

图书基本信息

书名：<<Visual C++程序设计与应用教程题解及课程设计>>

13位ISBN编号：9787302202486

10位ISBN编号：7302202486

出版时间：2009-7

出版时间：清华大学出版社

作者：马石安，魏文平 编著

页数：231

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>

前言

网络层。

传感器网络的网络层负责路由发现和维护，通常大多数结点无法直接与网关通信，需要通过中间结点以多跳路由的方式将数据传送至汇聚结点。

传输层。

传感器网络的传输层负责数据流的传输控制，主要通过汇聚结点采集传感器网络内的数据，并使用卫星、移动通信网络、因特网或者其他的链路与外部网络通信，是保证通信服务质量的重要部分。

(2) 网络管理平台 网络管理平台主要是对传感器结点自身的管理和用户对传感器网络的管理，包括拓扑控制、服务质量管理、能量管理、安全管理、移动管理、网络管理等。

网络管理平台主要包括如下内容： 拓扑控制。

一些传感器结点为了节约能量会在某些时刻进入休眠状态，这导致网络的拓扑结构不断变化，因而需要通过拓扑控制技术管理各结点状态的转换，使网络保持畅通，数据能够有效传输。

拓扑控制利用链路层、路由层完成拓扑生成，反过来又为它们提供基础信息支持，优化MAC协议和路由协议，降低能耗。

服务质量管理。

服务质量管理在各协议层设计队列管理、优先级机制或者带宽预留等机制，并对特定应用的数据给予特别处理。

它是网络与用户之间以及网络上互相通信的用户之间关于信息传输与共享的质量约定。

为了满足用户的要求，传感器网络必须能够为用户提供足够的资源，以用户可接受的性能指标工作。

能量管理。

在传感器网络中电源能量是各个结点最宝贵的资源。

为了使传感器网络的使用时间尽可能长，需要合理、有效地控制结点对能量的使用。

每个协议层次中都要增加能量控制代码，并提供给操作系统进行能量分配决策。

安全管理。

由于结点随机部署、网络拓扑的动态性和无线信道的不稳定，传统的安全机制无法在传感器网络中适用，因而需要设计新型的传感器网络安全机制，采用诸如扩频通信、接入认证/鉴权、数字水印和数据加密等技术。

移动管理。

在某些传感器网络的应用环境中，结点可以移动，移动管理用来监测和控制结点的移动，维护到汇聚结点的路由，还可以使传感器结点跟踪它的邻居。

网络管理。

网络管理是对传感器网络上的设备和传输系统进行有效监视、控制、诊断和测试所采用的技术和方法。

它要求协议各层嵌入各种信息接口，并定时收集协议运行状态和流量信息，协调控制网络中各个协议组件的运行。

内容概要

本书是作者编著的《Visual C++程序设计与应用教程》一书的配套辅助教材，其内容包括两部分。

第1部分是《Visual C++程序设计与应用教程》一书各章中的全部习题及参考解答，共分为12章，分别与主教材对应，全面具体地对教材中各章的习题进行了必要的分析和详细解答，对操作题给出了详细实现步骤、代码清单及其填写位置，填写的代码语句都有注释说明和分析，力求通过实例让读者全面掌握Visual C++程序设计的思路 and 开发技巧。

第2部分是“Visual C++课程设计实例”，共分为5章，第1章介绍了课程设计的目标及要求，第2~5章介绍了课程设计实例，各个实例相对独立，覆盖了MFC编程的绝大部分内容，深入浅出地说明了MFC编程中最典型和最有用处的程序设计方法。

本书实例的应用性极强，实例全部可以运行，读者可以根据这些实例进行研究、修改和扩展，使其符合自己的要求，是主教材的有益延伸和拓展。

本书可作为高等院校计算机及相关专业学习Visual C++程序设计的辅助教材，也可供自学者或教师参考。

书籍目录

第一部分 习题及上机操作题参考解答	第1章 Visual C++开发环境	第2章 MFC应用程序概述
第3章 图形与文本	第4章 菜单、工具栏和状态栏	第5章 对话框
第6章 Windows常用控件	第7章 文档与视图	第8章 打印编程
第9章 动态链接库编程	第10章 多线程编程	第11章 数据库编程
第12章 多媒体编程	第二部分 课程设计实例	第1章 课程设计说明
1.1 课程设计目的	1.2 课程设计步骤	1.3 课程设计技术要求
1.4 课程设计报告	1.5 考核方式	1.6 评价标准
第2章 小型CAD系统	2.1 功能描述	2.2 系统分析与设计
2.2.1 图元的数据结构设计	2.2.2 图元对象在系统中的存储方法	2.2.3 系统中数据的组织策略
2.2.4 文档/视图结构对图形对象的操作方法	2.3 系统详细设计	2.3.1 项目创建
2.3.2 界面设计	2.3.3 视图设计	2.3.4 图形的绘制
2.3.5 图形对象数据结构设计	2.3.6 文档设计	2.3.7 图形的选取
2.3.8 图形的编辑	2.4 小结	第3章 五子棋游戏的开发
3.1 功能描述	3.2 设计思路	3.2.1 总体思路
3.2.2 计算机的决策方式	3.3 系统详细设计	3.3.1 项目创建
3.3.2 窗体设计	3.3.3 光标及音效设计	3.3.4 核心程序设计
3.3.5 游戏启动封面的设计	3.4 小结	第4章 学生个人事务管理系统
4.1 功能描述	4.2 系统分析与设计	4.2.1 系统分析
4.2.2 数据库设计	4.3 系统详细设计	4.3.1 项目创建
4.3.2 主框架的设计	4.3.3 数据库操作的实现	4.3.4 左树视图中事务项的添加
4.3.5 中间及右边视图详细设计	4.3.6 视图切换的实现	4.3.7 记录的添加与删除
4.3.8 快捷菜单的实现	4.3.9 其他设计	4.4 小结
第5章 OpenGL图形程序的开发	5.1 OpenGL基础	5.1.1 什么是OpenGL
5.1.2 OpenGL的发展历史	5.1.3 OpenGL的特点	5.1.4 OpenGL开发组件
5.1.5 OpenGL常量和函数	5.1.6 OpenGL提供的基本操作	5.1.7 坐标变换
5.1.8 在OpenGL中使用颜色	5.1.9 光照和材质	5.1.10 三维动画程序设计
5.2 程序功能描述	5.3 程序的基本结构设计	5.4 程序详细设计
5.4.1 项目创建	5.4.2 界面设计	5.4.3 代码实现
5.5 小结	参考文献	

章节摘录

(2) MFC中线程有哪两种类型？

它们有何区别？

如何创建它们？

【问题解答】MFC中有两类线程，分别称之为作者线程和用户界面线程。创建一个工作者线程，首先需要编写一个希望与应用程序的其余部分并行运行的线程函数。

然后，在程序中合适的地方调用全局函数AfxBeginThread()创建线程，以启动线程函数。

创建用户界面线程，首先需要从CwinThread类派生一个新类，并重写派生类的InitInstance、ExitInstance及Run等函数，然后使用AfxBeginThread函数的另一个版本创建并启动用户界面线程。

(3) 什么是线程函数？

其作用是什么？

如何给线程函数传递参数？

【问题解答】线程函数是新线程创建后要执行的函数，新线程要实现的功能是由线程函数实现的。

线程函数带有一个参数LPVOID pParam，创建线程的AfxBeginThread全局函数的第二个参数LPVOID pParam的类型与线程函数的参数类型完全一致，该参数为启动线程时传递给线程函数的入口参数。

(4) 如何终止线程？

【问题解答】当一个工作者线程的线程函数执行一个返回语句或者调用AfxEndThread成员函数时，这个工作者线程就终止。

对于用户界面线程，当一个WM—QUIT消息发送到它的消息队列中，或者该线程中的一个函数调用AfxEndThread成员函数时，该线程就被终止。

一般来说，线程只能自我终止。

如果要从另一个线程来终止线程，必须在这两个线程之间设置通信方式。

(5) 如何使用自定义消息进行通信？

【问题解答】使用windows消息进行通信，首先需要定义一个自定义消息，然后，需要时在一个线程中调用全局函数：：PostMessage()向另一个线程发送自定义消息。

(6) 什么叫线程的同步？

为什么需要同步？

【问题解答】使隶属于同一进程的各线程协调一致地工作称为线程的同步。

在多线程的环境里，需要对线程进行同步。

这是因为在多线程处理时线程之间经常要同时访问一些资源，这有可能导致错误。

例如，对于像磁盘驱动器这样独占性系统资源，由于线程可以执行进程的任何代码段，且线程的运行是由系统调度自动完成的，具有一定的不确定性，因此就有可能出现两个线程同时对磁盘驱动器进行操作，从而出现操作错误。

又例如，对于银行系统的计算机来说，可能使用一个线程来更新其用户数据库，而用另外一个线程来读取数据库以响应储户的需要，极有可能读数据库的线程读取的是未完全更新的数据库，因为可能在读的时候只有一部分数据被更新过。

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>