第一图书网, tushu007.com

<<C++网络编程（卷2）>>

图书基本信息

书名：<<C++网络编程（卷2）>>

13位ISBN编号：9787030341983

10位ISBN编号：7030341988

出版时间：2012-7

出版时间：科学出版社

作者：Douglas C.Schmidt

页数：326

字数：523000

译者：叶斌

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

内容概要

你需要开发能被快速定制的灵活软件吗？
你需要给自己的软件添加框架的威力并提高其效率吗？
自适应通信环境（ACE）就是这样一种用于构建高性能网络应用和下一代中间件的开放源码工具包。
ACE的威力和灵活性来自面向对象的框架，通过它可以获得对网络应用的系统化复用。
ACE框架不仅能处理常用的网络编程任务，还能借助C++语言的各种特性对其进行定制，以生成完整的分布式应用。

本书重点讲述ACE的各个框架，全面涵盖了作为其架构的基础概念、模式和使用规则。
本书是设计面向对象框架的实用指南，同时也告诉开发者，怎样将框架应用到并发的网络应用中。
《C++网络编程卷1》主要介绍了ACE和wrapperfacade（包装器外观），它们是网络计算的基础成分，卷2则解释了如何在包装器外观之上创建框架，以提供更高级别的通信服务。

本书由ACE社区的两位专家撰写，内容包括：
◎ACE各个框架综述
◎网络服务的设计空间
◎对重要的ACE框架提供其关键能力的描述
◎大量C++代码示例，演示如何使用ACE框架
本书将教会开发者如何使用框架快速编写网络应用，同时降低开发难度和工作量。
对于任何从事网络应用开发的C++程序员，本书都是无法估量的财富。

作者简介

作者：（美国）休斯顿（Stephen D.Huston） （美国）施密特（Douglas C.Schmidt） 译者：王成章 白晓明 彭雪

书籍目录

第1章　用于网络编程的面向对象的框架
1.1 面向对象的框架综述
1.2 软件开发与复用技术的比较
1.2.1 框架与类库之间的比较
1.2.2 框架与组件之间的比较
1.2.3 框架与模式之间的比较
1.2.4 框架与模型集成式计算之间的比较
1.3 在网络编程中应用框架
1.4 漫游ACE框架
1.4.1 ACE综述
1.4.2 ACE框架概要
1.5 示例：网络日志服务
1.6 小结
第2章　服务以及配置的设计空间
2.1 服务以及服务器设计空间
2.1.1 短持续时间服务与长持续时间服务
2.1.2 内部服务与外部服务
2.1.3 有状态服务与无状态服务
2.1.4 分层式/模块化服务与整体式服务
2.1.5 单服务服务器与多服务服务器
2.1.6 一次性服务器与持续式服务器
2.2 设计空间的配置
2.2.1 静态命名与动态命名
2.2.2 静态链接与动态链接
2.2.3 静态配置与动态配置
2.3 小结
第3章　ACE Reactor框架
3.1 综述
3.2 ACE_Time_Value类
3.3 ACE_Event_Handler类
3.4 ACE定时器队列类
3.5 ACE_Reactor类
3.6 小结
第4章　ACE Reactor实现
4.1 综述
4.2 ACE_Select_Reactor类
4.3 ACE_TP_Reactor类
4.4 ACE_WFMO_Reactor类
4.5 小结
第5章　ACE Service Configurator框架
5.1 综述
5.2 ACE_Service_Object类
5.3 ACE_Service_Repository类
5.4 ACE_Service_Config类
5.5 小结

第6章　ACE Task框架
6.1 综述
6.2 ACE_Message_Queue类
6.3 ACE_Task类
6.4 小结
第7章　ACE Acceptor-Connector框架
7.1 综述
7.2 ACE_Svc_Handler类
7.3 ACE_Acceptor类
7.4 ACE_Connector类
7.5 小结
第8章　ACE Proactor框架
8.1 综述
8.2 异步I/O工厂类
8.3 ACE_Handler类
8.4 前摄式Acceptor-Connector类
8.5 ACE_Proactor类
8.6 小结
第9章　ACE Streams框架
9.1 综述
9.2 ACE_Module类
9.3 ACE_Stream类
9.4 小结
术语表
参考文献

章节摘录

版权页： 插图： 显式动态链接：允许一个应用程序去获取、使用，以及去除特定的与函数或者数据相关的，在DLL中定义的符号的运行时的地址绑定。
包括UNIX函数（dlopenO、dlsymO和dlcloseO）以及Windows函数（LoadLibraryO、GetProcAddressO和FreeLibraryO）在内的显式动态链接机制都是主流操作系统所支持的。
动态链接能够在减少内存中的进程以及存储在磁盘中的程序镜像的内存消耗方面提供帮助。
理想的情况是只有DLL代码的一个副本会存在，无论同时执行DLL代码的进程的数量有多少。
当在动态链接以及静态链接之间作出选择时，开发人员必须仔细地在灵活性、安全性，以及健壮性与时间以及空间效率的潜在益处之间进行权衡。
以下是采用动态链接的一些劣势。
安全性以及健壮性的问题。
一个动态链接应用程序可能不如静态链接应用程序的安全性以及健壮性高。
它可能会更加不安全，这是由于特洛伊木马可以被植入到DLL中。
由于一个错误DLL可能会破坏在相同应用程序进程中其他代码或者数据的状态，所以它的健壮性可能会更差。
运行时负载。
动态链接与静态链接相比较而言，可能会带来更多的运行时负载。
除了打开以及映射多个文件之外，在DLL中的外部符号地址必须在文件被加载的内存位置的基础上进行调整。
纵然懒散的链接能够减轻这一影响，但是它还是值得注意的，特别是在DLL首次被加载进入内存的时候。
此外，生成与位置无关的代码的编译器常常采用额外的间接层次去解析方法调用，以及访问DLL内部的全局变量。
过度的抖动。
时间关键的应用程序可能无法容忍链接DLL进入一个进程，以及动态解析方法地址的延迟。
因此，你应该凭经验来评估动态链接的影响，以便判定它对于你的应用程序是否真是一个问题。
2.2.3静态配置与动态配置 正如2.1节中描述的那样，网络应用程序常常提供或者采用各种各样的服务。
通过将在2.2.1以及2.2.2小节中描述的命名以及链接空间结合在一起，我们能够将网络应用程序服务分成被静态配置的和被动态配置的。
正如在2.2.2节中讨论的那样，在这一空间中主要的权衡包括可拓展性、安全性、可靠性，以及效率。
静态配置指的是与初始化一个包含静态命名服务（也就是说，将每一种服务开发成一个独立的函数或者类）的应用程序。

编辑推荐

《C++网络编程(卷2):基于ACE和框架的系统化复用》重点讲述ACE的各个框架，全面涵盖了作为其架构的基础概念、模式和使用规则。
《C++网络编程(卷2):基于ACE和框架的系统化复用》是设计面向对象框架的实用指南，同时也告诉开发者，怎样将框架应用到并发的网络应用中。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com