

◆ 一流大学计算机类专业核心课程教材

操作系统原理、 实现与实践

Operating System Principles,
Implementation and Practice

李治军 刘宏伟 编著

高等教育出版社



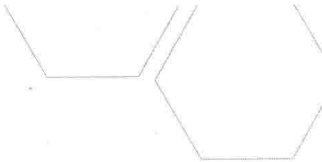
◆ 一流大学计算机类专业核心课程教材



操作系统原理、 实现与实践

Operating System Principles,
Implementation and Practice

李治军 刘宏伟 编著



高等教育出版社·北京

内容提要

本书充分突出操作系统作为一个系统保证整体高速运转的理念,同时亦强调各个组成部分必须有机、协调地分工协作,旨在让读者能真正理解操作系统的整体运转原理。

本书从硬件管理这个基本功能出发,依次对操作系统的CPU管理、内存管理、外设管理、文件管理等进行了详细的阐述;从上层应用提供的接口出发,对系统接口设计、进程管理、文件接口等内容进行了详细阐述。在单独模块阐述的基础上,深入地阐述了这些模块之间的关系,即这些模块是如何合作成为一个整体系统的。

为避免单纯的原理阐述让读者感到空泛、浮于表面,同时加强操作系统原理和实现的有机融合,本书设计了12个实践项目,其中包括8个初级认知型实践项目和4个高级综合型实践项目,几乎涵盖了操作系统的全部内容。

本书可作为理工科本科生操作系统课程教材或参考书,也可供操作系统开发以及在操作系统上进行相关研究的人员参考。

图书在版编目(CIP)数据

操作系统原理、实现与实践 / 李治军, 刘宏伟编著

—北京: 高等教育出版社, 2018.6

ISBN 978-7-04-049245-3

I. ①操… II. ①李… ②刘… III. ①操作系统 - 高等学校 - 教材 IV. ①TP316

中国版本图书馆CIP数据核字(2018)第002437号

Caozuo Xitong Yuanli、Shixian yu Shijian

策划编辑 张海波
插图绘制 杜晓丹

责任编辑 张海波
责任校对 王雨

封面设计 于文燕
责任印制 韩刚

版式设计 马云

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 廊坊市文峰档案印务有限公司
开本 787mm×1092mm 1/16
印张 22.75
字数 470千字
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.com.cn>
版 次 2018年6月第1版
印 次 2018年6月第1次印刷
定 价 45.00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 49245-00

本书配套的数字资源使用方法如下:

1. 访问 <http://abook.hep.com.cn/187715>, 或手机扫描二维码、下载并安装 Abook 应用。

2. 注册并登录, 进入“我的课程”。

3. 输入教材封底防伪标签上的数字课程账号 (20 位密码, 刮开涂层可见), 或通过 Abook 应用扫描封底数字课程账号二维码, 完成课程绑定。

4. 单击“进入课程”按钮, 开始本数字课程的学习。

课程绑定后一年为数字课程使用有效期。受硬件限制, 部分内容无法在手机端显示, 请按提示通过计算机访问学习。

如有使用问题, 请发邮件至 abook@hep.com.cn。



扫描二维码
下载Abook应用

前言

在计算机学科中, 计算机系统在越来越多的领域中发挥着越来越重要的作用。例如, 一台笔记本式计算机可以连接 Wi-Fi 网卡和蓝牙通信设备, 和周围工作的 Wi-Fi 热点设备、蓝牙音箱等组成一个计算机系统, 利用这个系统可以播放互联网上的歌曲。又如, 车载计算机可以连接 GPS、3G 通信设备等形成一个车载计算机系统, 借助这个系统, 我们可以在一个陌生的城市中游览。再如, 各个银行网点柜台的计算机和银行总部的数据中心通过互联网连接在一起形成一个巨大的分布式计算机系统, 正是基于这样的系统, 银行才可以实时地处理客户的存款、取款等各类业务。今天, 这样的计算机系统已经渗透到人们日常生活中的方方面面, 而将来, 这样的计算机系统会更多。

因此培养并建立关于计算机系统的“系统观”并在这种理念下去提出问题、分析问题和解决问题, 对于计算机科学工作者来说是非常必要而且也是非常重要的。很多时候, 一个想法在纸上实现是一回事, 而将其放在实际工作的计算机系统上去实现却完全是另外一回事。维基百科将“系统”定义为: 一个由多个相互关联、相互依赖的模块构成的错综复杂的总体。因此, 只有将问题求解放在这个错综复杂的整体背景下去考虑才有实际意义, 而立足于太多理想假设的求解最终也只能停留在纸面上。

在计算机系统中, 操作系统毋庸置疑处于一个核心位置上。因为从宏观的角度看, 可以将计算机系统分成三个基本组成部分: 底层的计算机硬件、中间层的操作系统以及上层的计算机应用程序, 操作系统属于承上启下的中间层, 所以它在计算机系统中的地位和作用尤为重要。因此, 以操作系统为基础建立对计算机系统的系统认识就显得非常合适了。

本书以“如何给一个拥有单核 CPU、内存、磁盘、键盘和显示器的基本计算机配备一个可实际运转的小操作系统”这一基本问题作为主线, 对这个基本操作系统中的六个基本模块, 即 CPU 管理、内存管理、外设管理、磁盘管理与文件系统、用户接口和启动模块, 以及这些模块之间的内在联系, 进行了深入细致的原理阐述、算法与数据结构设计, 并借鉴开源操作系统 Linux 全面分析了整个操作系统的代码实现。通过本书的学习, 读者可以体会一个完整的操作系统是如何良好运转的, 进而体会一个完整的计算机系统又是如何运转的; 通过本书的学习, 读者可以得到如何设计和实现一个完整操作系统的直接体验, 进而在未来从事计算机系统层面的工作时应用这些经验, 逐渐培养计算机系统能力。

为使操作系统原理和实现有机融合,本书设计了8个小型实践项目和4个大型实践项目,对这12个实践项目给出了非常详细的论述。并且在本书配套的电子资料中提供了这些实践项目所需要实验平台,相信读者在完成这12个实践项目以后,会对操作系统的认识、感受和操控力有质的飞跃。

作者

2017年10月

目录

第一章 系统启动 —— 打开电源以后发生的故事	1
1.1 什么是操作系统	1
1.2 操作系统历史概述	3
1.2.1 简单的批处理系统	3
1.2.2 OS/360 的出现	4
1.2.3 UNIX、Linux 的诞生	7
1.2.4 操作系统历史的另一个分支 —— 人机交互	8
1.3 操作系统核心轮廓	11
1.3.1 多进程视图	11
1.3.2 文件视图	12
1.3.3 学习操作系统从动手实践开始	13
1.4 操作系统启动过程	13
1.4.1 计算机工作机理	14
1.4.2 操作系统启动过程实例 —— 第一阶段	16
1.4.3 操作系统镜像的生成 —— Makefile	25
1.4.4 操作系统启动过程实例 —— 第二阶段	29
1.4.5 操作系统启动过程实例 —— 第三阶段	33
1.4.6 操作系统启动过程实例 —— 第四阶段	37
1.5 实践项目 1: 控制操作系统启动	38
习题一	39

第二章 系统接口 —— 通向操作系统内核的大门	41
2.1 用户使用计算机系统的基本形式	41
2.2 基本系统调用及其应用	43
2.2.1 fork、exec、wait、exit	43
2.2.2 open、read、write	45
2.2.3 printf、scanf	46
2.3 系统调用的实现机理	46
2.3.1 内核态与用户态	46
2.3.2 系统调用与 int 0x80	49
2.3.3 printf 的完整故事	51
2.4 通过系统调用窥见操作系统的基本脉络	54
2.5 实践项目 2: 添加系统调用	55
习题二	58
第三章 多进程 —— 操作系统最核心的视图	59
3.1 如何使用 CPU?	59
3.1.1 CPU 工作机理与使用 CPU 的直观想法	59
3.1.2 问题的引出与并发	60
3.2 进程概念与多进程视图	62
3.2.1 进程与程序	62
3.2.2 CPU 管理与多进程视图	64
3.3 多进程引起的基本问题	65
3.3.1 多个进程的组织与进程状态	65
3.3.2 多个进程的切换和调度	67
3.3.3 进程间的影响分离	69
3.3.4 进程间的通信与合作	70
3.4 实践项目 3: 打印进程日志	73
习题三	75

第四章 线程切换与调度 —— 操作系统的发动机	77
4.1 线程与进程	77
4.1.1 线程概念的引入	77
4.1.2 一个多线程实例	78
4.1.3 线程与进程	79
4.2 用户级线程的切换与创建	80
4.2.1 用户级线程之间的切换	81
4.2.2 用户级线程的创建	84
4.3 内核级线程的切换与创建	87
4.3.1 内核级线程的引出	87
4.3.2 内核级线程之间的切换	90
4.3.3 内核级线程的创建	93
4.4 创建 0 号/1 号进程 —— 多进程视图的起点	95
4.5 CPU 调度	97
4.5.1 CPU 调度的含义与算法准则	97
4.5.2 若干 CPU 调度的基本算法	99
4.5.3 多级反馈队列调度算法的一个具体实现	105
4.6 实践项目 4: 基于内核栈完成进程切换	107
习题四	116
第五章 进程同步 —— 让多个进程的推进合理有序	118
5.1 进程同步问题与睡眠/唤醒	118
5.2 从信号到信号量	120
5.2.1 利用信号解决同步问题	120
5.2.2 将信号扩展为信号量	122
5.2.3 生产者 - 消费者同步问题的信号量解法	124
5.3 临界区 —— 对信号量的保护	125
5.3.1 临界区的引出	125
5.3.2 临界区的软件实现	127

5.3.3	Lamport 面包店算法	130
5.3.4	临界区的硬件实现	132
5.4	信号量的实现与使用	134
5.4.1	信号量的使用	134
5.4.2	有正有负信号量的实现	136
5.4.3	只有正数的信号量的实现	137
5.5	死锁现象及死锁处理	138
5.5.1	死锁现象的出现	138
5.5.2	死锁出现的条件与死锁预防	140
5.5.3	死锁避免	143
5.5.4	死锁检测/恢复与死锁忽略	147
5.6	实践项目 5: 信号量的实现与应用	149
	习题五	150
第六章 内存管理 —— 给程序执行提供一个舞台		152
6.1	内存使用与程序重定位	153
6.1.1	从取指 — 执行到内存使用	153
6.1.2	程序重定位	154
6.2	分段	157
6.2.1	段的概念	157
6.2.2	分段机制下的地址转换	159
6.3	内存分区	162
6.3.1	可变分区与适配算法	162
6.3.2	内存碎片	164
6.4	分页	164
6.4.1	分页机制	164
6.4.2	多级页表与快表	166
6.5	段页式内存管理与虚拟内存	170
6.5.1	如何将段和页结合在一起	170
6.5.2	段、页结合时进程对内存的使用	171

6.5.3 一个真实的段页式内存机制	173
6.6 实践项目 6: 地址映射与共享	179
习题六	184
第七章 换入/换出 —— 用磁盘和时间来换取一个规整的虚拟内存	186
7.1 规整的虚拟内存	187
7.1.1 虚拟内存与换入/换出	187
7.1.2 请求调页	187
7.1.3 页面调入的具体实现	189
7.2 页面换出	190
7.2.1 基本的页面换出算法	191
7.2.2 LRU 算法的准确实现	192
7.2.3 clock 算法	194
7.2.4 页框个数分配与全局置换	196
习题七	198
第八章 设备驱动 —— 从文件视图到 out 指令	200
8.1 设备驱动的基本原理	200
8.1.1 外设的工作原理	200
8.1.2 文件视图	201
8.2 显示器的驱动	202
8.2.1 从 printf 开始	202
8.2.2 文件视图中的大量分支	204
8.2.3 最终到达 mov ax, [pos]	205
8.3 键盘的驱动	208
8.3.1 从键盘中断开始	208
8.3.2 从缓冲队列最终到 scanf	209
8.4 实践项目 7: 终端设备字符显示的控制	211
习题八	212

第九章 文件系统 —— 一个从磁盘到文件再到文件系统的漫长抽象	213
9.1 磁盘工作的基本原理	213
9.1.1 磁盘工作的原理	213
9.1.2 磁盘工作的过程	214
9.1.3 使用磁盘的直观方法	215
9.2 生磁盘的使用	216
9.2.1 第一层抽象: 从扇区到磁盘块请求	216
9.2.2 第二层抽象: 多个进程产生的磁盘请求队列	219
9.2.3 第三层抽象: 从磁盘请求到高速缓存	224
9.3 基于文件的磁盘使用	228
9.3.1 第四层抽象: 引出文件	228
9.3.2 文件的实现	232
9.3.3 第五层抽象: 将整个磁盘抽象成一个文件系统	235
9.3.4 目录解析的代码实现	239
9.4 实践项目 8: proc 文件的实现	242
习题九	246
第十章 大型实践项目一 —— 内核级线程的设计与实现	247
10.1 项目的主旨和基本任务	247
10.1.1 项目主旨	247
10.1.2 项目基本任务	248
10.2 TCB 与 PCB	250
10.3 创建线程与创建进程	252
10.3.1 对 fork 的改造	253
10.3.2 pthread_create 的实现机理分析	256
10.3.3 pthread_create 中的用户栈创建	258
10.3.4 pthread_create 处理 start_routine 的参数	259
10.3.5 内核级线程中的 TCB 创建与内核栈创建	260

10.4	以 TCB 为单位进行 CPU 调度	261
10.5	引入 TCB 以后的其他修改	263
10.5.1	进程退出 (exit)	263
10.5.2	进程状态、时间片等内容的操作	264
10.6	主线程与工作线程的设计与实现	265
	习题十	266
第十一章	大型实践项目二 —— 虚拟内存与交换分区的设计与实现	268
11.1	项目的主旨和基本任务	268
11.1.1	项目主旨	268
11.1.2	项目的基本任务	269
11.2	交换分区的驱动	269
11.2.1	给 Bochs 增加一块硬盘	269
11.2.2	交换分区结构设计	271
11.2.3	交换分区的初始化	273
11.3	进程页面的换出	274
11.3.1	clock 算法的设计	274
11.3.2	页面环形链表的建立和维护	275
11.3.3	扫描动作和换出动作的实现	276
11.3.4	交换分区的写出	278
11.4	进程页面的换入	279
11.5	交换分区使用情况的监控	280
	习题十一	282
第十二章	大型实践项目三 —— 鼠标驱动和简单的图形接口实现	283
12.1	项目的主旨和基本任务	283
12.1.1	项目主旨	283
12.1.2	项目的基本任务	284

12.2	鼠标的驱动	285
12.2.1	鼠标中断的捕获	285
12.2.2	键盘控制器 i8042 和中断控制器 8259A	286
12.2.3	鼠标输入数据的解码	290
12.3	显示器的图形工作模式	293
12.3.1	启动图形模式	294
12.3.2	建立像素点阵与显存之间的映射	296
12.3.3	设置屏幕分辨率	298
12.3.4	开始绘制屏幕	301
12.4	消息驱动框架	303
12.5	可视化应用程序的设计与实现	305
12.5.1	定义图形对象	305
12.5.2	创建一个定时器	306
12.5.3	应用程序的核心结构	307
	习题十二	308
第十三章 大型实践项目四 —— 网卡驱动与网络协议的设计与实现		309
13.1	项目的主旨和基本任务	309
13.1.1	项目主旨	309
13.1.2	项目的基本任务	309
13.2	网卡驱动	310
13.2.1	给 Bochs 安装网卡	310
13.2.2	网卡驱动的基本原理	313
13.2.3	读出网卡的 MAC 地址	316
13.2.4	网卡初始化	318
13.3	ARP 协议的设计与实现	324
13.3.1	ARP 协议的基本原理	324
13.3.2	ARP 数据包以及以太网物理帧的形成	325
13.3.3	利用网卡发送物理帧	328
13.3.4	利用网卡接收物理帧	331
13.3.5	ARP 数据包解析与 ARP 缓存表	334

13.4 ICMP 协议及 ping 命令	336
13.4.1 本项目涉及的协议栈结构	336
13.4.2 ICMP 协议的设计与实现	337
13.4.3 利用 ICMP 协议实现 ping 命令	339
习题十三	342
参考文献	344

第一章

系统启动 —— 打开电源以后发生的故事

操作系统在计算机系统中占据着核心位置,无论是要设计一个计算机系统,例如为卫星设计一个控制系统;还是要部署一个计算机系统,如为一家新公司部署一个信息管理系统;抑或是在已有的计算机系统之上开发一个新应用,如针对某手机开发一款更节能的新型浏览器,都需要对这些计算机系统中的操作系统有深入的理解和分析,甚至要进行设计和实现。

随着计算机技术的发展和普及,对于越来越多的计算机技术人员来说,操作系统已经不再是一个直接假定存在的黑盒子了,他们开始关注操作系统本身,在完成计算机相关开发时会利用操作系统的特性来实现性能提升,甚至会适当调整操作系统来提升计算机系统的整体性能。

本书的目标就是让读者对一个完整操作系统的原理、轮廓、设计和实现建立起完整而深入的认识,并通过大量的实例剖析和动手实践来提高操纵操作系统的能力,这里包括应用操作系统的能力、改造操作系统的能力甚至是自己编写操作系统的能力。

显然,理解操作系统是实现本书目标的基础。操作系统的理解从“走进操作系统”开始,系统启动无疑是操作系统的第一层面纱,“走进操作系统”的旅程从“走进系统启动”开始。本章的核心任务是将那黑色开机屏幕背后的所有故事逐层分析清楚。



1.1 什么是操作系统

实际上,并不需要对操作系统给出一个非常正式的定义,简而言之,操作系统就是安装在计算机硬件之上的一个实实在在的软件,人们通过这个软件可以方便而高效地使用计算机硬件。当计算机硬件环境发生变化时,这个软件就要跟着发生变化,例如底层硬件是一个由多台计算机构成的集群系统,那么操作系统软件就必须支持多机的磁盘共享和协作计算任务。而当计算机应用场景发生变化时,操作系统也要跟着发生变化,如要给导弹系统配备一个计算子系统来完成实时制导工作,那么操作系统软件就必须严格地保证制导程序执行的实时性。此外,当计算机系统的