

<<操作系统>>

图书基本信息

<<操作系统>>

内容概要

本书系统描述操作系统原理和实现，并富含大量解决问题的算法、背景信息、真实示例和编程练习。书中例用最通用的操作系统（包括Linux、UNIX和Windows）进行讲解，有助于深化读者对操作系统原理、概念和算法的理解，本书不但适合作为高校本科专业的操作系统教材，同时也适合专业技术人员自学参考。

<<操作系统>>

作者简介

Gary NuttZ (加里·纳特) 任教于科罗拉多大学计算机系。他曾在著名的施乐PARC研究中心和贝尔实验室担任研究员。他的研究领域包括操作系统、分布式系统、小型无线局域网和协作技术等。

<<操作系统>>

书籍目录

第1章 引言	1.1 计算机与软件	1.1.1 通常的系统软件	1.1.2 资源抽象	示例：磁盘设备抽象
	1.1.3 资源共享	1.1.4 虚拟机和透明资源共享	1.1.5 显式资源共享	1.2 操作系统策略
1.2.1 批处理系统	1.2.2 用户的观点	1.2.3 批处理技术	示例：批处理文件	1.2.4 分时系统
	1.2.5 用户的观点	1.2.6 分时技术	示例：UNIX分时系统	1.2.7 个人计算机和 workstation
1.2.8 用户的观点	1.2.9 操作系统技术	1.2.10 对现代操作系统技术的贡献	示例：微软Windows操作系统家族	1.2.11 嵌入式系统
	1.2.12 用户的观点	1.2.13 操作系统技术	1.2.14 对现代操作系统技术的贡献	示例：VxWorks
	1.2.15 小型通信计算机	1.2.16 用户的观点	1.2.17 操作系统技术	示例：Windows CE (Pocket PC)
	1.2.18 网络	1.2.19 现代操作系统的起源	1.3 小结	
1.4 习题	第2章 使用操作系统	2.1 程序员看到的虚拟机	2.1.1 顺序计算	2.1.2 多线程计算
2.2 资源	2.2.1 使用文件	示例：POSIX文件	示例：Windows文件	2.2.2 使用其他资源
2.3 进程和线程	2.3.1 创建进程和线程	2.3.2 FORK()、JOIN()和QUIT()：历史的观点	示例：使用FORK()、JOIN()和QUIT()	2.3.3 经典的进程创建
	2.3.4 现代进程和线程的创建	2.4 并发程序的编写	2.4.1 多单线程进程：UNIX模型	示例：在UNIX系统中执行命令
	2.4.2 多进程和进程中的多线程：Windows模型	示例：创建Windows进程	2.5 对象	2.6 小结
	2.7 习题	实验2.1：一个简单的shell	实验2.2：一个多线程的应用程序	第3章 操作系统的组织结构
3.1 基本功能	3.1.1 设备管理	3.1.2 进程、线程和资源管理	3.1.3 存储管理	3.1.4 文件管理
3.2 一般实现考虑	3.2.1 性能	3.2.2 资源独占性使用	3.2.3 处理器模式	3.2.4 内核
3.2.5 请求获得操作系统服务	3.2.6 软件模块化	3.3 当代的操作系统内核	3.3.1 UNIX内核	示例：Linux
3.3.2 Windows NT执行体和内核	3.4 小结	3.5 习题	实验3.1：观察操作系统的行为	第4章 计算机组织结构
4.1 冯·诺依曼体系结构	4.1.1 冯·诺依曼体系结构的发展	4.1.2 基本思想	4.2 中央处理单元	4.2.1 算术逻辑运算单元
4.2.2 控制单元	4.2.3 处理器的实现	4.3 主存储器	4.4 I/O设备	4.4.1 设备控制器
4.4.2 直接内存访问	4.4.3 存储映射I/O	4.5 中断	4.6 当代传统计算机	4.7 移动计算机
4.7.1 片上系统技术	4.7.2 电源管理	示例：Itsy移动计算机	4.8 多处理机和并行计算机	4.8.1 并行指令执行
4.8.2 阵列处理机	4.8.3 共享内存多处理机	4.8.4 分布式存储多处理机	4.8.5 工作站网络	4.9 小结
4.10 习题	第5章 设备管理	5.1 I/O系统	5.1.1 设备管理器抽象	5.1.2 在应用程序内I/O与处理器的交迭执行
5.1.3 多个线程间的I/O?处理器交迭执行	5.2 I/O策略	5.2.1 使用轮询的直接I/O	5.2.2 中断驱动I/O	5.2.3 中断I/O与轮询I/O的性能比较
5.3 设备管理器设计	5.3.1 设备相关的驱动程序基础框架	5.3.2 服务中断	示例：Linux设备I/O	5.4 缓冲
5.5 不同种类设备的特征	5.5.1 通信设备	示例：异步串行设备	5.5.2 顺序访问存储设备	示例：传统磁带
5.5.3 随机访问存储设备	示例：磁盘	示例：磁盘访问优化	示例：CD-ROM和DVD	5.6 小结
5.7 习题	实验5.1：软盘驱动程序	第6章 进程、线程和资源的实现	6.1 手边的任务	6.1.1 经典进程的虚拟机
6.1.2 支持现代进程和线程	6.1.3 资源	6.1.4 进程地址空间	6.1.5 操作系统家族	6.1.6 进程管理器的任务
6.2 硬件进程	6.3 虚拟机接口	6.4 进程抽象	示例：Linux进程描述表	示例：Windows NT/2000/XP进程描述表
6.5 线程抽象	示例：Linux线程描述表	示例：Windows NT/2000/XP线程描述表	6.6 状态图	示例：UNIX状态图
6.7 资源管理器	6.8 概括进程管理策略	6.8.1 精化进程管理器	6.8.2 专用的资源分配策略	6.9 小结
6.10 习题	实验6.1：内核计时器	实验6.2：操纵内核对象	第7章 调度	7.1 概述
7.2 调度机制	7.2.1 进程调度程序组织	7.2.2 保存上下文	7.2.3 自愿的CPU共享	7.2.4 非自愿的CPU共享
7.2.5 性能	7.3 策略选择	7.3.1 调度程序的特征	7.3.2 一个调度研究模型	示例：分解一个进程成多个小进程
7.4 非剥夺策略	示例：估计系统负载	7.4.1 先来先服务	示例：预测FCFS的等待时间	7.4.2 最短作业优先
7.4.3 优先级调度	7.4.4 期限调度	7.5 剥夺式策略	7.5.1 轮转	7.5.2 多级队列
7.6 调度程序的实现	示例：Linux调度机制	示例：BSD UNIX中的调度策略	示例：Windows NT/2000/XP中的线程调度	7.7 小结
7.8 习题	实验7.1：分析RR调度	第8章 基本同步原理	8.1 协作进程	8.1.1 临界区
8.1.2 死锁	8.1.3 资源共享	8.2 经典解决办法的改进	8.3 信号量：现代解决方法的基础	8.3.1 操作原理
示例：使用信号量	8.3.2 应用中要考虑			

<<操作系统>>

的因素 8.4 共享存储的多处理机中的同步 8.5 小结 8.6 习题 实验8.1：有限缓冲区问题第9章 高级同步技术与进程间通信 9.1 可选的同步原语 9.1.1 AND同步 示例：使用AND同步来解决哲学家就餐问题 9.1.2 事件 示例：使用通用事件 示例：Windows NT/2000/XP中的分派对象 9.2 管程 9.2.1 操作原理 9.2.2 条件变量 示例：使用管程 9.2.3 使用管程的一些实际状况 9.3 进程间通信 9.3.1 管道模型 9.3.2 消息传递机制 9.3.3 信箱 9.3.4 消息协议 9.3.5 使用send()和receive()操作 示例：同步的IPC 9.3.6 延迟的消息拷贝 9.4 小结 9.5 习题 实验9.1：使用管道 实验9.2：精炼shell程序第10章 死锁 10.1 背景 10.1.1 死锁预防 10.1.2 死锁避免 10.1.3 死锁检测和恢复 10.1.4 人工死锁管理 10.2 一个系统死锁模型 示例：单个资源类型 10.3 死锁预防 10.3.1 占有并等待 10.3.2 循环等待 10.3.3 允许剥夺 10.4 死锁避免 示例：使用银行家算法 10.5 死锁检测和恢复 10.5.1 顺序可重用资源 示例：顺序可重用资源图 10.5.2 可消费资源 10.5.3 一般资源系统 10.5.4 恢复 10.6 小结 10.7 习题第11章 存储管理 11.1 基本知识 11.2 地址空间抽象 11.2.1 管理地址空间 示例：静态地址绑定 11.2.2 用于数据结构的动态存储 11.2.3 现代存储绑定 11.3 主存分配 11.3.1 固定分区存储分配策略 11.3.2 可变分区存储分配策略 示例：移动程序的开销 11.3.3 现代存储分配策略 11.4 动态地址空间绑定 11.5 现代存储管理器策略 11.5.1 交换 11.5.2 虚拟存储器 示例：使用高速缓存存储器 11.5.3 共享存储器的多处理机 11.6 小结 11.7 习题 实验11.1：使用共享存储器第12章 虚拟存储器 12.1 概述 12.2 地址转换 12.2.1 地址空间映射 12.2.2 段式和页式 12.3 页式 示例：当代的页表实现 12.4 静态页面调度算法 12.4.1 取策略 12.4.2 请求调页算法 12.4.3 栈算法 12.4.4 实现LRU 12.4.5 页面调度性能 12.5 动态页面调度算法 12.5.1 驻留集算法 示例：驻留集算法 12.5.2 驻留集算法的实现 示例：利用分页实现IPC 示例：Windows NT/2000/XP虚拟存储器 示例：Linux虚拟存储器 12.6 段式 12.6.1 地址转换 12.6.2 实现 示例：Multics段式系统 12.7 存储映射文件 12.8 小结 12.9 习题 实验12.1：存储映射文件第13章 文件管理 13.1 概述 13.2 文件 13.2.1 低级文件 13.2.2 结构化文件 13.2.3 数据库管理系统 13.2.4 多媒体存储 13.3 低级文件实现 13.3.1 open()和close()操作 示例：UNIX中的open和close操作 13.3.2 块管理 示例：UNIX文件结构 示例：DOS下的FAT文件系统 13.3.3 读、写字节流 13.4 支持高级文件抽象 13.4.1 结构化顺序文件 13.4.2 索引顺序文件 13.4.3 数据库管理系统 13.4.4 多媒体文档 13.5 目录 示例：几个目录例子 13.6 目录实现 13.6.1 目录项 13.6.2 打开一个文件 13.7 文件系统 示例：ISO 9660文件系统 13.7.1 安装文件系统 13.7.2 异构文件系统 13.8 小结 13.9 习题 实验13.1：一个简单的文件管理器第14章 保护和安全的 14.1 问题 14.1.1 目标 14.1.2 策略和机制 14.1.3 保护和安全的上下文 14.1.4 保护机制的开销 14.2 认证 14.2.1 外部用户认证 示例：Windows NT/2000/XP用户认证 14.2.2 内部的线程/进程认证 14.2.3 网络中的认证 14.2.4 软件认证 14.3 授权 14.3.1 特别的授权机制 14.3.2 授权的通用模型 14.3.3 实现安全策略 14.3.4 实现通用的授权机制 14.3.5 保护域 14.3.6 访问矩阵的实现 14.4 密码技术 14.4.1 概述 14.4.2 私有密钥加密技术 14.4.3 公开密钥加密技术 示例：PGP 14.4.4 Internet信息发送 14.5 小结 14.6 习题第15章 网络 15.1 从计算机通信到网络 15.1.1 交换网络 15.1.2 网络硬件需求 15.1.3 网络软件需求 15.2 ISO的OSI网络体系结构模型 15.2.1 网络协议的演变 15.2.2 ISO的OSI模型 15.3 媒体访问控制(MAC)协议 15.3.1 物理层 示例：快速物理层 15.3.2 数据链路层 15.3.3 当代网络 15.4 网络层 15.4.1 Internet寻址 15.4.2 路由 15.4.3 网络层的使用 示例：在Internet上的延迟 15.5 传输层 15.5.1 通信端口 15.5.2 数据类型 15.5.3 可靠的通信 示例：数据报和虚电路性能 15.6 使用传输层 15.6.1 命名和地址 示例：域名服务 15.6.2 客户-服务器模型 15.7 网络安全 15.7.1 传输层安全：防火墙 15.7.2 网络层安全：IPsec 15.8 小结 15.9 习题 实验15.1：使用TCP/IP协议 示例：WinSock包第16章 远程文件 16.1 通过网络共享信息 16.1.1 显式的文件复制系统 16.1.2 无缝文件系统接口 16.1.3 工作分布 16.2 远程磁盘系统 16.2.1 远程磁盘操作 16.2.2 性能因素 16.2.3 可靠性 16.2.4 远程磁盘的未来 16.3 远程文件系统 16.3.1 通用的体系结构 16.3.2 块高速缓存 16.3.3 失效后的恢复 16.4 文件级高速缓存 16.4.1 Andrew文件系统 16.4.2 LOCUS文件系统 16.5 目录系统及其实现 16.5.1 文件名字 16.5.2 打开一个文件 16.6

<<操作系统>>

小结 16.7 习题第17章 分布式计算 17.1 分布式操作系统机制 17.2 分布式主存 17.2.1 远程存储器 示例：Linda程序设计语言 17.2.2 分布式共享存储器 17.3 远程过程调用 17.3.1 RPC如何工作 17.3.2 实现RPC 17.4 远程对象 17.4.1 Emerald系统 17.4.2 CORBA 17.4.3 Java远程对象 17.5 分布式进程管理 17.5.1 通用的进程管理 17.5.2 进程和线程创建 17.5.3 调度 17.5.4 迁移和负载平衡 17.5.5 分布式同步 17.6 小结 17.7 习题 实验17.1：使用远程过程调用
 第18章 分布式程序设计运行时系统 18.1 用中间件来支持分布式软件 18.2 传统的分布式应用程序 18.3 经典分布式程序设计的中间件支持 18.3.1 PVM 18.3.2 Beowulf集群计算环境 18.3.3 OSF分布式计算环境 18.4 Web上的分布式程序设计 18.5 移动代码的中间件支持 18.5.1 Java和Java虚拟机 18.5.2 ECMA?335通用语言基础设施 18.6 小结 18.7 习题第19章 设计策略 19.1 设计考虑 19.1.1 性能 19.1.2 可信软件 19.1.3 模块化 19.1.4 可移植性 19.2 单一内核 示例：MS?DOS 示例：UNIX内核 19.3 模块化组织结构 示例：Choices——面向对象的操作系统 19.4 可扩展内核或微内核组织结构 示例：Mach操作系统 19.5 分层的组织结构 19.6 用于分布式系统的操作系统 19.6.1 网络操作系统 示例：BSD UNIX 19.6.2 分布式操作系统 示例：CHORUS操作系统 19.7 小结 19.8 习题第20章 Linux内核 20.1 Linux内核 20.2 内核组织结构 20.2.1 使用内核服务 20.2.2 守护进程 20.2.3 启动内核 20.2.4 机器中的控制流 20.3 模块和设备管理 20.3.1 模块组织 20.3.2 模块的安装和移除 20.4 进程和资源管理 20.4.1 运行进程管理器 20.4.2 创建一个新任务 20.4.3 IPC和同步 20.4.4 调度程序 20.5 存储管理器 20.5.1 虚拟地址空间 20.5.2 缺页处理程序 20.6 文件管理 20.7 小结第21章 Windows NT/*!2000/*!XP内核 21.1 概述 21.2 NT内核 21.2.1 对象 21.2.2 线程 21.2.3 多处理机同步 21.2.4 自陷、中断和异常 21.3 NT执行体 21.3.1 对象管理器 21.3.2 进程和线程管理器 21.3.3 虚拟存储管理器 21.3.4 I/O管理器 21.3.5 高速缓存管理器 21.4 内核本地过程调用和IPC 21.5 子系统 21.6 小结术语表参考文献

<<操作系统>>

媒体关注与评论

因为在介绍现代操作系统的基本原理的同时，使用来自Linux、UNIX和Windows的实例进行实践，从而广受好评。

本书有助于深化读者对当代操作系统的理解和应用。

在第3版中，作者对操作系统的原理的介绍覆盖面更广，并让读者有更多的机会来实践现实世界的例子。

第3版中的新内容 使用最通用的操作系统作为原理举例及实验环境，包括Linux、UNIX和Windows。

包含了更多的实验！

比前一版本的例子要多一倍，给学生很多实际操作Linux、UNIX和Windows的机会。

加入或更新了以下信息： 手持和无线系统 安全 线程，包括UNIX和Windows线程 SMP/多处理机 存储媒体，包括DVD和其他ISO 9000设备

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>