

第三章作业

姓名：袁昊男 学号：2018091618008

1. 高级调度和低级调度的主要任务是什么？

高级调度的主要任务是根据某种算法，把外存上处于后备队列中的哪些作业调入内存，为它们创建进程、分配必要的资源，并将它们放入就绪队列。

低级调度的主要任务是根据某种算法，决定就绪队列中的哪个进程获得处理机，再把处理器分配给进程。

2. 说明低级调度的主要功能是什么？

主要功能：(1)保存处理机的现场信息；(2)按某种算法选取进程；(3)把处理机分配给进程。

3. 在抢占调度方式中，抢占的原则是什么？

- (1) 优先权原则：允许优先级高的进程抢占低优先级进程的处理机；
- (2) 短进程优先原则：允许新到的短进程抢占当前长进程的处理机；
- (3) 时间片原则：当前进程时间片用完后立即停止并重新调度。

4. 讨论轮转调度中，时间片的大小会有什么影响？

- (1) 时间片小：有利于短作业在一个时间片内完成；但会导致频繁进行进程调度和上下文切换，增加系统开销。
- (2) 时间片长：无法满足短作业和交互式用户的需求。

5. 为什么说多级反馈队列调度算法能较好满足各类用户的需求

- (1) 终端型用户：提交的作业大多属于较小的交互型作业，系统只要使这些作业在第一队列规定的时间片内完成，终端作业用户就会感到满意。
- (2) 短批处理作业用户：对于稍短作业，如果在第一队列中执行一个时间片段即可完成，便可获得与终端作业一样的响应时间。对于稍长作业，通常只需在第二和第三队列各执行一时间片即可完成，其周转时间仍然较短。
- (3) 长批处理作业用户：长作业将依次在第1, 2, ..., n个队列中运行，然后再按轮转方式运行，用户不必担心其作业长期得不到处理。

6. 何谓死锁？产生死锁的原因和必要条件是什么？

- (1) 定义：一组进程中的每一个进程都在等待仅由该组进程中的其他进程才能引发的事件，则这组进程处于死锁。换句话说，死锁是指多个进程在运行过程中因争夺资源而造成的一种僵局，当进程处于这种僵持状态时，若无外力作用，它们都将无法再向前推进。
- (2) 原因：竞争不可抢占或可消耗资源、进程间推进顺序非法。

(3) 必要条件：互斥条件、保持且请求条件、不剥夺条件、循环等待条件。

7. 解释银行家算法。

银行家算法分为以下几大部分：

(1) 数据结构：可利用资源Available（ $Available[j] = K$ 表示j类资源有K个）、最大需求矩阵Max（ $Max[i, j] = K$ 表示进程i需要j类资源K个）、分配矩阵Allocation（ $Allocation[i, j] = K$ 表示进程i已经得到K个j类资源）、需求矩阵Need（ $Need[i, j] = K$ 表示进程i还需要K个j类资源）。三个矩阵关系： $Need[i, j] = Max[i, j] - Allocation[i, j]$ 。

(2) 银行家算法：（ $Request_i[j] = K$ 表示进程i需要K个j类资源。

1) 若 $Request_i[j] \leq Need[i, j]$ （申请的比需求少），转2)；否则出错。

2) 若 $Request_i[j] \leq Available[j]$ （申请的比现有的少），转3)；否则等待。

3) 尝试分配资源，修改数值：

$Available[j] = Available[j] - Request_i[j]$ ；当前可用资源因为分配后变少

$Allocation[i, j] = Allocation[i, j] + Request_i[j]$ ；已分配资源得到分配后变多

$Need[i, j] = Need[i, j] - Request_i[j]$ ；当前进程还需要的资源因得到分配后变少

4) 执行安全性算法，检查是否因为分配进入不安全状态。安全则分配，不安全则恢复3)中操作。

(3) 安全性算法：

1) 当前可分配资源数向量Work（类似于Available）、进程完成标志Finish。

2) 找一个满足下列条件的进程：

a) $Finish[i] = false$ ；找一个未结束的进程i

b) $Need[i, j] \leq Work[j]$ ；当前可分配资源能满足i进程的资源分配需求找到i进程后转3)，否则4)。

3) i进程获得资源，结束后释放资源：

$Work[j] = Work[j] + Allocation[i, j]$ ；将i进程分配所得资源释放回Work

$Finish[i] = true$ ；标志i进程结束

go to step 2；寻找下一个未完成进程分配资源

4) 若所有进程都执行完毕，则表示系统处于安全状态；反之，则不安全。