<<Linux内核源码剖析（上下册）>>

图书基本信息

书名：<<Linux内核源码剖析（上下册）>>

13位ISBN编号：9787111323730

10位ISBN编号：7111323734

出版时间：2011-1

出版时间：机械工业出版社

作者：樊东东 莫澜

页数：1062

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>

<<Linux内核源码剖析（上下册）>>

前言

有人宣称Linux人才是未来20年IT职场中的王者。

无论这种说法有多夸张，有一个事实是不可否认的，那就是，近年来Linux的市场份额不断增长，Linux正在受到越来越多的关注乃至推崇。

由于Linux可广泛地应用到各种系统，包括很多嵌入式系统上，以及其他的诸多优点，如开放、高效、丰富的网络功能等，这种趋势在可预计的未来还将持续。

目前，国内对Linux各方面的研究工作没有国外那样广泛和深入，相关的出版物水准参差不齐。

特别是在网络的实现方面，有些著作针对性不强，有些则缺少了重要的传输层协议实现的论述，还有一些虽然有比较全面的介绍，却不够深入，选用的Linux版本也比较旧。

针对以上情况，本书选择了较新的2.6.20版本内核RCP / IP实现作详细的论述，在重要的细节处甚至逐行分析，并在此基础上，对代码背后的机制和原理作了深入的阐述，将各关键点连成一个整体，帮助读者理清整个Linux网络部分的脉络。

作者本着严谨的态度，在写作过程中参阅了大量的中英文资料及相关的文档。

相信本书可以成为那些希望深入了解Linux的TCP / IP协议栈、网络部分实现的人们的有力工具。

本书共有33章，通过自底向上的方法来论述TCP / IP的实现，从数据链路层开始，然后是网络层（IP、ICMP、IGMP、路由以及邻居子系统和IP组播），接下来是套接口，最后是传输层（TCP和UDP）

。在学习的时候也可以采用自顶向下的方法（从传输层开始向下），或者结合以上两种方法。要理解一个系统的运行机制，对于一个专业人员来说，代码是最为直接也最为可靠的资源。

Linux普及面不断扩展，越来越多的人会想通过研读内核代码来了解Linux系统，一来是更好地解决具体工作中的相关问题，二来是从Linux这个高质量的操作系统中学习到更多的编程、架构等技术。

本书的特点如下：选择的内核版本新，内容不会在短期内过时。

对代码作了详细的论述，在此基础上进一步分析了代码背后的机制和原理，为读者理清了整个框架的脉络，帮助读者避免迷失在细节中。

写作过程中参阅了大量中英文资料和相关的文档，对内核代码更是作了长时间深入细致的研究和分析，在细节处反复推敲，以确保本书的质量。

<<Linux内核源码剖析（上下册）>>

内容概要

本书详细论述了Linux内核2.6.20版本中TCP/IP的实现。

书中给出了大量的源代码，通过对源代码的详细注释，帮助读者掌握TCP/IP的实现。

本书根据协议栈层次，从驱动层逐步论述到传输层，包括驱动的实现、接口层的输入输出、IP层的输入输出以及IP选项的处理、邻居子系统、路由、套接口及传输层等内容，全书基本涵盖了网络体系架构全部的知识点。

特别是TCP，包括TCP连接的建立和终止、输入与输出，以及拥塞控制的实现。

本书适用于熟悉Linux的基本使用方法，对Linux内核工作原理以及网络知识有一定的了解，而又极想更深入理解各个机制在Linux中的具体实现的用户，包括应用程序员和嵌入式程序员，以及网络管理员等。

相关专业的科研人员在工作中遇到问题时，也可以查阅本书，理解相关内核部分的实现。

此外，计算机相关专业的本科高年级学生和研究生，在学习相关课程（如操作系统、计算机网络等）时，可将本书作为辅助教程，与理论相结合以便更好地理解相应的知识点。

<<Linux内核源码剖析 (上下册)>>

书籍目录

上册	前言	第1章 预备知识	1.1 应用层配置诊断工具	1.1.1 iputils	1.1.2 net-tools
	1.1.3 iproute2	1.2 内核空间与用户空间的接口	1.2.1 procfs	1.2.2 sysctl(/proc/sys目录)	
	1.2.3 sysfs(/sys文件系统)	1.2.4 ioctl系统调用	1.2.5 netlink套接口	1.3 网络I/O加速	
	1.3.1 TSO/GSO	1.3.2 I/O AT	1.4 其他	1.4.1 slab分配器	1.4.2 RCU
网络体系结构概述	2.1 引言	2.2 协议简介	2.3 网络架构	2.4 系统调用接口	2.5
无关接口	2.6 传输层协议	2.7 套接口缓存	2.8 设备无关接口	2.9 设备驱动程序	
网络模块源代码组织	第3章 套接口缓存	3.1 引言	3.2 sk_buff结构	3.2.1 网络参数和内核	
数据结构	3.2.2 SKB组织相关的变量	3.2.3 数据存储相关的变量	3.2.4 通用的成员变量	3.2.5 标志性变量	3.2.6 特性相关的成员变量
量	3.2.5 标志性变量	3.2.6 特性相关的成员变量	3.3 skb_shared_info结构	3.3.1	3.3.2
拷贝"技术"	3.3.2 对聚合分散I/O数据的支持	3.3.3 对GSO的支持	3.3.4 访	3.4.1 SKB的缓存池	3.4.2 分配SKB
问skb_shared_info结构	3.4 管理函数	3.4.1 SKB的缓存池	3.4.2 分配SKB	3.4.3 释	3.4.4
放SKB	3.4.4 数据预留和对齐	3.4.5 克隆和复制SKB	3.4.6 链表管理函数	3.4.7	3.4.8
添加或删除尾部数据	3.4.8 拆分数据: skb_split()	3.4.9 重新分配SKB的线性数据区	4.1 引言	4.2 网络模块	
: pskb_expand_head()	3.4.10 其他函数	第4章 网络模块初始化	4.1 引言	4.2 网络模块	
始化顺序	4.3 优化基于宏的标记	4.4 网络设备处理层初始化	第5章 网络设备	5.1 PCI设	
5.1.1 PCI驱动程序相关结构	5.1.2 注册PCI驱动程序	5.2 与网络设备有关的数据结构			
.....下册					

章节摘录

插图：尽管技术有了巨大的进步，但是TCP / IP协议栈的处理方式却几乎没有变化。

也就是说，即使用户使用最先进的CPU，依然要处理那些未经优化的TCP / IP协议，由此产生巨大的系统开销。

例如，TCP / IP的传输过程中需要封装、解包，这些动作对于处理器而言并不是一个复杂的过程，但是会占用处理器周期，而且网络带宽越高，这个问题越严重。

系统开销的增大不仅仅表现在占用较多的处理器周期，还会导致处理网络相关数据时的内存访问效率降低。

这又会进一步降低CPU效能和网络效率。

过去，网络流量较低，处理网络相关数据所产生的开销，远远低于用于执行正常任务的开销，所以并未引起重视。

现在，随着网络流量大幅度提升，处理网络相关数据所产生的系统开销越来越不能忽视，甚至已经影响到了正常应用。

现有几种解决方案：（1）TSO（TCP Segmentation Offload）通过网络设备上的专用处理器处理部分或者全部的封包，借此来降低对于系统处理器资源的占用，不过这种解决方案只对具有某些特征的数据包有效。

（2）RDMA（

Remote Direct Memory Access，远程直接内存访问）发送端系统直接将有效数据送至目的系统指定的内存中，无需移动数据包的时间消耗，因此大大提升了网络传输的效率。

但是这种技术需要专用的网络设备，应用程序也需要进行修改，甚至还增加了一个RDMA层的封装过程，而且这种操作风险较高，因此目前看来还不是一个吸引人的解决方案。

（3）Onloading技术 将系统处理器作为处理网络流量的第一引擎，尽可能地提升CPU处理网络数据包的效率，这种思想已经被英特尔借鉴。

<<Linux内核源码剖析（上下册）>>

编辑推荐

《Linux内核源码剖析:TCP/IP实现(套装上下册)》：套接口缓存网络设备IP：网际协议ICMP：Internet控制报文协议IP组播IGMP：Internet组管理协议邻居子系统路由表套接口层TCP：传输控制协议TCP连接的建立TOP拥塞控制的实现TCP的输出TCP的输入TCP连接的终止UDP：用户数据报

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>