

<<UNIX网络编程 卷2>>

图书基本信息

书名：<<UNIX网络编程 卷2>>

13位ISBN编号：9787115215116

10位ISBN编号：7115215111

出版时间：2009-11

出版单位：人民邮电出版社

作者：W.Richard Stevens

页数：1

字数：692000

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>

<<UNIX网络编程 卷2>>

前言

概述 大多数重要的程序都涉及进程间通信（Interprocess Communication, IPC）。这是受下述设计原则影响的自然结果：把应用程序设计为一组相互通信的小片断比将其设计为单个庞大的程序更好。

从历史角度看，应用程序有如下几种构建方法。

（1）用一个庞大的程序完成全部工作。

程序的各部分可以实现为函数，函数之间通过参数、返回值和全局变量来交换信息。

（2）使用多个程序，程序之间用某种形式的IPC进行通信。

许多标准的Unix工具都是按这种风格设计的，它们使用shell管道（IPC的一种形式）在程序之间传递信息。

（3）使用一个包含多个线程的程序，线程之间使用某种IPC。

这里仍然使用术语IPC，尽管通信是在线程之间而不是在进程之间进行的。

还可以把后两种设计形式结合起来：用多个进程来实现，其中每个进程包含几个线程。

在这种情况下，进程内部的线程之间可以通信，不同的进程之间也可以通信。

上面讲述了可以把完成给定任务所需的工作分到多个进程中，或许还可以进一步分到进程内的多个线程中。

在包含多个处理器（CPU）的系统中，多个进程也许可以（在不同的CPU上）同时运行，或许给定进程内的多个线程也能同时运行。

因此，把任务分到多个进程或线程中有望减少完成指定任务的时间。

本书详细描述了以下4种不同的IPC形式：（1）消息传递（管道、FIFO和消息队列）；

（2）同步（互斥量、条件变量、读写锁、文件和记录锁、信号量）；（3）共享内存（匿名的和具名的）；（4）远程过程调用（Solaris的rpc和Sun RPC）。

本书不讨论如何编写通过计算机网络通信的程序。

这种通信通常涉及使用TCP / IP协议族的套接字API，相关主题在第1卷[stevens 1998]中有详细讨论。

<<UNIX网络编程 卷2>>

内容概要

本书是一部UNIX网络编程的经典之作。

进程间通信（IPC）几乎是所有Unix程序性能的关键，理解IPC也是理解如何开发不同主机间网络应用程序的必要条件。

本书从对Posix IPC和System V IPC的内部结构开始讨论，全面深入地介绍了4种IPC形式：消息传递（管道、FIFO、消息队列）、同步（互斥锁、条件变量、读写锁、文件与记录锁、信号量）、共享内存（匿名共享内存、具名共享内存）及远程过程调用（Solaris门、Sun RPC）。

附录中给出了测量各种IPC形式性能的方法。

本书内容详尽且具权威性，几乎每章都提供精选的习题，并提供了部分习题的答案，是网络研究和开发人员理想的参考书。

<<UNIX网络编程 卷2>>

作者简介

W.Richard Stevens，国际知名的UNIX和网络专家，备受赞誉的技术作家。
他1951年2月5日出生于赞比亚，后随父母回到美国。
中学时就读于弗吉尼亚菲什伯恩军事学校，1973年获得密歇根大学航空和航天工程学士学位。
1975年至1982年，他在亚利桑那州图森市的基特峰国家天文台从

<<UNIX网络编程 卷2>>

书籍目录

Part Introduction Chapter 1 Introduction 1.1 Introduction 1.2 Processes, Threads, and the Sharing of Information 1.3 Persistence of IPC Objects 1.4 Name Spaces 1.5 Effect of fork, exec, and exit on IPC Objects 1.6 Error Handling: Wrapper Functions 1.7 Unix Standards 1.8 Road Map to IPC Examples in the Text 1.9 Summary Chapter 2 Posix IPC 2.1 Introduction 2.2 IPC Names 2.3 Creating and Opening IPC Channels 2.4 IPC Permissions 2.5 Summary Chapter 3 System V IPC 3.1 Introduction 3.2 key_t Keys and ft:ok Function 3.3 ipc_perm Structure 3.4 Creating and Opening IPC Channels 3.5 IPC Permissions 3.6 Identifier Reuse 3.7 ipcs and ~porto Programs 3.8 Kernel Limits 3.9 SummaryPart 2 Message PassingPart 3 SynchronizationPart 2 Shared MemoryPart 2 Remote Procedure CallsEpilogueAppendix AAppendix BAppendix CBibliographyIndex

章节摘录

1 The two processes on the left are sharing some information that resides in a file in the filesystem. To access this data, each process must go through the kernel (e.g., read, write, isek, and the like). Some form of synchronization is required when a file is being updated, both to protect multiple writers from each other, and to protect one or more readers from a writer.

2 The two processes in the middle are sharing some information that resides within the kernel. A pipe is an example of this type of sharing, as are System V message queues and System V semaphores. Each operation to access the shared information now involves a system call into the kernel.

3 The two processes on the right have a region of shared memory that each process can reference. Once the shared memory is set up by each process, the processes can access the data in the shared memory without involving the kernel at all. Some form of synchronization is required by the processes that are sharing the memory. Note that nothing restricts any of the IPC techniques that we describe to only two processes. Any of the techniques that we describe work with any number of processes. We show only two processes in Figure 1.1 for simplicity.

Threads Although the concept of a process within the Unix system has been used for a long time, the concept of multiple threads within a given process is relatively new. The Posix.1 threads standard (called "Pthreads") was approved in 1995. From an IPC perspective.

<<UNIX网络编程 卷2>>

媒体关注与评论

“我做专业程序员已经二十余年了，每当要写新程序的时候，我都要参考这本书。Stevens的书是我进入VoIP和音频CODEC网络编程领域的敲门砖。

” ——C. T. Vanderdecken “在我做过的项目中，已经记不清楚有多少次靠这套书挽狂澜于既倒……每位真正的UNIX系统程序员都应该买一套。

” ——David Sharpe “对每位UNIX平台的程序员而言，这本书都是必备的。

它深入阐释了UNIX平台上的各种IPC API。

既涵盖System V也涉及POSIX标准，对于IPC的论述无可匹敌。

书末的附录对各种管道、FIFO、Posix消息队列、System V消息队列、门、Sun RPC的性能进行了比较，我从来没有看到任何书能如此广泛而深入地介绍这一主题。

更何况，它出自权威人物之手。

” ——asingh “Stevens是UNIX网络编程的‘神’，至少是我的‘神’。

” ——Matthew MacGibbon

<<UNIX网络编程 卷2>>

编辑推荐

两卷本的《UNIX网络编程》是已故著名技术作家W.Richard Stevens的传世之作。卷2着重讨论如何让应用程序与在其他机器上的应用程序进行对话。

良好的进程间通信（IPC）机制是提高UNIX程序性能的关键。

《UNIX网络编程 卷2：进程间通信(英文版·第2版)》全面深入地讲解了各种进程间通信形式，包括消息传递、同步、共享内存及远程调用（RPC）。

书中包含了大量经过优化的源代码，帮助读者加深理解。

这些源代码可以从图灵网站《UNIX网络编程 卷2：进程间通信(英文版·第2版)》网页免费注册下载。

《UNIX网络编程 卷2：进程间通信(英文版·第2版)》是网络研究和开发人员公认的权威参考书，深入理解《UNIX网络编程 卷2：进程间通信(英文版·第2版)》内容，方能设计出良好的UNIX软件。

《UNIX网络编程 卷2：进程间通信(英文版·第2版)》中文版预计在2010年6月出版，敬请期待。

UNIX和网络专家W. Richard Stevens的传世之作 UNIX系统网络编程人员的必备经典 内容详尽权威，提供精选习题及答案 详细描述4种不同的IPC形式 在Unix系统上测试通过的真实、可运行的代码让读者轻松深入学习网络编程复杂的新技术。

<<UNIX网络编程 卷2>>

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>