



浙江省实验教学示范中心建设成果

计算机与软件工程实验指导丛书

操作系统实验指导

欧阳毅 主编



浙江工商大学出版社
ZHEJIANG GONGSHANG UNIVERSITY PRESS

操作系统实验指导

欧阳毅 主编



浙江工商大学出版社
ZHEJIANG GONGSHANG UNIVERSITY PRESS

图书在版编目(CIP)数据

操作系统实验指导 / 欧阳毅主编. — 杭州 : 浙江工商大学出版社, 2012. 8

ISBN 978-7-81140-568-2

I. ①操… II. ①欧… III. ①Linux 操作系统—高等学校—教材 IV. ①TP316.89

中国版本图书馆 CIP 数据核字(2012)第 173714 号

操作系统实验指导

欧阳毅 主编

责任编辑 任晓燕 柯 希

封面设计 王好驰

责任印制 汪 俊

出版发行 浙江工商大学出版社

(杭州市教工路 198 号 邮政编码 310012)

(E-mail: zjgsupress@163.com)

(网址: <http://www.zjgsupress.com>)

电话: 0571-88904980, 88831806(传真)

排版 杭州朝曦图文设计有限公司

印刷 杭州恒力通印务有限公司

开本 787mm×960mm 1/16

印张 16

字数 313 千

版印次 2012 年 8 月第 1 版 2012 年 8 月第 1 次印刷

书号 ISBN 978-7-81140-568-2

定价 32.00 元

版权所有 翻印必究 印装差错 负责调换

浙江工商大学出版社营销部邮购电话 0571-88804227

“计算机与软件工程实验指导丛书”编委会

主 任：凌 云（浙江工商大学）

委 员：赵建民（浙江师范大学）

陈庆章（浙江工业大学）

万 健（杭州电子科技大学）

汪亚明（浙江理工大学）

詹国华（杭州师范大学）

魏贵义（浙江工商大学）

总 序

以计算机技术为核心的信息产业极大地促进了当代社会和经济的发展,培养具有扎实的计算机理论知识、丰富的实践能力和创新意识的应用型人才,形成一支有相当规模和质量的技术人员队伍来满足各行各业的信息化人才需求,已成为当前计算机教学的当务之急。

计算机学科发展迅速,新理论新技术不断涌现,而计算机专业的传统教材特别是实验教材仍然使用一些相对落后的实验案例和实验内容,无法适应当代计算机人才培养的需要,教材的更新和建设迫在眉睫。目前,一些高校在计算机专业的实践教学和教材改革等方面做了大量工作,许多教师在实践教学和科研等方面积累了许多宝贵经验,将他们的教学经验和科研成果转化为教材,介绍给国内同仁,对于深化计算机专业的实践教学改革有着十分重要的意义。

为此,浙江工商大学出版社、浙江工商大学计算机技术与工程实验教学中心及软件工程实验教学中心邀请长期工作在学科教学科研第一线的专家教授,根据多年人才培养及实践教学的经验,针对国内外企业对计算机人才的知识和能力需求,组织编写了“计算机与软件工程实验指导丛书”。该丛书包括《操作系统实验指导》《嵌入式系统实验指导》《数据库系统原理学习指导》《Java 程序设计实验指导》《接口与通信实验指导》《My SQL 实验指导》《软件项目管理实验指导》《软件工程实践实验指导》《软件工程开源实验指导》《计算机应用技术(办公软件)实验指导》等书,涵盖了计算机及软件工程等专业的核心课程。



丛书的作者长期工作在教学、科研的第一线,具有丰富的教学经验和较高的学术水平。教材内容凸显当代计算机科学技术的发展,强调掌握相关学科所需的基本技能、方法和技术,培养学生解决实际问题的能力。实验案例选材广泛,来自于学生课题、教师科研项目、企业案例以及开源项目,强调实验教学与科研、应用开发、产业前沿紧密结合,体现实用性和前瞻性,有利于激发学生的学习兴趣。

我们希望本丛书的出版对国内计算机专业实践教学改革和信息技术人才的培养起到积极的推动作用。

“计算机与软件工程实验指导丛书”编委会

2012年7月

前 言

操作系统是计算机系统的核心软件,在计算机系统中占有极其重要的地位。学习操作系统就是要理解并熟练地掌握并运用它,为深入掌握计算机专业知识,进一步提升软件开发设计能力打下坚实的基础。而要真正理解操作系统的设计理念并非易事,一方面需要进行理论学习,另一方面则需要一个好的实验教材指导上机调试。

本教材针对操作系统内容庞杂、涉及面广的特点,在实验内容的选取上以 Linux 操作系统为平台,注重基础性、实践性、先进性。在框架的设计上注重逻辑性、完整性,将操作系统的内容组织成一个完整的逻辑整体。提炼现代操作系统中多用户、多任务的本质特性,始终贯穿并发、共享的主线。以 Linux 内核源码为基础,深入浅出地描述现代操作系统的动态特性,论述操作系统的重要概念——进程,支持多进程并发活动必需的数据结构,进程控制与进程调度等功能的实现技术,对于并发活动中进程同步和死锁等问题的解决策略与方法。

本教材针对操作系统实践性强的特征,以 Linux 操作系统为例,给出了一个操作系统实验体系。该实验体系涉及操作系统的核心实验和一些重要的实际应用,主要包括 Linux 系统中进程管理、进程间通信、进程同步、内存管理、死锁的处理和文件系统等模块。每个实验都是一个完整的小体系,包括:预备知识、实验目的、实验内容和实验中的程序范例。所有范例仅供参考,学生上机时需自行设计独立完成,同时给出更高层次的上机思考题。

在教材的内容上,操作系统原理的论述和实验内容的讨论相互关联,相互印证,读者能以操作系统原理为指导,针对每个实验去思考自己要做什么,通过实验知道如何去做,实验完成后思考为什么这样做,有没有更好的方法。这样,每次实验都会有所收获,能更好地理解操作系统中的相关概念,操作系统实现的机制和策略,体验操作系统各功能模块的实现技术。

本书由浙江工商大学计算机与信息工程学院的欧阳毅老师、浙江大学宁波理工学院计算机研究所的朱朝艳老师、浙江大学宁波理工学院计算机研究所的童有

成老师共同合作编著。欧阳毅老师负责第一、二、三、四、七章的全部实验,以及第五章的实验四和第六章的实验三,并负责统稿工作;朱朝艳老师负责第五章的实验一,实验二和实验三;童有成老师负责第六章的实验一和实验二。编写过程中,得到了浙江工商大学计算机与信息工程学院实验教学示范中心的大力支持,浙江工商大学出版社的柯希和蒋红群老师对本书进行详细校对并提出了宝贵的意见,使本书得以顺利出版。在此谨向他们表示衷心的感谢。

编著者

2012 年 5 月

目 录

绪 论	1
第一章 Linux 源码简介	3
实验一 Linux 内核编译	4
实验二 添加系统调用	21
第二章 进程管理	26
实验一 Linux 进程创建	29
实验二 Linux 进程间通信——1 管道通信和消息传递	43
实验三 Linux 进程间通信——2 共享内存	53
第三章 进程同步和互斥	64
实验一 Linux 系统进程同步和互斥	67
实验二 死锁——银行家算法	72
第四章 Linux 中进程调度	84
实验一 CPU 调度算法	86
实验二 CPU 各种调度算法	105
第五章 内存管理	116
实验一 动态分区管理的主存分配与回收	117
实验二 分页管理的主存分配与回收	133



实验三 虚拟存储器中的页面置换算法	148
实验四 统计操作系统缺页次数	156
第六章 Linux 文件系统	161
实验一 设计并实现一个二级文件系统程序	167
实验二 设计并实现一个 ext2 文件系统跟踪程序	197
实验三 Linux 中添加一个文件系统	207
第七章 磁盘调度	232
实验一 磁盘调度	233
实验二 磁盘调度	239
参考文献	245



绪 论

操作系统是现代计算机系统中不可缺少的基本系统软件。操作系统管理和控制计算机系统中的所有软、硬件资源,是计算机系统的灵魂和核心。除此之外,它还为用户提供计算机提供一个方便灵活、安全可靠的工作环境。学习并掌握计算机操作系统的基本原理和设计方法,不仅对计算机专业的学生 and 研究人员是必要的,而且对一般计算机应用人员也是非常有益的。

《操作系统原理》(或称《操作系统》)课程是计算机各专业的必修课程和主干课之一。它在学生的计算机专业知识结构中有着重要的地位和作用,它为学生学习以后各主要课程的学习打下扎实的基础。作为计算机基础理论的一部分,操作系统原理有一定难度。原因之一是除了要分析操作系统软件及实行算法等特性外,还要考虑计算机硬件等特性。本书介绍了一套系统的操作系统实验内容,它的设计思路、实现方法以及操作步骤,能引导学生参与操作系统内核交互,以观察其原理。本实验教材可以较好地帮助他们“理论联系实际”,学习操作系统原理。同时书中主要实验采用内核代码公开的 Linux 操作系统作为研究对象,以便学生可以对操作系统的实现细节展开深入学习。

本书作者在学习掌握操作系统原理的经典理论基础之上,通过大量阅读 Linux 内核代码来理解其实现的原理,进而设计了一组实验,包括 Linux 源码简介、Linux 内核编译、添加系统调用、进程管理实验、进程同步和互斥、CPU 调度算法、Linux 文件系统和磁盘调度实验。实验涉及的知识涵盖了操作系统主要模块功能。实验指导学生从内核分析入手深入操作系统各模块的构成和算法设计,在构思上与以往的操作系统原理实验相比更贴近内核,能更好的使学生理解操作系统原理。设计这套系统实验时融入了:高级语言编程 C/C++、内核裁剪、内核修改、内核编译以及系统编程等技术。

本教材的目的是使学生系统掌握操作系统的基本理论、设计方法和实现技术。课件内容安排具体如下:



第一章介绍 Linux 源码的组成、主要文件及目录结构、Linux 内核编译和添加系统调用。

第二章介绍进程管理采用的主要技术,其中包括 Linux 进程创建、管道通信和消息传递和共享内存。

第三章介绍进程同步和互斥,以及 Linux 系统进程同步和互斥实验和死锁——银行家算法实验。

第四章介绍 Linux 中进程调度,主要对操作系统原理理论课讲授的 CPU 各种调度算法进行实现。

第五章介绍内存管理,包括采用动态分区管理的主存分配与回收,分页管理的主存分配与回收,统计操作系统缺页次数和虚拟存储器中的页面置换算法实验。

第六章介绍 Linux 文件系统,通过设计并实现一个二级文件系统程序,设计并实现一个 ext2 文件系统跟踪程序和 Linux 中添加一个文件系统。

第七章介绍磁盘调度,对各种磁盘调度算法进行实现。



第一章 Linux 源码简介

本章重点介绍 Linux 操作系统内核代码的分布情况,和虚拟机 VMWare 的使用。本章描述了在 Linux 核心源程序中,应该从什么地方开始查看特定的核心功能。

Linux 的应用领域比较广泛。在某些领域中 Linux 具有其他操作系统无法比拟的优势,特别是在网络服务器方面, Linux 以其稳定及强大的性能占据了非常重要的地位。Linux 主要应用于网络服务器,工作站,嵌入式系统,个人用户桌面和视频制作等应用领域。

Linux 是当今流行的操作系统之一。由于其源码的开放性,现代操作系统设计的思想和技术能够不断运用于它的新版本中。因此,读懂并修改 Linux 内核源代码无疑是学习操作系统设计技术的有效方法。本章首先介绍 Linux 内核的特点、源码结构和重新编译内核的方法,讲述如何通过 Linux 系统所提供的 /proc 虚拟文件系统了解操作系统运行状况的方法。最后,对 Linux 编程环境中的常用工具也作简单介绍。

Linux 操作系统在短短几年时间内得到迅猛发展,与 Linux 具有良好特性是分不开的。简单地说, Linux 具有以下主要特性。

第一,开放性。开放性是指系统遵循世界标准规范,特别是遵循开放系统互联 (OSI) 国际标准。Linux 是一种开源软件,其源代码公开发放,同时使用者可以根据个人的具体要求对 Linux 的内核或外壳进行修改并发行,甚至构建个性化的系统。这使得 Linux 用户不断增多,功能也日趋完善。

第二,多用户。多用户是指系统资源可以被不同的用户各自拥有使用,即每个用户对自己拥有的资源有独立的权限,互不影响。Linux 提供虚拟控制台的访问方式,可以同时接受多个用户登录,还允许用户在同一时间从控制台进行多次登录,是一个真正意义上的多用户操作系统。

第三,多任务。多任务是现代操作系统的主要特点之一,它是指计算机同时执



行多个程序,同时各程序能独立运行,互不干扰。由于目前 CPU 处理速度非常快, Linux 对应用程序调度的开销也较少,因而系统运行起来更顺畅,启动多个应用程序看起来像在并行运行。

第四,良好的用户界面。Linux 向用户提供了两种界面:操作界面和程序设计界面。操作界面包括传统的基本文本的命令行界面(Shell,也称字符终端用户界面)和图像化用户界面。Shell 既可以联机使用也可以脱机使用,有很强的程序设计能力。Linux 也提供一个直观、易操作、且交互性能友好的图形化界面。系统调用是用户编程时使用的界面。用户可以在编程时直接使用系统提供的系统调用命令或函数。

第五,强大的网络服务。Linux 在通信和网络服务方面优于其他操作系统,为用户提供了完善的、强大的网络服务。完善的内置网络服务是 Linux 的一大特点。目前 Linux 支持的网络服务主要包括 WWW 服务、电子邮件服务、FTP 服务、DNS 和 DHCP 服务、Telnet 服务、NIS 和 NFS 服务等。

第六,设备独立性。Linux 具有设备独立性,它把所有外部设备当做文件来处理,只要安装它们的驱动程序,任何用户都可以像使用文件一样使用这些设备,而不必知道它们的具体实现形式。Linux 内核具有高度适应能力,使更多的硬件设备可以加入到各种 Linux 内核和发行版本中。同时,用户也可根据需要修改内核源代码,以便适应新增加的外部设备。

第七,良好的可移植性。Linux 是一种可移植的操作系统,可以在不同的平台上运行。当 Linux 操作系统从一个平台转移到其他平台上时仍然可以运行而无需对系统做修改。可移植性为 Linux 在不同设备之间的通信提供了可靠而稳定的连接方式,而不需要另外增加特殊且昂贵的通信接口。

实验一 Linux 内核编译

一、预备知识

作为自由软件,在广大爱好者的支持下, Linux 内核版本不断更新。新的内核修订了旧内核的 bug,并增加了许多新的特性。如果用户想要使用这些特性,或想根据自身系统量身定制一个更高效、更稳定的内核,就需要重新编译内核。通常,更新的内核会支持更多的硬件,具备更好的进程管理能力,运行速度更快、更稳定,并会修复旧版本中发现的许多漏洞等。经常升级更新的系统内核是 Linux 使用者

的必要操作内容。

要增加 Linux 对某部分功能的支持,比如网络服务,可以把相应部分编译到内核中,编译成模块以动态调用。如果将要增加的功能编译到内核中,在内核启动时可以自动支持相应部分的功能,其优点是方便快捷,机器一旦启动,就可以使用这部分功能;缺点是会导致内核变得庞大,不管是否需要这部分功能,它都存在。建议仅将经常使用的部分直接编译到内核中。而如果将某功能编译成模块,就会生成对应的 .ko 文件,使用时动态加载,优点是会使内核过分庞大,缺点是需要手工调用这些模块。

1. 下载内核源代码

Linux 操作系统有 GNU 通用公共许可证(GPL)保护,故其内核源代码是完全开放的。现在很多 Linux 的网站都提供内核代码的下载。推荐使用 Linux 的官方网站(<http://www.kernel.org>),在这里可以找到所有的内核版本,如图 1-1 所示。

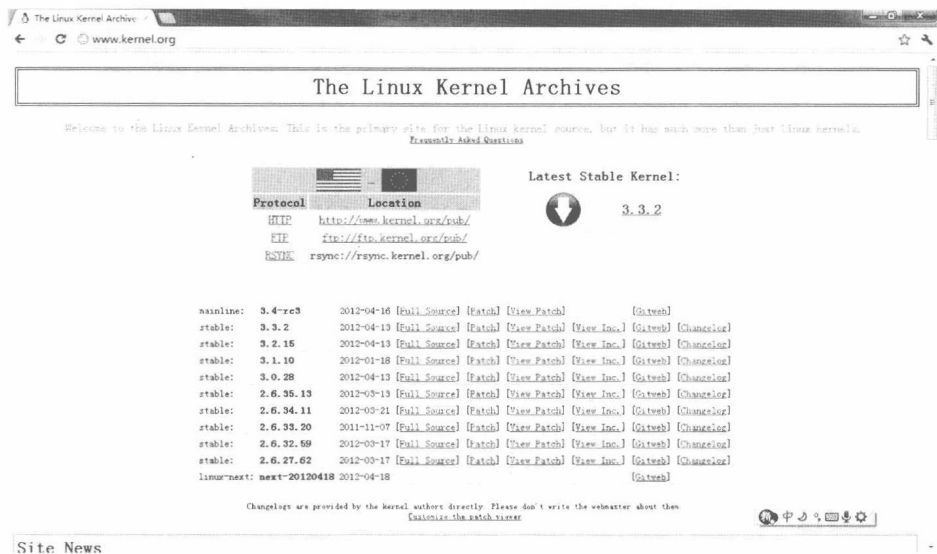


图 1-1 Linux 官方网站

(1) Linux Kernel Sources(Linux 核心源程序)。Linux 核心源程序有一个非常简单的编号系统。任何偶数的核心(例如 2.4.30)都是一个稳定的发行的核心,而任何奇数的核心(例如 2.5.42)都是一个开发中的核心。开发版的核心具有所有的最新的特点和设备支持,它们可能不稳定,但是让 Linux 社团测试最新核心是很重要的。这样可以让整个社团都进行测试。即使测试非生产用核心,最好也要备份系统。本书基于稳定的 2.4.32 源代码,在源程序树的最上层会看到一些目录:

Arch 子目录包括所有和体系结构相关的核心代码。它还有更深的子目录,每



一个代表一种支持的体系结构,例如 i386 和 alpha。

Include 子目录包括编译核心所需要的大部分 include 文件。它也有更深的子目录,分别对应所支持的各个体系结构。Include/asm 是这个体系结构所需要的真实的 include 目录的软链接,例如 include/asm-i386。为了改变体系结构,你需要编辑核心的 makefile,重新运行 Linux 的核心配置程序。

Init 子目录包含核心的初始化代码,这是研究核心如何工作的一个非常好的起点。

Mm 子目录包括所有的内存管理代码。和体系结构相关的内存管理代码位于 arch/* /mm/,例如 arch/i386/mm/fault.c

Drivers 系统所有的设备驱动程序。它们被划分成设备驱动程序类,例如 block。

Ipc 子目录包含核心的进程间通讯的代码。

Modules 这只是一个用来存放建立好的模块的目录。

Fs 所有的文件系统代码。被划分成子目录,每一个支持的文件系统一个子目录,例如 vfat 和 ext2。

Kernel 主要的核心代码。同样,和体系相关的核心代码放在 arch/* /kernel。

Net 核心的网络代码。

Lib 子目录放置核心的库代码。和体系结构相关的库代码放在 arch/* /lib。

Scripts 子目录包含脚本(例如 awk 和 tk 脚本),用于配置核心。

Memory Management(内存管理)代码大多存在 mm 下,但是和体系结构相关的代码在 arch/* /mm。Page fault 处理代码在 mm/memory.c 中,内存映射和页缓存代码在 mm/filemap.c 中。Buffer cache 在 mm/buffer.c 中实现,交换缓存在 mm/swap_state.c 和 mm/swapfile.c 中。

Kernel:大部分相对通用的代码存在 kernel 下,和体系结构相关的代码在 arch/* /kernel。调度程序在 kernel/sched.c, fork 代码在 kernel/fork.c。bottom half 处理代码在 include/linux/interrupt.h。task_struct 数据结构可以在 include/linux/sched.h 中找到。

Pci: Pci 伪驱动程序在 drivers/pci/pci.c,系统范围的定义在 include/linux/pci.h。每一种体系结构都有一些特殊的 Pci BIOS 代码, Alpha axp 的位于 arch/alpha/kernel/bios32.c。

Interprocess Communication:全部在 Ipc 目录。所有系统 V IPC 对象都包括 ipc_perm 数据结构,可以在 include/linux/ipc.h 中找到。系统 V 消息在 ipc/msg.c 中实现,共享内存在 ipc/shm.c 中,信号灯在 ipc/sem.c,管道在 ipc/pipe.c 中实现。



Interrupt Handling: 核心的中断处理代码几乎都是和微处理器(通常也和平台)相关。Intel 中断处理代码在 `arch/i386/kernel/irq.c` 中,它的定义在 `include/asm-i386/irq.h`。

Device Drivers (设备驱动程序): Linux 核心源代码的大部分代码行在它的设备驱动程序中。Linux 所有的设备驱动程序源代码都在 `drivers` 中,但是它们被进一步分类:

`/block` 块设备驱动程序。如果希望查看所有可能包含文件系统的设备是如何初始化的,可以看 `drivers/block/genhd.c` 中的 `device_setup()`。它不仅初始化硬盘,也初始化网络,因为安装 `nfs` 文件系统的时候需要网络。块设备包括基于 IDE 和 SCSI 设备。

`/char` 这里可以查看基于字符的设备,比如 `tty`,串行口等。

`/cdrom` 所有的 CDROM 代码。在这里可以找到特殊的 CD-Rom 设备(比如 Soundblaster CDROM)。注意 IDE CD 驱动程序是 `drivers/block` 中的 `ide-cd.c`,而 SCSI CD 驱动程序在 `drivers/scsi/scsi.c` 中。

`/pci` PCI 伪驱动程序。这是一个观察 PCI 子系统如何被映射和初始化的好地方。Alpha AXP PCI 整理代码也只能在 `arch/alpha/kernel/bios32.c` 中查看。

`/scsi` 在这里不但可以找到所有的 Linux 支持的 SCSI 设备的驱动程序,也可以找到所有的 SCSI 代码。

`/net` 在这里可以找到网络设备驱动程序比如 DEC Chip 21040 PCI 以太网驱动程序在 `tulip.c` 中。

`/sound` 所有的声卡驱动程序的位置。

File Systems (文件系统): `ext2` 文件系统的源程序都在 `fs/ext2` 子目录,数据结构的定义在 `include/linux/ext2_fs.h`,`ext2_fs_i.h` 和 `ext2_fs_sb.h` 中。虚拟文件系统的数据结构在 `include/linux/fs.h` 中描述,代码在 `fs/*` 中。Buffer cache 和 update 核心守护进程都是用 `fs/buffer.c` 实现的。

Network(网络): 网络代码放在 `net` 子目录,大部分的 `include` 文件在 `include/net`。BSD socket 代码在 `net/socket.c`,Ipv4 INET socket 代码在 `net/ipv4/af_inet.c` 中。通用协议的支持代码(包括 `sk_buff` 处理例程)在 `net/core` 中,TCP/IP 网络代码在 `net/ipv4`。网络设备驱动程序在 `drivers/net` 中。

Modules (模块): 核心模块代码部分在核心,部分则在 `modules` 包中。核心代码全部在 `kernel/modules.c`,数据结果和核心守护进程 `kerneld` 的消息则分别在 `include/linux/module.h` 和 `include/linux/kerneld.h` 中。也可在 `include/linux/elf.h` 中查看一个 ELF 目标文件的结构。

(2)配置内核。进行这项工作之前,先阅读 `/usr/src/linux` 目录下内核源代码