

图书基本信息

书名：<<CDIO工程教育系列教程·计算机专业基础类>>

13位ISBN编号：9787894360458

10位ISBN编号：7894360457

出版时间：2012-1

出版时间：东软电子出版社

作者：董祥千 编

页数：343

字数：564

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>

内容概要

本教材以Linux操作系统内核为基础，本着循序渐进的原则，采用通俗易懂的语言和丰富的实例，阐述操作系统的基本概念、原理及方法。

全教材共9章：第1章绪论；第2章操作系统的导引；第3章存储管理；第4章进程管理；第5章处理机管理；第6章操作系统用户接口；第7章设备管理；第8章文件管理；第9章Linux内核分析。

本教材适用于计算机专业本、专科《计算机操作系统》相关课程的教学。

书籍目录

第1章绪论

1.1学习引导

1.2操作系统的概念

1.2.1什么是操作系统

1.2.2操作系统的发展

1.3操作系统的功能

1.3.1处理机管理

1.3.2存储器管理

1.3.3设备管理

1.3.4文件管理

1.3.5用户接口

1.4操作系统的分类

1.4.1单用户操作系统

1.4.2多道批处理系统

1.4.3分时系统

1.4.4实时系统

1.4.5网络操作系统

1.4.6分布式操作系统

1.4.7嵌入式操作系统

1.5操作系统的特征和性能指标

1.5.1操作系统的特征

1.5.2操作系统性能指标

1.6Linux系统概述

1.6.1Linux系统形成和发展的基础

1.6.2Linux系统的特点

1.7单元项目：Linux内核结构

1.8扩展阅读

1.9本章小结

1.10习题

第2章操作系统的引导

2.1学习引导

2.2系统启动硬件基础

2.2.1主板与内存映射

2.2.2加电自检（POST）过程

2.3根文件系统

2.3.1根文件系统

2.3.2Busybox文件系统简介

2.4操作系统启动过程

2.4.1引导程序BootLoader的加载

2.4.2操作系统内核的加载

2.4.3计算机启动过程

2.5GDB调试

2.5.1GDB简介

2.5.2GDB命令

2.6单元项目

- 2.6.1 Linux内核的编译
- 2.6.2 制作带根文件系统的磁盘映像
- 2.6.3 制作带引导功能的启动磁盘
- 2.6.4 Linux内核调试
- 2.7 本章小结
- 2.8 习题
- 第3章 存储管理
- 3.1 学习引导
- 3.2 存储管理的基本概念
 - 3.2.1 存储管理的目的和功能
 - 3.2.2 存储管理的方式
- 3.3 连续分配方式
 - 3.3.1 单一连续分配
 - 3.3.2 固定分区存储管理
 - 3.3.3 动态分区存储管理
 - 3.3.4 交换技术
- 3.4 离散分配方式
 - 3.4.1 基本分页存储管理
 - 3.4.2 基本分段存储管理方式
 - 3.4.3 段页式存储管理方式
- 3.5 虚拟存储器
 - 3.5.1 虚拟存储器的引入
 - 3.5.2 虚拟存储器的实现
 - 3.5.3 虚拟存储器的特征
- 3.6 请求分页存储管理
 - 3.6.1 请求分页中的硬件支持
 - 3.6.2 内存的分配
 - 3.6.3 页面置换算法
 - 3.6.4 页面抖动现象
- 3.7 请求分段存储管理
 - 3.7.1 段表机制
 - 3.7.2 缺段中断机构
 - 3.7.3 地址变换机构
- 3.8 Linux内存管理
 - 3.8.1 Linux存储管理的重要数据结构
 - 3.8.2 页表的管理
 - 3.8.3 页面分配和回收
 - 3.8.4 Linux页面置换
- 3.9 单元项目
 - 3.9.1 动态分区分配模拟项目
 - 3.9.2 页面置换算法模拟项目
- 3.10 本章小结
- 3.11 习题
- 第4章 进程管理
- 4.1 学习引导
- 4.2 进程管理
 - 4.2.1 程序、程序的顺序执行及其特征

- 4.2.2程序的并发执行及其特征
- 4.2.3进程的定义及其特征
- 4.2.4进程的基本状态及其转换
- 4.2.5进程切换
- 4.2.6进程控制块PCB
- 4.2.7进程控制块的组织方式
- 4.2.8进程控制
- 4.3进程互斥与同步及信号量机制
 - 4.3.1临界资源和临界区
 - 4.3.2进程互斥与同步的概念
 - 4.3.3信号量机制
 - 4.3.4信号量集机制
- 4.4进程间通信
 - 4.4.1进程间通信的方式
 - 4.4.2进程地址空间
 - 4.4.3Linux进程间通信
 - 4.4.4System V共享内存实现
 - 4.4.5System V共享内存编程接口
 - 4.4.6System V信号量机制
- 4.5死锁
 - 4.5.1死锁产生的原因和必要条件
 - 4.5.2解决死锁的基本方法
- 4.6Linux进程的实现
 - 4.6.1Linux进程描述符
 - 4.6.2Linux进程的状态
 - 4.6.3Linux进程的虚拟内存布局
 - 4.6.4Linux进程的内核堆栈
 - 4.6.5Linux的init进程
 - 4.6.7Linux用户进程的创建
- 4.7线程
 - 4.7.1线程的基本概念
 - 4.7.2用户级线程与内核级线程
 - 4.7.3多线程模型
 - 4.7.4Linux线程技术
 - 4.7.5Glibc中的NPTL库编程接口
 - 4.7.6POSIX信号量
- 4.8单元项目
 - 4.8.1进程管理的模拟实现
 - 4.8.2生产者-消费者问题
 - 4.8.3利用共享内存实现读者写者问题
 - 4.8.4哲学家就餐问题
 - 4.8.5司机与售票员问题
- 4.9本章小结
- 4.10习题
- 第5章处理机管理
 - 5.1学习引导
 - 5.2基本知识

- 5.2.1 调度程序
- 5.2.2 分派程序
- 5.2.3 调度算法的若干准则
- 5.2.4 进程优先级
- 5.2.5 调度策略
- 5.2.6 进程调度层次
- 5.2.7 进程调度的功能
- 5.2.8 进程调度时机
- 5.2.9 作业的状态及其转换
- 5.3 常用的调度队列模型
 - 5.3.1 只有进程调度的调度队列模型
 - 5.3.2 具有高级和低级调度的调度队列模型
 - 5.3.3 同时具有三级调度的调度队列模型
- 5.4 常用调度算法
 - 5.4.1 先来先服务调度算法
 - 5.4.2 短作业调度算法 (SJF) 或短进程调度算法 (SPF)
 - 5.4.3 高响应比优先调度算法 (HRN)
 - 5.4.4 最高优先级 (FPF) 优先调度算法
 - 5.4.5 基于时间片轮转的调度算法
- 5.5 Linux 进程调度框架
 - 5.5.1 Linux 进程的分类
 - 5.5.2 时间片的选择
 - 5.5.3 Linux 2.6.26 的两种主流调度器
- 5.6 单元项目
 - 5.6.1 作业调度算法模拟
 - 5.6.2 进程调度算法模拟
 - 5.6.3 CFS 算法在 Linux 上的实现分析
- 5.7 本章小结
- 5.8 习题
- 第6章 操作系统用户接口
 - 6.1 学习引导
 - 6.2 操作系统的接口
 - 6.2.1 程序接口
 - 6.2.2 命令接口
 - 6.2.3 作业控制级接口
 - 6.3 Linux 的用户接口
 - 6.3.1 常用 Shell 命令
 - 6.3.2 Shell 编程
 - 6.4 单元项目
 - 6.4.1 “Hello world” 项目
 - 6.4.2 Shell 变量项目
 - 6.4.3 Shell 程序设计的流程控制项目
 - 6.5 扩展阅读
 - 6.6 本章小结
 - 6.7 习题
- 第7章 设备管理
 - 7.1 学习引导

- 7.2设备管理概述
 - 7.2.1设备管理的目的和功能
 - 7.2.2设备的分类
- 7.3I/O系统
 - 7.3.1设备控制器
 - 7.3.2I/O通道
 - 7.3.3总线系统
 - 7.4I/O数据传输方式
 - 7.4.1程序循环测试方式
 - 7.4.2中断驱动I/O控制方式
 - 7.4.3直接存储器存取（DMA）方式
 - 7.4.4I/O通道方式
 - 7.5缓冲技术
 - 7.5.1缓冲的引入
 - 7.5.2缓冲的类型
 - 7.6设备分配
 - 7.6.1设备分配中的数据结构
 - 7.6.2设备的独立性
 - 7.6.3分配中应考虑的因素
 - 7.6.4独占设备的分配程序
 - 7.7I/O处理
 - 7.7.1中断处理程序
 - 7.7.2设备驱动程序
 - 7.8虚拟设备
 - 7.8.1Spooling技术
 - 7.8.2共享打印机
 - 7.9磁盘管理
 - 7.9.1磁盘概述
 - 7.9.2磁盘调度
 - 7.9.3廉价磁盘冗余阵列
 - 7.10Linux设备管理
 - 7.10.1Linux中的设备文件
 - 7.10.2Linux的设备驱动程序
 - 7.10.3Linux的中断处理
 - 7.11单元项目：磁盘调度算法模拟实验
 - 7.12本章小结
 - 7.13习题

第8章文件管理

 - 8.1学习引导
 - 8.2文件的基本概念
 - 8.2.1文件属性
 - 8.2.2文件操作
 - 8.2.3文件格式
 - 8.2.4文件结构
 - 8.3文件存取方法
 - 8.3.1顺序存取
 - 8.3.2直接存取（随机存取）

- 8.3.3 按键存取
- 8.4 文件目录
 - 8.4.1 文件目录结构
 - 8.4.2 文件目录的实现
- 8.5 文件系统
 - 8.5.1 文件系统的功能
 - 8.5.2 文件系统的层次结构
- 8.6 空闲存储空间的管理
 - 8.6.1 空闲空间表法
 - 8.6.2 空闲块链法
 - 8.6.3 位示图法
 - 8.6.4 成组链接法
- 8.7 磁盘空间分配方式
 - 8.7.1 连续分配方式
 - 8.7.2 链接分配方式
 - 8.7.3 索引分配方式
- 8.8 文件共享
 - 8.8.1 早期文件共享的方法
 - 8.8.2 基于索引结点的共享方式（硬链接）
 - 8.8.3 利用符号链实现文件共享（软链接）
- 8.9 文件保护
 - 8.9.1 保护域
 - 8.9.2 访问矩阵
 - 8.9.3 访问矩阵的修改
 - 8.9.4 访问矩阵的实现
 - 8.9.5 分级安全管理
- 8.10 Linux文件系统分析
 - 8.10.1 Linux文件系统体系结构
 - 8.10.2 VFS文件系统
 - 8.10.3 ext2文件系统
 - 8.10.4 文件系统的挂载
- 8.11 单元项目
 - 8.11.1 Linux文件系统模拟
 - 8.11.2 给Linux系统添加新硬盘
- 8.12 本章小结
- 8.13 习题
- 第9章 Linux内核分析
 - 9.1 学习引导
 - 9.2 Linux内核代码组织
 - 9.2.1 微内核与单内核
 - 9.2.2 Linux内核源代码结构
 - 9.3 Linux内核分析方法
 - 9.4 单元项目
 - 9.4.1 Linux中的时钟和定时测量的初始化
 - 9.4.2 Linux内核编译过程分析
 - 9.4.3 Linux操作系统启动过程分析
 - 9.5 本章小结

9.6习题
参考文献

章节摘录

版权页：插图：4.4.3 Linux进程间通信 Linux下的进程通信手段基本上是从UNIX平台上的进程通信手段继承而来的。

而对UNIX发展做出重大贡献的两大主力AT&T的贝尔实验室及BSD（加州大学伯克利分校的伯克利软件发布中心在进程间通信方面的侧重点有所不同。

前者对UNIX早期的进程间通信手段进行了系统的改进和扩充，形成了“system V IPC”，通信进程局限在单个计算机内；后者则跳过了该限制，形成了基于套接口（socket）的进程间通信机制。

Linux则把两者继承了下来，如图4.15所示。

其中，最初UNIX IPC包括：管道、FIFO、信号；System V IPC包括：System V消息队列、System V信号灯、System V共享内存区；Posix IPC包括：Posix消息队列、Posix信号灯、Posix共享内存区。

这里有两点需要简单说明：（1）由于UNIX版本的多样性，电子电气工程协会（IEEE）开发了一个独立的UNIX标准，这个新的ANSI UNIX标准被称为计算机环境的可移植性操作系统界面（PSOIX）。

现有大部分Unix和流行版本都是遵循POSIX标准的，而Linux从一开始就遵循POSIX标准。

（2）BSD并不是没有涉足单机内的进程间通信（socket本身就可以用于单机内的进程间通信）。

事实上，很多UNIX版本的单机IPC留有BSD的痕迹，如4.4BSD支持的匿名内存映射、4.3+BSD对可靠信号语义的实现等。

图4.15表示出了Linux所支持的各种IPC手段，在本文接下来的讨论中，为了避免概念上的混淆，在尽可能少提及UNIX的各个版本的情况下，所有问题的讨论最终都会归结到Linux环境下的进程间通信上来。

对于Linux所支持通信手段的不同实现版本（如对于共享内存来说，有Posix共享内存区以及System V共享内存区两个实现版本），将主要介绍System V API，在4.5节将介绍Posix的信号量机制。

至此，可以总结出Linux下进程间通信的几种主要方式：（1）管道（Pipe）及有名管道（named pipe）：管道可用于具有亲缘关系进程间的通信，有名管道克服了管道没有名字的限制，因此，除具有管道所具有的功能外，它还允许无亲缘关系进程间的通信。

编辑推荐

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>