操作系统概述

## 操作系统的基本概念

### 操作系统的概念

#### 概念

操作系统，简称OS，是管理计算机硬件与软件资源的计算机程序

#### 计算机系统的构成

* 用户
* 应用程序
* 操作系统（OS）
* 硬件（裸机）

#### OS是一种系统软件

* 与硬件交互
* 对资源共享进行调度管理
* 解决并发操作处理中存在的协调问题
* 数据结构复杂，外部接口多样化，便于用户反复使用

#### 作为系统软件，OS做了哪些事

* 管理与配置内存
* 决定系统资源供需的优先次序
* 控制输入设备与输出设备
* 操作网络与管理文件系统等基本事务
* 提供一个让用户与系统交互的操作界面

### 操作系统的目标与功能

#### 目标

* 有效性——管理系统资源
  + 提高系统资源利用率
  + 提高系统的吞吐量
* 方便性——方便用户使用
* 可扩充性——作为扩充机器
* 开放性——作为扩充机器

#### 功能

* 作为就计算机系统资源的管理者
* 作为用户与计算机硬件系统之间的接口
* 实现了对计算机资源的抽象

#### 功能：作为就计算机系统资源的管理者

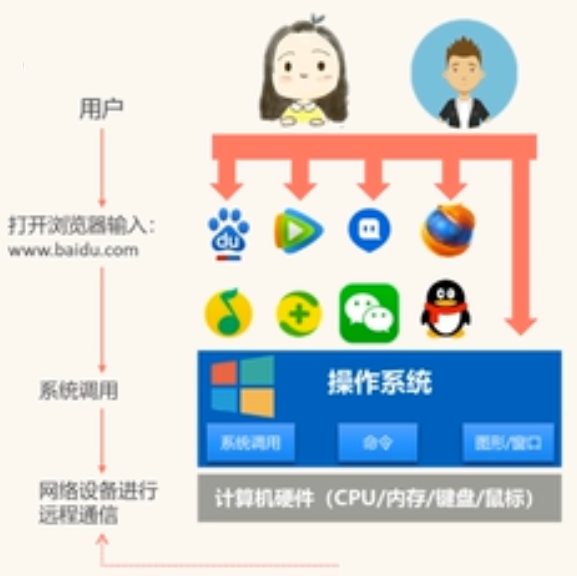
* 处理机管理
* 存储器管理
* I/O设备管理
* 文件管理

#### 功能：作为用户与计算机硬件系统之间的接口处理机管理

* 程序接口
* 命令接口
* GUI图形用户接口

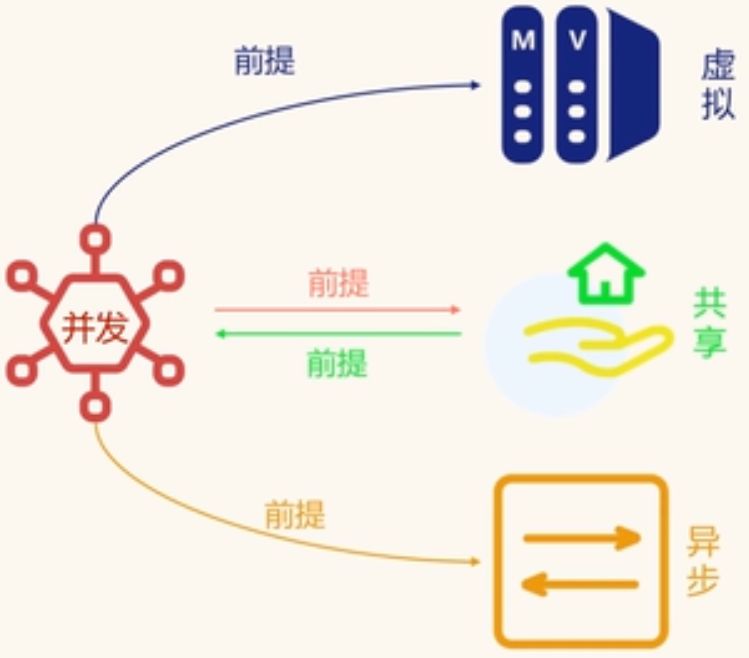
#### 功能：实现了对计算机资源的抽象

* 将具体的计算机硬件资源抽象成软件资源，方便用户使用
* 开放了简单的访问方式，隐藏了实现细节



### 操作系统的特征

* 并发
* 共享
* 虚拟
* 异步



#### 并发性

同一时间间隔内执行和调度多个程序的能力

* 宏观上，处理机同时执行多道程序
* 微观上，处理机在多道程序间高速切换（分时交替执行）
* 关注单个处理机同一时间段内处理任务数量的能力

#### 相似概念：并行

**并发**：同一时间间隔 (时间段）发生的事件数量

如：午餐时段，学校餐厅并发用餐人数为12人

**并行**：同一时刻（时间点）发生的事件数量

如：同一时间点 学校餐厅最多并行3人打餐

#### 共享性

即资源共享，系统中的资源供多个并发执行的应用程序共同使用

* 同时访问方式：同一时段允许多个程序同时访问共享资源
* 互斥共享方式：也叫独占式，允许多个程序在同一个共享资源上独立而不互相干扰的工作

#### 并发与共享互为存在条件

* 共享性要求OS中同时运行着多道程序。若只有单道程序正在运行，则不存在共享的可能
* 并发性难以避免的导致多道程序同时访问同一个资源，若多道程序无法共享部分资源（比如磁盘），则无法并发

#### 虚拟性

使用某种技术把一个物理实体变成多个逻辑上的对应物

* 时分复用技术（TDM）
  + 虚拟处理机技术：“四核八线程”
  + 虚拟设备技术：虚拟打印机
* 空分复用技术（SDM）
  + 虚拟磁盘技术：将一块硬盘虚拟出若干个卷
  + 虚拟存储器技术

#### 异步性

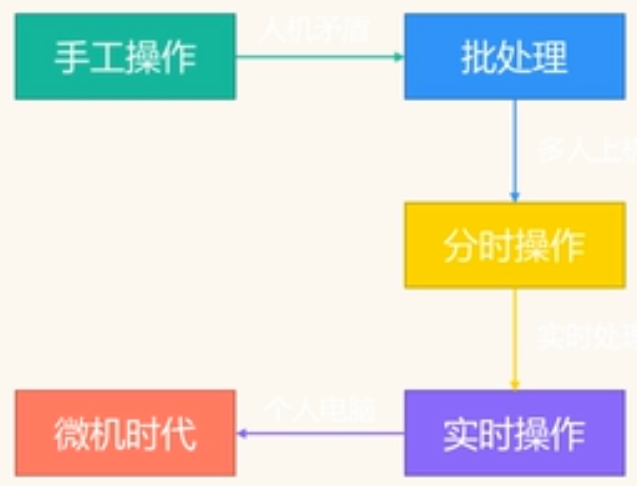
多道程序环境下，允许多个程序并发执行

单处理机环境下，多个程序分时交替执行

* 程序执行的不可预制性
  + 获得运行的时机
  + 因为什么而暂停
  + 每道程序需要多长时间
  + 不同程序的性能，比如计算多少，I/O多少
* 宏观上“一气呵成”，微观上“走走停停”

## 操作系统的发展与分类

* 手工操作阶段
  + 人工操作方式
  + 脱机输入/输出方式
* 批处理阶段
  + 单道批处理系统
  + 多道批处理系统
* 分时操作系统
* 实时操作系统
* 微机操作系统的发展



### 手工操作阶段：无操作系统的计算机系统

#### 人工操作方式

* 用户独占全机
* CPU等待人工操作

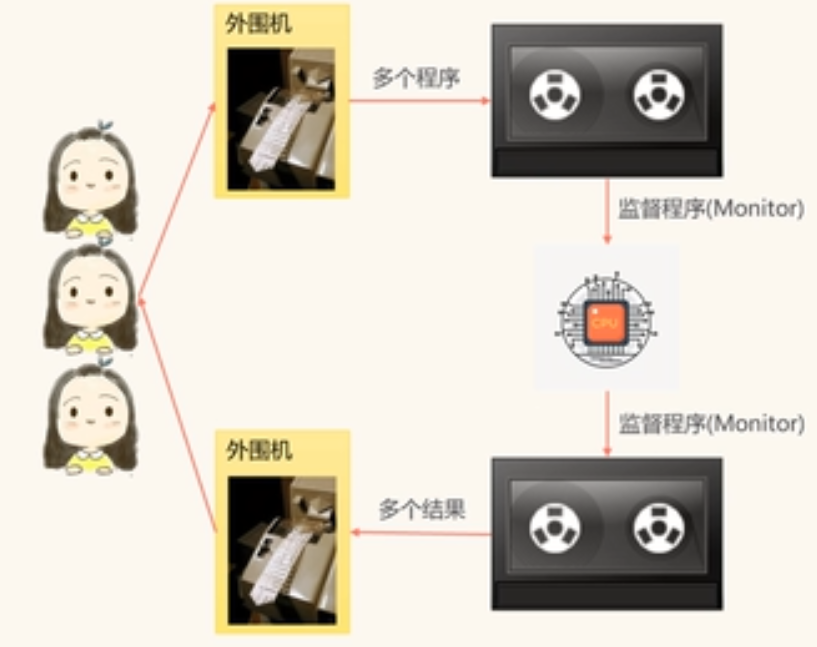
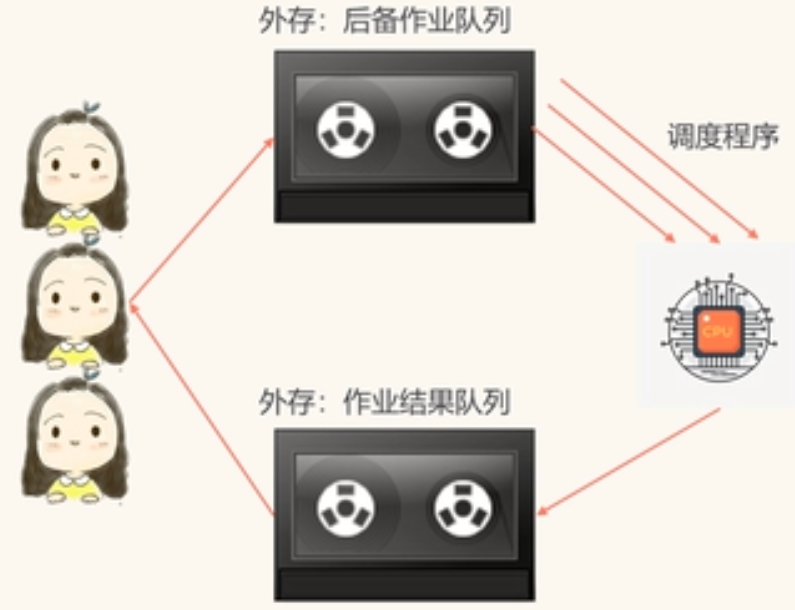
#### 脱机输入/输出方式

* 解决了人机矛盾
* 减少了CPU的空闲时间
* 提高了I/O速度
* 一次只能执行一个程序

### 批处理阶段：同时处理多道程序

#### 单道批处理系统（OS前身）

* 自动性
* 顺序性
* 单道性
* 内存中只有一道程序
* CPU需要等待I/O完成

#### 多道批处理系统

* 提高CPU的利用率
* 可提高内存和I/O设备利用率
* 增加系统吞吐量
* 平均周转时间长
* 无人机交互

#### 单道批处理系统与多道批处理系统对比

* 单道批处理系统：主要解决CPU、内存和I/O设备利用率不足的问题
* 多道批处理系统：主要解决I/O操作时CPU闲置问题

### 分时操作系统

一台主机连接多个带有显示器和键盘的终端，同时允许多个用户通过自己的终端，以交互方式使用计算机，共享主机中的资源。

#### 为什么需要分时系统？

* 人机交互
* 共享主机
* 便于用户上机

#### 关键问题

* 及时接收
* 及时处理（作业提前进入内存，并能够与用户交互）

#### 分时系统的特征

* 多路性：时间片轮转机制
* 独立性：用户彼此独立
* 及时性：用户能在短时间内获得响应
* 交互性：用户可以清求多种服务

#### 分时系统的缺点

作业/用户优先级相同，不能优先处理紧急任务

### 实时操作系统

系统能即时响应外部事件的请求，在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行

#### 应用需求

* 实时控制
* 实时信息处理

#### 实时任务

* 周期/非周期性实时任务（根据周期性）
* 硬/软实时任务（根据截止时间）

#### 与分时系统比较

* 多路性
* 独立性
* 及时性：以用户能接受的等待时间为准
* 交互性
* 可靠性：多级容错，保障系统和数据的安全

### 微机操作系统的发展

* 单用户单任务
  + CP/M、MS-DOS
  + 单用户多任务
  + Windows 1.0-XP
* 多用户多任务
  + UNIX OS:Solaris、Linux、Mac
  + MS-DOS:Windows 10

### 其它操作系统

* 网络操作系统
  + 资源共享
  + 远程通信
* 分布式操作系统
  + 分布性
  + 并行性

## 操作系统的运行环境

### 操作系统的运行机制

#### 基本概念

* 内核程序 —— 应用程序
* 核心态 —— 用户态
* 特权指令 —— 非特权指令



#### 时钟管理

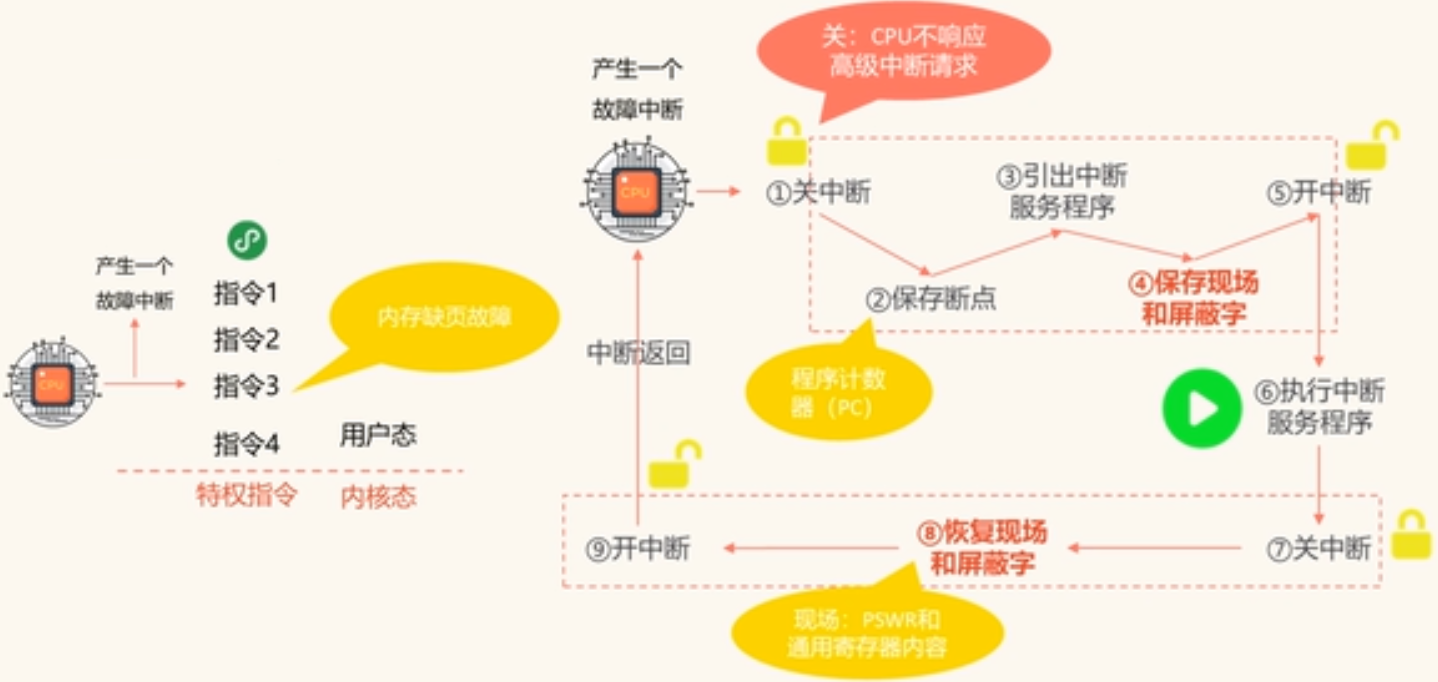
* 计时：提供系统时间
* 时钟中断：比如进程切换

#### 中断机制

* 提高多道程序环境下CPU利用率
* 外中断：中断信号来源于->外部设备
* 内中断：中断信号来源于->当前指令（内中断也叫异常/陷入/例外）
  + 陷阱/陷入(Trap)：由应用程序主动引发（由CPU产生）
  + 故障(fault)：由错误条件引发
  + 终止(abort): 由致命错误引发



#### 中断处理过程



#### 原语

* 由若干条指令组成
* 用来完成某个特定功能
* 执行过程不会被中断（原子性）

#### 系统数据结构

一般只涉及到对数据结构的操作，不涉及硬件

* 进程管理：作业控制块、进程控制块
* 存储器管理：存储器分配与回收
* 设备管理：缓冲区、设备控制块

#### 系统调用

系统调用的处理运行在核心态

* 由操作系统实现，给应用程序调用
* 是一套接口的集合
* 应用程序访问内核服务的方式

### 操作系统体系结构

#### 传统的操作系统结构（大内核）

* 第一代：无结构OS
  + 一系列过程（程序）的集合，过程间可以相互调用
  + 结构复杂且混乱，难以调试、阅读和维护
* 第二代：模块化结构OS: 模块-接口法OS
  + 基于“分解”和“模块化”原则
  + 按照功能战划分模块/子模块，规定模块间的接口
  + 模块独立性标准：高内聚、低耦合
* 第二代优点
  + 提高OS设计的正确性、可理解性和可维护性
  + 增强OS的适应性
  + 加速开发过程
* 第二代缺点
  + 模块接口设计难以扩展后续需求
  + 模块设计没有统一决策标准，导致模块接口设计不可靠
* 第三代：分层式结构OS
  + 有序分层法，自顶向下依次依赖
  + 设计时，自底向上：每一步建立在可靠的基础上
  + 优缺点：
    - 容易保证系统正确性
    - 容易扩充和维护
    - 自上而下的层次通信，导致系统效率降低

#### 微内核结构

* 足够小的内核，只实现OS核心功能
  + 与硬件处理紧密相关的部分，比如硬件处理、客户与服务器通信和其它基本功能
  + 一些较基本的功能
  + 客户和服务器之间通信(客户/服务器模式)
* 应用“机制与策略分离”原理
* 采用面向对象技术
* 微内核的OS的优点
  + 提高OS的可扩展性、可靠性、可移植性
  + 支持分布式系统
  + 融入了面向对象技术
* 微内核的OS的缺点
  + 相较早期OS,降低了一定的效率