

251
7P312JA
x 69
编程经典译丛

Java 网络编程技术内幕

第 二 版

Java Network Programming (2nd Edition)

Merlin Hughes Michael Shoffner
Derek Hamner with Umesh Bellur 著

刘先勇 高志勇 乐 莉 黄风雷 译

国防工业出版社

·北京·

著作权合同登记 图字:军-2001-010号

图书在版编目(CIP)数据

Java 网络编程技术内幕/(美)休斯(Hughes, M.)著;刘先勇等译. —北京:国防工业出版社, 2002.3

(编程经典译丛)

书名原文:Java Network Programming

ISBN 7-118-02780-4

I .J... II .①休...②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 004508 号

©1999 by Manning Publications Co. All rights reserved.

Published by arrangement with Manning Publications Co., USA.

本书中文简体版由 Manning Publications Co. 授予国防工业出版社(独家)出版。版权所有,侵权必究。

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 40 1234 千字

2002 年 3 月第 1 版 2002 年 3 月北京第 1 次印刷

印数:1—3000 册 定价:72.00 元

(本书如有印装错误,我社负责调换)

第一版前言

带着我们的承诺——为客户提供先进的商用互联网技术解决方案,我们建立了我们的公司——Prominence Dot Com。当我们首次开张时,关于互联网的热门新技术是 World Wide Web。我们的第一个商业定单是面向应用程序的 Web 站点。

1995 年夏,在 Java 的 alpha-2 阶段,我们开始用 Java 开发 Web 应用程序。那时,各种 CGI 的应用程序显现了基于 Web 的产品的前沿技术。Java 独一无二的、功能强大的特性立即显现了出来,它使得能够运行在任何 TCP/IP 网络上的“真正的”分布式应用程序成为可能。

当然,在 Internet 术语中,所有这些历史。差不多一年多以后,Java 就已经变得比激发其名字灵感的咖啡更加热门了。事实上,在本书写作期间,咖啡市场本身波澜不惊,但 Java 持续上涨的流行趋势却是汹涌澎湃。

Java 出现于恰当的时机和恰当的地点,带着对许多疑难问题的答案,它在全世界开发者的工具箱中赢得了一席之地。从开发者的立场来说,Java 的跨平台特性、安全特性、面向对象特性以及以网络为中心的特性使其非常有用。简洁的语法使其轻松自然、易于使用。最重要的是,其与生俱来的浏览器支持特性和作为唯一的跨平台高级语言的地位,使它成为严肃的面向 Web 和 Internet 开发的必需品——正是这种开发,需要使用本书讨论的安全联网代码集和技巧。

《Java 网络编程技术内幕》产生于一组专业课题;我们开发这些课题是为了响应全面引用 Java 语言先进特性的广泛需要。因此,本书基本覆盖了每一个细节,从 Java 联网编程的基本知识到实现高级的安全联网应用程序的深入技术。本书同时是一本关于联网编程和密码学的完整教程、一本 Java 联网编程 API 的参考书以及包含在所附 CD-ROM 中的产品代码的集合。

我们真诚地希望并且相信,本书中的资料对您将是非常有用的,就如它们对我们一样。请给《Java 网络编程技术内幕》一个好的家园,让书页卷角,以使它知道它正被您所爱、所深深的关注着。

第二版前言

好了,现在我们见到的是第二版。为什么我们要再次出版本书呢?

如以前一样,我们编写《Java 网络编程技术内幕》的主要目的,是为大家提供用 Java 开发联网应用程序的实用技巧和代码。

由于 Java 平台最近的修改,Java 在联网和分布式计算方面的能力得以显著提高。现在的 API 为开发重要的应用程序提供了新的选项。本书的重点集中在 Java 提供的基本的联网编程技术;流和基于 IP 的协议,例如 TCP、UDP 以及组播。在第二版中,我们增加了与这些技术相关的新 API 的特征,同时改善和扩充了第一版中的例子,进一步覆盖了这些变化以及更多的内容。

在其它方面,Java 平台的修改引入了新的联网编程能力,例如 RMI、CORBA 以及 servlet。为了在本书中提供良好的联网编程基础,第二版增加了有关这些重要主题的内容。当然,印刷页面的有限空间,限制了我们对这些联网 API 的深入讨论。因此,我们尽力充分地介绍这些主题,使你可以明智地选择自己应该使用的技术;倘若有某一主题使你入迷,其它的专著将提供比我们所希望的更好的服务。

最后,随着 Java 2 平台的出现,一个完整的影响深远的加密技术 API 已经被增加到 Java 的指令表中。与其把第一版的密码例子硬塞入新的 API,不如从第二版中完全地删除密码——我们选择了在一本单独的书《Applied Java Cryptography》中讨论它们,该书将在 Java API 稳定之后推出。

简而言之,有许多新资料需要推出,一些旧资料需要修正,而其它一些随机的事情又碰巧吸引了我们的注意力。此外,出版商仿佛用枪指着我们的头,所以我们别无选择。

祝愉快!

致 谢

把这本书拼凑在一起给我们带来了巨大的痛苦和折磨。如果没有人们的慷慨帮助,可能我们现在都已经累死了。感谢 Marjan Bace——我们的发行人以及他的助手,他们给我们许多有价值的建议并且一再容忍我们错过最后期限,感谢 Len Dorfman 使我们先于他人开始这项工作。感谢 Kevin Jeffay 审阅了关于联网编程的附录,感谢 Grant Gainey 和 Dave Zimmerman 启发我们增加了独创的 RMI。感谢 Dinesh Manocha,他在第一版写作期间显示了无限的耐心,同时也感谢 Jeffay, Don Smith, Bert Dempsey 和 Donal O' Mahony,几年间他们在联网编程方面给了我们许多有价值的建议。

我们所有人都要再一次感谢我们的朋友和家人,因为即使我们由于工作而暂时抛弃了他们,但是他们永远不离开我们。特别感谢 Dumsa 博士给予我们的最需要的指导和检查。

Merlin 尤其想感谢 Simon Rooney, David Graham 以及 TCD 工作人员给予我们的支持,并且也要特别感谢巴尔的摩的伙伴们对他的帮助。同时 Michael 想特别感谢北卡罗莱纳州 Sunsight 大学的全体同事们以及 UNC Geology 的伙伴们,感谢他们所有人给予的帮助。

第二版带来了又一轮的痛苦和折磨。但是我们现在已经习惯这样的痛苦和折磨,我们变得更加坚强了。除了所有第一版的伙伴们,我们想感谢 Catherine O'Sullivan, Jill Greeson, Marq Singer 和 Paul Jones 给予我们的耐心和帮助,感谢 Wang, Roy Tal, Andrew Chen, Claude Duguay, Michael Andreano, Michael Brundage, Scott Zimmerman 和 Victor A. Brennan 给予我们巨大的令人欣赏的反馈。

同样也感谢 Tony 和 Sherry Roberts, Elizabeth Martin, Aaron 和 Heather Lyon, Margaret Marynowski 和 Leslie Haimen,感谢他们这些年来一直为本书付出的辛勤劳动,感谢 Kevin Hall 的天才指导。

没有你们,就不会有本书的出版。

最诚挚的敬礼!

读者指南

亲爱的读者：

从初学者到经验丰富的开发者,《Java 网络编程技术内幕》都有您需要的内容。除了开发实用的联网类以外,本书还详细论述了像自定义流和远程方法调用这样的先进特性。为方便起见,书中开发的代码库可以从其伴随网站上在线获得,该网址是:<http://nitric.com/jnp/>。从该网址还可以获得将来更新的代码。

- 初学者可以首先阅读本书的“预备知识”部分,它包含了联网编程的概述以及 Java 安全模型、异常和多线程的重要部分。本书的附录详细地描述了关于联网编程的背景信息。书中的第二部分和第三部分完整地介绍了 `java.io` 和 `java.net` 软件包,它们是 Java 中网络编程的基础,也是本书其余部分的基础。

- 有经验的 Java 编程人员可以直接跳到最适合于他们需要的地方。熟悉原始的 Java API 的编程人员,可以浏览本书第二部分和第三部分中介绍的最新语言版本的新特征,例如组播和字符流,相应地有很多典型例子说明这些内容。然后,可以学习本书的第四部分和第五部分,这两部分包含了消息传递、RMI、CORBA 以及 `Servlet`,它们是本书中所包含的最新主题,并且是使用 Java 进行网络开发中功能最强大的一部分内容。

- 有经验的 C 和 C++ 编程人员将会发现,使用 Java 进行网络开发更清楚并且更容易得到回报。例如,Java 的独特处理机制有助于强有力地处理那些在 I/O 和联网操作中出现的常见问题,而线程提供了很容易就实现功能强大的服务器的方法。假定你熟悉 Java,在 Java 中联网应该自然地成为 C/C++ 开发者所实现。

第二版中的改变

本书的第二版与第一版相比发生了很大变化。这些变化可以归入三个方面:提供了额外的对新的 Java API 特性的论述、由于 Java API 的变化而导致的所有例子的更新,以及更多的做了修改的联网应用程序的例子。

语言 I/O 工具的讨论增加了关于字符流的全部细节以及对象流的增强的论述。除了包括 `finger`、`nslookup` 以及全功能的 Web 服务器这些完整的联网应用程序的新例子以外,本书的联网编程部分也覆盖了基类的变化并提供了关于组播工具的深入讨论。一个新的部分详细讨论了 RMI、CORBA 和 `Servlet` 等内容,它们是由于 Java 平台的更新而新增的性能。最后,密码部分已经删除了,因为一本完整的专著能够更好地论述关于新的 Java 密码 API 的内容。

使用 Java 进行网络编程

Java 编程语言对联网应用程序的编写非常理想。核心 API 带有一组标准类,它跨越所有使用 Java 虚拟机接口的平台,为联网协议提供了统一的访问接口。Java 语言覆盖了诸如硬件字节顺序这样的问题,这使得编程者不必再费心考虑跨平台互用性的一些传统问题。基本的网络访问由 `java.net` 软件包的类提供。`java.io` 软件包的类进一步补充了这些访问,该包提供了统一的基于流的信道接口。可以扩展这些类,以便为满足自定义的通信需要提供博大精深的高级功能。

应用程序和嵌入的 `Applet` 都可以从 Java 语言所提供的联网工具中受益。`Applet` 可与中心信息存储器通信,访问中心的信息存储,为环球网上分布的用户提供真正的实时协作。应用程序同样可以从上述的方

式中受益,并且超越了 Java 与传统编程语言相比所具有的常规优势。Java 的安全性和动态连接性允许开发可在运行时以新代码扩展的服务器。其例子既包括能够动态地、安全地执行客户端提供的代码的 Web 服务器,又包括能够以“searchlet”代码段这样的形式接受搜索的搜索引擎。

语言 API 内的其它软件包进一步屏蔽了底层的联网细节。这些扩充包括远程方法调用 API,提供了透明地访问驻留在远程虚拟机上的对象的方法。远程方法调用的参数和结果通过虚拟机之间的网络连接隐式传递。另一个 API 和工具组通过提供统一的 CORBA 接口而具有对 CORBA 的兼容性。其它的软件包提供了密码系统框架、联网的多媒体访问和电话产品接口以及商业框架。循此以往,Java 空间有希望成为真正的分布式应用的全球的多元空间。

内容和布局

在本书中,我们广泛地讨论了 `java.io`, `java.net`, `java.rmi` 以及 `javax.servlet` 软件包;以丰富的例子,展示了用 Java 开发联网应用程序所必需的所有技巧,包括线程和异常处理的序言以及某些支持 CORBA 的 API 的详细内容。与 API 的讨论一起,我们提供了详细的资料并讨论了许多联网应用程序开发的实际例子,包括一个基于流的高级通信库。下面有关联网的概述是提供给那些不太熟悉本题的读者的。

本书由六个部分组成:

第一部分是预备知识,包括对于联网的简介、Java 安全模型、线程以及异常。

第二部分详细讨论了 `java.io` 软件包,给出了 API 的详细内容、示例代码以及对基于流的扩充。此部分给出了 Java 所有主要的流类的全部细节;包括字节流、字符流和对象流。

第三部分讨论了 `java.net` 软件包,使用了大量的现实世界中的例子,包括多个 TCP 与 UDP 客户和服务器的组播示例;并且完整地讨论了 URL 类,包括自定义协议处理器的示例。

第四部分详细讨论了 Java 支持的三个专用的联网技术:远程方法调用(`java.rmi`)、`servlet`(`javax.servlet`)以及 CORBA(`org.omg.CORBA`);讨论中全程给出了广泛的示例。

第五部分讨论了基于消息的联网库的开发;该库极大地减轻了开发某些联网应用程序的难度,是许多其它可能使人感兴趣的基于流的联网库的基础。

第六部分由附录组成,包括一个全面深入的关于联网的概述以及一些有用的列表。

Java 平台正经历着不断的修改。随着错误的修正,API 得以净化并增加了新的特性。除了已由 JDK 及其相关产品的主要新版本引入的类、接口和方法以外,还增加了一个小标识符。更具体地讲,作者已尽其最大能力,使本书的内容符合下列版本之一:

- JDK1.2;Java 2 SDK,a.k.a. JDK 1.2
- JDK1.1;JDK 1.1.7B
- JDK1.0;JDK 1.0.2
- Java IDL;Java 2 SDK
- JSDK 2.1;JSDK 2.1 specification
- JSDK 2.0;JSDK 2.0
- JSDK 1.0;JSDK 1.0.1

下载代码

为省去你根据本书键入所有代码的麻烦以及购买一张不必要的 CD-ROM 的费用,伴随网站 <http://nitric.com/jnp/> 以各种方便的可下载格式提供了所有代码。

源代码也可以从出版商的网站 <http://www.manning.com/Hughes2> 下载。

译者序

作为唯一的跨平台高级语言,Java 的跨平台特性、安全特性、面向对象特性、以网络为中心的特性及其与生俱来的浏览器支持特性,已使它成为了面向 Web 和 Internet 开发的必不可少的工具。众所周知,Java 在电子商务、互联网、分布式仿真计算的解决方案中,目前已具有举足轻重的地位。

本书是一本 Java 语言教程,既适合 Java 的初学者、有经验的 Java 编程人员,也适合有经验的 C/C++ 编程人员。

本书产生于一组专业课件,为全面引用 Java 的先进特性,它基本覆盖了 Java 的全部内容;从 Java 编程的基本知识到实现高级的安全联网,都有可运行的广泛示例和精彩的深入讲述。

本书第一版出版后,赢得好评如潮。Knowledge 2000 公司的高级开发者 Scott Fauerbach 将该书作为其开发队伍的必读书籍;Atrieva 公司的软件开发经理 Claude Duguay 更认为该书是其所见过的 35 本 Java 书籍中最出色的一本。

本书的第二版推出以后,美国佛罗里达大学的 Victor A. Bennen 说,“该修订本使一本伟大的书更臻完美。”本书的主要译者是 Java 的爱好者和开发者,在译书过程中运行了书中的大部分示例程序段,对其精巧构思和实战性留下了深刻印象。在本书翻译前,译者全文学习过 Bruce Eckel 的《Thinking in Java》第二版英文版,感觉其示例离实用尚有距离;而《Java 网络编程技术内幕》第二版,除了生动的分析图示外,其例子大都可以直接用于解决实际问题。如果说,使用是最好的学习,那么译者以为,那正是本书的特点所在;几乎所有类型的开发者都可以从本书中找到可以模仿的实例。

本书翻译工作时间紧、工作量大,译者水平有限,难免有不少疏漏,敬请广大读者批评指正。衷心希望本书能够为读者学习和应用 Java 提供实际的、有价值的帮助。

本书由张庆明、战守义审稿,参加本书翻译工作的还有徐四平、袁长迎、郑珊、王荣兰、杨守义、严柏军、李健、付天光等同志。

刘先勇 高志勇 乐莉 黄风雷

目 录

第 1 章 网络技术入门 / 3

- 1.1 协议栈与层 / 3
- 1.2 因特网 / 5
- 1.3 TCP: 传输控制协议 / 7
- 1.4 UDP: 用户数据报协议 / 9
- 1.5 IP 服务家族 / 9
- 1.6 DNS / 11
- 1.7 防火墙和代理服务器 / 12
- 1.8 结论 / 13

第 2 章 Java 安全模型 / 14

- 2.1 可信及不可信代码 / 14
- 2.2 安全管理器 (SecurityManager) 及系统资源 / 15
- 2.3 网络 applet / 16
- 2.4 防火墙 / 18
- 2.5 安全 API / 19
- 2.6 结论 / 19

第 3 章 异常处理机制 / 21

- 3.1 异常的捕获 / 21

- 3.2 异常的主要类型 / 22

- 3.3 异常的产生 / 22

- 3.4 异常的处理 / 23

- 3.5 Throwable 类 / 25

- 3.6 自定义异常 / 26

- 3.7 总结 / 26

第 4 章 多线程综述 / 27

- 4.1 线程中变量的作用域 / 28

- 4.2 同步 / 28

- 4.3 Thread 类 / 32

- 4.4 Runnable 接口 / 34

- 4.5 ThreadGroup 类 / 35

- 4.6 Runnable 示例 / 36

- 4.7 Thread 子类示例 / 37

- 4.8 线程通知方法 / 38

- 4.9 生产者-消费者示例 / 39

- 4.10 高效多线程 / 41

- 4.11 总结 / 44

第 5 章 流简介 / 49

- 5.1 概览 / 49

- 5.2 字符流 / 50

- 5.3 OutputStream 类 / 50

- 5.4 OutputStream 示例 / 52

- 5.5 InputStream 类 / 54

- 5.6 InputStream 示例 / 56

- 5.7 Tee 类示例 / 57

- 5.8 各种基本流 / 59

- 5.9 总结 / 60

第 6 章 文件流 / 62

- 6.1 File 类 / 62

- 6.2 FileDescriptor 类 / 65

- 6.3 RandomAccessFile 类 / 65

- 6.4 FileOutputStream 类 / 68

- 6.5 FileInputStream 类 / 69

- 6.6 文件流示例 / 70

- 6.7 覆盖式 FileOutputStream / 70

- 6.8 可定位的 FileOutputStream / 72

- 6.9 具有标记/重置功能的

FileInputStream / 74

- 6.10 总结 / 76

第 7 章 流过滤器 / 77

- 7.1 高级通信 / 77

- 7.2 FilterOutputStream 类 / 79
- 7.3 FilterInputStream 类 / 80
- 7.4 标准流过滤器 / 80
- 7.5 使用流过滤器 / 82
- 7.6 创建文本流过滤器 / 84
- 7.7 总结 / 89
- 第 8 章 标准流过滤器 / 91**
 - 8.1 DataOutputStream 类 / 91
 - 8.2 DataInputStream 类 / 92
 - 8.3 BufferedOutputStream 类 / 94
 - 8.4 BufferedInputStream 类 / 95
 - 8.5 PrintStream 类 / 96
 - 8.6 SequenceInputStream 类 / 97
 - 8.7 LinenumberInputStream 类 / 98
 - 8.8 PushbackInputStream 类 / 99
 - 8.9 写入一个缓冲输入流 / 101
 - 8.10 建立一个 T 字型流过滤器 / 104
 - 8.11 总结 / 111
- 第 9 章 内存输入/输出流 / 112**
 - 9.1 ByteArrayOutputStream 类 / 112
 - 9.2 一个简单的 ByteArrayOutputStream 示例 / 113
 - 9.3 ByteArrayInputStream 类 / 114
 - 9.4 一个简单的 ByteArrayInputStream 示例 / 115
 - 9.5 PipedOutputStream 类 / 116
 - 9.6 PipedInputStream 类 / 117
 - 9.7 一个管道流的例子 / 118
 - 9.8 ByteArrayOutputStream 的自动重置 / 119
 - 9.9 本章内容摘要 / 121
- 第 10 章 字符流 / 122**
 - 10.1 概要 / 122
 - 10.2 字符编码 / 123
 - 10.3 **JDK1.1** Writer 类 / 126
 - 10.4 **JDK1.1** Reader 类 / 127
 - 10.5 **JDK1.1** OutputStreamWriter 类 / 128
 - 10.6 **JDK1.1** InputStreamReader 类 / 129
 - 10.7 一个编码转换的例子 / 130
 - 10.8 **JDK1.1** FileWriter 类 / 131
 - 10.9 **JDK1.1** FileReader 类 / 132
 - 10.10 文件流的应用 / 133
 - 10.11 总结 / 134
- 第 11 章 字符流过滤器 / 135**
 - 11.1 **JDK1.1** FilterWriter 类 / 135
 - 11.2 **JDK1.1** FilterReader 类 / 136
 - 11.3 **JDK1.1** BufferWriter 类 / 137
 - 11.4 **JDK1.1** BufferedReader 类 / 137
 - 11.5 **JDK1.1** LineNumberReader 类 / 138
 - 11.6 字符流过滤器的应用 / 139
 - 11.7 **JDK1.1** Printwriter 类 / 140
 - 11.8 **JDK1.1** PushbackReader 类 / 141
 - 11.9 建立一个文字型数据的 Reader / 141
 - 11.10 LineAtAtime Reader 类 / 147
 - 11.11 总结 / 150
- 第 12 章 基于内存的字符流 / 151**
 - 12.1 PipedWriter 类 / 151
 - 12.2 **JDK1.1** PipedReader 类 / 152
 - 12.3 **JDK1.1** CharArrayWriter 类 / 153
 - 12.4 **JDK1.1** CharArrayReader 类 / 154
 - 12.5 **JDK1.1** StringWriter 类 / 155
 - 12.6 **JDK1.1** StingReader 类 / 155
 - 12.7 另一种一次读一行的 Reader 类 / 156
 - 12.8 总结 / 157
- 第 13 章 对象流 / 158**
 - 13.1 简介 / 158
 - 13.2 **JDK1.1** ObjectOutputStream 类 / 159
 - 13.3 **JDK1.1** ObjectInputStream 类 / 162
 - 13.4 **JDK1.2** 对象流类异常 / 164
 - 13.5 **JDK1.2** 接口 Serializable / 164
 - 13.6 **JDK1.2** 接口 Externalizable / 165
 - 13.7 **JDK1.2** 接口 ObjectInputValidation / 166
 - 13.8 创建一个可序列化的类 / 166
 - 13.9 为对象流建立子类 / 173
 - 13.10 类的版本 / 178
 - 13.11 总结 / 180
 - 13.12 结论 / 180
- 第 14 章 客户端网络编程 / 183**
 - 14.1 InetAddress 类 / 183
 - 14.2 Socket 类 / 186
 - 14.3 建立一个 PostOutputStream 类 / 194

- 14.4 综述 / 198
- 第 15 章 TCP/IP 客户端编程示例 / 199
 - 15.1 Internet 协议 / 199
 - 15.2 finger 客户 / 200
 - 15.3 DNS 客户 / 206
 - 15.4 总结 / 228
- 第 16 章 服务器端网络编程 / 230
 - 16.1 SeverSocket 类 / 230
 - 16.2 建立回送服务器 / 232
 - 16.3 建立非阻塞式服务器 / 233
 - 16.4 建立多线程回送服务器 / 238
 - 16.5 总结 / 240
- 第 17 章 建立 Web 服务器 / 241
 - 17.1 Web 上的资源 / 241
 - 17.2 超文本传输协议 (HTTP) / 242
 - 17.3 HTTP 请求 / 242
 - 17.4 HTTP 响应 / 244
 - 17.5 通用网关接口 (CGI) / 246
 - 17.6 实现一个 Web 服务器 / 247
 - 17.7 结论 / 271
- 第 18 章 客户/服务器 Internet 应用程序示例 / 273
 - 18.1 建立一个聊天客户端 / 273
 - 18.2 建立一个多线程服务器 / 277
 - 18.3 带有对象流的网络技术 / 282
 - 18.4 分布式列表数据结构 / 283
 - 18.5 基于 Socket 的分布式列表 / 289
 - 18.6 总结 / 307
- 第 19 章 URL 类 / 308
 - 19.1 简介 / 308
 - 19.2 URL 类 / 309
 - 19.3 使用 URL 类 / 311
 - 19.4 URLConnection 类 / 312
 - 19.5 使用 URLConnection 类 / 315
 - 19.6 JDK1.1 HttpURLConnection 类 / 316
 - 19.7 URLStreamHandlerFactory 接口 / 318
 - 19.8 URLStreamhandler 类 / 319
 - 19.9 ContentHandlerFactory 接口 / 319
 - 19.10 ContentHandler 类 / 320
 - 19.11 HTTP 协议处理程序 / 321
 - 19.12 纯文本内容处理器 / 327
 - 19.13 指针协议处理程序 / 329
 - 19.14 URLEncoder 类 / 336
 - 19.15 JDK1.2 URLDecoder 类 / 336
 - 19.16 总结 / 337
- 第 20 章 数据报网络编程 / 338
 - 20.1 DatagramPacket 类 / 338
 - 20.2 DatagramSocket 类 / 339
 - 20.3 接收 UDP 包 / 341
 - 20.4 UDP 包的传输 / 342
 - 20.5 UDP 示例 / 343
 - 20.6 总结 / 350
- 第 21 章 数据报编程示例 / 352
 - 21.1 一个 ping 客户机 / 352
 - 21.2 UDP 上的 DNS / 359
 - 21.3 Daytime 服务器 / 362
 - 21.4 总结 / 364
- 第 22 章 组播网络编程 / 366
 - 22.1 组播 / 366
 - 22.2 广播 / 369
 - 22.3 **JDK 1.1** MulticastSocket 类 / 370
 - 22.4 使用组播 / 372
 - 22.5 一个对等的组播聊天系统 / 373
 - 22.6 一个客户/服务器组播聊天系统 / 379
 - 22.7 小结 / 383
 - 22.8 结论 / 383
- 第 23 章 远程方法调用 / 387
 - 23.1 引言 / 387
 - 23.2 RMI 日期服务器 / 390
 - 23.3 RMI 类文件定位 / 394
 - 23.4 RMI 相关信息包 / 396
 - 23.5 **JDK 1.1** 远端接口 / 397
 - 23.6 **JDK 1.1** 类命名 / 398
 - 23.7 JDK LocateRegistry 类 / 401
 - 23.8 **JDK 1.1** Registry 接口 / 402
 - 23.9 **JDK 1.1** RemoteObject 类 / 403
 - 23.10 **JDK 1.1** RemoteServer 类 / 404
 - 23.11 **JDK 1.1** UnicastRemoteObject 类 / 405
 - 23.12 **JDK 1.1** RemoteStub 类 / 406
 - 23.13 **JDK 1.1** Unreferenced 接口 / 407

- 23.14 **JDK1.1** RMISocketFactory 类 / 407
- 23.15 **JDK1.1** RMIFailureHandler 接口 / 408
- 23.16 **JDK1.2** RMIClientSocketFactory 接口 / 409
- 23.17 **JDK1.2** RMIServerSocketFactory 接口 / 409
- 23.18 RMI 对象激活 / 410
- 23.19 总结 / 410
- 第 24 章 RMI 示例 / 411
 - 24.1 一个 RMI 会话系统 / 411
 - 24.2 带有回叫的 RMI 会话 / 417
 - 24.3 RMI 分配列表 / 426
 - 24.4 RMI 对等服务 / 440
 - 24.5 RMI 及 Web 浏览器 / 445
- 第 25 章 CORBA / 447
 - 25.1 引言 / 447
 - 25.2 为什么分布使用 CORBA? / 448
 - 25.3 CORBA 核心结构 / 448
 - 25.4 接口定义语言 / 448
 - 25.5 一个银行账目例子 / 450
 - 25.6 CORBA 对象模型 / 451
 - 25.7 标准 CORBA 服务 / 453
 - 25.8 一个 CORBA 银行账目 / 454
 - 25.9 CORBA 银行客户 / 459
 - 25.10 结论 / 461
 - 25.11 参考文献 / 461
- 第 26 章 servlet / 462
 - 26.1 技术比较 / 462
 - 26.2 调用 servlet / 463
 - 26.3 开发 servlet / 465
 - 26.4 servlet 内部 / 466
 - 26.5 servlet 接口 / 468
 - 26.6 **JSDK2.0** SingleThreadModel 接口 / 469
 - 26.7 ServletConfig 接口 / 469
 - 26.8 ServletRequest 接口 / 470
 - 26.9 ServletResponse 接口 / 471
 - 26.10 ServletContext 接口 / 471
 - 26.11 ServletInputStream 类 / 472
 - 26.12 ServletOutputStream 类 / 472
 - 26.13 GenericServlet 类 / 473
 - 26.14 一个 server-side 包含的例子 / 474
 - 26.15 HttpServlet 类 / 475
 - 26.16 HttpServletRequest 接口 / 476
 - 26.17 HttpServletResponse 接口 / 478
 - 26.18 **jSDK2.0** HttpSession 接口 / 480
 - 26.19 **jSDK2.0** HttpSessionBindingListener 接口 / 481
 - 26.20 **jSDK2.0** HttpSessionBindingEvent 类 / 481
 - 26.21 **jSDK2.0** Cookie 类 / 481
 - 26.22 提供 HTTP 获取请求 / 482
 - 26.23 提供 HTTP 发送请求 / 484
 - 26.24 总结 / 486
- 第 27 章 servlet 的应用 / 487
 - 27.1 servlet 的持久性 / 487
 - 27.2 一个 servlet 访问计数器 / 492
 - 27.3 用 servlet 协作 / 495
 - 27.4 基于 servlet 的分布式列表 / 496
 - 27.5 结论 / 510
- 第 28 章 报文流 / 513
 - 28.1 报文的优点 / 513
 - 28.2 MessageOutput 类 / 514
 - 28.3 MessageInput 类 / 515
 - 28.4 MessageOutputStream 类 / 516
 - 28.5 MessageInputStream 类 / 518
 - 28.6 使用报文流 / 520
 - 28.7 MessageCopier 类 / 520
 - 28.8 一个进行事务处理的例子 / 523
 - 28.9 总结 / 529
- 第 29 章 队列报文流 / 530
 - 29.1 Queue 类 / 530
 - 29.2 QueueOutputStream 类 / 532
 - 29.3 QueueInputStream 类 / 533
 - 29.4 队列的填充 / 535
 - 29.5 队列的清空 / 536
 - 29.6 总结 / 537
- 第 30 章 多路复用报文流 / 538
 - 30.1 MultiplexOutputStream 类 / 538
 - 30.2 MultiplexInputStream 类 / 540
 - 30.3 Demultiplexer 类 / 542
 - 30.4 DeliveryOutputStream 类 / 544

- 30.5 Recipient 接口 / 546
- 30.6 一个简单的协作工具 / 546
- 30.7 总结 / 555
- 第 31 章 路由报文流 / 556**
 - 31.1 RoutingOutputStream 类 / 556
 - 31.2 RoutingInputStream 类 / 558
 - 31.3 Router 类 / 559
 - 31.4 总结 / 563
- 第 32 章 通用报文服务器 / 564**
 - 32.1 GenericServer 类 / 564
 - 32.2 GenericMessageCopier 类 / 566
 - 32.3 GenericHandler 类 / 567
 - 32.4 总结 / 572
- 第 33 章 构建通用报文客户类 / 573**
 - 33.1 GenericClient 类 / 573
 - 33.2 Client 接口 / 579
 - 33.3 创建一个高级的聊天系统 / 580
 - 33.4 扩展这些通用类的功能 / 585
 - 33.5 总结 / 586

附录 A 联 网 技 术 / 589

- A.1 网络概述 / 589
- A.2 局域网 / 590
- A.3 广域网 / 592
- A.4 Internet / 597
- A.5 参考书目 / 604

附录 B 数 据 表 / 605

- B.1 Java 转义符 / 605
- B.2 字符编码 / 605
- B.3 字符编码别名 / 607
- B.4 ASCII 字符集 / 610
- B.5 Unicode 2.0 块分配 / 612

- B.6 修正的 UTF-8 编码 / 615
- B.7 1 到 7 的乘法表 / 615
- B.8 IP 地址类别 / 616
- B.9 选定的众所周知的 UNIX TCP 和 UDP 服务 / 616
- B.10 HTTP 请求 / 617
- B.11 HTTP 响应 / 618
- B.12 HTTP / 619
- B.13 CGI 的环境变量 / 621
- B.14 MIME 类型 / 621
- B.15 许可对象 / 622

第一部分

预备知识

这部分将向程序员简要介绍 Java 网络编程和 Java 安全模型方面的一些重要概念。此外，还将概述异常处理以及 Java 多线程程序设计技术，本书后边的章节中将会用到这些概念。

第 1 章 网络技术入门

本章介绍 TCP/IP 网络的基本知识，其中包括协议栈和协议分层的概念、IP、TCP、UDP、端口、Socket 等内容。

第 2 章 Java 安全模型

本章概述了浏览器对访问系统资源所施加的限制。SecurityManager 类能够对不可信代码可能会执行的一些操作施加一定的限制。如果准备将网络程序作为 Java 的 applet 嵌入到网页当中，那么就必须了解这些限制。最后简要讨论了同防火墙相关的问题以及 Java 的“沙箱”安全模型。

第 3 章 异常处理机制

本章介绍了 Java 的异常处理模型以及同异常处理相关的类。另外还讲解了如何使用异常、如何定义你自己的异常以及为什么要定义自己的异常等问题。

第 4 章 多线程综述

Java 编程语言本身就支持多线程，而多线程能够在编写功能强大的客户端和服务端的应用程序时起非常重要的作用。本章简单介绍多线程的这种能力以及同多线程相关的一些类，最后讲解了由多线程所带来的一系列问题。

第 1 章 网络技术入门

简单地说，网络就是使用相同数据传输协议、相互连接的信息设备的集合。

1.1 协议栈与层

使用网络服务意味着把数据从网络上的一个设备传输到网络上的另一个设备。为了在设备之间进行通信，这些设备必须使用相同的数据传输语言。协议就是这样一种语言。

当在网络上传输特定长度的数据时，整个传输过程会使用到一系列相关的网络协议。其中，整个传输过程包括从推动物理链路上的电压信号（或光脉冲）到连续的数据流抵达要使用这些数据的应用程序的过程。这些协议间的相互关系可以非常形象地看作一个协议栈或分层系统。在这个系统中，最底层为硬件，最顶层为实际产生和使用数据的网络程序。位于中间的各层具有类似于把不同的物理网络连接成更大的网络等的辅助功能。网络中每一层完成一个相对独立的功能，并且仅与它的紧上一层和紧下一层打交道（见图 1.1）。

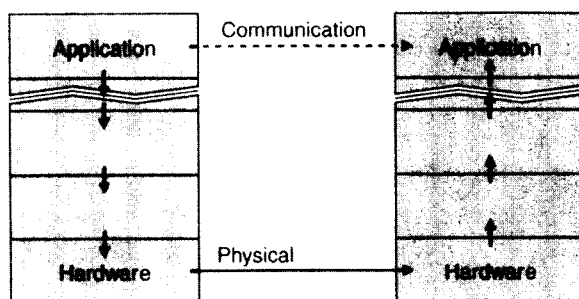


图 1.1 网络协议栈

协议栈这一抽象概念的关键在于，在协议栈中的每一层仅需要处理它自己的具体功能，而且，每一层的具体功能实际上都可以用若干个不同的、可互换的协议来实现。例如，在许多办公大厦中都能看到的标准的局域网以太网（Ethernet LAN）（它是一种同硬件相关的数据链路层协议）可以同时支持 IP、IPX(Novell)和 AppleTalk 协议，而所有这些都是互不相同的网络层协议。

1.1.1 OSI 七层参考模型

开放系统互联参考模型（OSI）是由国际标准化组织（ISO）认定和实施的一个七层协议栈模型。尽管真正的 OSI 协议从未在美国流行过，但是，当设计网络协议栈时这个模型具有一定的参考价值。OSI 参考模型经常用于阐述网络分层的原理（见图 1.2）。

1.1.2 IP 协议组

由于因特网（Internet）和内联网(Intranet)都是网际协议（IP），所以 Internet/Intranet 程序员都会对 IP

协议组非常感兴趣。IP 协议组从概念上把网络分为五层而不是七层（见图 1.3）。

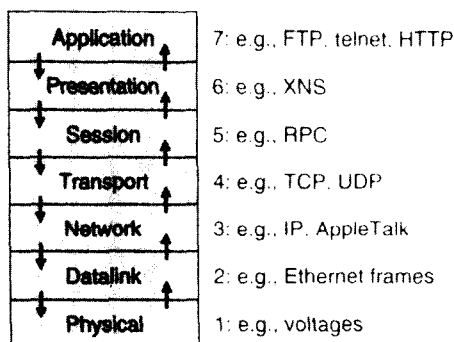


图 1.2 OSI 联网协议栈

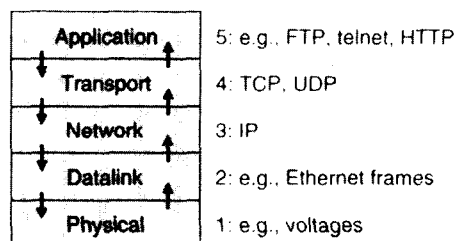


图 1.3 IP 网络协议栈

1.1.3 封装

协议栈中的每一层仅仅同紧挨着它的上一层和下一层打交道。这种通信方式是通过采用打包和解包的手段来实现的。封装指的是把每层的数据包嵌入到紧挨着它的下一层的数据包中的过程（见图 1.4）。解包指的是同打包相反的过程——当数据在接收方的协议栈中向上传输时，要去除较低层进行打包时所附加的信息。

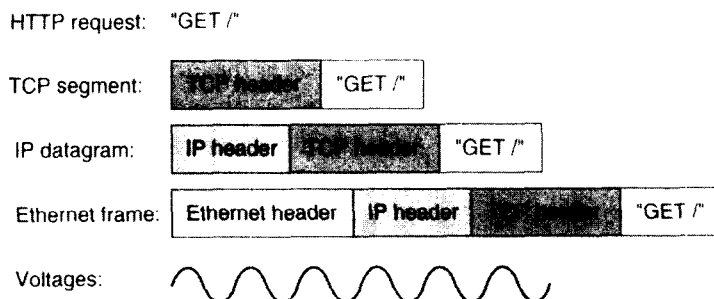


图 1.4 协议层间的封装

1.1.4 实际过程

TCP/IP 协议的通信过程如下：首先，某个应用程序产生了一个数据流；然后，TCP/IP 协议把数据流传递给传输层，该层把数据打包到 TCP 数据包中，并完成传输层的一些其它功能，如纠错等；接着，TCP 把 TCP 数据包传递给 IP 层，IP 层把它们放到 IP 数据报中，并执行网络层的一些其它功能，如路由选择等。下一步，IP 数据报将到达数据链路层。多数情况下，数据链路层是以太网。

以太网把 IP 数据报打包到帧中，为在物理电缆上传输数据做好准备。帧以电脉冲的形式在电缆中传输，直到由物理网络中的另一个以太网卡或网络接口接收为止。如果最初数据流是要传输到一台本地机器上，那么这本地目标机器上的以太网卡就能够直接从电缆上接收到这些帧。否则，这些帧将会由某种网关上的以太网接口所接收，这种网关可能是 IP 路由器。最后，IP 路由器将帧还原为 IP 数据包，并给这些数据包选择路由，再将它们为这个数据包将要通过的输出接口进行重新打包，把它们发送到目标机器所在的网络（见图 1.5）。

当远程主机接收到这些地址指向它的帧时，就开始进行解包过程。在把这些收到的数据一层一层地向应用层上传时，从包含数据包的数据包中可以通过解包得到其中的数据包。接收方的应用程序最终就接收到了发起这次数据通信的应用程序所产生的数据流。就作为接收方的应用程序而言，来自网络的数据