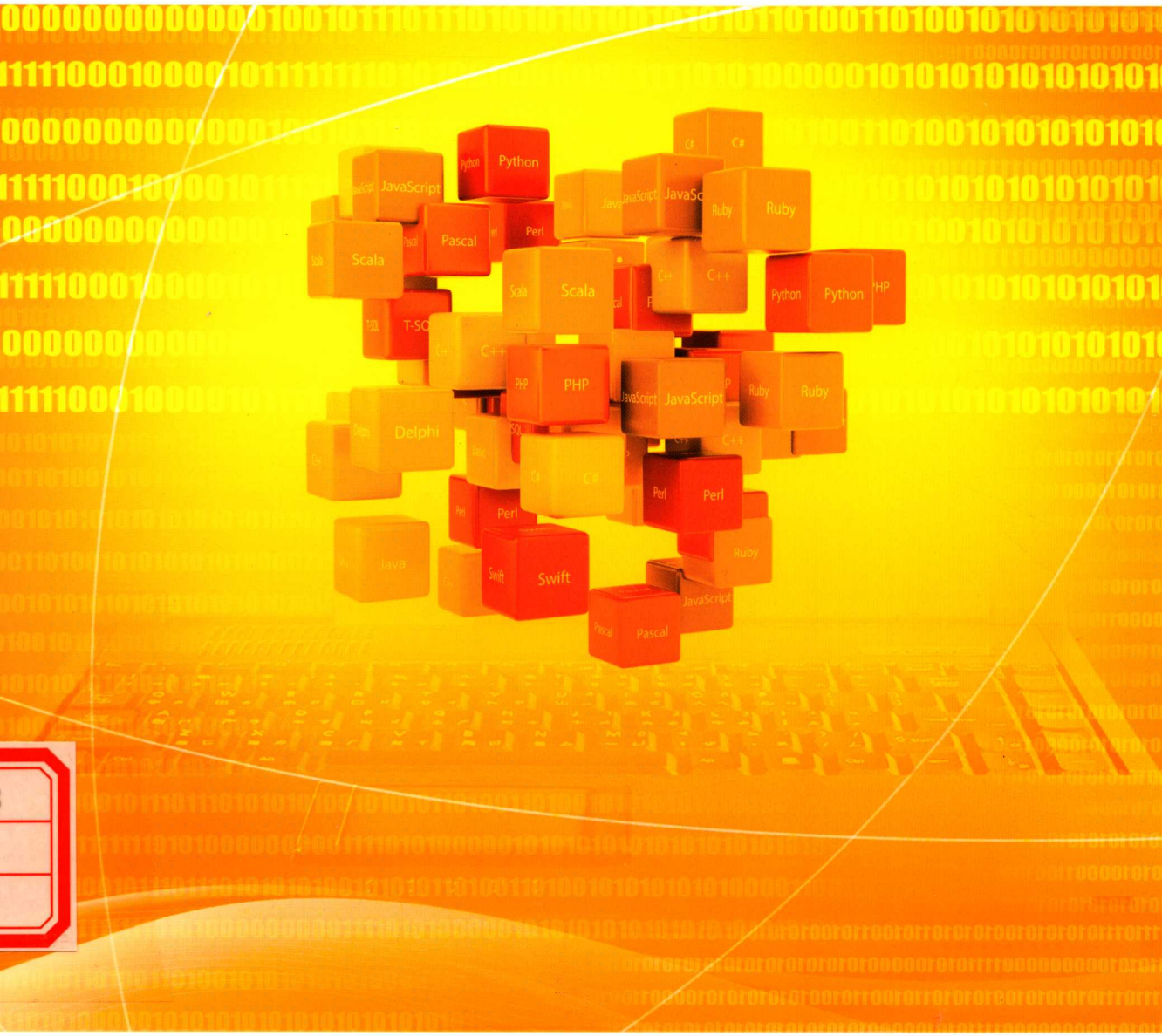
新编高等院校计算机科学与技术规划教材

网络程序设计——基于Java 8

Network Programming — Based on Java 8

刘海霞　编著



北京邮电大学出版社
www.buptpress.com

新编高等院校计算机科学与技术规划教材

网络程序设计

——基于 Java 8

刘海霞　编著

北京邮电大学出版社
www.buptpress.com

内容简介

本书是在设定读者已经有一定的Java编程经验的基础上编写而成的，并不涉及Java的基本结构、语法、面向对象、继承、多态、数组、常用类等基础内容。

本书专注于讲解Java的网络程序设计，并从Java的输入输出流开始，因为输入输出流是网络程序的基础。最终大部分的网络应用通常都会转化为输入输出流的操作。之后会按照TCP/IP协议栈逐层讲解基于IP、URL、TCP、UDP等协议的网络程序设计方法和开发包中的类。之后还会涉及最新的Java 8版本中关于NIO、NIO.2、异步通信等较新的接口和类库及其具体的使用方式。本书旨在使读者能够系统地了解Java关于网络程序开发的方方面面，从而能够进一步开发出自己的协议和应用。

本书编写了近百个程序实例，用来帮助读者更好地理解技术要点和使用方法。读者在实际开发中可以参考或直接使用。

本书的编写力求语言简练、注重思路并逐步深入，适用于需要使用Java进行网络程序设计的计算机专业人员和科技工作者，也可以作为高等学校计算机相关专业的专业教材和参考书。

图书在版编目(CIP)数据

网络程序设计：基于Java 8 / 刘海霞编著. -- 北京：北京邮电大学出版社，2016.12

ISBN 978-7-5635-4984-9

Ⅰ. ①网… Ⅱ. ①刘… Ⅲ. ①JAVA语言—程序设计 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2016)第297558号

书　　名：网络程序设计——基于Java 8

著作责任者：刘海霞　编著

责任编辑：徐振华　孙宏颖

出版发行：北京邮电大学出版社

社　　址：北京市海淀区西土城路10号(邮编：100876)

发 行 部：电话：010-62282185　传真：010-62283578

E-mail：publish@bupt.edu.cn

经　　销：各地新华书店

印　　刷：保定市中画美凯印刷有限公司

开　　本：787 mm×1 092 mm　1/16

印　　张：17.25

字　　数：431千字

版　　次：2016年12月第1版　2016年12月第1次印刷

ISBN 978-7-5635-4984-9　　定　价：36.00元

前　言

网络的硬件平台和架构经历了20年的迅猛发展，无论是百兆、千兆以太网，还是无线网，犹如骨干高速公路和密布的公路、街道已经连接了我们所有人。计算机网络是为了人们能够互相连接、共享信息而建，在信息高速公路上，最重要的还是各种各样的网络应用。

本书所指的网络程序设计，并不是指Web程序设计，二者之间还是有本质区别的。网络程序设计基于系统的底层网络体系架构和网络协议，创建基于网络通信的应用程序并多采用C/S模式。网络程序设计的原理和思路并不是Java特有的，毕竟遵循的都是网络和网络协议。实际上，本书讲解的很多技术和思路对于使用C语言或其他编程语言都是适用的，只是不同的语言封装的类或者API各不相同，表述也不尽相同。之所以选择Java进行网络程序设计，首先，Java天生具有网络的基因，非常适合编写网络应用；其次，Java提供了丰富和简洁的网络类库，使得开发变得更简单，从而使程序设计人员可以将精力更多地放在应用的业务逻辑上；此外，Java还有跨平台、线程安全、异常检测、自动垃圾回收等机制，都很适合进行网络应用的开发。

本书的内容分为7章。建议读者从第1章开始顺序阅读。

第1章讲述了网络的分层结构和参考模型，尤其是TCP /IP协议，这有助于帮助读者理解网络和协议分层，为应用层软件的开发奠定基础。

第2章讲解了Java的输入和输出，虽然可能有些读者已经熟悉，但还是建议读者浏览一遍，本章讲述的内容比一般的书籍更广泛和深入，并提供了很多的程序实例。

第3章到第6章的内容是逐步深入的。

第3章介绍了Java中IP地址的表示方法和InetAddress类，以及URL类、URLConnection类和URLStreamHandler类的使用方法。其中InetAddress类表示IP地址和网络主机。URL表示互联网网络资源，通过URL可以直接获取输入流来访问该资源。到URL网络资源的连接使用URLConnection类定义，它可以实现同URL资源的交互访问。

第 4 章介绍了基于 TCP 传输层协议的网络程序设计方法，基于 TCP 的网络通信是可靠的、有序的、差错控制的。其中，主要介绍了服务器端套接字 ServerSocket和客户端套接字 Socket。同时，还详细介绍了通信的机制和相关 Java 类的详细使用方法，以及如何控制网络参数。

第 5 章介绍了基于 UDP 传输层协议的网络程序设计方法，基于 UDP 的网络通信是不可靠的、无序的、无差错控制的。UDP 的通信机制和 TCP 完全不同。本章主要介绍了 UDP 套接字 DatagramSocket 和用来表示数据报的 DatagramPacket，并同样详细介绍了通信的机制和相关 Java 类的使用方法，以及如何控制网络参数。

第 6 章主要介绍了 Java 中的 NIO 和 NIO. 2。其中 NIO 是 New IO 的意思，它的定义比基本 IO 更加灵活和有效。NIO. 2 也称为异步 IO，它提供了异步 IO 的操作能力，更加节省资源和高效。本章重点介绍了 Java NIO 中的 Buffer 类、Channel 相关类和 Java NIO. 2 中的 AsynchronousSocketChannel 类、AsynchronousServerSocketChannel 类等。各章中的程序实例显示了应用不同的 IO 技术如何解决同一个问题。

第 7 章讲解了多线程和并发的内容。关于并发包中各个类的讲解是较新的内容，建议读者放在最后阅读。

书中所有的代码为了方便读者阅读，放在了灰色背景框中，并予以了解释。书中列举了一些 Java 定义的 API，读者可以参考 Java 的官方文档进行阅读。这些 API 使用的是 Java 8 的官方文档，和之前的版本略有差异。

本书在编写和出版的过程中得到了很多人的鼓励帮助，在此一并感谢。山东科技大学开设网络程序设计这门课以来，在多年的教学中，同学们的积极反馈使课程内容逐步完善。本书结合了笔者近十几年的开发经验和软件项目积累。这些项目长期而稳定的运行，证明了网络程序设计在网络应用中是非常重要的内容。

本书适合有 Java 编程基础的读者。本书的编写由浅入深，注重编程思路，力求语言简练，并提供了丰富的程序实例，适用于需要使用 Java 进行网络程序设计的计算机专业人员和科技工作者，也可以作为高等学校计算机相关专业的专业教材和参考书。本书的所有程序基于 Java SDK 的最新版本 Java 8，都经过了调试，可以正确运行。建议读者采用 Java 7 之后的开发包，异步通信中的一些类和 API 比更早期的版本还是有一些变化的。

本书的出版基于多年的开发经验和教学经验，即使如此，写作过程仍十分艰苦，唯恐描述不够严谨或者讲解不够准确，辜负了读者的期望。再熟悉不过的东西，当要落在纸上的时候，总想再求证一下，以免出现错误，因此也耗费了大量的精力和时间，但这都是值得的。即便如此，本书也难免有错误和不妥之处，欢迎读者和行业同仁批评指正。

目　录

第1章 概 述

众所周知，我们所处的是日新月异的网络和互联网时代。谈及网络，人们会有不同的理解和感受——移动应用 App、网络支付、新闻网站、4G、更快的网络速度、WiFi、光纤、电话……无论是看见、看不见的设备、通信线路，还是无时无刻影响着我们的网络应用，网络已经无处不在。

1.1 什么是网络

谈及网络，不同专业的人们会有不同的理解。简单地讲，计算机网络是由地理上分散的、具有独立功能的多台计算机，通过通信设备和线路互相连接起来，在相应的网络软件的配合下，实现计算机之间通信和资源共享的系统。

网络中连接的是不同操作系统、不同硬件体系结构、不同功能的设备和计算机。

网络实际用 6 个字就可以概括：开放、互连、共享。

如同人与人之间的交往，首先要敞开心扉，用语言来沟通。网络中的计算机要互相连接起来，就要彼此能够通信。如同人类世界，人们说着不同的语言，难以有效地沟通，所以要有统一的官方语言，或翻译成能够互相理解的语言。不同的计算机也因生产厂家不同、体系结构不同难以互相识别。所以计算机系统彼此之间要互连，就必须遵守统一的标准，称之为开放系统。没有开放系统，计算机系统就不可能互连。

计算机和人类的语言还是很不一样的。计算机系统包含硬件体系结构、操作系统和应用软件。所以在互连时也要实现从硬件到操作系统、软件不同层面的互相识别。所以开放系统在各个层面规定了统一的标准。最著名的是 ISO/OSI 参考模型和 TCP/IP 协议集。在互连的基础上，计算机之间能够有效地沟通，即共享数据，网络就变得有意义。

共享指共享网络资源。网络基础设施如同高速公路，无论高速公路多宽、多长，最终还是要有交通运输才有意义。网络之上，最终要“跑”应用，各种各样的网络应用。网络应用除了实现应用之间的通信，如社交软件，还可以彼此共享信息。有了越来越多的信息共享，就有了大数据分析，就可以实现异地业务的办理，就可以快捷地办理各种业务……网络已经改变了我们的生活。

1.1.1 ISO/OSI 参考模型

网络通信的核心是协议。协议是指进程之间交换信息与完成任务所使用的一系列规则和规范。协议规定了进程之间交换消息的顺序、格式。

通过定义协议，可以看出，两个进程只要遵循相同的协议，就可以相互交换信息，并且能够理解交换信息的格式和内容。两个进程可以使用不同的编程语言来编写，可以存在于两台完全不同的计算机上。国际标准化组织给出了一个通用的参考协议，称为国际标准化祖织开放式系统互连参考模型(International Organization for Standardization / Open System Interconnection Reference Modal，ISO/OSI RM)。

OSI 参考模型本身并不是一个完整的网络体系结构，因为它并没有明确地描述各层的协议和服务，它仅仅具有指导意义，声明每层的功能是什么。不过，ISO 已经为各层的网络协议制定了标准，它们并不是参考模型的一部分，而是作为独立的国际标准公布的。ISO 七层模型如图 1.1 所示。

层 次	名 称	协议内容
7	应用层	关于应用程序的规定
6	表示层	数据的表示方法
5	会话层	会话管理
4	传输层	完善下层功能
3	网络层	从多个计算机中选择通信对象
2	数据链路层	管理一对一的数据通信
1	物理层	关于硬件的规定

图 1.1　ISO 七层模型

其中，第 1 层物理层规定的是计算机体系结构中的最底层——硬件层面。第 7 层应用层规定的是应用软件部分的功能。其余各层为操作系统层面的功能。

(1) 各层的功能

物理层：实现网络连接，按比特流传送数据信息。

数据链路层：建立相邻节点之间的数据链路，按照数据帧(Data Frame)的格式组织数据，控制帧的传输，进行差错控制，以及提供数据链路通路的建立、维持和释放。

网络层：接收来自其他计算机的数据包，或发送数据包。

传输层：提供独立于具体通信协议的数据传输服务，在计算机之间建立通信通道。

会话层：在计算机之间组织会话。

表示层：处理数据的表示方法并进行转换，以消除不同的语义差异。

应用层：专门针对网络通信应用程序提供服务。

(2) 网络协议

人与人之间通过语言来交流，网络中的主机之间通过网络协议来交流。

计算机网络的目的是为了实现计算机之间的通信，而任何双方要成功地进行通信，必须遵守一定的信息交换规则和约定，这些信息交换规则和约定就称为通信协议(Protocol)。

计算机上的网络接口卡、通信软件、通信设备都是遵循一定的协议设计的，必须符合一定的协议规范。

为了减少协议设计的复杂性，大多数网络都按层或者级的方式来组织，每一层都建立在它的下层之上。不同的网络在分层数量和各层的名字、内容与功能上都不尽相同，然而，在所有的网络中，每一层的目的都是向它的上一层提供一定的服务，而把这种服务是如何实现的细节对上层加以屏蔽。

每一层都有一个或多个协议，几个层合成一个协议栈(Protocol Stack)。协议的分层模型便于协议软件按模块方式进行设计和实现，这样每层协议的设计、修改、实现和测试都可以独立进行，从而减少复杂性。

例如，物理层常用的协议是 RS-232、RJ-45；数据链路层常用的协议是 PPP；网络层常用的协议是 IP；传输层常用的协议是 TCP 和 UDP；应用层常用的协议是 HTTP、FTP 等，还包括自定义的各种协议。

不同机器内包含相同协议层的实体称为对等进程，对等进程是利用协议进行通信的主体。相邻层之间通过接口来定义相互关系。层和协议的集合称为网络体系结构。

1.1.2 TCP/IP 协议

TCP/IP(Transmission Control Protocol/Internet Protocol，传输控制协议/网间协议)是网络通信协议中应用最广泛的协议。互联网采用的就是 TCP/IP 协议，这是一种简化的 ISO/OSI 模型。有这样一种说法，正是因为 TCP/IP 协议的简化和开放，使得它得以广泛应用，并成为事实上的标准，从而有了互联网的蓬勃发展。

TCP/IP 协议体系是一组协议，因其两个著名的协议 TCP 和 IP 而得名，所以实际包含的协议远不止这两个协议。TCP/IP 协议体系在和 OSI 的竞争中取得了决定性的胜利，得到了广泛的认可，成为事实上的网络协议体系标准。

1. TCP/IP 分层模型

TCP/IP 协议体系和 OSI 参考模型一样，也是一种分层结构。如图 1.2 所示，它是由基于硬件层次上的 4 个概念性层次构成，即链路层、网络层、传输层和应用层，分别对应 ISO/OSI 模型的 2 层、3 层、4 层、7 层。

层 次	名 称
7	应用层
4	传输层
3	网络层
2	链路层

图 1.2 TCP/IP 分层结构

分层结构中每一层完成的功能与 OSI 参考模型是类似的。

2. 链路层

链路层也称为网络接口层、数据链路层，它是 TCP/IP 的最底层，但是 TCP/IP 协议并没有严格定义该层，它只是要求主机必须使用某种协议与网络连接，以便能在其上传递 IP

分组。链路层的协议有很多，如以太网协议、PPP 协议等。其中，IEEE802.3 是著名的以太网标准。

3. 网络层

网络层（Internet Layer）俗称 IP 层，它处理机器之间的通信。它接收来自传输层的请求，传输某个具有目的地址信息的分组。该层把分组封装到 IP 数据报中，填入数据报的首部，使用路由算法来选择是直接把数据报发送到目标机还是把数据报发送给路由器，然后把数据报交给下面的网络接口层中的对应网络接口模块。

为了将不同的 LAN 互连，需要一种实现这种连接的网络协议，即网络层协议。IP 就属于网络层协议，功能在于能从网络上众多的计算机中选出接收方，并使之与发送方建立连接。

IP 中的计算机地址称为 IP 地址。IP 地址的划分由网络管理者确定。现行的 IPv4 中，IP 地址采用句点分隔的 4 组 8 位二进制数来表示。

若 IP 地址在全球范围内不唯一，会使数据包不知发往何处。因此，IP 地址由 Internet NIC(Network Information Center)统一进行管理，并进一步由地区性的 NIC 负责某个范围内的具体分层管理。

4. 传输层

传输层的基本任务是提供应用层之间的通信，即端到端的通信。传输层管理信息流，提供可靠的传输服务，以确保数据无差错地按序到达。为了这个目的，传输层协议软件要进行协商，让接收方回送确认信息及让发送方重发丢失的分组。传输层协议软件将要传送的数据流划分成分组，并把每个分组连同目的地址交给下一层去发送。

5. 应用层

在这个最高层，用户调用应用程序来访问 TCP/IP 互联网络提供的多种服务。应用程序负责发送和接收数据。每个应用程序选择所需的传输服务类型，可以是独立的报文序列，或者是连续的字节流。应用程序将数据按要求的格式传送给传输层。

TCP/IP 是 Internet 的主要协议，定义了计算机和外部设备进行通信所使用的规则。TCP/IP 网络参考模型包括 4 个层次：应用层、传输层、网络层、链路层。ISO/OSI 网络参考模型则包括 7 个层次：应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。

大多数基于 Internet 的应用程序被看作 TCP/IP 网络的最上层——应用层，如 FTP、HTTP、SMTP、POP3、TELNET 等。

网络层对 TCP/IP 网络中的硬件资源进行标识。连接到 TCP/IP 网络中的每台计算机（或其他设备）都有唯一的地址，这就是 IP 地址。

在 TCP/IP 网络中，不同的机器之间进行通信时，数据的传输是由传输层控制的，这包括数据要发往的目标机器及应用程序、数据的质量控制等。TCP/IP 网络中最常用的传输协议就是 TCP(Transport Control Protocol)和 UDP(User Datagram Protocol)。

尽管 TCP/IP 协议的名称中，传输层只有 TCP 这个协议名，但是在 TCP/IP 的传输层同时存在 TCP 和 UDP 两个协议。

TCP 是一种面向连接的、保证可靠传输的协议。通过 TCP 协议传输，得到的是一个顺序的、无差错的数据流。

UDP 是一种无连接的协议，每个数据报都是一个独立的信息，包括完整的源地址和目

的地址，它在网络上以任何可能的路径传往目的地，因此能否到达目的地，到达目的地的时间以及内容的正确性都是不能保证的。

使用 UDP 时，每个数据报中都给出了完整的地址信息，因此不需要建立发送方和接收方的连接。对于 TCP 协议，由于它是一个面向连接的协议，在 Socket 之间进行数据传输之前必然要建立连接，所以在 TCP 中多了一个连接建立的时间。

使用 UDP 传输数据时是有大小限制的，每个被传输的数据报必须限定在 64 KB 之内。而 TCP 没有这方面的限制，一旦连接建立起来，双方的 Socket 就可以按统一的格式传输大量的数据。UDP 是一个不可靠的协议，发送方所发送的数据报并不一定以相同的次序到达接收方。而 TCP 是一个可靠的协议，它确保接收方完全正确地获取发送方所发送的全部数据。

TCP 在端点与端点之间建立持续的连接并进行通信。建立连接后，发送端将发送的数据印记了序列号和错误检测代码，并以字节流的方式发送出去；接收端则对数据进行错误检查并按序列顺序将数据整理好，数据在需要时可以重新发送，因此整个字节流到达接收端时完好无缺。这与两个人打电话的情形是相似的。TCP 协议具有可靠性和有序性，并且以字节流的方式发送数据，它通常被称为流通信协议。

为什么还要非可靠传输的 UDP 协议呢？主要的原因有两个：一是可靠的传输是要付出代价的，对数据内容正确性的检验必然占用计算机的处理时间和网络的带宽，因此 TCP 传输的效率不如 UDP 高；二是在许多应用中并不需要保证严格的传输可靠性，如视频会议系统，并不要求音频视频数据绝对正确，只要保证连贯性就可以了，这种情况下显然使用 UDP 会更合理一些。UDP 与通过邮局发送邮件的情形非常相似。

1.2 什么是网络程序设计

网络的目的是实现信息的共享，所以可以简单地认为基于网络的程序设计都属于网络程序设计的范畴。我们使用的软件哪些是网络程序设计的应用呢？

- 发送、接收电子邮件。
- 通过 FTP 上传、下载文件。
- 通过 HTTP 浏览 Web 网站。
- QQ 和微信。
- 远程登录服务器和网络设备。
- 在网站上查询个人信息，如网银。
- 网络游戏。
- 防病毒软件的自动更新。
- 360 安全卫士。

网络化增强了程序的功能。通过网络，一个程序可以和其他主机上的程序进行通信，获取任何其他计算机中共享的信息。

1.2.1 网络程序的模式

网络程序大致可以划分为两种模式：C/S 和 B/S。

1. C/S(客户端/服务器)

客户端指在使用者的计算机上安装客户端软件，使用者通过客户端软件连接服务器，获取服务器上的数据信息并进行计算和显示。C/S 模式的客户端也称为胖客户端。

客户端也可以向服务器提交数据，经过服务器的存储、计算、统计等功能，再重新获取更新的数据。客户端也可以连接多个服务器，分别获取信息、提交信息，按照程序设计的功能进行实时的交互。

服务器端接收来自于客户端的请求，对数据进行必要的计算和抽取，以一定的格式发送给用户。

服务端也可以不止一台主机，多台主机实现热备、冷备或者负载均衡。它们对客户端来说是透明的，就像一台服务器一样工作。

网络客户端可以使用标准协议与服务器进行通信，如 HTTP、FTP、SMTP 等，也可以自己定义协议，交互的数据也可以采取任意格式，只要客户端和服务器端的程序能够理解和识别交互的消息，就可以进行通信。

并行计算也是一种网络程序。当一些大规模的复杂运算无法由一台计算机来完成的时候，可以将复杂的问题进行分解，分配任务到不同的客户端计算机上，客户端完成任务后向服务器提交结果，从而共同完成复杂的并行计算。

本书所涉及的内容正是以 C/S 模式为主的网络应用程序设计。

2. B/S(浏览器/服务器)

使用浏览器作为客户端软件，通过 WWW 技术和 HTTP 协议，向 Web 服务器发起服务请求，并浏览结果。

客户端只需要安装浏览器，一般不需再安装其他客户端软件，维护的代价很小，所以也叫瘦客户端。

Web 服务器接收客户端请求，通常后台还会有其他的服务器完成业务功能，这些服务器共同完成数据计算功能。之后，把结果通过 Web 服务器以网页的形式呈现给客户端。

与 C/S 模式相比，B/S 模式的应用程序，其运算集中在 Web 服务器和其他业务服务器，更新和升级应用程序不需更新客户端程序。

B/S 模式的应用软件，虽然基于互联网或网络，但属于 Web 开发的范畴，本书并不涉及。

1.2.2 为什么使用 Java

Java 语言提供了强大的网络库来实现网络应用。Java 语言是完全基于面向对象的程序设计语言，对于网络功能的封装非常丰富。

Java 自诞生就具有网络的基因。Java 提供了对网络支持的多个完整软件包，并且 Java 关于网络的软件包简单易用，可以使开发者专注于网络应用业务逻辑的设计和开发，而不必

纠结于网络标准协议的具体实现。

所以本书采用 Java 来讲解网络程序开发的各种技术。

虽然藉由 Java,但其中的网络程序开发的原理和思路,对于各种开发语言都是适用的,对于各种操作系统平台也是适用的,并非仅限于 Java 语言本身。

不同的语言,其程序组织结构可能不同,开发包可能不同,API 的调用方式可能不同,但基本的网络程序设计原理是一样的。

学习网络程序设计还应该学习标准网络协议。绝大多数基于互联网的网络协议的技术细节都可以在 RFC(Request for Comment)中找到。

RFC 是描述互联网相关技术规范的文档,包含了互联网通信协议的几乎所有文档。想要了解一个网络协议的具体内容,RFC 是最基本的文档。

RFC 的技术文档是公开的,可以自行查阅网址:http://www.ietf.org/rfc.html。

在本书后面的章节中,将会逐一对网络程序设计中涉及的技术进行详细的讲解,并分析其原理和技术,通过程序示例来演示各种技术和 API。

第 2 章　Java 的输入和输出

本章重点

本章介绍了 Java 的基本输入流和输出流。

Java 的输入流和输出流实现了标准输入输出、文件的读写、基本数据类型的读写、数组的读写，以及各种进一步处理这些基本 IO 的过滤流类。

Java 的输入和输出与文件、数组、类对象的定义分离，只负责实现 IO 功能。

Java 网络程序、节点之间的通信会转换为输入流和输出流的读写，所以掌握 Java 的输入和输出是非常重要的。

本章还详细介绍了 Java 的各种输入流类和输出流类的具体使用方法，并对它们的应用场合进行了分析。

Java 的输入和输出可以说是 Java 里面最有趣也是初学者最不容易学习的一部分。Java 的文件操作、标准 I/O 操作、网络应用之间的通信最终都会归结为输入和输出的操作，即读和写的操作，非常简洁而且有效。

Java 的输入和输出不容易学习是因为它的层次结构比较复杂，涉及的类较多，最初学习时很难清楚如何选择合适的类，必须要有清晰的思路。但是相比其他设计语言，Java 的输入和输出定义的层次和概念还是非常明确的。

学习 Java 的网络程序设计，首先要熟悉 Java 的输入和输出。因为建立网络连接之后，通信双方传递消息也好，传递对象也好，最终就是双方建立一对读写通信流，选择和使用最合适的流类进行读写，读使用输入流，写使用输出流。

2.1　流

所有的计算机都有输入和输出设备。有些设备是输入设备，如键盘；有些设备是输出设备，如屏幕；当然还有一些既是输入设备也是输出设备，如硬盘。所谓输入，就是从设备获取或读取数据；所谓输出，就是将数据写入或者发送给设备。

输入即 Input，输出即 Output，输入输出即 IO。IO 操作通常称为读写。

应用程序也具有 IO 功能。Java 的 IO 使用“流”的概念来表示。IO 流涉及数据源和目的地。流是从源“流向”目的地的数据流。Java 将各种数据源和目标之间数据的传输统一