基本思想

SJF算法是以作业的长度来计算优先级，作业越短，其优先级越高。作业的长短是作业所要求的运行时间来衡量的。

算法性能评价

面向用户

周转时间

从作业被提交给系统开始，到作业完成为止的这段时间间隔（作业在后备队列上等待时间、进程在就绪队列上等待时间、进程在cpu上执行时间、进程阻塞时间）

1. 周转时间=完成时间-到达时间

2. 平均周转时间：周转时间/进程数

3. 带权周转时间：周转时间/服务时间

4. 平均带权周转时间：带权周转时间/进程数

###### **响应时间**

从用户通过键盘提交一个请求开始，到系统首次产生响应为止的时间，包括键盘输入时间、信息处理时间和信息回送时间。

###### **截止时间**

某任务必须开始执行的时间（开始截止时间）和执行完成时间（完成截止时间）

以例题来了解



系统有如下进程，计算平均周转时间和平均带权周转时间

非抢占式调度

非抢占式调度(Non-preemptiveMode)进程一旦获得处理机，只有在该进程任务完成或因某事件而阻塞时，才让出处理机，决不允许某进程抢占已经分配出去的处理机。

方法

找出最先到达的进程（该进程的完成时间=到达时间+服务时间）；

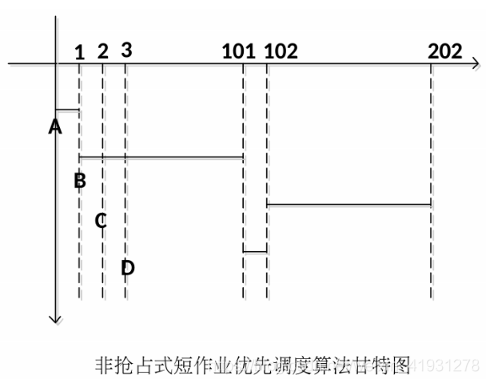
根据上一进程的完成时间，找到在这个完成时间内所有到达的进程，并找到这些进程中服务时间最短的那个，然后计算它的完成时间（该进程的完成时间=上一进程的完成时间+该进程服务时间）；

重复2直至完成所有进程的计算。

过程（A-B-D-C)



甘特图



平均周转时间：（1+100+99+200）/4=100

平均带权周转时间：（1+1+99+2）/4 =25.75

抢占式调度

抢占式调度(PreemptiveMode)允许调度程序根据某种原则，暂停某个占用处理机的进程，抢占已经分配出去的处理机。抢占的原则有优先权原则、短作业优先原则和时间片原则。

方法

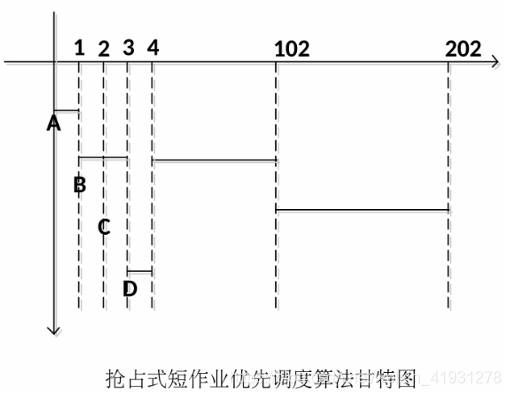
找出最先到达的进程（该进程的完成时间=到达时间+服务时间）

根据上一进程的完成时间，找到在这个完成时间内所有到达的进程，并找到这些进程中服务时间最短的那个，在进程运行中，有新到达的进程时，会比较两者的服务时间（该进程剩余的服务时间与新进程的服务时间进行比较，若小则继续运行，若大则暂停进程，新进程运行。）

重复2直至完成所有进程的计算。

过程

甘特图（A-B-D-C）



平均周转时间：（1+101+1+200）/4=75.75

平均带权周转时间：（1+1+99+2）/4 =1.25

结论

SJF特点是降低了系统平均周转时间。

但对长作业不利，长作业有可能长时间得不到调度。