


“基于系统能力培养的计算机专业课程建设研究”项目规划教材




操作系统 设计与实现

Operating System
Design and
Implementation

陈文智 主编

陈文智 施青松 龙鹏 编著

高等教育出版社



操作系统 设计与实现

Operating System
Design and
Implementation

陈文智 主编

陈文智 施青松 龙鹏 编著

内容提要

本书为“基于系统能力培养的计算机专业课程建设研究”项目规划教材。本书定位于初学者的动手实践类操作系统教材，可带领读者一步步从无到有，实现一个近 1 万行的小型内核 Lcore。

全书共 10 章。第 1 章简单回顾操作系统的基本原理；第 2 章从操作系统五大模块入手设计 Lcore 系统；第 3 章为 Lcore 添加 VGA 显示功能；第 4 章从时钟中断入手讨论如何在内核中实现中断处理；第 5 章实现键盘输入功能；第 6 章分别实现内存管理系统的 Bootmem、Buddy、Slub 分配器；第 7 章介绍进程管理初始化、进程创建、页异常处理和进程调度；第 8 章实现一个能够在 Lcore 内核中运行的类 FAT32 文件系统，同时也涉及如何编写一个 SD 卡的小驱动；第 9 章实现系统调用模块；第 10 章开发一款专属、简单的 shell。附录是硬件实验平台手册，说明平台的有关使用细节，以便读者在开发过程中随时查阅。

本书可作为本科计算机类专业操作系统课程配套的实践教材，也可供有关技术人员参考。

图书在版编目(CIP)数据

操作系统设计与实现 / 陈文智主编；陈文智，施青松，龙鹏编著. --北京：高等教育出版社，2017.2

ISBN 978-7-04-046752-9

I. ①操… II. ①陈… ②施… ③龙… III. ①操作系统-程序设计-高等学校-教材 IV. ①TP316

中国版本图书馆 CIP 数据核字(2016)第 275715 号

策划编辑 倪文慧
插图绘制 杜晓丹

责任编辑 倪文慧
责任校对 刘春萍

封面设计 张申申
责任印制 韩刚

版式设计 张杰

出版发行 高等教育出版社

社 址 北京市西城区德外大街 4 号

邮政编码 100120

印 刷 北京汇林印务有限公司

开 本 787mm×1092mm 1/16

印 张 19.25

字 数 360 千字

购书热线 010-58581118

咨询电话 400-810-0598

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

网上订购 <http://www.hepmall.com.cn>

<http://www.hepmall.com>

<http://www.hepmall.cn>

版 次 2017 年 2 月第 1 版

印 次 2017 年 2 月第 1 次印刷

定 价 36.00 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换

版权所有 侵权必究

物料号 46752-00

数字课程资源使用说明

与本书配套的数字课程资源发布在高等教育出版社易课程网站,请登录网站后开始课程学习。

一、注册/登录

访问 <http://abook.hep.com.cn/18780002>,点击“注册”,在注册页面输入用户名、密码及常用的邮箱进行注册。已注册的用户直接输入用户名和密码,即可登录进入“我的课程”页面。

二、课程绑定

点击“我的课程”页面右上方“绑定课程”,正确输入教材封底防伪标签上的 20 位密码,点击“确定”完成课程绑定。

三、访问课程

在“正在学习”列表中选择已绑定的课程,点击“进入课程”即可浏览或下载与本书配套的课程资源。刚绑定的课程请在“申请学习”列表中选择相应课程并点击“进入课程”。

四、与本书配套的易课程数字课程资源包括电子教案与源代码、交叉编译工具、模拟器等,方便读者学习使用。

账号自登录之日起一年内有效,过期作废。

如有账号问题,请发邮件至: abook@hep.com.cn。

序

从技术的角度看,现代计算机工程呈现出系统整体规模日趋庞大、子系统数量日趋增长且交联关系日趋复杂等特征。这就要求计算机工程技术人才必须从系统的高度多维度地研究与构思,综合运用多种知识进行工程实施,并在此过程中反复迭代以寻求理想的系统平衡性。上述高素质计算机专业人才的培养,是当前我国高校计算机类专业教育的重要目标。

经过半个多世纪的建设,我国计算机专业课程体系完善、课程内容成熟,但在高素质计算机专业人才的培养方面还存在一些普遍性问题。

(1) 突出了课程个体的完整性,却缺乏课程之间的融通性。每门课程教材都是一个独立的知识体,强调完整性,相关知识几乎面面俱到,忽略了前序课程已经讲授的知识以及与课程之间知识的相关性。前后课程知识不能有效地整合与衔接,学生难以系统地理解课程知识体系。

(2) 突出了原理性知识学习,却缺乏工程性实现方法。课程教学往往突出原理性知识的传授,注重是什么、有什么,却缺乏一套有效的工程性构建方法,学生难以实现具有一定规模的实验。

(3) 突出了分析式教学,却缺乏综合式教学。分析式教学方法有利于学习以往经验,却难于培养学生的创新能力,国内高校计算机专业大多是分析式教学。从系统论观点看,分析式方法是对给定系统结构,分析输入输出关系;综合式方法是对给定的输入输出关系,综合出满足关系的系统结构。对于分析式教学方法来说,虽然学生理解了系统原理,但是仍然难于重新构造系统结构。只有通过综合式教学方法,才能使得学生具有重新构造系统结构的能力。

在此背景下,教育部高等学校计算类专业指导委员会提出了系统能力培养的课题研究。这里所说的“系统能力”,是指能理解计算机系统的整体性、关联性、动态性和开放性,掌握硬软件协同工作及相互作用机制,并综合运用多种知识与技术完成系统开发的能力。以系统能力培养为目标的教学改革,是指将本科生自主设计“一台功能计算机、一个核心操作系统、一个编译系统”确立为教学目标,并据此重构计算机类课程群,即注重离散数学的基础,突出“数字逻辑”“计算机组成”“操作系统”“编译原理”4门课程群的融合,形成边界清晰且有序衔接的课群知识体系。在教学实验上,强调按工业标准、工程规模、工程方法以及工具环境设计与开发系统,提高学生设计开发复杂工程问

题解决方案的系统能力。

在课题研究的基础上,计算机类教学指导委员会研制了《高校计算机类专业系统能力培养研究报告》(以下简称《研究报告》)。其总体思路是:通过对系统能力培养的课程体系教学工作凝练总结,明确系统能力培养目标,展现各学校已有的实践和探索经验;更重要的是总结出一般性方法,推动更多高校开展计算机类专业课程改革。国内部分高校通过长期的系统能力培养教学改革探索与实践,不仅提高了学生的系统能力,同时还总结出由顶层教学目标驱动“课程群为中心”的课程体系建设模式,为计算机专业改革提供了有益参考。这些探索与实践成果,也为计算机类专业工程教育认证中的复杂工程问题凝练,以及解决复杂工程问题能力提供了很好的示范。

高水平的教材是一流专业教育质量的重要保证。在总结系统能力培养教学改革探索与实践经验的基础上,国内部分高校也组织了计算机专业教材编写。高等教育出版社为《研究报告》的研制以及出版这批具有创新实践性的系列教材提供了支持。这些教材以强化基础、突出实践、注重创新为原则,体现了计算机专业课程体系的整体性与融通性特点,突出了教学分析方法与综合方法的结合,以及系统能力培养教学改革的新成果。相信这些教材的出版,能够对我国高校计算机专业课程改革与建设起到积极的推动作用,对计算机专业工程教育认证实践起到很好的支撑作用。

教育部高等学校计算机类专业教学指导委员会秘书长

马殿富

2016年7月

引 言

互联网是一个充满创新的领域,技术革新非常快,云计算逐渐成为核心技术之一。伴随着 Xen、QEMU-KVM、Docker 等相继出现,它们的周边技术也在蓬勃发展,业内呈现一片欣欣向荣、百家争鸣的景象。作为计算机技术的基石——操作系统也随着云计算的火热发展而在不断进步,国内外越来越多的技术大牛投身到 Linux 内核的开发行列中。

同时,随着 IT 技术的飞速发展,信息安全领域面临的挑战也日益加剧,不仅关系到每个公民的切身利益,而且也直接影响到国家安全。各国政府在此方面从未掉以轻心,持续不断地进行技术投入,力求为公民、国家创造一个安全的网络环境。操作系统在计算机安全领域中的地位毫无疑问是根深蒂固的,国内外殿堂级的黑客们都有一个共同的特点:熟悉操作系统技术。

年轻而富有朝气的你还在等什么,赶紧加入内核 hacker 的队伍中吧!无论你将来从事云计算还是信息安全,抑或是 IT 其他领域,掌握一定的操作系统内核知识对你未来的发展绝对是有百利而无一害。如果你此前没有接触过内核,那么从这里开始,本书将引领你实现一款自己专属的小内核,或许不能让你的技能在短时间内突飞猛进,至少能够使你加深对操作系统的理解。

写作背景

“互联网+”和“大数据”的应用普及,使计算机人才培养面临许多机遇和挑战。新兴硬件的出现(如 FPGA 等),引发了计算机教学方法和平台的相应变化;新兴研究领域的出现(如类脑计算、智能硬件等),开启了新一轮计算机教学内容的整合和创新;计算模式的不断演变(如集中计算时代、并行计算时代、存处耦合时代),引领了计算机教育软硬件协同和垂直创新思维。

目前大部分计算机专业课程体系存在一些普遍性问题:各课程大多独立设计和实施,知识体系规划缺乏足够的工程训练规模与难度。现有的计算机教学内容和教学方法与实际需求严重脱节,高校计算机教学与业界需求水平差距越来越大。如何提高学生对计算机的兴趣、解决学生欺“软”怕“硬”的畏难情绪?如何树立学生的系统观、提高学生的系统能力?如何解决各门课程内容孤立封闭、关联脱节的问题?如何让不同基础和层次的学生都接受相应的系统能力训练?如何加强课内外实践结合、发挥学生

自主创造性? 这一系列问题已经成为国内外计算机教育界探讨的重要问题,也是必须解决的问题。

令人欣喜的是,近年来“系统能力培养”逐渐成为计算机人才培养的共识。作者认为,系统能力是指以计算思维为基本思想来理解计算机系统的整体性、关联性、层次性、动态性和开放性,掌握计算机软硬件协同工作及相互作用机制,并综合运用多种知识与技术完成全系统开发的能力。因此,提出面向系统设计能力培养的的总体思路,以深入理解计算机系统和高性能程序设计为目标,以计算机系统设计与实践的实践过程为手段,融合贯通数字逻辑、计算机组成、操作系统和编译技术等硬软件系列课程知识体系,以系统能力为统一视图,建立层次化、循序递进、开放式课程群的课程体系和实践体系,让学生在在校期间递进式逐步设计实现简易但完整的计算机系统,并用自己的计算机系统管理及运行自己的应用程序。

在教育部计算机类专业教学指导委员会的支持下,我们结合自身的教学实践,陆续编写了本系列系统能力培养丛书,内容涉及数字电路(逻辑与计算机设计)、计算机组成、计算机体系结构、计算机硬件综合、操作系统、编译原理、计算机系统综合等。

本书主要关注小型操作系统的开发和设计。

本书意义

在撰写本书之前,国内外已出版了多本操作系统内核的相关书籍,这些书籍可以大致分为以下三类。

① 内核原理类:从理论层面上为读者介绍操作系统设计与实现中所涉及的技术原理,代表作有 William Stallings 的《操作系统:精髓与设计原理》、Andrew S. Tanenbaum 的《现代操作系统》、Abraham Silberschatz 的《操作系统概念》。

② 内核剖析类:从代码实现角度为读者分析操作系统主要模块的设计与实现,代表作有 Marshall K. McKusick 的《FreeBSD 操作系统设计与实现》、Robert Love 的《Linux 内核设计与实现》、Daniel P. Bovet 的《深入理解 Linux 内核》。

③ 动手实践类:从零开始带领读者实现一个小型内核,代表作有于渊的《Orange's: 一个操作系统的实现》、川合秀实的《30 天自制操作系统》。

这三类图书简单分析如下。

首先,内核原理类书籍是学习操作系统技术时必须阅读的,其知识是无数操作系统先驱穷极一生而留下的瑰宝,是操作系统设计的指导方针。无论是 Linux、UNIX,还是 Windows,其根本原理都离不开这类书籍中叙述的各种设计思想。但是,学习技术仅精于理论是远远不够的,在掌握原理之后必须结合源码进行实践。

其次,类似 UNIX、Linux 这样的内核有着 20 年以上的历史,是古往今来所有相关领

域从业者的智慧结晶,它们分支繁多、内容庞大,代码量都是百万、千万行级别的,因此内核剖析类书籍只能从主要模块入手进行“庖丁解牛”。这类书对于初学者来说,有两个不足:

① 难度太大,刚掌握了基本理论便直接面对百万行的代码,无疑是让初学者崩溃的。更“可怕”的是这些内核中使用了太多精心的结构逻辑设计、高超的编程技巧,进一步加大了初学者学习的难度。

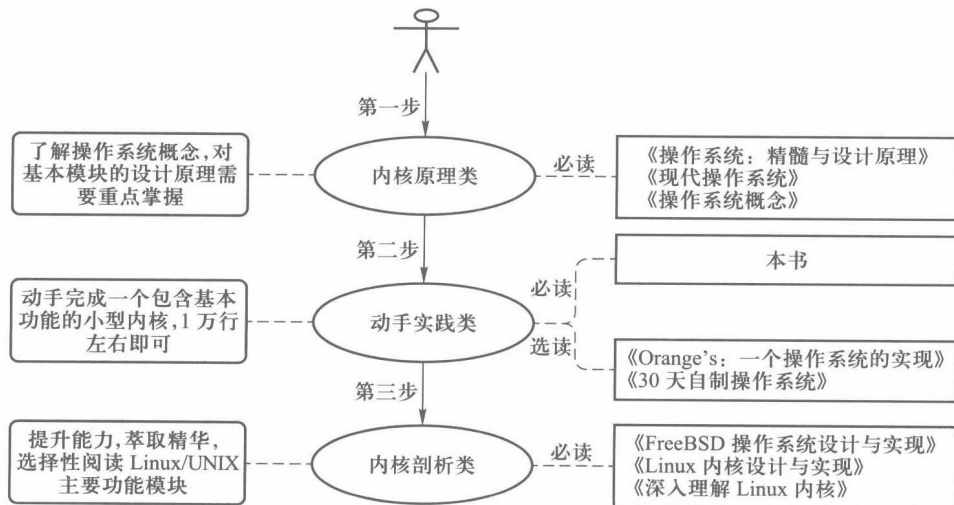
② 缺乏整体性,初学者即使了解了某个模块,也难以从整体层面对系统有深入的了解,宛如“盲人摸象”。

本书定位于上述的动手实践类书籍,同样是一步一步带领读者从无到有实现一个1万行不到的小型内核,对于初学者来说既具有挑战性,又具有合理的工作量。根据多年教学经验,我们认为在同类书籍基础上,还需具备以下要素方可利于初学者学习操作系统知识:

① 降低学习体系结构的成本。同类动手实践型图书大多基于 X86 架构,该架构对于初学者来说过于复杂,而操作系统基本原理是可以独立于体系架构的,因此本书立足于颇为简单的硬件平台,使读者能够专注于操作系统原理的实践。

② 增强系统实践的整体性。同类动手实践类图书侧重于操作系统的某个方面(如 GUI、嵌入式),而本书着重于系统的完整性,通过对进程管理、内存管理、文件管理、设备管理、用户接口这5个基本模块的设计与实现,使读者能够清楚地理解操作系统是如何运作、如何管理计算机的。

最后,我们为读者推荐一条合理的学习路线:



内容安排

本书将带领读者实现一个小型内核 Lcore, 内容包括 11 章和 1 个附录。章节先后顺序是根据 Lcore 系统开发顺序确定的, 方便读者梳理思路、理清逻辑。

第 1 章简单回顾了操作系统的基本原理(读者可以根据自身情况选择性地跳过此部分), 并介绍如何搭建实验所需的开发环境。

系统设计是软件开发中最为关键的一环, 也是在编码前必须要做的一项工作。第 2 章将从操作系统五大模块入手设计 Lcore 系统。

在系统开发过程中必然会遇到许多问题, 此时需要进行调试。由于是从零构建一个系统, 无法使用 GDB 等调试工具, 所以打印显示是一个靠谱的方法。第 3 章中将为 Lcore 添加 VGA 显示功能, 通过显示输出的信息来大致判断程序逻辑是否正确。

中断是现代操作系统中很重要的概念, 很多行为都是由中断来触发的, 最好较早建立中断处理框架。当中断产生时, 处理器暂停当前正在执行的程序转而调用相应的中断处理程序来响应中断, 通过这种方式能够加强处理器对多任务事件的处理能力。第 4 章将带领读者从时钟中断入手来学习如何在内核中实现对中断进行处理。

键盘是系统最基本、最简单的输入设备, 它依赖于中断机制。由于在第 4 章中添加了中断处理逻辑, 所以此时可以实现键盘输入功能了, 第 5 章将带领读者完成此项任务。

内存是计算机系统中极其重要的资源, 它的速度介于 CPU 缓存和外部存储之间, 是 CPU 能直接寻址的存储空间, 其作用是暂时存放运算数据、与磁盘等外部存储设备进行数据交换。进程管理、文件系统等模块都需要与内存模块打交道, 因此第 6 章实现内存管理功能, 将分别实现 Bootmem、Buddy、Slub 三种分配器。

有了内存管理模块之后, 接着来实现进程管理功能, 第 7 章将从进程管理初始化、进程创建、页异常处理、进程调度这 4 个方面展开。这一章学习完成之后, Lcore 小内核将会具备多任务功能。

第 8 章将带领读者实现一个能够在 Lcore 内核中运行的类 FAT32 文件系统, 同时也涉及如何编写一个 SD 卡的小驱动。完成本章的学习后读者的小内核就可以从 SD 卡上读写文件了。

通过前面 8 章内容的学习, 我们已经拥有了一个支持键盘输入、VGA 显示、中断处理、异常处理、内存管理、多进程支持的小型内核, 但是还缺乏一块重要的内容, 也就是第 9 章中将要实现的模块——系统调用。实现了系统调用, 用户将可以和内核进行交互, 进而拥有使用相关系统资源的功能。

第 10 章将通过第 9 章实现的系统调用功能来开发一款专属的、简单的 shell, 希望

能调动初学者的积极性,因为通过它能够执行一些命令,并且可以利用提供的用户态库来实现其他用户程序。

第 11 章总结全文,此前付出的一系列汗水在本章得到了回报!

附录是硬件实验平台手册,说明了平台的使用细节,可供读者在开发过程中随时查阅。

本书主要由陈文智、施青松、龙鹏等共同编写整理。清华大学向勇老师和北京航空航天大学王雷老师认真审阅了全书,并提出了中肯的修改意见,在此向他们表示衷心感谢。同时感谢浙江大学计算机系统能力教学改革小组的所有教师、学生的帮助,感谢高等教育出版社的大力支持!

作者

2016 年 12 月于浙江大学玉泉

目 录

第 1 章 操作系统开发介绍	1	2.2.4 其他初始化	44
1.1 操作系统原理	1	小结	45
1.1.1 进程管理	1	第 3 章 VGA 显示功能	46
1.1.2 内存管理	2	3.1 VGA 初始化	46
1.1.3 文件管理	3	3.1.1 实验平台 VGA 介绍	46
1.1.4 I/O 设备管理	3	3.1.2 实现 VGA 初始化	48
1.1.5 用户接口	4	3.2 实现 put_char 函数	53
1.2 实验平台简介	5	3.2.1 scroll_screen 函数	54
1.2.1 物理实验平台	5	3.2.2 put_char_ex 函数	57
1.2.2 计算机硬件体系结构	7	3.2.3 put_char 函数	57
1.2.3 QS 模拟器	9	3.3 实现 printk 函数	60
1.2.4 平台指令集	9	3.3.1 可变参数函数	61
1.3 开发环境搭建	10	3.3.2 相关子函数的实现	64
1.3.1 交叉编译工具链	10	3.3.3 printk 函数的实现	67
1.3.2 QS 模拟器使用	15	小结	70
1.3.3 其他工具	25	第 4 章 处理时钟中断	71
小结	26	4.1 硬件的外部中断	71
第 2 章 系统设计与初始化	27	4.1.1 机制剖析	71
2.1 Lcore 整体设计	27	4.1.2 相关硬件接口	72
2.1.1 进程管理	27	4.2 中断处理注册机制	74
2.1.2 内存管理	29	4.2.1 相关数据结构	74
2.1.3 文件系统	31	4.2.2 中断处理入口	80
2.1.4 I/O 设备	32	4.2.3 中断处理注册	86
2.1.5 用户接口	32	4.3 时钟中断实现	88
2.2 建立初始化框架	33	4.3.1 时钟中断注册	88
2.2.1 init_kernel 函数	33	4.3.2 显示信息规划	89
2.2.2 中断异常入口	35	4.3.3 系统时间显示	92
2.2.3 内核页表初始化	38	小结	97

第 5 章 键盘输入功能	98	7.4.1 调度初始化	180
5.1 键盘中断处理	98	7.4.2 执行调度	181
5.1.1 相关硬件接口	98	小结	185
5.1.2 实现键盘中断处理	99	第 8 章 实现文件系统	186
5.2 键盘扫描码解析	102	8.1 SD 卡驱动程序实现	186
5.2.1 键盘扫描码	102	8.1.1 SD 卡通信方式简介	186
5.2.2 解析扫描码	103	8.1.2 实现 SPI 通信	187
小结	108	8.1.3 指令处理	187
第 6 章 物理内存管理	109	8.1.4 实现 SD 卡初始化	189
6.1 内存管理策略	109	8.1.5 实现 SD 卡读操作	192
6.2 Bootmem 管理实现	111	8.1.6 实现 SD 卡写操作	196
6.2.1 Bootmem 数据结构	111	8.2 文件系统实现	202
6.2.2 Bootmem 初始化	112	8.2.1 文件控制块的实现	202
6.3 Buddy 系统实现	126	8.2.2 缓存机制的实现	203
6.3.1 Buddy 系统数据结构	126	8.2.3 FAT 表	210
6.3.2 Buddy 系统初始化	127	8.2.4 文件系统初始化	212
6.3.3 Buddy 系统内存释放	129	8.2.5 打开文件	216
6.3.4 Buddy 系统内存分配	131	8.2.6 关闭文件	222
6.4 Slub 系统实现	134	8.2.7 文件读取	224
6.4.1 Slub 系统数据结构	134	8.2.8 文件写入	227
6.4.2 Slub 系统初始化	135	8.2.9 创建文件	234
6.4.3 Slub 系统分配内存	139	小结	238
6.4.4 Slub 系统释放内存	144	第 9 章 添加系统调用	239
小结	146	9.1 系统调用框架	239
第 7 章 实现进程管理	147	9.1.1 系统调用入口	239
7.1 进程管理初始化	147	9.1.2 系统调用初始化	242
7.1.1 进程数据结构	147	9.1.3 系统调用定义	243
7.1.2 进程初始化	151	9.2 添加系统调用	244
7.2 进程创建	168	9.2.1 进程管理相关	244
7.3 页异常处理	174	9.2.2 输入显示相关	246
7.3.1 缺页异常处理	174	9.2.3 文件系统相关	248
7.3.2 写保护异常	176	小结	250
7.4 进程调度	180		

第 10 章 实现简单 shell	251	A. 1.3 CP0 寄存器组	270
10.1 实现用户态库	251	A. 1.4 中断处理	273
10.2 实现 shell 程序	254	A. 1.5 MMU 功能	274
10.2.1 解析程序	255	A. 1.6 外设接口	275
10.2.2 内建命令	258	A. 2 实验平台指令集	284
10.3 其他基本程序	261	A. 2.1 算术运算指令	284
10.3.1 ls 命令	261	A. 2.2 逻辑运算指令	285
10.3.2 ps 命令	262	A. 2.3 移位指令	285
10.3.3 kill 命令	263	A. 2.4 读写指令	285
小结	264	A. 2.5 跳转指令	286
结束语	265	A. 2.6 自陷指令	287
附录A 实验平台手册	267	A. 2.7 CP 指令	287
A. 1 实验平台详细介绍	267	A. 3 实验平台寄存器	287
A. 1.1 Sword 实验平台	267	A. 3.1 通用寄存器	287
A. 1.2 物理地址分配	270	A. 3.2 CP0 寄存器	288

第 1 章 操作系统开发介绍

古人云“不以规矩,不能成方圆”,这告诫我们:要想做好某些事情必定需要遵循与其相关的规则。对于开发操作系统而言,其规矩就是操作系统基本原理,因此在本章中将首先回顾一下操作系统的基本原理(当然,这里的介绍只能是很笼统的,因为有很多操作系统原理的经典书籍可供参考,读者可以根据自身情况选择性地跳过第 1.1 节),接着介绍我们的实验平台以及操作系统开发环境的搭建方法。

1.1 操作系统原理

操作系统的主要功能是管理计算机的硬件资源,为多任务处理提供良好的运行环境,以保证多任务能有条不紊、高效地运行,并能最大程度地提高系统中各种资源的利用率和方便用户的使用。为实现上述任务,操作系统应具有这样几方面的功能:进程管理、内存管理、设备管理和文件管理。为了方便用户使用操作系统,还须向用户提供方便的用户接口。此外,20 世纪 90 年代网络开始普及,越来越多的计算机接入网络中,为了方便计算机联网,又在操作系统中增加了面向网络的服务功能。

1.1.1 进程管理

在传统的多任务系统中,处理器的分配和运行都是以进程为基本单位,因而对处理器的管理可归结为对进程的管理;在引入了线程的操作系统中,也包含对线程的管理。处理器管理的主要功能是创建和撤销进程(线程),对诸进程(线程)的运行进行协调,实现进程(线程)之间的信息交换,以及按照一定的算法把处理器分配给进程(线程)。

一个程序的指令如果不被 CPU 执行,那么这个程序什么都不会做。目前在计算机领域一般将正在处理器上调度执行(运行态或者其他阻塞状态等)的程序称为进程。例如,分时用户程序如编译程序是进程;由个人计算机上的单个用户所运行的 Word、Excel、网页浏览器等是进程;系统任务,如将输出发送到打印机也是进程。

进程完成其任务需要一定的资源,包括 CPU 时间、内存、文件和 I/O 设备等。这些资源可以在创建进程时赋予进程或在执行进程时分配给进程。除了在创建时进程所得

到的各种物理和逻辑资源外,也会接受传输过来的各种初始化数据。例如,考虑这样一个进程,其功能是在终端屏幕上显示文件的状态。该进程会得到一个文件名作为输入,会执行适当的指令和系统调用以得到并在终端上显示所要的信息。当进程终止时,操作系统会收回所有可再用的资源。

进程是以异步方式运行的,并以人们不可预知的速度向前推进。为使多个进程能有条不紊地运行,系统中必须设置进程同步机制。进程同步的主要任务是为多个进程(含线程)的运行进行协调。有两种协调方式:

① 进程互斥方式。这是指诸进程(线程)在对临界资源进行访问时,应采用互斥方式。

② 进程同步方式。这是指在相互合作去完成共同任务的诸进程(线程)间,由同步机制对它们的执行次序加以协调。

为了实现进程同步,系统中必须设置进程同步机制,最常用的进程同步机制是信号量机制。最简单的用于实现进程互斥的机制是为每一个临界资源配置一把锁,当锁打开时,进程(线程)可以对该临界资源进行访问;而当锁关上时,则禁止进程(线程)访问该临界资源。

1.1.2 内存管理

内存管理的主要任务是为多任务的运行提供良好的环境,方便用户使用内存,提高内存的利用率以及能从逻辑上扩充内存。为此,内存管理应具有内存分配、内存保护、地址映射和内存扩充等功能。

内存分配的主要任务是为每个程序分配内存空间,使它们“各得其所”;提高内存的利用率,以减少不可用的内存空间;允许正在运行的程序申请附加的内存空间,以适应程序和数据动态增长的需要。操作系统在实现内存分配时,可采取静态和动态两种方式。在静态分配方式中,每个进程的内存空间是在进程启动时确定的,在进程启动后的整个运行期间,不允许该进程再申请新的内存空间,也不允许进程在内存中“移动”。在动态分配方式中,每个进程所要求的基本内存空间也是在装入时确定的,但允许进程在运行过程中继续申请新的额外内存空间,以适应程序和数据的动态增长,也允许进程在内存中“移动”。

内存保护的主要任务是确保每个用户程序都只在自己的内存空间内运行,彼此互不干扰;绝不允许用户程序访问操作系统的程序和数据;也不允许用户程序转移到非共享的其他用户程序中去执行。为了确保每个程序都只在自己的内存区中运行,必须设置内存保护机制。一种比较简单的内存保护机制是设置两个界限寄存器,分别用于存放正在执行程序的上界和下界。系统须对每条指令所要访问的地址进行检查,如果发

生越界即发出越界中断请求,以停止该程序的执行。如果这种检查完全用软件实现,则每执行一条指令便需增加若干条指令去进行越界检查,这将显著降低程序的运行速度。因此,越界检查都由硬件实现。当然,对发生越界后的处理,还需与软件配合来完成。

内存管理中的内存扩充任务并非是去扩大物理内存的容量,而是借助于虚拟内存技术,从逻辑上去扩充内存容量,使用户所感觉到的内存容量比实际内存容量大得多,以便让更多的用户程序并发运行。这样既满足了用户的需要,又改善了系统的性能。

1.1.3 文件管理

在现代计算机管理中,总是把程序和数据以文件的形式存储在外部存储设备(如磁盘、U盘)上,供所有的或指定的用户使用。为此,在操作系统中必须提供文件管理机制。文件管理的主要任务是对用户文件和系统文件进行管理,以方便用户使用,并保证文件的安全性。为此,文件管理应具有对文件存储空间的管理、目录管理、文件的读/写管理,以及文件的共享与保护等功能。

为了方便用户的使用,一些当前需要使用的系统文件和用户文件必须放在可随机存取的外部存储设备上。在多用户环境下,若由用户自己对文件的存储进行管理,不仅非常困难,而且也必然是十分低效的。因而,需要由文件系统对诸多文件及文件的存储空间实施统一的管理,其主要任务是为每个文件分配必要的外部存储空间,提高存储设备的利用率,并能有助于提高文件系统的读写速度。为此,系统应设置相应的数据结构,用于记录文件存储空间的使用情况,以供分配存储空间时参考;系统还应具有对存储空间进行分配和回收的功能。为了提高存储空间的利用率,对存储空间的分配通常采用离散分配方式,以减少存储碎片,并以盘块为基本分配单位。盘块的大小通常为1~8 KB。

为了使用户能方便地在外部存储设备上找到自己所需的文件,通常由系统为每个文件建立一个目录项。目录项包括文件名、文件属性、文件在设备上的物理位置等。若干个目录项又可构成一个目录文件。目录管理的主要任务是为每个文件建立其目录项,并对众多的目录项加以有效的组织,以实现方便的按名存取,即用户只需提供文件名便可对该文件进行存取。其次,目录管理还应能实现文件共享,这样,只需在外存上保留一份该共享文件的副本即可。此外,还应能提供快速的目录查询手段,以提高对文件的检索速度。

1.1.4 I/O 设备管理

I/O 设备管理用于管理计算机系统中所有的外围设备。设备管理的主要任务是完成用户进程提出的 I/O 请求,为用户进程分配其所需的 I/O 设备,提高 CPU 和 I/O 设