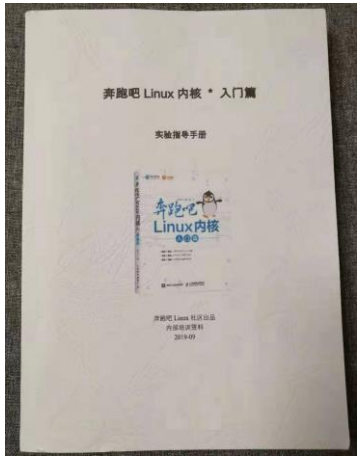
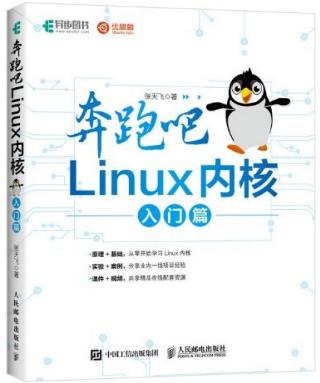
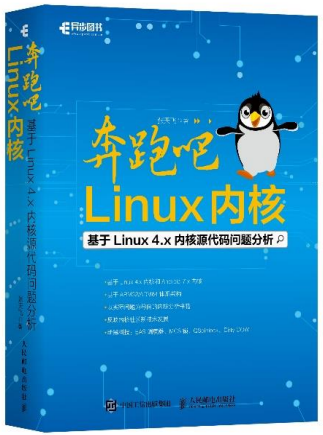
操作系统课程实验新尝试和探索

关于我

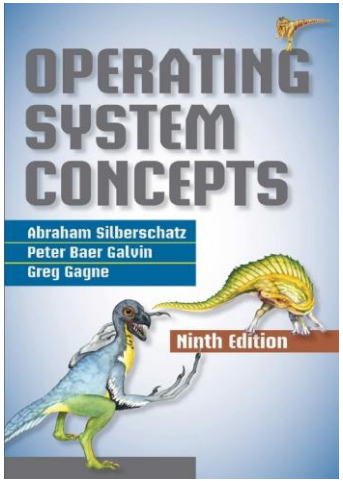
- 网名：笨叔
- 从事10几年Linux驱动和内核相关工作
- 2017年出版《奔跑吧Linux内核》
- 2019年出版《奔跑吧Linux内核 * 入门篇》
- 2019年公开小册子《Linux实验指导手册》

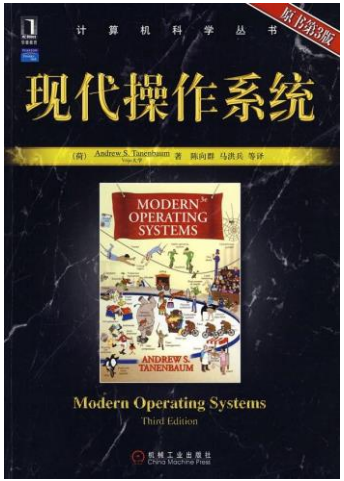


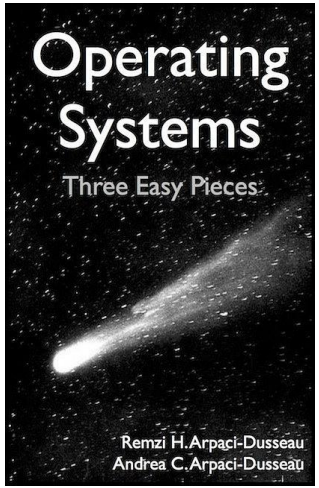


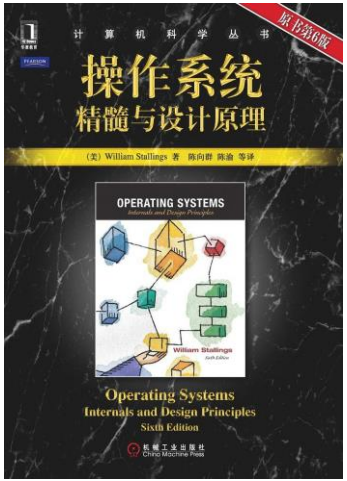


国外经典操作系统教材

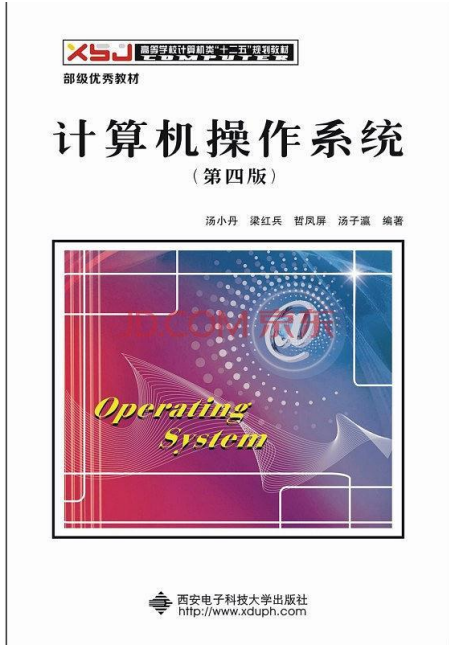

OPERATING
SYSTEM
CONCEPTS
Abraham Silberschatz
Peter Baer Galvin
Greg Gagne
Ninth Edition

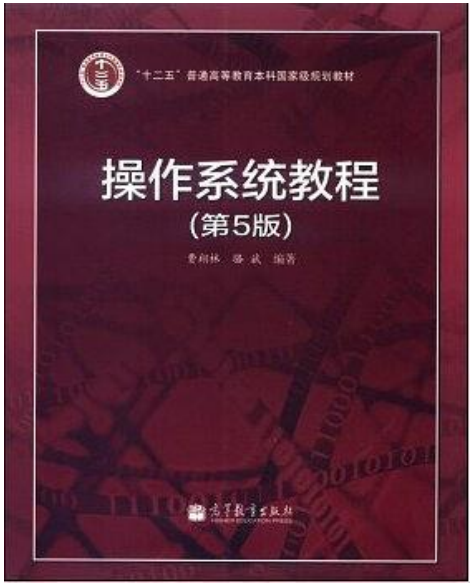

计 算 机 科 学 丛 书
现代操作系统
MODERN OPERATING SYSTEMS
ANDREW S. TANENBAUM
Modern Operating Systems
Third Edition


Operating
Systems
Three Easy Pieces
Remzi H.Arpaci-Dusseau
Andrea C.Arpaci-Dusseau


计 算 机 科 学 丛 书
操作系统
精髓与设计原理
OPERATING SYSTEMS
Operating Systems
Internals and Design Principles
Sixth Edition

国内经典操作系统教材

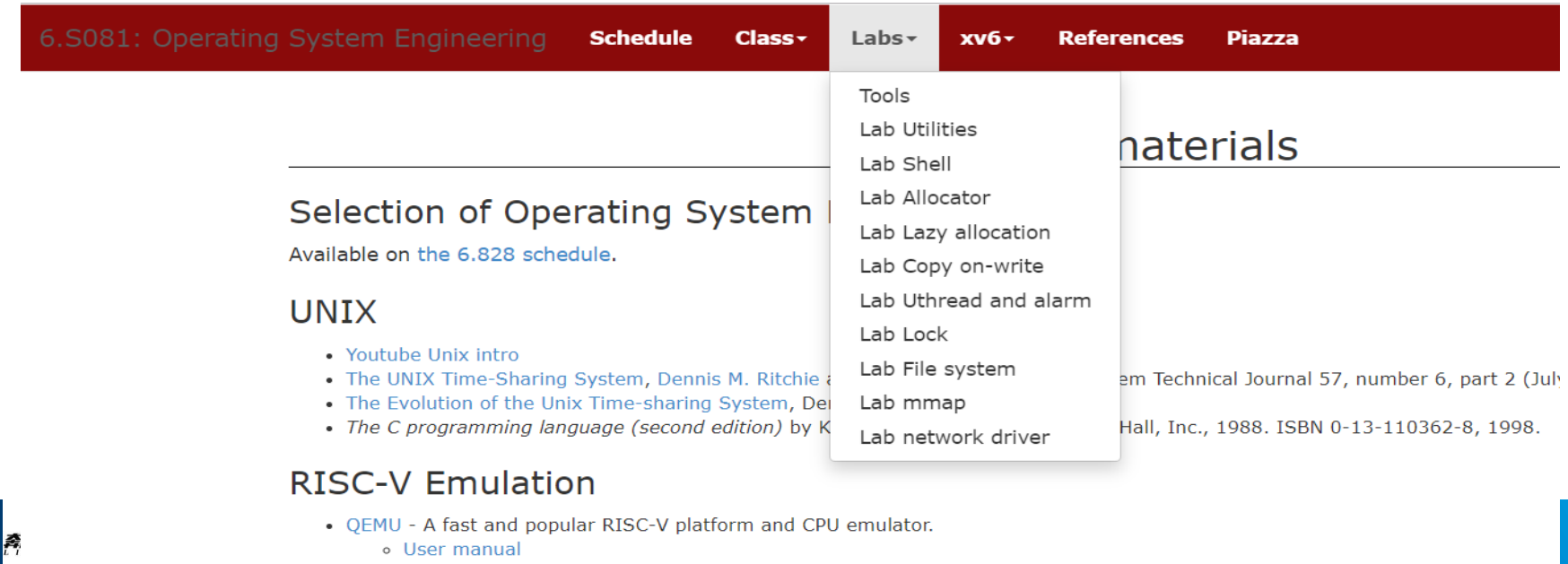

部级优秀教材
计算机操作系统
（第四版）
汤小丹 梁红兵 哲凤屏 汤子瀛 编著
Operating System
西安电子科技大学出版社
http://www.xduph.com


操作系统教程
（第5版）

操作系统课程实验

- 大部分985高校的操作系统课程采用xv6作为实验素材 - 基于MIT 6.828课程

- 2019年的6.828课程 实验采用RISC-V的xv6
- https://pdos.csail.mit.edu/6.828/2019/index.html



xv6实验优缺点

- 优点：
  - ✓ 搭建了一个小os，循序渐进地引导学生实现一些核心的功能
  - ✓ 对体系结构（比如RSIC-V，X86）会有深入理解
  - ✓ 对操作系统基本原理和实现有更深入理解

- 缺点：
  - ✓ 对初学者难度比较大，比如cow实验
  - ✓ 对操作系统不感兴趣的同学可以直接放弃了
  - ✓ 有些学生从github直接拷贝和粘贴答案

xv6的不足

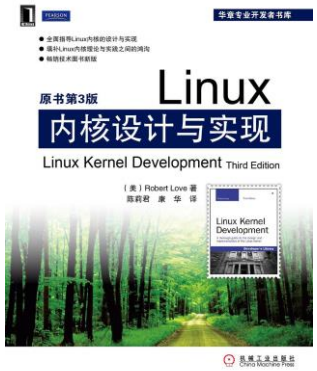
- 基于教学的小型os，类似Linux 0.11
- 离现在工业界使用的操作系统成熟度还差很远
- 离实际工业界需求有点远
- 不能立马学以致用，到企业之后还要重新学习

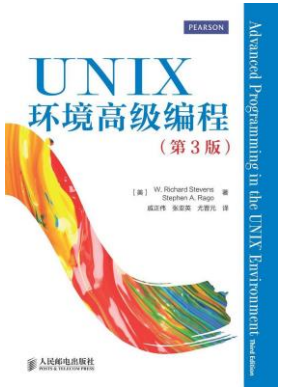
个人建议

- 对于学有余力或者对OS方向感兴趣的同学，独立完成xv6实验后，可以选择一个工业界主流的操作系统来研究：比如开源的Linux内核
- 对于没有采用xv6实验的高校，也可以直接采用Linux为实验对象

本科Linux内核和应用编程课程 （选修）

- 部分高校本科有Linux内核和应用编程课程 （西邮、上交大等）
- 以介绍Linux内核基本框架和基本概念为主要内容
- 介绍Linux应用编程接口





高级操作系统课程 （某985高校研究生课程）

高级操作系统主要教学计划：（八周，每周4节课，32学时，2学分）

1. 操作系统相关理论，1周；

2.Linux操作系统内核结构、进程管理、存储管理、中断、系统调用、设备驱动，2周；

3.虚拟化与容器，分布式系统，1周；

4. 第5到8周，学生分组做报告，主要涉及：Linux内核近2年新增模块/功能分析，嵌入式开源OS分析，操作系统前沿技术研究（针对OSDI、SOSP、FAST、USENIX Security等国际知名OS会议上的新技术动向开展研究）。

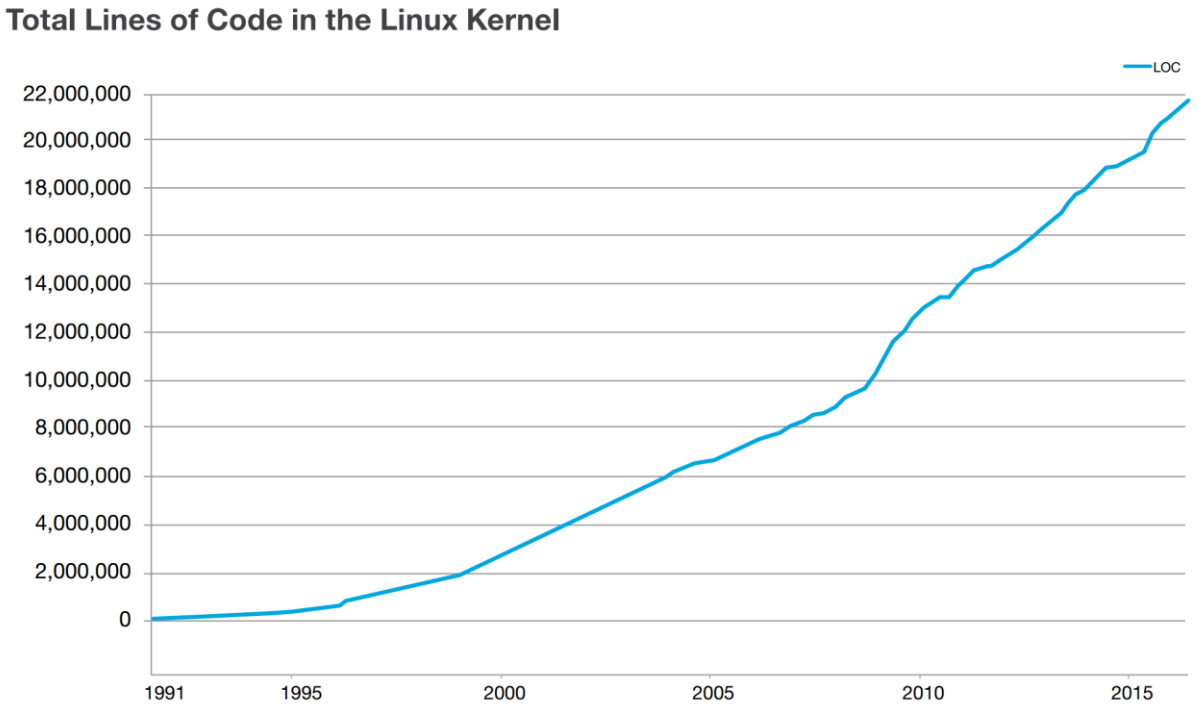
采用Linux作为实验对象的好处

- 优点：
  - ✓ 学以致用，到企业里立马上手
  - ✓ 学习最新最先进的操作系统理念和技术，比如EAS调度器，NUMA，异构计算等
  - ✓ 学习开源编程规范、理念、文化

- 缺点：
  - ✓ Linux内核代码太庞大，学习曲线骤增
  - ✓ Linux内核更新太频繁
  - ✓ 没有很好的入门教程和实验素材

Linux内核行数的变化

- Linux内核从1991年的1w行代码发展到超过200w行代码



搭建一个开放的实验平台

- 基于QEMU + Linux 4.x （Linux 5.x）
- 支持x86_64、ARM64、RISC-V等主流体系结构
- 基于busybox小文件系统或者Debian Rootfs文件系统
- “00”编译Linux内核
- Eclipse + GDB 图形化调试内核
- 70多个实验案例，部分实验案例源自实际项目
- Git仓库：https://github.com/figozhang/runninglinuxkernel_4.0
- 400多页实验指导手册免费下载
- 提供统一的实验平台：vmware镜像

我们想做的事情…

- 免费和开放的平台
- 入门实验 和免费的实验指导手册（目前提供70多个入门实验）
- 持续不断增加有趣的实验和修订实验指导手册
- 从企业中实际项目中抽闲出有趣的实验 （比如新增的特工队实验）
- 从广大一线工程师中收集有趣的实验
- …

特工队实验例子

7.9 实验 9：特工队：修改敌军计算机的系统调用（新增）

1．实验目的

- 通过本实验可以深刻理解操作系统中的页表是怎么构建的。
- 如何去遍历页表。
- 如何修改页表。
- 深刻理解各级页表中页表项的各个字段是什么含义。
- 系统调用在内核是如何被调用的。
- 当编写的驱动发生 crash 和 panic 时，如何去 debug？

注意：本实验有相当难度，学有余力的同学可以尝试做本实验。

2．实验要求

假设你是一名安全特工，正在执行一项秘密任务，这项秘密任务就是要深入到敌军的作战指挥中心的计算机里安装一个窃听的程序，简单来说就是把计算机的系统调用动态替换掉。假设你的同事已经帮你把敌军计算机的 root 密码给破解了，接下来就看你如何动态修改系统调用了。注意：编写的内核模块不能让敌军的计算机重启、crash/panic，否则就暴露行踪，秘密行动失败。

1）编写一个内核模块。

实验环境：ARM64 体系结构，Linux 5.0 内核。

要求替换系统调用表（sys_call_table）中某一项系统调用，替换成自己编写的系统调用处理函数（例如：my_new_syscall()），在新的系统调用函数中打印一句“hello, I have hacked this syscall”，然后再调用回原来的系统调用处理函数。

比如以 ioctl 系统调用为例，它在系统调用表中的编号就是__NR_ioctl。那么需要修改系统调用表 sys_call_table[__NR_ioctl]的指向，让其指向 my_new_syscall()函数，然后在 my_new_syscall()函数中打印一句话，调用原来的 sys_call_table[__NR_ioctl]指向的处理函数。

2）卸载模块时候把系统系统表恢复原样。

3）用 clone 系统调用来验证你的驱动，clone 系统调用号是__NR_clone。

提高初学者兴趣

- 源自实际项目的案例抽象
- 历经4次Linux宕机
- 涉及Linux多个核心模块
  - ARM64体系结构
  - ARM64内存管理
  - Linux内核内存管理
  - 系统调用
  - Linux宕机分析

```
root@benshushu:/mnt# insmod testsyscall_issue.ko
[  191.649281] testsyscall_issue: loading out-of-tree module taints kernel.
[  191.679779] testsyscall_issue: module verification failed: signature and/or required key missing - tainting kernel
[  191.817883] Found the sys_call_table at ffff000011732b20.
[  191.964776] replace system call ...
[  191.965620] walk_pagetable get pte=0x7FFFC003
[  191.965807] mkwrite pte=0x800007FFFC403
[  191.967559] walk_pagetable: pte=0x800007fffc403
[  191.968034] got sys_call_table[29] at              0.
[  191.984000] Unable to handle kernel write to read-only memory at virtual address ffff000011732c08
[  191.984900] Mem abort info:
[  191.985070]   ESR = 0x9600004f
[  191.985330]   Exception class = DABT (current EL), IL = 32 bits
[  191.985566]   SET = 0, FnV = 0
[  191.985720]   EA = 0, S1PTW = 0
[  191.985911] Data abort info:
[  191.986069]   ISV = 0, ISS = 0x0000004f
[  191.987128]   CM = 0, WnR = 1
[  191.988815] swapper pgtable: 4k pages, 48-bit VAs, pgdp = (____ptrval____)
[  191.989668] [ffff000011732c08] pgd=000000007ffff003, pud=000000007fffe003, pmd=000800007fffc403, pte=00e000004173279
3
[  191.994840] Internal error: Oops: 9600004f [#1] SMP
[  191.996028] Modules linked in: testsyscall_issue(OE+)
[  191.998209] CPU: 1 PID: 565 Comm: insmod Kdump: loaded Tainted: G           OE     5.0.0 #1
[  191.999715] Hardware name: linux,dummy-virt (DT)
[  192.000771] pstate: 60000005 (nZCv daif -PAN -UAO)
[  192.003707] pc : hack_syscall_init+0x49c/0x1000 [testsyscall_issue]
[  192.004347] lr : hack_syscall_init+0x438/0x1000 [testsyscall_issue]
[  192.004973] sp : ffff80002451f5b0
[  192.005440] x29: ffff80002451f5b0 x28: ffff800024549c00
[  192.006139] x27: 0000000000000000 x26: 0000000000000000
[  192.007305] x25: 0000000056000000 x24: 0000000000000015
[  192.008007] x23: 0000000040001000 x22: 0000ffff88ca4d44
```

谢谢观赏

Backup

实验指导手册　实验目录

第 9 章 同步管理

第 10 章 中断管理

第 12 章 开源社区

第 13 章 块设备和文件系统（新增）

第 11 章 调试和性能优化

实验指导手册下载

登录“奔跑吧linux社区”微信公众号：

- 输入“实验指导手册”下载实验指导手册。
- 输入“全套资料”下载全套配套实验平台和资料。
- 输入“免费视频”下载免费视频。

Linux内核架构概貌