

计算机系统

教师：王雷

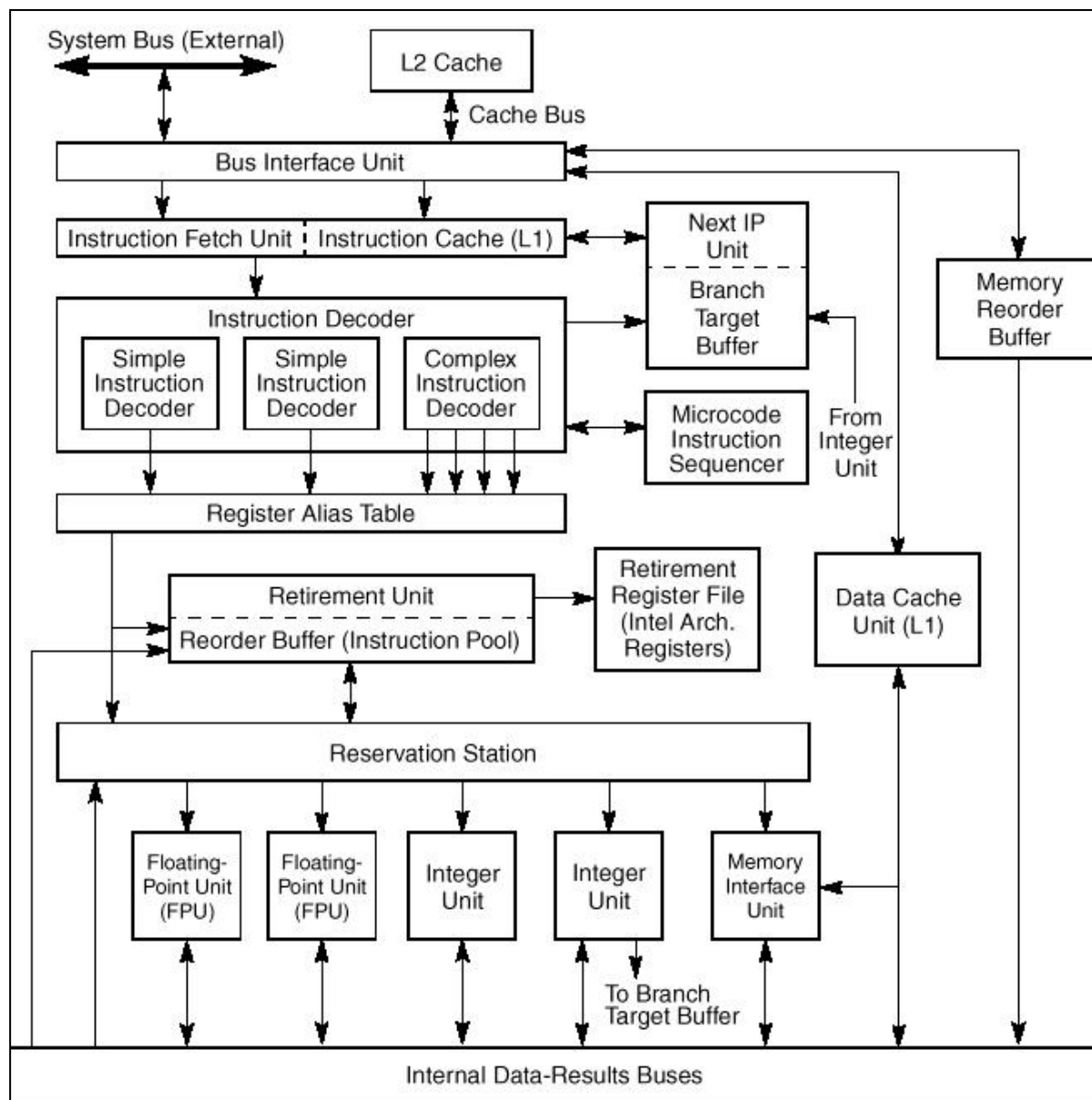
82316284, wanglei@buaa.edu.cn

操作系统引论

- 计算机硬件简介
- 操作系统的基本实现机制
- 操作系统的基本类型
- 操作系统的特征和功能
- 操作系统结构

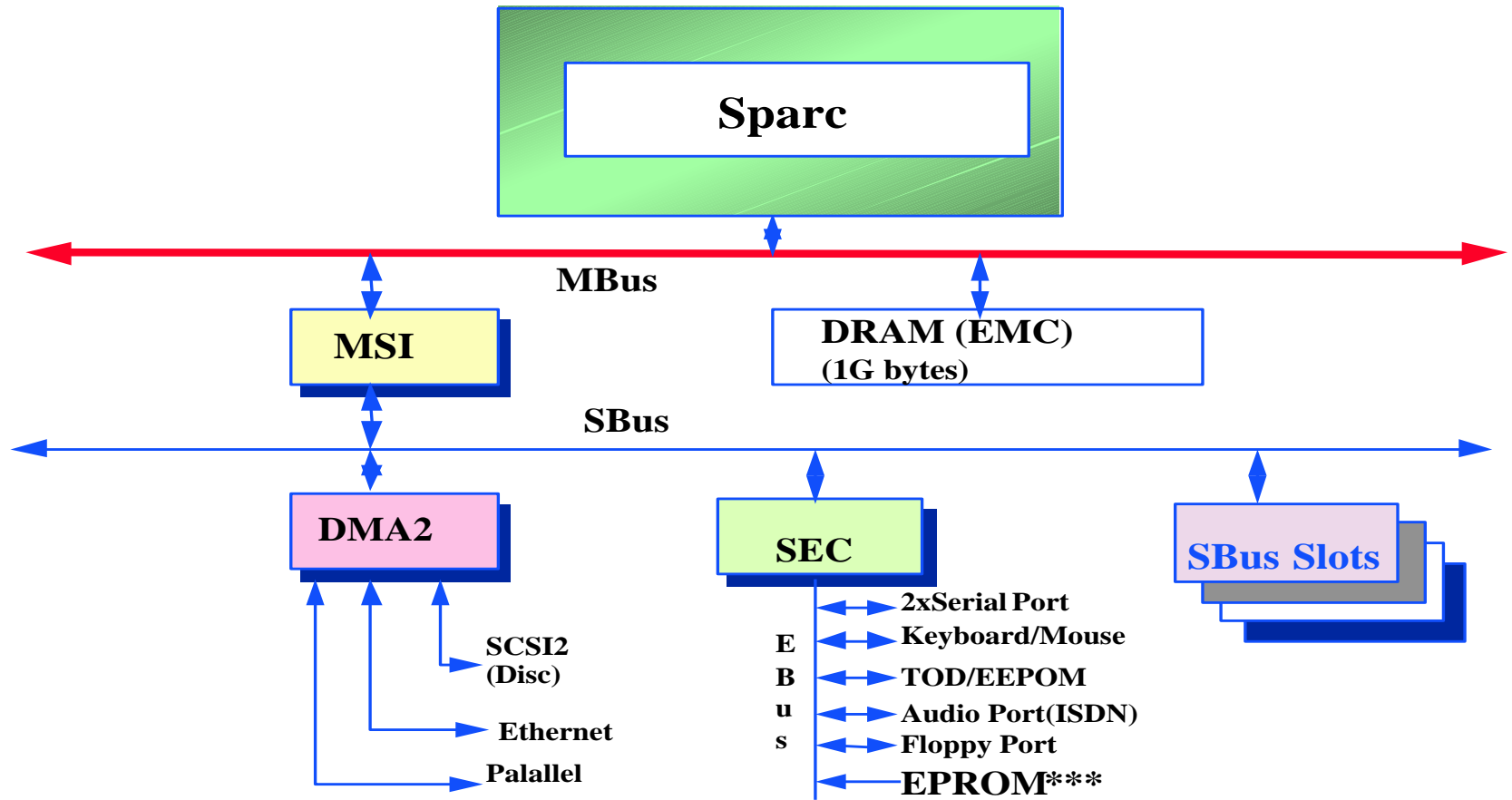
计算机硬件简介

- 硬件是基础，必须了解硬件。
- 但不必象硬件工程师那样，我们了解的是功能、接口和状态。



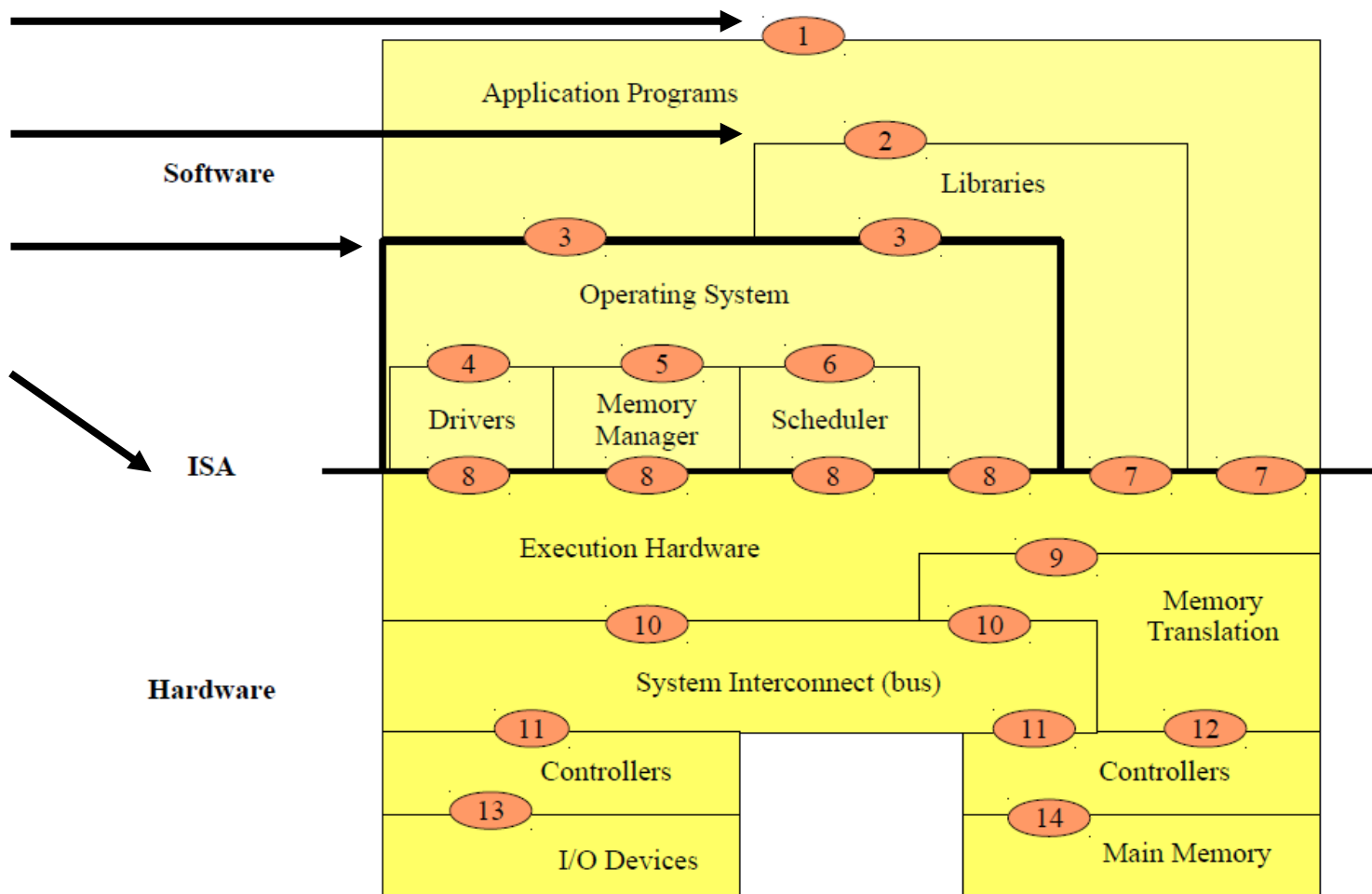
Sparc

Diagram of Sparc Workstation



计算机体系结构中的“接口”

- UI
- API
- ABI
- ISA



Olivier Gruber,

Virtual Machines. http://sardes.inrialpes.fr/~gruber/Courses/MosigM2_AdvancedOS/Intro-RealMachines.pdf

计算机的存储结构

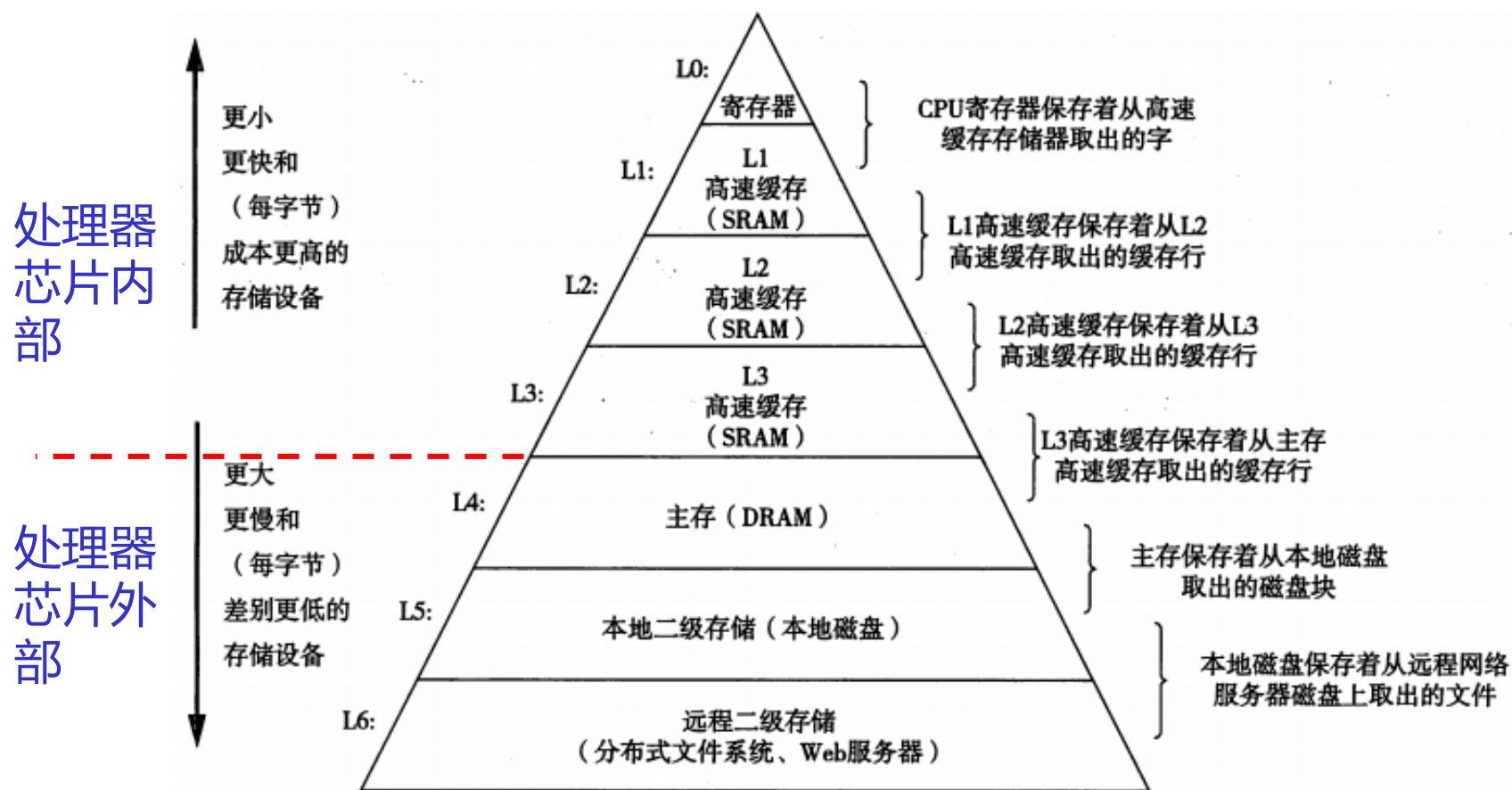
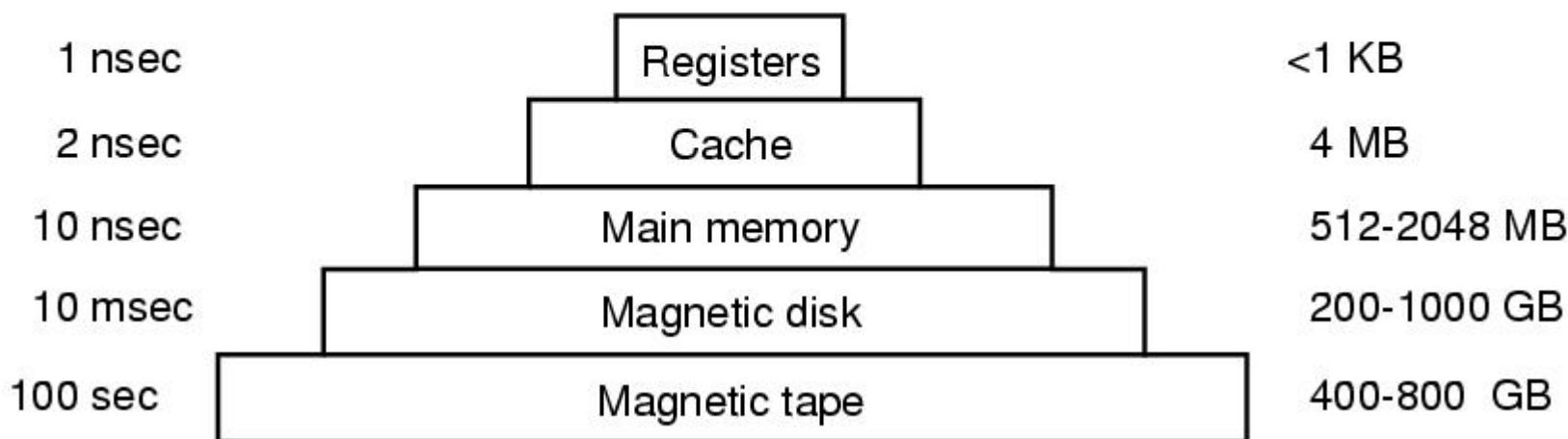


图 6-23 存储器层次结构

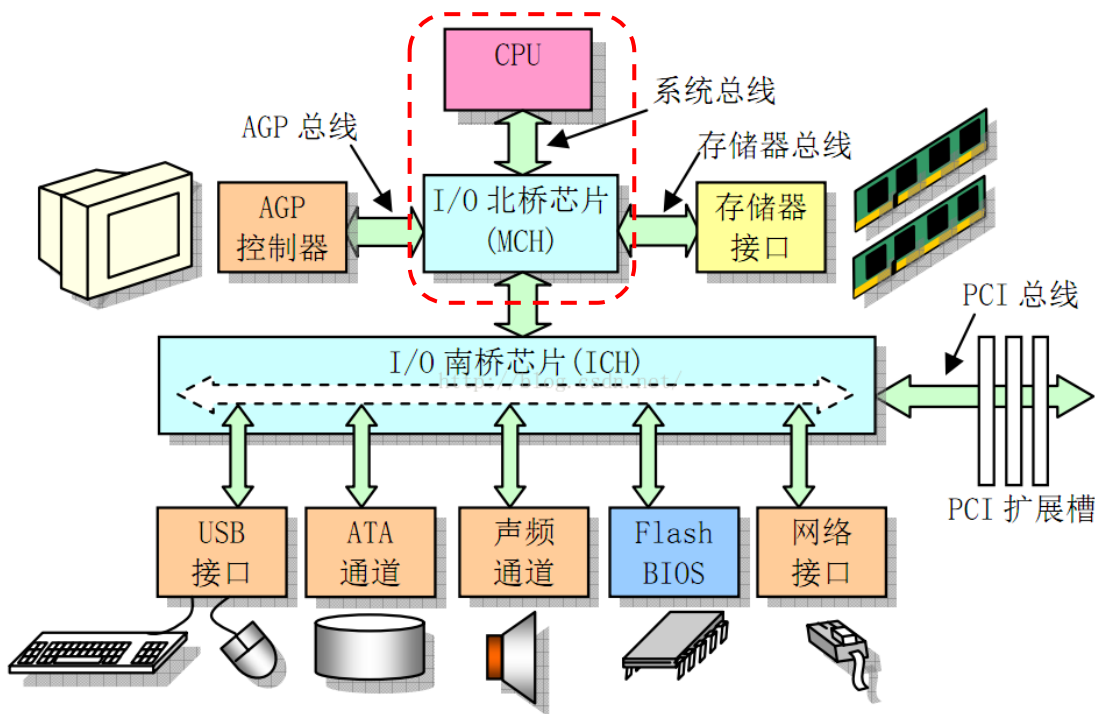
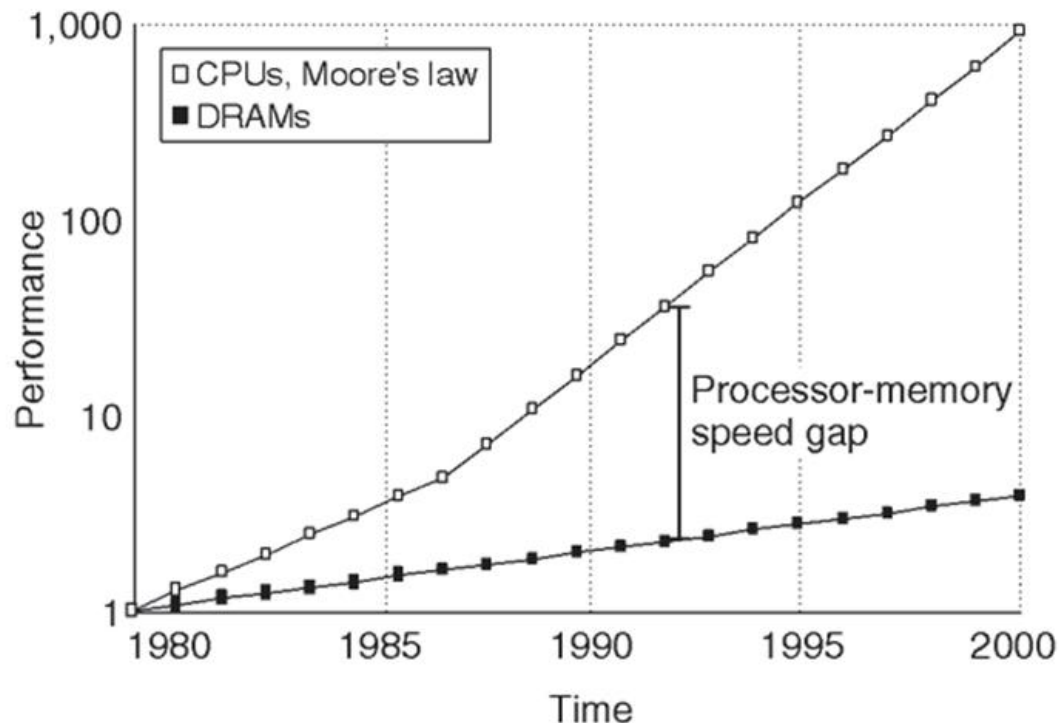
计算机的存储结构

Typical access time

Typical capacity



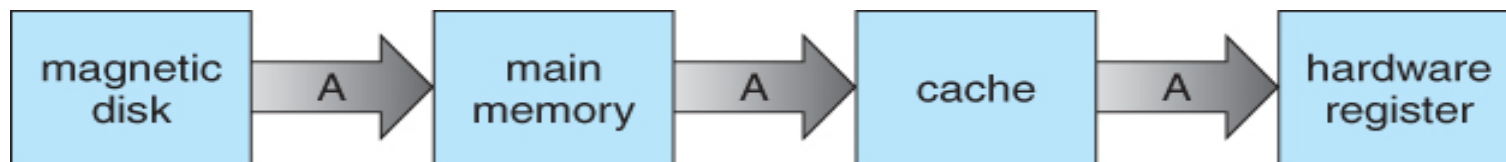
1980~2000这20年间,
处理器计算速度提升了
1000倍 (每年60%) ,
而DRAM的速度近提升了
不到4倍 (每年7%) !



高速缓存 (Cache) 通常
位于处理器内部, 而内存
及IO设备通常处于处理器
芯片的外部。

各层存储设备的对比

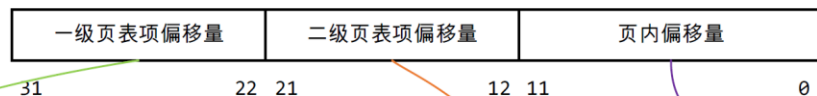
Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape



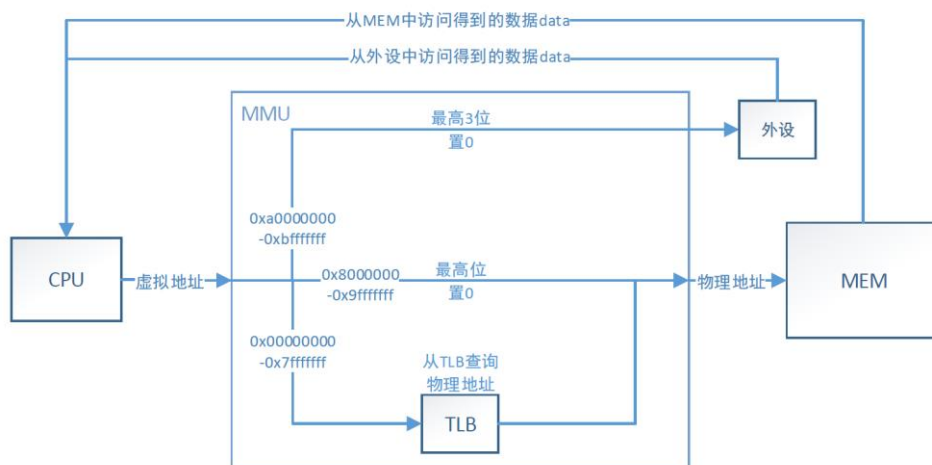
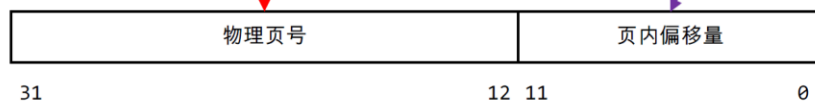
如何管理？

MIPS内存管理

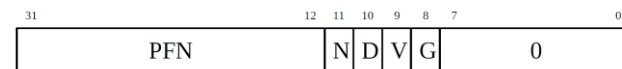
虚拟地址



物理地址



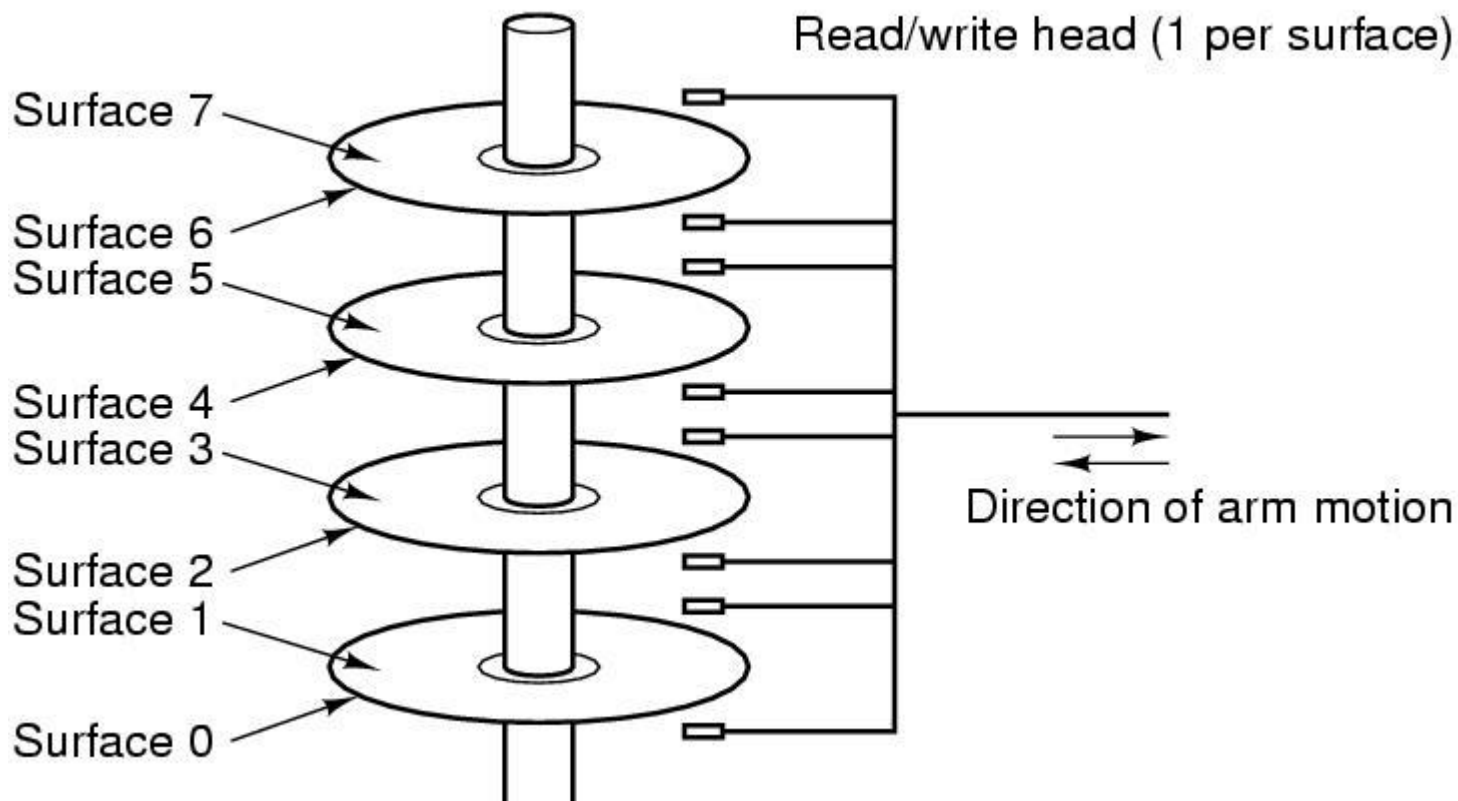
EntryHi Register (TLB Key Fields)



EntryLo Register (TLB Data Fields)

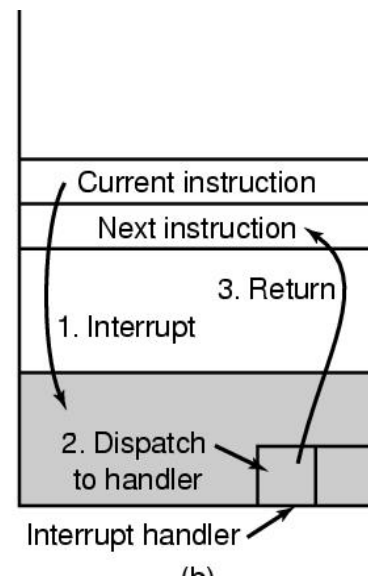
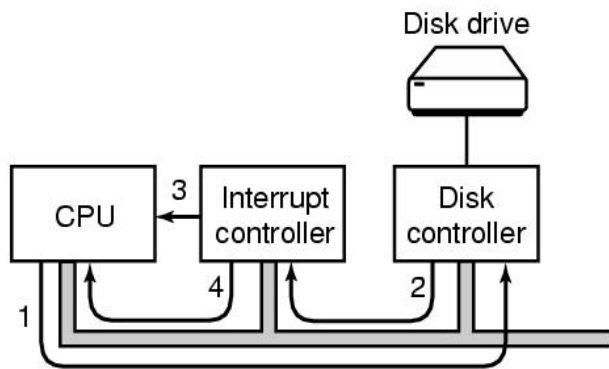
磁盘

- 柱面→磁道→扇区 (512字节)

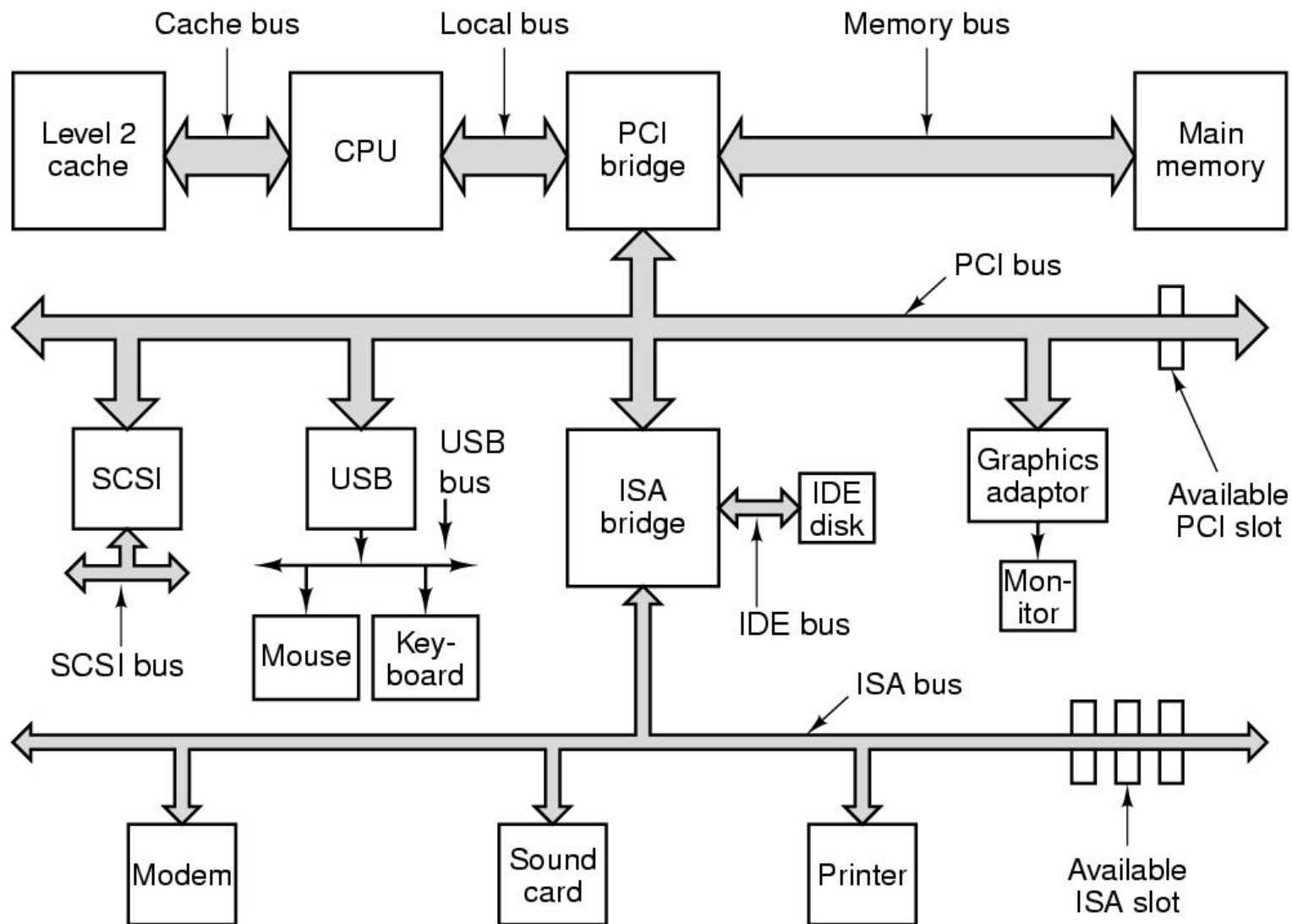


I/O设备

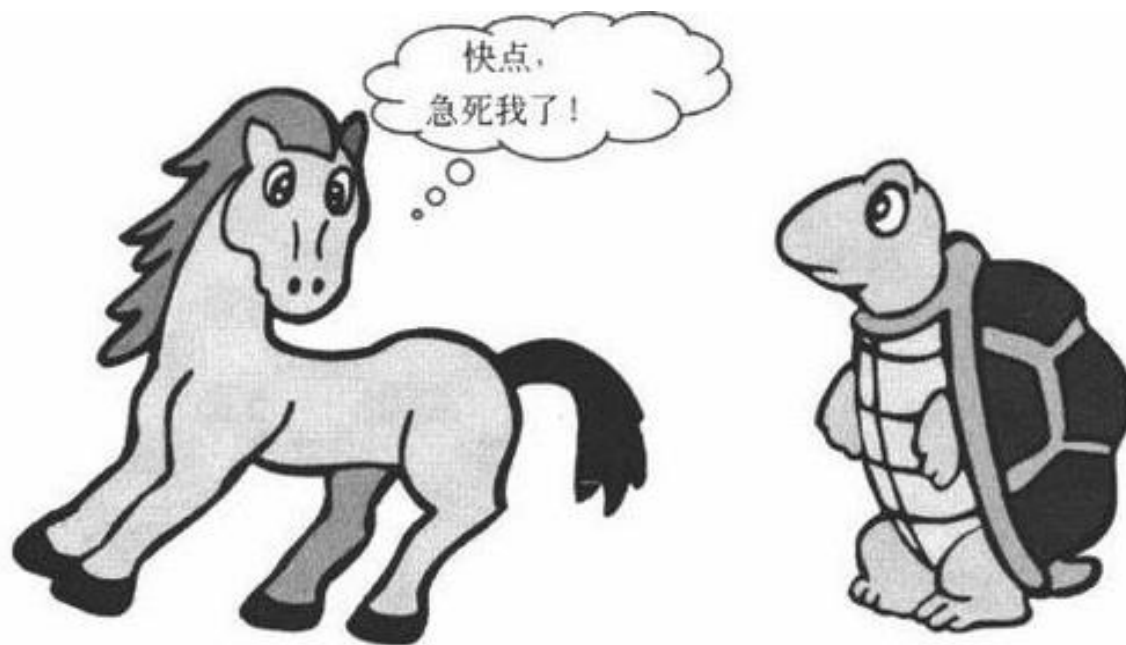
- 设备控制器
- 设备驱动程序（一般在内核态运行），通过设备寄存器与设备控制器通信
- CPU处理速度快，访问I/O设备慢，如何处理？
 - 忙等待
 - 中断
 - DMA



总线



计算与访存及IO速度失配



计算速度很快

访存及IO速度很慢

■ 原因:

- 处理器与内存的制造工艺差异;
- 内存及IO设备处于处理器的外部, 离处理器“远”。

该特点在电子计算机诞生的第一天就存在, 而且越来越突出

问题1

- 操作系统如何提高硬件平台的使用效率？

操作系统引论

- 计算机硬件简介
- 操作系统的基本实现机制
- 操作系统的基本类型
- 操作系统的特征和功能
- 操作系统结构

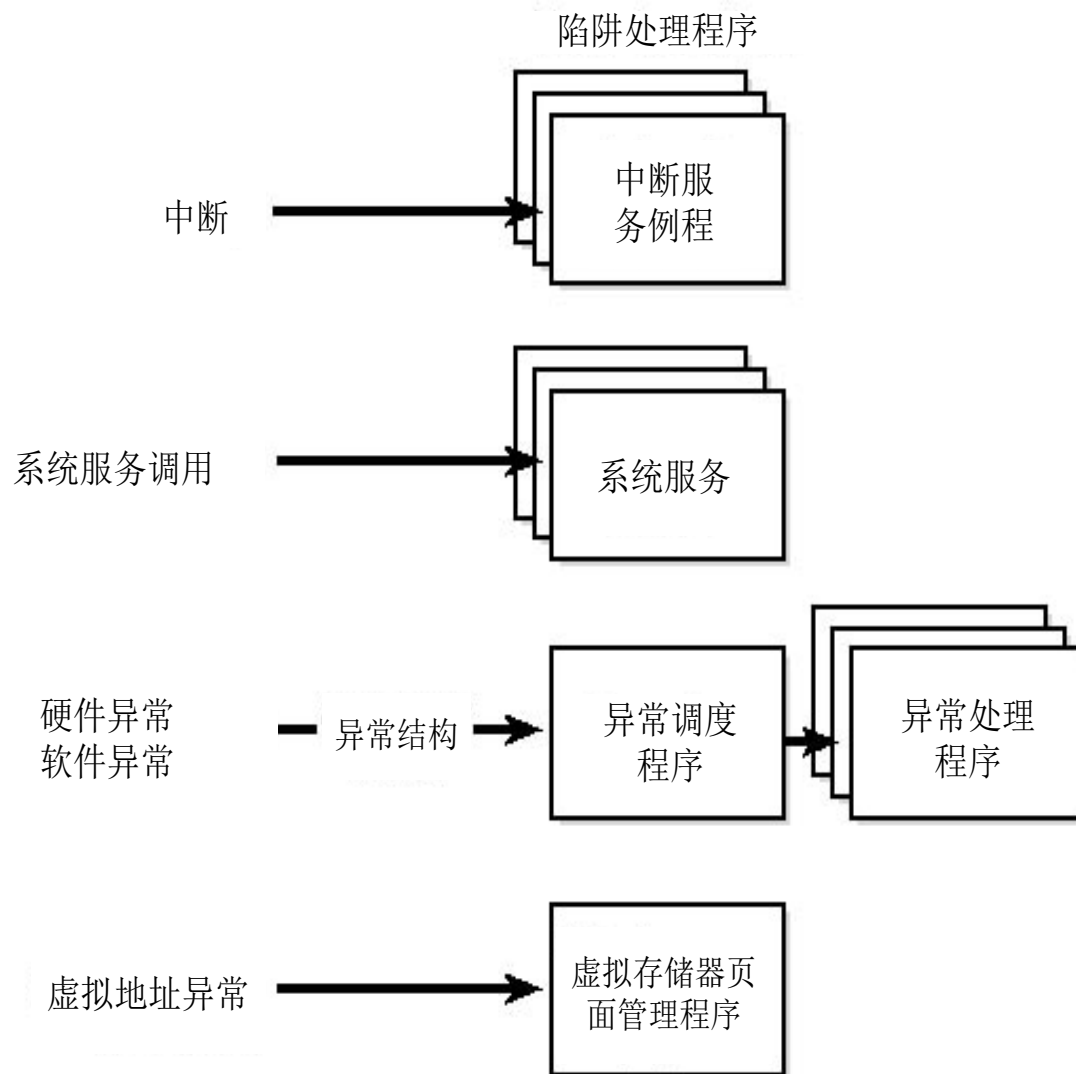
异常(exception): 陷阱(trap)和中断(interrupt)

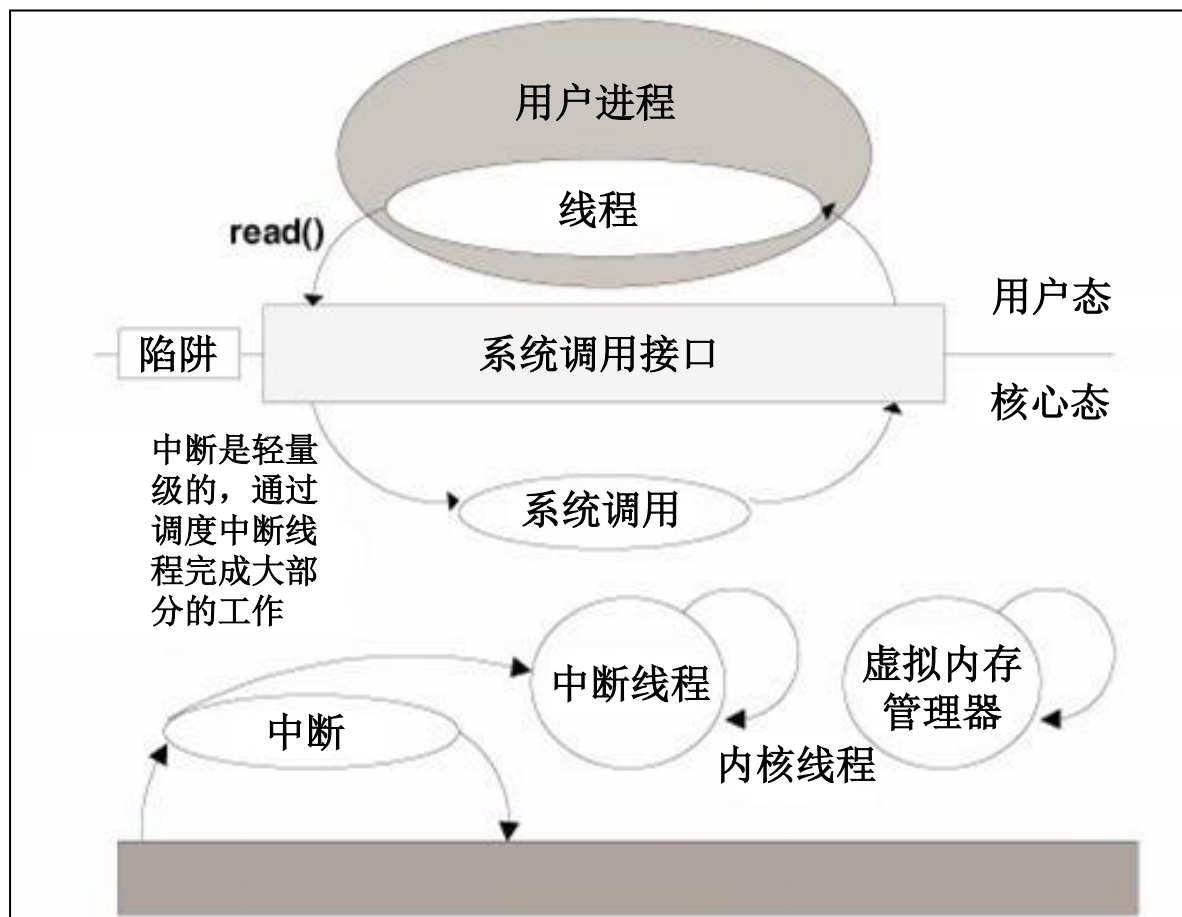
- 中断是异步异常，可能随时发生，与处理器正在执行的内容无关。中断主要由I/O设备、处理器时钟或定时器产生，可以被启用或禁用。 (*Asynchronous Exceptions*)
- 同步异常，它是某一特定指令执行的结果。在相同条件下，异常可以重现。例如内存访问错误、调试指令以及被零除。 (*Synchronous exceptions*)
- 系统调用也视作同步异常，或trap。
- 软件和硬件都可以产生中断。
- 陷阱(trap)帧：完整的线程描述表的子集，用于现场保护
- 陷阱处理程序处理少量事件，多数转交给其他的内核或执行体模块处理

参考：<http://flint.cs.yale.edu/cs422/doc/art-of-asm/pdf/CH17.PDF>

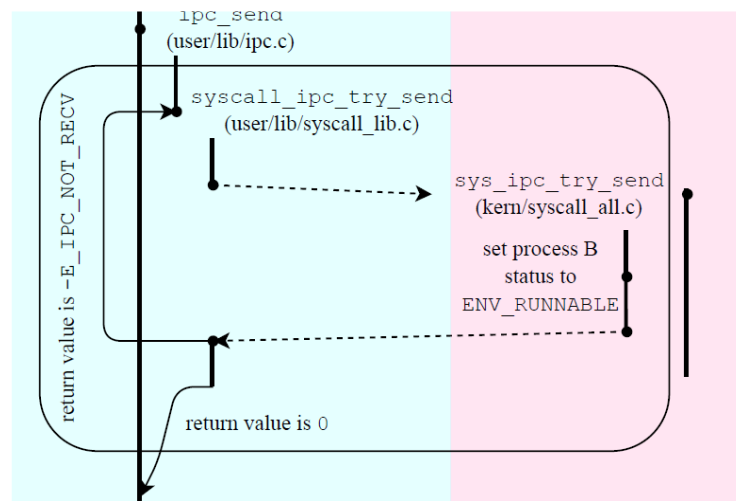
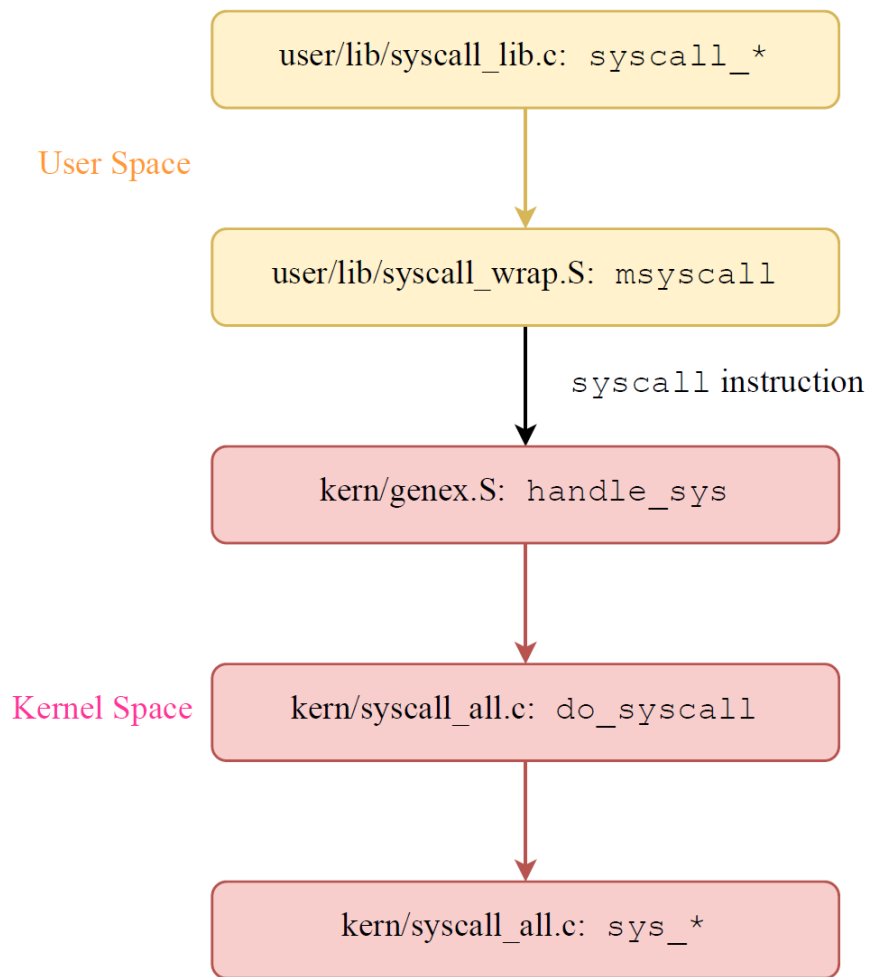
异常(exception): 陷阱(trap)和中断(interrupt)

类别				原因	返回行为	例子
异常	异步	中断 (interrupt)	可屏蔽中断	来自 I/O 设备的信号	总是返回到下一条指令	所有的 IRQ 中断
			不可屏蔽中断			电源掉电和物理存储器奇偶校验
	同步	陷阱 (trap)		程序内部有意设置	总是返回到下一条指令	系统调用、信号机制等 (通过中断指令实现)
		故障 (fault)		潜在可恢复的错误	返回到当前指令	缺页异常、除 0 错误、段错误
		终止 (abort)		不可恢复的错误	不会返回	硬件错误





■ 系统调用指令：syscall、int 80等



操作系统引论

- 计算机硬件简介
- 操作系统的基本实现机制
- 操作系统的基本类型
- 操作系统的特征和功能
- 操作系统结构

操作系统的基本类型

- 批处理系统
- 分时系统
- 实时系统

- 混合型

操作系统引论

- 计算机硬件简介
- 操作系统的基本实现机制
- 操作系统的基本类型
- 操作系统的特征和功能
- 操作系统结构

操作系统的特征

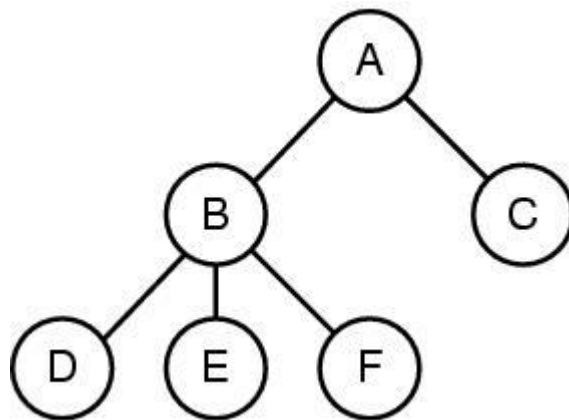
- 并发
- 共享
 - 互斥共享（打印机、变量）
 - 同时访问（宏观）
- 虚拟
 - 多道程序中的CPU
 - **SPOOLING**（外围设备同时联机操作）
 - 虚拟存储
- 异步性
 - 由于资源等因素的限制，程序的执行通常都不可能“一气呵成”，而是以“停停走走”的方式运行。

操作系统的功能

- 处理机管理
- 存储器管理（内存）
- 设备管理
- 文件管理
- 作业控制

处理机管理

- 进程（线程）控制；
- 进程（线程）同步；
- 进程通信；
- 进程（线程）调度。



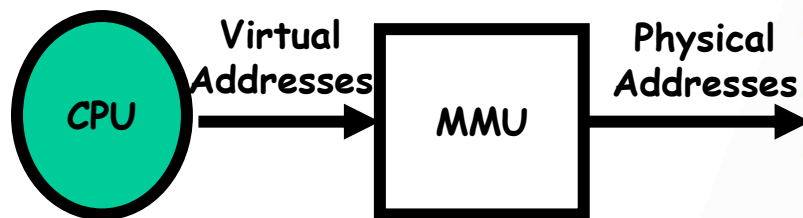
存储器管理

■ 任务：

- 为多道程序的并发提供良好的环境
- 便于用户使用存储器
- 提高存储器利用率
- 为尽量多的用户提供足够大的存储空间

■ 功能：

- 内存分配：静态和动态分配。
- 内存保护；
- 地址影射；
- 内存扩充。



设备管理

- 任务：
 - 为用户程序分配I/O设备
 - 完成用户程序请求的I/O操作
 - 提高CPU和I/O设备的利用率：中断；通道。
 - 改善人机界面
- 功能：
 - 缓冲管理；
 - 设备分配；
 - 设备处理；
 - 虚拟设备功能。

文件系统

- 文件存储空间的管理；
- 目录管理；
- 文件读、写管理；
- 文件保护；
- 向用户提供接口。

作业控制

- 作业调度；
- 作业控制。
 - 批量型作业
 - 终端型作业

操作系统应解决的基本问题

- 提供解决各种冲突（资源竞争引起）的策略
 - 处理机调度、进程调度、内存分配、设备分配等
- 协调并发活动的关系（提供流程控制和避免运行结果不确定性）
 - 进程之间的通信，同步与互斥
- 保证数据的一致性
 - 读写数据时，数据结构中的内容是否真实地记录了数据的实际情况
 - 在分布式处理时的共享数据的不同副本是否一致
- 实现数据的存取控制
 - 共享程度、隐私程度、安全程度的控制

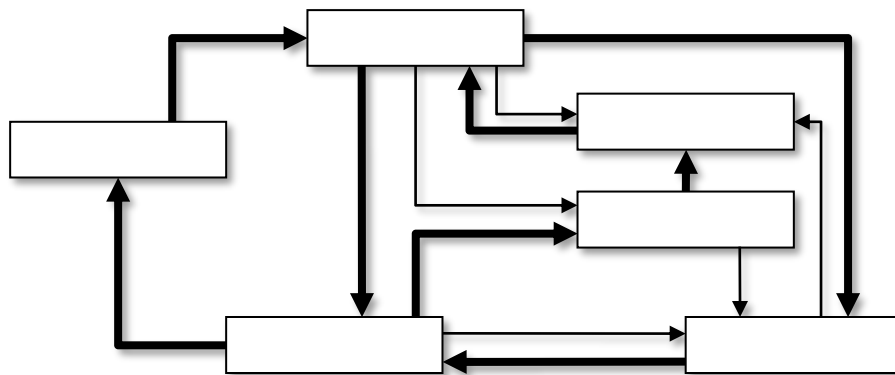
操作系统引论

- 计算机硬件简介
- 操作系统的基本实现机制
- 操作系统的基本类型
- 操作系统的特征和功能
- 操作系统结构

操作系统结构

- 模块接口
- 有序分层法
- 虚拟机
- 微内核结构
- 机制与策略分开

模块接口



- 内聚性
- 耦合性
- 优点：加速了操作系统的研制过程、增加了操作系统的灵活性、便于修改和维护。
- 缺点：接口定义困难、无序性。

有序分层法

- 自底向上法、自顶向下法
- 层次设置原则
 - 调用关系
 - 与硬件的关系
 - 与虚存的关系
 - 可扩充性
- 调用方式
 - 只能调用直接下层
 - 可以调用所有下层
 - 可以部分调用下层。

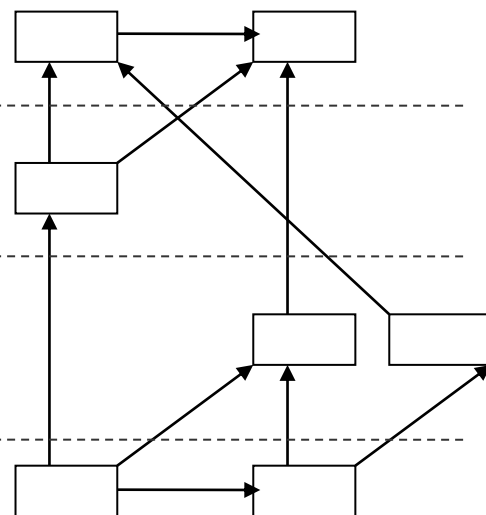
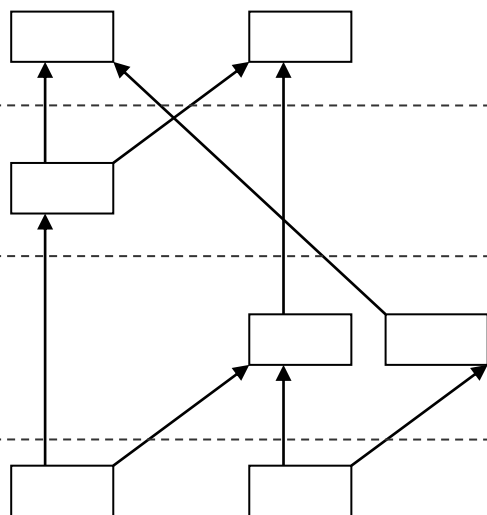
层次结构的层间关系

第3层

第2层

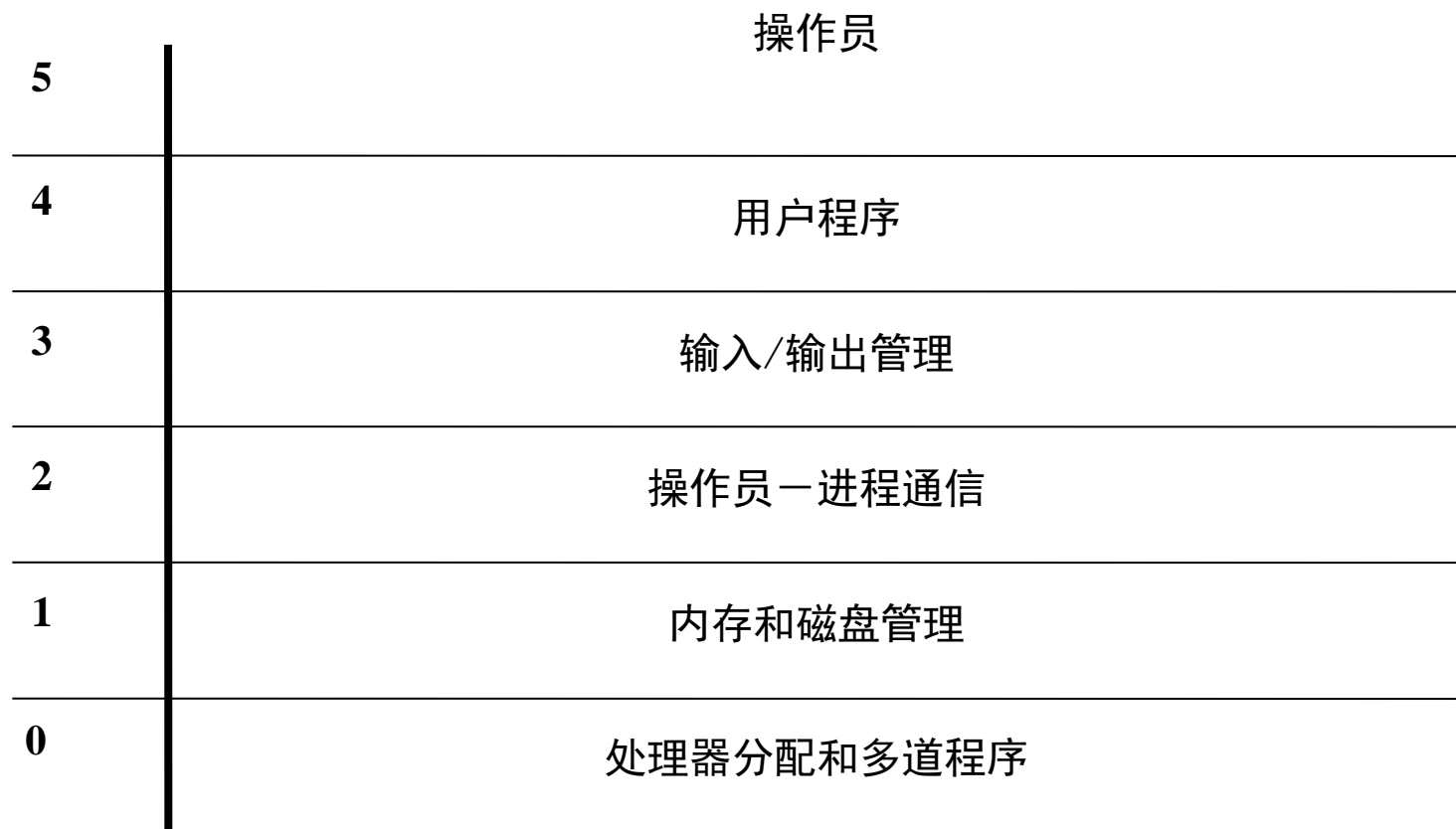
第1层

第0层



■ 层次结构

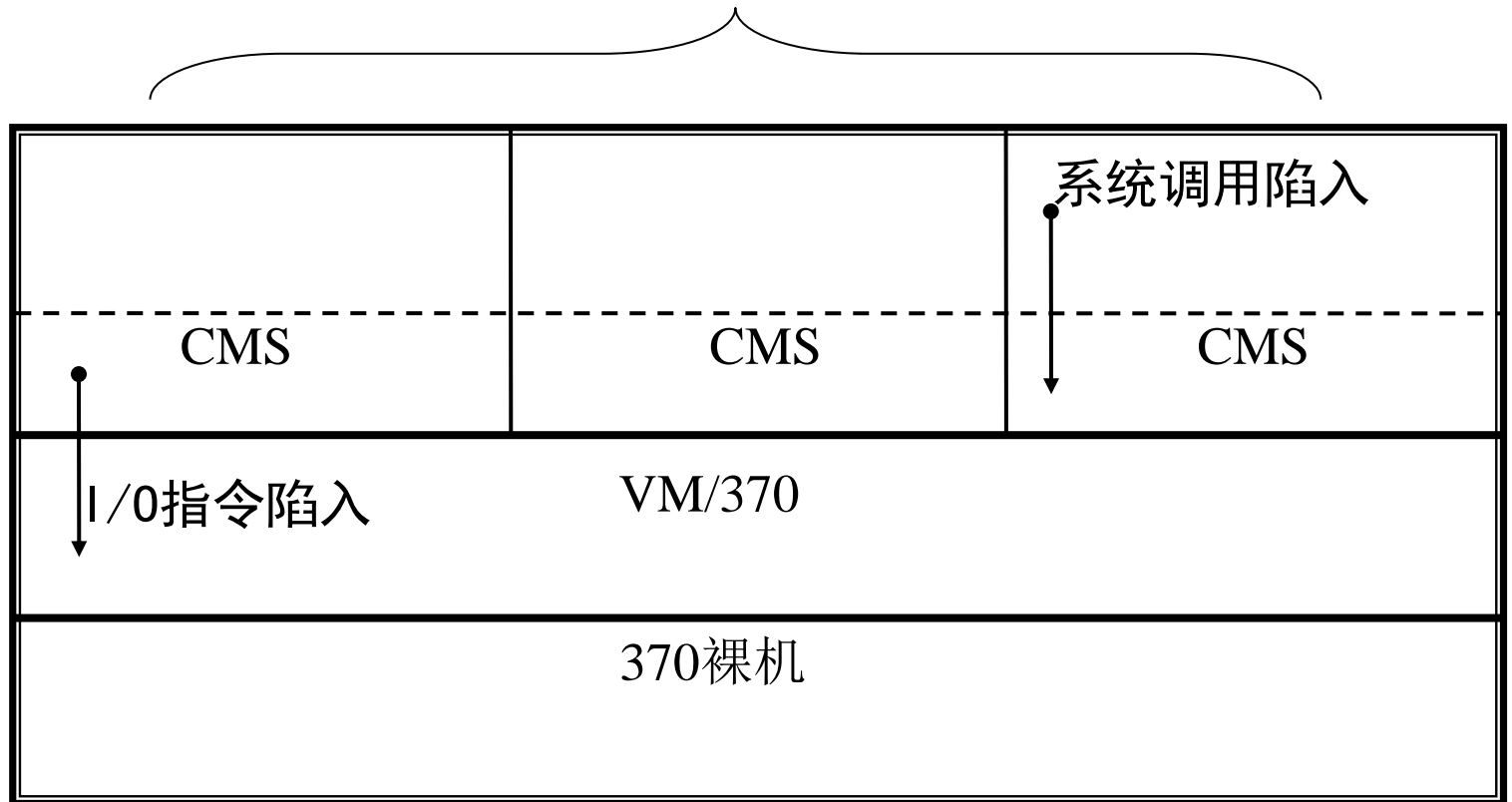
- 层次结构设计方法的核心思想
- 例如E.W.Dijkstra的THE系统：



虚拟机结构

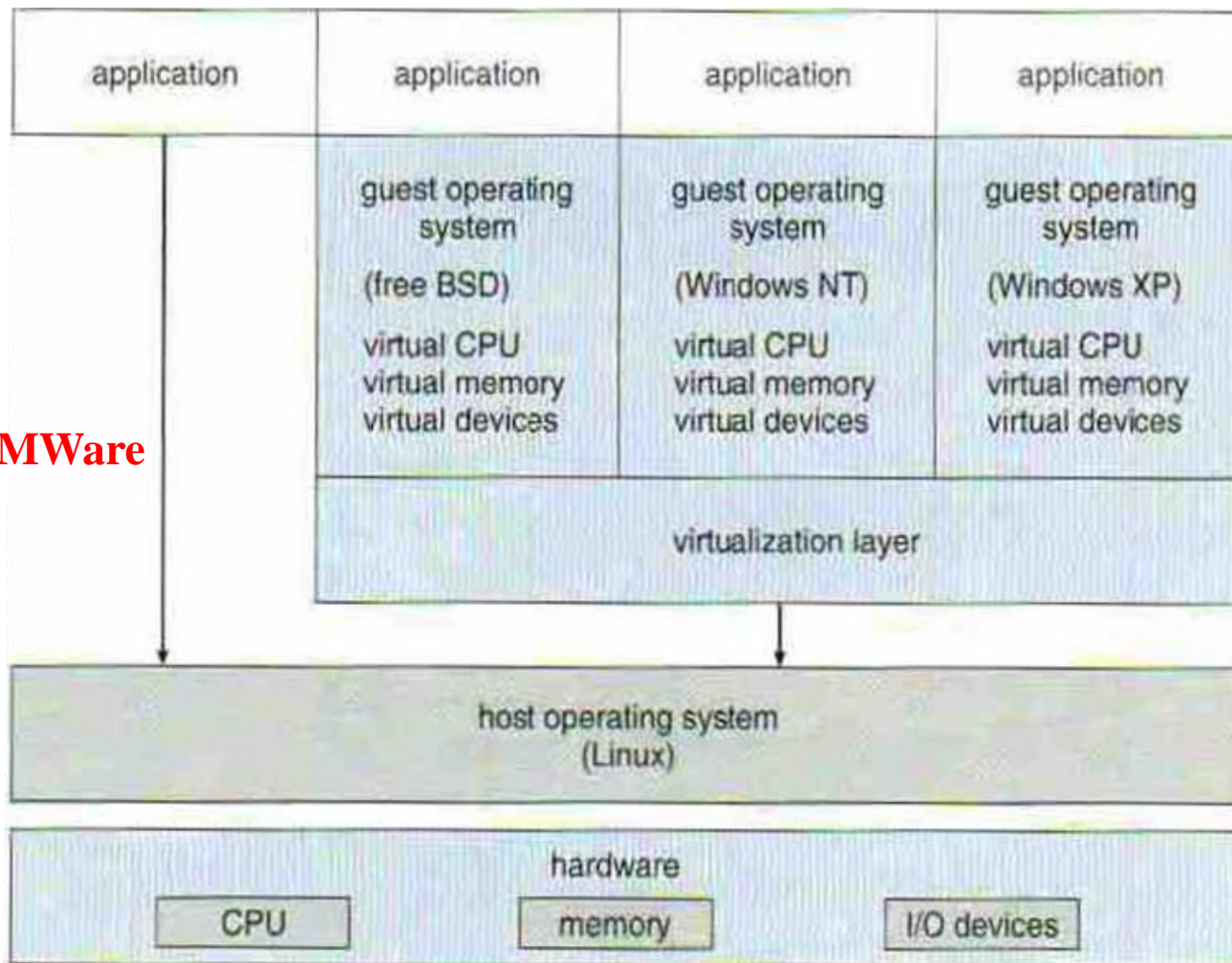
- 如IBM大型机上的系列操作系统
- 基本思想：系统应该提供 1) 多道程序能力 2) 一个比裸机更方便扩展界面的计算机。但是二者的实现应该相互独立
- 优缺点
 - 虚拟机概念可以实现完全保护
 - 用软件从硬件逐层扩展
 - 虚拟机方法把多道程序和扩充机器的功能完全分开了，使每一部分都简单、灵活，更易于维护。
 - 性能开销比较大

若干个370虚拟机



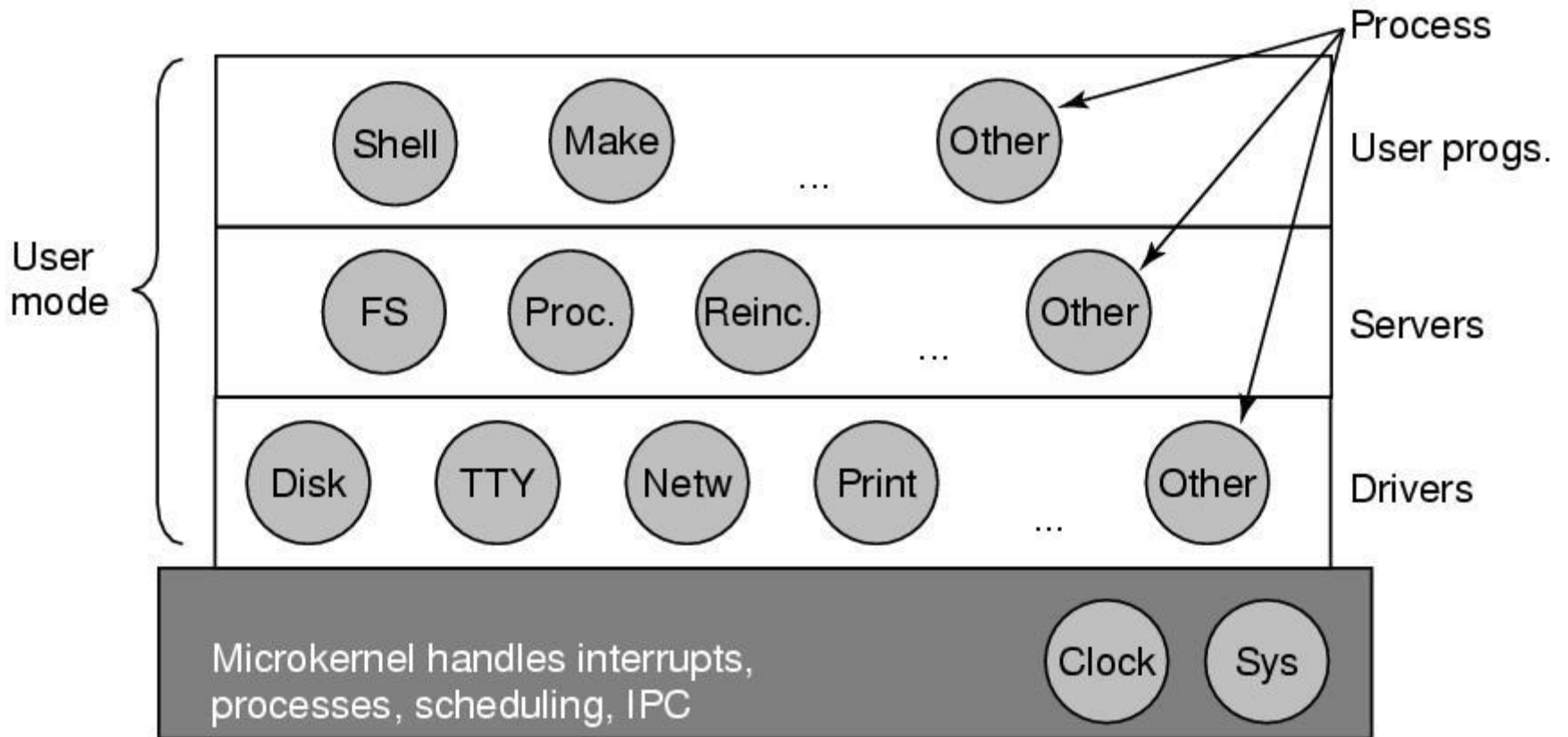
VMware 体系结构

建议：试用VMWare Workstation



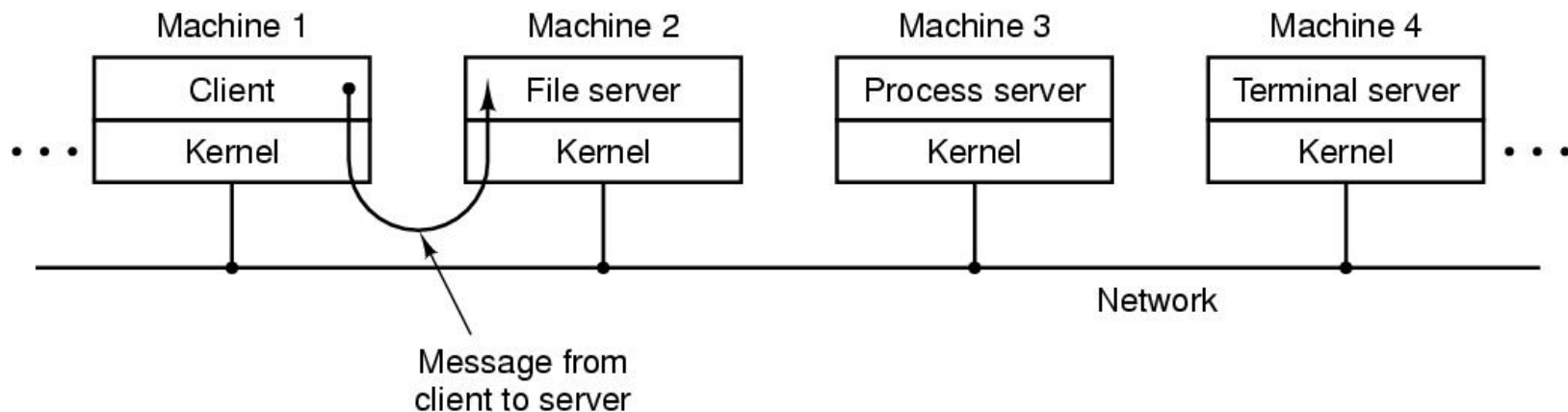
微内核结构

- 内核中只包括中断处理、进程通信（IPC）、基本调度等
- 文件系统、网络功能、内存管理、设备管理等作为服务在微内核上运行。
- 优点：
 - 内核易于实现、可移植性好、配置灵活、适应分布式环境（本地内核与远程内核对服务同样的支持）
- 缺点：
 - 速度较慢。（扩大内核减少切换；减少内核提高其他优点）



Structure of the MINIX 3 system.

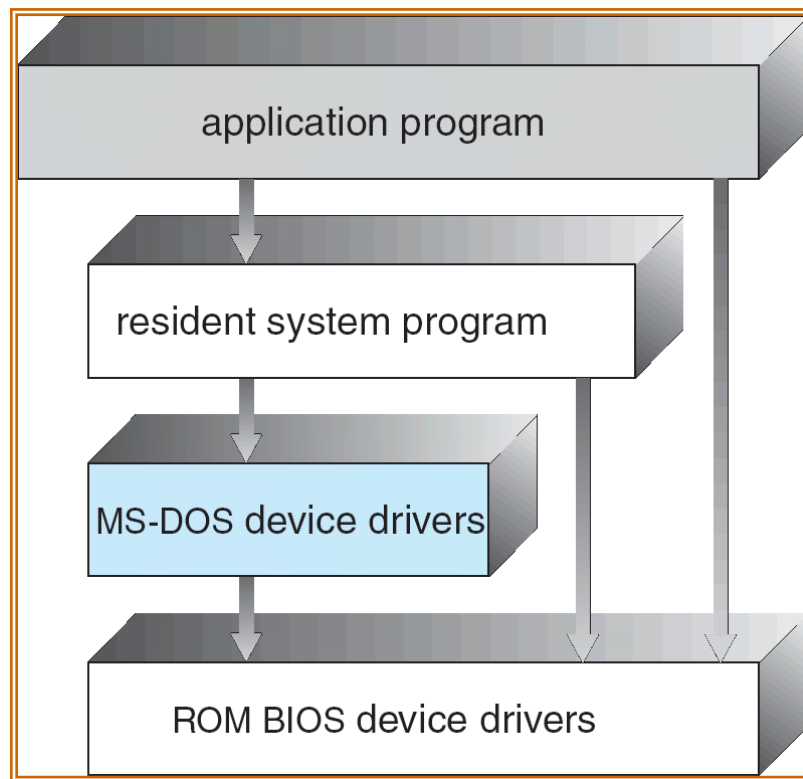
客户端-服务器模型



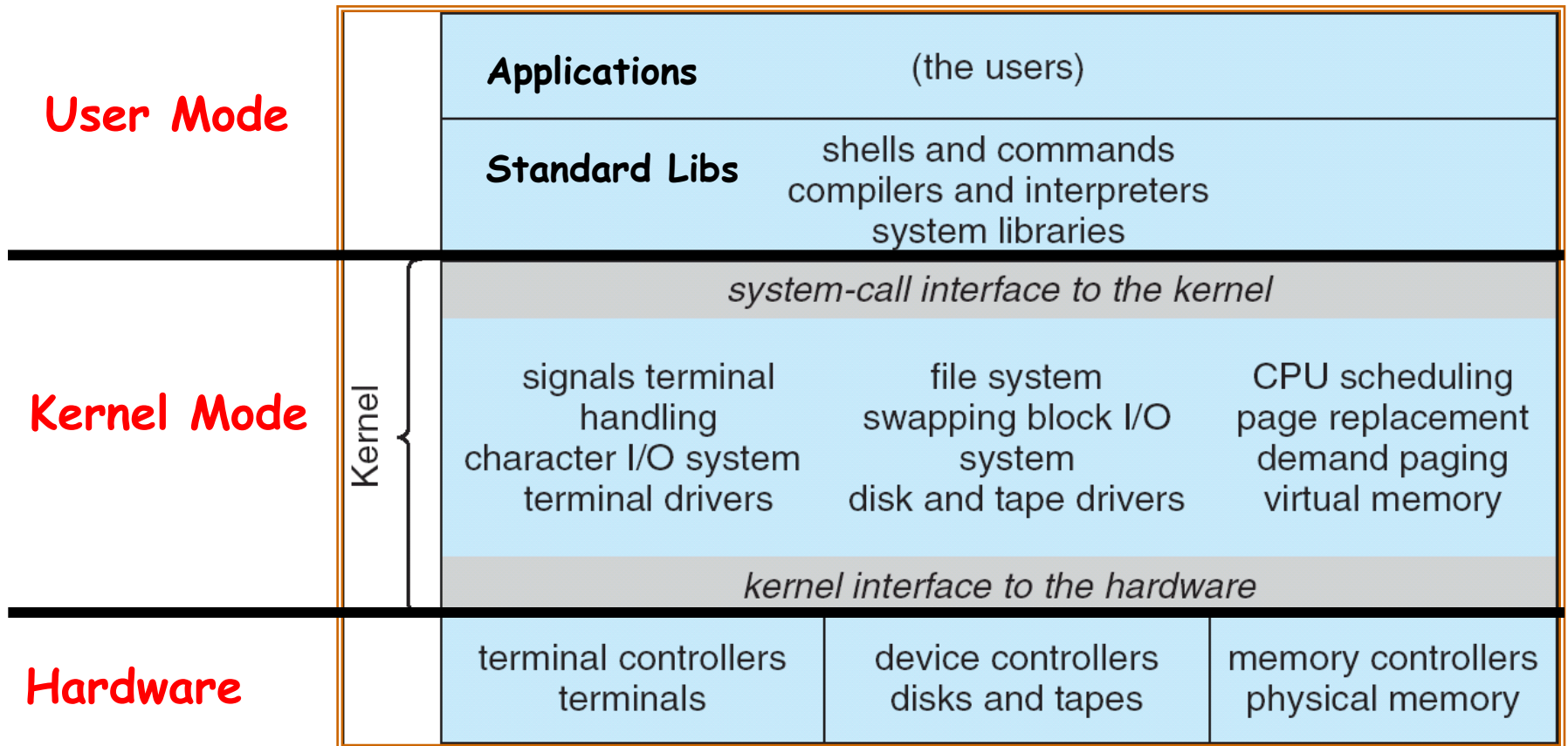
机制与策略分开

- 灵活
- 可扩展

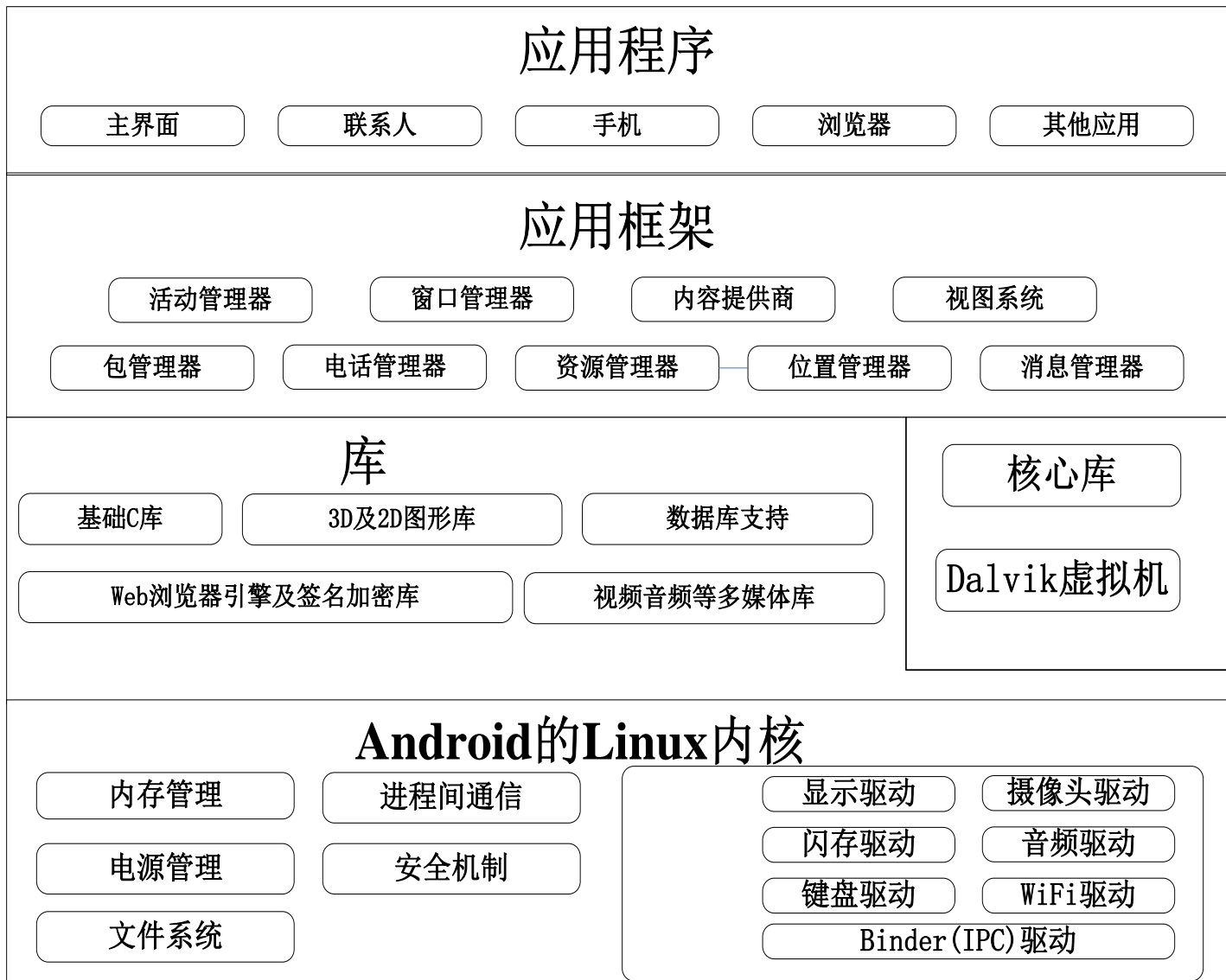
DOS的结构



UNIX System Structure

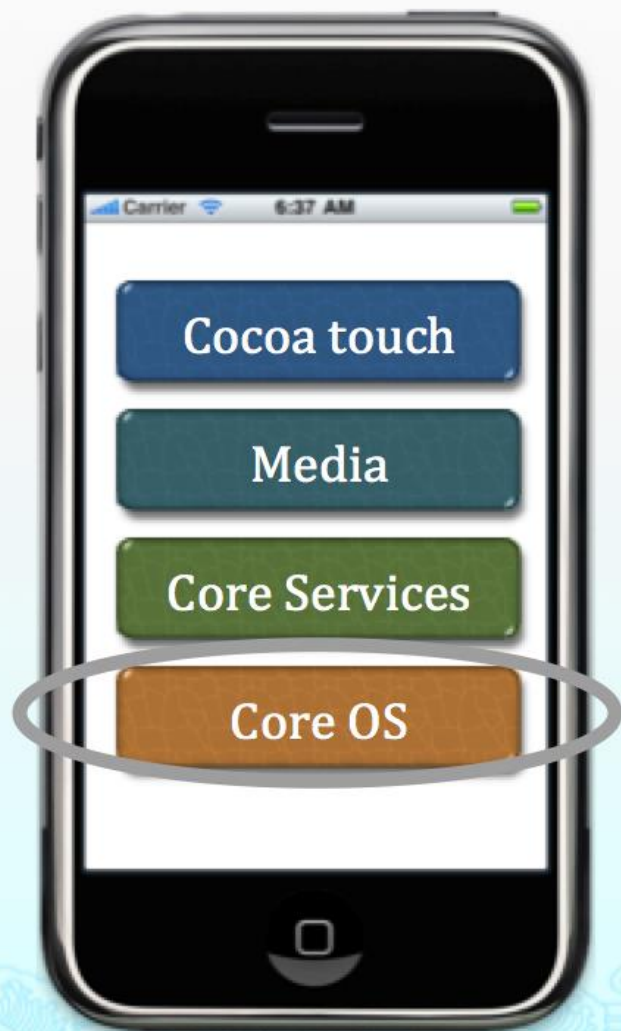


Android



iOS

- ◆ 分层架构，包含四层

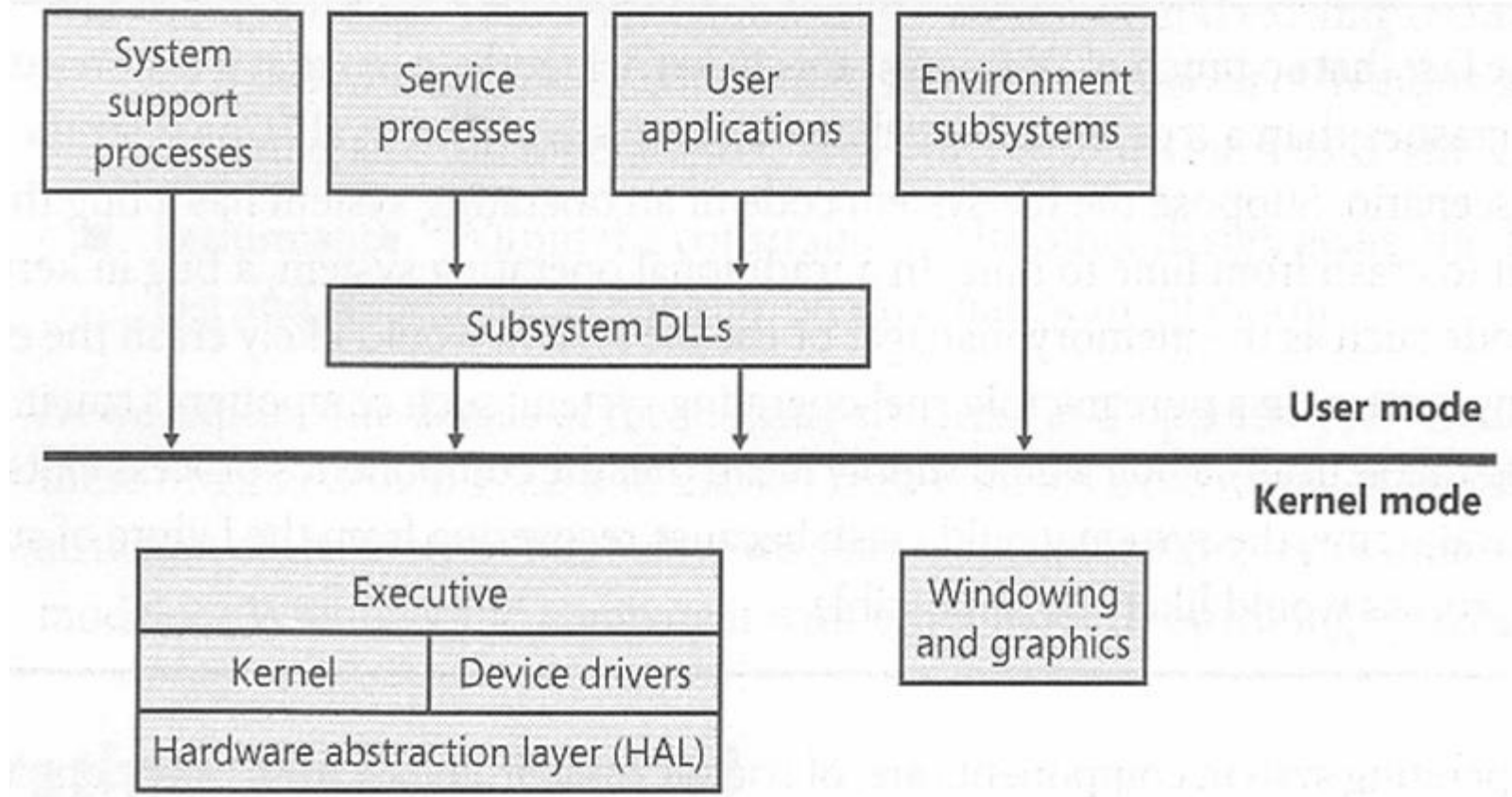


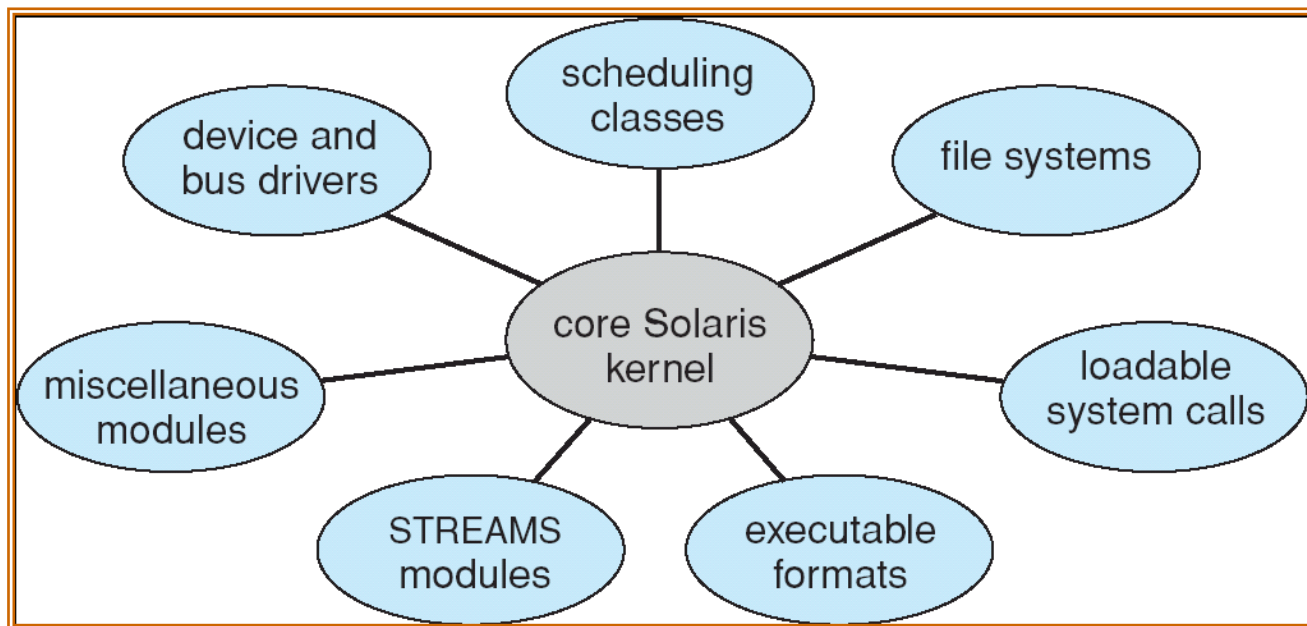
Core OS

1. 系统内核OSX
2. Mach 3.0 (CMU 微内核操作系统)
3. 能源管理
4. 文件系统
5. Sockets
6. 安全框架

C 语言

简化的Windows体系结构：类微内核





概念理解练习题

1. 操作系统的四项主要资源管理的功能是什么？各自面临哪些问题？
2. 什么是陷阱？与中断的区别是什么？什么是系统调用？
3. 如何建立一个可移植的操作系统？
4. 在设计操作系统时，一些设计指标是相互矛盾的，例如资源利用率、吞吐量、处理时间、健壮性等。请给出一对相互矛盾的设计实例。

<https://gitee.com/osbuaa/mos/>