**处理机调度经典算法比较**

【自主回顾】

1.调度的基本概念：在多道程序系统中，进程的数量往往多于处理机的个数，进程争用处理机的情况就在所难免。处理机调度是对处理机进行分配，就是从就绪队列中，按照一定的算法（公平、髙效）选择一个进程并将处理机分配给它运行，以实现进程并发地执行。

·处理机调度是多道程序操作系统的基础，它是操作系统设计的核心问题。

2.调度的基本准则：不同的调度算法具有不同的特性，在选择调度算法时，必须考虑算法所具有的特性。为了比较处理机调度算法的性能，人们提出很多评价准则，主要的几种如下：

1) CPU利用率。CPU是计算机系统中最重要和昂贵的资源之一，所以应尽可能使CPU 保持“忙”状态，使这一资源利用率最髙。

2) 系统吞吐量。表示单位时间内CPU完成作业的数量。长作业需要消耗较长的处理机时间，因此会降低系统的吞吐量。而对于短作业，它们所需要消耗的处理机时间较短，因此能提高系统的吞吐量。调度算法和方式的不同，也会对系统的吞吐量产生较大的影响。

3) 周转时间。是指从作业提交到作业完成所经历的时间，包括作业等待、在就绪队列中排队、在处迤机上运行以及进行输入/输出操作所花费时间的总和。

·作业的周转时间可用公式表示如下：

周转时间 = 作业完成时间 - 作业提交时间

平均周转时间是指多个作业周转时间的平均值：

平均周转时间 = (作业1的周转时间 + … + 作业 n 的周转时间) / n

带权周转时间是指作业周转时间与作业实际运行时间的比值：

平均带权周转时间是指多个作业带权周转时间的平均值：

平均带权周转时间 = (作业1的带权周转时间 + … + 作业 n 的带权周转时间) / n

4) 等待时间。是指进程处于等处理机状态时间之和，等待时间越长，用户满意度越低。处理机调度算法实际上并不影响作业执行或输入/输出操作的时间，只影响作业在就绪队列中等待所花的时间。因此，衡量一个调度算法优劣常常只需简单地考察等待时间。

5) 响应时间。是指从用户提交请求到系统首次产生响应所用的时间。在交互式系统中，周转时间不可能是最好的评价准则，一般釆用响应时间作为衡量调度算法的重要准则之一。从用户角度看，调度策略应尽量降低响应时间，使响应时间处在用户能接受的范围之内。

【经典算法比较】

1. **先来先服务算法（FCFS）**

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

FCFS调度算法属于不可剥夺算法。从表面上看，它对所有作业都是公平的，但若一个长作业先到达系统，就会使后面许多短作业等待很长时间，因此它不能作为分时系统和实时系统的主要调度策略。但它常被结合在其他调度策略中使用。例如，在使用优先级作为调度策略的系统中，往往对多个具有相同优先级的进程按FCFS原则处理。

FCFS调度算法的特点是算法简单，但效率不高；对长作业比较有利，但对短作业不利（相对SJF和高响应比）；有利于CPU繁忙型作业，而不利于I/O繁忙型作业。现在操作系统中，已很少用该算法作为主要调度策略，尤其是在分时系统和实时系统中。但它常被结合在其它调度策略中使用。

1. **优先级调度算法（PSA）**

基本思想：随着计算机的发展，特别是实时操作系统的的出现，越来越多的应用场景需要根据任务的紧急程度来决定处理顺序。

优先级调度算法又称优先权调度算法，该算法既可以用于作业调度，也可以用于进程调度，该算法中的优先级用于描述作业运行的紧迫程度。

在作业调度中，优先级调度算法每次从后备作业队列中选择优先级最髙的一个或几个作业，将它们调入内存，分配必要的资源，创建进程并放入就绪队列。在进程调度中，优先级调度算法每次从就绪队列中选择优先级最高的进程，将处理机分配给它，使之投入运行。

优点是可以灵活调整对各种进程的偏好。但若有源源不断的高优先级进程到来，则可能导致饥饿。

1. **短作业优先（SJF）**

短作业主要用于作业调度,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。类似地，用于进程调度的是短进程优先调

优点：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。  
缺点：对长作业不利；没有考虑作业的紧迫程度；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。

1. **高响应比优先调度算法（HRRN）**

高响应比优先调度算法主要用于作业调度，该算法是对FCFS调度算法和SJF调度算法的一种综合平衡，同时考虑每个作业的等待时间和估计的运行时间。在每次进行作业调度时，先计算后备作业队列中每个作业的响应比，从中选出响应比最高的作业投入运行。

响应比的变化规律可描述为：

IMG_256

根据公式可知：

·当作业的等待时间相同时，则要求服务时间越短，其响应比越高，有利于短作业。

·当要求服务时间相同时，作业的响应比由其等待时间决定，等待时间越长，其响应比越高，因而它实现的是先来先服务。

·对于长作业，作业的响应比可以随等待时间的增加而提高，当其等待时间足够长时，其响应比便可升到很高，从而也可获得处理机。克服了饥饿状态，兼顾了长作业。

**5.时间片轮转调度算法（RR）**

时间片轮转调度算法主要适用于分时系统。在这种算法中，系统将所有就绪进程按到达时间的先后次序排成一个队列，进程调度程序总是选择就绪队列中第一个进程执行，即先来先服务的原则，但仅能运行一个时间片。

在时间片轮转调度算法中，时间片的大小对系统性能的影响很大。如果时间片足够大，以至于所有进程都能在一个时间片内执行完毕，则时间片轮转调度算法就退化为先来先服务调度算法。如果时间片很小，那么处理机将在进程间过于频繁切换，使处理机的开销增大，而真正用于运行用户进程的时间将减少。因此时间片的大小应选择适当。

时间片的长短通常由以下因素确定：系统的响应时间、就绪队列中的进程数目和系统的处理能力。

RR 是一种非常公平的算法。若一个时间片尚未用完进程便已经完成，就立即再调度就绪队列中队首进程运行，并启动一个新的时间片。如果在一个时间片用完时进程尚未运行完毕，则剥夺 CPU，调度程序把它送往就绪队列的末尾。

响应时间T = 时间片q × 就绪队列进程数n

## **多级反馈队列调度算法（集合了前几种算法的优点）**

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。

·多级反馈队列的优势有：

终端型作业用户：短作业优先。

短批处理作业用户：周转时间较短。

长批处理作业用户：经过前面几个队列得到部分执行，不会长期得不到处理。