

国外计算机科学教材系列

用 TCP/IP 进行网际互连

第3卷：客户机 - 服务器编程和应用（第2版）

Internetworking With TCP/IP

Vol III : Client - Server Programming And Application (Second Edition)

DOUGLAS E. COMER

DAVID L. STEVENS

著

赵刚 林瑶 蒋慧 等译

谢希仁 校



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



PRENTICE HALL 出版公司

内 容 提 要

本书是一部计算机网络经典性教科书(现在是第二版)。它是目前美国大多数大学里所开设的计算机网络课程的主要参考书。目前国内外能见到的各种有关 TCP/IP 的书籍,其主要内容均出自本书。本书的特点是:强调原理,概念准确,深入浅出,内容丰富且新颖。全书共分为三卷。第 3 卷主要讨论应用软件如何使用 TCP/IP,重点研究了客户机-服务器范例,并考察了在分布式程序中的客户机和服务器,举例说明了各种设计,讨论了应用网关和隧道技术。全书共 28 章,各章之后附有很多很好的习题。本书可供计算机和通信专业的研究生、高年级本科生作为教科书和学习参考书,也可供从事科研和技术开发的人员参考。

© 1996 by Prentice-Hall, Inc.

本书中文简体版由电子工业出版社和美国 Prentice Hall 出版公司合作出版。未经许可,不得以任何手段和形式复制或抄袭本书内容。版权所有,侵权必究。

丛 书 名: 国外计算机科学教材系列

原 书 名: Internetworking With TCP/IP Vol. III: Client-Server Programming and Applications(Second Edition)

书 名: 用 TCP/IP 进行网际互连第 3 卷:客户机-服务器编程和应用(第 3 版)

著 者: DOUGLAS E. COMER & DAVID L. STEVENS

译 者: 赵 刚 林 瑶 蒋 慧 等译

审 校 者: 谢希仁

责任编辑: 张月萍

印 刷 者: 顺义县天竺颖华印刷厂印刷

出版发行: 电子工业出版社出版、发行 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话(68214070)

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 27 字数: 657 千字

版 次: 1998 年 6 月第 1 版 1998 年 6 月第 1 次印刷

印 数: 10000 册

书 号: ISBN 7-5053-4632-6/TP·2206

定 价: 40.00 元

著作权合同登记号 图字: 01-97-2040

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻印必究

出版说明

计算机科学的迅速发展是 20 世纪科学发展史上最伟大的事件之一。从 1946 年第一台笨重而体积庞大的计算机的发明至今, 仅仅半个多世纪, 计算机已经变得小巧无比却又能力非凡。它的应用已经渗透到了社会的各个方面, 成为当今所谓的信息社会的最显著的特征。

处于世纪之交科技进步的大潮中, 我国正在加强计算机科学的高等教育, 着眼于为下一世纪培养高素质的计算机人才, 以适应信息社会加速度发展的需要。当前, 全国各类高等院校已经或计划在各专业基础课程规划中增加计算机科学的课程内容, 而作为与计算机科学密切相关的计算机、通信、信息等专业, 更是在酝酿着教学的全面革新, 以期规划出一整套面向 21 世纪的、具有中国高校计算机教育特色的课程计划和教材体系。值此, 我们不妨借鉴并引进国外具有先进性、实用性和权威性的大学计算机教材, 洋为中用, 以更好地服务于国内的高校教育。

美国 Prentice Hall 出版公司是享誉世界的高校教材出版商, 自 1913 年公司成立以来, 即致力于教育图书的出版。它所出版的计算机教材在美国为众多大学所采用, 其中有不少是专业领域中的经典名著。许多蜚声世界的教授学者成为该公司的资深作者, 如: 道格拉斯·科默 (Douglas Comer), 安德鲁·坦尼伯姆 (Andrew Tanenbaum), 威廉·斯大林 (William Stallings)……几十年来, 他们的著作教育了一批批不同肤色的莘莘学子, 使这些教材同时也成为全人类的共同财富。

为了保证本系列教材翻译出版的质量, 电子工业出版社和 Prentice Hall 出版公司共同约请北京地区的清华大学、北京大学、北京航空航天大学, 上海地区的上海交通大学、复旦大学, 南京地区的南京大学、解放军通信工程学院等全国著名的高等院校的教学第一线的几十位教师参加翻译工作。这中间有正在讲授同类教材的年轻教师和博士, 有积累了几十年教学经验的教授和博士生导师, 还有我国著名的计算机科学家。他们的辛勤劳动保证了本系列丛书得以高质量地出版面世。

如此大规模地引进计算机科学系列教材, 在我们还是第一次。除缺乏经验之外, 还由于我们对计算机科学的发展, 对中国高校计算机教育特点认识的不足, 致使在选题确定、翻译、出版等工作中, 肯定存在许多遗憾和不足之处, 恳请广大师生和其他读者提出批评、建议。

电子工业出版社

URL: <http://www.phei.com.cn>

Prentice Hall 出版公司

URL: <http://www.prenhall.com>

前 言

Douglas E. Comer 博士的系列著作——《用 TCP/IP 进行网际互连》是一套引人瞩目的书。能向读者介绍该书第三卷的修订版,对我来说实在是荣幸之至。这套书开始创作于 1987 年,那时它还不为很多人所知。然而现在它已成为学习 Internet 协议族的首选材料,而 Internet 协议族业已使与厂商无关的计算机通信成为可能。

在我看来,这部创新的著作是最有望抵制所谓“Internet 要变成哑巴了”的论调。当媒介和企业家们让大众的想像被“Internet 神秘主义”所填满时,是 Comer 博士清晰地阐述了那些使 Internet 成为可能的技术的真实状况。

尽管我已学习了这一套书的所有三本,但我觉得,Doug 与 David L. Stevens 合著的第三卷——客户机-服务器编程和应用,与今天的 Internet 特别有关。这本书教给我们如何设计和构建客户机-服务器应用,而且更重要的是,它还教给我们如何理解每种设计决策中所蕴涵的权衡。

因此,我邀请您加入一个难忘的行程——学习网络互连技术的理论、设计以及实现的原理和内容。

Marshall T. Rose
Theorist, Implementor, and Agent Provocateur
Del Mar, California

序 言

我们很高兴为大家介绍网际互连系列中第三卷的修订版。粗略地讲,第一卷研究了“什么是 TCP/IP 互连网?”的问题,第二卷研究了“TCP/IP 软件是怎样工作的?”,它比第一卷阐述了更多的细节,且探讨更为深入。本卷将研究“应用软件如何使用 TCP/IP?”这个问题,它重点研究了客户机-服务器范例,并考察了在分布式程序中的客户机和服务器这两部分。本卷举例说明了每种设计,并讨论了包括应用层网关和隧道在内的技术。另外,本卷还重温了几个标准应用协议,并用它们来说明一些算法和实现技术。

所做修订都遵从最近的标准。例如,例子中的代码已使用 ANSI C 重写,并且有关 NFS 讨论的章节已改变为第 3 版。另外,还增加了一些新章节解释一些程序中所蕴涵的概念,像 slirp 程序,它提供用拨号电话连接进入到 Internet。对客户机-服务器系统发生故障方式的讨论也加以扩充:用整个新的一章着重讨论死锁(deadlock)和活锁(livelock)。该章考察了问题发生的原因,以及避免发生这些问题的技术。最后,原书行文中的少量印刷错误和意义不够明确的地方现都已纠正。

本书代码可在线获得。要通过 Web 得到一个拷贝,可在如下网址的连网书表中查找第三卷:

<http://www.cs.purdue.edu/homes/comer/books.html>

要通过 FTP 访问代码,请使用:

<ftp://ftp.cs.purdue.edu/pub/Xinu/TCPIP-vol3.bsd.dist.tar.Z>

本书的组织与上一版相同。开始几章介绍了客户机-服务器模型,以及应用程序用于接入到 TCP/IP 协议软件的插口(socket)接口。此外,还描述了并发进程和用于创建他们的操作系统函数。随后的几章介绍了客户机和服务器设计。

本书阐明了各式各样可能的设计并不是没有规则的。实际上,这些设计都是遵循了一种模式。在考虑了并行性和传输的选择后,就可以理解这一观点。例如,有一章讨论了一个使用面向连接传输(如 TCP)的非并发服务器设计,而另一章讨论了一个相似的设计,但它使用无连接传输(如 UDP)。

我们描述了每个设计如何适应于各种可能的实现,但是,我们并没有试图开发一种客户机-服务器交互的抽象“理论”。我们只是强调实用的设计原则,以及对程序员来说是重要的技术。每种技术在一些情况下都有其优点,并且每种技术都已用于正在工作的软件中。我们相信,理解这些设计之间的概念联系,将有助于读者理解每种方法的优缺点,并使在它们之间进行选择更为容易。

本书包含了多个例子程序,他们展现了各种设计实际上是如何进行的。大多数例子实现了标准的 TCP/IP 应用协议。在每一种情况下,我们都试图选择一个应用协议,使它可表达一种设计思路而又不太难理解。因此,虽然很少有令人激动的例子程序,但这里的每一个例子都说明了一个重要的概念。在第三卷的这个版本中,所有的例子程序都使用 BSD UNIX 插口机制;相关版本中含有使用 AT&T 的 TLI 协议接口的相同例子。

后几章讨论远程过程调用的概念,并描述它是怎样被用于构造分布式程序的。这些章将远程过程调用技术与客户机-服务器模型相联系,并说明如何使用软件从一个远程过程调用描述生成出客户机和服务器程序。有关 TELNET 的章节展现了细枝末节如何在一个实际工作的程序中占据支配地位,以及即使是实现一个简单的、面向字符的协议,其代码如何会变得复杂。

本书很大部分的重点在并发处理。写过并发程序的学生可能熟悉我们所描述的许多概念,因为这些概念适用于所有的并发程序,而不仅仅是网络应用。没写过并发程序的学生可能会觉得这些概念很难。

本书适于作为向高年级或研究生介绍连网知识的一个学期的课程。由于本书重点是如何使用互连网,而不是互连网是如何工作的,因此学生几乎不需要太多的网络背景知识就能理解这些内容。只要学生们按合适的进度循序渐进,较低级课程中不会有特别的概念令人感到太难。介绍操作系统概念或并发编程实际经验的基础课程,可提供最佳背景材料。

学生只有亲手使用教材后,才会欣赏它。因此,任何课程都应安排编程实践,强迫学生将其想法运用到实际程序中。大学本科生可通过反复设计其他的应用协议来学习基本概念。研究生则应构建更为复杂的分布式程序,这些程序强调一些细微的技术(如第 15 章中并发管理技术,和第 17 章中互连技术)。

在此,要感谢许多人的帮助。珀杜(Purdue)大学的 Internet 研究小组的成员们给本书原稿提供了许多技术信息和建议。Christine Comer 进行了修订并改进了行文和一致性。

Douglas E. Comer

David L. Stevens

目 录

第 1 章 引言和概述	(1)
1.1 TCP/IP 的使用	(1)
1.2 为分布式环境设计应用程序	(1)
1.3 标准和非标准的应用协议	(2)
1.4 使用标准应用协议的例子	(2)
1.5 一个连接的例子	(3)
1.6 使用 TELNET 访问其他服务	(3)
1.7 应用协议和软件的灵活性	(4)
1.8 从提供者的角度看服务	(5)
1.9 本教材的其余部分	(5)
1.10 小结	(6)
第 2 章 客户机-服务器模型与软件设计	(7)
2.1 引言	(7)
2.2 动机	(7)
2.3 术语和概念	(8)
2.3.1 客户机和服务器	(8)
2.3.2 特权和复杂性	(8)
2.3.3 标准的和非标准的客户机软件	(9)
2.3.4 客户机的参数化	(9)
2.3.5 无连接的和面向无连接的服务器	(10)
2.3.6 无状态的和有状态的服务器	(11)
2.3.7 一个有状态的服务器的例子	(11)
2.3.8 无状态是一个协议问题	(13)
2.3.9 作为客户机的服务器	(13)
2.4 小结	(14)
第 3 章 客户机-服务器软件中的并发处理	(16)
3.1 引言	(16)
3.2 网络中的并发性	(16)
3.3 服务器中的并发性	(17)
3.4 术语和概念	(18)
3.4.1 进程概念	(18)

3.4.2	程序与进程	(19)
3.4.3	过程调用	(19)
3.5	一个创建并发进程的例子.....	(20)
3.5.1	一个顺序执行的 C 实例	(20)
3.5.2	程序的并发版	(21)
3.5.3	时间分片	(22)
3.5.4	使各进程分离	(23)
3.6	执行新的代码.....	(24)
3.7	上下文切换和协议软件设计.....	(24)
3.8	并发性和异步 I/O	(25)
3.9	小结.....	(25)
第 4 章	协议的程序接口	(27)
4.1	引言.....	(27)
4.2	不精确指明的协议软件接口.....	(27)
4.2.1	优点与缺点	(27)
4.3	接口功能.....	(28)
4.4	概念性接口的规格说明.....	(28)
4.5	系统调用.....	(29)
4.6	网络通信的两种基本方法.....	(29)
4.7	UNIX 中所提供的基本 I/O 功能	(30)
4.8	用 UNIX I/O 结合 TCP/IP	(31)
4.9	小结.....	(31)
第 5 章	插口接口	(33)
5.1	引言.....	(33)
5.2	Berkeley 的插口	(33)
5.3	指明一个协议接口.....	(33)
5.4	插口的抽象.....	(34)
5.4.1	插口描述符和文件描述符	(34)
5.4.2	针对插口的系统数据结构	(35)
5.4.3	使用插口	(35)
5.5	指明端点地址.....	(36)
5.6	类属地址结构.....	(36)
5.7	与插口一同使用的主要的系统调用.....	(38)
5.7.1	socket 调用	(38)
5.7.2	connect 调用	(38)
5.7.3	write 调用	(38)
5.7.4	read 调用	(39)

5.7.5 close 调用	(39)
5.7.6 bind 调用	(39)
5.7.7 listen 调用	(39)
5.7.8 accept 调用	(40)
5.7.9 TCP 所使用的插口调用小结	(40)
5.8 用于整数转换的实用例程	(41)
5.9 在程序中使用插口调用	(41)
5.10 插口调用的参数所使用的符号常量	(41)
5.11 小结	(42)
第 6 章 客户机软件设计中的算法和问题	(44)
6.1 引言	(44)
6.2 不是研究细节而是学习算法	(44)
6.3 客户机体系结构	(44)
6.4 标识一个服务器的位置	(45)
6.5 分析地址参数	(46)
6.6 查找域名	(47)
6.7 由名字查找某个知名端口	(47)
6.8 端口号和网络字节序	(48)
6.9 由名字查找协议	(48)
6.10 TCP 客户机算法	(49)
6.11 分配一个插口	(49)
6.12 选择一个本地协议端口号	(50)
6.13 选择本地 IP 地址中的一个基本问题	(50)
6.14 将一个 TCP 插口连接到某个服务器	(51)
6.15 使用 TCP 与服务器通信	(51)
6.16 从一个 TCP 连接中读取响应	(52)
6.17 关闭 TCP 连接	(52)
6.17.1 对部分关闭(partial close)的需要	(52)
6.17.2 部分关闭的操作	(53)
6.18 UDP 客户机的编程	(53)
6.19 连接的和非连接的 UDP 插口	(54)
6.20 对 UDP 使用 connect	(54)
6.21 使用 UDP 与服务器通信	(54)
6.22 关闭一个使用 UDP 的插口	(54)
6.23 对 UDP 的部分关闭	(55)
6.24 关于 UDP 不可靠性的警告	(55)
6.25 小结	(55)

第 7 章 客户机软件举例	(57)
7.1 引言	(57)
7.2 小例子的重要性	(57)
7.3 隐藏细节	(57)
7.4 针对客户机程序的过程库例子	(58)
7.5 connectTCP 的实现	(58)
7.6 connectUDP 的实现	(59)
7.7 构成连接的过程	(59)
7.8 使用例子库	(62)
7.9 DAYTIME 服务	(63)
7.10 针对 DAYTIME 的 TCP 客户机实现	(63)
7.11 从 TCP 连接中进行读	(65)
7.12 TIME 服务	(65)
7.13 访问 TIME 服务	(66)
7.14 精确时间和网络时延	(66)
7.15 针对 TIME 服务的 UDP 客户机	(66)
7.16 ECHO 服务	(68)
7.17 针对 ECHO 服务的 TCP 客户机	(68)
7.18 针对 ECHO 服务的 UDP 客户机	(70)
7.19 小结	(72)
第 8 章 服务器软件设计的算法和问题	(74)
8.1 引言	(74)
8.2 概念性的服务器算法	(74)
8.3 并发服务器和循环服务器	(74)
8.4 面向连接的和无连接的访问	(75)
8.5 面向连接的服务器	(75)
8.6 无连接的服务器	(76)
8.7 故障、可靠性和无状态	(76)
8.8 优化无状态服务器	(77)
8.9 四种基本类型的服务器	(79)
8.10 请求处理时间	(79)
8.11 循环服务器的算法	(80)
8.12 一种循环的、面向连接的服务器的算法	(80)
8.13 用 INADDR_ANY 绑定一个知名端口	(80)
8.14 将插口置于被动模式	(81)
8.15 接受连接并使用这些连接	(81)
8.16 一个循环的、无连接的服务器的算法	(81)

8.17	在无连接的服务器中构造一个应答	(82)
8.18	并发服务器的算法	(82)
8.19	主进程和从进程	(83)
8.20	并发的、无连接的服务器的算法	(83)
8.21	并发的、面向连接服务器的算法	(84)
8.22	把单独的程序作为从进程来使用	(84)
8.23	使用单个进程获得表面上的并发性	(85)
8.24	各服务器类型所适用的场合	(85)
8.25	服务器类型小结	(86)
8.26	重要问题——服务器死锁	(87)
8.27	其他的实现方法	(88)
8.28	小结	(88)
第 9 章	循环的、无连接服务器(UDP)	(90)
9.1	引言	(90)
9.2	创建一个被动插口	(90)
9.3	进程结构	(93)
9.4	TIME 服务器举例	(94)
9.5	小结	(95)
第 10 章	循环的、面向连接的服务器(TCP)	(97)
10.1	引言	(97)
10.2	分配一个被动的 TCP 插口	(97)
10.3	用于 DAYTIME 服务的服务器	(98)
10.4	进程结构	(98)
10.5	DAYTIME 服务器举例	(99)
10.6	关闭连接	(101)
10.7	连接终止和服务器的脆弱性	(101)
10.8	小结	(102)
第 11 章	并发的、面向连接的服务器(TCP)	(103)
11.1	引言	(103)
11.2	并发 ECHO	(103)
11.3	循环与并发实现的比较	(103)
11.4	进程结构	(104)
11.5	并发 ECHO 服务器举例	(104)
11.6	清除游离(errant)进程	(108)
11.7	小结	(108)

第 12 章 单进程、并发服务器(TCP)	(110)
12.1 引言	(110)
12.2 服务器中的数据驱动处理	(110)
12.3 用单个进程进行数据驱动处理	(111)
12.4 单个进程服务器的进程结构	(111)
12.5 单进程 ECHO 服务器举例	(112)
12.6 小结	(114)
第 13 章 多协议服务器(TCP, UDP)	(116)
13.1 引言	(116)
13.2 减少服务器数量的动机	(116)
13.3 多协议服务器的设计	(116)
13.4 进程结构	(117)
13.5 一个多协议 DAYTIME 服务器的例子	(117)
13.6 共享代码的概念	(120)
13.7 并发多协议服务器	(121)
13.8 小结	(121)
第 14 章 多服务服务器(TCP, UDP)	(123)
14.1 引言	(123)
14.2 合并服务器	(123)
14.3 无连接的、多服务服务器的设计	(123)
14.4 面向连接的、多服务服务器的设计	(124)
14.5 并发的、面向连接的、多服务服务器	(125)
14.6 单进程的、多服务服务器的实现	(125)
14.7 从多服务服务器调用单独的程序	(126)
14.8 多服务、多协议设计	(127)
14.9 多服务服务器的例子	(128)
14.10 静态的和动态的服务器配置	(134)
14.11 UNIX 超级服务器, inetd	(135)
14.12 inetd 服务器的例子	(137)
14.13 小结	(138)
第 15 章 服务器并发性的统一、高效管理	(140)
15.1 引言	(140)
15.2 在循环设计和并发设计间选择	(140)
15.3 并发等级	(141)
15.4 需求驱动的并发	(141)

15.5	并发的代价	(141)
15.6	额外开销和时延	(142)
15.7	小时延可能出麻烦	(142)
15.8	进程预分配	(143)
15.8.1	UNIX 中的预分配	(143)
15.8.2	面向连接服务器中的预分配	(144)
15.8.3	无连接服务器中的预分配	(145)
15.8.4	预分配、突发通信量和 NFS	(146)
15.8.5	多处理器上的进程预分配	(146)
15.9	延迟的进程分配	(146)
15.10	两种技术统一的基础	(147)
15.11	技术的结合	(147)
15.12	小结	(148)
第 16 章	客户机中的并发	(150)
16.1	引言	(150)
16.2	并发的优点	(150)
16.3	运用控制的动机	(150)
16.4	与多个服务器的并发联系	(151)
16.5	实现并发客户机	(152)
16.6	单个进程的实现	(152)
16.7	一个使用 ECHO 的并发客户机例子	(153)
16.8	并发客户机的执行	(158)
16.9	例子代码中的并发性	(159)
16.10	小结	(159)
第 17 章	运输层和应用层的隧道技术	(161)
17.1	引言	(161)
17.2	多协议环境	(161)
17.3	混合网络技术	(162)
17.4	动态电路分配	(163)
17.5	封装和隧道技术	(164)
17.6	通过 IP 互连网的隧道技术	(164)
17.7	客户机和服务器之间的应用级隧道技术	(165)
17.8	隧道技术、封装和电话拨号线	(165)
17.9	小结	(166)
第 18 章	应用级网关	(168)
18.1	引言	(168)

18.2	在受约束的环境中的客户机和服务器	(168)
18.2.1	多种技术的现实	(168)
18.2.2	有限功能的计算机	(168)
18.2.3	安全性引起的连通性约束	(169)
18.3	使用应用网关	(169)
18.4	通过邮件网关互操作	(169)
18.5	邮件网关的实现	(170)
18.6	应用网关与隧道技术的比较	(171)
18.7	应用网关和有限功能系统	(172)
18.8	为解决安全问题而使用的应用网关	(173)
18.9	应用网关和额外跳问题	(174)
18.10	应用网关举例	(175)
18.11	一个应用网关的实现	(176)
18.12	应用网关的代码	(177)
18.13	网关交换的例子	(179)
18.14	使用 rfc2 和 UNIX 的 .forward 文件	(179)
18.15	一个通用的应用网关	(179)
18.16	SLIRP 的运行	(180)
18.17	SLIRP 如何处理连接	(180)
18.18	IP 寻址和 SLIRP	(181)
18.19	小结	(182)
第 19 章	外部数据表示(XDR)	(184)
19.1	引言	(184)
19.2	计算机中的数据表示	(184)
19.3	N 平方转换问题	(185)
19.4	网络标准字节序	(185)
19.5	外部数据表示的事实上的标准	(186)
19.6	XDR 数据类型	(186)
19.7	隐含类型	(187)
19.8	使用 XDR 的软件支持	(188)
19.9	XDR 库例程	(188)
19.10	一次一片地构造一个报文	(188)
19.11	XDR 库中的转换例程	(189)
19.12	XDR 流、I/O 和 TCP	(191)
19.13	记录、记录边界和数据报 I/O	(191)
19.14	小结	(192)

第 20 章 远程过程调用(RPC)的概念	(194)
20.1 引言	(194)
20.2 远程过程调用模型	(194)
20.3 构建分布式程序的两种模式	(194)
20.4 传统的过程调用的概念性模型	(195)
20.5 过程模型的扩充	(195)
20.6 传统过程调用的执行和返回	(196)
20.7 分布式系统中的过程模型	(197)
20.8 客户机-服务器和 RPC 之间的类比	(197)
20.9 作为一个程序的分布式计算	(198)
20.10 Sun Microsystems 的远程过程调用定义	(199)
20.11 远程程序和过程	(199)
20.12 减少参数的数量	(200)
20.13 标识远程程序和过程	(200)
20.14 适应远程程序的多个版本	(202)
20.15 远程程序中的互斥	(202)
20.16 通信语义	(202)
20.17 至少一次语义	(203)
20.18 RPC 重传	(203)
20.19 将一个远程程序映射到一个协议端口	(204)
20.20 动态端口映射	(204)
20.21 RPC 端口映射器算法	(205)
20.22 ONC RPC 的报文格式	(206)
20.23 对远程过程进行参数序列化	(207)
20.24 鉴别	(208)
20.25 RPC 报文表示的一个例子	(209)
20.26 UNIX 鉴别字段的例子	(209)
20.27 小结	(209)
第 21 章 分布式程序的生成	(212)
21.1 引言	(212)
21.2 使用远程过程调用	(212)
21.3 支持 RPC 的编程工具	(213)
21.4 将一个程序划分成本地过程和远程过程	(214)
21.5 为 RPC 增加代码	(214)
21.6 stub 过程	(215)
21.7 多个远程过程和分派	(215)
21.8 客户机端的 stub 过程的名字	(216)

21.9	使用 rpcgen 生成分布式程序	(217)
21.10	rpcgen 输出和接口过程	(217)
21.11	rpcgen 的输入和输出	(218)
21.12	使用 rpcgen 构建客户机和服务器	(218)
21.13	小结	(219)
第 22 章	分布式程序的生成(rpcgen 的例子)	(221)
22.1	引言	(221)
22.2	一个说明 rpcgen 的例子	(221)
22.3	查找字典	(221)
22.4	一个分布式程序的八个步骤	(222)
22.5	步骤 1: 构建一个常规的应用程序	(223)
22.6	步骤 2: 将程序划分成两部分	(227)
22.7	步骤 3: 创建一个 rpcgen 规格说明	(232)
22.8	步骤 4: 运行 rpcgen	(234)
22.9	rpcgen 产生的 .h 文件	(234)
22.10	rpcgen 产生的 XDR 转换文件	(235)
22.11	rpcgen 产生的客户机代码	(236)
22.12	rpcgen 产生的服务器代码	(237)
22.13	步骤 5: 编写 stub 接口过程	(240)
22.13.1	客户机端接口例程	(240)
22.13.2	服务器端接口例程	(242)
22.14	步骤 6: 编译并链接客户机程序	(244)
22.15	步骤 7: 编译和链接服务器程序	(247)
22.16	步骤 8: 启动服务器和执行客户机	(249)
22.17	使用 UNIX 的 make 实用程序	(249)
22.18	小结	(251)
第 23 章	网络文件系统(NFS)的概念	(253)
23.1	引言	(253)
23.2	远程文件存取和传输	(253)
23.3	对远程文件的操作	(254)
23.4	异构计算机之间的文件存取	(254)
23.5	无状态服务器	(254)
23.6	NFS 和 UNIX 的文件语义	(254)
23.7	UNIX 文件系统的回顾	(255)
23.7.1	基本定义	(255)
23.7.2	无记录界限的字节序列	(255)
23.7.3	文件拥有者和组标识符	(255)

23.7.4	保护和存取	(255)
23.7.5	打开-读-写-关闭范例	(256)
23.7.6	数据传输	(257)
23.7.7	允许搜索目录	(257)
23.7.8	随机存取	(258)
23.7.9	搜索超过文件的结束	(258)
23.7.10	文件位置和并发存取	(258)
23.7.11	在并发存取时的“写(write)”语义	(259)
23.7.12	文件名和路径	(260)
23.7.13	索引结点(inode):存储在文件中的信息	(260)
23.7.14	stat 操作	(261)
23.7.15	文件命名机制	(262)
23.7.16	文件系统 mount	(262)
23.7.17	UNIX 文件名解析	(264)
23.7.18	符号链接	(265)
23.8	NFS 下的文件	(265)
23.9	NFS 的文件类型	(265)
23.10	NFS 文件模式	(266)
23.11	NFS 文件属性	(267)
23.12	NFS 客户机和服务器	(267)
23.13	NFS 客户机操作	(268)
23.14	NFS 客户机与 UNIX	(269)
23.15	NFS mount	(269)
23.16	文件句柄	(270)
23.17	句柄取代路径名	(271)
23.18	在 UNIX 中的 NFS 客户机	(272)
23.19	无状态服务器的文件定位	(272)
23.20	对目录的操作	(272)
23.21	无状态地读目录	(273)
23.22	NFS 服务器中的多个分层结构	(273)
23.23	安装(mount)协议	(274)
23.24	小结	(274)
第 24 章	网络文件系统协议(NFS, Mount)	(276)
24.1	引言	(276)
24.2	用 RPC 定义协议	(276)
24.3	用数据结构和过程定义协议	(276)
24.4	NFS 常数、类型和数据声明	(277)
24.4.1	NFS 常数	(277)