操作系统

第一章

课件. 操作系统的功能概念定义

- 程序接口：即系统调用 ppt8页
  - 程序通过系统调用使用程序接口
  - 程序接口由一系列系统调用组成
- 系统调用：广义指令
- 用户接口：命令接口 + 程序接口

课件1 操作系统的特征

- 四大特征：
  - 并发
    - 并发和并行
  - 共享
    - 同时共享
    - 互斥共享
  - 虚拟
    - 空分复用？（虚拟存储器技术？） ppt 4页
    - 时分复用（虚拟处理器）
  - 异步：走走停停

课件2 操作系统的发展与分类

- 实时操作系统：
  - 软实时
  - 硬实时

课件3 操作系统的运行机制

- 两种指令：
  - 特权指令
  - 非特权指令
- 两种状态：
  - 内核态：管态
    - 可以执行特权指令和非特权指令
  - 用户态：目态
- 内核态、用户态的切换：
  - 内核->用户：执行特权指令，主动让出
  - 用户->内核：中断，强行夺取cpu
- 原语：
  - 原子性：运行不可中断 一气呵成
  - 最接近硬件的部分
- 大内核和微内核区别：
  - 进程管理
  - 存储器管理
  - 设备管理

课件4 中断和异常

- 中断作用：由用户态->内核态；操作系统夺取cpu的唯一途径
- 中断分类：
  - 内中断（异常）：来自cpu内部；与当前执行指令有关
    - 非法指令：特权指令；除法除数为0
    - 陷入指令：主动请求操作系统服务；（系统调用）
    - 陷入（陷阱）、故障（内核来修复）、终止（内核来终止）
  - 外中断：来自cpu外部；与当前执行指令无关
    - IO中断
    - 时钟中断
- 中断向量表：根据中断信号类型，找到中断处理程序的程序地址（指针）

课件5 系统调用

- 程序接口：系统调用
- 系统调用与库函数：
  - 某些库函数封装了系统调用供程序员使用
  - 大佬也可以自己写系统调用的操作，不调用库函数的封装
- 系统调用的作用：
  - 经由操作系统内核，统一分配控制共享资源
  - 涉及共享资源的使用：都需要统一经由操作系统内核提供服务来完成

- 系统调用入口程序：中断处理程序
- 系统调用处理程序：由入口程序（中断处理程序）调用
- 传参指令：本质是往寄存器传参

课件6 操作系统的体系结构

- 大内核和微内核区别：
  - 进程管理
  - 存储器管理
  - 设备管理
  - 这些管理 更多是数据结构的操作（信息状态操作） 而不直接涉及硬件
- 大内核和微内核的变态次数：？
  - 大内核较少
  - 微内核较多

无课件7

课件8 操作系统的体系结构

- 分层结构：
  - 优点：便于自底向上调试验证（底层无调用 可排查是否有问题）
  - 易于扩充，只要满足上层接口调用，按规定调用下层接口即可
- 模块化：主模块 + 可加载内核模块
- 宏内核：运用了一定的模块化思想
- 微内核：运用了一定的模块化思想
- 外核：负责为用户进程分配未经抽象的硬件资源，并且由外核负责保证资源使用安全

课件9 操作系统引导Boot（开机过程）

- 主引导记录：（MBR）
  - 磁盘引导程序
  - 分区表
- 分区引导记录：（在C盘PBR）
  - 负责找到启动管理器
- 开机过程?
  - cpu从主存特定地址rom中读取引导程序（先硬件自检，再开机）
  - 磁盘第一块，主引导记录，执行磁盘引导程序，扫描分区表
  - 从活动分区（主分区）中第一块，读入分区引导记录，执行程序
  - 找到根目录下的，启动管理器，完成开机一系列动作
- ram和rom
  - ram：随机存取
  - rom：只读

课件10 虚拟机

- vmm：虚拟机管理程序
  - 与虚拟机的区别?
- 第一类vmm和第二类对比
  - 是否有host os

    - 是否直接控制分配硬件资源
    - 资源分配方式
    - 运行模式
- 指令等级?
    - 支持虚拟化通常分更多指令等级?
    - 即不再单纯分为：特权指令和非特权指令两种

第二章

课件. 进程的概念、组成、特征

- 程序：静态的，可执行文件exe，存放在磁盘中
- 进程：动态的，程序的一次执行过程
- PCB（process control block）：进程控制块
    - 进程存在的唯一标志：
        - 进程创建时，开始运行前，创建PCB
        - 进程销毁时，回收PCB
    - 进程描述信息
    - 进程控制和管理信息
    - 资源分配清单
    - 处理机相关信息
- 进程的组成：
    - PCB
    - 程序段
    - 数据段
- 进程是动态的，进程实体是静态的（反应了某一时刻进程的状态）
- **进程是操作系统进行资源分配和调度的一个独立（基本）单位**
- 进程的特征：
    - 动态性：**基本特征**
    - 并发性
    - 独立性
    - 异步性：**需要操作系统同步机制来解决异步带来的问题**
    - 结构性

课件1 进程的状态与转换

- 进程的状态：
    - 创建态
    - 就绪态
    - 运行态
    - 阻塞态：
        - 进程运行过程中可能会请求等待某个事件（等待系统资源的分配或者其他进程的响应等等）的发生，在事件发生之前，进程无法继续执行，进入阻塞态
    - 终止态：exit系统调用
        - 父进程可能还要用到子进程的资源，因此子进程不能完全杀掉?
- 进程的组织方式：将同一状态的进程组织起来，进行统一管理
    - 链接方式（通过队列组织）：

        - 运行队列
        - 就绪队列
        - 阻塞队列（不同操作系统会根据阻塞原因不同，分多个阻塞队列）
    - 索引方式：
        - 运行表指针
        - 就绪表指针
        - 阻塞表指针
        - 索引表里记录各个进程

课件2 进程控制

- 进程控制：就是实现进程状态的转换
    - 如何实现：原语
    - 为何需要原语，一气呵成：否则造成操作系统中某些数据结构信息的混乱（eg：阻塞队列中出现state为就绪的进程，原因是没来得及放入就绪队列，就因为中断而停止了进程状态转换）
- 进程的创建：
    - 创建原语
    - 引起进程创建的事件：
        - 用户登录：分时操作系统中，用户登录成功，系统会为其建立一个新的进程？
        - 作业调度：内核处理程序运行的进程
        - 提供服务
        - 应用请求
- 进程的终止：（有几种状态可以直接变成终止态？我认为是只有运行态？ ppt p5）
    - 撤销原语：
        - 特别注意：终止所有子进程
        - 特别注意：将资源归还给父进程或者操作系统
    - 引起进程终止地事件：
        - 正常结束
        - 异常结束
        - 外界干预
- 进程的阻塞和唤醒：
    - 进程的阻塞：
        - 阻塞原语
            - 注意：保护进程运行现场
        - 引起阻塞的事件
    - 进程的唤醒：
        - 唤醒原语
        - 引起唤醒的事件
- 进程的切换：（运行态和就绪态的切换）
    - 切换原语
        - 运行环境信息：进程上下文，存入PCB
    - 引起进程切换的事件
        - 当前进程主动阻塞
        - 当前进程终止（这两种情况不是应该是阻塞原语和撤销原语吗？ ppt p6 p12）
- 程序是如何运行的：
    - 除了数据段，cpu寄存器中会存放程序运行中所需要的某些数据
        - cpu指令执行急用的数据，放寄存器，记忆存储类型的数据（不急用的），放数据段

    - PSW:程序状态字寄存器（记录内核态还是用户态），与PCB中的state不同
    - PC：程序计数器，存放下一条指令的地址
    - IR：指令寄存器，存放当前正在执行的指令
    - 通用寄存器：存放其他的必要信息
    - 当进程运行一半，某条指令执行完后，另一个进程上cpu
        - 解决：**在切换进程前，在PCB中保存进程的运行环境（一些必要的寄存器值，也称为进程上下文）**
- 原语干的三件事：
    - 更新PCB信息
        - 更新进程状态
        - 保存运行环境
        - 恢复运行环境
    - 更新PCB队列
    - 分配/回收资源
        - 杀死子进程
        - 回收PCB

课件3 进程通信IPC

- 共享存储：
    - 操作系统内核提供的同步互斥工具？ ppt p3
        - 由通信进程自己实现互斥？
    - 共享内存区映射到进程自己的内存空间？
    - 基于数据结构的共享（低级）？
    - 基于存储区的共享（高级）？
- 消息传递：
    - 直接通信方式
        - 接收进程PCB中的消息队列
    - 间接通信方式
        - 操作系统内核程序的地址空间，有信箱
    - 发送/接收消息原语：
- 管道通信：
    - 半双工通信（实现双向通信需要设置两个管道）
    - 各进程互斥地访问管道（操作系统实现）
    - 管道写满，写进程阻塞
    - 管道为空，读进程阻塞
    - 多个进程读同一个管道可能会错乱？
        - 解决方案：
            - 一个读进程
            - 多个读进程（由操作系统协调读）

课件4 线程概念

- 线程概念：
    - 一个进程内可有多个线程
    - 引入线程后：
        - 进程：除cpu外系统资源的分配单元

        - 线程：cpu（处理机）的分配单元
        - 即进程是资源分配的基本单位，线程是调度的基本单位
    - 线程也有线程ID、线程控制块TCB

课件5 线程的实现方式 多线程模型

- 用户级线程：
    - 线程库是什么，有什么作用，如何实现其作用?
    - 从用户视角看到的线程，由线程库实现
- 内核级线程：**才是处理机分配的基本单位**
    - 从操作系统角度看到的线程
- 多线程模型：
    - 一对一（类似内核级线程）
        - 一个用户级线程映射到一个内核级线程
    - 多对一（类似用户级线程）
        - 多个用户级线程映射到一个内核级线程
    - 多对多（内核级线程和用户级线程的中和）
        - n个用户级线程映射到m个内核级线程（n >= m）
    - 用户级线程：代码逻辑的载体
    - 内核级线程：运行机会的载体

课件6 线程的状态与转换

- TCB（线程控制块）： +
- 线程表：

课件7 调度算法

- 先来先服务
- 最短作业优先
- 最高相应比优先
- 作业和进程的区别?