

<<深入浅出Linux TCP/IP协议栈>>

图书基本信息

书名：<<深入浅出Linux TCP/IP协议栈>>

13位ISBN编号：9787115216274

10位ISBN编号：7115216274

出版时间：2010-1

出版时间：人民邮电出版社

作者：罗钰

页数：342

字数：524000

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>

<<深入浅出Linux TCP/IP协议栈>>

前言

这是一本介绍协议栈实现源代码的书，不是介绍“协议”的书。

很多人对网络非常感兴趣，但却不理解内部是如何运作的，或知之不多，那么最终的结果就是行之不远。

本书尝试用浅显的语言和合理的安排带领读者到Linux内核网络模块的代码丛林中一游，既让你有所知，也让您有所思。

为什么选定Linux来介绍网络实现?首先，某些操作系统不是开源的，本人看不到也不能拿来解说；其次，Linux的市场占有率还是比BSD要高的；最后，Linux是经过成千上万用户使用过并且还将不断发展的。

目前Linux在服务器市场上已经证明其设计的精巧和健壮，特别是当内核也从非抢占式发展成为抢占式后，嵌入设备市场上也将要掀起一股风浪。

于是Linux内核分析的资料层出不穷，但有的太老(内核的代码还使用2.2的)，有的对网络部分的分析不甚详细，于是作者萌发了分析整个Linux网络协议栈的想法。

希望能在研究一些经典代码时发现与时俱进的部分，抛砖引玉，吸引更多的人参与到研究网络协议栈的实现技巧以及移植工作上，而不用对照枯燥的RFC文档和代码。

本人做过一些网络通信产品开发，所以对网络内部实现很感兴趣，于是在学习和工作时间之余记录下分析和调试Linux的网络协议栈的心得体会。

2005年开始计划写这本书，当时研究的内核版本是2.6.5，但是现在我使用RHEL5作为研究平台时，内核却升级到了2.6.18版本，协议栈的部分代码就已经发生了很多变化，最新的内核版本已经是2.6.26版本，再一次感叹计划赶不上变化的同时，还是锁定了这个版本，毕竟这是知名公司服务器使用的内核版本，有问题还可以到社区寻找援助。

本书的整体风格是比较随性的记录。

我的目标读者是广大对Linux和TCP/IP协议栈实现感兴趣的读者，而且要有一定研发经验和编程基础。

只对协议感兴趣的读者，本书是不太适合的，要看最好是去看RFC或者相关的手册。

<<深入浅出Linux TCP/IP协议栈>>

内容概要

本书主要对Linux 2.6.18内核协议栈源代码做了一些基本的分析，这些分析基于作者在操作系统方面的研究和网络协议开发过程中的经验和笔记，编写本书的目的主要是使读者能够在尽可能短的时间内掌握Linux内核协议栈的工作机理，为移植和扩展协议栈打下基础。

本书首先介绍了内核源码的整体概况及协议栈初始化过程，然后结合配置、用户使用协议栈的方法，采取深入浅出、由上及下的策略对协议栈的代码进行了分解和注释。

最后还介绍了通信界里较流行的VLAN技术和LACP协议。

本书适合Linux网络开发人员以及对Linux内核感兴趣的读者阅读。

<<深入浅出Linux TCP/IP协议栈>>

作者简介

罗钰，贵州人，国防科学技术大学硕士毕业，多年Linux底层开发经验，精通软件分析与设计、TCP / IP协议，曾开发Windows / Linux / Vxworks / FreeBSD等平台的设备驱动，开发过以太网芯片驱动、二层协议以及OSPF路由协议，对MPLS架构设计有丰富的经验，擅长编译器、CPU技术，近年来一直致力于无线网络产品系统的分析与设计工作。

<<深入浅出Linux TCP/IP协议栈>>

书籍目录

第1章 操作系统、网络、协议栈、代码	1.1 Linux操作系统介绍	1.1.1 Linux操作系统架构简介
1.1.2 网络协议发展介绍	1.2 本书的组织和安排	1.2.1 基本的数据结构和计算机术语
1.2.2 图片风格演示	1.2.3 本书的组织	第2章 内核系统初始化
2.1 系统初始化流程简介	2.2 内核文件解读	2.2.1 ELF文件格式
2.2.2 Link Scripts知识	2.2.3 Linux内核镜像解析	2.3 中断及任务调度管理
2.3.1 中断及软中断模型	2.3.2 各种语境下的切换	2.3.3 内核下的同步与互斥
2.3.4 各种异步手段	2.4 虚拟文件系统	2.5 网络协议栈各部分初始化
2.5.1 网络基础系统初始化	2.5.2 网络内存管理	2.5.3 网络文件系统初始化
2.5.4 网络协议初始化	2.5.5 初步了解路由系统	2.6 Linux设备管理
2.6.1 底层PCI模块的初始化	2.6.2 网络设备接口初始化例程	第3章 配置网络系统
3.1 配置过程分析	3.1.1 配置是如何下达到内核的?	3.1.2 socket系统调用
3.1.3 ioctl代码的实现	3.1.4 loopback接口的配置过程	3.1.5 IP别名的实现
3.2 回顾FIB系统初始化	3.3 深入FIB系统	3.4 FIB系统发生了什么样的变化
3.5 直接访问路由表	3.6 接口状态变化的处理过程	第4章 网络层实现的初步研究
4.1 从ping 127.0.0.1开始旅程	4.2 再次相遇Socket系统调用	4.3 IP数据报文格式
4.4 send系统调用	4.5 在路由系统中游历	4.5.1 查找出口
4.5.2 当目的地址是远端主机时	4.5.3 创建对应路由cache表项	4.5.4 创建对应邻居表项
4.6 回到发送的路径	4.6.1 IP层发送过程	4.6.2 揭密hh_cache
4.7 ARP的作用	4.7.1 ARP的机制	4.7.2 ARP报文格式
4.7.3 Linux ARP协议的实现	4.8 到达设备驱动层	4.8.1 数据链路层帧格式
4.8.2 Loopback设备的发送过程	4.9 接收过程：从中断到路由系统	4.10 ICMP
4.10.1 ICMP报文格式	4.10.2 ping本机地址及回环地址	4.10.3 ping外部地址
4.11 从内核到用户	第5章 传输层实现的研究	第6章 Select系统调用的实现机制
第7章 数据链路层协议实现	后记	参考文献

章节摘录

插图：因此RCU实际上是一种改进的rwlock，读操作几乎没有什么同步开销，它不需要锁，不使用原子指令，因此不会导致锁竞争、内存延迟以及流水线停滞。

不需要锁也使得使用更容易，因为死锁问题就不需要考虑了。

写操作的同步开销比较大，它需要延迟数据结构的释放，复制被修改的数据结构，它也必须使用某种锁机制同步并行的其他写操作的修改操作。

读操作必须提供一个信号给写操作，以便写操作能够确定数据可以被安全地释放或修改的时机。

有一个专门的垃圾收集器来探测读操作的信号，一旦所有的读操作都已经发送信号告知它们都不在使用被RCU保护的数据结构，垃圾收集器就调用回调函数完成最后的数据释放或修改操作。

RCU与rwlock的不同之处是：它既允许多个读操作同时访问被保护的数据，又允许多个读操作和多个写操作同时访问被保护的数据（注意：是否可以有多个写操作并行访问取决于写操作之间使用的同步机制），读操作没有任何同步开销，而写操作的同步开销则取决于使用的写操作间同步机制。

但RCU不能替代rwlock，因为如果写操作比较多时，对读操作的性能的提高不能弥补写操作导致的损失。

2.3.4各种异步手段1.Timer（定时器函数）作为一个有经验的开发者，一定知道所谓的定时器函数，其实都是一些回调函数而已，只不过本书中涉及的定时器都是在内核中执行，内核中的Timer不是线程，它们运行在中断级，所以timer函数不应该做任何精细的工作。

如果需要进一步处理，那么应该在tasklet里完成，因为tasklet可以被中断抢占。

创建Timer的方式有好几种，出于篇幅的原因，这里只举一个在协议栈经常使用的方式：（1）定义一个timer_list{}结构比如名叫atimer；（2）调用init_timer（&atimer）；（3）指定atimer.expires为执行周期，atimer.function为回调函数，timer.data为回调函数的参数；（4）执行add_timer（&atimer）；（5）当执行atimer.expires之后，执行回调，并且在回调函数中再执行（4），依次重复。

为什么要采用这种方式而非其他，是因为要保证这种rimer的优先级高于其他方式创建的Timer。

<<深入浅出Linux TCP/IP协议栈>>

后记

写完这本书，我只能感叹，时间真是靠挤出来的。

尽管《深入理解Linux网络内幕》已经面市，但我还是觉得我要继续完成这部分工作，一来了却自己的愿望，二来希望用自己的语言和见解来分析Linux的协议栈，使之更符合我国软件工程师的需要，我的目的就达到了。

阅读本书能大概了解Linux是如何实现TCP / IP协议栈的，但却不是能用来当枕头的“红宝书”，所以，读者必须结合协议的RFC文档来看本书。

本人将选择Linux和

FreeBSD的网络协议栈作为自己的业余研究对象。

所以本套书应该有两本，该本是Linux协议栈的剖析另一本还在计划之中。

为何选择这两者？

其一：代码是公开且免费的，不会有公司来找我麻烦，我也不需要采用极端的方式获取这些代码。

其二：这两种代码已成为网络应用之事实标准，Linux在当前大行其道，自不必说，而FreeBSD是BSD家族中应用最广泛者，其协议栈为VxWorks等工业级操作系统采用，而且是故去的Steven所著的《TCP-IP详解卷2实现》的代码基。

为投开发人员之所好，我决定研究这两个操作系统的网络协议栈。

本书就是我的第一本关于Linux方面的文档。

将来有时间我将写出FreeBSD的协议栈文档。

我们知道如果想研究BSD的协议栈可以看Gray R.Wright和W.Richard Stevens编写的《TCP / IP Illustrated Volume2 : The Implementation》（中文译名：《TCP / IP详解卷2：实现》），但是：FreeBSD已经进化到7.0版本了，其内部的结构已经有相当大的变化，光看那本书估计还不够。

而且，根据我的经验，作为高端嵌入式软件市场占有率最大的嵌入式操作系统VxWorks，其网络协议栈即从BSD发展而来，却对BSD协议栈进行了大量的修改，连代码风格都不一样。

但看这本书就更不能完全理解VxWorks的网络协议栈。

于是许多研究者在研究路由器操作系统的协议栈时，不知从何处研究起。

在本文完成之际，作者谨向所有给予我指导、关心、支持和帮助的老师、Leader、同事、同学和亲人致以衷心的感谢！

<<深入浅出Linux TCP/IP协议栈>>

编辑推荐

《深入浅出Linux TCP/IP协议栈》学Linux TCP/IP，从这里开始!从Linux 2.6内核基础讲起，引导读者快速入门运用大量示意图，清晰讲解函数之间的调用关系详细分析核心代码实现思想，代码注释详尽《深入浅出Linux TCP/IP协议栈》首先介绍了Linux内核源码的整体概况及协议栈初始化过程，然后结合配置、用户使用协议栈的方法，采取深入浅出、由上及下的策略对协议栈代码进行分解和注释，涵盖以下内容：操作系统、网络、协议栈、代码内核系统初始化配置网络系统网络层的实现传输层的实现Select系统调用的实现机制数据链路层协议实现

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>