2018010985 徐铭婕 第三次作业：处理机典型算法比较

1. 先来先服务算法（FCFS）
   1. 基本思想：按作业或进程到达的先后顺序进行调度，即每次在就绪进程中选择最先到达的作业或进程投入运行
   2. 特点：最简单、既能用于作业调度又能用于进程调度
   3. 有利于CPU繁忙型的作业，不利于I/O繁忙型作业；同时有利于长作业，不利于短作业
2. 短作业优先算法（SJF/SPF）
   1. 基本思想：按作业或进程估计运行时间长短来组织后备作业队列，每次选择运行时间最短的作业或进程投入运行，目的是提高系统吞吐率，减少进程的平均周转时间。后来的作业不抢占正在执行的作业
   2. 特点：适合作业调度和进程调度；对长作业不利，长作业会饥饿；该作业未考虑作业的紧迫程度；作业的估计执行时间值不一定准确
3. 高优先权优先调度算法
   1. 非抢占式优先权算法：主要用于批处理系统，一旦分配不可抢占
   2. 抢占式优先权算法：主要用于要求比较严格的实时系统，分配后可能被优先级更高的抢占
   3. 静态优先权：进程创建时确定，之后就不改了
   4. 动态优先权：进程创建时确定，但进程的优先数会随着进程推进过程或等待时间的增加而改变
4. 最高响应比优先调度算法（HRRN）
   1. 基本思想：HRRN算法同时兼顾每个作业的等待时间和运行时间两个方面的因素，每次在作业后备队列中挑选响应比最高的作业投入运行
   2. 响应比：响应时间/要求服务时间=（等待时间+要求服务时间）/要求服务时间
   3. 特点：有利于短作业（要求服务时间短）；实现了先来先服务（要求服务时间相同时，等待时间长）；对于长作业，其优先级随着等待时间增加而提高；既照顾了短作业，又考虑了作业到达次序，也不会使得长作业饥饿
5. 时间片轮转调度算法（RR）
   1. 基本思想：把CPU处理时间分成固定时间的时间片，每次调度时，把CPU分配给队首进程，时间片用完时，中断并把该进程送往就绪队列的末尾，再把CPU分给新的队首
   2. 特点：时间片过长->进程可能在一个时间片内就做完了，CPU空转；时间片过短->进程上下文切换次数过多，系统资源开销大
6. 多级反馈队列轮转法（RRMF）
   1. 基本思想：有多个就绪队列，优先级高的队列分配的时间片最小，随着队列级别增加其时间片也增加；支持抢占式；该时间片内某进程运行不完，就给它降级；同级别里先来先服务
   2. 特点：有利于交互性作业用户（第一队列），短批处理作业用户（前面的几次降级就做完了），长批处理用户（每一级都能运行，不会饥饿）