

操作系统导论

吴化尧

hywu@nju.edu.cn

2025 Spring

课程信息

- 周二 2-4节 @ 仙 II -405
- 课程 QQ 群: 1034852757
 - 修改群昵称为 姓名 + 学号
 - 统计随堂练习提交的依据
- 课程网站: <https://gist.nju.edu.cn/course-os/>
 - 课件下载 (每次课前更新)
 - 课堂示例代码
 - 实验说明

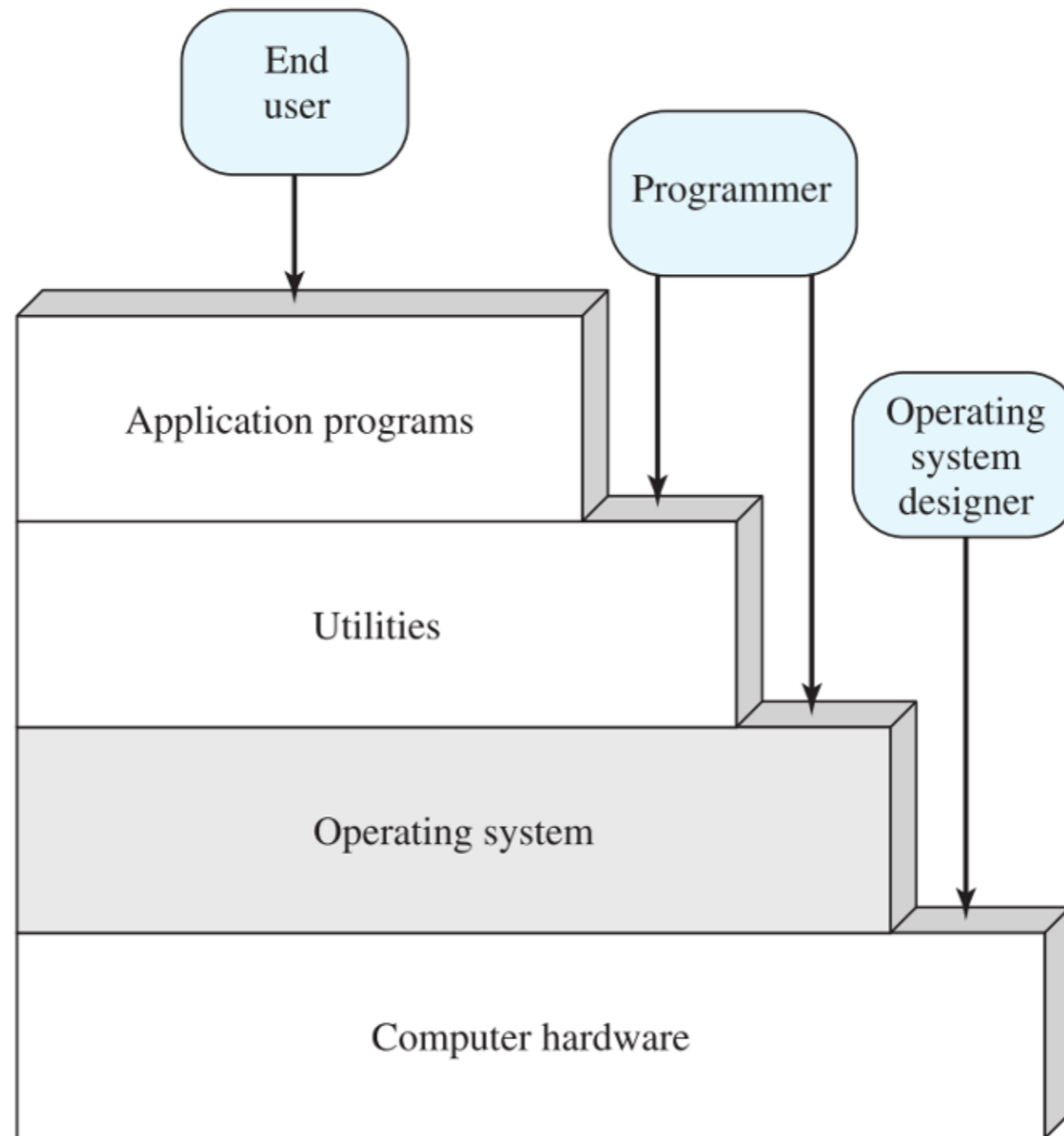
为什么要学操作系统？

操作系统在我们的生活中无处不在

- 每天使用的各种设备上都在运行着某个 OS
- 写的每个程序通常都运行在某个 OS 之上，并且其行为和性能会受 OS 的影响

为什么要学操作系统？

知道系统如何工作、以及如何工作得更好



为什么要学操作系统？

知道系统如何工作、以及如何工作得更好

- 为什么键盘输入能变成屏幕上的文字？
- 100 个 APP 是如何在一个 8 core CPU / 16 GB RAM 上运行的？
- 电脑里的“文件夹”到底是个什么？
- 如何写一个让多核 CPU 跑满 100% 的程序？
- GDB 是如何实现的？
- ...

为什么要学操作系统？

Make you a more capable programmer

- 系统的实现能力
 - 理解一个复杂系统是如何构建和工作的
 - 掌握 OS 设计和实现中相关的基本动机和基本方法
- 系统的编程能力
 - 熟悉 OS 提供的各种对象及其使用方式
- 系统的思考能力
 - OS 的各种核心思想体现在计算机科学的方方面面
 - 抽象、虚拟化、并发、资源管理、性能和可靠性等

为什么要学操作系统？

Make our lives better (or worse)

为什么操作系统难学？

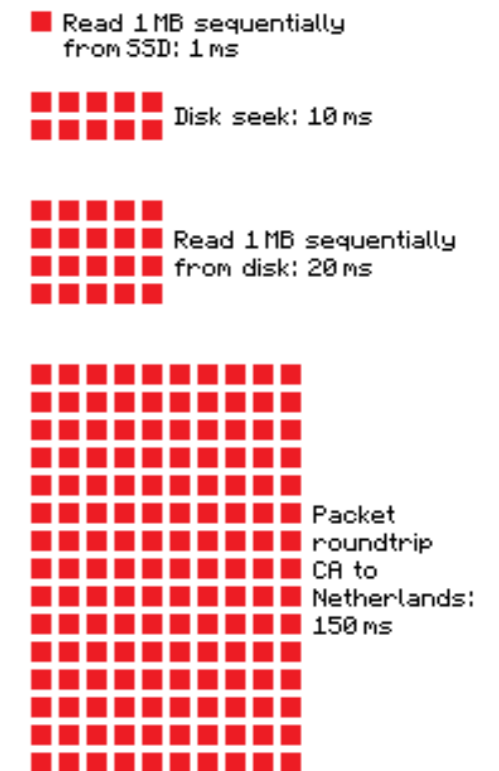
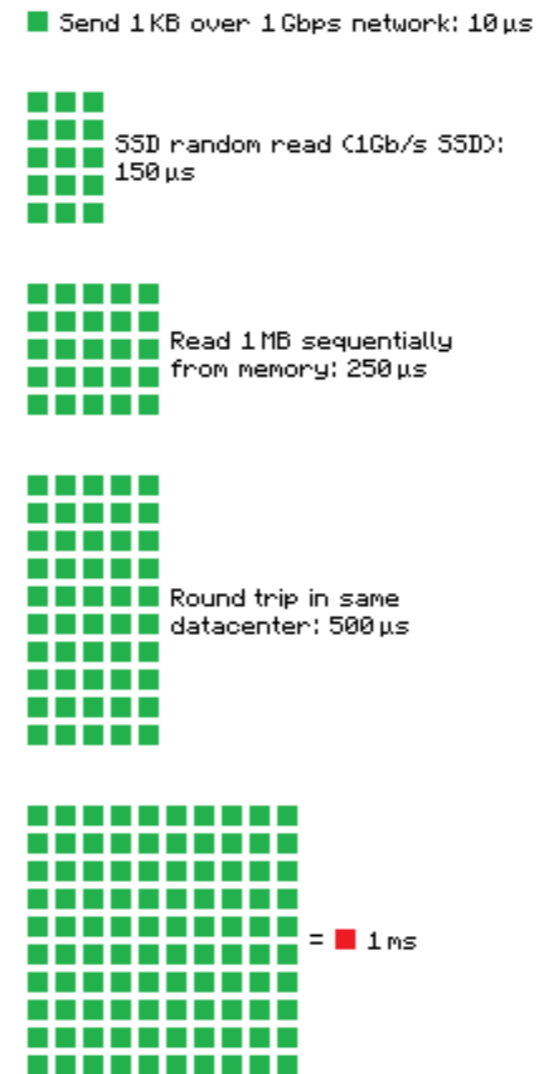
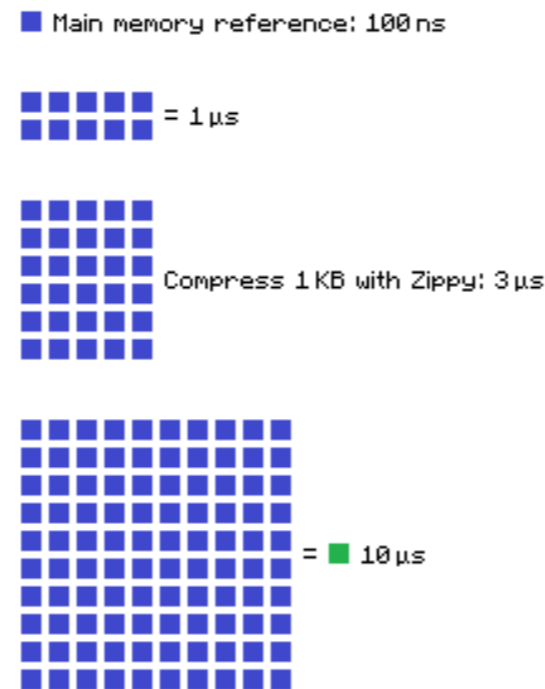
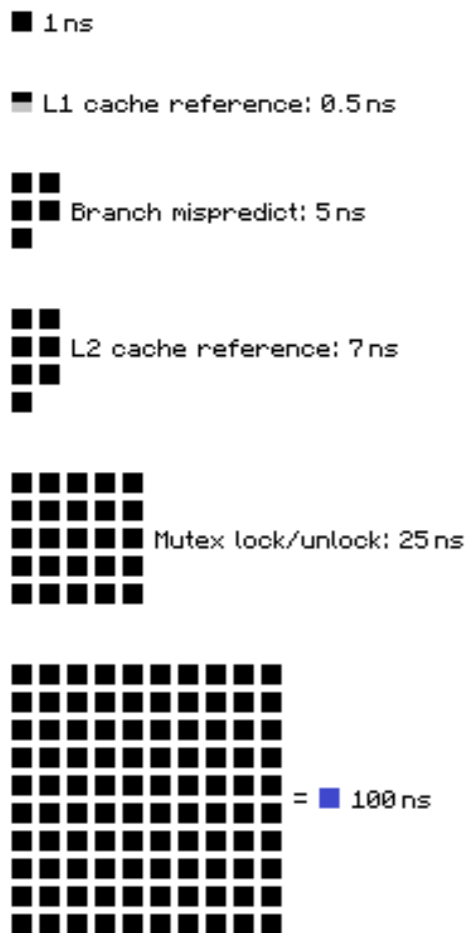
横跨很多不同的物理设备



为什么操作系统难学？

横跨很多不同的时间尺度

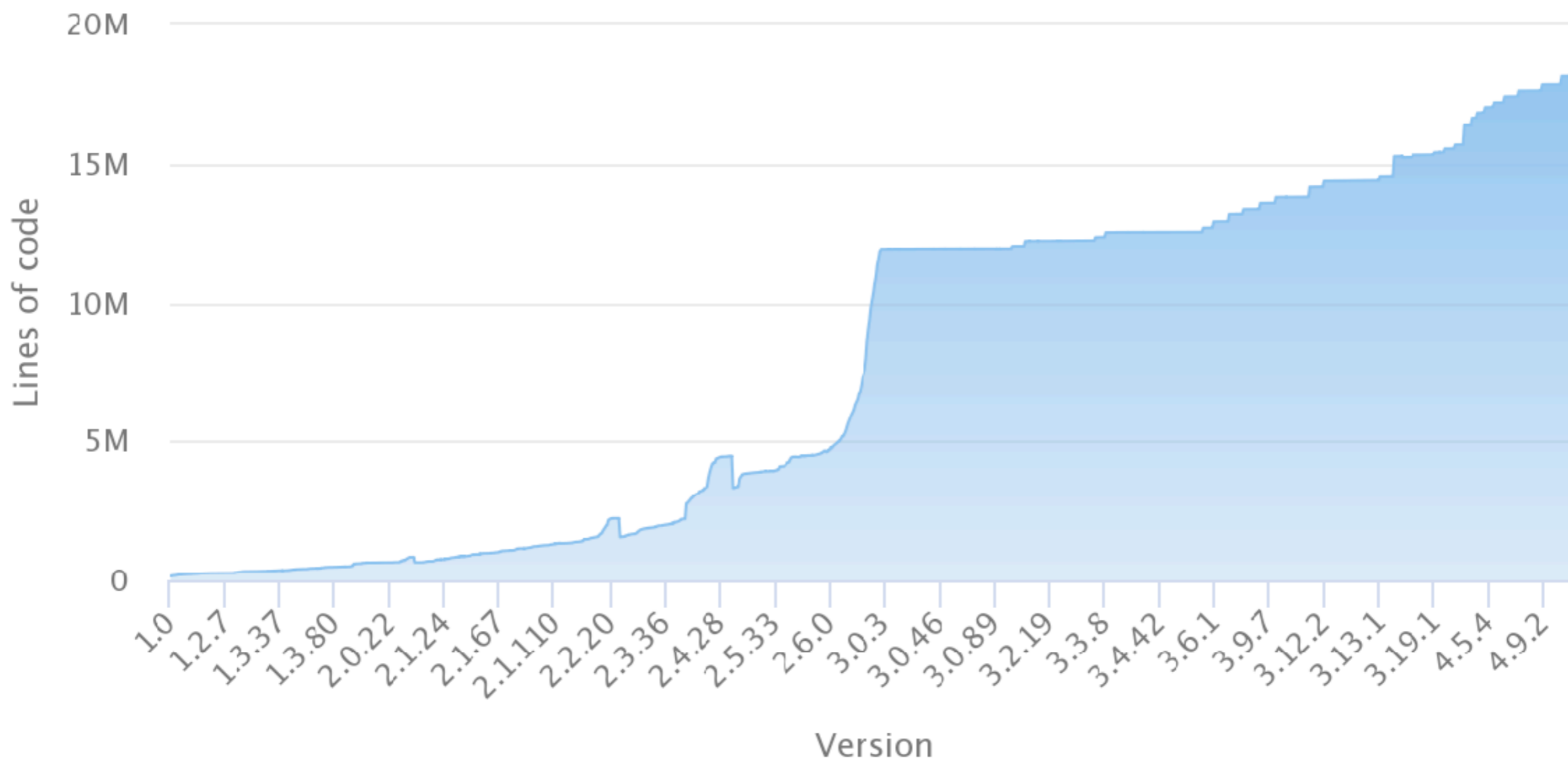
Latency Numbers Every Programmer Should Know



Source: <https://gist.github.com/2841832>

为什么操作系统难学？

规模和复杂度急剧增加



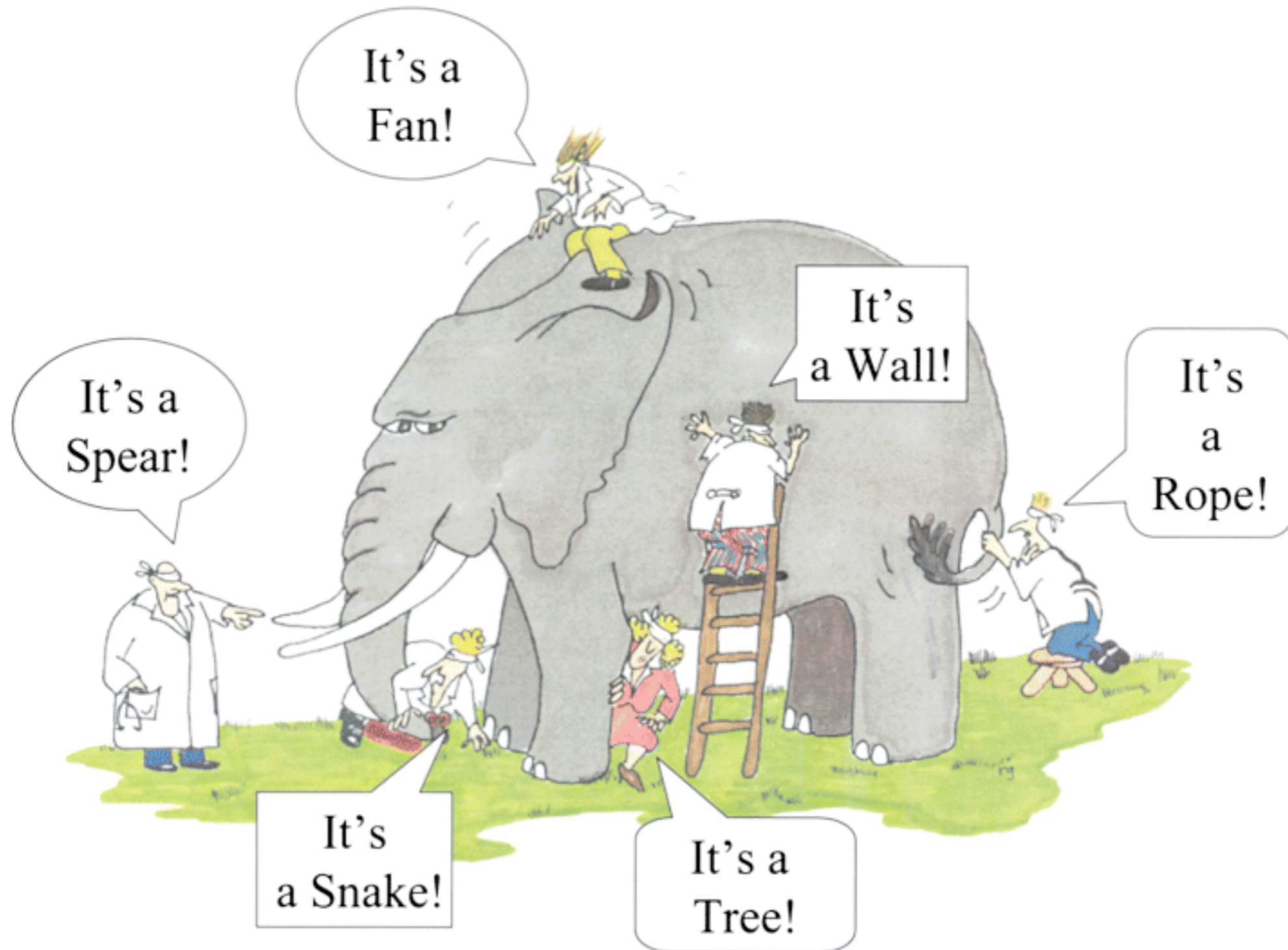
Lines of Code per Linux Kernel

为什么操作系统难学？

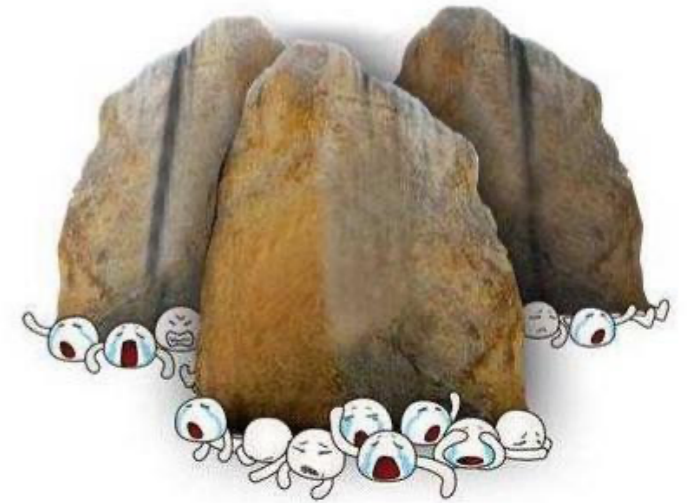
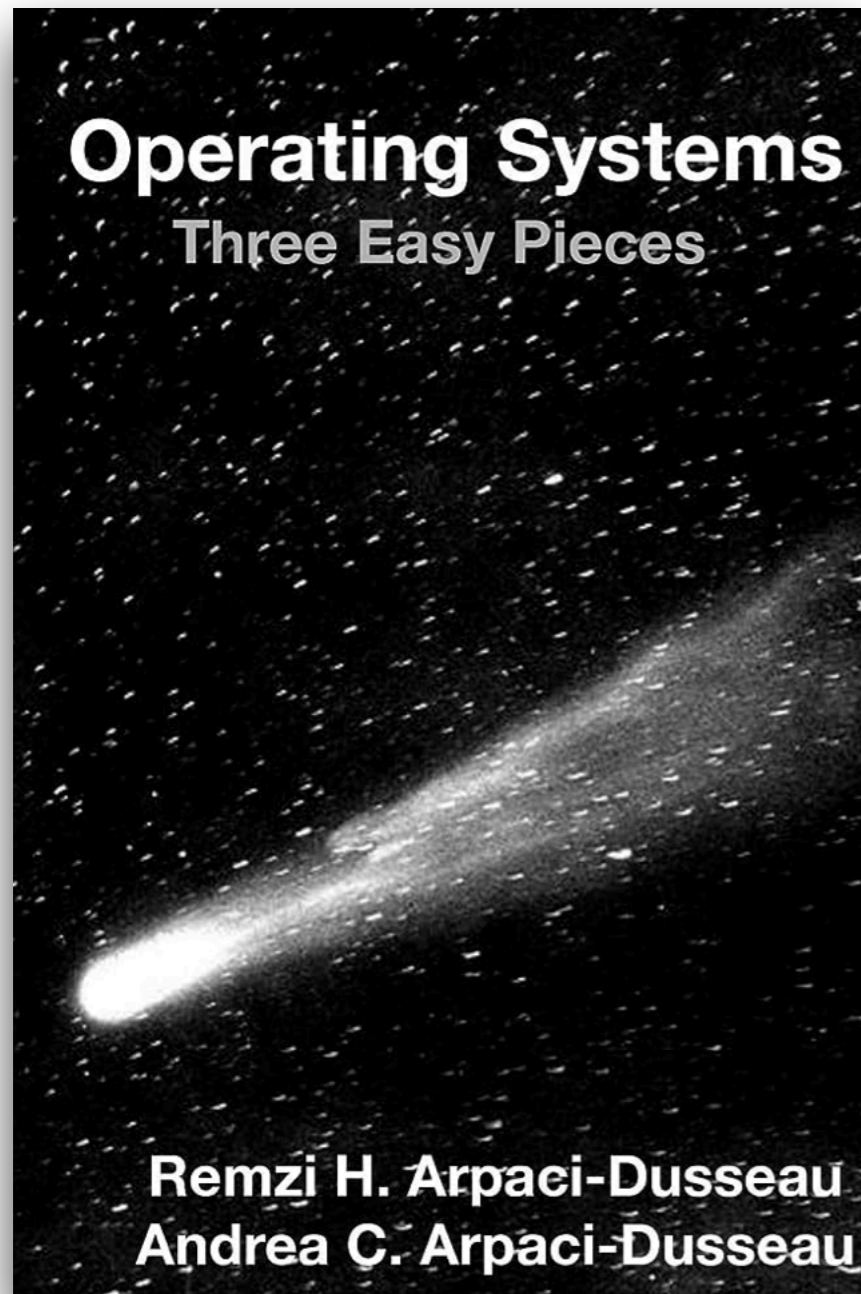
规模和复杂度急剧增加

- 更智能的硬件设备
- 更高的可靠性和安全性需求
- 更快的性能需求
- 更低的能耗需求
- 遗产系统和兼容性

如何理解如此复杂的系统？



教材



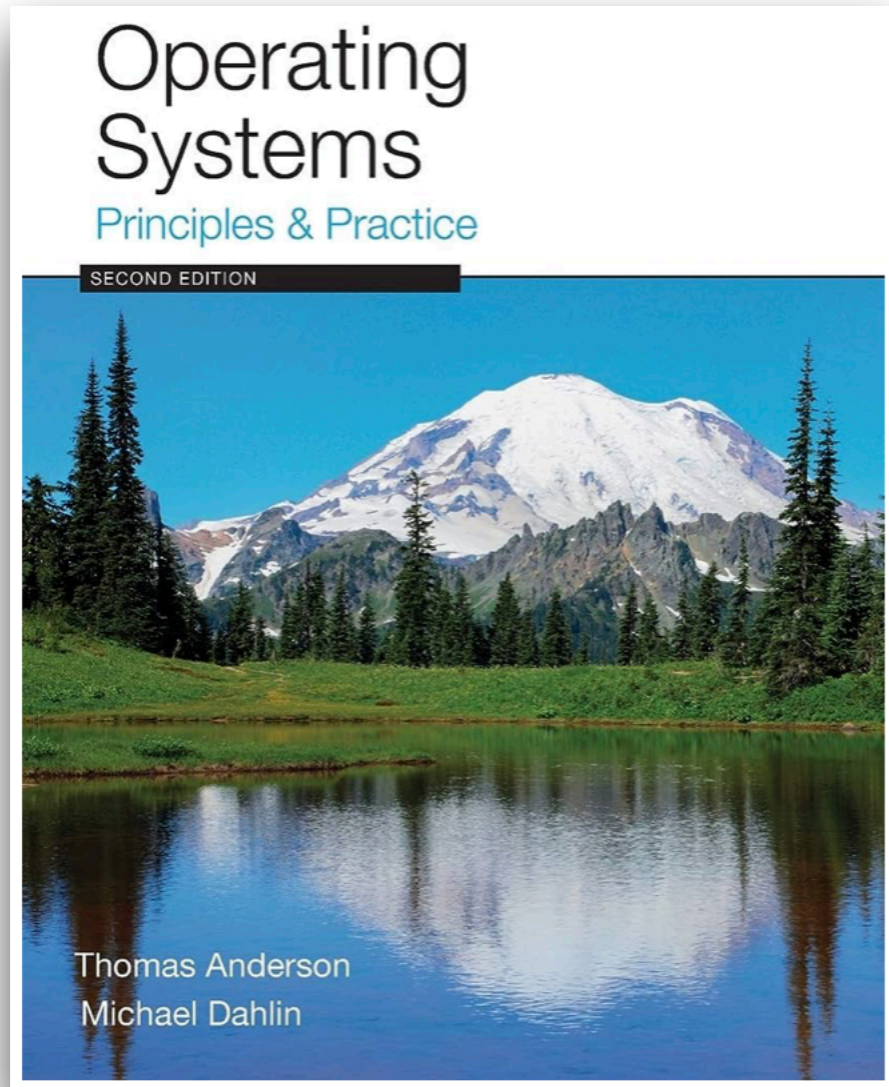
Operating Systems: Three Easy Pieces

Remzi H. Arpaci-Dusseau & Andrea C. Arpaci-Dusseau

<https://pages.cs.wisc.edu/~remzi/OSTEP/>

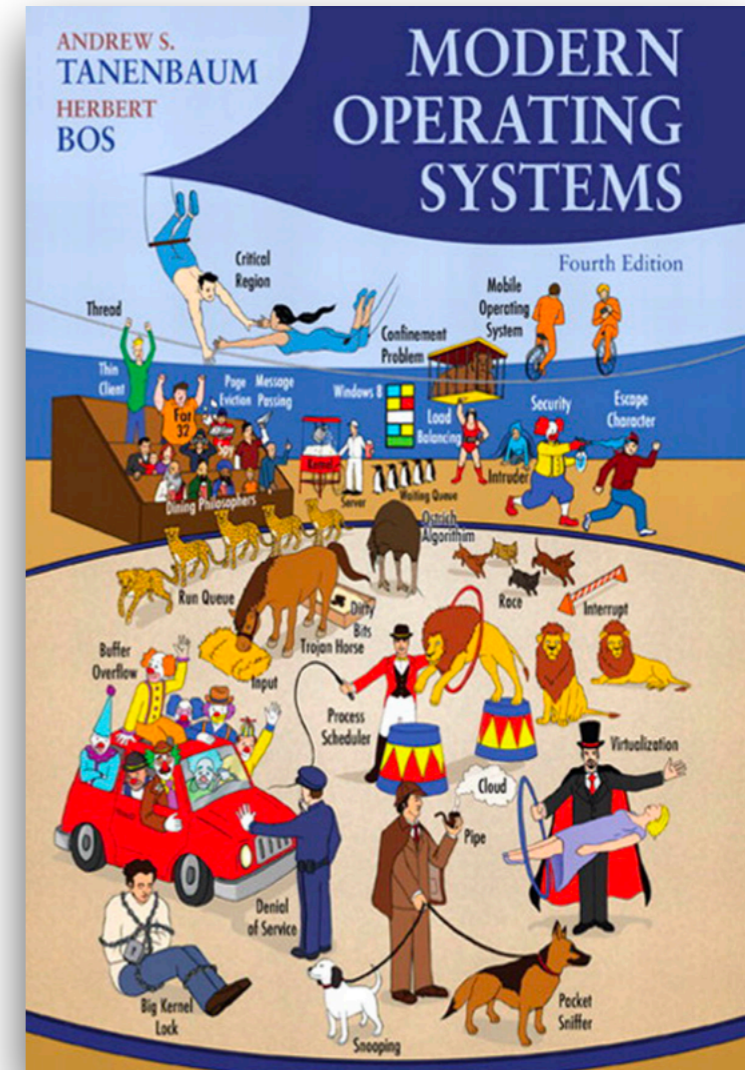
《操作系统导论》 人民邮电出版社

参考资料



Operating Systems: Principles & Practice

Thomas Anderson & Mike Dahlin

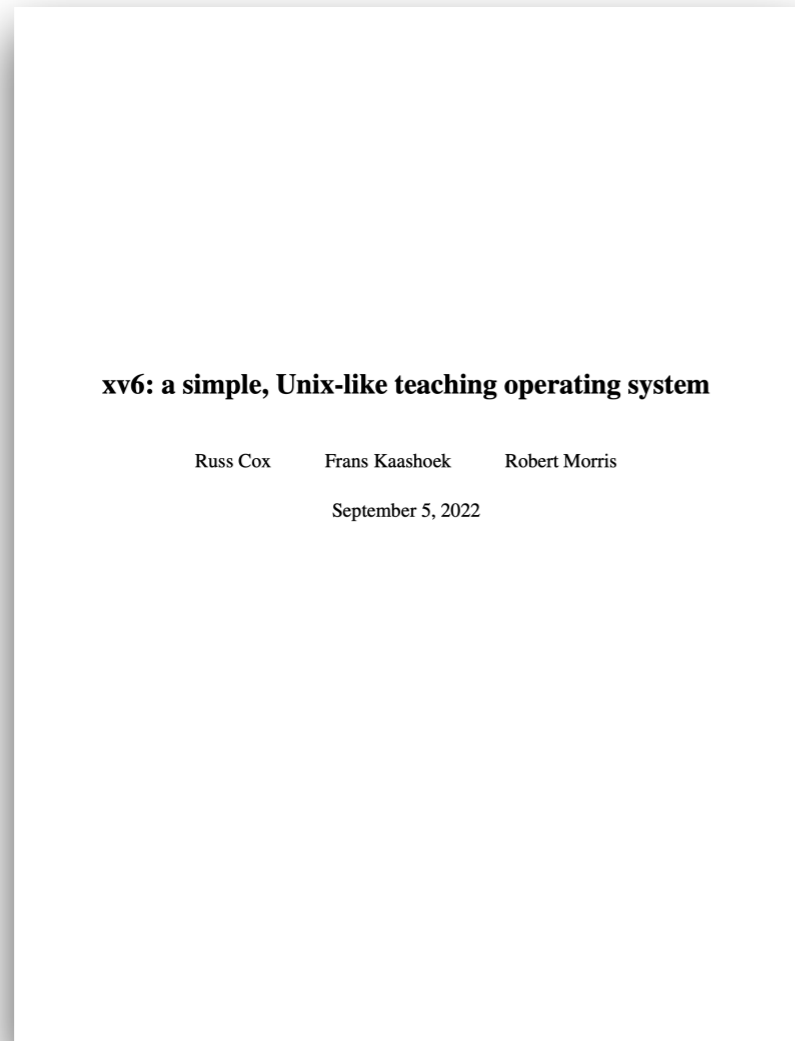


Modern Operating Systems

Andrew Tanenbaum & Herbert Bos

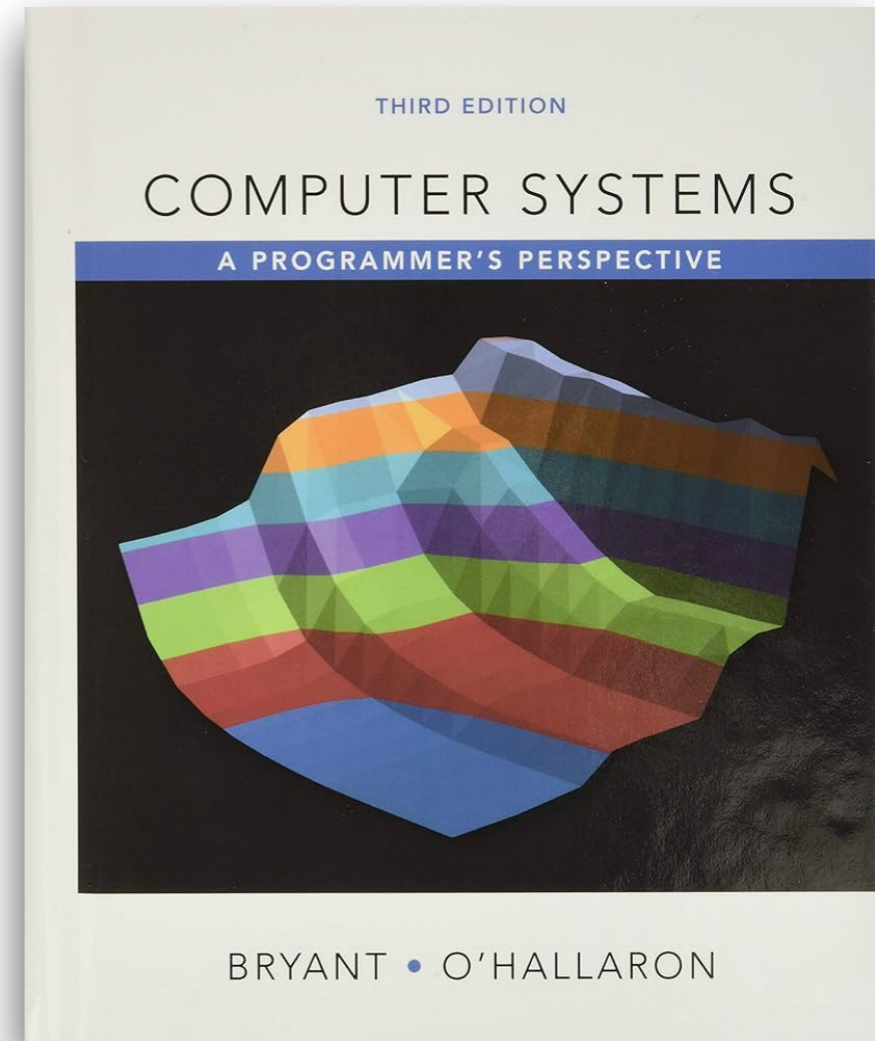
《现代操作系统》机械工业出版社

参考资料



xv6: a simple, Unix-like teaching operating system

Russ Cox, Frans Kaashoek & Robert Morris



Computer Systems: A Programmer's Perspective

Randal Bryant & David O'Hallaron

《深入理解计算机系统》机械工业出版社

参考资料



Large Language Models

大胆提问、谨慎求证

课程内容安排

- 操作系统概述
- 虚拟化 (Virtualization): 处理器管理、内存管理
- 并发 (Concurrency): 同步和死锁
- 持久化 (Persistence): 文件系统、设备管理

课程实验

- 总共 4 次课程实验 (Lab)
 - 基于 Linux (利用系统提供的 APIs 实现某种能力)
 - 每次实验需提交
 - 源代码: Online Judge 评分 + 部分人工检查
 - 实验报告 (1~2页): 人工阅读评分
- 实验截止时间
 - Soft Deadline: 在此之前完成提交将获得一定加分
 - Hard Deadline (本学期末): 所有实验最终提交 (更新) 截止

学术诚信

独立完成实验是对自己最好的训练

- 禁止抄袭别人的代码（包括来自互联网 / 学长学姐 / AI 生成），也不要将实验代码上传到互联网
 - 我们会有一些检测措施
 - 对 copy-er & copy-ee 的 minimum penalty 是 -100%
- 可以互相讨论遇到的问题 and 思路、以及使用 LLM 来帮助澄清概念或解释 API 的使用方式
 - 在实验报告中 acknowledge 相关参与方的贡献

成绩构成

- 课程实验: 40%
- 期中考试 (闭卷) + 平时成绩 (bonus): 20%
- 期末考试 (闭卷): 40%

* 重修/免修不免考仍需要按时完成实验、并参加期中和期末考试

* 补考只更新期末考试成绩

Tips

- Do not panic. 毕竟这门课的名称是“操作系统导论”
- 课堂出勤
 - 考试范围以课程 slides 为主
 - 核心考点都会在课堂提及
 - 随堂练习 = 样卷
- 完成实验
 - 少量考试题目会和 Lab 内容相关
 - 仅提交部分实验结果 (an empty .c file) 也将获得一定的分数

Tips

- 正确看待 OJ 评分
 - 不要面向 OJ 编程
 - $\neq 100$ 时不要惊慌 (有可能是 OJ 有 bugs)
 - $= 100$ 时也不要得意 (有可能是我们放水了)