OS

操作系统

介绍

陈　渝
清华大学计算机系

内容摘要

- 课程概述
- 什么是操作系统?
- 为什么学习操作系统?
- 如何学习操作系统?
- 操作系统实例
- 操作系统的演变
- 操作系统结构

课程概述

- 基本概念及原理
  - 操作系统介绍
  - 中断及系统调用
  - 内存管理
  - 进程及线程
  - 调度
  - 同步
  - 文件系统
  - I/O 子系统
- 练习
  - 在uCore 操作系统上做实验
  - https://github.com/chyyuu/mooc_os_lab
- 延伸
  - 讨论一些相关的话题或故事

OS课程概述

- 操作系统实验
  - 实验0: 准备
  - 实验1: 系统启动及中断
  - 实验2: 物理内存管理
  - 实验3: 虚拟内存管理
  - 实验4: 内核线程管理
  - 实验5: 用户进程管理
  - 实验6: CPU 调度
  - 实验7: 同步与互斥
  - 实验8: 文件系统

- 预备知识:
  - 计算机结构原理 (Intel 80386+)
  - 数据结构
  - C 与汇编 程序设计

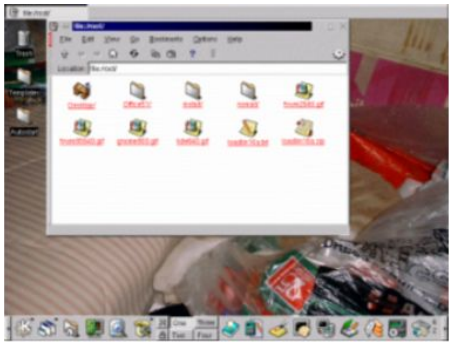
内容摘要

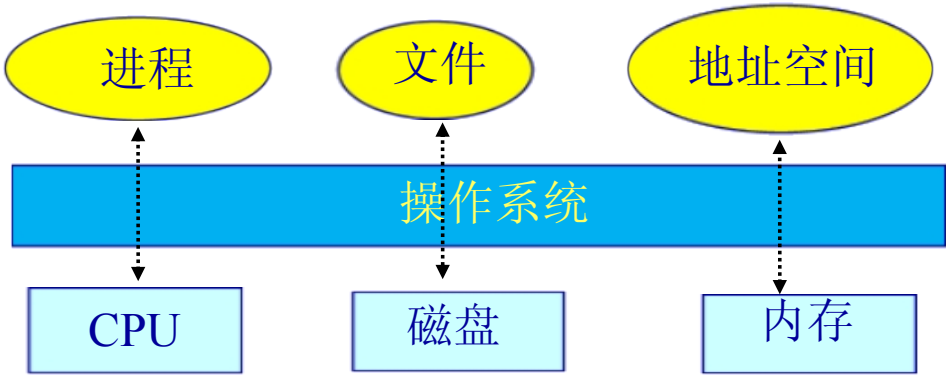
- 课程概述
- 什么是操作系统?
- 为什么学习操作系统?
- 如何学习操作系统?
- 操作系统实例
- 操作系统的演变
- 操作系统结构

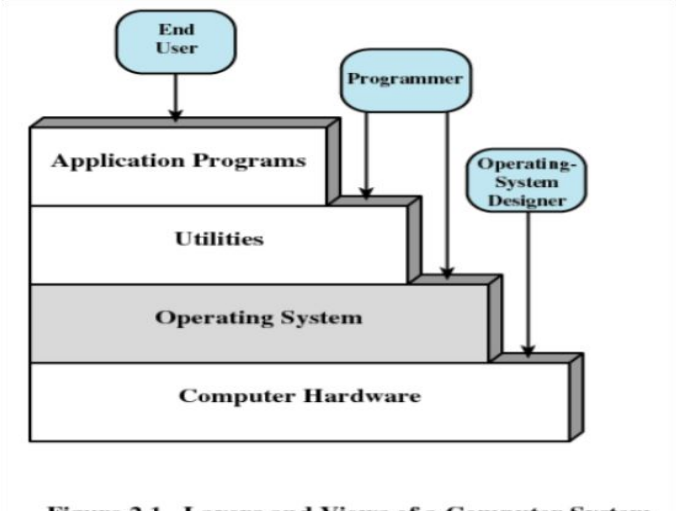
什么是操作系统?

操作系统定义

- **没有公认的精确定义**
- **操作系统是一个控制程序**
  - 一个系统软件
  - 控制程序执行以防止错误和计算机的不当使用
  - 执行用户程序和给用户程序提供各种服务
  - 使计算机系统方便使用
- **操作系统是一个资源分配器**
  - 应用程序与硬件之间的中间层
  - 管理各种计算机资源（包括硬件外设等）
  - 有效解决冲突请求并确保资源的公平使用
  - 提供高效的手段使用计算机硬件

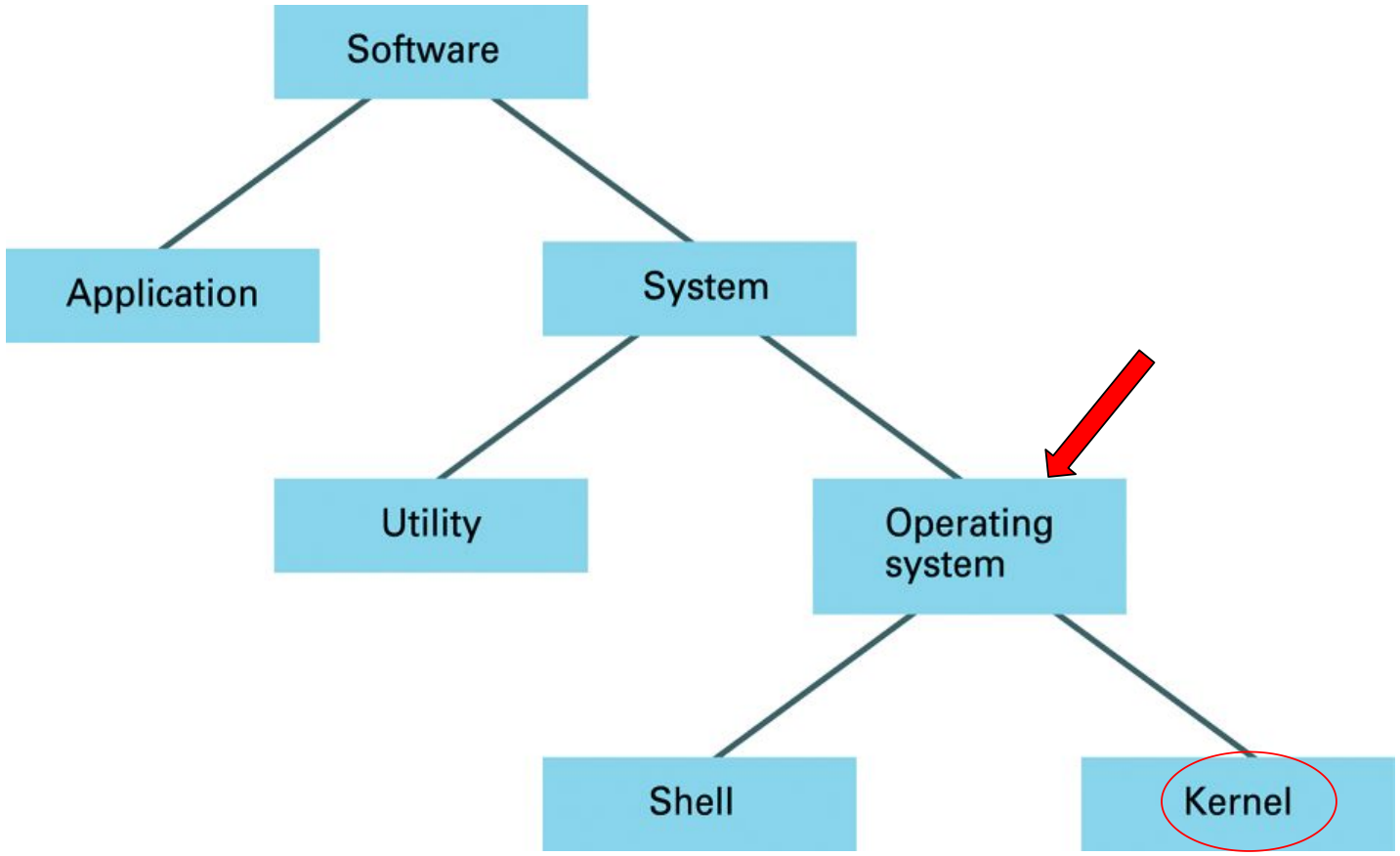






什么是操作系统?

– 在计算机系统内的软件体系结构中，操作系统所属的具体类别在哪里？

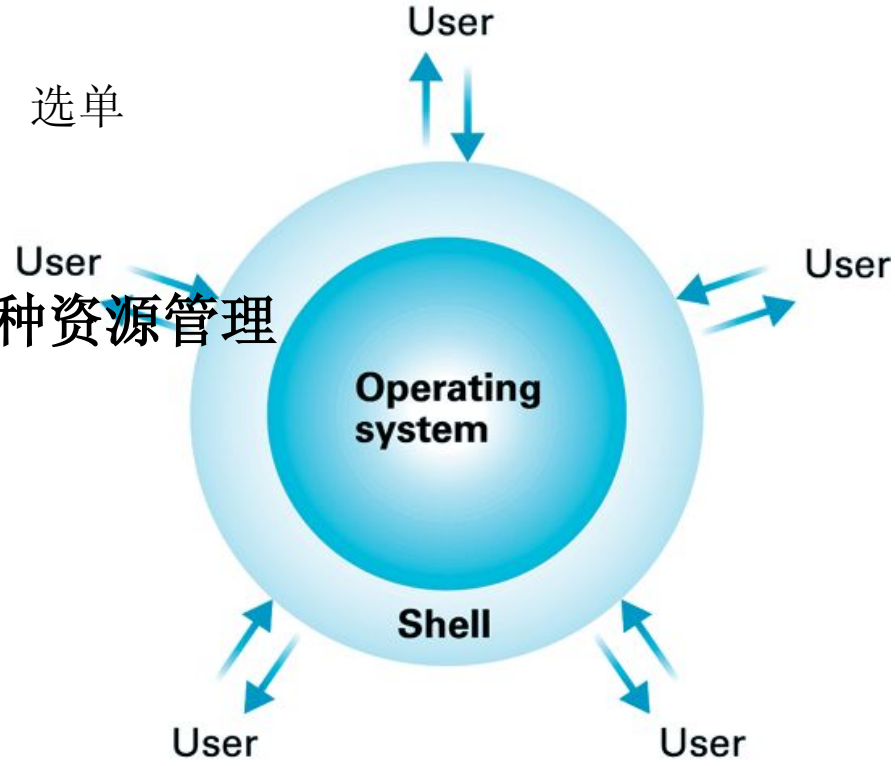


什么是操作系统?

- **Shell** -- 文本信息 ✹ **GUI**: 图形表示, 图标
- 通过键盘方向键, 命令, 鼠标, 轨迹球, 触摸板, 等操纵。
- 窗口管理 – 在屏幕上为应用程序分配空间, 管理空间, 等.
- **GUI**--桌面隐喻

  WIMP（视窗（Window）、图标（Icon）、选单（Menu）、指标（Pointer））

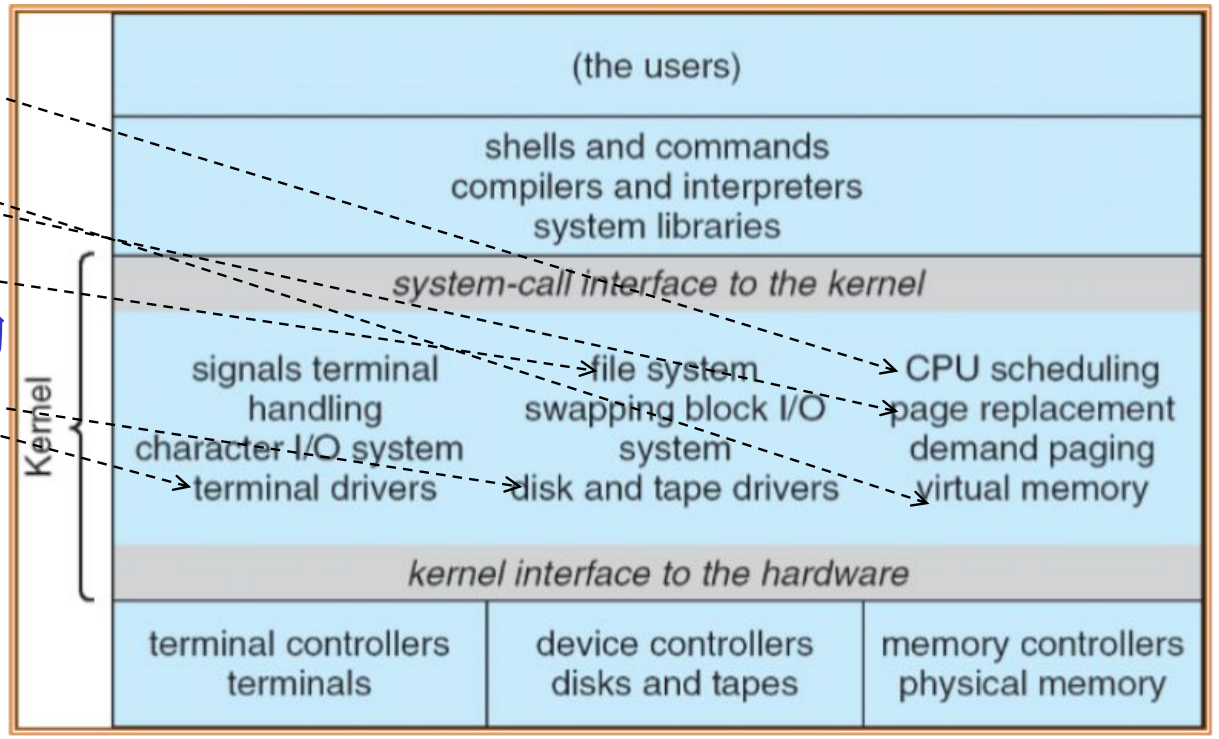
  直接操作和所见即所得
- **Kernel** -- 操作系统的内部 – 执行各种资源管理等功能，是这门课关注的重点



什么是操作系统?

Kernel – 操作系统内部 组件，包括:

- **CPU调度器**
- **物理内存管理**
- **虚拟内存管理**
- **文件系统管理**
- **中断处理与设备驱动**



uCore操作系统

什么是操作系统?

OS Kernel的特征:

- 并发
  - 计算机系统中同时存在多个运行的程序，需要OS管理和调度
- 共享
  - “同时”访问
  - 互斥共享
- 虚拟
  - 利用多道程序设计技术，让每个用户都觉得有一个计算机专门为他服务
- 异步
  - 程序的执行不是一贯到底，而是走走停停，向前推进的速度不可预知
  - 但只要运行环境相同，OS需要保证程序运行的结果也要相同

内容摘要

- 课程概述
- 什么是操作系统?
- 为什么学习操作系统?
- 如何学习操作系统?
- 操作系统实例
- 操作系统的演变
- 操作系统结构

为什么学习操作系统?

- 综合课程 - 结合许多不同的课程
  - 程序设计语言
  - 数据结构
  - 算法
  - 计算机体系结构

- 材料
  - 操作系统概念和原理, 源代码
- 技能
  - 操作系统的设计和实现

为什么学习操作系统?

- 我使用的操作系统运行的很好, 我怀疑我将来的工作不会涉及到写一个OS
  - 例如 Windows, Linux.
- 操作系统开发人员已经解决了所有的事情吗? 还有什么要做的?
- 作为一个本科生，为什么我要学习它?

写操作系统很酷!

掌握操作系统是一个挑战!

操作系统很有用!

我要参与系统软件开发

我想了解操作系统到底是如何工作的?

为什么学习操作系统?

操作系统: 计算机科学研究的基石之一

- 计算机系统的基本组成部分
- 由硬件的发展和应用需求所驱动
- 学术和工业的持续推进

为什么学习操作系统?

哪里在做OS研究

- 顶尖大学的计算机科学部门
- 计算机产业
  - 旧时: Xerox (PARC), IBM, DEC (SRC), Bell Labs
  - 现代: Microsoft, Google, Yahoo, IBM, HP, Sun, Intel, VMware, Amazon, …
- 研究协会
  - ACM SIGOPS
  - USENIX

为什么学习操作系统?

操作系统研究的顶级会议

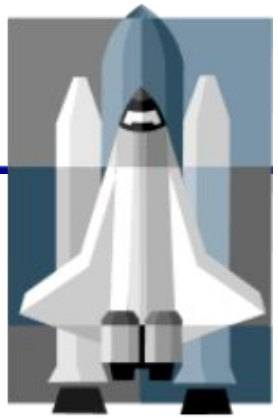
- ACM 操作系统原理研讨会 (SOSP)
  - ACM SIGOPS
  - 每两年 (奇数: 1967-)
  - ~20 论文
- USENIX 操作系统设计和实现研讨会 (OSDI)
  - USENIX
  - 每两年 (偶数: 1994-)
  - ~20 论文

为什么学习操作系统?

最具影响力的操作系统论文

- SIGOPS Hall-of-Fame Awards
  - 论文必须发表在同行评议的文献中至少十年
  - 到目前为止有三十多篇论文获奖
- 假如你想做操作系统研究
  - 需要阅读和理解这些论文
  - http://www.sigops.org/award-hof.html

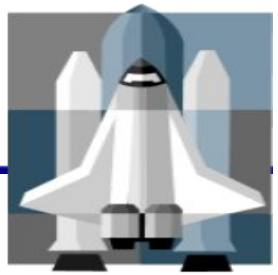
为什么学习操作系统?



掌握操作系统具有挑战性

- 操作系统很大
  - Windows XP 有4500万行
- 操作系统管理并发
  - 并发导致有趣的编程挑战
- 操作系统代码管理原始硬件
  - 时间依赖行为, 非法行为, 硬件故障
- 操作系统代码必须是高效的, 低耗CPU、内存、磁盘的
- 操作系统出错，就意味着机器出错
  - 操作系统必须比用户程序拥有更高的稳定性
- 操作系统是系统安全的基础

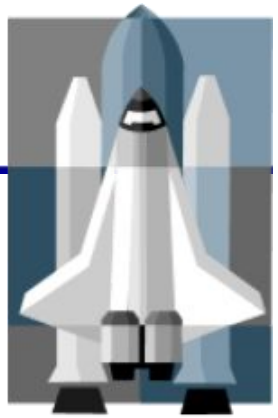
为什么学习操作系统?



掌握操作系统具有挑战性

- 操作系统并不仅仅关于并发性和琐碎的调度算法
  - 并发性是一小部分
    - 内核里不存在管程和哲学家问题
    - 内核中的锁问题需要太多的背景知识
  - 磁盘调度大多是不相干的 (SCSI 已经做了这些)
  - 进程调度是个比较小话题

为什么学习操作系统?



掌握操作系统具有挑战性

- 操作系统是关于:
  - 权衡
    - 时间与空间
    - 性能与可预测性
    - 公平与性能(哪种设计能工作？为什么?)
  - 硬件
    - 如何让中断、异常、上下文切换真正有效?
    - TLB是如何工作的？这对页表又意味着什么？
    - 如果你不展示任何汇编代码，那么你就不是教操作系统的!

内容摘要

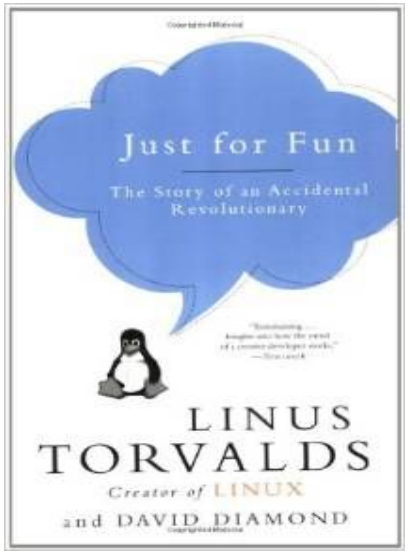
- 课程概述
- 什么是操作系统?
- 为什么学习操作系统?
- 如何学习操作系统?
- 操作系统实例
- 操作系统的演变
- 操作系统结构

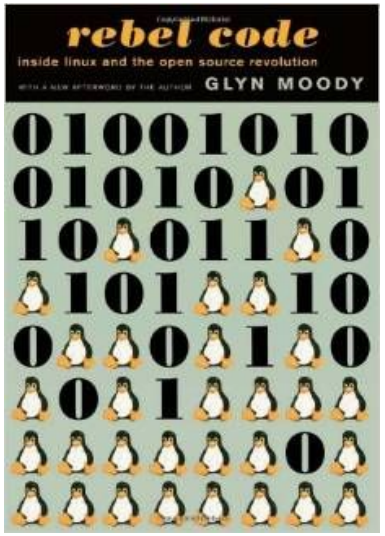
如何学习操作系统?

"我听到的我会忘记,
我看到的我能记住,
只有我做过的我才能理解."
-- 中国谚语

"天才是1% 的灵感加上
99% 的汗水"
-- Thomas Edison

“困难,最好的和**最有趣**的三年课程!”



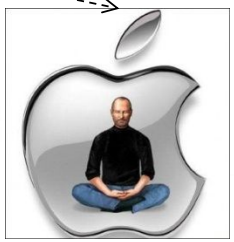


OS 概要

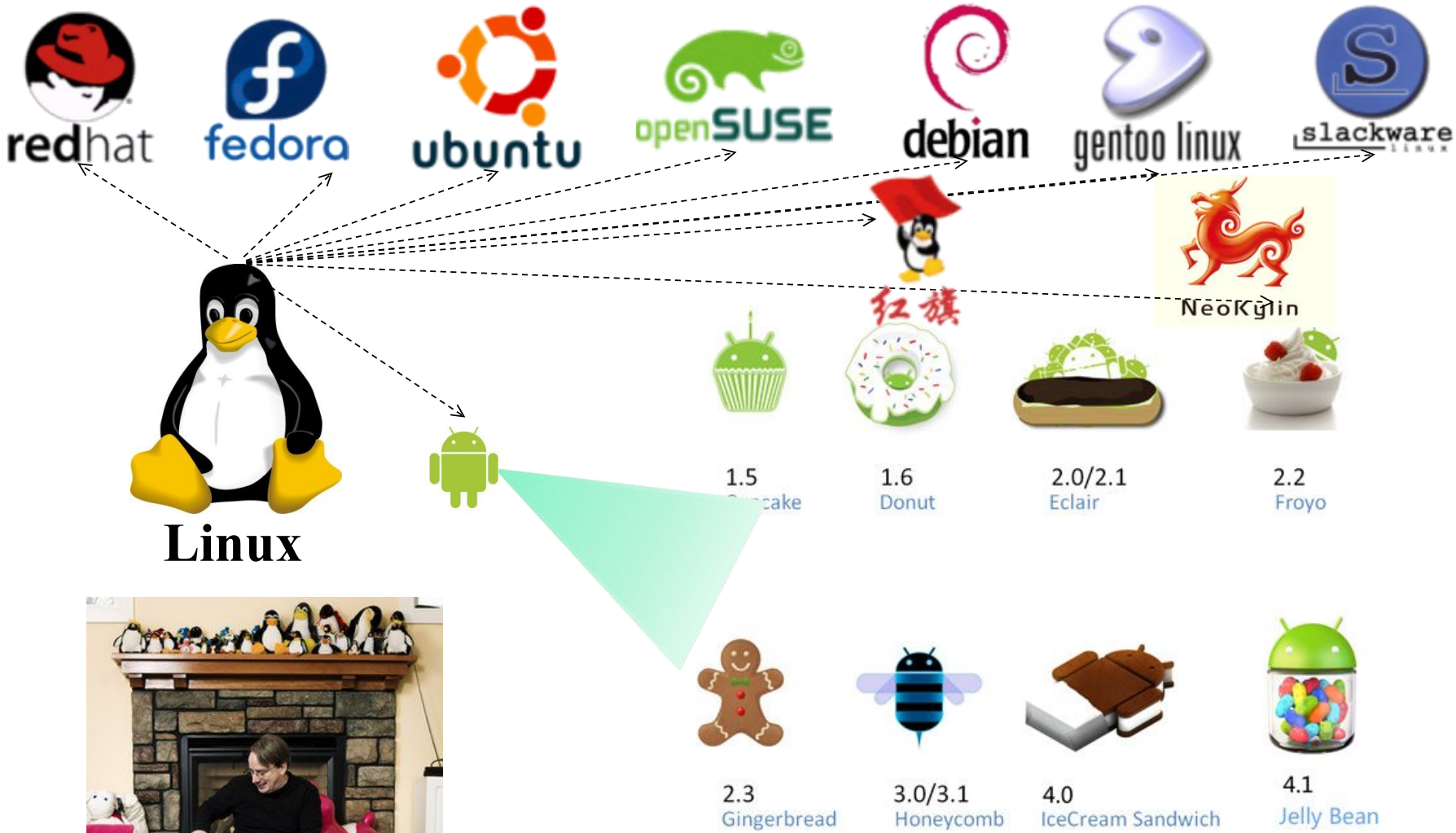
- 课程概述
- 什么是操作系统?
- 为什么学习操作系统?
- 如何学习操作系统?
- 操作系统实例
- 操作系统的演变
- 操作系统结构

UNIX家族

UNIX BSD (伯克利软件发行版)



Linux家族


redhat
fedora
ubuntu
openSUSE
debian
gentoo linux
slackware
NeoKylin
红旗
Linux
1.5
1.6
Donut
2.0/2.1
Eclair
2.2
Froyo
2.3
Gingerbread
3.0/3.1
Honeycomb
4.0
IceCream Sandwich
4.1
Jelly Bean

Windows家族

微软从DEC聘请 Dave Cutler 做 Windows NT 主要设计师

Windows 1.0 | Windows 3.1 | Windows 95 | Windows 98 | Windows me

Windows 2000 | Windows XP | Windows Vista | Windows 7 | Windows 8

内容摘要

- 课程概述
- 什么是操作系统?
- 为什么学习操作系统?
- 如何学习操作系统?
- 操作系统实例
- 操作系统的演变
- 操作系统结构

操作系统的演变

- 操作系统为什么改变?
  - 主要功能: 硬件抽象和协调管理
  - 原则: 设计随着各种相关技术的改变而做出一定的改变
  - 在过去二十年底层技术有极大的改变 !!

- 从 1981到 2012计算机系统的对比

Vital statistic	1981 IBM personal computer	2001 Dell OptiPlex GX150	2012 Dell XPS 8300
Price	$3045	$1447	$1090
CPU	4.77-MHz 8088	933-MHz Pentium III	3.4GHz Intel Core i7-2600
MIPS	0.33-1 MIPS	1,354 MIPS at 500 MHz	76,383 MIPS at 3.2 GHz
RAM	64KB	128MB	8GB DDR3 SDRAM at 1333MHz
Storage	160KB floppy drive	20GB hard drive, CD-RW and 1.44MB floppy drives	1TB - 7200RPM, SATA 3.0Gb/s

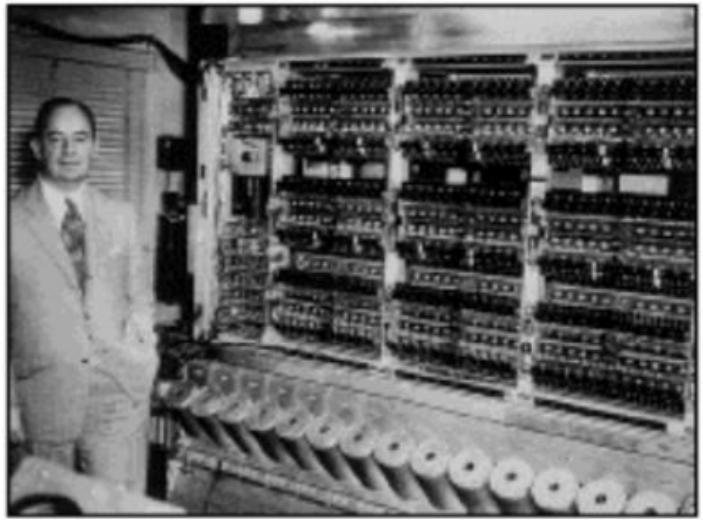
操作系统的演变

- 单用户系统
- 批处理系统
- 多程序系统
- 分时
- 个人计算机: 每个用户一个系统
- 分布式计算:每个用户多个系统

单用户系统 ('45-'55)

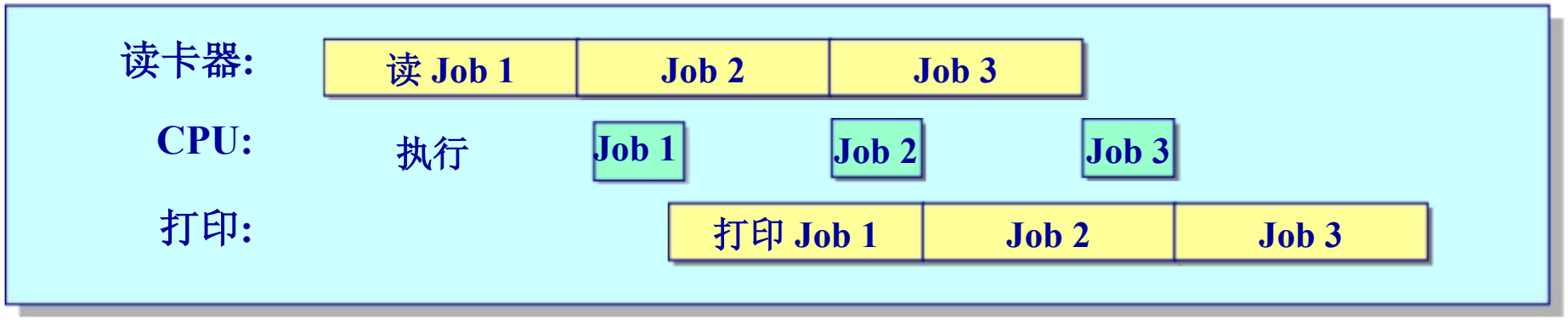
- 操作系统 = *装载器* + *通用子程序库*
- 问题: 昂贵组件的低利用率

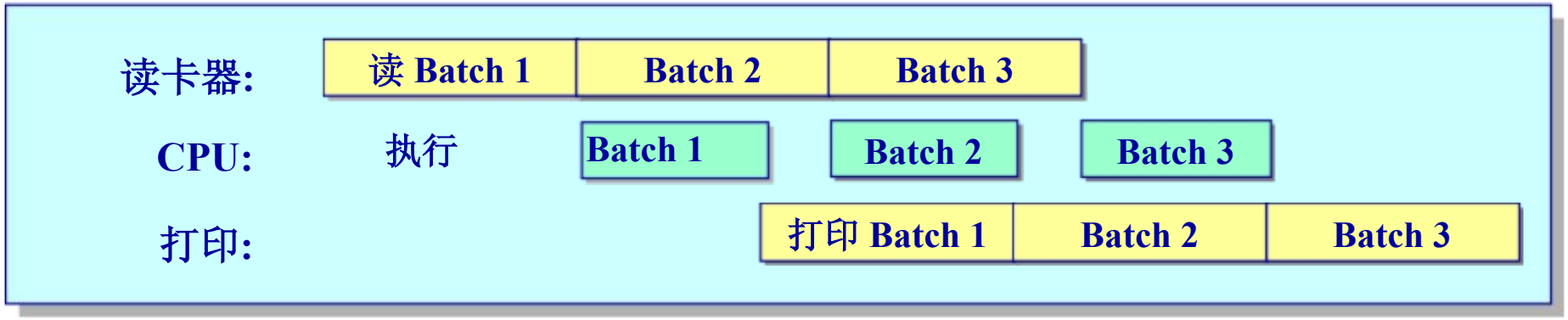
$$\frac{\textit{执行时间}}{\textit{执行时间} + \textit{读卡时间}} = \%\ \textit{利用率}$$



成批/离线处理 (‘55-‘65)

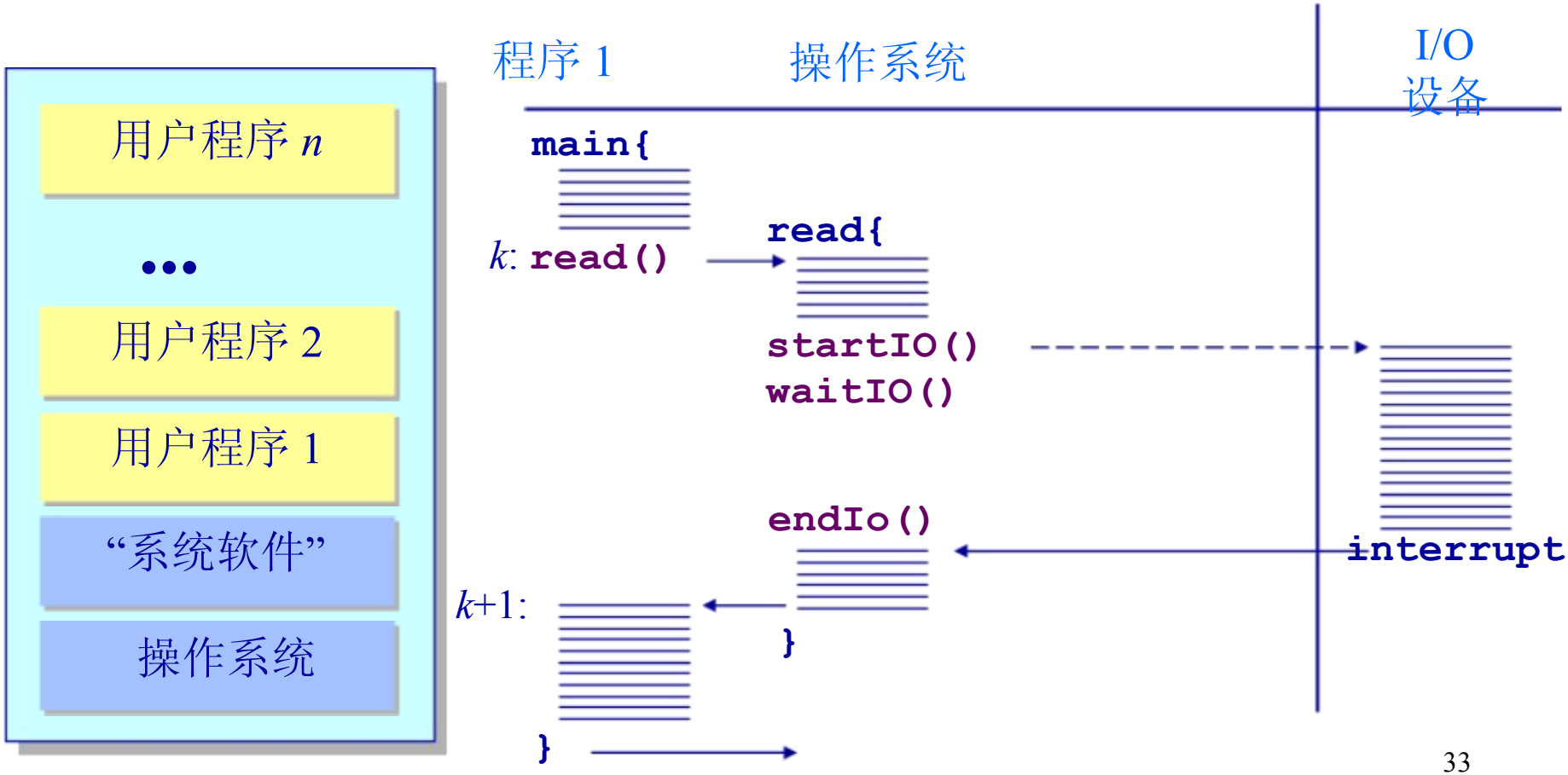
- 顺序执行与批处理





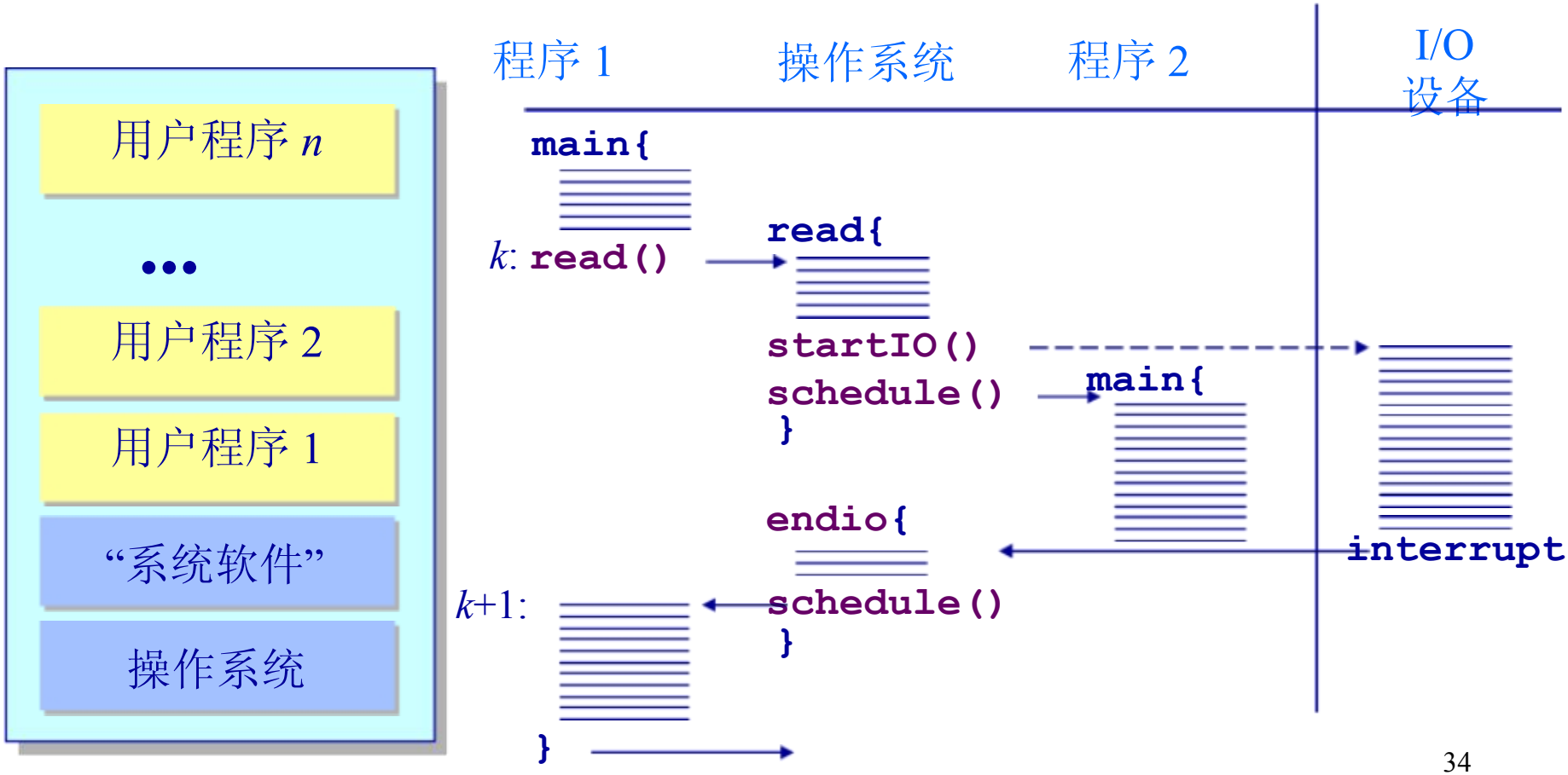
多程序 (‘65-‘80)

- 保持多个工作在内存中并且在各工作间复用CPU



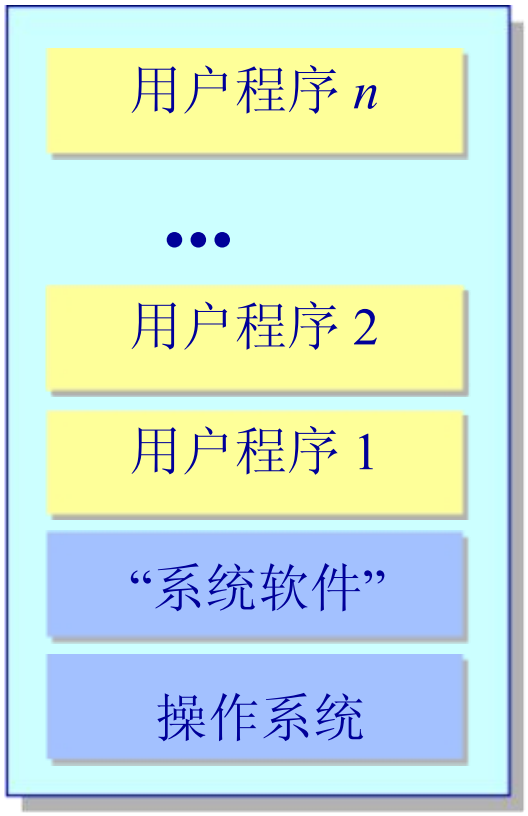
多程序 (‘65-‘80)

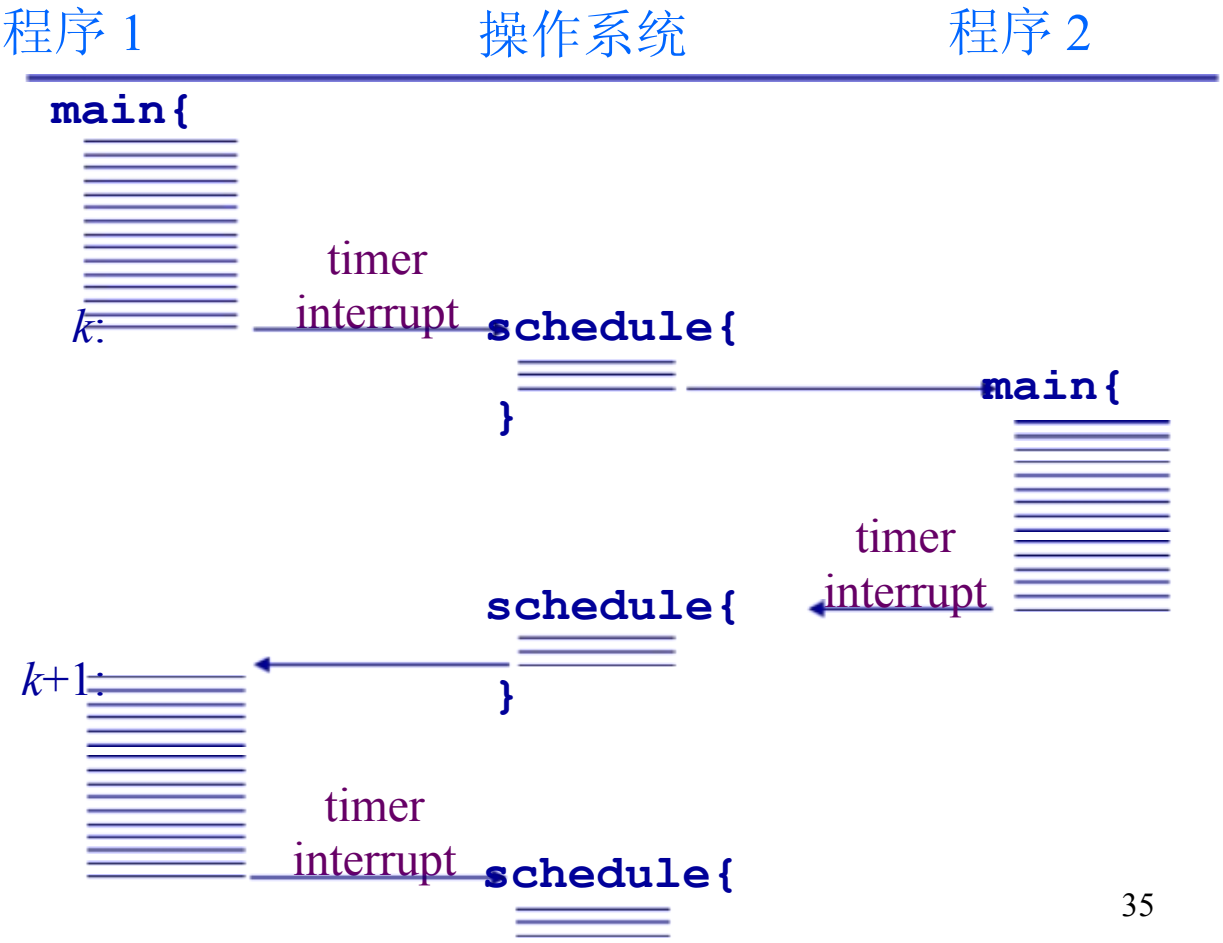
- 保持多个工作在内存中并且在各工作间复用CPU



分时 (‘70-)

- 定时中断用于工作对CPU的复用





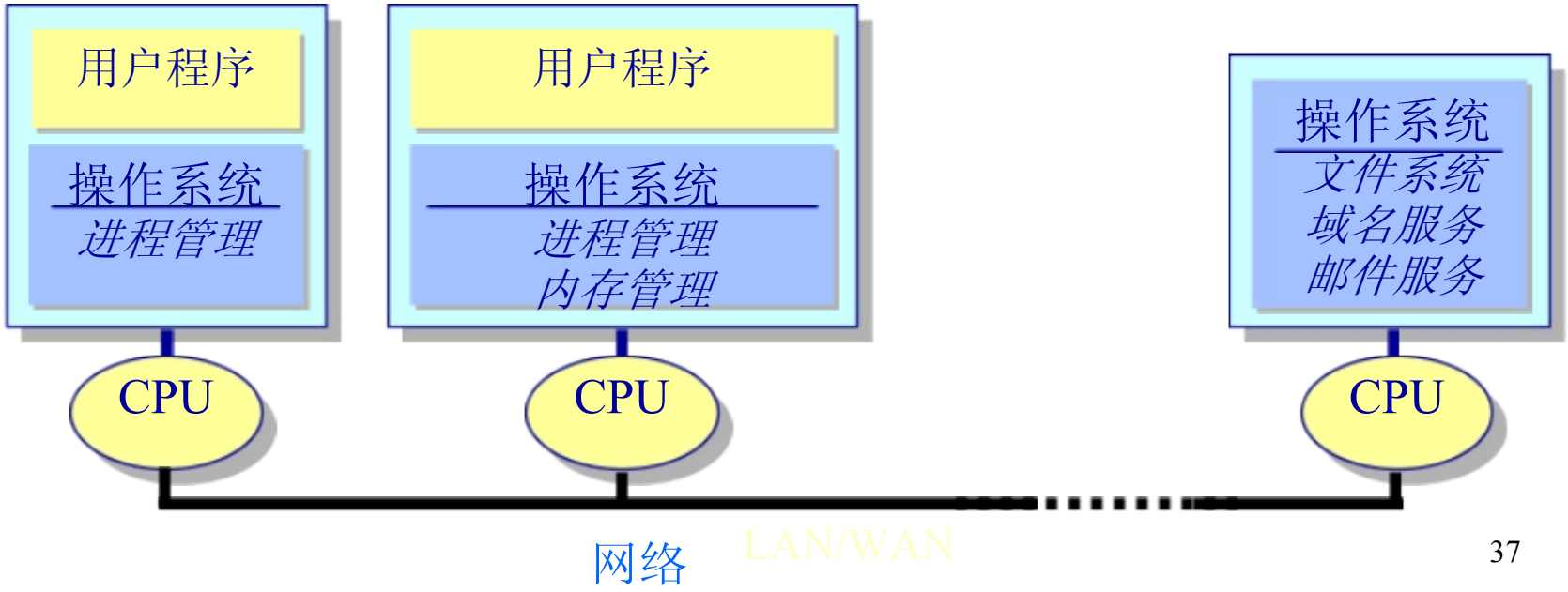
个人电脑操作系统

- 个人电脑系统
  - 单用户
  - 利用率已不再是关注点
  - 重点是用户界面和多媒体功能
  - 很多老的服务和功能不存在

- 演变
  - 最初: 操作系统作为一个简单的服务提供者 (简单库)
  - 现在：支持协调和沟通的多应用系统
  - 越来越多的安全问题 (如，电子商务、医疗记录)

分布式操作系统

- 网络支持成为一个重要的功能
- 通常支持分布式服务
  - 跨多系统的数据共享和协调
- 可能使用多个处理器
  - 松、紧耦合系统
- 高可用性与可靠性的要求



操作系统的演变

Future OS

普适计算（Pervasive computing），移动计算，云计算，大数据处理，许多联网设备为许多人提供个性化的服务

iOS/Android,...

Internet服务

Windows/Linux/BSD,...

网络计算（Internet computing）

AIX/HP-UX

个人机计算（Personal computing）

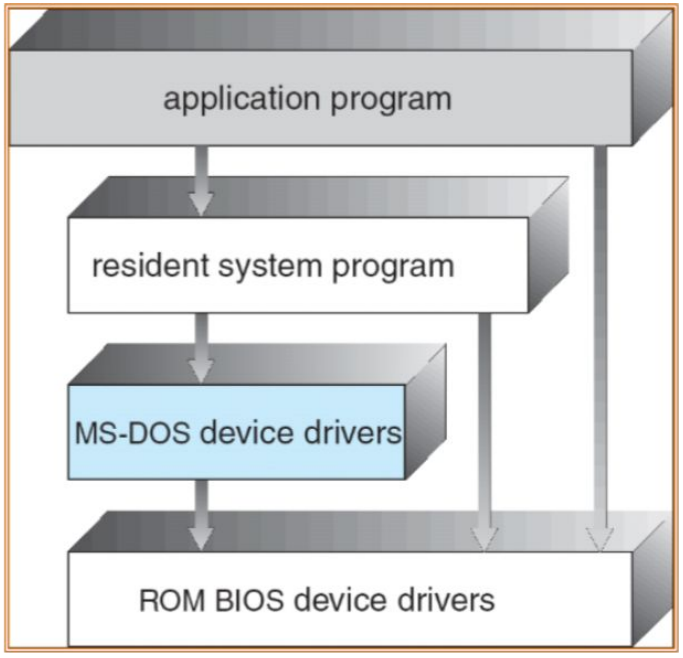
主机型计算（Mainframe computing）

内容摘要

- 课程概述
- 什么是操作系统?
- 为什么学习操作系统?
- 如何学习操作系统?
- 操作系统实例
- 操作系统的演变
- 操作系统结构

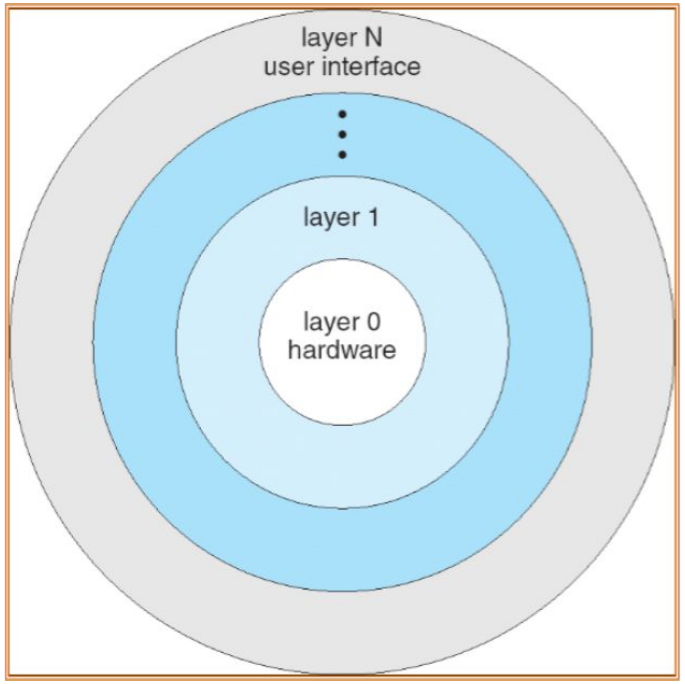
简单结构

- MS-DOS – 在最小的空间，设计用于提供大部分功能 (1981~1994)
  - 没有拆分为模块
  - 虽然 MS-DOS 在接口和功能水平没有很好地分离，主要用汇编编写



分层方法

- 操作系统分为很多层 (levels)
  - 每层建立在低层之上
  - 最底层 (layer 0), 是硬件
  - 最高层(layer N) 是用户界面
- 使用模块化, 每一层仅使用更低一层的功能（操作）和服务。

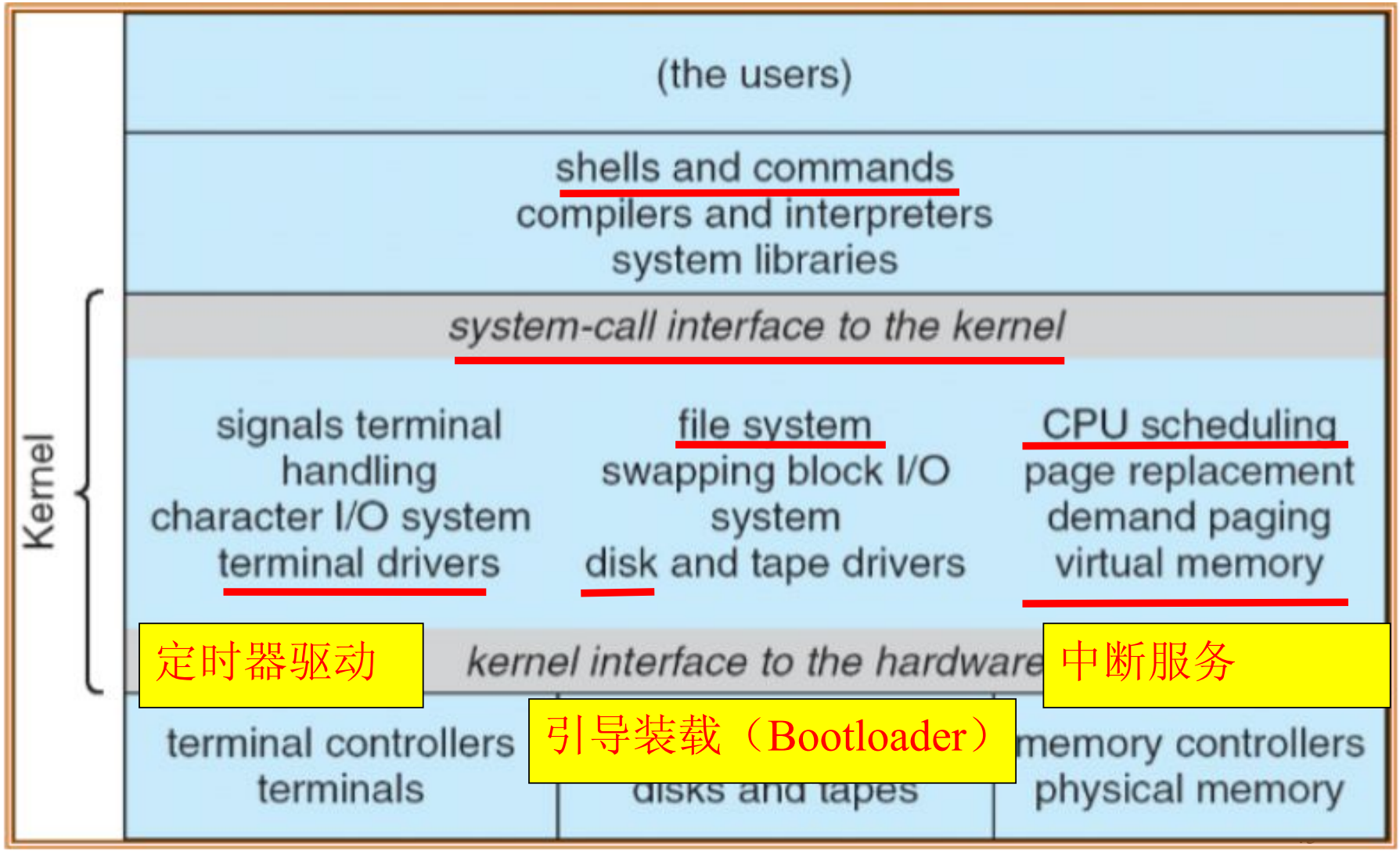


UNIX

- 1972由 Kenneth Thompson和Dennis Ritchie在贝尔实验室设计.
- 设计用于 UNIX 操作系统的编码例程.
- “高级”系统编程语言创建可移植操作系统的概念

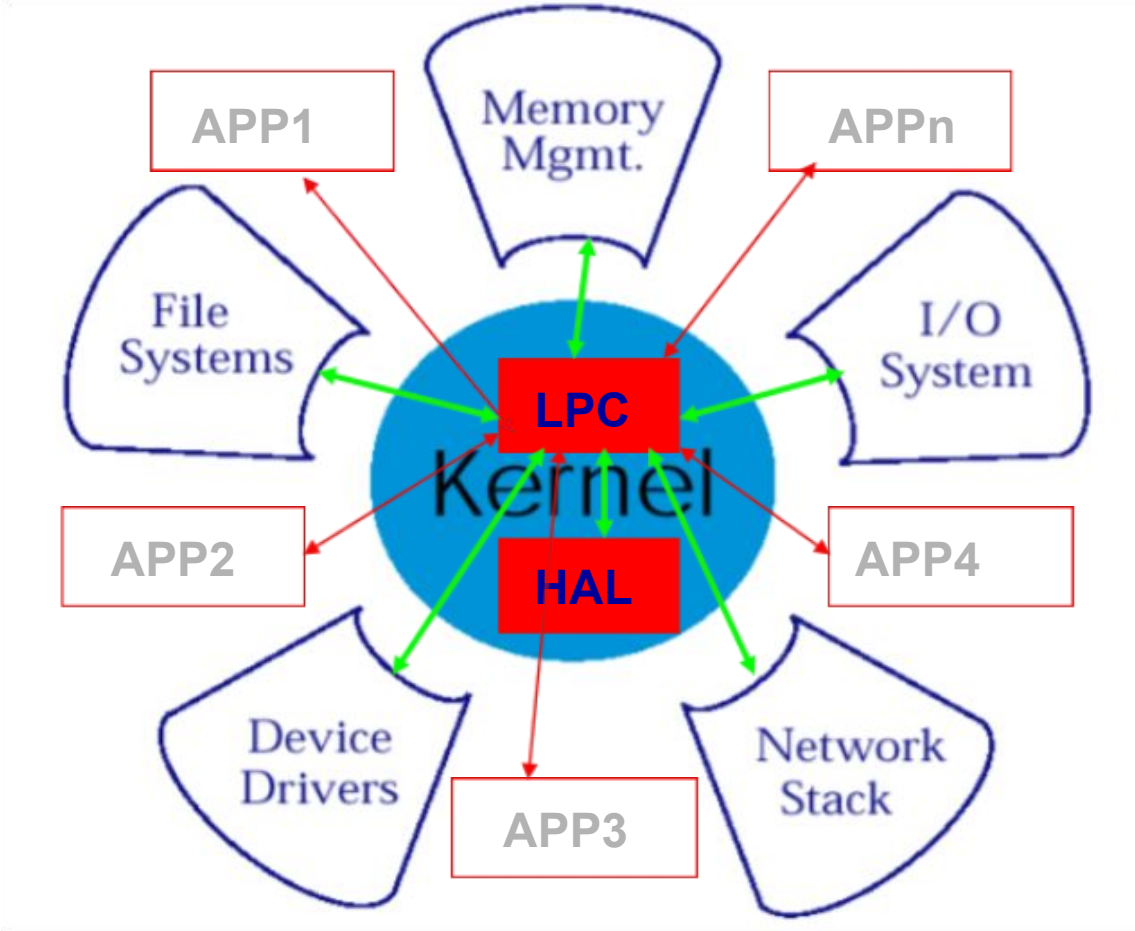
K. Thompson and D. Ritchie

uCore 操作系统实验



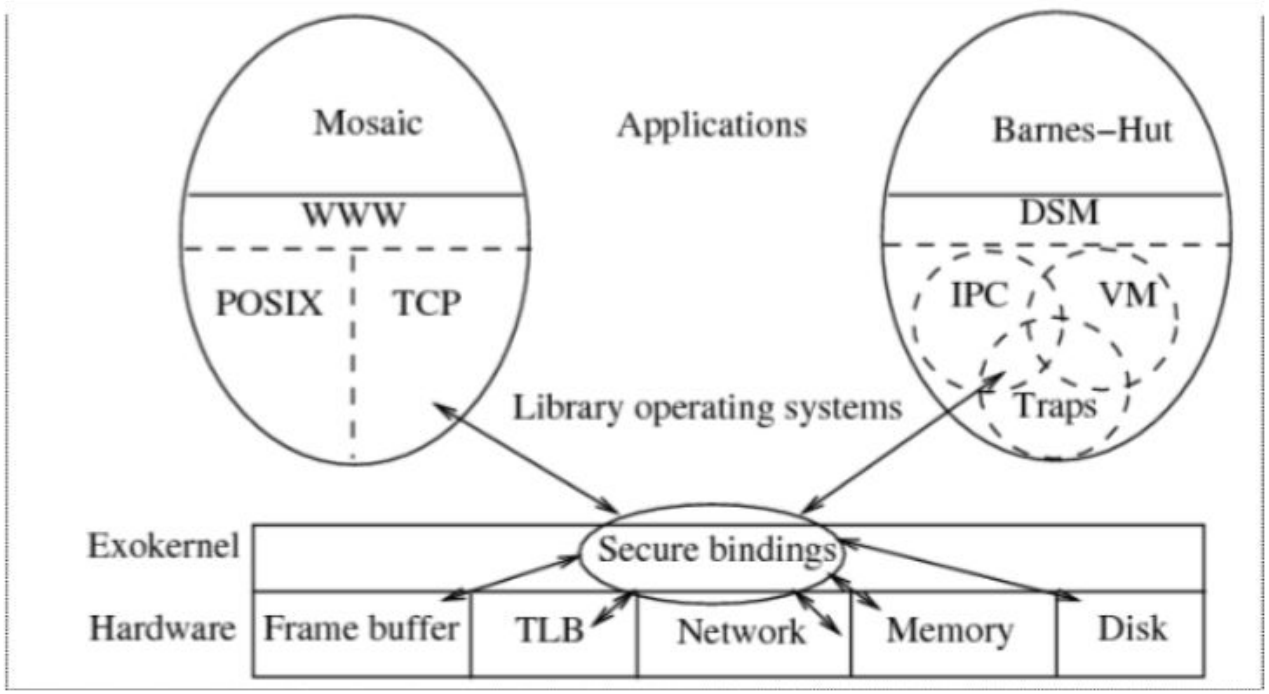
微内核系统结构

- 尽可能把内核功能移到用户空间
- 用户模块间的通信使用消息传递
- 好处: 灵活/安全...
- 损害: 性能

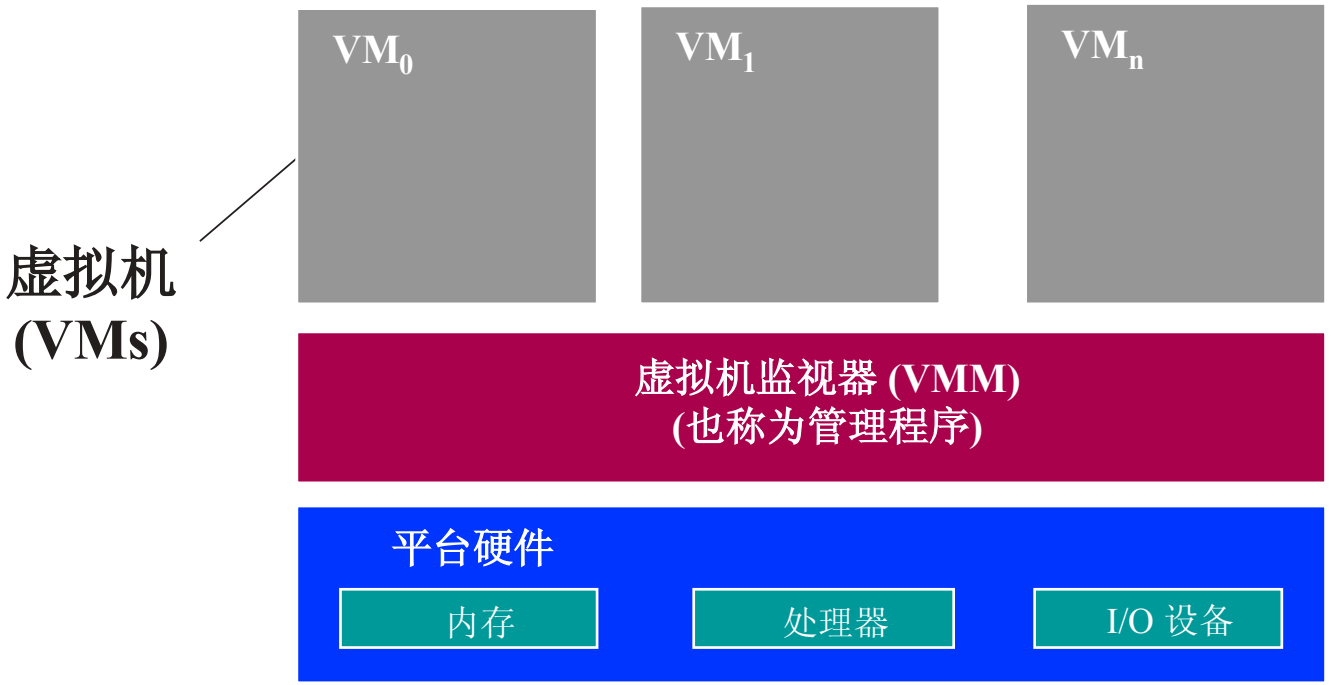


外核结构

- 概要
  - 让内核分配机器的物理资源给多个应用程序, 并让每个程序决定如何处理这些资源.
  - 程序能链接到操作系统库(libOS) 实现了操作系统抽象
  - 保护控制转移, PCT

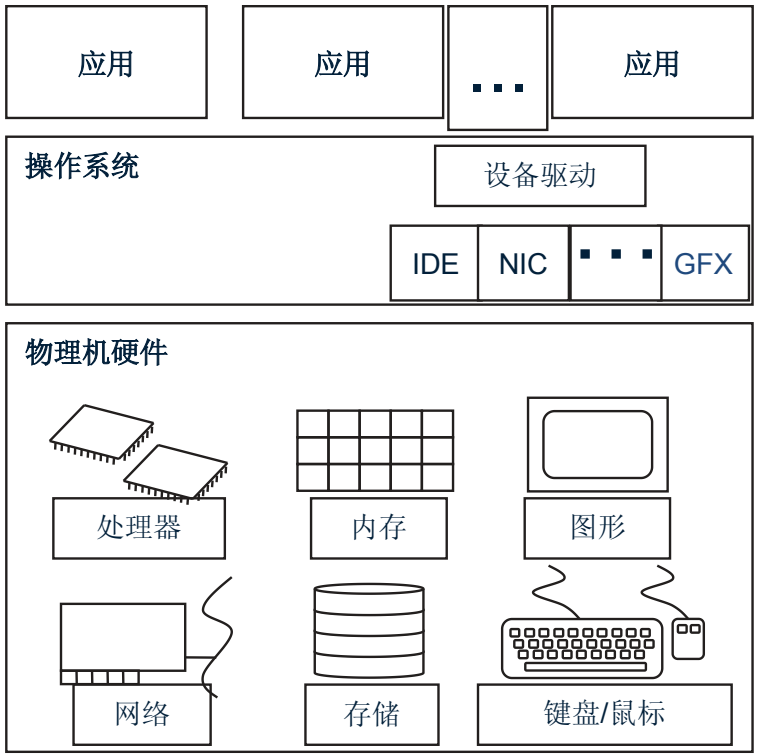


VMM（虚拟机监视器）

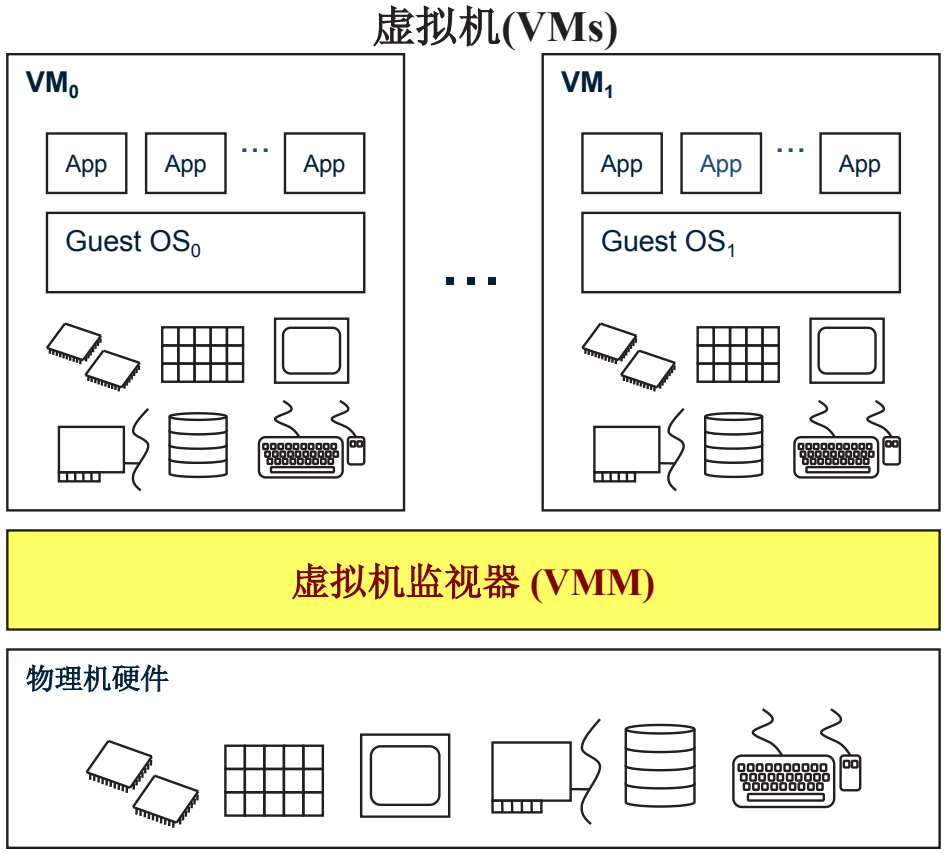


- VMM 将单独的机器接口转换成很多的幻象．每个这些接口（虚拟机）是一个原始计算机系统的有效副本，并完成所有的处理器指令。
  - Robert P. Goldberg.

VMM（虚拟机监视器）



无虚拟机: 单操作系统拥有所以硬件资源



有虚拟机: 多操作系统共享硬件资源

- VMM 将单独的机器接口转换成很多的幻象．每个这些接口（虚拟机）是一个原始计算机系统的有效副本，并完成所有的处理器指令。
  - Robert P. Goldberg.

内容摘要

- 课程概述
- 什么是操作系统?
- 为什么学习操作系统?
- 如何学习操作系统?
- 操作系统实例
- 操作系统的演变
- 操作系统结构

结论

操作系统很有趣，可以管理和控制整个计算机!

但 …

- 它是不完备的
  - Bug、性能异常、功能缺失，有很多的挑战和机遇。
- 它是庞大的
  - 有许多概念、原理和代码需要了解。
- 我们能做到!
  - ... 至少靠你自己的恒心和投入，完全可以在一个学期理解OS的原理和ucore OS的实现