# 操作系统

# **第一部分：概论**

#### **一、计算机系统：**

硬件系统、软件系统

#### **二、操作系统定义：（属于软件层）**

1. 组织和管理计算机系统中的硬件资源和软件资源（PCB、系统设备表等）
2. 有效性 （减少内存、硬盘空间的浪费等）
3. 合理（公平对待）
4. 方便（命令、图形界面、程序API接口）

#### **三、操作系统特征：**

1. 并发性
2. 共享性
3. 随机性

#### **四、研究操作系统的观点：**

1. 软件的观点
   1. 外在特性：接口
   2. 内在特性：与硬件交互

2、资源管理的观点

(1) 操作系统进行分配

3、进程的观点

(1) 侧重分析系统各部分的并行工作，管理各个进程及人物之间的管理（如：竞争资源、进程通信）

4、虚机器的观点

(1) 把所有对设备和文件的操作抽象成统一的打开、关闭、读、写，用户感觉不到底层的操作差异

5、服务提供者的观点

(1) 比裸机功能更强、服务质量更高、更方便灵活的虚拟机

#### **五、操作系统的功能：**

1. 进程管理
   1. 对中央处理器进行管理
   2. 为了描述多道程序技术并发而引入
   3. 进程管理的内容：进程控制、进程同步、进程间的通信、调度
2. 存储管理
   1. 任务：管理计算机内存的资源
   2. 功能：
      1. 内存的分配与回收
      2. 存储保护
      3. 内存扩充
3. 文件管理
   1. 任务：有效支持文件的存储、检索和修改，用户方便、安全的访问文件
   2. 功能：
      1. 文件存储空间的管理
      2. 目录管理
      3. 文件系统的安全性
4. 设备管理
   1. 含义：除了处理器和内存之外的所有输入输出设备的管理
   2. 功能：复杂外部设备的分配、启动和故障处理
5. 用户接口
   1. 用户观点：用户与计算机之间的接口
   2. 任务：提供用户一个使用系统的良好环境、有效组织自己的工作流程，并使整个系统高效地运行

**—————————————————————————**

#### **一、操作系统的体系结构：**

1. Windows 操作系统
   1. 内核
   2. 硬件抽象层HAL
   3. 执行体
   4. 系统进程和系统线程
2. Unix 操作系统
   1. 内核
   2. 系统层
   3. 应用层
3. Linux操作系统
   1. 内核
   2. Shell
   3. 文件系统
   4. 应用程序
4. Android操作系统
   1. 应用程序层
   2. 应用框架层
   3. 系统运行库层
   4. Linux内核层

**—————————————————————————**

#### **一、操作系统的发展：**

1. 手工阶段
2. 监控程序
3. 多道批处理
4. 分时与事实操作系统
5. UNIX通用操作系统
6. 个人计算机操作系统
7. Android操作系统

**—————————————————————————**

#### **操作系统的分类：**

基本类型

1. 批处理系统
   1. 特点：成批处理，用户不能干预自己作业的运行
   2. 目标：系统资源利用率，作业吞吐率高
   3. 分类：简单批处理和多道批处理
   4. 一般指令和特权指令：
      1. 操作系统的运行模式
      2. 处理器的状态
      3. 机器指令
      4. 系统调用
2. 分时系统
   1. 特点：
      1. 多路性
      2. 交互性
      3. 独占性
      4. 及时性
3. 实时系统
   1. 分类：硬实时、软实时
   2. 能力
      1. 实时时钟管理
      2. 过载防护
      3. 高可靠性

新类型

1. 个人操作系统
2. 网络操作系统
3. 分布式操作系统
4. 嵌入式操作系统
   1. 定义：安装在电器、电子和智能机械上，嵌入安装着各种微处理器或微控制芯片

**—————————————————————————**

#### **操作系统的设计过程：**

1. 功能设计
2. 算法设计
3. 结构设计

#### **操作系统的设计目标：**

1. 可靠性
2. 高效性
3. 易维护性
4. 可移植性
5. 安全性
6. 简明性

#### **操作系统的结构设计：**

1. 系统模块化
2. 模块标准化
3. 通信规范化

#### **操作系统的结构：**

1. 整体性结构
2. 层次性结构
3. 微内核（客户/服务器）结构

## 第二部分：运行环境 （操作系统的基础）

#### **处理器：**

1. 构成与基本工作原理
   1. 处理器中的寄存器
   2. 控制和状态寄存器
2. 特权指令和非特权指令
   1. 特权指令：只能有操作系统的指令
   2. 非特权指令
3. 处理器的工作状态
   1. 管态（高权限）和目态（普通状态）
   2. 处理器工作状态的转换
   3. 限制用户程序执行特权指令
4. 程序状态字
   1. CPU的工作状态码：管态和目态
   2. 条件码
   3. 中断屏蔽码

#### **计算机系统硬件部件：**

1. 存储器的类型
   1. 类型：
      1. 读写型存储器（RAM）
      2. 只读存储器 （ROM）
   2. 存储分块：
      1. 存储最小单位：位（bit）
      2. 最小编制单位：字节
      3. 分块（以单位分块）
2. 存储器的层次机构
   1. 容量、速度和成本的匹配
   2. 存储访问局部性原理
   3. 存储器保护（每个程序独立运行互不干扰）（界地址寄存器）
3. I/O部件
4. 时钟部件

#### **中断机制：**

1. 概念
   1. 中断与异常：
      1. 中断：处理器对系统中或系统外部发生的异步事件的响应
      2. 异步事件：指随机发生的事件
      3. 中断源
      4. 中断请求
      5. 中断处理程序
      6. 断点
      7. 中断返回
      8. 中断向量表
      9. 异常
   2. 中断与异常的分类：
      1. 典型的中断：
         1. 时钟中断
         2. 输入输出中断
         3. 控制台中断
         4. 硬件故障中断
      2. 典型的异常：
         1. 程序性中断
         2. 访问指令异常
2. 中断系统
   1. 中断请求的接收
   2. 中断响应
      1. 接收器接收中断信号
      2. 保护现场
      3. 分析中断向量
      4. 将处理器的PC值为中断程序的入口地址
      5. 调用中断处理程序
   3. 中断处理
      1. 接收和响应中断
      2. 保护现场
      3. 分析中断向量，调用中断处理程序
      4. 中断处理结束恢复现场，原有 程序继续执行
   4. 几种典型中断处理
      1. I/O中断
      2. 时钟中断
      3. 硬件故障中断
      4. 程序性中断
      5. 系统服务请求
3. 中断优先级、中断屏蔽与中断嵌套
   1. 多级中断与中断优先级
   2. 中断屏蔽
   3. 中断嵌套

#### **系统调用：**

1. 概念
   1. 操作系统提供给编程人员的唯一接口
   2. 特殊的调用过程，由特殊的机器指令实现，这条指令将系统转入管态
2. 系统调用（管态）和函数调用（目态）的区别
   1. 运行状态不同
   2. 状态的转换
   3. 返回问题
   4. 嵌套调用
3. 系统调用的分类
   1. 进程控制类
   2. 文件操作类
   3. 进程通信类
   4. 设备管理类
   5. 信息维护类
4. 系统调用的处理过程
   1. 进程控制类

## 第三部分：进程管理 （进程控制、同步、通信、调度、死锁）

#### 第一部分：进程与线程

#### **多道程序设计：**

1. 程序的执行顺序
   1. 顺序程序设计：程序是一个事件上严格按照次序前后相继的操作序列
   2. 特点:
      1. 顺序性
      2. 封闭性
      3. 程序执行结果确定性
      4. 程序执行结果的可在见性

#### **程序的并发执行：**

1. 含义：指两个或者两个以上程序在计算机系统中，同时处于开始执行且尚未 结束的状态
2. 特征：
   1. 在执行期间相互制约
   2. 程序与计算不再一一对应
   3. 并发程序的执行结果不可再现
   4. 程序的并行执行和程序的并发执行：程序的并发执行是宏观上的同时，微观是顺序，并行则是微观上是同时的

#### **多道程序设计：**

1. 环境特点概念：
   1. 就是允许多个程序同时进入内存并运行。根本目的是提高整个系统的效率
   2. 吞吐量：单位时间内系统所处理进程的道数，是衡量系统效率的尺度
2. 特点：
   1. 独立性
   2. 随机性
   3. 资源共享性
3. 缺陷
   1. 可能延长程序的执行时间
   2. 系统效率的提高有一定的限度

**—————————————————————————**

#### **进程：**

1. 定义：具有一定的独立功能的程序在某个数据集合上的一次运行活动，是系 统进行资源分配和调度的一个独立单位。分为：系统进程和用户进程

2、进程与程序的联系和区别

* 1. 程序是静态的，进程是动态的
  2. 二者是多对多的关系

3、可再入程序：同时被多个程序调用

1. 特征：
   1. 并发性
   2. 动态性
   3. 独立性
   4. 交往性
   5. 异步性
   6. 结构性

#### **进程的状态与转换：**

1. 三状态进程模型
   1. 运行状态
   2. 就绪状态
   3. 等待状态（阻塞状态）
2. 五状态进程模型
   1. 运行状态
   2. 就绪状态
   3. 堵塞状态
   4. 创建状态
   5. 结束状态
3. 七状态模型 + 上
   1. 就绪/挂起
   2. 阻塞/挂起

#### **进程控制块：**

1. 概念：便于系统控制和描述过程的活动过程，定义一个专门的数据结构，称 为PCB（进程存在的唯一标志）
2. PCB 的内容
   1. 调度信息（名字、进程号、地址信息、优先级、状态、消息队列指针等）
   2. 现场信息（进程运行情况，主要是CPU寄存器的信息）
3. 进程的组成
4. PCB 的组织
   1. 线性方式
   2. 索引方式
   3. 链接方式
5. 进程的队列
   1. 就绪队列 一个或多个
   2. 等待队列 多个
   3. 运行队列
6. 进程队列的组成
   1. 进程队列实际是PCB 的链接，链接分为：单向链表和双向链表
   2. 出队、入队、插队

#### **进程控制：**

1. 概念：对进程整个生命周期中各种状态之间的转换进行的控制
2. 进程控制原语：
   1. 创建原语：建立进程控制块PCB
   2. 撤销原语：撤销进程快PCB
   3. 阻塞原语
   4. 唤醒原语

**—————————————————————————**

#### **线程：**

1. 基本概念：
   1. 进程的属性：
      1. 一个可拥有资源的独立单位
      2. 可独立调度和分派的基本单位
   2. 引入线程的目的：
      1. 为了使多个进程并发执行
      2. 为了减少程序并发执行时候所付出的时空开销
   3. 什么是线程：
      1. 一个线程可以创建和撤销另一个线程；同一个进程中的多个线程可并发执行
   4. 线程的属性：
      1. 每个线程有一个唯一的标识和一张线程描述表
      2. 不同的线程可以执行相同的程序
      3. 同一个进程中的各个线程共享该进程的内存地址空间
      4. 多个线程可以并发执行
      5. 一个线程拥有生命周期，经历等待、就绪、运行等状态的变化
   5. 引入线程的好处：
      1. 创建一个新线程花费时间少
      2. 线程之间的切换花费时间少
      3. 线程之间的通信无需调用内核，不需要额外的通信机制
2. 进程和线程的比较：
   1. 调度：同进程中线程切换不引起进程切换，不同进程程线程才会引起进程切换
   2. 并发性：都可以并发
   3. 拥有资源：线程仅拥有隶属进程的资源；进程是拥有资源的独立单位
   4. 系统开销：线程低，进程高
3. 线程实现机制
   1. 用户级线程：仅存在用户空间
   2. 内核级线程
   3. 混合方式

**—————————————————————————**

#### **进程调度：**

1. 进程调度的主要功能：
   1. 记录系统中所有进程的执行情况
   2. 根据一定的调度算法，从就绪队列中选出一个进程，准备把处理器分配给它
   3. 分配处理器
2. 进程调度的时机
   1. 正在执行的进程运行完毕
   2. 正在执行的程序由某种错误终止运行
   3. 时间片完
   4. 正在执行的进程调用阻塞原语将自己阻塞起来
   5. 创建了新的进程
   6. 正在执行的进程调用了唤醒原语操作激活了等待资源的进程
3. 处理器的调度方式
   1. 非抢占方式
   2. 抢占式

#### **调度算法的设计原则：**

1. 进程行为：I/O密集型和计算密集型
2. 系统分类：批处理、交互式、实时处理
3. 调度算法的设计目标：
   1. 共同目标：资源利用率高、公平、平衡强行策略
   2. 批处理目标：平均周转时间短、系统吞吐量高、处理机利用效率好
   3. 分时系统目标：响应速度快、均衡性
   4. 实时系统目标：介质时间的保证、可预测性

#### **进程调度算法：**

1. 先来先服务
   1. 优点：实现简单
   2. 缺点：没有考虑进程优先级
2. 最短进程优先
   1. 同时可运行算法优先
3. 最短剩余时间优先算法
4. 最高响应比优先算法
   1. 优先调度响应比最大的进程
   2. 先来先服务、优先算法的折中
   3. 响应比 = 1 + 等待时间 / 运行时间
   4. 周转时间 = 完成时间 - 到达时间
5. 轮转算法
6. 最高优先级算法
7. 多级反馈队列算法

**—————————————————————————**

#### **系统内核：**

1. 概念：为了提高系统运行效率，把常用的基本操作集中安排，形成一个操作系统的核心
2. 内核的位置：常驻内存
3. 内核的功能：（通过执行原语操作来实现）
   1. 中段处理程序
   2. 进程同步与互斥
   3. 进程调度
   4. 进程控制与通信
   5. 存储管理
   6. 时钟管理

#### 第二部分：进程同步与互斥

#### **进程间相互作用：**

1. 相关进程：在逻辑上有某种联系的进程
2. 无关进程：在逻辑上没有联系的进程
3. 对时间有关的错误**：**
   1. 交替使用共享资源就会形成与时间相关的错误

#### **进程同步与互斥：**

1. 相关同步：一种直接的协同工作关系，一些进程相互合作共同完成一项任务
2. 进程互斥：多个进程竞争使用一个服务
3. 临界区：
   1. 临界资源：某种服务资源一次只允许一个进程使用
   2. 临界区：访问临界资源的那段代码
   3. 相关临界区：若干进程共享某一临界资源
      1. 使用规则：
         1. 有空让进，有效利用资源
         2. 无空等待，互斥进入
         3. 多种选一
         4. 有限等待，避免死等
         5. 让权等待，避免忙等

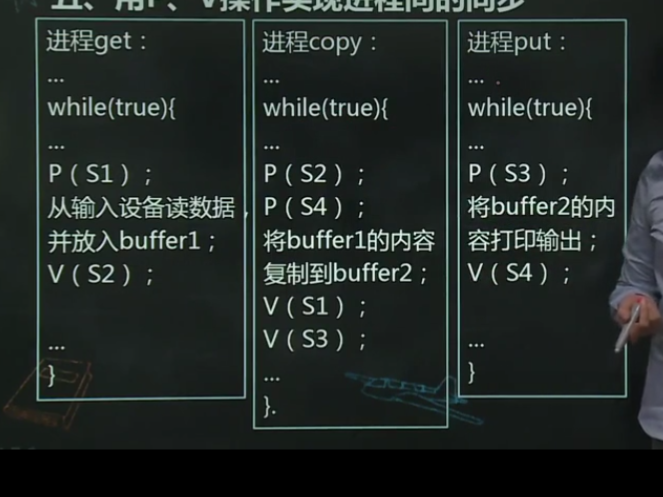
#### **信号量及P、V操作**

1. 信号量：
   1. 首先提出：1965荷兰学着Dijkstra
   2. P、V操作的使用：使用P(S) 和V()
2. P、V操作
   1. 执行一次P: 申请请求一个资源
   2. 执行一次V: 表示释放一个资源

#### **用P、V操作实现进程之间的互斥**

## 

## 

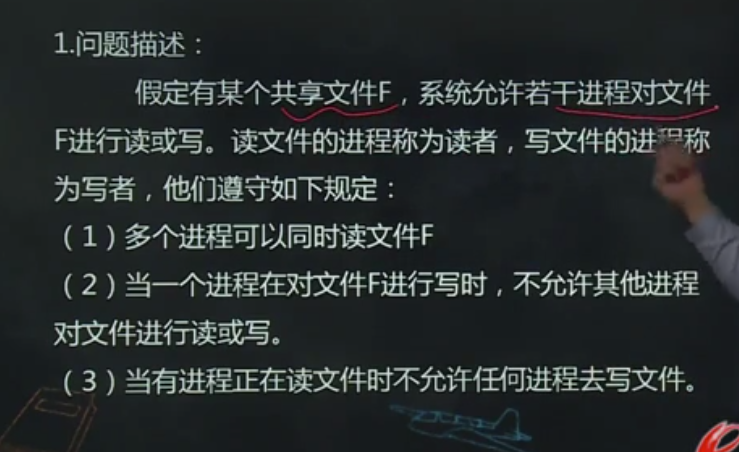


#### **信号量用P、V操作总结**

1. P、V操作必须成对出现
2. 互斥操作时候，P、V操作出现在同一进程
3. 同步操作时候，P、V操作出现在不同进程
4. 既有同步，又有互斥操作时候，同步信号量P操作在互斥信号量P操作之后， V操作顺序不限

#### **经典进程同步问题**

1. 简单生产者--消费者问题
2. 多个生产者--消费者问题
3. 读者--写者问题



## 

## 

#### **同步与互斥的综合应用**

**—————————————————————————**

#### **管程**

#### **进程通信**

1. 共享内存
2. 消息机制--消息缓冲--信箱
3. 管道通信

#### 第三部分：死锁

#### **定义：（随机产生的错误）**

1. 多道程序系统中，一组进程中的每一个进程均无限期地等待被该组进程中的 另一个进程所占用且永远不会释放的资源
2. 处于死锁状态的进程称为死锁进程

#### **产生的原因：**

1. 资源的概念：
   1. 永久性资源
   2. 临时性资源
2. 产生的原因：
   1. 竞争资源
   2. 进程推进顺序不合理
3. 产生死锁的四个必要条件
   1. 互斥条件
   2. 不可剥夺条件
   3. 请求和保持条件
   4. 循环等待条件
4. 解决死锁的方法
   1. 预防死锁
   2. 避免死锁
   3. 检测与解除死锁
   4. 忽略死锁

#### **死锁的预防：**

1. 基本思想：防患与未然
2. 具体做法：破坏产生死锁的四个必要条件之一

## 第四部分：存储管理 （内存的管理）

## 第五部分：文件系统 （文件存储空间的管理）

## 第六部分：设备管理 （I/O设备的管理）