

# 操作系统中常用调度算法的比较

哈森格日乐

(兴安盟电大分校, 内蒙古 乌兰浩特 137400)

在操作系统中存在着多种调度算法, 有的适于作业调度, 有的适于进程调度, 也有的调度算法对二者都可用。本文介绍常用的几种算法。

## 一、实现方法的比较

1. 先来先服务法 先来先服务方法的实现思想是“排队买票”的办法。对于进程调度来说, 采用先来先服务法, 就是每次调度从就绪队列中选择一个最先进入该队列的进程, 把 CPU 分给它, 令其投入运行。该进程一直运行下去, 直至完成或者由于某些原因而阻塞, 才放弃 CPU。这样, 当一个进程进入就绪队列时, 它的 PCB 就链入就绪队列的末尾。每次进程调度时把队头进程从该队列中摘下, 分给它 CPU, 使它运行。

2. 时间片轮转法 时间片轮转法主要用于分时系统中的进程调度。为实现轮转调度, 系统把所有就绪进程按先进先出的原则排成一个队列。新来的进程加到就绪队列末尾。每当执行进程调度时, 进程调度程序总是选出就绪队列的队首进程, 让它在 CPU 上运行一个时间片的时间。当进程用充分给它的时间片后, 系统的计时器发出时钟中断, 调度程序便停止该进程的运行, 并把它放入就绪队列的末尾; 然后, 把 CPU 分给就绪队列的队首进程, 同样也让它运行一个时间片, 如此往复。

3. 优先级法 优先级法的实现思想是哪个进程的优先级高, 就先运行哪个。非抢占式优先级法是当前占用 CPU 的进程一直运行下去, 直到完成任务或者因等待某种事件而主动让出 CPU 时, 系统让另一个优先级高的进程占用 CPU。

## 二、平均周转时间和平均带权周转时间的比较

现举一例子, 对上述三种算法进行比较。假定在单 CPU 条件下有下列要执行的作业:

作业	运行时间	优先级
1	10	3
2	1	1
3	2	3
4	1	4
5	5	2

设作业到来的时间是按作业编号顺序进行的。这时我们比较一下利用三种算法计算出的平均周转时间和平均带

权周转时间。

我们首先来定义一下周转时间和带权周转时间。周转时间为从作业提交到作业完成的时间间隔。带权周转时间为周转时间除以实际运行时间。

### 1. 利用先来先服务法

作业	到达时间	运行时间	完成时间	周转时间	带权周转时间
1	0	10	10	10	1.0
2	1	1	11	10	10.0
3	2	2	13	11	5.5
4	3	1	14	11	11.0
5	4	5	19	15	3.0
平均周转时间				11.4	
平均带权周转时间				6.1	

### 2. 利用时间片轮转法

作业	到达时间	运行时间	完成时间	周转时间	带权周转时间
1	0	10	19	19	1.9
2	1	1	2	1	1.0
3	2	2	8	6	3.0
4	3	1	5	2	2.0
5	4	5	16	12	2.4
平均周转时间				8.0	
平均带权周转时间				2.06	

### 3. 利用非抢占式优先级法

作业	到达时间	运行时间	完成时间	周转时间	带权周转时间
1	0	10	10	10	1.0
2	1	1	19	18	18.0
3	2	2	13	11	5.5
4	3	1	11	8	8.0
5	4	5	18	14	2.8
平均周转时间				12.2	
平均带权周转时间				7.06	

由上表可看出, 先来先服务法有利于长作业(进程), 而不利短作业(进程)。因为短作业运行时间很短, 如果让它等待较长时间才得到服务, 那么, 它的带权周转时间就会很高。在轮转法中, 一次轮回时间内分给任何进程的 CPU 时间都不会大于一个时间片。如果一个进程在一个时间片内没有做完自己的事情, 那么在时间片用完时, 该进程就被剥夺对 CPU 的控制权, 放回到就绪队列的末尾。所以, 一个需运行较长时间的进程要经过多次轮转才能完成工作。非抢占式优先级法中, 如果有一个进程在运行过程中, 即使系统中出现一个优先更高的进程, 也要等到当前进程运行完毕, 而不迫使它放弃 CPU。

[责任编辑: 张建荣]