

## <<计算机操作系统>>

### 图书基本信息

书名：<<计算机操作系统>>

13位ISBN编号：9787302245636

10位ISBN编号：7302245630

出版时间：2011-6

出版时间：清华大学出版社

作者：何炎祥 等编著

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>

## &lt;&lt;计算机操作系统&gt;&gt;

## 前言

FOREWORD操作系统是计算机系统中最关键的系统软件，计算机系统愈复杂，操作系统的作用和地位就愈重要。

本书结合现在操作系统的设计并考虑操作系统的发展方向，着重讨论操作系统设计的基本概念、基本原理和典型技术。

全书共分12章，下面介绍各章内容组成。

第1章简要介绍操作系统的基本概念、功能、发展历史以及主要成就等。

第2章介绍了进程的概念，以及操作系统对进程进行控制和管理时采用的数据结构，还讨论了与进程相关的线程等内容。

第3章介绍了在单一系统中并行处理的关键技术--互斥和同步机制。

第4章描述了死锁的性质，并讨论了解决死锁问题的一些方法。

第5章讨论了多种内存管理方法，并讨论了用于支撑虚拟内存所需的硬件结构和操作系统用来管理虚拟内存的软件方法。

第6章分析了各种不同的进程调度方法，包括实时调度策略等方面的内容。

第7章论述了操作系统对输入/输出设备的控制和管理，尤其是对系统性能影响较大的磁盘I/O的调度和控制。

第8章对文件的组织、存储、使用和保护等方面的内容进行了讲解。

第9章和第10章描述了分布式操作系统的一些关键设计领域，包括Client/Server结构，用于消息传递和远程过程调用的分布式通信机制、分布式进程迁移、中间件以及解决分布式互斥和死锁问题的原理与技术。

第11章简要讨论了保证操作系统安全性的相关理论和方法。

第12章以小型操作系统MINIX为例，用解释性的方式介绍了MINIX设计和实现的具体过程，以期达到理论联系实际，学以致用，突出实践性的目的。

本书由何炎祥、李飞、李宁共同编写，何炎祥统编了全书。

在编写过程中得到了武汉大学计算机学院领导和同事们热情帮助，清华大学出版社为本书的出版给予了大力支持，文中还参考、引用了国内外一些专家学者的论著和研究工作，以及一些公司的产品介绍，在此一并表示诚挚的感谢。

随着操作系统技术的发展，本书在前一版的基础上，对部分内容进行了修订和改编，以适应教学需要。

限于水平，书中错误难免，敬请读者不吝赐教。

编者 2011年2月

## <<计算机操作系统>>

### 内容概要

本书着重讨论操作系统设计的基本概念、基本原理和典型技术，讲述构造操作系统过程中可能面临的种种问题及其解决办法；介绍操作系统设计中的一些非常重要的进展，包括线程、实时系统、多处理器调度、进程迁移、分布式计算模式、分布式进程管理、中间件技术、微核技术、操作系统的安全性等。

为了帮助读者更好地理解操作系统的概念、原理和方法，更好地将理论与实际设计相结合，笔者选择了目前具有代表性、典型性的操作系统Windows

NT、UNIX、Linux作为实例贯穿全书，并专门介绍了一个小型操作系统--MINIX的设计与实现。

目的是尽可能清晰、全面地向读者展现较新的操作系统的设计原理与基本实现技术，以便读者深入了解现在操作系统的全貌，为今后进行大型软件研制与系统开发打下坚实基础。

本书可供大专院校计算机专业及相关专业的师生作为教材，也可供在计算机软件企业和IT行业工作的科技工作者与管理者学习和参考。

## <<计算机操作系统>>

### 作者简介

何炎祥，博士、教授、博士生导师，国家教学名师，享受国务院政府特殊津贴。

武汉大学计算机学院院长。

美国Oregonl大学计算机及信息科学系硕士，武汉大学计算机科学系博士。

主要研究方向为分布并行处理、可信软件、软件工程等。

主持和主要承担包括863高技术计划项目、国家自然科学基金重大研究计划项目、国家自然科学基金面上项目、省市科技攻关项目等40余项，发表论文200多篇，出版学术著作和教材15种。

获包括湖北省科技进步一等奖在内的省部级科技成果和教学成果奖多项。

主持“编译原理”国家精品课程。

国家自然科学基金委、国家科学技术奖、教育部科学技术奖评审专家，兼任教育部科技委信息学部委员，全国高校计算机教育研究会副理事长，中国计算机学会教育专委会副主任，清华大学出版社“高等学校计算机教育规划教材”编委会主任，湖北省计算机学会副理事长，湖北省软件行业协会副理事长，《计算机研究与发展》、《计算机科学与探索》、《Intelligence Control and Automation》等杂志编委，《中国科学》、《软件学报》、《计算机学报》等审稿人，中山大学、华中师范大学、广东外语外贸大学等高校兼职教授，国家重点基础研究发展规划（973）项目“下一代互联网信息存储的组织模式与核心技术研究”和“面向复杂应用环境的数据存储系统理论与技术基础研究”项目专家组专家。多个国际学术会议的大会合作主席、程序委员会合作主席、委员等。

## &lt;&lt;计算机操作系统&gt;&gt;

## 书籍目录

## 第1章 操作系统概论

## 1.1 操作系统的作用

- 1.1.1 硬件系统和应用程序间的界面
- 1.1.2 资源管理者
- 1.1.3 推动操作系统发展的因素

## 1.2 操作系统的演变

- 1.2.1 串行处理系统
- 1.2.2 简单批处理系统
- 1.2.3 多道程序批处理系统
- 1.2.4 分时系统
- 1.2.5 实时系统
- 1.2.6 网络操作系统
- 1.2.7 分布式操作系统
- 1.2.8 嵌入式操作系统

## 1.3 操作系统的主要成就

- 1.3.1 进程
- 1.3.2 存储器管理
- 1.3.3 信息保护和安全性
- 1.3.4 调度和资源管理
- 1.3.5 系统结构

## 1.4 操作系统举例

- 1.4.1 Windows NT
- 1.4.2 UNIX System V

## 1.5 操作系统的主要研究课题

## 小结

## 习题

## 第2章 进程描述与控制

## 2.1 进程状态

- 2.1.1 进程产生和终止
- 2.1.2 进程状态模型
- 2.1.3 进程挂起

## 2.2 进程描述

- 2.2.1 操作系统控制结构
- 2.2.2 进程控制结构
- 2.2.3 进程属性

## 2.3 进程控制

- 2.3.1 执行模式
- 2.3.2 进程创建
- 2.3.3 进程切换
- 2.3.4 上下文切换
- 2.3.5 操作系统的运行
- 2.3.6 微核

## 2.4 线程和SMP

- 2.4.1 线程及其管理
- 2.4.2 多线程的实现

## <<计算机操作系统>>

2.4.3 进程与线程的关系

2.4.4 SMP

2.5 系统举例

2.5.1 UNIX System V

2.5.2 Windows NT

2.5.3 Linux

小结

习题

第3章 并发控制--互斥与同步

第4章 死锁处理

第5章 内存管理

第6章 处理机调度

第7章 I/O设备管理

第8章 文件管理

第9章 分布计算

第10章 分布式进程管理

第11章 操作系统的安全性

第12章 一个小型操作系统的实现

参考文献

## &lt;&lt;计算机操作系统&gt;&gt;

## 章节摘录

版权页：插图：系统程序设计员用于开发早期的多道程序和多用户系统的原则是中断。

任何作业在遇到规定的事件（如I/O）时就会被挂起，处理器保存某些内容（例如，程序计数器和其他寄存器），然后转到中断处理程序，中断处理程序将确定中断、处理中断，然后继续执行用户被中断的作业和其他一些作业。

设计系统软件以协调各种组件间的运作是很困难的。

在同一时刻有许多任务，而每个任务又包括大量必须依次执行的步骤，因此，对所有事件的可能的组合顺序进行分析是不可能的。

由于系统缺乏协作的方法，常会产生一些错误。

这些错误很难诊断，因为必须将它们与应用程序的出错及硬件出错区分开。

即使发现了错误，也很难确定原因，因为出现错误的精确环境是很难再现的。

通常，有4种原因会导致这样的错误：不合适的同步。

一个进程常常因等待某事件发生而必须挂起。

例如，一个程序执行初始化I/O读操作后，必须等待，直到缓冲区中有所需的数据。

不合适的信号机制可能会导致信号丢失或收到多个信号。

失败的互斥。

有些情况下，不止一个用户或程序企图同时使用共享资源。

例如，在一个航空订票系统中，两个用户可能分别试图读取数据库，如果有空余座位，就订下座位并更新数据库。

如果访问没有得到控制，就可能出现错误。

必须有一种互斥机制，即在同一时刻只允许一个进程对数据区域进行更新。

这种互斥的实现很难在所有可能的事件顺序下被证明是正确的。

非确定的程序操作。

一个特定程序的运行结果通常是由该程序的输入决定而不依赖于一个共享系统中其他程序的运行。

但是，如果多个程序共享存储器并都被处理器调入内存，就有可能修改共享存储器区，从而互相干扰。

这样，各程序的调度次序就会影响其他程序的输出结果。

死锁。

在某些情况下，可能有两个或多个程序都在互相等待彼此占用的资源。

例如，两个程序可能都需要两个I/O设备，但每个程序都控制了其中一个设备而等待另一个程序释放另一个设备。

解决这些问题所必需的基本条件是，要有一个系统的方法去监视和控制处理器中运行的程序。

进程的概念提供了这个基本条件。

进程由以下3部分组成：一个可执行的程序；该程序所需的相关数据（变量、工作空间、缓冲区等）；该程序的执行上下文（context）。

其中，最后一项是必不可少的。

操作系统中所有用来管理进程和处理器执行进程的信息都包括在执行上下文中。

这个上下文包括寄存器的内容、进程的优先级以及进程是否等待I/O事件的完成等信息。

根据上述分析，结合各种观点，我们可以把“进程”定义为：可并发执行的程序在一个数据集合上的运行过程。

图1.10显示了进程可能实现的一种方法。

两个进程A和B存在于内存中。

<<计算机操作系统>>

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>