1_概述

1.1_1 操作系统的概念、功能和目标

- 操作系统是系统资源的管理者
  - 提供的功能
    - 处理机管理
    - 存储器管理
    - 文件管理
    - 设备管理
  - 目标
    - 安全
    - 高效
- 向上层提供方便易用的服务
  - 封装思想：将抽象的硬件功能封装成简单易用的服务
  - GUI：图形化用户接口
  - 命令接口（联机命令接口|脱机命令接口）
    - 联机命令接口 = 交互式命令接口
      - 用户说一句，系统做一句
    - 脱机命令接口 = 批处理命令接口
      - 用户说一堆，系统做一堆
  - 程序接口
    - 程序中系统调用来使用程序接口
    - 用户不能直接使用，只能通过代码间接使用
- 是最接近硬件的一层软件
  - 实现对硬件机器的扩展

1.1_2 操作系统的特征

并发和共享是最基本的特征，二者互为存在条件

- 并发
  - 并发|并行
    - 并发：多个事件交替发生（宏观同时发生、微观交替进行）
    - 并行：多个事件同时发生
  - 单核|多核CPU
    - 单核CPU同一时刻只能执行一个程序，各个程序只能并发的执行
    - 多核CPU同一时刻可以执行多个程序，各个程序可以并行的执行
- 共享
  - 互斥共享方式|同时共享方式
    - 互斥共享方式：一个时间段内只允许一个进程访问该资源
    - 同时共享方式：允许一个时间段内由多个进程“同时”对它们进行访问
- 虚拟
  - 概念：把一个物理上的实体变为若干个逻辑上的对应物
  - 空分复用计数|时分复用计数
- 异步
  - 概念：在多道程序环境下，允许多个程序并发执行，但由于资源有限，进程的执行不是一贯到底的，而是走走停停的，以不可预知的速度向前推进。只有系统拥有并发性，才有可能导致异步性。

1.1_3 操作系统的发展与分类

- 手工操作阶段
  - 纸袋机实现
  - 缺点：
    - 用户独占全机
    - 人机速度矛盾导致资源利用率极低

- 批处理阶段
  - 单道批处理系统
    - 引入脱机输入/输出技术，并由监督程序负责控制作业的输入输出
    - 缺点：
      - 内存中仅有一道程序运行
      - CPU大量时间处于空闲等待I/O完成
  - 多道批处理系统
    - 操作系统开始出现
    - 优点：
      - 可以并发运行
      - 共享计算机资源
      - 资源利用率大幅度提升
    - 缺点：
      - 用户交互能力差
  - 分时操作系统
    - 以时间片为单位，轮流处理作业
    - 优点：
      - 可以实现人机交互问题
    - 缺点：
      - 不能优先处理紧急任务
  - 实时操作系统
    - 根据优先处理紧急任务
    - 硬实时系统|软实时系统
      - 硬实时系统：必须在严格的时间内完成处理
      - 软实时系统：可以偶尔犯错
  - 网络操作系统
  - 分布式操作系统
  - 个人计算机操作系统

1.1_4 操作系统的运行机制与体系结构

- 两类程序
  - 内核程序
    - 整个系统的管理者
  - 应用程序
- 两类指令
  - 特权指令
  - 非特权指令
- 两种处理器状态
  - 内核态 = 核心态 = 管态
    - 说明此时运行的是内核程序，此时可以执行特权指令
  - 用户态 = 目态
    - 说明此时运行的是应用程序，此时只能执行非特权指令
  - 扩展：CPU中有一个寄存器叫程序状态字寄存器（PSW），采用二进制方式区分
  - 如何变态：
    - 内核态->用户态：执行一条特权指令--修改PSW的标志位为用户态
    - 用户态->内核态：由“中断”引发，硬件自动完成变态过程
      - 但凡需要操作系统介入的地方，都会出发中断信号

1.1_5 中断和异常

- 因为可以实现中断，所以可以实现并发运行
- “中断”是让操作系统内核夺回CPU使用权的唯一途径
- 内中断|外中断
  - 内中断（也称为异常）：与当前执行的指令有关，中断信号来源于CPU内部
    - 陷阱、陷入（trap）
      - 陷入指令：程序主动将控制权还给操作系统内核

            - 非特权指令
        - 故障（fault）
            - 由错误条件引起，可被内核程序修复
        - 终止（abort）
            - 致命错误引起，内核无法修复
    - 外中断（也称为中断）：与当前执行的指令无关，中断信号来源于CPU外部
        - 时钟中断：实现计时功能，一定时间后，会给CPU发送中断信号
        - I/O中断：由输入/输出设备发送的中断信号
- 中断机制的基本原理
    - 不同的中断信号，需要不同的中断处理程序来处理
    - 查询“中断向量表”
    - 中断处理程序一定是内核程序

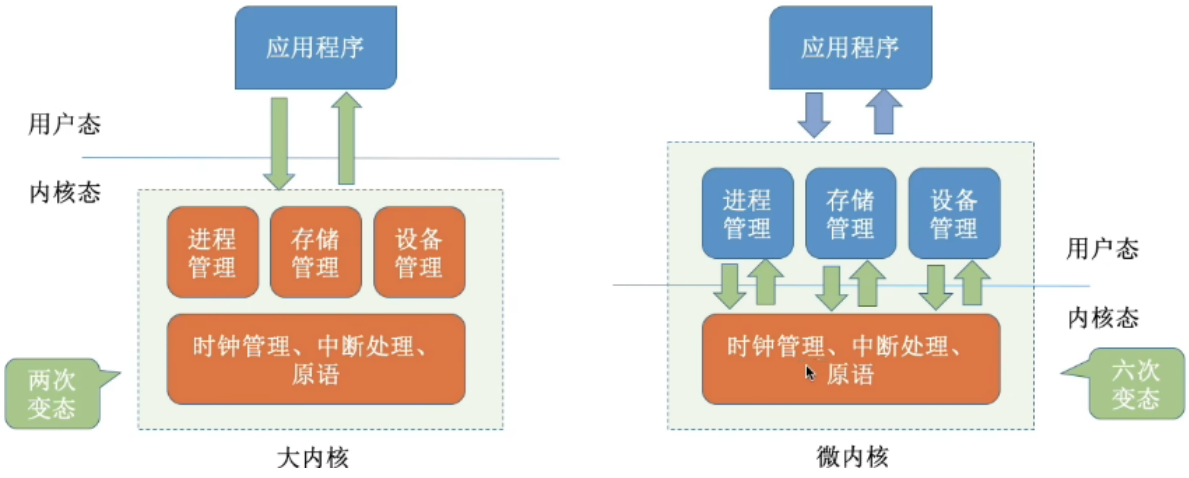
1.1_6 系统调用

- 概念：应用程序通过系统调用请求操作系统的服务。凡是与共享资源有关的操作（如存储分配、I/O操作、文件管理等），都必须通过系统调用的方式向操作系统提出服务请求。可以保证系统的稳定性和安全性。
- 系统调用和库函数的区别：
    - 系统调用是操作系统向上层提供的接口
    - 有的库函数是对系统调用的进一步封装
    - 当今编写的应用程序大多是通过高级语言提供的库函数间接地进行系统调用
- 系统调用过程
    - 传递系统调用参数->执行陷入指令（用户态）->执行相应的内请求核程序处理系统调用（核心态）->返回应用程序
        - 陷入指令是在用户态执行的，执行陷入指令后立即引发一个内中断，使CPU进入核心态

- 发出系统调用请求是在用户态，而对系统调用的相应处理在核心态
- 陷入指令=trap指令=访管指令

1.1_7 操作系统体系结构

- 大内核
  - 时钟管理（实现计时功能）
  - 中断处理
  - 原语（程序运行具有原子性，不可中断）
  - 进程管理
  - 存储器管理
  - 设备管理
- 微内核
  - 时钟管理（实现计时功能）
  - 中断处理
  - 原语（程序运行具有原子性，不可中断）
- 大内核|微内核
  - 区别：是否含有对系统资源进行管理的功能



  - 变态：CPU状态的转换
  - 频繁的变态会降低系统性能
  - 优点：

        - 大内核：高性能
        - 微内核：内核功能少，结构清晰，方便维护
    - 缺点：
        - 大内核：内核代码庞大，结构混乱，难以维护
        - 微内核：需要频繁在用户态和核心态之间切换，性能低
- 分层结构
    - 每层只能调用更低的相邻的接口提供的功能
    - 优点：
        - 便于调试和验证
        - 便于扩充和维护
    - 缺点：
        - 仅可调用相邻的低层，难以合理定义各层的边界
        - 效率低，不可跨层调用，系统执行时间长
- 模块化
    - 内核=主模块+可加载内核模块
        - 主模块：只负责核心部分，如进程调度，内存管理
        - 可加载内核模块：可动态加载新模块到内核
    - 优点：
        - 逻辑清晰，易于维护，可实现多模块同时开发
        - 可动态加载新的内核模块，增强OS的适应性
        - ==任何模块可直接调用其他模块，无需通过消息传递进行通信，效率高
    - 缺点：
        - 接口定义未必合理、实用
        - 调试和验证困难
- 外核
    - 内核负责进程调度，进程通信等功能，外核负责为用户进程分配未经抽象的硬件资源，且由外核负责保证资源使用安全
    - 优点：

        - 外核可直接给用户进程分配“不虚拟，不抽象”的硬件资源，使用户可以灵活的使用硬件资源
        - 减少了虚拟硬件资源的“映射层”，提升效率
    - 缺点：
        - 降低了系统的一致性
        - 使系统变复杂

1.1_8 操作系统的引导（boot）

- 主存：
    - RAM|ROM
        - RAM：断电后，数据会丢失
        - ROM（BIOS）：断电后不会丢失
            - 包含：ROM引导程序，即自举程序

1.1_9虚拟机

- 两类虚拟机
    - 直接运行在硬件之上
        - 虚拟机管理程序运行在内核态
        - 操作系统运行在用户态（虚拟内核空间）
    - 运行在宿主操作系统之上
        - 虚拟机管理程序运行在用户态

两类虚拟机管理程序（VMM）的对比

	第一类VMM	第二类VMM
对物理资源的控制权	直接运行在硬件之上，能直接控制和分配物理资源	运行在Host OS之上，依赖于Host OS为其分配物理资源
资源分配方式	在安装Guest OS时，VMM要在原本的硬盘上自行分配存储空间，类似于"外核"的分配方式，分配未经抽象的物理硬件	GuestOS 拥有自己的虚拟磁盘，该盘实际上是 Host OS 文件系统中的一个大文件。GuestOS分配到的内存是虚拟内存
性能	性能更好	性能更差，需要HostOS作为"中介"
可支持的虚拟机数量	更多，不需要和 Host OS 竞争资源，相同的硬件资源可以支持更多的虚拟机	更少，Host OS 本身需要使用物理资源，Host OS 上运行的其他进程也需要物理资源
虚拟机的可迁移性	更差	更好，只需导出虚拟机镜像文件即可迁移到另一台 HostOS 上，商业化应用更广泛
运行模式	第一类VMM运行在最高特权级（Ring 0），可以执行最高特权的指令。	第二类VMM部分运行在用户态、部分运行在内核态。GuestOS 发出的系统调用会被 VMM 截获，并转化为 VMM 对 HostOS 的系统调用