

21世纪高等学校计算机规划教材

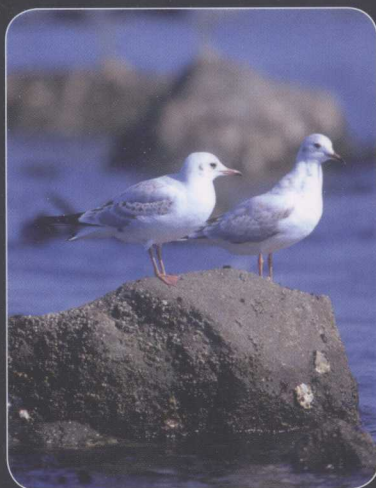
21st Century University Planned Textbooks of Computer Science

操作系统

Operating Systems

宗大华 宗涛 陈吉人 编著

- 突出操作系统的理论性，对内容精心选择和安排
- 注重操作系统的实用性，配有大量的实例和习题
- 关切操作系统的基本性，深入浅出地分析和说明



精品系列



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

操作系统

Operating Systems

宗大华 宗涛 陈吉人 编著



人民邮电出版社 精品系列

人民邮电出版社

北京

样书

专用章

图书在版编目 (CIP) 数据

操作系统 / 宗大华, 宗涛, 陈吉人编著. —北京: 人民邮电出版社, 2009.10
21世纪高等学校计算机规划教材
ISBN 978-7-115-20217-8

I. 操… II. ①宗…②宗…③陈… III. 操作系统—高等学校—教材 IV. TP316

中国版本图书馆CIP数据核字 (2009) 第154537号

内 容 提 要

本书是一部关于操作系统基本概念、管理策略及实现原理的教材, 目的是尽可能清晰、全面地介绍现代操作系统涉及的内容和技术, 展现现代操作系统的本质和特点, 跟上现代操作系统发展的速度和步伐。

全书共9章, 第1章为操作系统概述, 第2章介绍进程与线程, 第3章介绍处理机管理, 第4章和第5章介绍存储管理, 第6章介绍文件管理, 第7章介绍设备管理, 第8章介绍并发进程的设计技术, 第9章介绍死锁和系统安全。

本书内容涵盖《2009 计算机考研大纲》和《高等学校计算机科学与技术专业公共核心知识体系与课程》中有关操作系统部分的要求; 从认知的实际过程出发, 对全书的内容做了精心安排; 对于重点内容或难以理解之处, 本书或通过举例、或不惜用较多的笔墨进行说明和解释; 最后以附录的形式给出各章部分习题或详或简的参考答案。

本书可作为普通高等院校本科计算机专业及相关专业的操作系统教材和考研参考书, 也可以作为 IT 行业有关人员的自学参考书。

21 世纪高等学校计算机规划教材

操 作 系 统

- ◆ 编 著 宗大华 宗涛 陈吉人
责任编辑 滑 玉
执行编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 20.25
字数: 534 千字
印数: 1—3 000 册
- 2009 年 10 月第 1 版
2009 年 10 月河北第 1 次印刷

ISBN 978-7-115-20217-8

定价: 33.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

操作系统是加载在计算机硬件上的第一层软件,是对计算机功能的首次扩充,其重要性不言而喻,它早已成为现代计算机系统中不可或缺的一种系统软件。人们通过操作系统管理和控制系统中的所有软硬件资源,使这些资源的使用更加合理、有效,也使计算机操作更加方便、安全、快捷。

本书是一部关于操作系统基本概念、管理策略及实现原理的教材,目的是尽可能清晰、全面地介绍现代操作系统涉及的内容和技术,展现现代操作系统的本质和特点,跟上现代操作系统发展的速度和步伐。

全书共 9 章,具体安排如下。

第 1 章为操作系统概述,简要介绍操作系统的定义和功能,硬件支撑环境,常采用的结构模式,以及分类。

第 2 章和第 3 章介绍处理机管理,引入进程和线程的概念、它们之间的联系与区别、各种作业调度算法、各种进程调度算法以及实时处理与实时调度算法。

第 4 章和第 5 章介绍存储管理,包含的内容有分区、分页式、分段式、段页式、请求分页式及请求分段式等存储管理策略,涉及存储区的分配和回收、存储区的共享和保护及地址重定位的实现等。

第 6 章介绍文件管理,包含的内容有文件的逻辑结构、文件的物理结构、文件的目录结构、磁盘存储空间的管理、“按名存取”的实现过程以及文件的使用等。

第 7 章介绍设备管理,包含的内容有 I/O 系统的组织结构、I/O 的 4 种实现方式、缓冲技术、虚拟设备与 SPOOLing 技术以及磁盘的移臂调度策略。

第 8 章介绍并发进程的设计技术,同时给出了互斥与同步的概念,信号量与 P、V 操作的概念,用信号量实现互斥、同步、资源分配的方法,以及进程间的高级通信。

第 9 章介绍死锁和系统安全,包含的内容有死锁的概念和产生死锁的必要条件,死锁的预防、避免、检测与恢复,安全与保护概述,各种安全威胁及具体的安全防护措施等。

本书有如下几个特点。

(1) 本书的内容涵盖了《2009 计算机考研大纲》中有关操作系统部分的要求,同时也涵盖了《高等学校计算机科学与技术专业公共核心知识体系与课程》中有关操作系统部分的要求。

(2) 从认知的实际过程出发,对全书的内容做了精心安排。例如,把通常在处理机管理中讲述的进程并发性抽出来单独在最后成章介绍,使学生在有大量感性认识的基础上学习这些较为困难和理论性的问题,有利于难点的分散,同时也有利于学生的接受。

(3) 对于重点内容或难以理解之处,本书或通过举例、或不惜用较多的笔墨进行说明和解释。例如,对实时调度算法、工作集和管程等均通过实例做进一步的解释;再例如,对段的动态链接、互斥和同步的典型样例均给出了非常详细的

说明，并附有流程图。这样的做法有利于学生更快理解和掌握。

(4) 在本书的最后，以附录的形式给出各章部分习题或详或简的参考答案。这种必要的提示信息会给学生带来启发，从而提高他们学习的兴趣。关于本书习题的详细解答可在人民邮电出版社的教学资源与服务网 (<http://www.ptpedu.com.cn>) 免费下载。

本书是参与写作的全体同仁共同努力的结果，在此对他们的辛勤劳动和精诚合作表示真挚的感谢。他们是：王晓宇、陈新、黄芳、沈寄云、江汇、沈佳、梁发寅、宗杰、余楠、蒋玮、梁辉。

由于编者的水平有限，书中肯定会有不妥甚至错误之处。恳请广大读者批评指正！

编者

2009 年 4 月 于北京

目 录

第 1 章 操作系统概述.....1

- 1.1 操作系统的定义与功能.....1
 - 1.1.1 操作系统的定义1
 - 1.1.2 操作系统的功能3
- 1.2 操作系统的接口与内核模式.....5
 - 1.2.1 操作系统的两种接口5
 - 1.2.2 操作系统内核的结构模式9
- 1.3 操作系统的基本分类.....11
 - 1.3.1 批处理系统11
 - 1.3.2 分时系统13
 - 1.3.3 实时系统14
 - 1.3.4 网络系统16
- 1.4 硬件支撑环境.....17
 - 1.4.1 计算机硬件的基本构成17
 - 1.4.2 中断19
 - 1.4.3 存储器的结构22
- 1.5 Linux 操作系统的诞生24
 - 1.5.1 操作系统的发展简史24
 - 1.5.2 Linux 操作系统的诞生25
- 习题.....27

第 2 章 进程与线程.....29

- 2.1 进程的概念.....29
 - 2.1.1 多道程序设计环境下的程序特点29
 - 2.1.2 进程的定义32
 - 2.1.3 进程的状态及状态变迁34
- 2.2 进程的管理.....37
 - 2.2.1 进程控制块37
 - 2.2.2 进程控制块队列40
 - 2.2.3 进程控制的系统调用命令41
- 2.3 线程.....45
 - 2.3.1 线程的概念45
 - 2.3.2 线程的实现48
 - 2.3.3 线程与进程的关系49
- 2.4 Linux 的进程51

- 2.4.1 Linux 进程51
- 2.4.2 Linux 的几种链接信息54
- 2.4.3 Linux 进程的生存过程56
- 习题59

第 3 章 处理机管理.....61

- 3.1 处理机调度概述61
 - 3.1.1 处理机调度的 3 个层次61
 - 3.1.2 进程调度的功能、时机和基本策略63
 - 3.1.3 调度算法的性能评价指标64
- 3.2 作业调度算法66
 - 3.2.1 先来先服务调度算法66
 - 3.2.2 短作业优先调度算法69
 - 3.2.3 最短剩余时间优先调度算法70
 - 3.2.4 最高响应比调度算法71
- 3.3 进程调度算法73
 - 3.3.1 先来先服务调度算法73
 - 3.3.2 轮转调度算法73
 - 3.3.3 优先级调度算法75
 - 3.3.4 多级队列调度算法77
 - 3.3.5 多级反馈队列调度算法78
- 3.4 实时处理与实时调度算法79
 - 3.4.1 实时处理的特征79
 - 3.4.2 最早截止时间优先调度算法81
 - 3.4.3 速率单调调度算法82
- 3.5 Linux 的处理机调度84
 - 3.5.1 涉及调度的进程分类84
 - 3.5.2 Linux 的可运行队列85
 - 3.5.3 Linux 的进程调度算法87
- 习题89

第 4 章 基本存储管理.....92

- 4.1 存储管理预备知识92
 - 4.1.1 用户程序的 4 个处理阶段92
 - 4.1.2 地址重定位94

4.1.3 程序的链接	97	6.1.4 记录的成组与分解	166
4.1.4 存储管理的功能	98	6.2 文件的目录结构	167
4.2 分区存储管理	98	6.2.1 目录	167
4.2.1 固定分区存储管理	99	6.2.2 目录的层次结构	169
4.2.2 可变分区存储管理	102	6.2.3 路径名	171
4.2.3 其他管理技术: 覆盖、交换、 伙伴系统	109	6.3 文件在磁盘上的组织	172
4.3 分页式存储管理	111	6.3.1 磁盘存储空间的管理	172
4.3.1 分页式存储管理的基本思想	111	6.3.2 文件的物理结构	175
4.3.2 分页式存储管理的地址转换	113	6.3.3 文件的存取方式	178
4.3.3 页帧的分配与回收	117	6.3.4 “按名存取”的实现	179
4.4 分段式和段页式存储管理	119	6.4 文件的使用	180
4.4.1 分段式存储管理	119	6.4.1 文件的操作	180
4.4.2 段页式存储管理	124	6.4.2 文件共享	182
习题	125	6.4.3 文件保护	184
第 5 章 虚拟存储管理	129	6.5 Linux 的文件管理	185
5.1 请求页式虚拟存储管理基础	129	6.5.1 Linux 文件系统概述	186
5.1.1 虚拟存储器	129	6.5.2 Linux 的文件类型	188
5.1.2 请求页式虚拟存储管理	131	6.5.3 Linux 的二次扩展文件系统—— Ext2	190
5.2 请求页式的替换策略	137	6.5.4 Linux 的虚拟文件系统——VFS	194
5.2.1 替换策略综述	137	习题	197
5.2.2 请求页式静态替换策略	137	第 7 章 设备管理	199
5.2.3 关于静态替换策略的进一步讨论	139	7.1 设备管理概述	199
5.2.4 请求页式动态替换策略	144	7.1.1 I/O 系统的组织结构	199
5.3 请求段式虚拟存储管理	147	7.1.2 计算机设备的分类	201
5.3.1 请求段式虚拟存储管理	147	7.1.3 设备管理的目标与功能	205
5.3.2 段的动态链接	148	7.1.4 设备管理的数据结构	205
5.4 Linux 的存储管理	150	7.2 I/O 的 4 种实现方式	208
5.4.1 Linux 存储管理的硬件基础	150	7.2.1 程序循环控制 I/O	208
5.4.2 Linux 多级页表的地址转换	152	7.2.2 中断驱动 I/O	209
5.4.3 内存空间的管理	155	7.2.3 直接内存访问 I/O	210
5.4.4 管理虚拟存储空间的数据结构	156	7.2.4 通道管理 I/O	212
习题	158	7.3 缓冲技术、虚拟设备与 SPOOLing 技术	214
第 6 章 文件管理	161	7.3.1 I/O 缓冲	214
6.1 文件系统概述	161	7.3.2 虚拟设备与 SPOOLing 技术	217
6.1.1 文件系统概述	161	7.4 磁盘及磁盘的移臂调度策略	218
6.1.2 文件系统的功能	164	7.4.1 磁盘的格式化	218
6.1.3 文件的逻辑结构	165	7.4.2 磁盘的性能参数	221

7.4.3 磁盘的移臂调度策略	222	8.4.2 哲学家就餐问题	262
7.4.4 独立磁盘冗余阵列: RAID	225	8.4.3 理发师理发问题	266
7.5 Linux 的 I/O 管理	228	8.5 高级进程通信	268
7.5.1 Linux 设备管理综述	229	8.5.1 消息缓冲通信	268
7.5.2 Linux 对字符设备的管理	231	8.5.2 信箱通信	269
7.5.3 Linux 对块设备的管理	232	习题	270
7.5.4 Linux 的磁盘调度	233		
习题	235		
第 8 章 并发性: 互斥和同步	238	第 9 章 死锁、系统安全	273
8.1 互斥和同步	238	9.1 死锁概述	273
8.1.1 互斥和临界区	238	9.1.1 死锁的概念	273
8.1.2 同步	240	9.1.2 资源分配图	276
8.2 实现互斥的方法讨论	243	9.1.3 产生死锁的必要条件	277
8.2.1 实现互斥的硬件方法	243	9.2 死锁的预防、避免、检测与恢复	278
8.2.2 实现互斥的软件方法	244	9.2.1 死锁预防	278
8.3 信号量与 P、V 操作	246	9.2.2 死锁避免	280
8.3.1 信号量与 P、V 操作定义	246	9.2.3 死锁检测与恢复	286
8.3.2 用 P、V 操作实现互斥	247	9.3 系统的安全与保护	290
8.3.3 用 P、V 操作实现同步	250	9.3.1 安全与保护概述	290
8.3.4 用 P、V 操作实现资源分配	252	9.3.2 具体的安全威胁	296
8.3.5 管程	254	9.3.3 具体的安全防护措施	298
8.4 互斥、同步的样例分析	258	习题	302
8.4.1 读者-写者问题	258		
		附录 各章部分习题解答	305
		参考文献	316

第 1 章

操作系统概述

21 世纪的重要特征是数字化、网络化和信息化，而这些特征的出现，完全是基于 20 世纪最伟大的创造——计算机。自 1946 年第一台计算机诞生至今，其技术得到了突飞猛进的发展。目前，计算机不仅被广泛应用于科学计算、过程控制、数据处理、军事技术等领域，而且也渗透到了日常办公、教育学习、家庭生活等诸方面。计算机已成为社会信息化的重要支柱和人类文明高度发展的重要象征。

操作系统（operating system）是一系列软件的汇集。可以这么说：凡接触过计算机的人，都听说过“操作系统”这个术语，也都使用过“操作系统”这个软件，从这一点即可看出操作系统在计算机中扮演角色和所起作用的重要性。

那么，到底什么是“操作系统”？它具有哪些功能？用户如何使用它？它需要有哪些硬件方面的支持？本章将逐一解答这些问题。

本章主要讲述以下几个方面的内容。

- （1）操作系统的定义与功能。
- （2）操作系统的基本分类。
- （3）操作系统向用户提供的两种接口。
- （4）操作系统运行所需的基本硬件支撑环境。
- （5）Linux 操作系统的形成过程。

1.1 操作系统的定义与功能

1.1.1 操作系统的定义

在计算机的发展和应用过程中，人们最常听说和接触的操作系统有：磁盘操作系统 DOS、图形界面操作系统 Windows、多用户操作系统 UNIX 以及开放源代码操作系统 Linux 等。这表明，计算机科学与技术的不断创新，推动着操作系统不断地更新与改进。从另一个方面来说，也正是这种不断更新与改进，才使得操作系统至今还没有一个统一的定义和解释。

一般来说，可以从以下两种不同的观点（或角度）出发去认识和理解操作系统存在的本质。

1. 操作系统是硬机器的扩展：虚拟机的观点

通常，把未配置任何软件的计算机称为“裸机”。面对裸机，用户只能使用机器指令编写程序。这样的机器语言难懂、难记、难用，同时还要求编程者事事都要深入到计算机的硬件中，都要明

白相关设备的动作细节，都要过问输入/输出的具体琐事。

例如，在裸机环境下，没有你所熟悉的“文件”实体，所以为了在磁盘上存放数据，你必须清楚地知道磁盘的使用情况，即哪些地方已经存放有数据，哪些地方空闲可以使用；你必须指明数据具体存放在什么地方，包括哪个柱面，哪个盘面，以及哪个扇区等；你必须详细了解磁盘的各种操作，包括如何启动，如何驱动，以及如何读/写等；……于是，你的精力根本不可能集中到如何使用计算机去解决自己的实际问题，计算机本身的效率也不可能充分地得到发挥。想必任何一个用户，都不会愿意在这样的“恶劣”环境下去使用计算机。

为了能使人们从复杂的硬件环境中脱出身来，最好的解决办法就是开发一种软件，它紧紧地位于计算机硬件之上，能够把计算机底层硬件的物理特性隐藏起来，向用户提供一个可以集中精力于自身应用的、较为舒适便利的开发环境。这种扩展计算机系统功能、方便用户使用的软件集合就是所谓的“操作系统”，如图 1-1 所示。

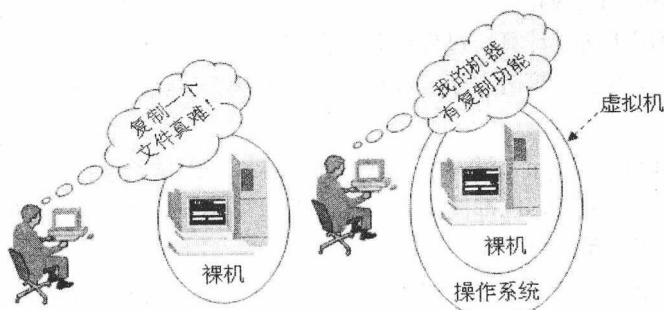


图 1-1 操作系统是硬机器的扩展

操作系统是在裸机上安装的第一层软件，它把计算机硬件与使用者隔离开。用户使用计算机时，面对的是操作系统提供的图形或命令界面，通过这种界面，他们可以方便地打开文件、读/写文件、修改目录、拷贝复制等，而无须再顾及硬件的特性和具体的动作。

裸机（也就是计算机）由输入/输出设备、存储器、运算和逻辑部件以及控制器组成。当计算机安装了操作系统后，用户可能会感到机器的功能增强了，使用起来也更加简单、方便了。但是，现实生活中并不存在具有这种增强功能的真实硬机器，它只是用户的一种“感觉”罢了。因此，人们称这种由于在裸机上安装了操作系统使功能得以扩展的机器为“扩展机”或“虚拟机（virtual machine）”。此时虚拟机即成为了一个向用户提供的新的工作平台。

下面将这样的“虚拟”概念加以扩充：在裸机上安装操作系统后，机器的功能得到第一次扩展，形成一台虚拟机；在操作系统的基础上，安装第二层软件，机器的功能得到第二次扩展，形成一台功能更强的虚拟机；……依此类推，人们就可以得到一台台虚拟机，后一台虚拟机建立在前一台虚拟机的基础之上，并具有更加强大的功能，如图 1-2 所示。

2. 操作系统是机器的管理者：资源管理的观点

计算机系统是由硬件（CPU、存储器和各种外部设备）和软件（系统程序、应用程序）组合而成的。这些硬件和软件是计算机运行过程中各个用户都需要使用的资源。按照性质的不同，可以把这些资源划分成 4 类：处理机（即 CPU），存储器，外部设备，程序和数据。前 3 种属于硬资源，第 4 种属于软资源。

在计算机的运行过程中，对于每种硬资源都有如下的 4 个问题需要解决。

（1）记住资源当前状态：是否被使用以及谁在使用。

(2) 制定资源分配策略：如何分配、何时分配、分配多少以及应分配给谁。

(3) 实施资源分配：根据分配策略完成分配。

(4) 完成资源回收：使用结束后收回资源，以便进行下次分配。

设想一下，如果把上述4个问题一次性地全部交由用户去解决，其结果可想而知：效率极其低下。

为了使人们从繁杂琐碎的资源管理事务中脱出身来，最好的解决办法还是开发一种软件，它同样紧紧地地位于计算机硬件之上，承揽“记住资源当前状态、制定资源分配策略、实施资源分配、完成资源回收”的任务，把计算机系统的硬软件资源统一管理起来，向用户提供一种简单、有效地使用系统资源的方法，提供一个能够集中精力于自身应用的新开发环境。这种扩展计算机系统功能、方便用户使用的软件集合就是所谓的“操作系统”。

软件是相对于硬件而言的，它是计算机程序、过程、规则和相关文档资料的总称。按照所起的作用和要求的运行环境，可把软件大致划分为应用软件和系统软件两类。应用软件是为解决某类需要或某个特定问题而编制的程序，它涉及计算机应用的各个领域，图1-2中列出的财务管理系统、航空订票系统等是它的代表。系统软件则是另外一类软件，它们不是针对特定需要或特殊问题编制的程序，而是对计算机系统的资源实施管理、控制，为其他程序的运行提供支持和服务的通用软件，一般情况下，系统软件都是由计算机生产厂家提供的，图1-2中列出的操作系统、语言编译程序及汇编程序等是它的代表。

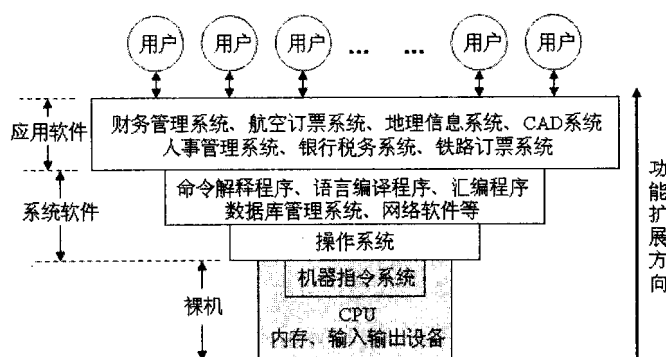


图 1-2 每一层的扩展均向上构成新的虚拟机

综上所述，我们可以这样来给出关于“操作系统”的描述性定义：操作系统是控制和管理计算机硬件和软件资源、合理地组织计算机工作流程以及方便用户使用计算机的一个大型系统软件。

1.1.2 操作系统的功能

配置操作系统，就是要通过对整个系统资源的合理组织、控制和调度，达到提高资源利用率、提高任务运行效率以及方便用户使用的目的。为此，操作系统应具有以下4个方面的基本功能：处理机管理、存储管理、设备管理、文件管理。这4个方面的功能相互配合，协同工作，从而实现对计算机系统资源的管理和控制程序的执行。

1. 处理机管理

中央处理机（也称中央处理器）（CPU）是计算机系统的一种核心硬资源。它具有两个特点，一是用户程序进入内存后，只有当获得CPU的使用权后才能真正得到执行；二是相比系统中的其他硬件设备，CPU的工作速度要快得多。因此，有关它的调度分配策略将会对整个计算机系统性

能的发挥和提高产生重要的影响。

为了提高 CPU 的利用率,尽量不使其空闲,系统应在内存中同时存放几个用户的作业程序(即所谓的“多道程序设计”)。这样,当一个程序因等待某事件(如输入/输出)的完成而暂时不能使用 CPU 时,操作系统即可将 CPU 重新分配给其他可运行的作业程序。例如,用户在进行文本编辑的同时收听优美动听的歌曲就是提高 CPU 利用率的极好示例,即在文本编辑进行过程中两次敲击键盘按键的间隙时间, CPU 被分配转去执行 mp3 歌曲。

处理机管理的主要工作如下。

(1) 记住系统中当前每个用户作业程序的状态,这样,在需要对 CPU 重新进行调度时即可在候选的程序中快速选取。

(2) 制定处理机的调度策略,它是挑选候选程序时应遵循的规则,其中包括调度时机(即将处理机从运行当前程序切换到运行新的程序)、把 CPU 分配给谁使用(因为满足条件的作业可能有多个)、分配多长时间等问题需要回答。

(3) 实施具体的 CPU 分配(即处理机调度)。这里有两个问题需要解决,一是如何做才能使 CPU 从运行一个程序切换到运行另一个程序,以便让获得 CPU 的作业程序真正投入运行;二是如何做才能让暂时被打断使用 CPU 的作业程序,在下次又有机会获得 CPU 时,能够从被打断的断点处正常地往下继续运行。

2. 存储管理

存储器是计算机的记忆装置。在计算机系统中,存储器可分为内存储器(或主存储器)和外存储器(或辅助存储器)两种。内存储器(简称内存)位于主机内部, CPU 可通过指令直接访问它,其特点是存取速度快,存储容量小,价格昂贵。内存一般用来存放计算机当前正在运行的程序和数据,由于断电后其上存储的信息即刻消失,因此它是一种易失性存储器,只能起到临时存储的作用。

外存储器(简称外存)位于主机外部,由于 CPU 不能通过指令直接访问它,因此要使用存储在其中的信息时,必须先将其调入内存,然后信息才能得到相应的利用或处理。相比内存,外存的存取速度较慢,价格低廉,但存储容量较大。由于断电后外存上的信息仍能保存,因此它是一种非易失性存储器,通常可作为内存的延伸和后援,存放暂时不用的或需要长期、永久保存的程序和数据。

由于内存的容量较小,而且信息又只有在进入内存之后才能得到利用和处理,因此内存在整个计算机系统中是一种极为宝贵的资源。多个作业程序都要进入内存,必定会形成对内存资源的竞争。

另外,还需要指出的是,操作系统中所谓的“存储器管理”是针对内存而言的,即管理对象是内存,至于外存的管理,一般则交由文件系统去完成。

存储管理的主要工作如下。

(1) 记住内存各部分的使用情况,包括哪些已经分配,哪些为空闲待分配。此时空闲的存储区域就是进行存储分配时的分配对象。

(2) 制定内存的分配策略。这里需要考虑的问题较多,比如:当有多个用户提出存储请求时,决定接受谁提出的请求;是按要求尺寸全部满足申请者还是部分满足;是分配内存的一个连续存储区还是分配不连续的存储区;……如此等等。

(3) 由于用户作业程序事先并不知道自己会进入内存的哪个区域,因此必须对程序指令中的地址加以调整,这样才能保证程序能够得到正确的运行。这就是存储管理中所谓的“地址重定位”问题。

(4) 实施内存的具体分配和回收。

(5) 建立安全机制,既要确保内存中各独立作业程序的安全、互不侵扰,又要允许不同作业

程序能够共享一些系统或用户的程序。

(6) 随着计算机应用的发展, 涉及的问题越来越复杂, 需要的内存量也就越来越大。受内存容量和价格条件的限制, 存储管理必须借助外存储器来解决“内存”的扩充问题, 使“作业程序比内存大时也能正确运行”。这就是存储管理中的“虚拟存储器”技术。

3. 设备管理

在计算机系统中, 除了处理机和内存外其他全部均为设备管理的对象, 主要是一些输入/输出设备, 如打印机、键盘、鼠标、磁盘驱动器等。外部设备品种繁多, 性能千差万别, 控制各异。因此, 设备管理在整个操作系统软件中所占比例最大, 是最复杂、最庞大的部分。

设备管理的主要工作如下。

- (1) 记住各类设备的使用状态, 并按各自不同的性能特点进行分配和回收。
- (2) 为各类设备提供相应的设备驱动程序、启动程序、初始化程序、控制程序等, 保证输入/输出操作的顺利完成。
- (3) 外部设备的运转速度较慢, 与 CPU 的工作速度极不匹配。因此, 设备管理必须采用缓冲技术, 以提高 CPU 与外部设备、外部设备与外部设备之间操作的并行程度。
- (4) 根据不同的设备特点制定优化策略, 使对具体设备的使用更趋合理和有效。
- (5) 为了方便, 设备管理应使用户在编制程序时不直接使用真实的设备名, 这样有利于设备分配的灵活性, 同时也有利于处理外部设备的故障, 这就是所谓的“设备无关性”, 或称为“设备独立性”。
- (6) 有些设备一次只能允许一个用户使用, 利用率较低。为此, 设备管理应借助大容量的外存, 把这种独占设备改造成为可以多个用户“共享”的设备, 这就是所谓的“虚拟设备”技术。

4. 文件管理

程序与数据都是以文件的形式存放在外存(如硬盘、软盘)上, 是计算机系统的软资源。文件系统应向用户提供各种命令, 以便有效地支持对文件的存储、检索、分类等; 应合理地组织和管理磁盘空间, 以确保文件的安全和共享; 应实施“按名存取”, 使用户只需通过文件的名称即可访问所需要的文件。

文件管理的主要工作如下。

- (1) 要对磁盘空间进行组织和管理, 随时记住磁盘上文件存储空间的使用情况, 包括哪些已经分配, 哪些为空闲待分配。
- (2) 制定文件存储空间的分配策略, 实施具体的分配和回收。
- (3) 维持目录表, 每个目录项中登记有对应文件的名称和有关信息。当用户要使用某个文件时, 只需通过文件名查目录表找到它的目录项, 即可得到该文件的各种信息, 完成所需的读/写操作。
- (4) 确保存放在外存上文件的安全、保密和共享。
- (5) 提供一系列文件使用命令, 以便用户能对文件进行存取、检索、更新等操作。

1.2 操作系统的接口与内核模式

1.2.1 操作系统的两种接口

用户是通过操作系统为其提供的接口来使用计算机的。在用户编写的程序中使用所谓的

“系统调用命令”，获得操作系统提供的各种功能服务，这是操作系统在程序一级给予用户的支持。另外，用户还可以使用操作系统提供的各种操作命令，通过键盘（或鼠标）控制和完成程序的运行，这是操作系统在作业控制一级给予用户的支持。这样的两种支持通常被称为是用户与操作系统之间的接口，即前者被称为“程序接口（program interface）”，后者被称为“命令接口（command interface）”。

1. 特权指令、核心态、用户态

在内存中存放的多个作业程序共享着系统的各种资源。正是由于要实现对资源的“共享”，所以涉及资源管理和使用的硬指令就不能随便使用。例如，如果每个用户作业都有权启动打印机进行输出，那么打印机就会同时打印多个用户的结果，大家的输出内容就交织在一起，必然造成混乱。因此，常把 CPU 指令系统中的指令划分为两类，一类是大家（指操作系统和用户）都能使用的指令，另一类是只能由操作系统使用的指令。前者称为“非特权指令”，后者称为“特权指令”。

例如，传送程序状态字的指令负责从内存单元中取出程序状态字，送到 CPU 的程序状态寄存器中，以便改变程序的执行状态，此时它应是特权指令。

例如，启动、测试和控制设备的指令会直接导致设备的执行，此时它们应是特权指令。

又例如，完成存取特殊寄存器（如指令计数器（PC）、中断寄存器、时钟寄存器等）的指令，也都应该是特权指令。

为了确保只有在操作系统范围内使用特权指令，计算机系统让 CPU 取两种工作状态：核心态（kernel mode）和用户态（user mode）。核心态也称为“管态”，用户态也称为“目态”。当 CPU 处于核心态时，可以执行包括特权指令在内的一切机器指令；当 CPU 处于用户态时，禁止使用特权指令，而只能执行非特权指令。如果在用户态下发现取到了一条特权指令，则中央处理机就会拒绝执行，并产生“非法操作”中断。于是，一方面会转交给操作系统去处理该事件，另一方面会出示“程序中有非法指令”的信息，通知用户进行修改。

CPU 是处在核心态还是处在用户态，硬件会自动设置与识别：当 CPU 的控制权移到执行操作系统程序时，硬件就会把 CPU 工作的方式设置成核心态；当操作系统选择用户程序占用处理机时，CPU 的工作方式就会由核心态转换成为用户态。

2. 程序接口：系统调用命令

当有了操作系统后，资源的管理和使用权都被集中到操作系统手中，用户只是资源的使用者。因此必须提供一种方式，将用户对资源的使用请求告知操作系统，以便由操作系统代之完成。这种方式就是利用操作系统提供的多种系统调用命令来实现的。

操作系统中预先编制了很多不同功能的子程序，而用户在自己的程序中调用这些子程序，以取得操作系统提供的这些功能服务。这些功能子程序被称为“系统功能调用”程序，简称为“系统调用（system call）”。当用户程序中调用这些系统调用提供的功能时，就称为系统调用命令。

不过，用户程序只能在用户态下运行，而系统调用命令的程序代码属于操作系统，它应在核心态下运行。为了解决从用户态到核心态的切换，计算机系统必须提供一种手段，通过它能够使 CPU 从用户态变换成为核心态，进而达到调用有关操作系统功能程序的目的。提供这种手段的是一条名为“访管”的非特权硬指令。

访管指令的功能是当执行它时即产生一个访管软中断（或称“扑获”、“陷阱”），它能够使 CPU 从用户态转为核心态，从而实现进入操作系统处理该中断。

访管指令需通过语言编译程序来实现其具体应用,即在编译时,编译程序把用户编写的源程序中的每个系统调用都做这样的翻译:一是转换成一条访管指令,二是把具体调用的功能转换成相应的编号。这样,编译完成后 CPU 运行程序时,就会由执行访管指令而从用户态进入核心态,然后再根据功能编号转到相应的系统调用功能处理程序执行。

例 1-1 分析 UNIX 中系统调用命令的实现过程。

例如,在 C 语言中, `write (fd,buf,count)` 是有关“文件写”的系统调用命令。通过它,用户可以实现向一个文件中进行写的操作:把 `buf` 指向的内存缓冲区中的 `count` 个字节内容写到文件号为 `fd` 的磁盘文件上。因此, `write` 表示一个系统调用命令,且是要求调用文件写功能的系统调用命令,命令括号中的 `fd`、`buf` 和 `count` 是由用户提供的、表示要求系统按何种条件去完成文件写操作的 3 个参数。

C 编译程序在编译 C 源程序中的这条系统调用命令时,总是把系统调用命令 `write` 翻译成能够引起软中断的访管指令 `trap`。该指令长两个字节,第 1 个字节为操作码,第 2 个字节为该系统调用命令的功能编号。例如, `trap` 的 16 进制操作码为 89, `write` 的功能号为 04。那么通过编译, `write` 将被翻译成一条二进制值为“1000100100000100”的机器指令(其八进制是 104404)。`write` 命令括号中的参数将由编译程序把它们顺序存放在 `trap` 指令的后面。于是,源程序中的语句 `write (fd,buf,count)` 经过编译后,即对应于图 1-3 (a) 所示的 `trap` 机器指令。

`trap` 指令中的功能编号是用来区分不同的系统功能调用的。在 UNIX 操作系统中有一张“系统调用程序入口地址表”。该表表目从 0 开始且以系统调用命令对应的功能号为索引顺序进行排列。例如, `write` 的功能号是 04,那么该表中的第 5 个表目内容就是对应于 `write` 的。系统调用程序入口地址表的每个表目形式如图 1-3 (b) 所示。它由两部分组成,一是给出该系统调用所需要的参数个数,一是给出该系统调用功能处理程序的入口地址。

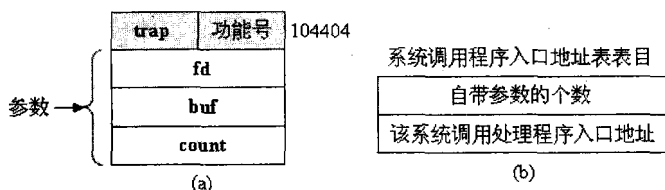


图 1-3 `trap` 指令格式与“系统调用程序入口地址表”表目内容

现在可以用图 1-4 来描述该系统调用的整个处理过程了。C 语言编译程序把系统调用命令 `write (fd,buf,count)` 翻译成 `trap` 指令 104404,简记为 `trap 04`。当处理机执行到 `trap 04` 这条指令时,即会产生中断,硬件把处理机的工作方式由用户态转变为核心态。于是 CPU 就可以去执行操作系统中的 `trap` 中断处理程序。该程序根据 `trap` 中的功能号 04,从系统调用处理程序入口地址表中的第 5 个表目中,既可得到该系统调用应有 3 个参数(它跟随着在目标程序 `trap 04` 指令的后面),又可从表目中得到该系统调用处理程序的入口地址。于是,就携带这 3 个参数去具体执行 `write` 的处理程序,从而完成用户提出的输入/输出操作要求。执行完毕,又把处理机恢复到用户态,返回目标程序中 `trap` 指令的下一条指令(即断点)继续执行。这里需要注意的是,由于系统调用处理程序入口表第 1 个表目的功能号是 0,因此图 1-4 中第 5 个表目的功能号为 4。

不同操作系统所提供的系统调用命令在数量、使用格式以及功能上都会有所不同,但对系统调用的处理过程则大致相同。

从功能上看,可以把系统调用命令分成 5 大类:一是关于处理机管理和控制的;二是关于外部设

备输入/输出的；三是关于磁盘文件管理的；四是关于访问系统信息的；五是关于存储申请与释放的。

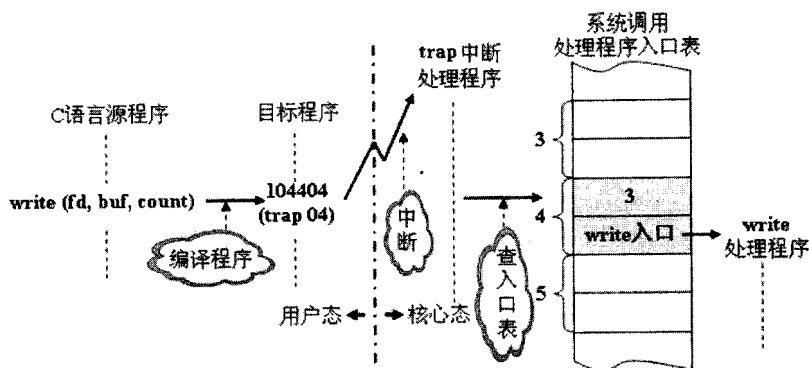


图 1-4 系统调用处理过程示意图

从形式上看，虽然操作系统提供的系统调用与一般程序设计中的过程调用（或称子程序调用）相似，但它们也有着如下明显的区别。

（1）一般的过程调用，调用者与调用者都运行在相同的 CPU 状态，即或都处于用户态（用户程序调用用户程序），或都处于核心态（操作系统程序调用操作系统程序），但当系统调用时，发出调用命令的调用者运行在用户态，而被调用的对象则运行在核心态。

（2）一般的过程调用，是直接通过转移指令转到被调用的程序，但当系统调用时，则只能通过软中断指令（如 trap）提供的一个统一的入口由用户态进入核心态，然后再经分析后才转向相应的命令处理程序。

（3）一般的过程调用，在被调用者执行完成后，即直接返回断点继续执行，但系统调用可能会导致用户程序（即“进程”）运行状态的变化，从而引起系统重新分配处理机。因此，当系统调用处理结束后，不一定是返回调用者断点处继续执行。

3. 命令接口：命令行和图形用户界面

用户通过操作系统提供的命令行和图形用户界面两种命令接口组织和控制作业程序的执行，管理所使用的计算机系统。命令行是传统的命令接口，而图形用户界面则是目前流行的命令接口，它通过鼠标点击屏幕上的对象来完成各种控制和管理任务。

图形用户界面是读者熟悉的操作界面，在此不再赘述。

命令行接口有两种使用方式：脱机命令行接口和联机命令行接口。

- 所谓“脱机命令行接口”是指系统向用户提供作业控制语言（Job Control Language, JCL）。首先用户使用 JCL 编写作业说明书规定完成一个作业过程中所需的一系列作业步，描述每个步骤的特性、使用的资源及文件，然后将该作业说明书连同程序和数据一起提交给系统，系统按照作业说明书上的信息控制作业的执行。

- 所谓“联机命令接口”是指操作系统提供一组操作命令、终端处理程序以及命令解释程序。首先用户通过键盘输入所需要的命令，终端处理程序接收并在显示器上回显命令，当命令输入完毕后，由命令解释程序对它进行分析，然后执行相应的命令处理程序，完成用户的一次请求。如此反复，直到任务完成为止。这是一种人-机交互控制程序执行和管理计算机系统的方法。各种操作系统所提供的联机命令，在数量、功能、格式上都不尽相同。

例 1-2 Linux 的联机命令接口。

Linux 提供了丰富的命令集，用户利用这些命令可以有效地进行诸如文件操作及目录管理等

工作。Linux 命令的一般格式是：

命令名 [选项] [参数 1] [参数 2]……

其中，命令名由小写英文字母组成，方括号表示这部分内容可选。使用时，可直接在系统命令提示符（对一般用户，默认的提示符是“\$”）的后面输入命令，也可在命令名后跟随选项和参数。选项是对命令的进一步定义，以“-”开始；参数通常是一些文件名，告诉命令从哪里可以得到输入数据以及把输出存放在什么地方。

例如，ls 命令用来列出指定目录下的文件和目录的清单，常用的选项有：

- -a：列出指定目录下所有文件与子目录，包括隐藏文件。
- -l：以长格式显示文件的详细信息，如所有者、建立日期、大小和权限等。

假定用户 zong 的当前目录为/home/zong，目录下有文件 test.txt、子目录 dl 及 backup，那么在提示符\$下发命令：

```
$ls -a
```

屏幕上就会列出当前目录下的所有文件（包括隐藏文件）如下：

```
.   test.txt .bash_history .bash_profile    dl      .gtkrc
..  backup  .bash_logout .bashrc         .emacs
```

1.2.2 操作系统内核的结构模式

目前，在设计一个操作系统时，其内核（即核心部分）的结构主要有两种模式：单内核模式和微内核模式。

1. 单内核模式

单内核模式也称为集中模式或整体模式，整个系统是一个大的模块。这时，操作系统提供的工作流程是应用主程序用给出的参数值去执行操作系统中的各种系统调用命令。于是，一方面由于系统调用命令位于操作系统的内核，因此 CPU 的工作状态将由执行用户程序的用户态切换为执行系统程序的核心态；另一方面，操作系统根据具体的参数值调用特定的系统调用服务程序，而这些服务程序又会根据需要再调用位于内核底层的各种支持函数，从而完成所需的功能。在完成了应用程序要求的服务后，操作系统又改变 CPU 的状态，使其从核心态切换回到用户态，从而返回到应用程序中继续执行后面的指令。

单内核模式结构的系统虽然是一个大的模块，但可以粗略地划分成 3 个层次：调用系统服务的主程序层、执行系统调用的服务层以及支持系统调用的底层函数。后两层即是操作系统的内核。当从第 1 层进入第 2 层时，CPU 由用户态切换成为核心态；当从内核返回应用主程序时，则 CPU 又由核心态切换成为用户态，如图 1-5 所示。

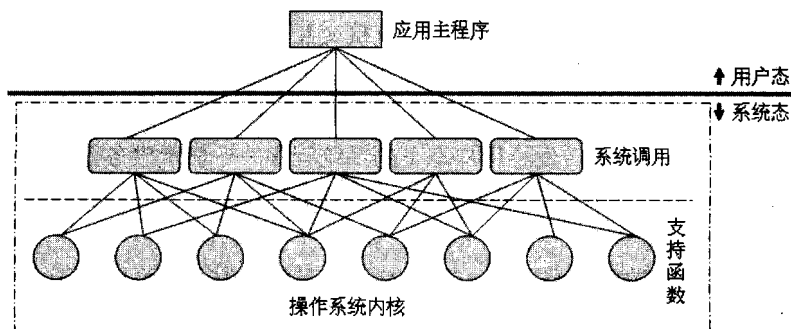


图 1-5 单内核模式的结构示意图