

<<Linux 操作系统原理与应用>>

图书基本信息

书名：<<Linux 操作系统原理与应用>>

13位ISBN编号：9787302278368

10位ISBN编号：7302278369

出版时间：2012-1

出版时间：清华大学出版社

作者：陈莉君，康华 编著

页数：255

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 操作系统原理与应用>>

内容概要

本书是Linux内核及动手实践的入门教程。

在庞大的Linux内核中，选取最基本的内容——进程管理、中断、内存管理、系统调用、内核同步、文件系统、i / o设备管理等进行阐述。

从原理出发，基于Linux内核源代码但又不局限于代码，分析原理如何落实到代码，并通过简单有效的实例说明如何调用Linux内核提供的函数进行内核级程序的开发。

主要章节给出了具有实用价值的小型应用，从而让读者在实践中加深对原理的理解和应用能力。

本书对于希望深入Linux操作系统内部、阅读Linux内核源代码以及进行内核级程序开发的读者具有较高的参考价值。

本书可作为高等院校计算机相关专业的本科生、研究生的教材，Linux应用开发人员、嵌入式系统开发人员等均可从本书中获益。

<<Linux 操作系统原理与应用>>

书籍目录

第1章 概述

- 1.1 认识操作系统
 - 1.1.1 从使用者角度看
 - 1.1.2 从程序开发者的角度看
 - 1.1.3 从操作系统在整个计算机系统所处位置看
 - 1.1.4 从操作系统设计者的角度看
 - 1.1.5 操作系统组成
- 1.2 开放源代码的UNIX / Linux操作系统
 - 1.2.1 UNIX诞生和发展
 - 1.2.2 Linux诞生
 - 1.2.3 操作系统标准POSIX
 - 1.2.4 GNU和Linux
 - 1.2.5 Linux的开发模式
- 1.3 Linux内核
 - 1.3.1 Linux内核的技术特点
 - 1.3.2 Linux内核的位置
 - 1.3.3 Linux内核体系结构
- 1.4 Linux内核源代码
 - 1.4.1 Linux内核版本
 - 1.4.2 Linux内核源代码的结构
 - 1.4.3 Linux内核源代码分析工具
- 1.5 Linux内核模块编程入门
 - 1.5.1 模块的定义
 - 1.5.2 编写一个简单的模块
 - 1.5.3 应用程序与内核模块的比较
- 1.6 Linux内核中链表的实现及应用
 - 1.6.1 链表的演化
 - 1.6.2 链表的定义和操作
 - 1.6.3 链表的应用
- 1.7 小结

习题

第2章 内存寻址

- 2.1 内存寻址
 - 2.1.1 Intel x86 CPU寻址的演变
 - 2.1.2 80x86寄存器简介
 - 2.1.3 物理地址、虚拟地址及线性地址
- 2.2 段机制
 - 2.2.1 段描述符
 - 2.2.2 地址转换及保护
 - 2.2.3 Linux中的段
- 2.3 分页机制
 - 2.3.1 页与页表
 - 2.3.2 线性地址到物理地址的转换
 - 2.3.3 分页举例
 - 2.3.4 页面高速缓存

<<Linux 操作系统原理与应用>>

- 2.4 Linux中的分页机制
- 2.5 Linux中的汇编语言
 - 2.5.1 AT&T与Intel汇编语言的比较
 - 2.5.2 AT&T汇编语言的相关知识
 - 2.5.3 GCC嵌入式汇编
- 2.6 Linux系统地址映射举例
- 2.7 小结

习题

第3章 进程

- 3.1 进程介绍
 - 3.1.1 程序和进程
 - 3.1.2 进程的层次结构
 - 3.1.3 进程状态
 - 3.1.4 进程举例
- 3.2 Linux系统中的进程控制块
 - 3.2.1 进程状态
 - 3.2.2 进程标识符
 - 3.2.3 进程之间的亲属关系
 - 3.2.4 进程控制块的存放
 - 3.2.5 当前进程
- 3.3 Linux系统中进程的组织方式
 - 3.3.1 进程链表
 - 3.3.2 哈希表

.....

- 第4章 内存管理
- 第5章 中断和异常
- 第6章 系统调用
- 第7章 内核中的同步
- 第8章 文件系统
- 第9章 设备驱动
- 参考文献

章节摘录

版权页：插图：为此，Linux内核设置了一个定期将页面换出的守护进程kswapd。

从原理上说，kswapd相当于一个进程，它有自己的进程控制块task_struct结构。

与其他进程一样受内核的调度。

而正因为内核将它按进程来调度，所以可以让它在系统相对空闲的时候来运行。

不过，与普通进程相比，kswapd有其特殊性。

首先，它没有自己独立的地址空间，所以在近代操作系统理论中把它称为“线程”或“守护进程”以与进程相区别。

那么，kswapd到底隔多长时间运行一次，这由内核设计时所确定的一个常量HZ决定。

HZ决定了内核中每秒时钟中断的次数，用户可以在编译内核前的系统配置阶段改变其值，但是一经编译就确定下来了。

在Linux2.4的内核中，每秒钟kswapd被调用一次。

kswapd的执行路线分为两部分，第一部分是发现物理页面已经短缺的情况下才进行的，目的在于预先找出若干页面，且将这些页面的映射断开，使这些物理页面从活跃状态转入不活跃状态，为页面的换出做好准备。

第二部分是每次都要执行的，目的在于把已经处于不活跃状态的“脏”页面写入交换区，使它们成为不活跃的“干净”页面继续缓冲，或进一步回收这样的页面成为空闲页面。

在本章的学习中，有一点需特别向读者强调。

在Linux系统中，CPU不能按物理地址访问存储空间，而必须使用虚拟地址。

因此，对于Linux内核映像，即使系统启动时将其全部装入物理内存，也要将其映射到虚拟地址空间中的内核空间，而对于用户程序，其经过编译、链接后形成的映像文件最初存于磁盘上，当该程序被运行时，先要建立该映像与虚拟地址空间的映射关系，当真正需要物理内存时，才建立地址空间与物理空间的映射关系。

内存管理实例我们希望能通过访问用户空间的内存达到读取内核数据的目的，这样便可以进行内核空间到用户空间的大规模信息传送，从而应用于高速数据采集等性能要求高的场合。

因为通过外设采集的数据首先会由驱动程序放入内核，然后才传送到用户空间由应用程序做进一步的处理。

由于内核内存是受保护的，因此，要想将其数据拷贝到用户空间，通常的方法是利用系统调用，但是系统调用的缺点是速度慢，这会成为数据高速处理的瓶颈。

因此我们希望能从用户空间直接读取内核数据，从而省去数据在两个空间拷贝的过程。

具体地讲，我们要利用内存映射功能，将内核中的一部分虚拟内存映射到用户空间，使得访问用户空间地址等同于访问被映射的内核空间地址，从而不再需要数据拷贝操作。

4.6.1 相关背景知识我们知道，在内核空间中调用kmalloc（）分配连续物理空间，而调用vmalloc（）分配非连续物理空间。

在这里，把kmalloc（）所分配的内核空间称为内核逻辑空间（KernelLogicSpace）。

它所分配的内核空间虚地址是连续的，所以能很容易获得其对应的实际物理地址，即“内核虚地址-PAGE_OFFSET=实际的物理地址”。

另外，由于系统在初始化时就建立了内核页表swapper_pg_dir，而kmalloc（）分配过程所使用的就是该页表，因此也省去了建立和更新页表的工作。

<<Linux 操作系统原理与应用>>

编辑推荐

《Linux 操作系统原理与应用(第2版)》编辑推荐：教学目标明确，注重理论与实践的结合。
教学方法灵活，培养学生自主学习的能力。
教学内容先进，强调计算机在各专业中的应用。
教学模式完善，提供配套的教学资源解决方案。

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>