

# 操作系统引论

2021年11月1日 20:13

## 1、操作系统的目标和作用

1. 操作系统的目标
  - a. 方便性
  - b. 有效性
  - c. 可扩充性
  - d. 开放性
2. 操作系统的作用
  - a. 作为用户与计算机硬件之间的接口
  - b. 作为计算机系统资源的管理者
  - c. 实现了对计算机系统资源的抽象
3. 推动操作系统发展的主要动力
  - a. 不断提高计算机资源利用率
  - b. 方便用户
  - c. 器件的不断更新换代
  - d. 计算机体系结构的不断发展
  - e. 不断提出新的应用需求

## 2、操作系统的发展过程

1. 未配置操作系统的计算机系统
  - a. 人工操作方式
    - i. 用户独占全机
    - ii. CPU等待人工操作

人工操作方式严重降低计算机资源的利用率（人机矛盾）
  - b. 脱机IO方式
    - i. 减少了CPU空闲时间
    - ii. 提高了IO速度

数据的输入和输出在外围机控制下完成
2. 单道批处理系统SBPS
  - a. 处理过程
    - i. 以脱机方式将一批作业输入到磁带上
    - ii. 将磁带上的第一个程序装入内存，将控制权交由程序
    - iii. 一个接一个程序运行，直到所有作业全部完成
  - b. 缺点
    - i. 内存中只存在一道程序
    - ii. 等待IO使得CPU利用率低
3. 多道批处理系统MBPS
  - a. 基本概念
    - i. 用户提交的作业在外存上形成后备队列

- ii. 作业调度程序按一定算法选择若干进程调入内存
    - iii. 内存中多道程序共享计算机资源
    - iv. 运行中的程序因IO操作暂停时调度另一道程序运行
    - v. 保持CPU忙碌
  - b. 优缺点
    - i. 资源利用率高
    - ii. 系统吞吐量大
    - iii. 平均周转时间长
    - iv. 无交互能力
  - c. 需要解决的问题
    - i. 处理机争用
    - ii. 内存分配与保护
    - iii. IO设备分配
    - iv. 文件组织与管理
    - v. 作业管理
    - vi. 用户与系统的接口
4. 分时系统TSS
- a. 分式系统的引入
    - i. 人机交互
    - ii. 共享主机
  - b. 分时系统实现的关键问题
    - i. 及时接收
    - ii. 及时处理
  - c. 分时系统的特征
    - i. 多路性
    - ii. 独立性
    - iii. 及时性
    - iv. 交互性
5. 实时系统RTS
- a. 实时系统的类型
  - b. 实时任务的类型
    - i. 周期性实时任务/非周期性实时任务
    - ii. 硬实时任务/软实时任务
  - c. 实时系统与分时系统特征的比较
    - i. 多路性
    - ii. 独立性
    - iii. 及时性
    - iv. 交互性
    - v. 可靠性
6. 微机操作系统的发展
- a. 单用户单任务操作系统
  - b. 单用户多任务操作系统

- c. 多用户多任务操作系统

### 3、操作系统的基本特性

1. 并发
  - a. 并行与并发
  - b. 进程
2. 共享
  - a. 互斥共享方式
  - b. 同时访问方式
3. 虚拟
  - a. 时分复用技术
  - b. 空分复用技术
4. 异步

### 4、操作系统的主要功能

1. 处理机管理
  - a. 进程控制
  - b. 进程同步
  - c. 进程通信
  - d. 调度
2. 存储器管理
  - a. 内存分配
  - b. 内存保护
  - c. 地址映射
  - d. 内存扩充
3. 设备管理
  - a. 缓冲管理
  - b. 设备分配
  - c. 设备处理
4. 文件管理
  - a. 文件存储空间管理
  - b. 目录管理
  - c. 文件读写保护与管理
5. 操作系统与用户之间的接口
  - a. 用户接口
  - b. 程序接口
6. 现代操作系统的新功能

### 5、OS结构设计

1. 传统OS结构
  - a. 无结构操作系统
  - b. 模块化结构操作系统
    - i. 模块化程序设计技术
    - ii. 模块独立性

- 1) 内聚性
      - 2) 耦合度
    - iii. 模块接口法的优点
      - 1) 提高正确性, 可理解性, 可谓互性
      - 2) 增强可适应性
      - 3) 加速开发过程
    - iv. 模块接口法的缺点
  - c. 分层次结构操作系统
    - 各层之间只存在着单向的依赖关系, 高层仅依赖紧邻的低层
    - i. 优点
      - 1) 易保证正确性
      - 2) 易扩充, 易维护
    - ii. 缺点
      - 1) 系统效率降低
2. 客户/服务器模式 (C/S模式)
- a. 客户/服务器模式的组成
    - i. 客户机
    - ii. 服务器
    - iii. 网络系统
  - b. 客户/服务器模式的交互
    - i. 客户机发送请求
    - ii. 服务器接收请求
    - iii. 服务器回送消息
    - iv. 客户机接收消息
  - c. 客户/服务器模式的优点
    - i. 数据的分布处理、存储
    - ii. 便于集中管理
    - iii. 灵活性和可扩充性
    - iv. 易于改编应用软件
  - d. 客户/服务器模式的缺点
    - i. 不可靠
    - ii. 瓶颈问题
3. 面向对象程序设计 (OOP)
- a. 基本概念
    - i. 对象
    - ii. 类
    - iii. 继承
  - b. OOP技术的优点
    - i. 通过重用提高质量与生产率
    - ii. 更好的易修改、易扩展性
    - iii. 易于保证正确性、可靠性
4. 微内核OS结构

- a. 微内核OS结构的基本概念
  - i. 足够小的内核
  - ii. 基于C/S模式
  - iii. 应用“机制与策略分离”原理
  - iv. 采用OOP技术
- b. 微内核OS结构的基本功能
  - i. 进程（线程）管理
  - ii. 低级存储器管理
  - iii. 中断与陷入处理
- c. 微内核OS结构的优点
  - i. 提高可扩展性
  - ii. 增强可靠性
  - iii. 可移植性强
  - iv. 提供了对分布式系统的支持
  - v. 融入了OOP技术
- d. 微内核OS结构的缺点
  - i. 运行效率降低
    - 1) 多次上下文切换