

<<计算机操作系统实验与实践>>

图书基本信息

书名：<<计算机操作系统实验与实践>>

13位ISBN编号：9787302228950

10位ISBN编号：7302228957

出版时间：2010-9

出版时间：清华大学出版社

作者：秦明，李波 编著

页数：270

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>

<<计算机操作系统实验与实践>>

前言

“ 计算机操作系统 ” 是一门重要的专业基础课，也是计算机专业的学位课和考研课程。

该课程涉及概念较多，内容抽象，很多初学者常感到不易理解。

想要学好该课程，实验环节尤为重要。

选择一本适合的实验教材，对于学习者非常重要。

我们认为，教材并非难度大、有深度、面面俱到就是最好；教材应该和学习者的基础和层次紧密相关。

目前关于操作系统的实验教材不多，尤其是同时提供Linux和windows两种主流操作系统实验内容的实验教材更少。

我们希望能够拥有一本适合学生层次，同时具备windows和Linux两种主流操作系统内容的实验教材，该教材能够对学生的实验方向有所指引、对实验过程有适当提示和引导，同时，也能够充分调动学生的积极性、发挥他们的主观能动性。

基于以上认识，我们编写了本教材。

本教材是与一般高校计算机专业（本科生）操作系统原理这门专业课相配套的实验指导书，主要介绍了基于Windows操作系统内核和Linux操作系统内核的操作系统理论课程实验和课程设计的主要实验内容，对操作系统的5大管理功能——进程管理、资源管理、存储管理、设备管理、文件管理进行了各种模拟实验，通过这些模拟实验，学生能更深入地理解操作系统管理计算机各类资源的方法，对操作系统原理有更深刻的认识和体会。

本教材部分内容已在教学实践中多次内部使用，读者对象定位在一般高校本科生，注重应用，语言通俗易懂。

为了使读者能更好地理解这些实验的目的，本书还对大多数实验的实验结果进行深入分析和总结，力图使学生通过各个实验来更好地理解操作系统管理计算机系统各类资源的机理。

与一般教材不同，本教材对实验进行启发式引导，并不给出全部参考源代码，而是给出实验步骤提示。

<<计算机操作系统实验与实践>>

内容概要

“ 计算机操作系统 ” 是一门非常重要的专业课程，本教材所介绍的 “ 基于Windows和Linux内核的操作系统实验体系 ” 可使学生对于目前应用面最广的Windows和Linux操作系统的内部原理有更深入的理解。

整个实验体系分别架构在Windows和Linux这两个内核的基础上，并按照操作系统的进程管理、资源管理、存储管理、设备管理和文件管理设计相应的模拟实验。

本教材分为上下两篇，上篇介绍基于Windows内核的围绕操作系统5大管理功能的各个实验，共5章，每章通过一个典型的实验对操作系统的其中一个管理功能进行模拟；下篇除了介绍基于Linux内核的围绕操作系统5大管理功能的各个实验之外，还介绍了Linux socket编程实验(第16章)以及Linux操作系统的预备实验(第7章和第8章)，其余各章分别围绕操作系统的5大管理功能分别介绍了相应的模拟实验。

本教材所介绍的实验几乎涵盖了操作系统所有的重要原理和机制。

本教材既可作为大专院校、计算机职业专科学校或者电子类各专业 “ 操作系统 ” 的实验教学课程教材，也可供从事计算机方向研究的专业技术人员使用。

<<计算机操作系统实验与实践>>

书籍目录

上篇 Windows操作系统实验与实践	第1章 生产者-消费者实验	1.1 实验目的	1.2 预备知识
1.2.1 基本概念	1.2.2 Windows SDK简介	1.2.3 并发程序设计	1.2.4 Windows程序设计简介
1.2.5 API函数	1.2.6 相关函数	1.3 实验内容	1.4 实验指导
1.4.1 程序流程图	1.4.2 参考部分源程序代码	1.4.3 程序运行结果	1.4.4 简要分析和说明
1.4.5 深入探讨和研究	第2章 进程死锁模拟实验	2.1 实验目的	2.2 预备知识
2.2.1 基本概念	2.2.2 句柄简介	2.2.3 内核对象简介	2.2.4 Windows API简介
2.2.5 相关函数	2.3 实验内容	2.4 实验指导	2.4.1 程序流程图
2.4.2 参考程序源代码	2.4.3 程序运行结果及简要分析	2.4.4 深入探讨和研究	第3章 页式存储管理部分功能模拟实验
3.1 实验目的	3.2 基本概念	3.2.1 虚拟存储器	3.2.2 程序的局部性原理
3.3 实验内容	3.4 实验指导	3.4.1 实验步骤	3.4.2 程序流程图
3.4.3 参考部分源程序代码	3.4.4 程序运行结果及简要分析	3.4.5 深入探讨和研究	第4章 设备管理模拟实验
4.1 实验目的	4.2 预备知识	4.2.1 基本概念	4.2.2 设备处理程序
4.2.3 Windows设备驱动程序简介	4.2.4 相关函数	4.3 实验内容	4.4 参考源程序代码
第5章 文件管理模拟实验	5.1 实验目的	5.2 预备知识	5.2.1 文件的组织结构
5.2.2 文件存储空间的管理	5.2.3 文件目录管理	5.2.4 文件系统的接口	5.2.5 相关函数
5.3 实验内容	5.4 实验指导	5.4.1 实现文件基本操作的源程序代码(仅供参考)	5.4.2 实现文件属性操作的源程序代码(仅供参考)
下篇 Linux操作系统实验与实践	第6章 预备知识	6.1 Red hat Linux简介	6.2 Ubuntu简介
6.2.1 安装与设置Ubuntu	6.2.2 Ubuntu常用服务安装	6.3 Linux的特性	6.4 Linux操作系统的核心结构
6.5 Linux操作系统的特点	第7章 初步使用Linux	7.1 实验目的	7.2 实验指导
7.2.1 C语言使用简介	7.2.2 文件编辑器vi简介	7.2.3 GNU C编译器	7.2.4 gdb调试工具
7.3 实验基本原理	7.4 练习程序	第8章 系统调用	8.1 实验目的
8.2 实验基本原理	8.3 实验参考程序	8.4 提高与深化	第9章 进程创建
9.1 实验目的	9.2 实验内容	9.3 实验指导	9.3.1 进程的基本概念
9.3.2 进程映像	9.3.3 进程所涉及的系统调用	9.4 参考程序源代码	9.5 运行结果及分析
9.6 深入理解fork()	第10章 进程控制实验	10.1 实验目的	10.2 实验内容
10.3 实验指导	10.3.1 实验所涉及的系统调用	10.3.2 参考源程序代码	10.3.3 深入分析和讨论
第11章 进程互斥实验	11.1 实验目的与实验内容	11.1.1 实验目的	11.1.2 实验内容
11.2 实验指导	11.2.1 实验所涉及的系统调用	11.2.2 部分源程序代码	11.2.3 运行结果及分析
11.3 提高与深化	第12章 进程同步实验	12.1 实验目的	12.2 实验的基本原理
12.2.1 PV操作的基本概念	12.2.2 Linux的进程同步原语	12.3 参考源程序	第13章 进程通信实验
13.1 信号机制	13.1.1 实验目的	13.1.2 实验内容	13.1.3 实验指导
13.1.4 本次实验所涉及的中断调用	13.1.5 部分源程序代码	13.1.6 源程序分析及思考	13.2 进程管道通信
13.2.1 实验目的	13.2.2 实验内容	13.2.3 实验指导	13.2.4 部分源程序代码及结果分析
13.3 消息的发送与接收	13.3.1 实验目的	13.3.2 实验内容	13.3.3 实验指导
13.3.4 部分源程序代码	13.3.5 运行结果简要分析和说明	13.4 共享存储区通信	13.4.1 实验目的
13.4.2 实验内容	13.4.3 实验指导	13.4.4 部分源程序代码	13.4.5 源程序运行结果简单分析
第14章 存储管理一页面置换算法模拟	14.1 实验目的	14.2 实验基本原理	14.3 实验内容
14.4 实验前的准备工作	14.5 实验指导	14.5.1 虚拟存储系统	14.5.2 页面置换算法
14.5.3 参考源程序代码	14.5.4 该程序运行结果及对该结果的简要分析	第15章 虚拟文件系统模拟实验	15.1 实验目的
15.2 实验基本原理	15.3 实验指导	15.3.1 参考源程序代码	15.3.2 源程序的简要说明
第16章 Linux socket编程实验	16.1 实验目的	16.2 实验基本原理	16.2.1 socket的基本概念
16.2.2 socket的建立	16.2.3 socket的配置	16.3 参考源程序代码	第17章 Linux设备驱动程序
17.1 实验目的	17.2 实验基本原理	附录A Windows API函数清单	附录B Linux常用命令简介
B1 Linux的登录与退出	B2 Linux命令格式	B3 常用命令	B4 用cat命令查看/proc动态文件系统目录下的文件, 辨识其中的系统信息
附录C Linux部分源代码	附录D 操作系统实验报告要求参考文献		

章节摘录

插图：线程（thread）-有时称轻量级进程（Light Weight Process，LWP），是一个CPU调度单位，是进程中的一个执行路径。

它由线程ID、程序计数器、寄存器集合和堆栈组成。

它与同属于一个进程的其他线程共享其代码段、数据段和其他操作系统的资源（如打开文件和信号）

。在引入线程的操作系统中，进程是作为独立分配资源的单位，线程是进程的一个实体，是被独立调度和分派处理机的基本单位。

由于同一个进程内的多个线程都可以访问进程的所有资源，因此，线程之间的通信要比进程之间的通信方便得多；同一进程内线程的切换也因为线程的轻装而简单得多。

线程是操作系统中能够独立执行的实体（控制流），是处理器调度和分派的基本单位。

线程是进程的组成部分，每个进程内允许包含多个并发执行的实体（控制流），这就是多线程。

在多线程环境下，进程是操作系统中进行保护和资源分配的基本单位。

多线程是为了同步完成多项任务，其目的是提高资源使用效率来提高系统的效率。

线程是在同一时间需要完成多项任务的时候实现的。

每一个进程至少有一个主执行线程，它无须由用户去主动创建，是由系统自动创建的。

用户根据需要在应用程序中创建其他线程，多个线程并发地运行于同一个进程中。

一个进程中的所有线程都在该进程的虚拟地址空间中，共同使用这些虚拟地址空间、全局变量和系统资源，所以线程间的通信非常方便，多线程技术的应用也较为广泛。

多线程可以实现并行处理，避免了某项任务长时间占用CPU时间。

要说明的一点是，目前大多数的计算机都是单处理器（CPU），为了运行所有这些线程，操作系统为每个独立线程安排一些CPU时间，操作系统以轮换方式向线程提供时间片，这就给人一种假象，好像这些线程都在同时运行。

由此可见，如果两个非常活跃的线程为了抢夺对CPU的控制权，在线程切换时会消耗很多的CPU资源，反而会降低系统的性能。

这一点在多线程编程时应该注意。

Windows SDK函数支持进行多线程的程序设计，并提供了操作系统原理中的各种同步、互斥和临界区等操作。

Visual C++6.0（本书后面简称VC++）中，使用MFC类库也实现了多线程的程序设计，使得多线程编程更加方便。

<<计算机操作系统实验与实践>>

编辑推荐

《计算机操作系统实验与实践:基于Windows与Linux》：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

<<计算机操作系统实验与实践>>

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>