操作系统概述

2024年5月17日 14:02

第一章 操作系统概述

第一章分为三小节

- 操作系统基本概念
- 操作系统的发展和分类
- 操作系统的运行环境

什么是操作系统?

- 操作系统的概念
  - 操作系统（OperationSystem），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）：应用程序都安装在操作系统
    - 硬件（裸机）：CPU、内存、键盘、鼠标
  - 操作系统是一个软件，负责什么?
    - 与硬件交互
    - 对资源共享(硬件资源，如CPU、内存等)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
  - 操作系统是系统软件，做了什么?
    - 管理和配置内存
    - 决定系统资源的供需的优先次序
    - 控制输入设备和输出设备
    - 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?

- 管理系统资源

  有效性、提高系统资源利用率、提高系统的吞吐量
- 方便用户使用

  方便性
- 作为扩充机器

  可扩充性、开放性

操作系统有哪些功能?

- 作为计算机系统资源的管理者
  - 处理机管理
    - 进程控制、进程同步、进程通信、调度
  - 储存器管理
    - 内存分配、内存保护、地址映射、内存扩充
  - IO设备管理
    - 缓冲管理、设备分配、设备处理
  - 文件管理
    - 文件储存空间的管理、目录管理、文件的读写管理和保护
- 作为用户与计算机硬件系统的接口
  - 程序接口
  - 命令接口
  - GUI图像化接口
- 实现对计算机资源的抽象
  - 将具体硬件资源抽象为软件资源，方便用户使用
  - 开放简单的访问方式，隐藏实现细节

操作系统的特征?

- OS的并发性
  - 同一时间间隔内执行和调度多个程序的能力
  - 相对概念：并行：同一时刻发生的事件数量，一个处理器同一时刻只能做一件事，有物理极限
  - 宏观上，处理机同时执行多个程序
  - 微观上，处理机在多道程序间告诉切换（时间片）
  - 关注单个处理机同一时间段内处理任务数量的能力
- OS的共享性
  - 即资源共享，系统中的资源公多个并发指向的应用程序共同使用
  - 同时访问方式：同一时刻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互不干扰的工作
  - 如：共享打印机、音频设备、视频设备等
- 并发和共享互为存在条件
  - 共享性要求OS同时运行多个程序，如果单程序运行不存在共享
  - 并发性难以避免的导致多个程序同时访问同一资源，如果多程序无法共享部分资源（磁盘），就无法并发
- OS的虚拟技术
  - 使用某种技术将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，如（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：将磁盘分为若干卷
        - 虚拟储存器技术
- OS的异步性
    - 多程序环境下，允许程序并发执行。单处理机下，多程序分时交替执行
    - 异步原因：程序执行的不可预知性
        - 获得运行的时机、因何暂停、每个程序的执行时间、不同程序的性能

操作系统的发展?

- 手工操作阶段
    - 手工操作方式：用户独占全机，CPU等待人工操作
    - 脱机输入/输出方式：纸带打印机
        - 解决人机矛盾、减少CPU空闲时间、提高IO速度
        - 一次只能执行一个程序
- 批处理阶段
    - 单道批处理系统（OS前身）
        - 自动、顺序、单道
        - 内存中只有一个程序、CPU要等到IO完成
    - 多道批处理系统（内外存划分为输入执行输出三个执行器并行执行）
        - 提高CPU利用、提高内存IP设备利用、增加系统吞吐量
        - 平均周转时间长、无人机交互
- 分时操作系统
    - 一台主机连接多个显示器和键盘终端，运行多个用户通过终端交互使用计算机，共享主机资源
    - 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
        - 及时接收、及时处理（作业提前进入内存，能与用户交互）
    - 特征
        - 多路性、独立性、及时性、交互性、
    - 但是用户的优先级相同，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，在规定时间内完成对时间的处理，并控制所有实时任务协调一致地运行
    - 需求：实时控制、实时信息处理
    - 实时任务：
        - 周期/非周期实时任务
        - 硬/软实时任务
    - 多路性、独立性、及时性、交互性
    - 但是对可靠性要求高。可靠性：多级容错、保证系统和数据安全

- 微机操作系统的发展
  - 单用户多任务->多用户多任务
  - UNIX发展到Linux和MacOS，Linux发展到Windows
  - 其他操作系统：
    - 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

操作系统的运行机制?

- 基础概念
  - 用户空间
    - 非内核功能、应用程序
  - 内核空间
    - 进程管理、储存器服务器、文件管理服务器
  - 区分概念：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：如进程切换
- 中断机制
  - 提高多道程序环境下CPU的利用率，是由硬件实现
  - 外中断：中断信号来源于外部设备
  - 内中断（异常/陷入/例外）：中断信号来源于当前指令
    - 陷阱/陷入：应用程序主动引发，再由CPU产生（主要是系统调用时会使用）
    - 故障：错误条件引发，可预知的故障中断，有对应的熔断机制
    - 终止：致命错误引发，未预知的故障或者系统无法自己处理的故障
  - 中断处理过程
    - 产生中断
      - 关中断、保存断点、引出中断服务程序、保存现场和屏蔽字、开中断
      - 开关中断类似加锁，不同的中断有不同的优先级。开关中断让cpu不响应高级中断请求
    - 执行中断
      - 执行中断服务程序
    - 处理中断
      - 关中断、恢复现场和屏蔽字、开中断，然后中断返回对应信息
- 原语
  - 原语是一个程序段。原子性
    - 由若干指令组成
    - 来完成某个特定功能
    - 执行过程不能被中断
  - 原语运行在内核空间
- 系统数据结构
  - 进程管理：作业控制块、进程控制块

    - 存储器管理：储存器分配与回收
    - 设备管理：缓冲区、设备控制块
- 系统调用
    - 系统调用是（由陷入/陷阱中断实现的）
        - 由OS实现给应用程序调用的
        - 一套用于访问内核服务的接口的集合
    - 系统调用的处理运行在核心态

操作系统的结构如何设计的?

- 传统的操作系统结构
    - 第一代：无结构OS，
        - 一系列功能集合，过程中相互调用，
        - 结构复杂混乱，可读性差
    - 第二代：模块化结构OS：模块-接口法OS
        - 模块独立性标准：高内聚、低耦合
        - 难扩展、没有统一决策标准
    - 第三代：分层式结构OS
        - 有序分层，自顶而下依赖。但设计时：自底而上
        - 保证系统正确性、容易扩展和维护
        - 自上而下层次通信，系统效率低
    - 第四代：微内核OS结构
        - 足够小的内核，采用面向对象技术，与硬件处理紧密关联，只实现OS核心功能和基本功能
        - 提高扩展性、可靠性、可移植性、支持分布式系统
        - 相比于早期OS，效率下降