

os 作业 3

处理机调度

高级调度（长程/作业/宏观调度），中级调度（交换调度），低级调度（短程/CPU/进程/微观调度）

各种调度算法比较：

一、作业调度

1、先来先服务（fcfs）

按进程（作业）进入就绪（后备）队列的先后次序来分配处理机。一般采用非剥夺的调度方式。

比较简单，但是效率不高，有利于 CPU 繁忙型作业，不利于 I/O 繁忙型作业，常被结合在其它调度策略中使用。

2、短作业优先（sjf）

用于作业调度

主要任务是从后备队列中选择一个或若干个估计运行时间最短的作业，将它们调入内存运行。能有效降低作业的平均等待时间，能有效缩短进程的平均周转时间，提高了吞吐量。但是对长作业不利，作业执行时间、剩余时间仅为估计。SJF 算法虽然是优化的，但在 CPU 调度中很难实现。

3、优先级调度算法（psa）

优先级调度算法以作业的紧迫程度为优先级，系统选择优先级最高的几个作业装入内存。优先级调度算法也用于进程调度，系统在可运行的进程中选择优先

级最高者使其投入运行。

优先级的类型分为静态优先级和动态优先级

静态优先级

优先权在创建进程时确定，且在进程的整个运行期间保持不变。一般用整数表示，小表示优先级高。优点是简单，开销小，但公平性差。

动态优先级

动态优先级在进程的存在过程中不断发生变化，动态优先级的变化取决于进程的等待时间，进程的运行时间，进程使用资源的情况。动态优先权确定方法的资源利用率高，公平性好，但开销较大，实现较为复杂。

4、高响应比优先调度算法（hrrn）

FCFS 只考虑了作业的等待时间，忽略了运行时间。SJF 只考虑了作业的运行时间，忽略了等待时间。高响应比优先调度算法既考虑了作业的等待时间，也考虑了作业的运行时间，是一种动态优先级调度算法。

二、进程调度

1:最短剩余时间调度算法（srt）

最短剩余时间调度算法是针对 SPF 增加了抢占机制的一种调度算法。选择预期剩余时间最短的进程。只要新进程就绪，且有更短的剩余时间，调度程序就可能抢占当前正在运行的进程。

SRT 不像 FCFS 偏向长进程，也不像轮转法（下个算法）产生额外的中断，从而减少了开销。平均周转时间和带权平均周转时间说明了 SRT 好于前面的任何一个算法（因为它具有抢占的特点）。但是必须记录过去的服务时间，从而增加了开销

2:时间片轮转调度算法 (rr)

时间片轮转调度算法：系统将所有就绪进程按 FCFS 的原则，排成一个队列依次调度，把 CPU 分配给队首进程，并令其执行一个时间片，通常为 10-100ms。时间片用完后，系统的计时器发出时钟中断，该进程将被剥夺 CPU 并插入就绪队列末尾。RR 是一种非常公平的算法。

3:多级队列调度算法

非抢占式优先级调度算法：一旦把处理机分配给就绪队列中优先级最高的进程后，该进程便一直执行下去直至完成，或者因该进程发生某事件而放弃处理机时，系统方可将处理机重新分配给另一优先级最高的进程。

抢占式优先级调度算法：只要出现了另一个其优先级更高的进程，调度程序就将处理机分配给新到的优先级最高的进程

4:多级反馈队列调度算法 (mfq)

多级反馈队列调度算法是时间片轮转算法和优先级调度算法的综合和发展，通过动态调整进程优先级和时间片大小，不必事先估计进程的执行时间。