



# 操作系统

Operating System

# 课程信息

主讲教师：	向勇、陈渝
助教：	沈游人，施鹤远，李瞳等

# 课程信息

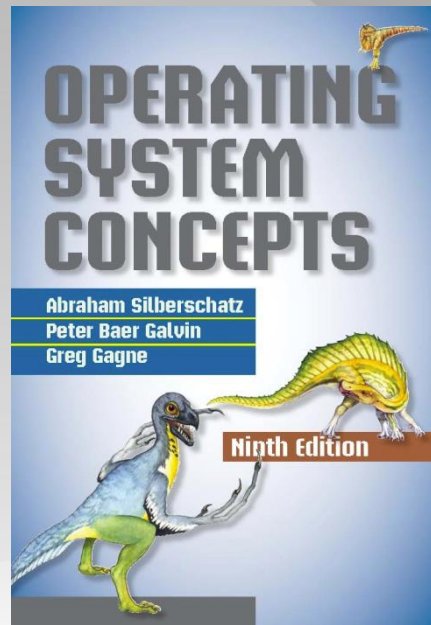
Wiki:	<a href="http://os.cs.tsinghua.edu.cn/oscourse/OS2017spring">http://os.cs.tsinghua.edu.cn/oscourse/OS2017spring</a>
学堂在线:	<a href="http://www.xuetangx.com/courses/course-v1:TsinghuaX+30240243X_2015_T2+2015_T2/about">http://www.xuetangx.com/courses/course-v1:TsinghuaX+30240243X_2015_T2+2015_T2/about</a>
Piazza讨论区:	<a href="https://piazza.com/tsinghua.edu.cn/spring2015/30240243x/home">https://piazza.com/tsinghua.edu.cn/spring2015/30240243x/home</a>
练习和实验:	<a href="http://github.com/chyyuu/ucore_os_lab">http://github.com/chyyuu/ucore_os_lab</a> <a href="http://github.com/chyyuu/ucore_os_doc">ucore_os_doc</a> <a href="http://github.com/chyyuu/os_course_info">os_course_info</a> <a href="http://github.com/chyyuu/simple_os_book">simple_os_book</a>  <a href="http://crl.ptopenlab.com:8811/">http://crl.ptopenlab.com:8811/</a>

# 预备知识

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

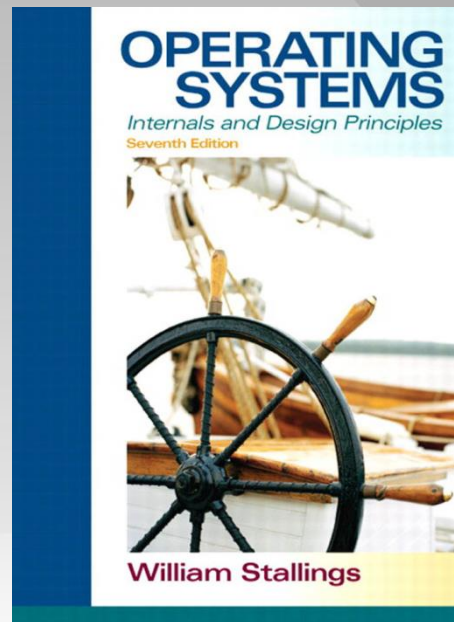
# 参考教材

- Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating system concepts (9th Edition), John Wiley & Sons, 2012
- 操作系统概念（第七版）；Silberschatz、Galvin和Gagne著、郑扣根译；高等教育出版社，2010年；



# 参考教材

- William Stallings, Operating Systems-Internals and Design Principles (8th Edition), Prentice Hall, 2011
- ▶ 操作系统——精髓与设计原理（第七版）；William Stallings著，陈向群、陈渝译；电子工业出版社，2012年；



# 成绩评定

- **实验：20分**
  - 独立完成8个教学实验，并提交实验报告
- **考试或课程设计：80分**
  - 期中考试：35分
  - 期末考试：45分
  - 有余力和兴趣的同学，可用课程设计替代考试

**总成绩加权方法：**上述各项成绩的总和会做一次调整，基本原则是，各分数段保持一定的比例，可能的参考比例为A+/A/A-占25%、B+/B/B-占45%、C+/C/C-占20%和D+/D/F占10%。

# 教学内容

- 操作系统结构
- 中断及系统调用
- 内存管理
- 进程及线程
- 处理机调度
- 同步互斥
- 文件系统
- I/O子系统



# 练习与实验

- 操作系统练习

- ▣ 课堂练习

- 操作系统实验

- ▣ 实验0: 实验环境准备

- ▣ 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

# 课程设计

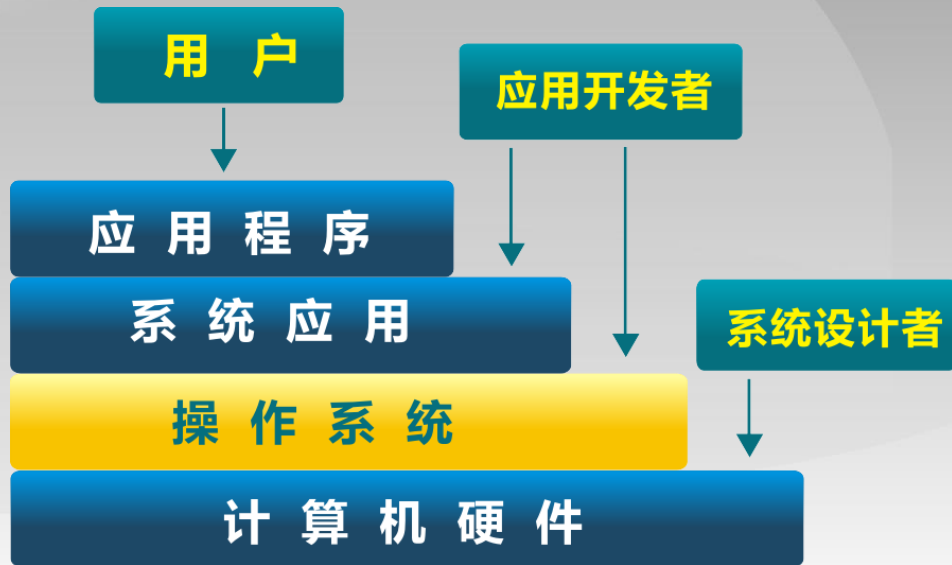
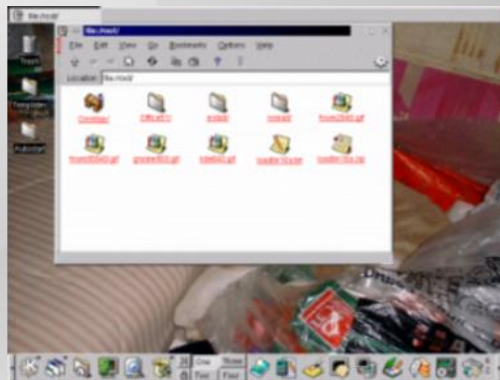
- ucore on RISC-V CPU, etc.



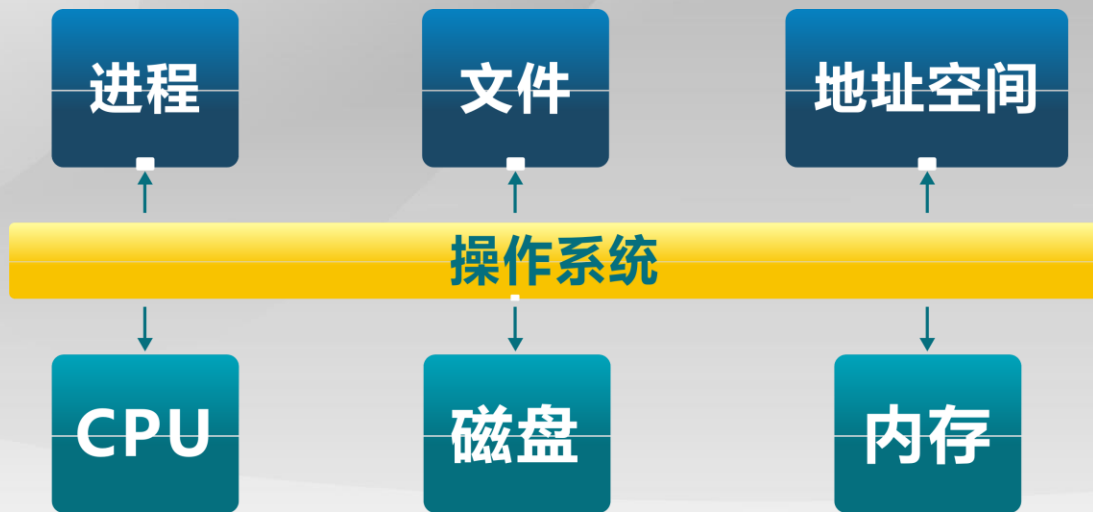
# 操作系统定义

- 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
  - ▣ 控制程序执行过程，防止错误和计算机的不当使用
  - ▣ 执行用户程序，给用户程序提供各种服务
  - ▣ 方便用户使用计算机系统
- 操作系统是一个资源管理器
  - ▣ 应用程序与硬件之间的中间层
  - ▣ 管理各种计算机软硬件资源
  - ▣ 提供访问计算机软硬件资源的高效手段
  - ▣ 解决资源访问冲突，确保资源公平使用

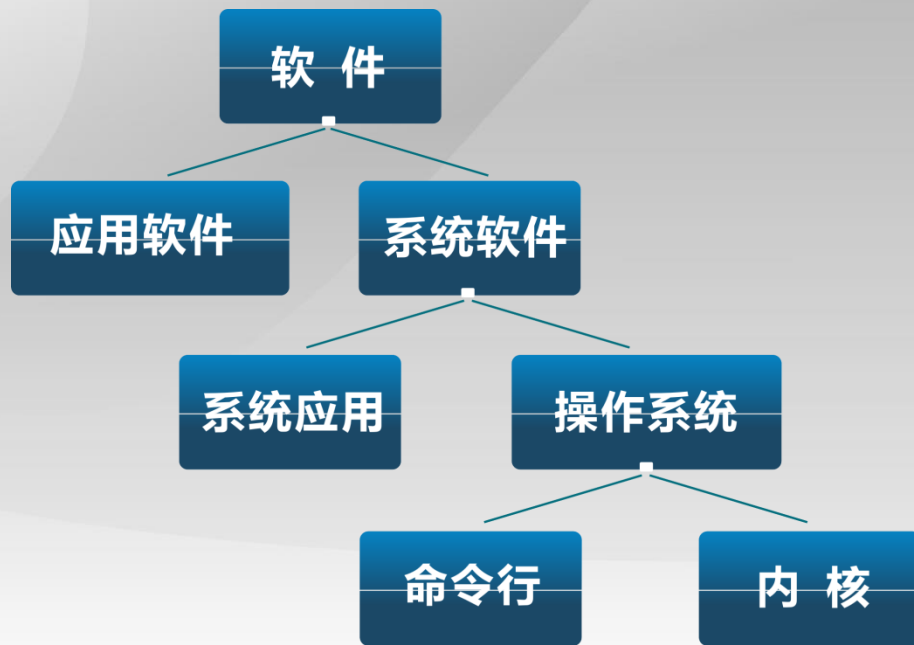
# 操作系统的地位



# 操作系统的地位



# 操作系统软件分类



# 操作系统软件的组成

- **Shell**--命令行接口

- ▣ 通过键盘操纵。
- ▣ 方便用户进行命令输入

- **GUI**--图形用户接口

- ▣ WIMP

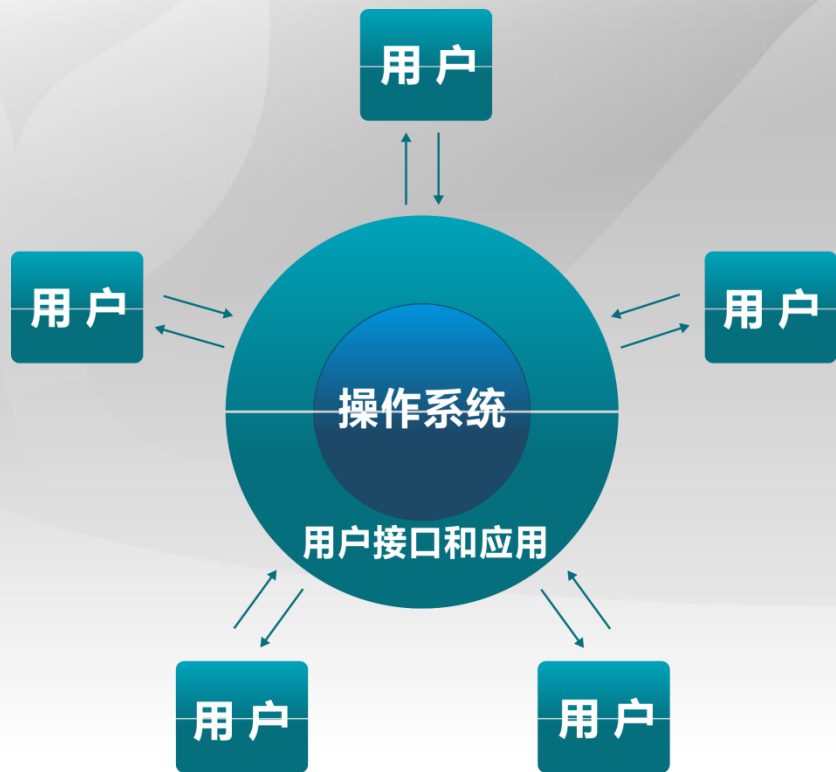
(视窗 (Window) 、图标 (Icon) 、选单 (Menu) 、指标 (Pointer) )

- ▣ 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

# 操作系统软件的组成





# ucore教学操作系统内核

内核	应用程序		
	命令行程序 编译器 解释器 系统库		
	系统调用接口		
	信号	文件管理系统	CPU 调度
	字符设备I/O	块设备I/O	虚拟内存管理
	串口驱动	磁盘驱动	物理内存管理
	硬件抽象层		
	串口控制器 终端设备	块设备控制器 磁盘和磁带	存储控制器 物理内存

# 操作系统内核特征

- 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

# 课程设计

- 在真实系统上运行ucore+
  - ▣ RISC-V
  - ▣ Others.
- 自选操作系统题目

<http://os.cs.tsinghua.edu.cn/oscourse/OS2017/>





# 操作系统

Operating System

# 操作系统课是多门课程的综合

- 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

# 学习操作系统的目的

- 已有操作系统很好，我将来的工作不会写操作系统
  - ▣ Windows, Linux.
- 已有操作系统是否解决了所有的事？
- 为什么我要学习它？



写操作系统很酷！

操作系统很有用！

我想了解操作系统到底是如何工作的？

掌握操作系统是一个挑战！

我要参与系统软件开发

# 操作系统软件的地位

- 操作系统：计算机科学研究的基础之一
  - ▣ 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

# 哪里在做操作系统研究？

- 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, ...
  - ▣ 国内： 阿里巴巴、百度、华为...
- 研究协会
  - ▣ ACM SIGOPS
  - ▣ USENIX



# 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

# 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>

# 掌握操作系统具有挑战性（1）

- 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

## 掌握操作系统具有挑战性 (2)

- 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

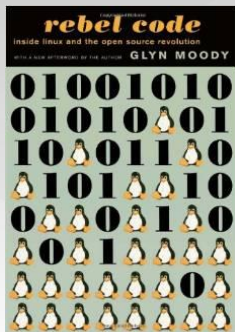
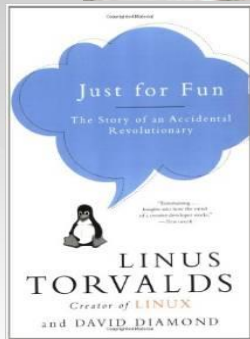
# 掌握操作系统具有挑战性（3）

- 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！

# 如何学习操作系统？

- “不闻不若闻之，闻之不若见之，见之不若知之，知之不若**行之**；学至于行之而止矣。”  
--荀子《儒效篇》
- “天才是1%的灵感加上**99%**的汗水”  
-- Thomas Edison
- “困难，最好的和最**有趣**的三年级课程！”

# 如何学习操作系统?



```
for (i=1; i<=13; i++) {  
  1. 预习，完成第i周课前视频学习(optional)  
  2. 完成第i周基本练习(optional)  
  3. 上课听讲，提问/被提问  
  4. 在deadline前，按序完成ucore_lab实验  
  5. 复习/做课后练习，if (碰到问题) 到piazza提问;  
  6. if (i==7) 参加期中考试;  
  7. if (i==13) 参加期末考试;  
}
```



# 操作系统

Operating System



# UNIX家族



## UNIX BSD



iOS 6

# Linux家族



Linux



1.5  
Cupcake



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





# 操作系统

Operating System

# 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	0.33-1MIPS	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 144MB floppy drives	1TB-7200RPM, SATA 3.0Gb/s

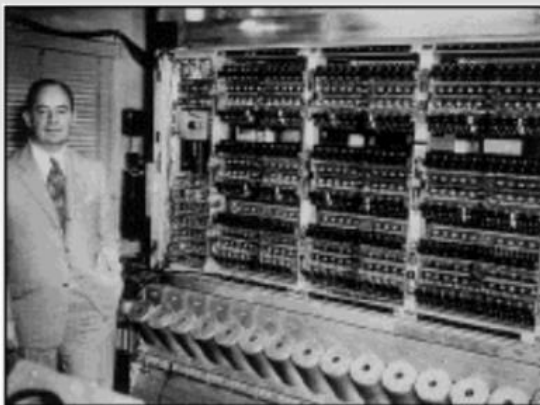
# 操作系统的演变

- 单用户系统
- 批处理系统
- 多道程序系统
- 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)

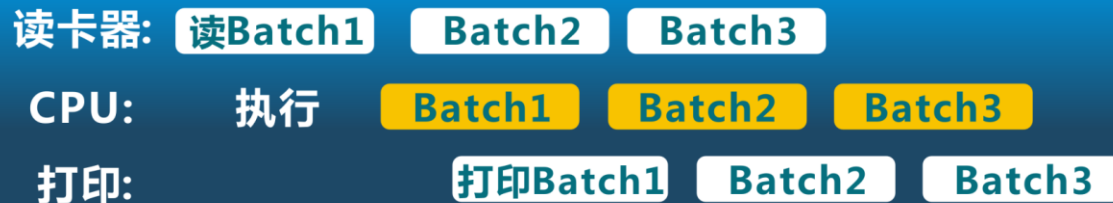
- 操作系统=装载器+通用子程序库
- 问题：昂贵组件的低利用率

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



# 批处理 ('55-'65)

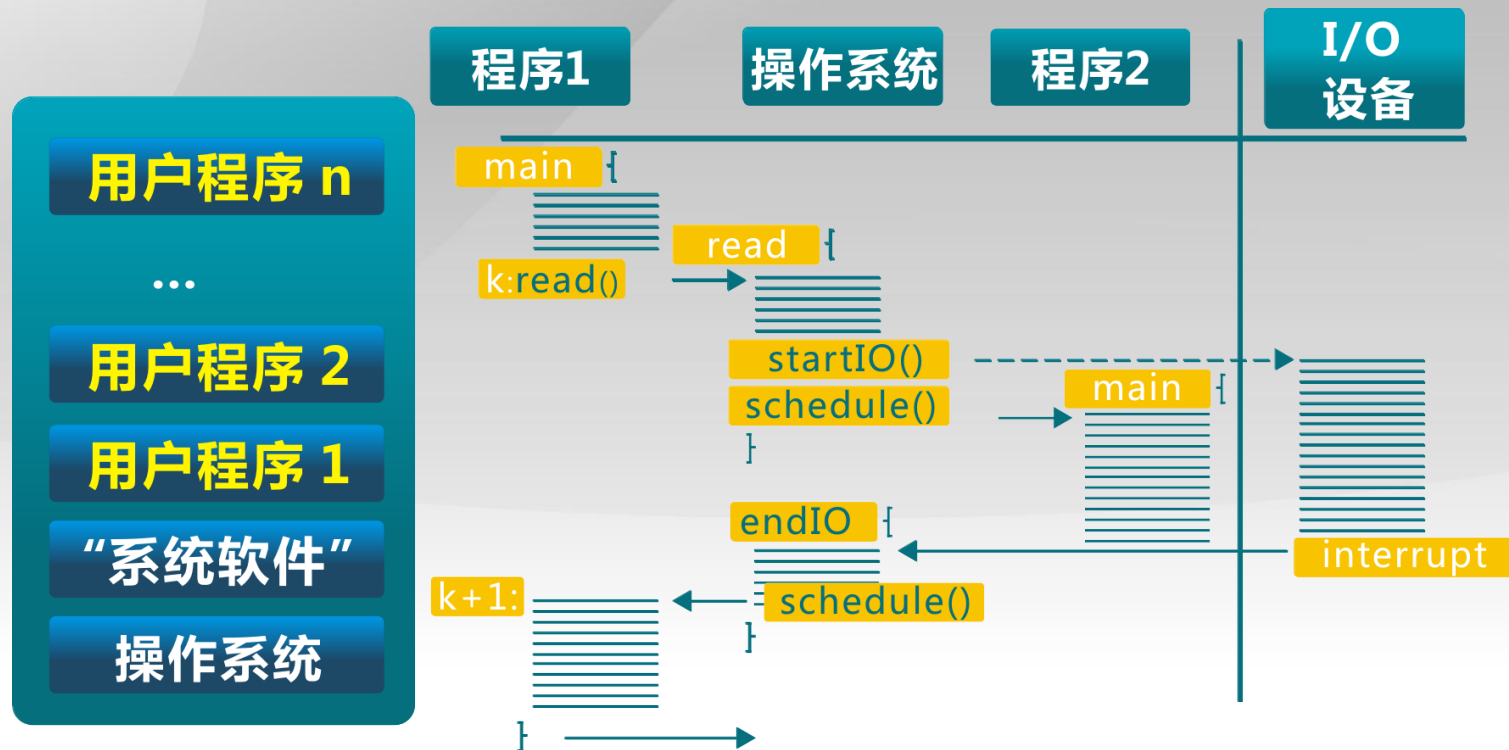
## ■ 顺序执行与批处理





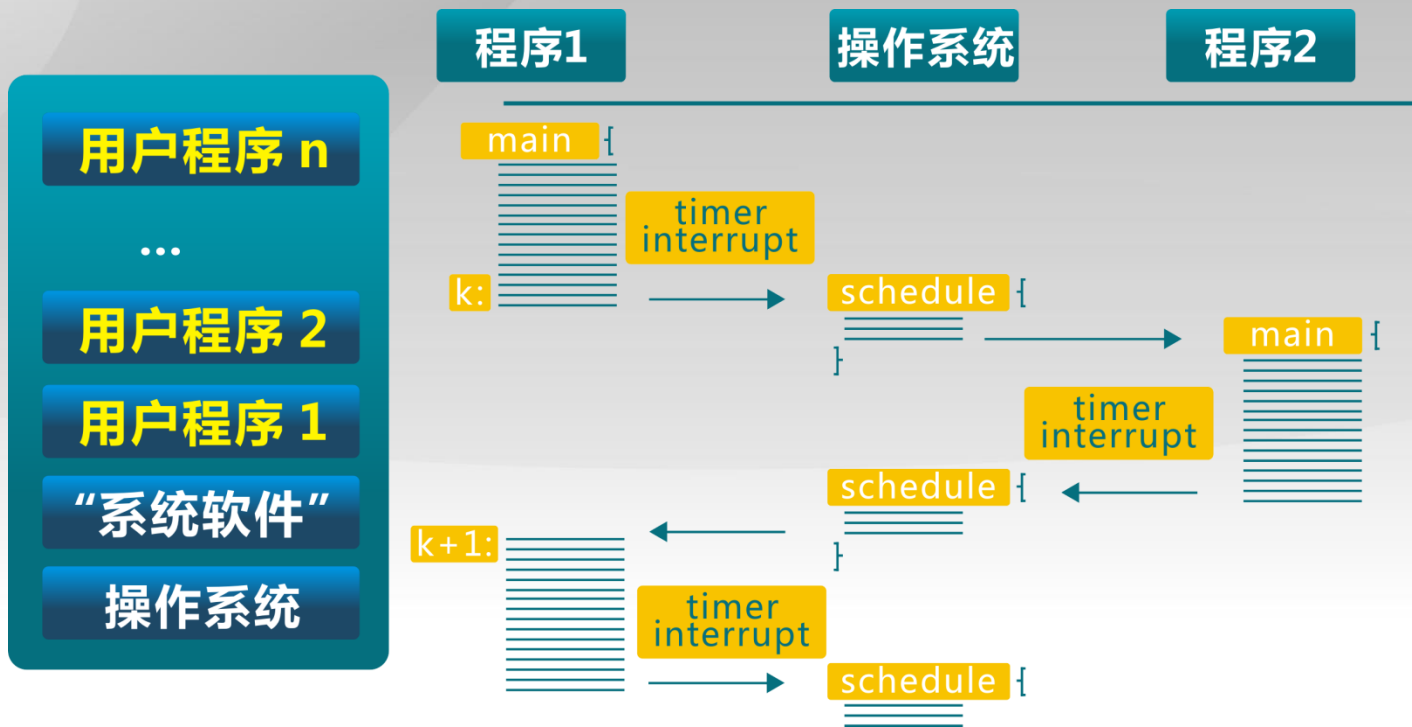
# 多道程序 ('65-'80)

- 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)

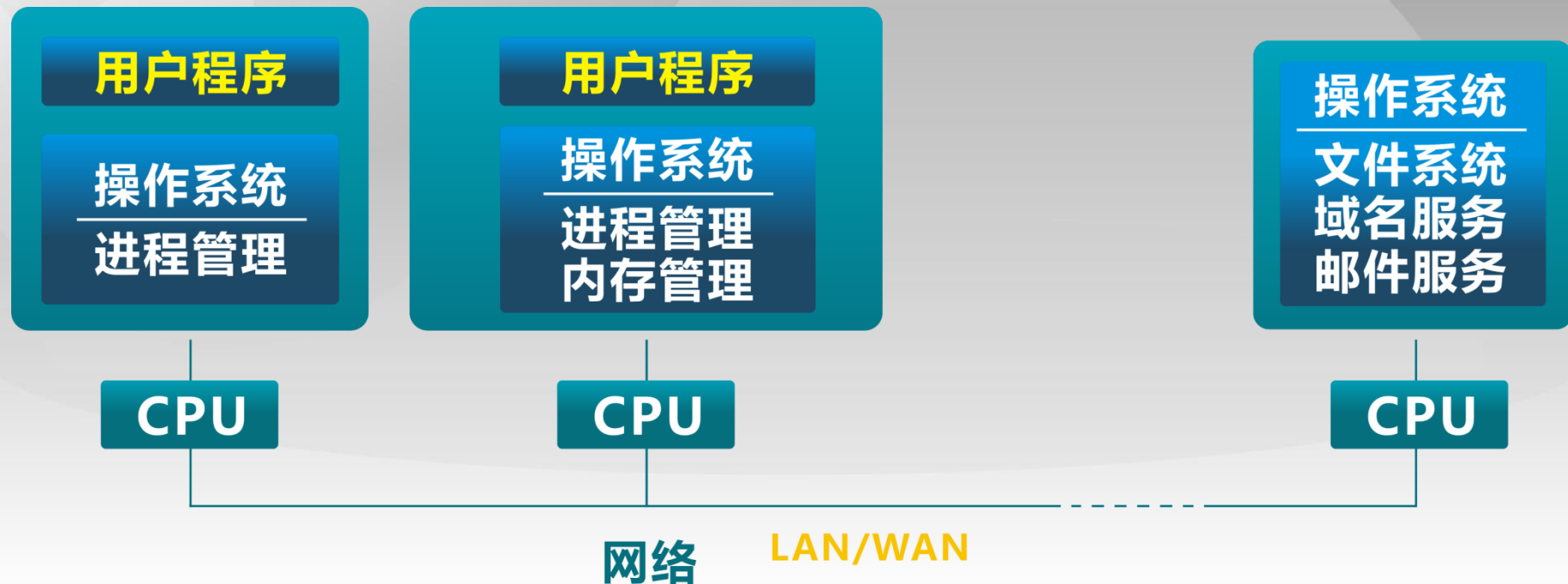
- ▣ 现在: 支持协调和沟通的多应用系统

- ▣ 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

# 分布式操作系统



# 操作系统演变中的计算机系统

Future OS

iOS/Android,...

Windows/Linux/BSD,...

AIX/HP-UX



主机型计算(Mainframe computing)

Internet服务

网络计算(Internet computing)

个人机计算(Personal computing)



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

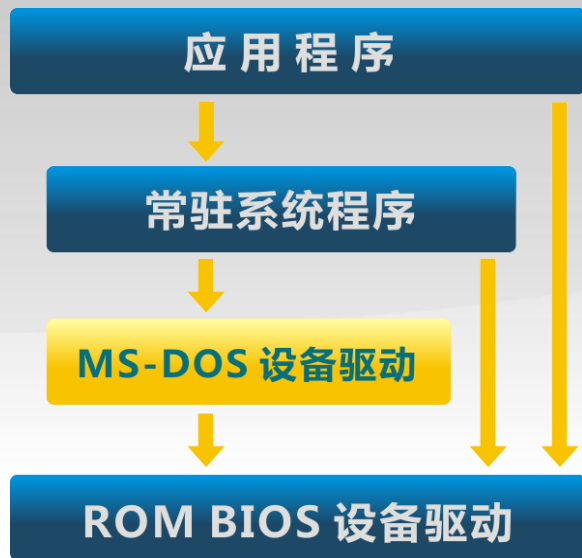


# 操作系统

Operating System

# 简单结构

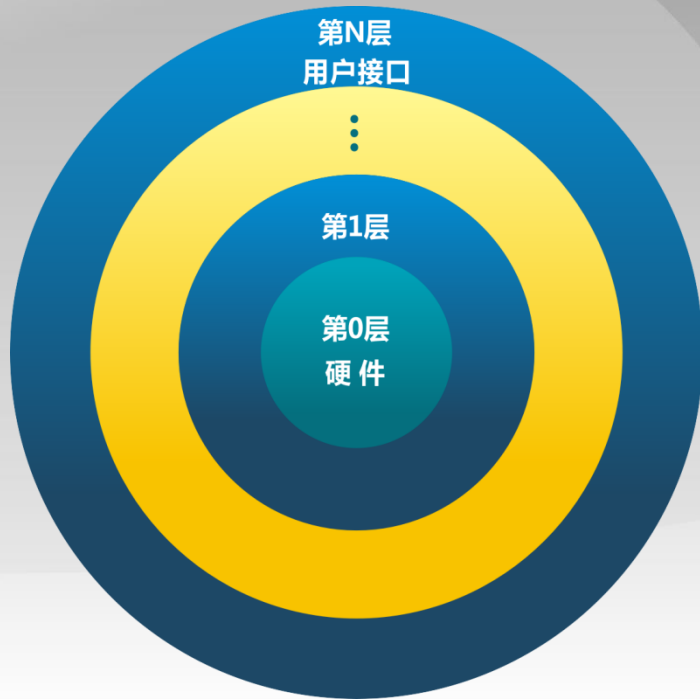
- MS-DOS – 在最小的空间，设计用于提供大部分功能 (1981~1994)
  - ▶ 没有拆分为模块
  - ▶ 虽然 MS-DOS 在接口和功能水平没有很好地分离，主要用汇编编写





# 分层结构

- 将操作系统分为多层 (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
- 每一层仅使用更低一层的功能(操作) 和服务。



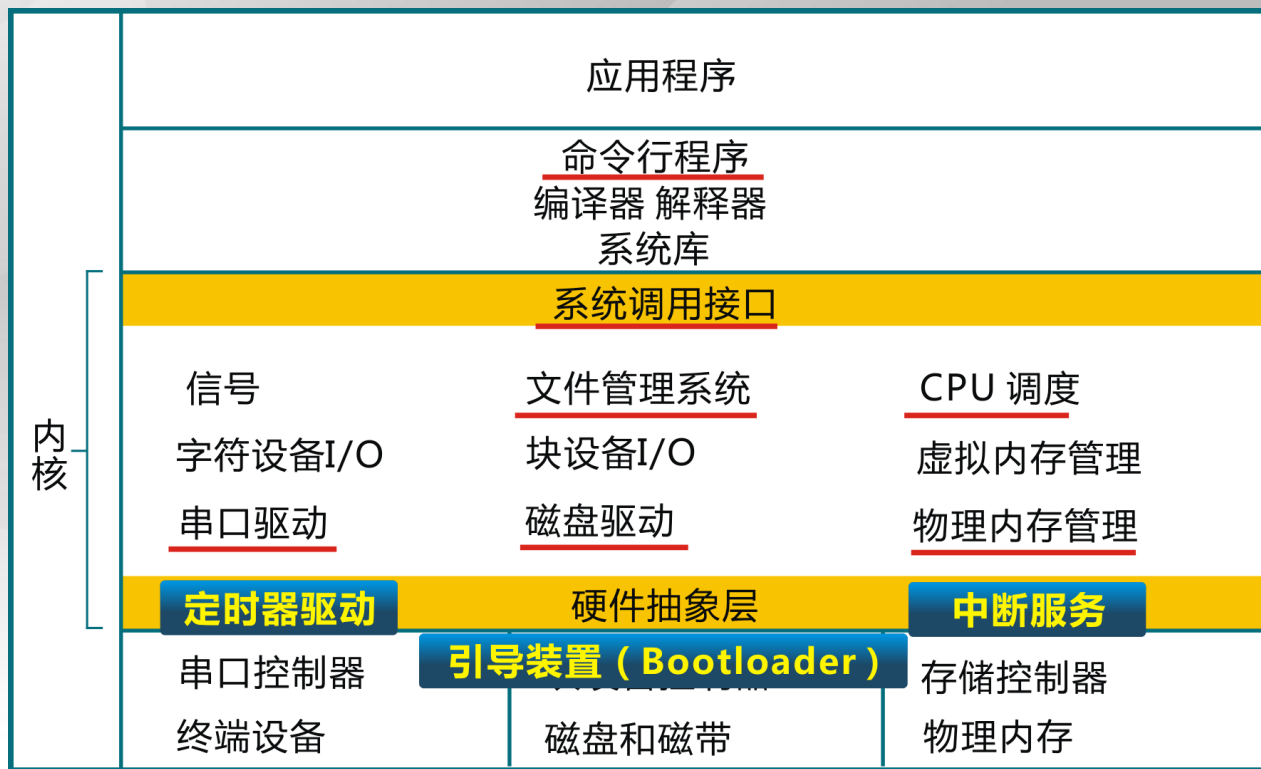
# UNIX操作系统与C语言

- 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操作系统结构



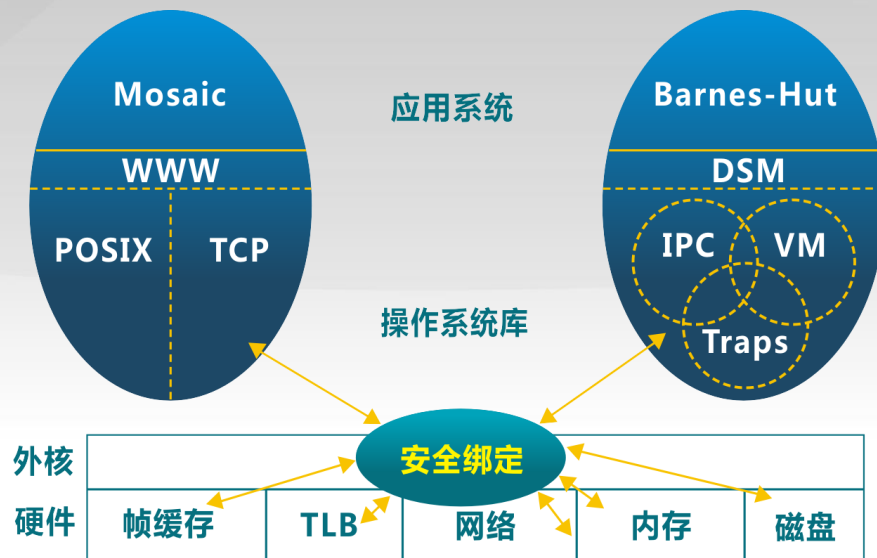
# 微内核结构 (Microkernel)

- 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...
- 缺点: 性能



# 外核结构 (Exokernel)

- 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
- 保护与控制分离



# VMM (虚拟机管理器)

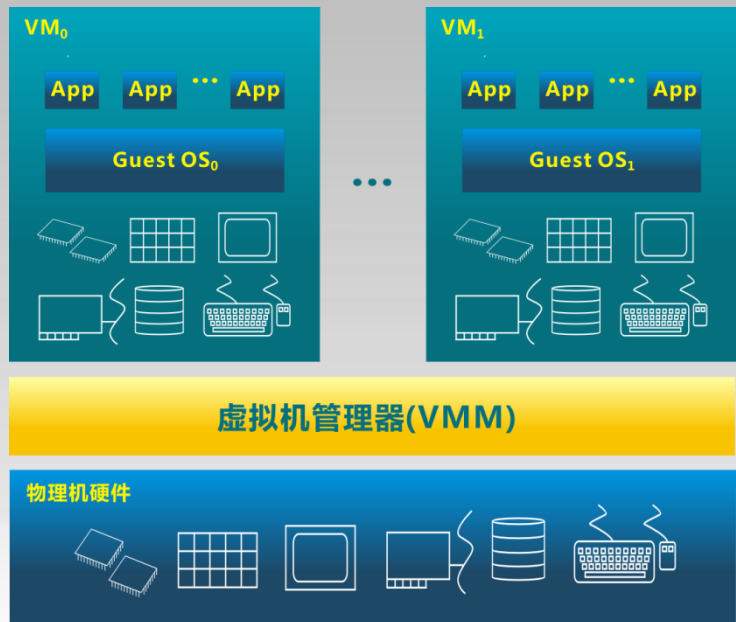


- 虚拟机管理器将单独的机器接口转换成很多的虚拟机，每个虚拟机都是一个原始计算机系统的有效副本，并能完成所有的处理器指令。

# VMM (虚拟机管理器)



无虚拟机：单操作系统拥有所有硬件资源



有虚拟机：多操作系统共享硬件资源

# 小结

- 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。