

<<Linux 2.6内核标准教程>>

图书基本信息

书名：<<Linux 2.6内核标准教程>>

13位ISBN编号：9787115187116

10位ISBN编号：7115187118

出版时间：2008-11

出版时间：人民邮电出版社

作者：华清远见嵌入式培训中心，河秦，王洪涛 编著

页数：368

字数：567000

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 2.6内核标准教程>>

前言

写作背景自由、开放的Linux操作系统正在蓬勃发展，得到了广泛的应用，Linux操作系统的用户数量迅速增长，很多Linux爱好者希望能够学习、掌握Linux内核的原理、机制，能够阅读Linux内核代码，并能够加以应用，但在实际的学习过程中常遇到以下问题。

Linux内核学习门槛较高，初学者总想迅速读懂内核源代码，往往在不清楚内核运行机制和内核代码结构的情况下就开始阅读Linux内核代码，会遇到很大障碍。

随着Linux内核的发展，内核的代码量日益增加，系统规模不断扩大，复杂度不断提高。

<<Linux 2.6内核标准教程>>

内容概要

Linux内核是Linux操作系统中最核心的部分，用于实现对硬件部件的编程控制和接口操作。

本书深入、系统地讲解了Linux内核的工作原理，对Linux内核的核心组件逐一进行深入讲解。

全书共8章，首先讲解Linux系统的引导过程；然后对Linux内核的3大核心模块——内存管理、进程管理、中断和异常处理进行了深入的分析；在此基础上，对时间度量、系统调用进行了分析和讨论；最后讲解了Linux内核中常见的同步机制，使读者掌握每处理器变量和RCU这两种新的同步机制。

本书适合Linux内核爱好者、Linux驱动开发人员、Linux系统工程师参考使用，也可以作为计算机及相关专业学生深入学习操作系统的参考书。

<<Linux 2.6内核标准教程>>

书籍目录

第1章 Linux内核学习基础 1.1 为什么研究Linux内核 1.1.1 Linux的历史来源 1.1.2 Linux的发展现状 1.1.3 Linux的前景展望 1.2 选择什么版本进行研究 1.3 内核基本结构 1.3.1 内核在操作系统中的地位 1.3.2 Linux 2.6内核源代码目录树简介 1.3.3 Linux 2.6内核的新特性 1.4 如何阅读本书 1.4.1 内核探索工具 1.4.2 推荐阅读方法 第2章 引导过程分析 2.1 内核镜像的构建过程 2.1.1 编译内核的步骤及分析 2.1.2 内核镜像构建过程分析 2.2 系统引导过程分析 2.2.1 傀儡引导扇区 2.2.2 探测系统资源 2.2.3 解压内核镜像 2.2.4 进入保护模式 2.2.5 系统最终初始化 2.3 系统引导过程总结 第3章 内存管理 3.1 基础知识 3.1.1 存储器地址 3.1.2 分段机制 3.1.3 分页机制 3.2 内核页表的初始化过程 3.2.1 启用分页机制 3.2.2 构建内核页表 3.3 物理内存的描述方法 3.3.1 内存节点 3.3.2 内存区域 1 3.3.3 物理页框 3.4 物理内存的初始化过程 3.4.1 探测系统物理内存 3.4.2 初始化内存分配器 3.5 物理内存的分配与回收 3.5.1 伙伴分配算法 3.5.2 对象缓冲技术 3.6 内核地址空间 3.6.1 常规映射地址空间 3.6.2 固定映射地址空间 3.6.3 长久内核映射空间 3.6.4 临时内核映射空间 3.6.5 非连续映射地址空间 第4章 进程管理 4.1 进程与线程的概念 4.1.1 程序与进程 4.1.2 进程与线程 4.2 进程描述符 4.2.1 进程标识符 4.2.2 进程的状态 4.2.3 进程上下文 4.2.4 当前进程 4.3 进程的组织形式 4.3.1 进程标识符构成的哈希表 4.3.2 所有进程构成的双向链表 4.3.3 执行态进程组成的运行队列 4.3.4 阻塞态进程组成的等待队列 4.4 进程的创建过程 4.4.1 进程创建的接口函数 4.4.2 进程创建的处理过程 4.5 进程调度算法 4.5.1 进程的分类 4.5.2 进程优先级 4.5.3 时间片分配 4.5.4 进程调度时机 4.6 进程切换过程分析 4.6.1 选取合适进程 4.6.2 完成上下文切换 4.7 空闲进程的初始化 4.7.1 空闲进程的内存态栈 4.7.2 空闲进程的内存描述符 4.7.3 空闲进程的硬件上下文 4.7.4 空闲进程的任务状态段 第5章 中断和异常 5.1 基础知识 5.1.1 中断和异常的定义 5.1.2 中断和异常的分类 5.1.3 中断和异常的对比 5.2 处理机制 5.2.1 IA32架构下的处理机制 5.2.2 Linux内核的实现策略 5.3 中断描述符表的初始化 5.3.1 中断描述符表的初步初始化 5.3.2 中断描述符表的最终初始化 5.4 具体处理过程 5.4.1 公用的硬件处理阶段 5.4.2 中断的软件处理阶段 5.4.3 异常的软件处理阶段 5.5 延迟处理机制 5.5.1 softirq延迟处理 5.5.2 tasklet延迟处理 5.5.3 work queue延迟处理 第6章 时间度量 6.1 硬件支持 6.1.1 实时钟RTC 6.1.2 系统时钟 6.2 软件架构 6.2.1 相对时间 6.2.2 墙上时间 6.2.3 内核定时器 6.3 时间度量的初始化过程 6.3.1 内核定时器初始化 6.3.2 系统时钟的初始化 6.3.3 初始化时钟中断源 6.4 时钟中断处理过程 6.4.1 找回遗失的时钟中断 6.4.2 更新jiffies_64、xtime 6.4.3 对当前进程记账 6.4.4 时钟中断处理小结 6.5 内核定时器工作原理 6.5.1 初始化内核定时器节点 6.5.2 激活内核定时器节点 6.5.3 内核定时器的处理过程 6.6 微妙级延迟 6.6.1 微妙级延迟的访问接口 6.6.2 微妙级延迟的实现方法 第7章 系统调用 7.1 系统服务接口的种类 7.1.1 系统调用接口 7.1.2 应用编程接口 7.2 系统调用的访问手段 7.2.1 使用封装函数 7.2.2 使用通用接口 7.2.3 使用内嵌汇编 7.3 系统调用的工作机制 7.3.1 系统调用的基本要素 7.3.2 系统调用门的初始化 7.3.3 系统调用的处理过程 7.4 系统调用的参数传递 7.4.1 少量参数的情况 7.4.2 大量参数的情况 7.5 如何添加新系统调用 7.5.1 前期准备工作 7.5.2 添加处理函数 7.5.3 测试新系统调用 7.6 什么是快速系统调用 7.6.1 工作机制 7.6.2 实现策略 7.6.3 处理过程 第8章 内核同步机制 8.1 同步基本原理 8.1.1 原子变量 8.1.2 中断禁用 8.1.3 内核态抢占 8.2 系统引导过程分析 8.2.1 普通自旋锁 8.2.2 读写自旋锁 8.2.3 顺序自旋锁 8.3 信号量机制 8.3.1 普通信号量 8.3.2 读写信号量 8.4 其他同步机制 8.4.1 每处理器变量 8.4.2 RCU同步机制 附录A Linux内核双向链表 A.1 内核链表表头 A.2 内核链表遍历 A.3 内核链表遍历 附录B 跟踪调试内核 B.1 安装辅助工具 B.2 准备内核镜像 B.3 准备根文件系统 B.4 进行源码级调试 附录C Linux内核汇编语法 C.1 常规汇编语法 C.1.1 寄存器前缀 C.1.2 立即数前缀 C.1.3 操作数顺序 C.1.4 操作数宽度 C.1.5 内存寻址格式 C.2 内嵌汇编语法 C.2.1 内嵌汇编举例 C.2.2 内嵌汇编格式——格式框架 C.2.3 内嵌汇编格式——语句模板 C.2.4 内嵌汇编格式——输出列表 C.2.5 内嵌汇编格式——输入列表 C.2.6 内嵌汇编格式——修饰字符 C.2.7 内嵌汇编格式——破坏描述 附录D 参考文献 D.1 关于IA32体系结构的资源 D.2 关于Linux内核的相关资源 D.3 关于计算机基本原理的资源 D.4 其他相关资源

<<Linux 2.6内核标准教程>>

章节摘录

插图：第1章 Linux内核学习基础1.1 为什么研究Linux内核本节先对Linux系统的来源、历史背景和现状作简单介绍，然后讨论自由软件的价值所在及自由软件如何满足用户的需要，最后讲解Linux系统的应用前景和未来。

1.1.1 Linux的历史来源谈到Linux内核，不能不介绍GNU项目。

假若没有GNU计划和GNU推出的自由软件，Linux内核不可能如此成功；Linux内核的开发工作都是基于GNU推出的自由软件完成的。

反过来，没有Linux内核，GNU计划也只是一个空谈——它无法向用户提供一个完全自由的类UNIX操作系统。

GNU和Linux相辅相成、缺一不可。

GNU工程始于1984年，由自由软件之父Richard Stallman组织，目的在于开发一套自由、完整的UNIX操作系统。

该系统是一个完全“自由”的软件体系，与此相应的有一份通用公共许可证（General Public License，GPL）。

和大多数软件许可证相反，GNU通用公共许可证力图保证用户共享和修改自由软件的自由——保证自由软件对所有用户是自由的，这里指的是自由而不是免费。

GPL的详细信息，请参见<http://www.gnu.org/licenses/licenses.html>。

Linux内核及其他相关的大量软件都是在GPL的推动下开发和发布的。

各种使用Linux作为内核的GNU操作系统正被广泛地使用着，虽然这些系统通常被称作为Linux，但是它们应该被更确切地称为GNU/Linux系统。

原因在于，Linux实际上是操作系统的内核，使这个内核变得非常有用的大量应用程序都是GNU软件，都是在GPL许可证下发布的。

例如，窗口管理系统、编译器、各种shell、脚本解释器、浏览器、编辑器以及其他多不胜数的实用工具。

基于该原因，GNU/Linux更适合作为操作系统的名称，而Linux则适合作为内核的名称。

GNU项目的开发策略是先完成现有UNIX系统中实用程序（如ls、cat、autoconf、automake）的自由实现，然后开发GNU操作系统的核心——Hurd内核，只要Hurd内核开发出来，GNU就是一套真正自给自足、完整的自由操作系统。

但与此同时，Linux内核在短短几年内迅速崛起，也使用GNU的通用公共许可证，填补了GNU Hurd内核的空缺。

因为Linux内核已经十分成熟，目前几乎所有的GNU系统都采用Linux作为操作系统的核心。

如果说Richard Stallman创立并推动了自由软件的发展，那么Linus毫不犹豫奉献给GNU的Linux，则把自由软件的发展带入了一个全新的境界。

Linux内核是一个极其成功的自由软件，是自由软件中的一个典范。

<<Linux 2.6内核标准教程>>

编辑推荐

理解Linux内核的工作原理是做好嵌入式Linux开发的关键，也是嵌入式Linux系统开发的难点之一。

《Linux2.6内核标准教程》紧扣这一主题，以简洁的语言进行了深入浅出的讲解，值得一读。

北京麦克泰软件技术有限公司（BMR）董事长 / 电子产品世界杂志编委何小庆《Linux2.6内核标准教程》有非常突出的特点—图文并茂、分析透彻，全书配以大量图表引导读者建立起对Linux内核的认识和理解，是作者多年嵌入式Linux从业经验的总结，读者一定会从中受益。

中国软件行业协会嵌入式系统分会秘书长郭淳学在开源技术盛行的今天，基于Linux内核的计算机系统正在发挥巨大的作用，越来越多的人参与到Linux的相关开发之中，《Linux2.6内核标准教程》是深入学习Linux内核的最佳选择。

<<Linux 2.6内核标准教程>>

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>