

<<Java2网络协议内幕>>

图书基本信息

书名：<<Java2网络协议内幕>>

13位ISBN编号：9787508411521

10位ISBN编号：7508411528

出版时间：2002-7

出版时间：中国水利水电

作者：(美)A1 Williams

页数：437

字数：622

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>

<<Java2网络协议内幕>>

内容概要

本书涵盖了Java程序设计中各个层次的网络编程，是使用Java进行网络编程的优秀的指导书。本书的主要内容有：Internet基础、基础套接字、Telnet、FTP和TFTP、SMTP、POP3、NNTP、HTTP、HTTPS等协议的网络程序设计。内容系统而且全面，概念清晰，易于理解，并且每章都给出了大量的实例及分析。可从中国水利水电出版社网站（www.waterpub.com.cn）上下载完整的程序清单。

本书适合于有一定Java基础的程序员和高级程序员作为编程指南，也适合对网络协议感兴趣的程序员阅读。

<<Java2网络协议内幕>>

作者简介

A1 Williams从Univac1106机开始就是一位资深程序员。

从那时起，他曾经作过Unix的编码人员、MS-DOS程序员和Windows开发人员，现在他发现自己最常用的是Java。

A1还是《Web技术》杂志中的Java和开发专栏作家，他还为其他几个杂志撰写专栏文章，包括《Dr.Dobb'Sourcebook》和《可

<<Java2网络协议内幕>>

书籍目录

译者序

致谢

作者简介

前言

第1章 Internet基础

1.1 深入介绍

1.1.1 Java怎么样

1.1.2 协议包

1.1.3 Internet地址

1.1.4 DNS：Internet上的电话簿

1.1.5 URL，URI，URN

1.1.6 层（Layers）

1.1.7 基本协议

1.1.8 网络硬件

1.1.9 协议的学习

1.2 快速解决方案

1.2.1 确定你的真IP地址

1.2.2 使用动态重定向

1.2.3 确定IP地址的类型

1.2.4 选择端口号

1.2.5 使用Ping和其他工具

1.2.6 自己动手练习一个协议

1.2.7 查找RFC

1.2.8 为代理服务器设置Java

第2章 Java网络编程

2.1 深入介绍

2.1.1 套接字编程的实质

2.1.2 I/O（输入/输出）流

2.1.3 高级套接字方法

2.1.4 线程

2.2 快速解决方案

2.2.1 解析主机名

2.2.2 向服务器端打开TCP套接字

2.2.3 打开服务器端套接字

2.2.4 创建UDP套接字

2.2.5 向TCP套接字发送数据

2.2.6 从TCP套接字接收数据

2.2.7 压缩套接字数据

2.2.8 设定套接字的最长读时间

2.2.9 设定服务器端最长接收时间

2.2.10 设定SoLinger

2.2.11 设定套接字的延时行为

2.2.12 设定保持活动选项

2.2.13 设定缓冲区的大小

2.2.14 处理套接字异常

<<Java2网络协议内幕>>

2.2.15 创建多线程服务器程序

2.2.16 自动处理多线程服务器

2.2.17 使用线程池为客户端程序服务

第3章 简单协议

3.1 深入介绍

3.1.1 Echo协议

3.1.2 Finger

3.1.3 Whois协议

3.1.4 基本时间协议 (BasicTime)

3.2 快速解决方案

3.2.1 使用Echo协议

3.2.2 编写TCP Echo服务程序

3.2.3 编写UDP Echo服务程序

3.2.4 合并TCP和UDP服务程序

3.2.5 使用Finger服务

3.2.6 编写Finger服务器

3.2.7 创建一个简单的代理

3.2.8 使用Whois

3.2.9 查询对人可读格式的时间

3.2.10 对NIST时间串进行解码

3.2.11 查询机器可读格式中的时间

3.2.12 编写时间服务程序

3.2.13 选用Unicode作字节映射

第4章 TFTP协议

4.1 深入介绍

4.1.1 关于TFTP协议

4.1.2 Play by Play

4.1.3 TFTP客户端应用

4.1.4 TFTP服务器端应用

4.1.5 更简单的一种方法

4.1.6 TFTP与FTP的对比

4.2 快速解决方案

4.2.1 探寻TFTP的规范

4.2.2 创建一个TFTP类

4.2.3 创建一个TFTP的客户端应用程序

4.2.4 创建一个TFTP的服务器端应用程序

4.2.5 使用GNU的TFTP类

4.2.6 配置GNU的TFTP服务器

第5章 Telnet协议

5.1 深入介绍

5.1.1 Telnet回顾

5.1.2 NVT回顾

5.1.3 特殊命令

5.1.4 要协商的地方

5.1.5 Telnet实践

5.1.6 一个基本的Java客户端

5.1.7 创建一个Telnet服务器端应用

<<Java2网络协议内幕>>

- 5.1.8 定制服务器端
- 5.1.9 Telnet开放源码
- 5.2 快速解决方案
 - 5.2.1 探寻Telnet协议规范
 - 5.2.2 发送Telnet命令同时发送数据
 - 5.2.3 模拟NVT
 - 5.2.4 协商Telnet的选项
 - 5.2.5 防止循环
 - 5.2.6 处理子选项
 - 5.2.7 从基类创建一个Telnet客户端
 - 5.2.8 从基类创建一个Telnet服务器端
 - 5.2.9 使用Telnetwapper
- 第6章 FTP协议
 - 6.1 深入介绍
 - 6.1.1 基础
 - 6.1.2 传输
 - 6.1.3 响应
 - 6.1.4 登录
 - 6.1.5 创建连接
 - 6.1.6 FTP命令细节
 - 6.1.7 考虑客户端
 - 6.1.8 考虑服务器端
 - 6.2 快速解决方案
 - 6.2.1 查找FTP规范
 - 6.2.2 连接到FTP服务器
 - 6.2.3 解释FTP的响应
 - 6.2.4 管理当前目录
 - 6.2.5 读文件目录
 - 6.2.6 传输文件
 - 6.2.7 选择主动方式还是被动方式
 - 6.2.8 使用FTP的开放源码
- 第7章 SMTP协议
 - 7.1 深入介绍
 - 7.1.1 验证
 - 7.1.2 超时、多行和透明性
 - 7.1.3 扩展的SMTP
 - 7.1.4 题头
 - 7.1.5 编码
 - 7.1.6 实现
 - 7.1.7 使用 SMTP
 - 7.1.8 附件
 - 7.1.9 SMTP的问题 (Twists)
 - 7.2 快速解决方案
 - 7.2.1 探寻SMTP规范
 - 7.2.2 连接一个SMTP服务器
 - 7.2.3 通过SMTP发送邮件
 - 7.2.4 解释响应码

<<Java2网络协议内幕>>

- 7.2.5 形成地址
- 7.2.6 选择题头
- 7.2.7 格式化消息文本
- 7.2.8 使用可引用可打印编码对消息文本编码
- 7.2.9 使用Base 64编码对消息文本编码
- 7.2.10 格式化多部分消息
- 7.2.11 使用MailMessage对象
- 7.2.12 使用SMTP对象
- 第8章 POP3协议
- 8.1 深入介绍
- 8.1.1 POP3协议
- 8.1.2 一个POP3类
- 8.1.3 一个常用列表管理器
- 8.1.4 代码
- 8.1.5 作用
- 8.1.6 进一步开发
- 8.1.7 关于IMAP
- 8.1.8 使用JavaMail
- 8.2 快速解决方案
- 8.2.1 探寻POP3协议规范
- 8.2.2 探寻IMAP规范
- 8.2.3 解释POP3服务器的响应
- 8.2.4 使用POP3授权
- 8.2.5 了解邮箱状态
- 8.2.6 确定消息细节
- 8.2.7 读一个邮件消息
- 8.2.8 删除一个消息
- 8.2.9 创建一个POP3客户端类
- 8.2.10 使用comjthomas.pop包
- 8.2.11 安装JavaMail
- 8.2.12 使用JavaMail Message对象工作
- 8.2.13 使用JavaMail Session对象工作
- 8.2.14 在POP邮件服务器中使用JavaMail
- 8.2.15 在IMAP邮件服务器中使用JavaMail
- 第9章 NNTP协议
- 9.1 深入介绍
- 9.1.1 关于 News
- 9.1.2 NNTP内幕
- 9.1.3 封装 NNTP
- 9.1.4 Web上的NNTP
- 9.2 快速解决方案
- 9.2.1 探寻NNTP规范
- 9.2.2 连接一个News服务器
- 9.2.3 选择一个组
- 9.2.4 列出所有的组
- 9.2.5 寻找新组
- 9.2.6 读取文章

<<Java2网络协议内幕>>

- 9.2.7 改变当前的文章
- 9.2.8 查找新文章
- 9.2.9 投递文章
- 9.2.10 使用NewsClient类
- 9.2.11 显示Web上的文章
- 9.2.12 读Web上的文章
- 9.2.13 通过Web投递文章
- 第10章 HTTP客户端
- 10.1 深入介绍
- 10.1.1 HTTP协议
- 10.1.2 状态码
- 10.1.3 常用标题
- 10.1.4 表单
- 10.1.5 Cookies
- 10.1.6 Applets (小程序)
- 10.2 快速解决方案
- 10.2.1 探寻HTTP协议规范
- 10.2.2 创建简单请求
- 10.2.3 创建1.0版的请求
- 10.2.4 创建1.1版的请求
- 10.2.5 读状态码
- 10.2.6 通过HTML向服务器发话表单数据
- 10.2.7 使用Java发话表单数据到服务器
- 10.2.8 URL数据编码
- 10.2.9 自动提交表单
- 10.2.10 发送和接收Cookies
- 10.2.11 打开浏览器到浏览器的通信
- 10.2.12 检查合法链接
- 第11章 协议操作者
- 11.1 深入介绍
- 11.1.1 URL内幕
- 11.1.2 URLConnection内幕
- 11.1.3 URLConnection子类
- 11.1.4 协议和内容操作者
- 11.2 快速解决方案
- 11.2.1 获取URL的数据
- 11.2.2 获取URL的内容
- 11.2.3 设置请求标题
- 11.2.4 读取响应标题
- 11.2.5 使用特定的HTTP连接
- 11.2.6 传送数据到服务器
- 11.2.7 打开一个JAR文件作为URL
- 11.2.8 创建一个客户协议操作者
- 11.2.9 安装一个客户协议操作者
- 11.2.10 创建一个客户内容操作者
- 11.2.11 安装一个客户内容操作者
- 第12章 解释HTML

<<Java2网络协议内幕>>

12.1 深入介绍

12.1.1 显示

12.1.2 处理HTML

12.1.3 实现Ad Hoc

12.1.4 Ad Hoc细节

12.1.5 使用AHParse

12.1.6 处理图像

12.1.7 属性解析

12.1.8 改进可用性

12.1.9 再次访问Swing

12.2 快速解决方案

12.2.1 通过Swing使用HTML

12.2.2 使用 JEditorPane显示 HTML

12.2.3 通过超链接显示HTML

12.2.4 使用 AHParse

12.2.5 通过Swing解析标签

12.2.6 通过Swing解析属性

12.2.7 通过Swing解析文本

第13章 HTML服务

13.1 深入介绍

13.1.1 关于 JSP

13.1.2 定制Tandem中的服务

13.1.3 通过代理创建的Web

13.1.4 拍卖服务器

13.2 快速解决方案

13.2.1 使用JSP进行服务器端编程

13.2.2 从JSP中读输出

13.2.3 在JSP中向浏览器写数据

13.2.4 使用JSP页的定向功能

13.2.5 写一个简单的Web服务器程序

13.2.6 配置HttpServer (Http服务器)

13.2.7 定制HttpServer

13.2.8 写一个代理服务器

13.2.9 调试一个代理服务器

第14章 XML

14.1 深入介绍

14.1.1 进入XML

14.1.2 XML语法

14.1.3 有效的XML

14.1.4 文档对象模型 (DOM)

14.1.5 名字空间

14.1.6 Java对XML的支持

14.1.7 XML库

14.1.8 使用 SAX

14.1.9 使用 DOM

14.2 快速解决方案

14.2.1 安装 Java XML扩展

<<Java2网络协议内幕>>

- 14.2.2 在 JSP里创建 XML
- 14.2.3 创建一个解析器
- 14.2.4 创建一个验证解析器
- 14.2.5 创建一个理解命名的解析器
- 14.2.6 使用一个SAX解析器
- 14.2.7 使用SAX进行验证
- 14.2.8 在XML文件里创建DOM
- 14.2.9 读属性
- 14.2.10 构造一个DOM
- 14.2.11 编写一个 DOM
- 第15章 安全性略谈
 - 15.1 深入介绍
 - 15.1.1 加密技术回顾
 - 15.1.2 Java安全性
 - 15.1.3 关于证书
 - 15.1.4 隐藏数据
 - 15.2 快速解决方案
 - 15.2.1 创建一个安全的套接字工厂
 - 15.2.2 创建一个安全套接字
 - 15.2.3 与一个安全的Web服务器相连接
 - 15.2.4 使用Steganography
 - 15.2.5 包含证书
 - 15.2.6 显示证书
 - 15.2.7 导入证书
 - 15.2.8 导出证书
- 附录A 一些有用的RFC
- 附录B 端口的分配

<<Java2网络协议内幕>>

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>