和指向http://3631937589同样的网站。

如果你不熟悉字节次序的数字表示方法，那么把216.122.248.53变形成3631937589后，会有些令人费解。利用任何科学型计算器，或者就用Windows自带的科学型计算器，很容易完成这个转换运算。要完成这个转换，必须将地址216.122.248.53的每一个字节转换成对应的十六进制数。你可以轻易地将Windows的计算器转换成十进制模式，输入数字，然后转换成十六进制模式。做完这些后，反映出的结果如以下显示：

十进制	十六进制
216	D8
122	7A
248	F8
53	35

现在每一个字节都是十六进制，要产生一个集成了4个字节的十六进制数，只需将每个字节按次序连续排列就可以了，如下所示：

D8 7A F8 35 或 D87AF835

现在得到了一个和IP地址相等的数值，惟一的问题是该数字是十六进制的，其实没什么问题，科学型计算器很容易将十六进制转换成十进制。这样做了以后，得到了数字3,631,937,589，它和URL: http://3631937589中使用的数字一样。

为什么我们需要IP地址的两种形式呢？216.122.248.53这种形式比3631937589强在什么地方呢？原因主要在于前者更容易记住。虽然这两个数字都挺难记的，但Internet设计者认为按字节段分开的符号（216.122.248.53）比冗长的数字符号（3631937589）更好记。在实际环境中，终端用户通常两种形式都看不到，因为IP地址几乎都和主机名相关联。

什么是主机名

由于216.122.248.53，或3631937589这样的地址对于一般计算机用户来说太难记了，因此使用了主机名。例如，我的主机名是www.heat-on.com，它被设定指向216.122.248.53，人们记住www.heat-on.com这个名称就比记住216.122.248.53要容易很多。

主机名不应该与URL（Uniform Resource Locator，统一资源定位器）相混淆。一个主机名只是URL上的一个组成部分。例如，我的站点上某一个页面的URL是http://www.jeffheaton.com/java/advanced/，而主机名www.jeffheaton.com只是这个URL中的一部分。它指定服务器传送被申请的文件。主机名只指定某一IP地址属于哪台服务器，URL则指