

<<Linux系统编程>>

图书基本信息

书名：<<Linux系统编程>>

13位ISBN编号：9787564115197

10位ISBN编号：756411519X

出版时间：2009-7

出版时间：东南大学出版社

作者：Robert Love

页数：382

译者：O'Reilly Taiwan公司

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系统编程>>

前言

内核开发者之所以会这么说，目的是于任何用户空间程序代码执行失败的时候撇清所有责任。当所发生的问题绝对不是内核的过错时，他们所关心的是用户空间开发者应该去修正他们自己的程序代码。

为了证明这通常不是内核的过错，3年多前，一位具领导地位的Linux内核开发者曾在挤满人的会议室里，以“why user Space Sucks”（为何用户空间程序很糟糕）为题发表演讲，他举出实例说明我们每个人每天使用了哪些可怕的用户空间程序代码。

其他内核开发者则以自己创建的工具来展示差劲的用户空间程序如何滥用硬件，并耗尽无预警的笔记本电脑电池的电量。

尽管用户空间程序对嘲笑它的内核开发者而言可能只是一个“试验负载”，不过这些内核开发者也是每天都得依靠这些用户空间程序。

如果没有用户空间程序可用，所有的内核充其量就只能在屏幕上交替输出ABABAB样式的信息。

而今，Linux已经成为有史以来最灵活、最强大的操作系统，随处都可以看到它的踪迹，不仅最小型的手机和嵌入式装置运行它，全世界前500台速度最快的超级计算机中也有70%以上的在运行它。

其他操作系统从未有过这么好的规模，也不会遭受各种硬件和环境的挑战。

如同内核一样，在Linux的用户空间上运行的程序也得运作在各种平台上，以人们所依赖的应用程序和公用程序提供给全世界使用。

<<Linux系统编程>>

内容概要

在某些时刻，几乎所有的程序员都要与其程序所处操作系统中的系统调用和程序库打交道。

《Linux系统编程》主要讨论如何编写Linux系统软件——代码位于底层，并且直接跟内核及核心系统程序库对话。

《Linux系统编程》描述了使用标准接口包括使用Linux独有的高级接口时，在功能和性能之间如何进行权衡取舍的策略。

该书同样也是一本内行人士编写灵活高效代码的学习指南。

作为内核黑客和《Linux系统编程》的作者，Robert Love不仅阐释了系统接口应该如何工作，还介绍了它们实际上是如何工作的，以及怎样安全有效地使用它们。

《Linux系统编程》包含了帮助你在任何层面编写更佳代码的实用技巧。

《Linux系统编程》主题包括：
读写文件以及其他文件I/O操作，包括Linux内核如何实现和管理文件I/O，内存映射与优化
技术进程管理的系统调用，包括实时进程
文件与目录——创建、移动、复制、删除和管理
内存管理——内存分配接口，管理内存，以及优化内存访问
信号及其在Unix系统中的角色，以及基本和高级信号接口
时间、休眠和时钟管理，从基础开始讲述，并且涵盖POSIX时钟和高精度计时器
拥有《Linux系统编程》，你将从理论和应用的角度深入了解Linux，可以最大限度地利用系统的潜能。

<<Linux系统编程>>

作者简介

作者：(美国)Robert Love 译者：O'Reilly Taiwan公司

<<Linux系统编程>>

书籍目录

序前言第一章 介绍与基本概念系统编程API与ABI标准Linux编程的概念向系统编程迈进第二章 文件I/O打开文件以read()进行读取操作以write()进行写入操作同步化I/O关闭文件使用lseek()查找文件位置针对特定位置的读取与写入截短文件多任务式I/O内核内部结束语第三章 缓冲式I/O用户缓冲式I/O标准I/O打开文件经文件描述符打开流关闭流从流中读取使用缓冲式I/O的简单程序查找一个流刷新一个流错误与EOF取得相应的文件描述符控制与缓冲机制线程安全标准I/O的缺陷结束语第四章 高级文件I/O分散-聚集I/O事件轮询接口将文件映射至内存对一般文件I/O的用法提供建议同步化、同步及异步操作I/O调度程序与I/O性能结束语第五章 进程管理进程ID运行一个新进程终止一个进程等待已终止的子进程用户与组会话与进程组守护进程结束语第六章 高级进程管理进程的调度让出处理器进程优先级实时系统资源限制第七章 文件和目录管理文件与其元数据目录链接文件的复制以及移动设备节点带外通信第八章 内存管理进程地址空间分配动态内存管理数据段匿名内存映射高级内存分配调试内存分配基于堆栈的分配选择内存分配机制操作内存锁定内存投机取巧的分配策略第九章 信号信号的概念基本的信号管理发送一个信号可重人性信号集阻挡信号高级信号管理以payload送出信号结束语第十章 时间时间的数据结构POSIX时钟取得当前时间设定当前时间操作时间调整系统时钟休眠与等待定时器附录GCC对C语言的扩展参考书目

<<Linux系统编程>>

章节摘录

插图：Linux编程的概念本节将对Linux系统所提供的服务做一个简明的概述。

所有的unix系统，包括Linux，会提供一组共同的抽象概念以及接口。

的确，这种通用性定义了unix。

文件和进程、用于管理pipe和socket的接口等抽象概念就是unix的基础组件。

本节假设你已经熟悉Linux的操作环境：你可以在shell中游走、使用基本的命令以及编译简单的c程序

。本节的内容并非Linux或其编程环境的概述，而是Linux系统编程的基本知识。

文件以及文件系统文件是Linux中最基本且最重要的抽象概念。

Linux遵循了“一切皆文件”的哲学（尽管做得不如其他系统，例如Plan9（注2））。

因此，许多操作是通过对文件的读写进行的，即使所操作的对象并非你平日所使用的文件。

文件必须先被打开才可以被访问。

文件可以被打开以备读取、写入或是读写。

你可以通过一个独一无二的描述符（descriptor）来引用一个已打开的文件，此描述符让我们可以从与已打开文件相关的元数据映射回相应文件。

在Linux内核内部，此描述符的操作是通过一个称为文件描述符的整数（这是c的int数据类型）来进行

。由用户空间程序所共享，而用户空间程序可以直接使用来访问文件。

Linux系统编程多半就是用来进行文件的打开、操纵、关闭以及其他工作。

常规文件一般人所谓的文件就是Linux所称的常规文件。

常规文件内含数据的字节，而且被组织成一个称为字节流（bytestream）的线性数组（Lineararray）。

在Linux中，不会再为文件指定任何格式。

这些字节可以具有任何值，而且可以通过任何方式被组织在一起。

从系统层来看，除了字节流之外，Linux不会再对文件套上任何结构。

有些操作系统，例如VMs，会提供高度结构化的文件来支持像记录（record）这样的概念。

Linux则不会这么做。

<<Linux系统编程>>

编辑推荐

《Linux系统编程》是由东南大学出版社出版的。

<<Linux系统编程>>

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>