

<<操作系统原理>>

图书基本信息

书名：<<操作系统原理>>

13位ISBN编号：9787111306238

10位ISBN编号：7111306236

出版时间：2010-7

出版时间：机械工业出版社

作者：孟庆昌

页数：268

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>

<<操作系统原理>>

前言

我国高等学校计算机专业建立于20世纪50年代。

经过近60年的迅速发展，经历了从精英化教育到大众化教育的发展阶段，目前在校生多达40余万人，已成为我国规模最大的理工科专业，为国家建设培养了大批信息技术人才。

2006年，教育部计算机科学与技术专业教学指导委员会发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》），提出了以“按培养规格分类”为核心思想的专业发展建议，把计算机专业人才划分为研究型、工程型、应用型三种不同类型。

在《规范》的方针指导下，培养合格的计算机本科人才。

教育包括知识、能力、素质三个方面。

知识是基础、载体和表现形式，能力是技能化的知识及其综合体现，素质是知识和能力的升华。

专业教育不仅要重视知识的传授，更应突出专业能力的培养，实施能力导向的教育。

如何以知识为载体实现能力的培养和素质的提高，特别是实现专业能力和素质的提高是非常重要的。

对计算机专业本科教育而言，要想实现能力导向的教育，首先要分析专业能力的构成并考虑如何将其培养落实到教学实践中。

为此，教育部高等学校计算机科学与技术专业教学指导委员会开展了计算机科学与技术专业专业能力（简称为计算机专业能力）的培养研究。

该项研究明确计算机专业本科人才应具有的四大基本能力—计算思维能力、算法设计与分析能力、程序设计与实现能力、系统能力，并将这四大基本能力分解为82个能力点，探讨如何面对不同类型学生的教育需求，在教学活动中进行落实。

针对计算机应用型人才的培养，由于其培养数量巨大、社会需求广泛和多样化，所以培养应用型人才的专业能力在具体教学实践上有其自身的特点。

计算机应用型人才的培养目标是为国家、企事业信息系统的建设与运行培养信息化技术型人才。

本类型人才应能承担信息化建设的核心任务，掌握各种计算机软、硬件系统的性能，善于进行系统的集成和配置，有能力管理和维护复杂信息系统的运行，研究如何实现服务及方便有效地利用系统进行计算等。

计算机应用型人才的培养凸显了职业特征，使企业与学校的合作更加紧密，部分课程设置凸显能力培养特征，教学模式也呈现了职业化趋势。

为体现研究成果在教学活动中的实现，我们根据《高等学校计算机科学与技术专业专业能力构成与培养》和计算机应用型人才培养的特点和社会需求出版了这套教材。

本套教材面向高等院校计算机应用型人才培养从知识传授向能力培养转型的需求，在内容的选择、体系安排和教学方法按照专业能力和职业特征的需要进行了探索和诠释。

<<操作系统原理>>

内容概要

本书全面系统地介绍现代操作系统的基本知识和最新技术，并以Linux系统为背景介绍具体实现。全书共分7章：第1章概述操作系统的定义、功能、特征、主要类型以及系统结构；第2章至第6章分别讲述进程管理、处理机调度、存储管理、文件系统及设备管理；第7章简述现代操作系统的发展和安全生产保护机制。

为强化操作系统课程的实践环节，在附录A中给出7个实验指导；附录B给出部分习题参考答案；附录C和D分别给出Linux常用系统调用和常用命令，供教师和学生参考。

本书可作为大学本科及专科计算机科学与技术专业以及相关专业的教材，也可作为其他专业或成人教育的参考书，以及计算机工作者的自学用书。

书籍目录

出版者的话丛书序言丛书编委会前言第1章 操作系统概述 1.1 操作系统的概念 1.1.1 计算机硬件结构 1.1.2 系统初启一般过程 1.1.3 什么是操作系统 1.1.4 操作系统的目标和地位 1.1.5 操作系统的特征和服务 1.2 操作系统的主要功能 1.3 操作系统的形成和基本类型 1.3.1 操作系统的形成和发展 1.3.2 操作系统的基本类型 1.4 操作系统的主要结构 1.5 UNIX和Linux系统的核心结构 小结 习题第2章 进程管理 2.1 进程的概念 2.1.1 程序顺序执行的特征 2.1.2 程序并发执行及其特征 2.1.3 进程概念的引入和定义 2.2 进程状态描述及组织方式 2.2.1 进程的状态及其转换 2.2.2 进程的组成 2.2.3 进程组织方式 2.3 进程管理和有关命令 2.3.1 进程图和进程管理 2.3.2 Linux进程管理 2.3.3 有关进程操作的命令 2.3.4 有关进程管理的系统调用 2.4 线程概念 2.4.1 什么是线程 2.4.2 线程的实现方式 2.5 进程间的同步与互斥 2.5.1 进程间的关系 2.5.2 竞争条件和临界区 2.5.3 进程同步机制 2.5.4 信号量的一般应用 2.6 进程通信 2.7 管程 2.8 经典进程同步问题 2.9 死锁 2.9.1 死锁概述 2.9.2 死锁的预防 2.9.3 死锁的避免 2.9.4 死锁的检测和恢复 2.9.5 活锁和饥饿 小结 习题第3章 处理机调度 3.1 调度的作用和级别 3.2 作业调度 3.2.1 作业状态 3.2.2 作业管理和调度 3.3 进程调度 3.3.1 进程调度的功能和时机 3.3.2 两级调度模型 3.3.3 三级调度模型 3.4 调度性能的评价 3.4.1 调度策略的选择 3.4.2 性能评价标准 3.5 常用调度算法 3.6 实时调度 3.7 Linux系统中的进程调度 3.7.1 Linux进程调度方式 3.7.2 Linux常用调度命令 3.8 中断处理和系统调用 3.8.1 中断处理的一般过程 3.8.2 系统调用处理 3.9 shell基本工作原理 小结 习题第4章 存储管理 4.1 地址空间与重定位 4.1.1 用户程序的地址空间 4.1.2 重定位概念 4.1.3 对换技术 4.2 分区管理技术 4.2.1 分区法 4.2.2 可重定位分区分配 4.3 分页技术 4.3.1 分页的基本概念 4.3.2 分页系统中的地址映射 4.3.3 页的共享和保护 4.3.4 页表的构造 4.4 分段技术 4.4.1 分段的基本概念 4.4.2 分段系统中的地址映射 4.4.3 段的共享和保护 4.5 虚拟存储管理 4.5.1 虚拟存储器的概念 4.5.2 虚拟存储器的特征 4.6 请求分页技术 4.6.1 请求分页的基本思想 4.6.2 硬件支持及缺页处理 4.6.3 页面置换算法 4.7 内存块分配和抖动问题 4.7.1 内存块分配 4.7.2 抖动问题 4.7.3 工作集 4.8 段式虚拟存储器 4.8.1 基本工作过程 4.8.2 动态链接和链接中断处理 4.9 段页式结合系统 4.10 Linux系统的存储管理技术 4.10.1 对换 4.10.2 请求分页技术 小结 习题第5章 文件系统 5.1 文件系统概述 5.1.1 文件及其分类 5.1.2 文件系统的功能 5.2 文件的逻辑组织和物理组织 5.2.1 文件的逻辑组织 5.2.2 用户对文件的存取方法 5.2.3 文件的物理组织 5.3 目录文件 5.3.1 文件控制块和文件目录 5.3.2 目录结构 5.4 文件存储空间的管理 5.5 文件系统的可靠性 5.5.1 坏块管理 5.5.2 文件的后备和恢复 5.5.3 文件系统的一致性 5.6 文件共享和保护 5.6.1 文件共享 5.6.2 文件保护 5.7 Linux文件系统 5.7.1 文件系统的格式 5.7.2 虚拟文件系统 5.7.3 管道文件 5.7.4 对文件的主要操作 小结 习题第6章 设备管理 6.1 设备管理概述 6.1.1 设备分类和标识 6.1.2 I/O系统结构 6.1.3 直接存储器访问方式 6.1.4 缓冲技术 6.1.5 设备管理的功能 6.2 设备分配技术 6.2.1 设备分配技术和算法 6.2.2 SPOoLing系统 6.3 I/O软件构造原则 6.3.1 I/O软件目标 6.3.2 设备驱动程序 6.3.3 与设备无关的I/O软件 6.3.4 用户空间I/O软件 6.3.5 处理输入输出请求的步骤 6.4 磁盘调度和管理 6.4.1 磁盘硬件 6.4.2 磁盘调度算法 6.5 Linux系统设备管理 6.5.1 设备管理概述 6.5.2 设备驱动程序与内核间的接口 小结 习题第7章 操作系统的发展和安全性 7.1 现代操作系统发展概述 7.1.1 推动操作系统发展的动力 7.1.2 现代操作系统的发展 7.2 系统安全性 7.2.1 信息安全问题 7.2.2 一般性安全机制 7.2.3 保护机制 7.3 系统性能评价 小结 习题附录A实验指导附录B 部分习题参考答案附录C Linux常用系统调用附录D Linux常用命令参考文献

<<操作系统原理>>

章节摘录

插图：另一个用户程序读取数据，并且可以在终端或打印机上显示第三个用户程序的结果。

在多道程序设计系统中，CPU在各程序之间来回进行切换：在这个程序上运行一会儿（如几十或几百毫秒），在另一个程序上再运行一会儿。

也就是说，各个程序是并发执行的。

由于多道程序并发执行时共享系统资源，共同决定这些资源的状态，因此系统中各程序在执行过程中就出现了相互制约的新关系，程序的执行出现“走走停停”的新状态。

这些都是程序的动态过程中发生的。

而程序本身是机器能够翻译或执行的一组动作或指令，或者写在纸面上，或者存放在磁盘等介质上，是静止的。

很显然，直接从程序的字面上无法看出它什么时候运行、什么时候停顿，也看不出它是否影响其他程序或者一定受其他程序的影响。

综上所述，用程序这个静态概念已不能如实反映程序并发执行过程中的这些特征。

为此，人们引入“进程”（process）这一概念来描述程序动态执行过程的性质。

2.进程的定義 进程（或任务）是在20世纪60年代中期由美国麻省理工学院（MIT）的J.H.Saltzer首先提出的，并在所研制的MULTICS系统上实现。

IBM公司把进程叫做任务（task），并在TSS/360系统中实现了。

进程是操作系统的最基本、最重要的概念之一，是对正在运行程序的抽象。

引进这个概念对于理解、描述和设计操作系统都具有极其重要的意义。

但是，迄今为止，对进程概念还没有形成统一的定义，可以从不同的角度来描述它的基本特征。

进程最根本的属性是动态性和并发性。

我们将进程定义为：程序在并发环境中的执行过程。

为了说明进程和程序的关系，我们举一个生活中的事例作比喻：假如你正在按照菜谱上的指导来烹饪，厨房里有鱼、肉、鸡蛋、油、盐和各种调料。

菜谱就相当于程序，你就相当于处理器（CPU，各种原料就相当于数据，你按照菜谱上的指令一步步地加工，这一系列动作的总和就是进程。

如果你在你切菜时，电话铃响了，你会停下手中的活（保留现场），然后按规定的步骤接听电话：拿起话机、打招呼、交谈，最后挂上电话。

这一系列步骤是程序，打电话的整个过程是进程。

之后，你接着做菜。

这样，CPU就在进程间实施切换。

可以看出，进程和程序有密切的关系，但又是两个完全不同的概念，它们在以下4个方面有重要区别。

程序是静态、被动的概念，本身可以作为一种软件资源长期保存。

进程是程序的一次执行过程，是动态、主动的概念，有一定的生命期，会动态地产生和消亡。

例如，从键盘上输入一条命令：\$date则系统就会针对这条命令创建一个进程，这个进程执行date命令所对应的程序（以可执行文件的形式存放在系统所用的磁盘上）。

当工作完成后，显示出当前日期和时间，这个进程就终止了，并从系统中消失，而date命令所对应的程序仍保留在磁盘上。

传统的进程是一个独立运行的单位，能与其他进程并发执行。

进程是作为资源申请和调度单位存在的。

通常的程序是不能作为一个独立运行的单位而并发执行的。

程序在CPU上才能真正地执行。

系统中以进程为单位进行CPU的分配，因为进程不仅包括相应的程序和数据，还有一系列描述其活动情况的数据结构。

系统中的调度程序能够根据各个进程当时的状况，从中选出一个最适合运行的进程，将CPU控制权交

<<操作系统原理>>

给它，令其运行。

而程序是静态的，系统无法区分内存中的哪一个程序更适合运行，所以，程序不能作为独立的运行单位。

<<操作系统原理>>

编辑推荐

《操作系统原理》按照工程应用型高等院校“操作系统”课程的教学要求和IT行业对人才的需求，并参考研究入学考试大纲编写而成。

书中以Linux系统为实例，本着重基础、重能力、求创新、凸显职业性的总体思想，全面系统地讲述现代操作系统的关键技术与实现方法。

《操作系统原理》特点：突出基础、强调实用，讲清基本概念，讲透核心技术，讲明典型实现。既讲解经典理论，又介绍最新开发应用技术。

在讲授方法上由浅入深，由表及里，先引出问题，再给出概念、实现技术和典型算法。

强化实践环节，每章后面给出大量习题，并在附录中给出7个上机实验指导。

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>