

深入 C++ 系列



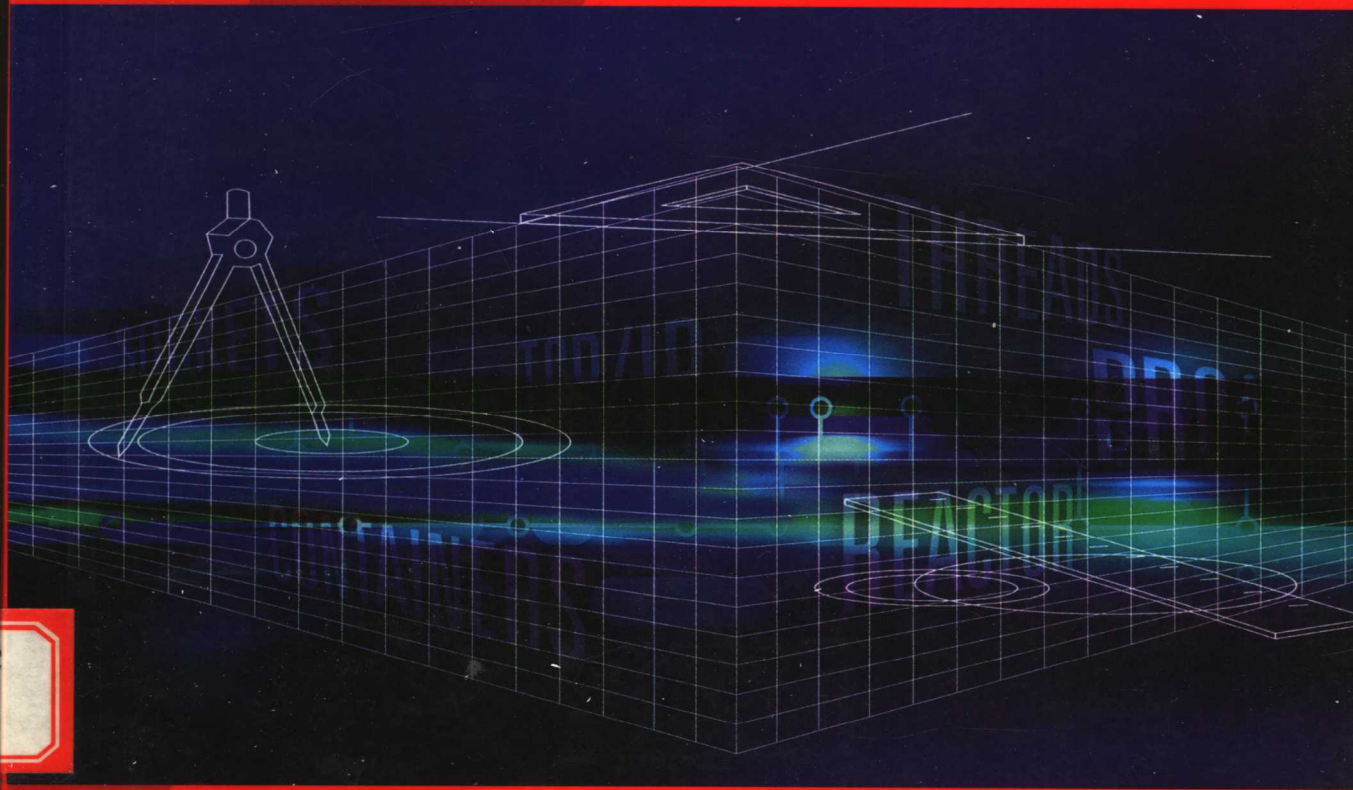
The ACE Programmer's Guide

Practical Design Patterns for Network and Systems Programming

ACE 程序员指南

网络与系统编程的实用设计模式

[美] Stephen D. Huston/James CE Johnson/Umar Syyid 著
马维达 译



随书所附光盘上包含了书中所有示例的完整源码，
针对许多流行平台编译好的可安装版本以及 ACE
的所有类的完整参考文档。



中国电力出版社

www.infopower.com.cn

深入 C++ 系列

The ACE Programmer's Guide

Practical Design Patterns for Network and Systems Programming

ACE 程序员指南

网络与系统编程的实用设计模式

[美] Stephen D. Huston/James CE Johnson/Umar Syyid 著
马维达 译



中国电力出版社
www.infopower.com.cn

The ACE Programmer's Guide: Practical Design Patterns for Network and Systems Programming
(ISBN 0-201-69971-0)

Stephen D. Huston, James CE Johnson, Umar Syyyid

Copyright ©2004 by Pearson Education, Inc.

Original English Language Edition Published by Pearson Education, Inc.

All rights reserved.

Translation edition published by PEARSON EDUCATION ASIA LTD and CHINA ELECTRIC POWER PRESS,
Copyright © 2004.

本书翻译版由 Pearson Education 授权中国电力出版社独家出版、发行。
未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 Pearson Education 防伪标签，无标签者不得销售。
北京市版权局著作权合同登记号 图字：01-2004-4935 号

图书在版编目 (CIP) 数据

ACE 程序员指南：网络与系统编程的实用设计模式 / (美) 约翰逊等编著；马维达译. —北京：中国电力出版社，2004

(深入 C++ 系列)

ISBN 7-5083-2734-9

I. A... II. ①约...②马... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 094361 号

丛 书 名：深入 C++ 系列

书 名：ACE 程序员指南：网络与系统编程的实用设计模式

编 著：(美) Stephen D. Huston 等

翻 译：马维达

责任编辑：牛贵华

出版发行：中国电力出版社

地址：北京市三里河路6号 邮政编码：100044

电话：(010) 88515918 传 真：(010) 88518169

印 刷：北京丰源印刷厂

开本尺寸：185×233 1/16 印 张：24.75 字 数：538千字

书 号：ISBN 7-5083-2734-9

版 次：2004 年 11 月北京第 1 版 2005 年 7 月第 2 次印刷

定 价：48.00 元

版权所有 翻印必究

译者序

或许我已不用再介绍 ACE (ADAPTIVE Communication Environment) 是什么, 随着 Douglas C. Schmidt 及其合作者撰写的《Pattern-Oriented Software Architecture》(卷 2) 和《C++ Network Programming》(两卷本) 的中译本在国内出版, 已经有越来越多的国内开发者在学习使用这个强大的网络软件开发工具包。自 ACE 诞生之日起, 其应用领域就一直在不断扩大, 如果你访问 <http://www.cs.wustl.edu/~schmidt/ACE-users.html>, 你就会看到, ACE 的应用领域遍及电信、航空、保险、军事、天文、游戏等众多领域; 在国内, ACE 也走进了实际的应用开发中——举个例子, 假如你购买了“金山词霸 2005”软件, 安装好, 突然发现在 C:\Program Files\Common Files\KingSoft\KSG 下有一个叫 ACE.dll 的文件; 你用工具软件查看这个 dll, 会发现像 ACE_Reactor、ACE_Mutex、ACE_Log_Msg 这样的函数名。这, 显然不是一个巧合。

那么对于许多开发者来说, 问题或许已不在于是否应该使用 ACE 开发网络软件, 而在于怎样尽快掌握它并实际应用它。有好些读者朋友曾给我写信, 说 ACE 太难学了, 我想, 这并不是一件出人意料的事情——ACE 的难学, 恰恰反映了网络软件开发的困难, 以及一些先进的设计思想的复杂和深奥。一个人只有在深入掌握了 C++ 语言、面向对象编程、网络编程及设计模式等方面的基础知识, 才能开始学习 ACE。与此同时, 尽管已经有了三本与 ACE 直接相关的书, 尽管有了大量的中文技术文档 (<http://www.flyingdonkey.com/ace/>), 面对 ACE 这样一个庞大而复杂的开发包, 你或许仍然会觉得资料不足。

没关系, 现在有了本书。本书由 ACE 的几位核心开发者撰写, 适用于 ACE 的初学者, 也可供有经验的 ACE 开发者作参考手册用。本书从介绍 ACE 开始, 循序渐进, 详细地讨论了 ACE 的 IPC 设施、进程及线程管理以及一些高级的话题, 比如共享内存、ACE Streams 框架、ACE Service Configurator 框架、定时器管理类、ACE Naming Service, 等等。在这个过程中, 作者给出了许多源码例子, 并做了详尽的解释。

有读者也许会产生这样的疑问: 那么到底应该先阅读《C++ Network Programming》还是本书。我的看法是应该先阅读前者, 即《C++ Network Programming》, 原因是该书中给出了更多的概念性阐述; 本书则更偏重于实践性的问题。

在本书末尾的“参考文献”中列出了一些与 ACE 有关的重要文献, 比如网络编程著作之父 Richard Stevens 的一些著作, 以及 MINIX 的发明者 Andrew S. Tanenbaum 的《计算机网络》, 等等。如果你能“按图索骥”, 全面而深入地学习包括本书在内的所有这些文献, 相信你一定

能成为一名真正的“王牌”(ACE)程序员。

最后，我要感谢中国电力出版社及本书的策划高军先生，使我又有了翻译 ACE 书籍的机会——能够通过翻译的方式，把 ACE 介绍给国内的程序员，始终是我的荣幸。

马维达 于上海

序

我是在上个世纪 80 年代的中期开始计算机编程的。那时和今天一样，各种计算与通信系统是异构的；也就是说，有许多不同的编程语言、操作系统以及网络协议。而且，由于通过低级和不可移植的应用编程接口（application programming interface, API）进行编程会带来各种偶发复杂性（accidental complexity），软件开发者与最终用户可以使用的网络化应用的类型相对来说很原始，在很大程度上就只有 e-mail、远程登录以及文件传输。这样一来，许多应用或是被集中在大型主机和小型计算机上，或是被放在本地的单机 PC 或工作站上。

二十年来，计算与通信信息技术（information technology, IT）已经取得了长足的进步。各种语言、编程环境、操作系统、网络协议还有中间件已变得更为成熟，更为标准化。例如，C++ 已经成为 ISO/ANSI 标准，并正被用于广泛的应用领域。同样，UNIX、Windows 和 TCP/IP 标准也已经无处不在。甚至像 CORBA、.NET 和 J2EE 这样的分布式计算中间件标准，也正日益为 IT 行业和最终用户所广泛接受。

但是，即使有了这些进步，在网络化软件设计空间中，仍然有一个层面——主机基础设施中间件（host infrastructure middleware），用于性能驱动的、多平台的、网络化和/或并发的软件系统——未能很好地得到位于其他抽象层面的标准解决方案的处理。在通过像 Sockets 或 POSIX threads（Pthreads）这样的原生 OS API 开发网络化应用时，需要完成许多繁琐、易错和不可移植的工作，主机基础设施中间件封装了原生操作系统（OS）的并发和进程间通信（interprocess communication, IPC）机制，能使上述工作变得更为容易。性能驱动的系统还具有严格的服务质量（Quality of Service, QoS）需求。

对于以下类型的系统而言，为现在流行的各种计算环境提供主机基础设施中间件特别重要：（1）高性能计算系统，比如支持科学可视化、分布式数据库服务器以及在线金融交易的系统；（2）对现实世界的各种制品进行监控的分布式实时及嵌入系统，比如航空任务和飞行管制软件、监视控制与数据采集系统以及汽车制动系统；（3）必须可移植地跨越局域和广域网运行的多平台应用。这些类型的系统正日益受到以下趋势的影响：

- 为满足 QoS 需求而访问原生 OS 机制的需要。例如，对需要长时间、双向字节流通信服务的多媒体应用而言，传统的分布中间件提供的同步请求/响应模型并不适用。同样，许多分布式中间件实现会造成相当大的开销，并且缺少足够的挂钩（hook）用以操纵其他与 QoS 有关的属性，比如等待时间、吞吐量以及抖动。
- 苛刻的开销限制和巨大的推出时间压力。全球竞争和政府对市场干预的减少正在造成

内部开发和软件质量保证 (quality assurance, QA) 预算的缩减, 特别是 OS 和中间件基础设施软件。而且, 受性能驱动的用户常常无法或不大情愿付钱购买特制的私有基础设施软件。

- **用户需要自己进行定制。**因为高性能软件常常会冲击技术的极限, 这样的软件必须针对特定的运行时环境和应用需求进行优化。通用的全能型软件解决方案的性能常常无法让人接受。

随着这些趋势的发展速度的继续加快, 给今后的网络化软件系统的开发者带来了许多挑战。特别地, 为了在今天这样竞争激烈、快速变化的 IT 行业中取得成功, 成功的中间件和应用软件必须展现出以下八个特征:

1. 可负担性 (affordability), 确保软件采购和演化 (evolution) 的总开销不会高得惊人
2. 可扩展性 (extensibility), 支持连续的快速更新和扩展, 从而满足新的需求和占领新兴的市场
3. 灵活性 (flexibility), 支持范围不断扩展的数据类型、传输流以及端到端 (end-to-end) QoS 需求
4. 可移植性 (portability), 减少在异种 OS 平台和编译器上支持各种应用所需的工作量
5. 可预测性和高效性 (predictability and efficiency), 给对延迟敏感的实时应用提供低等待时间、给有高带宽要求的应用提供高性能, 并在像无线链接这样的低带宽网络上提供可用性
6. 可靠性 (reliability), 确保应用健壮、容错
7. 可伸缩性 (scalability), 使应用有能力同时处理大量客户
8. 可信性 (trustworthiness), 在互连的系统中确保完整性、机密性和可用性

在过去二十年间, 在开发许多研究性及产品性的、性能驱动的软件系统时, 我遇到过所有这些挑战。1992 年, 为了应对并发网络编程的各种挑战, 我开始开发一个名为 ADAPTIVE Communication Environment (ACE) 的软件工具包。ACE 是一个用 C++ 编写的、可移植的面向对象主机基础设施中间件, 当时的设计目标是要解决非常实际的问题——为我的博士论文项目的实现节省时间, 论文的主题是在 SunOS 4.x 和 SunOS 5.x 操作系统上实现并行协议处理引擎。在此之前的几年里, 我已经参与过若干自由软件项目的开发, 因此我立刻就意识到, 以开放源码的形式发布 ACE 将有助于创立一个社群, 其他开发者和用户可以藉此在 QA、文档和支持方面提供帮助, 从而把 ACE 转化为高质量、产品级的软件工具包。

在过去十年里, ACE 的 30 余位核心开发者和我一起, 与全世界 1700 多位开发者协同努力, 使 ACE 发展成了一个封装并增强了广泛的原生 OS 功能的工具包——要支持性能驱动的软件系统, 这些功能是必不可少的。ACE 中的各种核心框架和 wrapper facade 提供了一组集成的可复用面向对象类, 能够简化并自动化这样一些事务: 连接建立、事件多路分离、安全而高效的进程间通信、整编/解编、应用组件的动态配置、并发以及同步。最重要的是, 开发者可以轻松地对 ACE 的各种框架进行定制, 从而支持广泛的运行时环境和应用需求。

今天, ACE 已能在许多的硬件和 OS 平台上运行, 其中包括 Windows 和 UNIX 的大多数版本, 还有许多实时和嵌入式操作系统。由于 ACE 的成熟、高效和广泛的应用范围, 数以千

计的开发团队正在使用 ACE 进行开发，范围从财富 500 强公司、小型的新兴公司直到大学和行业实验室的高级研究项目。其开放源码的开发模型及知识渊博的用户社群，在精神、热情及生产率上，可与 Linux 操作系统、Apache web 服务器以及 GNU 编译器集的开发模型及用户社群相比拟。

但多年来，学习 ACE 的惟一途径是阅读源码及应用实例、阅读描述其模式及框架设计的技术论文以及一些在线教程。尽管有数量惊人的 ACE 用户设法通过这样的方式掌握了 ACE，但这样的学习过程既耗费时日，又容易出错。在 2002 和 2003 年，我和 Steve Huston 一起撰写了两卷本的《C++ Network Programming》(C++NP)，阐述了 ACE 中的大多数常用的 wrapper facade 类和框架的意图、设计、实现以及用法。

本书所取的方向与 C++NP 不同，它快速而直接地解释怎样充分利用 ACE。除了在解释 ACE 的各种关键功能时采用了“nuts-and-bolts (具体细节)”方法，本书还讨论了 C++NP 没有涵盖的一些重要话题，包括怎样构建 ACE 及使用 ACE 的程序、ACE 的 OS 和编译器可移植性特性、ACE 日志设施、命令行选项处理、容器类、信号处理以及共享内存。本书还比 C++NP 更深入地讨论了某些问题，包括并发模型、双向流以及异步 I/O。贯穿全书的许多 ACE 应用实例将告诉你怎样应用各种关键的模式和设计原则，去成功地开发面向对象的网络化软件。通过本书的学习，你将能更好地设计并实现性能驱动的、无法在市场上买到的网络化软件，从而始终在竞争中走在前列。

我们很幸运，Steve、James、Umar 在他们异常紧张的日程表中抽出时间撰写本书。可移植、灵活、可扩展、可预测、可靠、可负担——如果你正在设计这样的软件和系统，本书及 ACE 工具包将使你在所有这些方面都变得更为得力。即使是在用了十余年时间开发 ACE，并用它构建了许多网络化软件应用之后，我发现我还是从本书中学到了许多东西。我深信，你也会一样。

——Douglas C. Schmidt

田纳西州，纳什维尔市

范德比尔特大学电子工程与计算机科学系教授，ACE 的发明者

前言

ACE (ADAPTIVE Communication Environment) 是一个强大的 C++ 工具包, 能帮助你更轻松、更迅速地开发可移植、高性能的应用, 尤其是网络化和/或多线程化应用的同时, 拥有更多的灵活性及更少的错误数量。同时, 因为 ACE 具有先进的设计与框架, 与其他开发途径相比, 你可以用少得多的代码做到所有这些事情。我们使用 ACE 已有多年, 发现它对我们很有帮助, 值得你付出任何代价。更好的是, 它是自由软件! 一直以来, 许多开发者为使用 ACE 所要付出的代价, 是克服陡峭的学习曲线。这是一个大工具包, 具有大量功能。直到最近, 最好的文档一直是源码 (当然可以自由获取) 和一批由 ACE 的研发人员撰写的学术论文 (只有高级专业人员和高年级学生才有能力研究)。这样的学习障碍使得 ACE 的力量、易用与优雅成了软件开发中隐藏最深的秘密之一。这正是我们为何如此兴奋地撰写这本书的原因! 它将使 ACE 的学习曲线变得平缓, 并把 ACE 的力量、优雅及各种能力带给所有人。

本书将告诉你与 ACE 有关的以下内容: 它的一些历史及开发方式、它的组织方式、怎样着手使用它, 以及怎样使用它的一些更为高级的能力。我们将在本书中告诉你怎样以“ACE 方式”做各种事情, 但我们不可能让本书成为一本参考大全。你应该通过本书着手学习 ACE, 并开始在你的工作中使用它。如果你已经有了一点使用 ACE 的经验, 那么书中描述的有些功能, 你可能并未见过, 所以对于有经验的 ACE 用户而言, 这本书同样也会很有用。

谁应当阅读本书

我们既想使本书成为 ACE 初学者的入门指南, 也想使它成为有经验的 ACE 用户的快速参考资料。如果你是 ACE 初学者, 我们建议你从头开始依次阅读各章。如果你是有经验的用户, 知道自己想要阅读什么, 你可以快速地找到所需的部分, 而无需阅读前面的章节。

本书是为这样的 C++ 程序员撰写的: 你已经接触过某些更高级的 C++ 特性, 比如虚继承和类模板。你还应该接触过你计划使用的操作系统所提供的各种基本设施。例如, 如果你计划编写使用 TCP/IP sockets 的程序, 你至少应该熟悉创建 sockets、建立连接及传输数据的一般方式。

对于那些从事教学工作的人而言, 本书也是一个极好的资料来源: 无论是商业培训还是高校教学。就怎样设计面向对象软件、怎样使用 C++ 设计和编写高性能且易于维护的软件系统而言, ACE 是一个极好的范例。

篇章安排

本书是一本关于怎样有效地使用 ACE 的实用指南。书中的许多源码示例将阐释怎样适当地使用所描述的 ACE 功能。我们尽量让源码示例保持简短，并切中问题的关键。有时，为了使读者的注意力集中在某个话题上，我们对源码作了删节。所有示例的完整源码可在随书附带的光盘上找到，也可在 Riverace 公司的网站上下载。在附带的光盘上还包含了 ACE 源码包的副本、针对许多流行平台编译好的可安装版本以及 ACE 的所有类的完整参考文档。

本书在起始部分讲述了许多 ACE 用户都需要了解的基本功能，并进而在此基础上加以延伸，描述各种把行为抽象进强大模式中的更高级特性。

- 第一部分介绍 ACE，并提供与 ACE 提供的设施有关的一些一般性的有用信息。第一部分还解释怎样配置并构建 ACE，以及怎样构建你的 ACE 应用。另外还将描述被广泛使用的编程辅助功能（比如日志与跟踪、命令行处理与配置访问），以及 ACE 的容器类。
- 第二部分讨论 ACE 的进程间通信（IPC）设施——从基本的、低级的 TCP/IP Sockets 包装（wrapper）类开始，一直到怎样用 ACE 的 Reactor（反应器）和 Proactor（前摄器）框架同时处理多个 sockets 及其他事件，比如定时器和信号。第二部分还描述了 ACE 的 Acceptor-Connector（接受器—连接器）框架，并在最后讨论了 ACE 提供的其他一些 IPC 包装类——这些类有许多都可用于替换所涵盖的框架中使用的 TCP/IP 包装类。
- 第三部分涵盖的是与 ACE 的进程及线程管理有关的广泛话题。这一部分解释了怎样使用 ACE 的进程管理类，怎样处理信号，接着又用三章讨论多线程化编程、线程管理以及线程安全和同步的关键内容。第三部分的最后将讨论 Active Object（主动对象），以及使用 ACE 的线程池的各种方式——这是在高性能应用中有效使用多线程的关键所在。
- 第四部分涵盖高级的 ACE 话题：共享内存，用于装配模块化数据处理流的 ACE Streams 框架，以及怎样通过使用 ACE Service Configurator（服务配置器）框架使你的应用更灵活、可配置性更好。第四部分的结尾将深入讨论 ACE 的定时器管理类和 ACE Naming Service，后者是 ACE 的网络服务组件之一，这些组件的用途是为常用的网络化应用编程任务提供帮助。

在本书的最后给出了参考书目和详尽的主题索引。

本书采用的约定

所有的 ACE 类都以 ACE_ 开头。当我们所指的是模式而不是实现它们的类时，我们会略去该前缀。例如，Reactor 模式是由 ACE_Reactor 类实现的。

所有的类成员变量都以“_”为后缀。ACE 源码采用了这一约定，我们也把它带到了本书的各个示例中。

致谢

我们应当感谢阅读并评论了整个手稿的评阅团队。Craig L. Ching、Dave Mercer、Johnny Willemsen 以及 Steven P. Witten 提供了深刻而有益的反馈和意见，极大地改善了本书的质量。

我们还感谢世界各地的许多 ACE 用户社群成员，他们无偿地花费时间评阅许多份手稿，提供了有益的更正和建议。和许多为 ACE 自身作出贡献的人一样，这些人展示了许多开放源码开发者/用户社群的协作性——特别是以下这些 ACE 用户：Bill Cassanova、Ufuk Çoban、Todd Cooper、Ning Cui、Alain Decamps、John Fowler、Chris D. Gill、Kelly F. Hickel、Don Hinton、Robert Kindred、Michael Kleck、Franz Klein、Sven Köster、Dieter Knüppel、Theo Landman、Mark Levan、Alexander Libman、John Lilley、Stephen McDonald、Mike Mullen、Mats Nilsson、Jaroslaw Nozderko、Rick Ohnemus、Wojtek Pilorz、Sunanda C. Prasad、Dietrich Quehl、Irma Rastegayeva、Michael Searles、Rich Siebel、Chris Smith、Scott Smith、Edward Thompson、Alain Totouom、Bill Trudell 以及 Lothar Werzinger。

在本书漫长的撰写过程中，Addison-Wesley 的编辑团队给了我们极大的帮助和鼓励。感谢我们的编辑：Marina Lang、Debbie Lafferty、Peter Gordon 及 Bernie Gaffney。感谢我们的文字编辑 Evelyn Pyle，是 Evelyn 完成了一项令人惊叹的工作，把我们的不同方法和风格塑造成了统一的整体。非常感谢我们的制作协调人 Elizabeth Ryan，她运用自己了不起的技能、雅致和耐心，帮助完成了全部三本与 ACE 有关的书籍。

Steve 的致谢

与我在所有工作中所感受到的感激一样，我深深地感谢我伟大的妻子 Jane，为她在本书漫长的撰写过程中所给予的持续的爱、不变的支持以及我极其需要的帮助。你的牺牲远远超出了任何人应当做到的程度，你真的是上帝赐予我的礼物——谢谢你！正如智慧的所罗门曾经说过的：“著书多，没有穷尽；读书多，身体疲倦。”（《传道书》12:12，NIV）我有一点疲倦，感谢上帝给了我完成本书所需的精力。我还要感谢我已故的母亲 Karen L. Anderson，我想，我终于学到的经验会让她满意的。

James 的致谢

我要感谢我的妻子 Karla 和我的儿子 Riley（他是在本书四年的撰写过程中出生的），感谢他们在我每次长达数小时地消失在地下办公室中时所给予我的耐心和理解。没有他们的不断支持和鼓励，我想我肯定不会走到这里。我还要感谢我的父母，他们总是鼓励我尽自己最大的努力，鼓励我永不在“第二好”的地方停下来。最后，感谢 Doug Schmidt，是他在多年前创造了 ACE，给予了我们撰写这本书的机会。

Umar 的致谢

首先我要感谢我的妻子 Ambreen 和我的儿子 Hassan。如果不是为了他们，我也许会把我的几乎所有时间用在计算机上；谢谢你们打电话让我回家。我应该特别感谢 Ambreen，感谢她多年来忍受我的喋喋不休。没有她的支持，我连一页文字也写不出来。我还要感谢我的母亲，感谢她在我生活中非常艰难的时刻鼓励我、支持我。此外，我还要感谢我的父亲，是他使我想向他证明：我能够做到。最后，感谢 Doug Schmidt 驱车来到伊利诺依州立大学，主持了一次研究生研讨会，从而使我知道了一个叫作 ACE 的框架。

结束语

这本书是真正的合著，我们每个人都撰写了数量相等的内容。因此，如果你在文本中发现任何问题，我们都应该同等地承担责任。无论你发现了什么，都请将其发送至 ACE 用户邮件列表 (ace-users@cs.wustl.edu)，或发送给我们 (ace-tutorial@tragus.org)。撰写一本主题像 ACE 这样广泛的书籍是非常困难的任务。我们拥有一个作者团队这一事实使得这项任务容易了许多。此外，来自 DOC 组织和 ACE 用户社群的帮助也展示出了无法估量的价值。

Steve Huston
James CE Johnson
Umar Syyid

目录

译者序
序
前言

第 1 部分 ACE 基础知识

第 1 章 ACE 简介	3
1.1 ACE 的历史	3
1.2 ACE 的优势	4
1.3 ACE 的组织结构	5
1.4 模式、类库和框架	5
1.5 把你的代码移植到多种操作系统	6
1.6 消除 C++ 编译器之间的差异	8
1.7 同时使用窄字符和宽字符	15
1.8 到哪里寻求更多的信息和支持	16
1.9 小结	17
第 2 章 怎样构建 ACE 并在你的程序中使用它	18
2.1 关于 ACE 版本的说明	18
2.2 ACE Distribution 指南	18
2.3 怎样构建 ACE	20
2.4 怎样在你的应用中包含 ACE	22
2.5 怎样构建你的应用	22
2.6 小结	26
第 3 章 使用 ACE 日志设施	27
3.1 基本的日志记录和跟踪	27
3.2 启用和禁用日志严重级别	32
3.3 定制 ACE 日志宏	35
3.4 重定向日志输出	41
3.5 使用回调	45
3.6 日志客户和服务器看守	49

3.7	LogManager 类.....	53
3.8	通过 ACE 日志策略进行运行时配置	56
3.9	小结	58
第 4 章	收集运行时信息	59
4.1	命令行参数和 ACE_Get_Opt.....	59
4.2	访问配置信息	63
4.3	构建参数向量	65
4.4	小结	66
第 5 章	ACE 容器	67
5.1	容器的概念	67
5.2	序列容器	69
5.3	关联容器	80
5.4	分配器	89
5.5	小结	92

第 2 部分 进程间通信

第 6 章	基本的 TCP/IP Socket 用法	95
6.1	一个简单客户	95
6.2	给客户增加健壮性	99
6.3	构建一个服务器	104
6.4	小结	108
第 7 章	处理事件及多个 I/O 流	109
7.1	Reactor 框架总览	109
7.2	处理多个 I/O 源	110
7.3	信号	120
7.4	通知	122
7.5	定时器	123
7.6	使用 Acceptor-Connector 框架	131
7.7	反应器实现	141
7.8	小结	144
第 8 章	异步 I/O 与 ACE Proactor 框架	145
8.1	为何使用异步 I/O?	145
8.2	怎样发送和接收数据	146
8.3	建立连接	153
8.4	ACE_Proactor 完成多路分离器	155
8.5	使用定时器	156

8.6	其他 I/O 工厂类	156
8.7	结合 Reactor 与 Proactor 框架	156
8.8	小结	158
第 9 章	其他 IPC 类型	159
9.1	基于 UDP/IP 的主机间 IPC	159
9.2	主机内通信	163
9.3	小结	165

第 3 部分 进程与线程管理

第 10 章	进程管理	169
10.1	派生新进程	169
10.2	使用 ACE_Process_Manager	174
10.3	使用 ACE_Process_Mutex 进行同步	178
10.4	小结	181
第 11 章	信号	182
11.1	使用包装	182
11.2	事件处理器	185
11.3	保护临界区	190
11.4	通过反应器管理信号	192
11.5	小结	192
第 12 章	基本的多线程编程	193
12.1	入门	193
12.2	基本的线程安全性	194
12.3	任务间通信	199
12.4	小结	206
第 13 章	线程管理	207
13.1	线程的类型	207
13.2	优先级与调度类	209
13.3	线程池	212
13.4	使用 ACE_Thread_Manager 进行线程管理	213
13.5	信号	216
13.6	线程启动挂钩	219
13.7	取消	220
13.8	小结	223
第 14 章	线程安全与同步	224
14.1	保护原语	224

14.2	线程同步	234
14.3	线程专有存储	240
14.4	小结	241
第 15 章	主动对象	242
15.1	模式	242
15.2	模式使用	245
15.3	小结	251
第 16 章	线程池	252
16.1	理解线程池	252
16.2	半同步/半异步模型	253
16.3	领导者/跟随者模型	263
16.4	线程池与反应器	267
16.5	小结	270

第 4 部分 ACE 高级特性

第 17 章	共享内存	273
17.1	ACE_Malloc 与 ACE_Allocator	273
17.2	ACE_Malloc 的持久功能	275
17.3	与位置无关的分配	278
17.4	用于容器的 ACE_Malloc	281
17.5	包装	293
17.6	小结	294
第 18 章	ACE Streams 框架	296
18.1	综述	296
18.2	使用单向流	297
18.3	双向流	312
18.4	小结	328
第 19 章	ACE Service Configurator 框架	330
19.1	综述	330
19.2	配置静态服务	331
19.3	设置动态服务	336
19.4	设置流	338
19.5	在执行过程中重新配置服务	339
19.6	使用 XML 配置服务与流	340
19.7	不用 svc.conf 配置服务	341
19.8	单体与服务	342

19.9	小结	342
第 20 章	定时器	343
20.1	定时器概念	343
20.2	定时器队列	343
20.3	预先构建的分派器	351
20.4	管理事件处理器	353
20.5	小结	357
第 21 章	ACE Naming Service	358
21.1	ACE_Naming_Context	358
21.2	单进程命名上下文: PROC_LOCAL	359
21.3	在一个节点上共享命名上下文: NODE_LOCAL	367
21.4	在网络上共享命名上下文: NET_LOCAL	372
21.5	小结	374
参考书目	375