

精通

Windows Sockets

网络开发 — 基于

Visual C++ 实现

孙海民 编著

全面讲解Windows Sockets网络开发的相关技术
按照原理、实例程序和开发实例的模式进行讲解
覆盖Windows Sockets的2种模式和5种I/O模型
详细介绍7个C/S模式的网络开发实例



人民邮电出版社
POSTS & TELECOM PRESS

精通

TP393.09/180

2008

Windows Sockets

网络开发 — 基于 Visual C++ 实现

孙海民 编著

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

精通 Windows Sockets 网络开发: 基于 Visual C++ 实现 /
孙海民编著. —北京: 人民邮电出版社, 2008.7
ISBN 978-7-115-17911-1

I. 精… II. 孙… III. ①窗口软件—网络软件—程序设计
②C 语言—程序设计 IV. TP316 TP312

中国版本图书馆 CIP 数据核字 (2008) 第 046915 号

内 容 提 要

Windows Sockets 是当前主要的网络开发技术之一。本书由浅入深、循序渐进地讲解如何使用 Windows Sockets 开发网络应用程序。全书内容包括准备开发环境、TCP/IP 基本介绍、Windows 套接字基础、协议特征、基本 TCP 套接字编程、基本 UDP 套接字编程、套接字选项、套接字阻塞模式开发、套接字非阻塞模式开发、Select 模型开发、WSAAsyncSelect 模型开发、WSAEventSelect 模型开发、重叠 I/O 模型开发和完成端口模型开发。为使读者深入理解套接字的理论知识, 增加实践项目经验, 本书最后 7 章, 每章讲解一个具有实践意义的网络应用程序。在这些实例中包含许多软件开发技术知识, 如 STL、ADO、多线程、MFC、软件设计模式等。

本书适合广大的网络应用程序设计初学者、Windows 网络应用程序开发人员、VC++ 开发工程师、网络游戏开发人员、软件培训机构学员和高等学校的学生阅读。

精通 Windows Sockets 网络开发——基于 Visual C++ 实现

- ◆ 编 著 孙海民
责任编辑 屈艳莲
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 29.75
字数: 727 千字
印数: 1—4 000 册
- 2008 年 7 月第 1 版
2008 年 7 月北京第 1 次印刷

ISBN 978-7-115-17911-1/TP

定价: 55.00 元

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

前 言

当前网络技术迅猛发展，互联网走进千家万户，各种各样的网络软件应运而生。由于 Windows 平台被使用得最为广泛，所以 Windows 网络编程技术备受开发人员关注。从技术角度来看，Windows 网络编程技术可以分为传统技术和 Windows Sockets 网络编程技术两大类。

传统网络编程技术包括 NetBIOS (Network Basic Input/Output System)、重定向器、邮槽和命名管道。这些技术与 Windows Sockets 网络编程技术相比，具有独到之处，但往往只适合应用在一些特殊环境中。

而 Windows Sockets 网络编程技术由于其通用性，广泛地被应用在互联网软件的开发上，例如文件传输系统、电子邮件系统、网络电视、网络游戏、P2P 软件、网络聊天工具等。

Windows Sockets 网络编程技术为开发人员提供一套简单的网络应用程序接口，以其开放性、支持多种协议的特点，在 Windows 下的各种网络编程技术中脱颖而出。经过多年的不断完善和版本升级，在 Intel、Microsoft、Sun 等公司的支持下，目前已成为 Windows 网络编程事实上的标准。

目前，越来越多的软件人员从事网络软件的开发。事实上，能否熟练运用 Windows Sockets API 进行网络软件开发，已经成为应聘 Windows 应用开发工程师时的一项必备条件。最近，笔者见到一则新闻，百度公司研发 IM（即时通信软件）有了最新消息和确切进展，百度公司在招聘页面中发布了关于招聘即时通信产品经理和研发人才的招聘信息，如图 1 所示。

开发 Windows Sockets 应用程序，开发人员需要掌握 TCP/IP 协议族、协议特征、IP 地址、套接字模式、套接字 I/O 模型和套接字选择项等知识。为帮助读者快速掌握具有实用价值的网络开发技术，作者根据学习者的需求，从 Windows Sockets 实际的应用出发，并结合项目经验来安排全书的内容。学习完本书，读者能够掌握从事网络软件开发所应该具备的重要理论知识，提高软件开发技能。

本书特点

1. 内容翔实，重点突出

以开发实用 Windows Sockets 应用程序为目标，选择学习内容。重点讲解 Windows Sockets

阻塞模式和 5 种 I/O 模型。同时对涉及到的周边知识也进行了讲解和介绍。例如 TCP/IP 协议、多线程技术等。

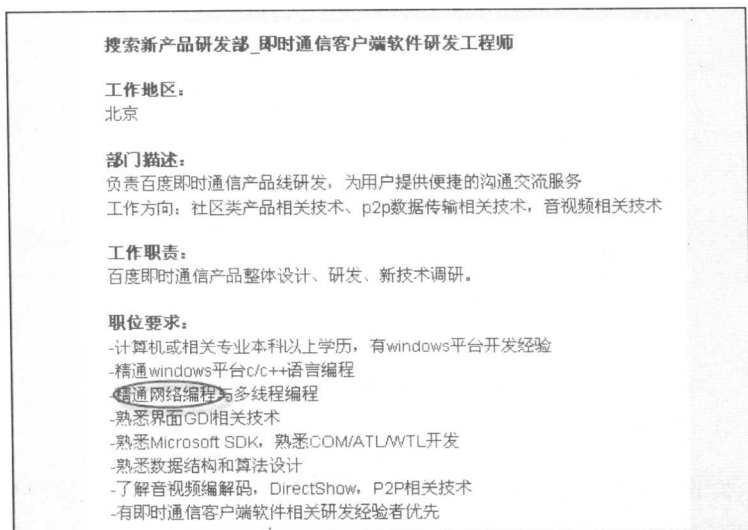


图 1

2. 精选实例, 深入剖析

当前 Windows Sockets 广泛地应用于互联网软件的开发。网络游戏、P2P 软件、网络聊天工具等软件的开发都离不开 Windows Sockets 技术。本书在实例选择上力求覆盖这些领域, 实例包括了网络聊天系统、网络五子棋游戏、考试系统、网络字典等, 这些实例基本上都是按照相应软件的业务逻辑进行讲解的。

3. 以点带面, 拓宽知识

开发网络应用程序时涉及到许多其他技术。本书能帮助读者以点带面地掌握软件开发技术, 例如 MFC 类库, ADO 访问数据库技术。开发实例中包含了许多编程技巧和经验。读者可以将其作为今后从事开发网络程序的样例。相信读者通过对本书的学习, 不仅能掌握了网络开发技术, 而且能拓宽软件开发的知识面。

4. 注重实训, 讲解清晰

第 4 章~第 10 章基本上按照这样的步骤编排内容: 相关原理、实例程序、开发实例。实例程序中, 重点突出利用所讲原理开发应用程序的框架。开发实例都是具有实际意义的应用程序。第 5 章~第 10 章的实例程序的代码都具有一定的规模。书中对于原理性的内容和复杂的软件程序设计都配有插图, 在讲解 Windows Sockets API 时, 配有实例代码。

5. 循序渐进, 由浅入深

首先介绍如何准备开发环境, 讲解创建 Windows 应用程序的方法以及调试两个工程的技术。接着讲解 Windows Sockets 的基础知识。然后, 给出了简单的 TCP 和 UDP 应用程序实例,

使读者对开发 Windows Sockets 应用程序有一个整体感知和思维框架。最后讲解套接字的工作模式和 I/O 模型。

本书内容

第 1 章 本章介绍如何检查网络连接状态, 如何利用 Microsoft Visual Studio 6.0 开发工具创建控制台和 MFC 类型工程, 以及同时调试两个 Visual C++ 工程的技巧。本章还讲解了开发 Windows Sockets 应用程序时的文件配置方法。

第 2 章 本章介绍了开放系统互连参考模型, 以及与开发 Windows Sockets 应用程序密切相关的 TCP/IP 协议族。

第 3 章 本章讲解开发 Windows Sockets 应用程序所必备的基础知识。内容包括套接字的概念、协议特征、IP 地址、基本 TCP 套接字编程、基本 UDP 套接字编程。

第 4 章 本章详细讲解了套接字的阻塞模式, 及如何运用该模式开发客户端与服务器相互问候程序。

第 5 章 本章详细讲解了套接字的非阻塞模式, 及如何运用该模式开发的远程算数运算程序。本章实例的客户端和服务器程序中充分利用了多线程技术。

第 6 章 本章详细讲解了套接字的 Select 模型, 及如何运用该模型开发远程文件下载程序。Select 模型是比较常用的一种套接字 I/O 模型。

第 7 章 本章详细讲解了套接字 WSAAsyncSelect 模型, 及如何利用 MFC 的 CSocket 类开发简易局域网聊天程序。本章还详细分析了 CSocket 类的实现, 归纳了利用该类开发 Windows Sockets 应用程序的编程模型, 并且就如何满足 MFC 的序列化条件进行了详细的讲解。

第 8 章 本章详细讲解了套接字 WSAEventSelect 模型, 及如何运用该模型开发的网络五子棋程序。

第 9 章 本章详细讲解了套接字重叠 I/O 模型, 及如何运用该模型开发网络字典程序。本章还在服务器程序中详细讲解了运用 ADO 访问数据库的技术。

第 10 章 本章详细讲解了套接字完成端口模型, 以及如何运用该模型开发计算机文化基础课考试系统。

学习方法

本书专门讲解如何基于 TCP/IP 协议族开发 Windows Sockets 应用程序。读者在学习本书之前应具备一定的 C/C++ 程序设计基础, 如 Windows SDK 的架构、MFC 和 STL 等相关内容。

本书在内容安排上, 由浅入深, 循序渐进。从准备开发环境配置, 到理解 TCP/IP 协议, 再到 Windows Sockets 的基础知识, 最后是套接字的工作模式和 I/O 模型。

在学习方法上, 可以参考如下步骤。

- (1) 学习原理。
- (2) 理解实例。
- (3) 独立编写一个实例。
- (4) 重温原理。

只有在实际开发中才能理解掌握所学知识。一个看似简单的原理或者知识点, 在实际应

用中可能会遇到预想不到的问题。解决这些问题的过程,会促使开发人员深入思考一些问题,从而加深对所学原理的掌握。对于软件开发人员来说,实际操作能力是非常重要的。当读者独立开发了一个实例之后,再重温所应用的原理,可以从一个更高的层次理解这些知识。

适合读者

- 网络应用程序设计的初学者。
- Windows 网络应用程序开发工程师。
- C++/Visual C++开发工程师。
- 网络游戏开发人员。
- 软件培训机构学员。
- 高等学校学生。
- 编程爱好者。

关于作者

本书由孙海明主持编写,同时参与编写和代码调试的还有陈杰、何世晓、何颖、卢顺科、吴恒奎、宋智广、宋燕、宋翔、寇苏朋、尚兴隆、张立全、戎剑、陈冠军、李东博、李丹、李春萌、杨涵、杨选举、梁其学、梁艺娟、苏鹏、陈励、鞠明君、黄志,在此一并表示感谢。

由于本书编写时间仓促,疏漏之处在所难免,请广大读者批评指正。

编者

2008年4月

目 录

第 1 篇 网络开发基础篇

第 1 章 准备开发环境.....3	1.4.2 将一个工程加入到另一个工程 空间..... 13
1.1 Windows Sockets 开发概述.....3	1.5 配置开发环境..... 14
1.1.1 网络程序开发应用.....3	1.6 小结..... 15
1.1.2 网络程序结构——C/S、B/S.....3	第 2 章 TCP/IP 简介..... 16
1.1.3 网络程序通信基础——网络 协议.....4	2.1 开放系统互连参考模型..... 16
1.1.4 网络程序通信技术——Windows Sockets 介绍.....4	2.2 TCP/IP 协议概述..... 17
1.2 连接网络.....6	2.2.1 TCP/IP 模型..... 17
1.2.1 TCP/IP 设置.....6	2.2.2 UDP..... 18
1.2.2 TCP/IP 是否工作正常.....8	2.2.3 TCP..... 18
1.2.3 系统与网络适配器间的通信.....8	2.2.4 端口..... 19
1.2.4 默认网关.....8	2.3 小结..... 19
1.2.5 ping 其他计算机 IP 地址.....9	第 3 章 Windows Sockets 基础..... 20
1.3 创建应用程序.....9	3.1 Windows Sockets..... 20
1.3.1 控制台程序.....10	3.1.1 应用程序与 Windows Sockets 的关系..... 20
1.3.2 MFC 应用程序.....11	3.1.2 套接字..... 21
1.4 调试两个应用程序.....12	3.2 协议特征..... 22
1.4.1 启动两个工程.....13	3.2.1 面向连接与面向无连接..... 22

3.2.2 可靠性与次序性	22	3.4.8 closesocket()函数	31
3.2.3 面向消息	22	3.4.9 shutdown()函数	31
3.2.4 部分消息	23	3.4.10 connect()函数	32
3.2.5 从容关闭	23	3.5 TCP 示例程序	32
3.2.6 路由选择	24	3.5.1 服务器实现	33
3.2.7 广播数据	24	3.5.2 客户端实现	35
3.3 IP 定址	24	3.6 基本 UDP 套接字编程	37
3.3.1 IP 定址	25	3.6.1 recvfrom()函数	37
3.3.2 字节顺序问题	25	3.6.2 sendto()函数	37
3.4 基本 TCP 套接字编程	26	3.7 UDP 示例程序	38
3.4.1 WSASStartup()函数	26	3.7.1 服务器实现	38
3.4.2 socket()函数	28	3.7.2 客户端实现	40
3.4.3 bind()函数	28	3.8 套接字选项	41
3.4.4 listen()函数	29	3.8.1 getsockopt()函数	41
3.4.5 accept()函数	29	3.8.2 setsockopt()函数	42
3.4.6 recv()函数	30	3.8.3 SOL_SOCKET 选项级别	43
3.4.7 send()函数	31	3.9 小结	44

第 2 篇 Visual C++网络模式开发篇

第 4 章 阻塞模式开发	47	4.4.6 接收数据	56
4.1 套接字的阻塞模式	47	4.4.7 发送数据	58
4.1.1 阻塞模式	47	4.4.8 退出	59
4.1.2 设置套接字的阻塞模式	49	4.5 客户端设计	59
4.1.3 阻塞模式套接字的优势和不足	49	4.5.1 启动	59
4.2 客户端与服务器相互问候	50	4.5.2 连接服务器	60
4.3 服务器设计	50	4.5.3 接收数据	60
4.3.1 启动	50	4.5.4 退出	60
4.3.2 接受请求	50	4.6 客户端实现	60
4.3.3 接收数据	51	4.6.1 初始化	61
4.3.4 退出	51	4.6.2 创建套接字	61
4.3.5 错误提示	51	4.6.3 连接服务器	62
4.4 服务器实现	51	4.6.4 发送数据	63
4.4.1 初始化	52	4.6.5 接收数据	63
4.4.2 创建套接字	54	4.6.6 退出	64
4.4.3 绑定套接字	54	4.7 小结	64
4.4.4 监听	56	第 5 章 非阻塞模式开发	65
4.4.5 接受请求	56	5.1 套接字的非阻塞模式	65
		5.1.1 非阻塞模式	65

第7章 WSAAsyncSelect 模型开发 ... 161

7.3.9 消息循环	170	7.10 客户端设计	218
7.3.10 程序退出	170	7.10.1 界面设计	218
7.3.11 窗口例程	170	7.10.2 通信层架构设计	220
7.3.12 CClient 类	171	7.10.3 显示的消息格式设计	220
7.3.13 管理客户端套接字的链表	171	7.11 客户端实现	220
7.3.14 网络事件消息处理函数	172	7.11.1 静态类图	220
7.4 调用 WSAAsyncSelect() 函数注意 问题	173	7.11.2 初始化	223
7.4.1 接收不到网络事件	173	7.11.3 接收服务器数据	227
7.4.2 关于 accept() 函数	173	7.11.4 显示离线消息	228
7.4.3 关于 FD_READ 网络事件	174	7.11.5 创建聊天窗口	229
7.4.4 如何判断套接字已经关闭	174	7.11.6 在线聊天	231
7.4.5 发送数据失败	174	7.11.7 发送离线消息	234
7.5 发生网络事件的条件	174	7.11.8 被动聊天	234
7.5.1 FD_READ 事件	174	7.11.9 保存聊天记录	235
7.5.2 FD_WRITE 事件	174	7.12 小结	236
7.5.3 FD_ACCEPT 事件	175		
7.5.4 FD_CONNECT 事件	175	第 8 章 WSAEventSelect 模型开发	238
7.5.5 FD_CLOSE 事件	175	8.1 套接字 WSAEventSelect 模型	238
7.6 WSAAsyncSelect 模型优势和不足	175	8.1.1 WSAEventSelect 模型	238
7.7 局域网简易聊天程序	176	8.1.2 与其他模型比较	238
7.7.1 需求分析	176	8.2 套接字 WSAEventSelect 模型实现	239
7.7.2 用户状态设计	177	8.2.1 WSAEventSelect() 函数	239
7.7.3 数据包类设计	179	8.2.2 WSACreateEvent() 函数	240
7.8 服务器设计	180	8.2.3 WSAResetEvent() 函数	241
7.8.1 界面设计	180	8.2.4 WSACloseEvent() 函数	241
7.8.2 通信层架构设计	182	8.2.5 WSAWaitForMultipleEvents() 函数	241
7.9 服务器实现	182	8.2.6 WSAEnumNetworkEvents() 函数	242
7.9.1 开发前的准备	182	8.3 WSAEventSelect 模型示例程序	244
7.9.2 静态类图	183	8.3.1 CClient 类	245
7.9.3 启动服务器	186	8.3.2 管理客户端套接字的链表	245
7.9.4 初始化	191	8.3.3 定义事件对象数组等变量	246
7.9.5 接受客户端请求	194	8.3.4 创建套接字	246
7.9.6 接收数据	196	8.3.5 为监听套接字注册网络事件	246
7.9.7 处理数据	203	8.3.6 开始监听	247
7.9.8 删除用户	210	8.3.7 等待网络事件	247
7.9.9 用户离线	212		
7.9.10 退出	213		

8.3.8 获取发生的网络事件	247	8.9 客户端设计	279
8.3.9 FD_ACCEPT 网络事件	248	8.9.1 多线程设计	279
8.3.10 FD_READ 网络事件	249	8.9.2 界面设计	279
8.3.11 FD_WRITE 网络事件	249	8.10 客户端实现	281
8.3.12 FD_CLOSE 网络事件	249	8.10.1 静态类图	282
8.3.13 退出	250	8.10.2 创建切分窗口	287
8.4 调用 WSAEventSelect()函数注意 的问题	250	8.10.3 初始化	289
8.4.1 多次调用 WSAEventSelect() 函数	250	8.10.4 工作线程	294
8.4.2 一个套接字不要关联多个事件 对象	250	8.10.5 与服务器通信	295
8.4.3 取消为套接字注册的网络 事件	251	8.10.6 更新用户列表	299
8.4.4 关于 accept()函数	251	8.10.7 开始游戏	301
8.4.5 关于 FD_READ 网络事件	251	8.10.8 下棋	306
8.4.6 关于 FD_WRITE 网络事件	251	8.10.9 结束游戏	317
8.4.7 关于 FD_CLOSE 网络事件	251	8.10.10 退出	318
8.5 WSAEventSelect 模型的优势和 不足	251	8.11 小结	319
8.6 网络五子棋游戏	252	第9章 重叠 I/O 模型开发	320
8.6.1 需求分析	252	9.1 套接字重叠 I/O 模型	320
8.6.2 游戏规则	253	9.1.1 重叠 I/O 模型	320
8.6.3 用户状态设计	253	9.1.2 与其他模型的比较	321
8.6.4 数据包设计	253	9.2 套接字重叠 I/O 模型实现概述	322
8.7 服务器设计	254	9.3 Win32 重叠 I/O 相关技术	322
8.7.1 多线程设计	254	9.3.1 CreateFile()函数	322
8.7.2 界面设计	255	9.3.2 OVERLAPPED 结构	323
8.8 服务器实现	255	9.3.3 ReadFile()函数	323
8.8.1 静态类图	255	9.3.4 WriteFile()函数	324
8.8.2 初始化	260	9.3.5 GetOverlappedResult()函数	324
8.8.3 启动服务	262	9.3.6 示例程序	325
8.8.4 服务器运行	264	9.4 套接字重叠 I/O	327
8.8.5 接收客户端请求	266	9.4.1 创建套接字	327
8.8.6 接收数据	270	9.4.2 接收数据	327
8.8.7 更新用户列表	273	9.4.3 WSAOVERLAPPED 结构	328
8.8.8 发送用户列表	277	9.4.4 WSAWaitForMultipleEvents() 函数	329
8.8.9 客户端退出	278	9.4.5 WSAGetOverlappedResult() 函数	329
8.8.10 服务器退出	278	9.4.6 发送数据	329
		9.5 事件通知	330
		9.5.1 CClient 类	331

9.5.2 管理客户端的链表	332	9.11.3 连接服务器	381
9.5.3 声明变量和函数	333	9.11.4 工作线程	382
9.5.4 套接字的创建、绑定和监听	333	9.11.5 发送数据	385
9.5.5 创建服务线程	334	9.11.6 接收数据	388
9.5.6 接受客户端连接请求	334	9.11.7 显示单词含义	390
9.5.7 服务线程	335	9.11.8 退出	391
9.6 完成例程	336	9.12 小结	391
9.6.1 CClient 类	337	第 10 章 完成端口模型开发	392
9.6.2 管理客户端的链表	339	10.1 套接字完成端口模型	392
9.6.3 套接字的创建、绑定和监听	339	10.1.1 利用完成端口接收数据	392
9.6.4 接受客户端连接请求	339	10.1.2 完成端口模型	393
9.6.5 服务线程	340	10.1.3 与重叠 I/O 模型比较	394
9.7 实例开发——网络词典	340	10.2 完成端口模型实现	395
9.7.1 需求分析	341	10.2.1 创建完成端口对象	395
9.7.2 数据包设计	341	10.2.2 创建服务线程	396
9.7.3 数据库设计	342	10.2.3 套接字与完成端口关联	397
9.8 服务器设计	342	10.2.4 发起重叠 I/O 操作	398
9.8.1 多线程设计	342	10.2.5 等待重叠 I/O 操作结果	399
9.8.2 界面设计	343	10.2.6 取消异步操作	400
9.9 服务器实现	345	10.2.7 投递完成通知包	400
9.9.1 开发前的准备	346	10.3 完成端口示例程序	401
9.9.2 静态类图	347	10.3.1 声明单 I/O 操作数据结构	402
9.9.3 初始化	351	10.3.2 CClient 类	403
9.9.4 启动服务	358	10.3.3 管理客户端的链表	404
9.9.5 服务器运行	359	10.3.4 声明变量和函数	405
9.9.6 接收数据	361	10.3.5 创建完成端口	405
9.9.7 查询单词含义	364	10.3.6 创建服务线程	405
9.9.8 发送数据	366	10.3.7 套接字的创建、绑定和监听	405
9.9.9 查看未知单词	367	10.3.8 套接字与完成端口关联	406
9.9.10 数据库操作	370	10.3.9 发起重叠 I/O 操作	407
9.9.11 退出	375	10.3.10 等待 I/O 操作结果	407
9.10 客户端设计	376	10.4 考试系统	409
9.10.1 多线程设计	376	10.4.1 需求分析	409
9.10.2 界面设计	376	10.4.2 数据包设计	410
9.11 客户端实现	377	10.4.3 数据库设计	410
9.11.1 静态类图	377	10.5 服务器设计	411
9.11.2 启动客户端	380	10.5.1 多线程设计	411

10.5.2 界面设计	412	10.6.11 服务器退出	443
10.6 服务器实现	414	10.7 客户端设计	446
10.6.1 静态类图	414	10.8 客户端实现	447
10.6.2 初始化	419	10.8.1 静态类图	448
10.6.3 启动服务	427	10.8.2 初始化	450
10.6.4 接受客户端请求线程	430	10.8.3 连接服务器并接收试卷	450
10.6.5 服务线程	432	10.8.4 初始化试卷头	454
10.6.6 接收客户端数据	433	10.8.5 初始化试卷	455
10.6.7 验证学号	437	10.8.6 答卷	459
10.6.8 发送数据	438	10.8.7 发送心跳包	460
10.6.9 考生交卷	440	10.8.8 交卷	460
10.6.10 跟踪客户端连接状态	442	10.9 小结	462

精通 Windows Sockets 网络开发——基于 Visual C++ 实现

第 1 篇 网络开发基础篇

第 1 章 准备开发环境

第 2 章 TCP/IP 简介

第 3 章 Windows Sockets 基础

第 1 章

准备开发环境

本章讲解开发 Windows Sockets 应用程序的前期准备知识。首先简述了开发 Windows Sockets 应用程序的背景知识,然后讲解开发人员应该具备的一些技术。技术包括如何检查网络连接,确保主机能够正常访问网络中其他计算机;如何利用 Microsoft Visual Studio 6.0 开发工具创建两种类型工程;如何调试两个 Windows 应用程序;如何配置 Windows Sockets 应用程序的开发环境。

1.1 Windows Sockets 开发概述

本节简述开发 Windows 网络程序的背景知识,内容包括什么是网络程序开发、网络软件结构、网络协议、Windows Sockets 的历史和开发人员应该具备的技术。

1.1.1 网络程序开发应用

网络程序开发可以简单地被理解为:基于局域网或互联网,利用网络开发技术,开发能够运行在网络上的软件(系统)。例如,开发常用的 IM 工具 QQ,使用该网络程序用户可以和朋友在线聊天、视频等。不论朋友身处何方,只要双方的主机连接到互联网上就可以实现通信。在 QQ 上我们还可以建立或者加入 QQ 群。将兴趣相同的朋友聚集在一起,研究和讨论问题。

IM 工具仅仅是网络程序开发的一个应用而已。随着网络技术的迅速发展,各种各样的网络软件(系统)应运而生,例如文件传输系统、电子邮件系统、网络电视、网络游戏、P2P 软件、网络聊天工具等。特别是 P2P(peer-to-peer)技术,被普遍认为在加强网络上人与人之间交流、实现文件交换和分布计算等方面大有前途。

1.1.2 网络程序结构——C/S、B/S

虽然网络应用程序很多,但是软件结构基本类似。网络软件可以分为 C/S 和 B/S 两种结构。

C/S (Client/Server),即客户端与服务器结构。客户端和服务端分别承担不同的任务。Client 将用户的需求提交给 Server,再将 Server 返回的结果以一定的形式提供给用户。Server 的任务是接收 Client 提出的服务请求,进行相应的处理,并将结果返回给 Client。