第一图书网, tushu007.com

<<Windows网络编程>>

图书基本信息

书名：<<Windows网络编程>>

13位ISBN编号：9787115248541

10位ISBN编号：7115248540

出版时间：2011-4

出版时间：人民邮电出版社

作者：罗莉琴，詹祖桥　主编

页数：348

字数：586000

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

随着internet技术的应用和普及，人类社会已经进入了信息化的网络时代。
大多数应用程序都是运行在网络环境下，这就要求程序员能够在应用最广泛的windows操作系统上开发网络应用程序。
《windows网络编程》结合大量的实例，介绍了开发windows网络应用程序的必备知识，并完整地讲述了几个windows网络应用程序实例的开发过程。
这些实例包括局域网探测器、基于p2p技术的bt下载工具和基于winpcap技术的网络数据包捕获、过滤和分析工具等。

《windows网络编程》可以作为大学本科、大专及高职相关专业的教材，也可作为广大windows网络应用程序开发人员的参考资料。

书籍目录

第1篇　基础协议
　第1章　internet与网络通信模型概述
　　1.1　internet概述
　　1.1.1　internet的发展历史
　　1.1.2　internet的管理机构
　　1.1.3　国内internet网络建设的现状
　　1.2　网络通信模型和协议簇
　　1.2.1　osi参考模型
　　1.2.2　tcp/ip协议簇体系结构
　　习题
　第2章　tcp/ip协议簇及其应用
　　2.1　ip
　　2.1.1　ip基础
　　2.1.2　ip的关键机制
　　2.2　tcp和udp
　　2.2.1　tcp的网络功能
　　2.2.2　tcp段结构
　　2.2.3　tcp的基本工作流程
　　2.2.4　udp
　　2.3　其他常用协议
　　2.3.1　arp
　　2.3.2　icmp
　　2.3.3　telnet
　　2.3.4　ftp
　　2.3.5　smtp和pop3
　　习题
第3章　ip地址和子网规划
　　3.1　ip地址
　　3.1.1　ip地址的结构
　　3.1.2　ip地址的分类
　　3.1.3　特殊的ip地址
　　3.2　子网划分
　　3.2.1　子网
　　3.2.2　子网掩码
　　3.2.3　cidr表示法
　　3.2.4　单播、组播和广播地址
　　习题
第2篇　网络编程
　第4章　网络编程基础
　　4.1　网络编程相关的基本概念
　　4.1.1　网络编程与进程通信
　　4.1.2　internet中网间进程的标识
　　4.1.3　网络协议的特征
　　4.1.4　高效的用户数据报协议
　　4.1.5　可靠的传输控制协议

4.2 三类网络编程
4.2.1 基于tcp/ip协议栈的网络编程
4.2.2 基于www应用的网络编程
4.2.3 基于.net框架的web services网络编程
4.3 客户机/服务器交互模式
4.3.1 网络应用软件的地位和功能
4.3.2 客户机/服务器模式
4.3.3 客户机与服务器的特性
4.3.4 容易混淆的术语
4.3.5 客户机与服务器的通信过程
4.3.6 网络协议与c/s模式的关系
4.3.7 错综复杂的c/s交互
4.3.8 服务器如何同时为多个客户机服务
4.3.9 标识一个特定服务
4.4 p2p模式
4.4.1 p2p技术的兴起
4.4.2 p2p的定义和特征
4.4.3 p2p的发展
4.4.4 p2p的关键技术
4.4.5 p2p系统的应用与前景
习题
第5章 socket编程基础
5.1 socket网络编程接口的产生与发展
5.1.1 socket编程接口起源于unix操作系统
5.1.2 socket编程接口在windows和linux操作系统中得到继承和发展
5.2 socket的工作原理和基本概念
5.2.1 socket协议的工作原理
5.2.2 什么是socket
5.2.3 socket的服务方式和类型
5.3 winsock编程基础
5.3.1 构建winsock应用程序框架
5.3.2 ip地址的表示形式
5.4 面向连接的socket编程
5.4.1 面向连接的socket通信流程
5.4.2 socket()函数
5.4.3 bind()函数
5.4.4 listen()函数
5.4.5 accept()函数
5.4.6 recv()函数
5.4.7 send()函数
5.4.8 closesocket()函数
5.4.9 shutdown()函数
5.4.10 connect()函数
5.4.11 tcp socket服务器应用程序编程实例
5.4.12 tcp socket客户端应用程序编程实例
5.5 面向非连接的socket编程
5.5.1 面向非连接的socket通信流程

5.5.2 sendto()函数
5.5.3 recvfrom()函数
5.6 socket选项
5.6.1 调用getsockopt()函数获取socket选项
5.6.2 调用setsockopt()函数设置socket选项
习题
第6章 探测网络中的在线设备
6.1 获取本地计算机的网络信息
6.1.1 使用ipconfig命令获取本地网络信息
6.1.2 获取本地网络信息的开发接口ip helper api
6.1.3 获取本地网络适配器信息
6.1.4 获取本地主机名、域名和dns服务器信息
6.1.5 获取本地计算机网络接口的基本信息
6.1.6 获取本地计算机ip地址表
6.1.7 添加和删除ip地址
6.2 扫描子网中的地址
6.2.1 计算指定子网内包含的所有ip地址
6.2.2 实现ping的功能
6.2.3 扫描子网
习题
第7章 netbios网络编程技术
7.1 netbios协议及应用
7.1.1 netbios协议
7.1.2 使用nbtstat命令
7.2 netbios开发接口
7.2.1 netbios操作
7.2.2 ncb结构体
7.2.3 其他常用netbios结构体
7.2.4 netbios()函数
7.2.5 获取lana上的所有netbios名字
7.2.6 获取网络适配器上的mac地址
7.3 在程序中实现nbtstat命令的功能
7.3.1 本实例的工作原理
7.3.2 定义的结构体
7.3.3 为获取netbios信息而定义的函数
7.3.4 实现nbtstat命令功能的主函数
习题
第8章 高级socket编程技术
8.1 socket编程模型概述
8.2 阻塞与非阻塞模式socket编程
8.2.1 设置非阻塞模式socket
8.2.2 非阻塞模式服务器应用程序编程实例
8.2.3 非阻塞模式客户端应用程序编程实例
8.2.4 基于非阻塞模式的多线程服务器应用程序编程实例
8.3 基于select模型的socket编程
8.3.1 select()函数
8.3.2 基于select模型的服务器应用程序实例

8.4 基于wsaasyncselect模型的socket编程
8.4.1 wsaasyncselect()函数
8.4.2 创建窗口
8.4.3 窗口例程
8.4.4 基于wsaasyncselect模型的服务器编程
8.5 基于wsaeventselect模型的socket编程
8.5.1 wsaeventselect()函数
8.5.2 创建和管理事件对象
8.5.3 wsawaitformultipleevents()函数
8.5.4 wsaenumnetworkevents()函数
8.5.5 基于wsaeventselect模型的服务器编程
8.6 基于重叠i/o模型的socket编程
8.6.1 wsasocket()函数
8.6.2 调用wsasend()函数发送数据
8.6.3 调用wsarecv()函数接收数据
8.6.4 getoverlappedresult()函数
8.6.5 使用事件通知来管理重叠i/o操作
8.6.6 使用完成例程来管理重叠i/o操作
8.7 基于完成端口模型的socket编程
8.7.1 完成端口模型的工作原理
8.7.2 创建完成端口对象
8.7.3 等待重叠i/o的操作结果
8.7.4 基于完成端口模型的服务器应用程序实例
习题
第9章 基于winpcap技术的网络数据包捕获、过滤和分析技术
9.1 winpcap技术基础
9.1.1 winpcap的体系结构
9.1.2 nic驱动器和ndis
9.1.3 网络组包过滤(npf)模块
9.1.4 捕获数据包的原理和步骤
9.2 下载和安装winpcap开发包
9.2.1 下载winpcap
9.2.2 安装winpcap
9.2.3 源代码的目录结构
9.3 在visual c++中使用winpcap技术
9.3.1 环境配置
9.3.2 获取与网络适配器绑定的设备列表
9.3.3 获取网络适配器的高级属性信息
9.3.4 打开网络适配器并实现抓包功能
9.3.5 不使用事件处理器进行抓包
9.3.6 过滤数据包
9.3.7 分析数据包
习题
第3篇 实例应用
第10章 设计局域网探测器
10.1 局域网探测器的主要功能
10.2 基础模块设计

10.2.1 基础函数
10.2.2 本地主机类clocalhost
10.2.3 设备类cdevice
10.2.4 子网类csubnet
10.3 系统主界面设计
10.3.1 系统主界面中包含的控件
10.3.2 设计菜单项
10.4 加载和退出主界面
10.4.1 加载主界面的代码实现
10.4.2 在文本编辑框中输出描述信息
10.4.3 自动调整控件的大小
10.4.4 退出系统并保存自定义子网
10.5 管理子网
10.5.1 添加和编辑子网
10.5.2 删除子网
10.6 扫描指定的子网
10.6.1 设计执行扫描子网操作的对话框
10.6.2 启动子网扫描
10.7 检测子网的状态
10.7.1 设计检测子网状态的对话框
10.7.2 启动状态检测
第11章 设计基于p2p技术的bt下载工具
11.1 p2p技术的工作原理和应用
11.1.1 p2p技术的工作原理
11.1.2 p2p网络模型
11.1.3 bt下载
11.1.4 ftkernelapi兼容bt协议网络内核库
11.2 系统主界面设计
11.2.1 系统主界面中包含的控件
11.2.2 设计菜单项
11.2.3 设计工具栏
11.3 加载主窗口
11.3.1 加载主窗口的代码实现
11.3.2 在startcontext()函数中初始化ftkernelapi环境
11.3.3 在initnattunnel()函数中初始化穿透内网的操作
11.4 实现bt下载
11.4.1 打开种子文件
11.4.2 开始下载
11.4.3 停止下载
11.4.4 显示下载进度
11.4.5 删除文件
11.4.6 打开目录

章节摘录

版权页：插图：如果一个服务器存在多个副本，客户机是怎样与正确的副本进行交互的呢？
进一步说，收到的请求是怎样被传给正确的服务器副本的呢？
这个问题的答案在于传输协议用来标识服务器的方法。
前面说过，每个服务被赋予一个唯一的标识，每个来自客户端的请求包含了服务标识，这使得服务器端计算机的传输协议有可能将收到的请求匹配到正确的服务器。
在实际应用中，大多数传输协议给每个客户机也赋以一个唯一的标识，并要求客户机在提出请求时包含这个标识。
服务器端计算机上的传输协议软件使用客户机和服务器的标识来选择正确的服务器副本。
作为一个实例，下面来看一下TCP连接中所使用的标识。
TCP要求每个客户机选择一个没有被赋给任何服务的本地协议端口号。
当客户机发送一个TCP段时，它必须将它的本地协议端口号放入S01.IRCEPORT域中，将服务器的协议端口号放入DESTINATIONPORT域中。
在服务器端计算机上，TCP使用源协议端口号和目的协议端口号的组合（同时也用客户机和服务器IP地址）来标识特定的通信。
这样，信息可以从多个客户机到达同一个服务器而不引起问题。
TCP将每个收到的段传给处理该客户机的服务器副本。
总之，传输协议给每个客户机也给每个服务器赋予一个标识。
服务器端的计算机上的协议软件使用客户机标识和服务器标识的结合来选择正确的并发服务器的副本。

编辑推荐

体现作者多年的Windows网络编程开发经验　　讲解大量实用技巧，重点突出，便于灵活掌握　提供典型应用范例及其源代码，分析详细，实用性强

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