

2022/1/29

第三章

1. 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那些作业调入内存。低级调度是保存处理中的现场信息,按某种算法先取进程,再把处理完分配给进程。引入中级调度的主要目的是为了提离内存利用率与系统吞吐量。使那些暂时不能运行的进程不再占用内存资源,将它们调至外存等待,把进程状态改为就绪/外存状态或挂起状态。

6. 高响应比优先调度算法是一种高优先调度算法,于其中的优先权,即响应比等于:

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

因此,它具有以下的优点:

(1) 如果作业(进程)的等待时间相同,则要求服务时间最短的作业(进程)的优先权最高,因此它有利短作业(进程),从而可降低作业(进程)的平均周转时间,提高系统吞吐量。

(2) 如果作业(进程)的要求服务时间相同,则其优先权取决于作业到达(或进程进入就绪状态)的先后次序,因此体现了公平的原则。

(3) 如果作业(进程)较长,它的优先权将随着等待增长而提高,从而使长作业不会长期得不到服



夸克扫描王

极速扫描,就是高效



11. 一旦确定下来, 进程在运行期间它的优先级一直不变化的优先级叫静态优先级。系统首先为一个进程只赋予初始优先级, 该优先级会随着进程的运行而改变, 这样的优先级叫动态优先级。确定静态优先级的依据: ①进程类型 ②进程对资源的需求 ③用户需求。

12. 时间片应略大于一次典型的交互需要的时间。一般应考虑三个因素: 系统对相应时间的要求、就绪队列中进程的数目、系统的处理能力。

20. 按调度方式的不同, 实时调度算法分为非抢占调度算法、抢占调度算法。由于非抢占式调度算法比较简单, 易于实现, 故在一些小型实时系统或要求不太严格的实时控制系统中经常采用之, 可以分为非抢占式轮转调度算法和非抢占式优先级调度算法; 在要求较严格的(如实时性数十毫秒以下)的实时系统中, 应采用抢占式优先级调度算法。根据抢占与发生时间的不同而进一步分为基于时钟中断的抢占式调度算法和立即抢占的优先级。

23. “优先级倒置”是指高优先级进程(或线程)被低优先级进程(或线程)延迟或阻塞的现象。解决方法: ①优先级继承 ②优先级天花板



27. 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。产生死锁的原因为竞争资源和进程间推进顺序非法。其必要条件是：互斥条件、请求和保持条件、不剥夺条件、环路等待条件。

30. ~~P0发出Request(0,1,0), 按银行家算法进行检查。① Request(0,1,0) ≤ Need(7,4,3) ② Request(0,1,0) ≤ Available(2,3,0) ③ 系统暂时先假定可为P0分配资源, 修改 Available, Allocation 和 Need:~~
~~Available[j] := Available[j] - Request[i,j]; Allocation[i,j] := Allocation[i,j] + Request[i,j]; Need[i,j] := Need[i,j] - Request[i,j]; 即: 计算结果:~~

进程 \ 资源情况	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P ₀	7	5	3	0	1	0	7	4	3	3	3	2
P ₁	3	2	2	2	0	0	1	2	2			
P ₂	9	0	2	3	0	2	6	0	0			
P ₃	2	2	2	2	1	1	0	1	1			
P ₄	4	3	3	0	0	2	4	3	1			

可以分配



31.

(1)	work				Need				Allocation				Work + Allocation				Finish
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	
P₀	1	6	2	2	0	0	1	2	0	0	3	2	1	6	8	4	True
P ₃	1	6	5	4	0	6	5	2	0	3	3	2	1	9	8	6	True
P ₄	1	9	8	6	0	6	5	6	0	0	1	4	1	9	9	10	True
P ₁	1	9	9	10	1	7	5	0	1	0	0	0	2	9	9	10	True
P ₂	2	9	9	10	2	3	5	6	1	3	5	4	3	12	14	14	True

此刻能找到一安全序列 $\{P_0, P_3, P_4, P_1, P_2\}$, 此刻系统安全。

2) 若 P₂ 申请 (1, 2, 2, 2) 后, 系统判断 Request (1, 2, 2, 2) 预分配。但分配后 Available 为 (0, 4, 0, 0), 已无法满足任何一个进程的资源需求, 系统不能将资源分配给它。

