# 处理机调度与死锁

2021年11月3日 22:25

## 1、处理机调度的层次与调度算法的目标

1. 处理机调度的层次
    a. 高级调度
        i. 长程调度，作业调度
        ii. 作业为调度对象
        iii. 主要用于多道批处理系统
    b. 低级调度
        i. 短程调度，进程调度
        ii. 进程为调度对象
        iii. 最基本的调度
    c. 中级调度
        i. 内存为调度对象
        ii. 目的为提高内存利用率与系统吞吐量
2. 处理机调度算法的目标
    a. 共同目标
        i. 资源利用率

        $$\text{CPU利用率} = \frac{\text{CPU有效工作时间}}{\text{CPU有效工作时间+CPU空闲等待时间}}$$

        ii. 公平性→各进程获得合理的CPU时间
        iii. 平衡性→系统资源使用平衡
        iv. 策略强制执行
    b. 批处理系统的目标
        i. 平均周转时间短
        ii. 系统吞吐量高→短作业
        iii. 处理机利用率高→计算量大的作业
    c. 分时系统的目标
        i. 响应时间快
        ii. 均衡性→响应时间与服务复杂度适应
    d. 实时系统的目标
        i. 截止时间的保证
        ii. 可预测性

## 2、作业与作业调度

1. 批处理系统中的作业
    a. 作业和作业步
        i. 作业 较程序更广泛，除程序和数据还包括作业说明书
        ii. 作业步

1) 编译
2) 装配链接
3) 运行

b. 作业控制块JCB
c. 作业运行的三阶段三状态
   i. 收容阶段 后备状态
   ii. 运行阶段 运行状态
   iii. 完成阶段 完成状态

2. 作业调度的主要任务
   a. 接纳多少个作业
      i. 作业过少：吞吐量低
      ii. 作业太多：内存不足发生中断概率增加，周转周期延长
   b. 接纳哪些作业
3. 先来先服务和短作业优先调度算法
   a. 先来先服务调度算法FCFS

      到达时间越早，优先级越高
   b. 短作业优先调度算法SJF

      作业越短，优先级越高

      缺点：
      1) 必需准确预知作业运行时间
      2) 对长作业不利
      3) 无人机交互性
      4) 不考虑作业紧迫程度
4. 优先级调度算法和高响应比优先调度算法
   a. 优先级调度算法PSA

      基于作业紧迫程度由外部赋予优先级，根据优先级调度
   b. 高响应比优先调度算法HRRN

      优先权=响应比$R_p$

      $R_p = \frac{\text{等待时间+要求服务时间}}{\text{要求服务时间}} = \frac{\text{响应时间}}{\text{要求服务时间}}$

## 3、进程调度

1. 进程调度的任务，机制和方式
   a. 进程调度的任务
      i. 保存处理机现场信息
      ii. 按某种方法选取进程
      iii. 把处理器分配给进程
   b. 进程调度机制
      i. 排队器
      ii. 分配器
      iii. 上下文切换器
   c. 进程调度方式

i. 非抢占方式
ii. 抢占方式
1) 优先权原则
2) 短进程优先原则
3) 时间片原则

2. 轮转调度算法RR
   a. 轮转法的基本原理
   b. 进程切换时机
   c. 时间片大小的确定
3. 优先级调度算法PSA
   a. 优先级调度算法的类型
      i. 非抢占式优先级调度算法
      ii. 抢占式优先级调度算法
   b. 优先级的类型
      i. 静态优先级
         1) 进程类型
         2) 进程对资源的需求
         3) 用户要求
      ii. 动态优先级
4. 多队列调度算法MQ

   将一个就绪队列拆分为多个，就绪队列中的进程可以有优先级，不同就绪队列本身可以有不同优先级
5. 多级反馈队列调度算法MFQ
   a. 调度机制
      i. 设置多个就绪队列，队列有不同优先级，优先级越高的队列时间片越短
      ii. 每个队列采用FCFS算法，时间片内未完成的插入下一级就绪队列
      iii. 按队列优先级调度
   b. 调度算法的性能
6. 基于公平原则的调度算法
   a. 保证调度算法

      进程得到相同的时间片

      明确的性能保证
   b. 公平分享调度算法

      用户得到相同的时间片

## 4、实时调度

1. 实现实时调度的基本条件
   a. 提供必要的信息
      i. 就绪时间
      ii. 开始截止时间和完成截止时间
      iii. 处理时间

iv. 资源要求
v. 优先级

b. 系统处理能力强
c. 采用抢占式调度机制
d. 具有快速切换机制

2. 实时调度算法的分类
   a. 非抢占式调度算法
      i. 非抢占式轮转调度算法
      ii. 非抢占式优先级调度算法
   b. 抢占式调度算法
      i. 基于时钟中断的抢占式优先级调度算法
      ii. 立即抢占的优先级调度算法
3. 最早截止时间优先算法EDF

   任务的截止时间越早，优先级越高
4. 最低松弛度优先算法LLF

   任务的紧急度越高，优先级越高

   松弛度 = 必须完成时间-本身运行时间-当前时间
5. 优先级倒置

   高优先度进程与低优先度进程共享临界资源，低优先度进程未释放锁时被打断，使高优先度进程被阻塞，此过程中如有中等有限度进程则会延长高优先度进程的阻塞时间

   解决方案：直接由低优先度进程继承高优先度进程的优先级

## 5、死锁概述

1. 资源问题
   a. 可重用性资源与消耗性资源
      i. 可重用性资源
      ii. 可消耗性资源
   b. 可抢占性资源和不可抢占性资源
      i. 可抢占性资源
      ii. 不可抢占性资源
2. 计算机系统中的死锁
   a. 竞争不可抢占性资源引起死锁
   b. 竞争可消耗性资源引起死锁
   c. 进程推进顺序不当引起死锁
3. 死锁的定义、必要条件和处理方法
   a. 死锁的定义

      如果一组进程中的每一个进城都在等待仅由该组中的其他进程才能引发的事件，那么该组进程是死锁的
   b. 死锁产生的必要条件
      i. 互斥条件

ii. 请求和保持条件
iii. 不可抢占条件
iv. 循环等待条件

c. 处理死锁的方法
i. 预防死锁
ii. 避免死锁
iii. 检测死锁
iv. 解除死锁

## 6、预防死锁

1. 破坏“请求和保持”条件
   a. 第一种协议
      进程在开始前，必须一次性的申请到其在整个运行过程中所需要的全部资源
      i. 资源被严重浪费，恶化资源的利用率
      ii. 使进程经常会发生饥饿现象
   b. 第二种协议
      允许一个进程只获得运行初期所获的资源便开始运行，运行过程中逐步释放分配给自己的，已经用毕的全部资源，再请求新的所需资源
2. 破坏“不可抢占”条件
   当一个已经保持了某些不可被抢占资源的进程，提出新的资源请求而不能得到满足时，必须释放已保持的所有资源，待以后需要时再重新申请
3. 破坏“循环等待”条件
   对系统所有资源类型进行线性排序，并赋予不同的序号，每个进程必须按序号递增的顺序请求资源

## 7、避免死锁

1. 系统安全状态
   a. 安全状态
      i. 安全状态指系统能按某种进程推进顺序为每个进程分配其所需资源，直至满足每个进程对资源的最大需求，使每个进程都可顺利完成。称序列为安全序列
      ii. 系统只要处于安全状态便不会进入死锁状态
   b. 由安全状态向不安全状态的转换
2. 利用银行家算法避免死锁

## 8、死锁的检测与解除

1. 死锁的检测
   a. 资源分配图
   b. 死锁定理
2. 死锁的解除
   a. 抢占资源
      从一个或多个进程中抢占足够多的资源分配给死锁进程

b. 终止进程

终止一个或多个死锁进程，直到打破循环环路，使系统从死锁状态中解脱

i. 终止所有死锁进程

ii. 逐个终止进程

c. 付出最小代价的死锁解除算法