# 操作系统复习

第六章： CPU调度

第六章总纲：

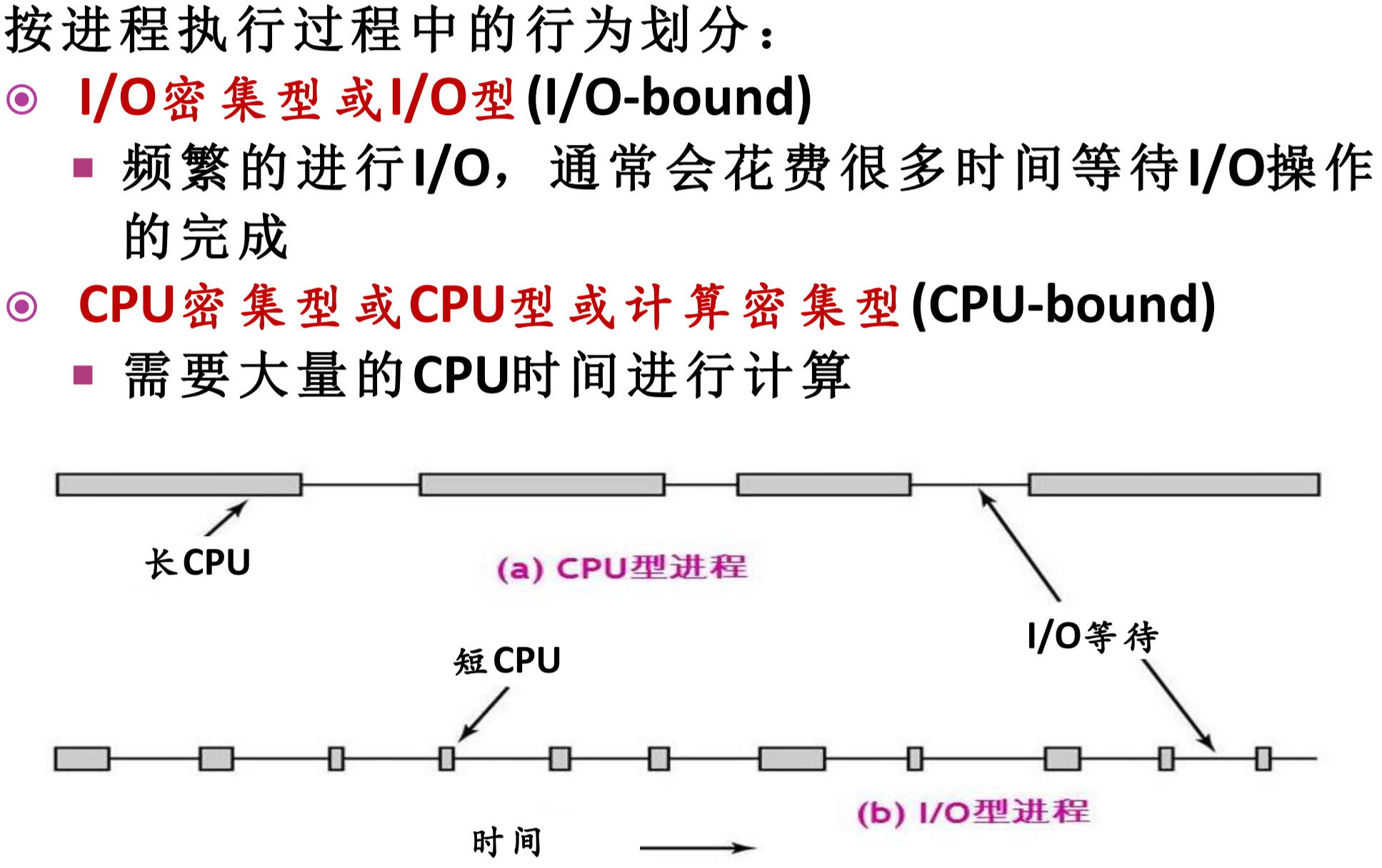
* Basic Concepts
* Scheduling Criteria
* Scheduling Algorithms
* Thread Scheduling
* Multiple-Processor Scheduling\*
* Real-Time CPU Scheduling
* Operating Systems Examples
* Algorithm Evaluation\*

**#基本概念**

CPU调度是多道程序操作系统的基础。通过在进程之间切换CPU，可以提高计算机的吞吐量。

**\*CPU区间和IO区间**

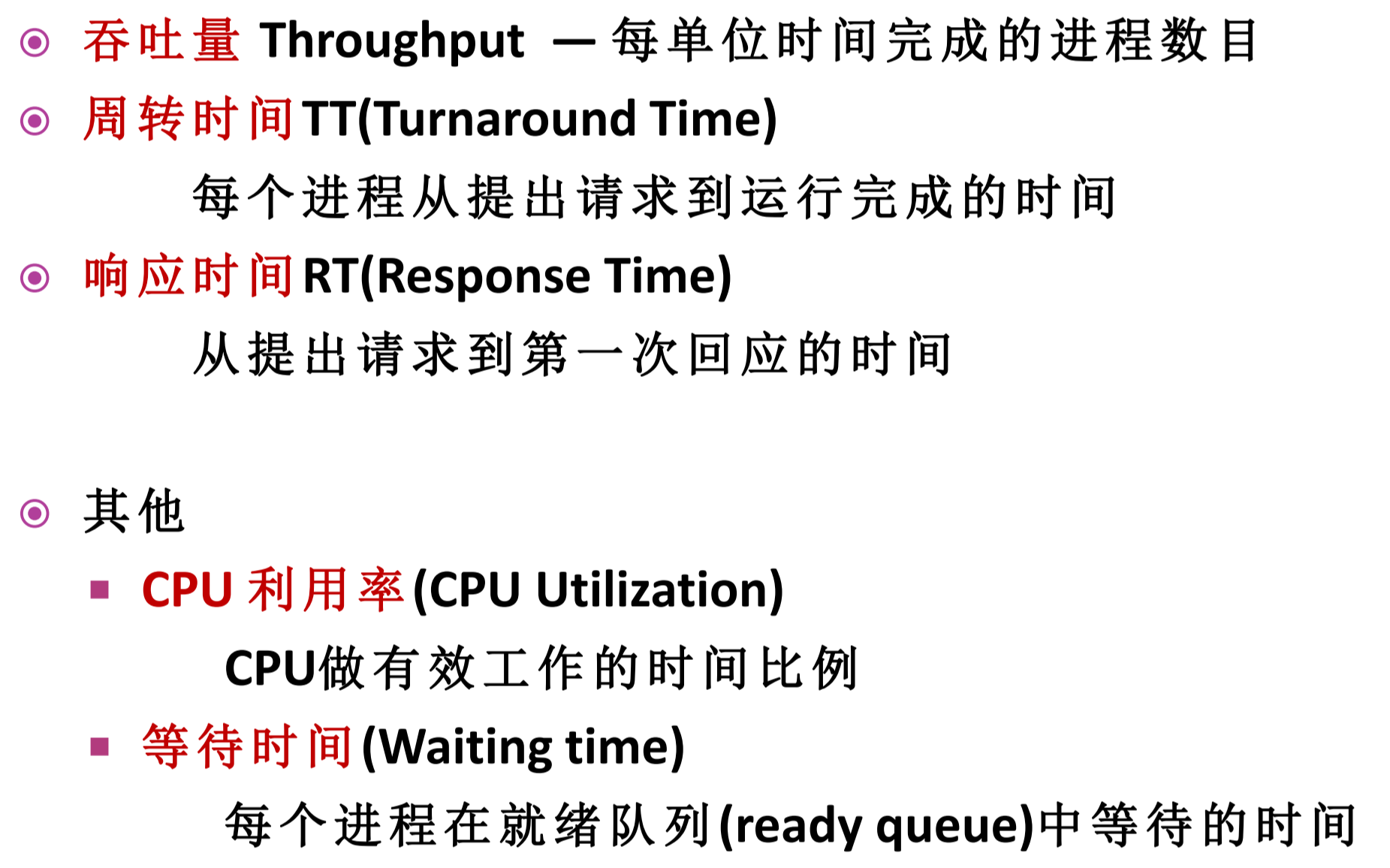
计算机进程执行是由CPU执行和IO等待周期组成。进程在者两个状态切换，由CPU开始，由CPU结束。



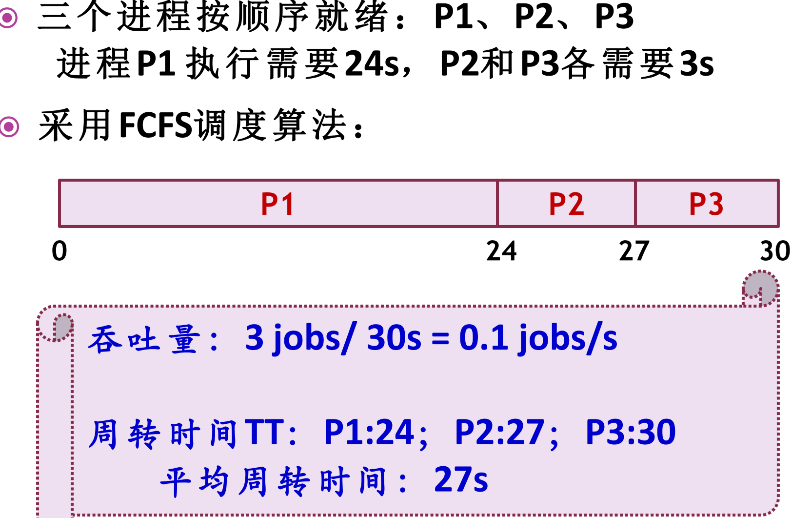
**\*CPU调度发生情况：**

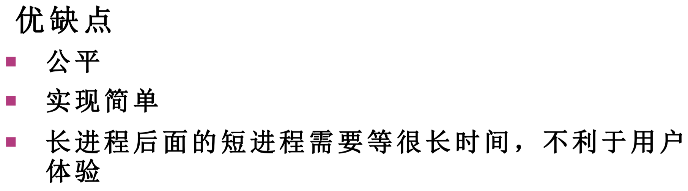
* 1. 当一个进程从运行状态切换为等待状态
  2. 当一个进程从运行状态切换为就绪状态
  3. 当一个进程从等待状态切换为就绪状态
  4. 当一个进程终止

**#调度准则Scheduling Criteria**

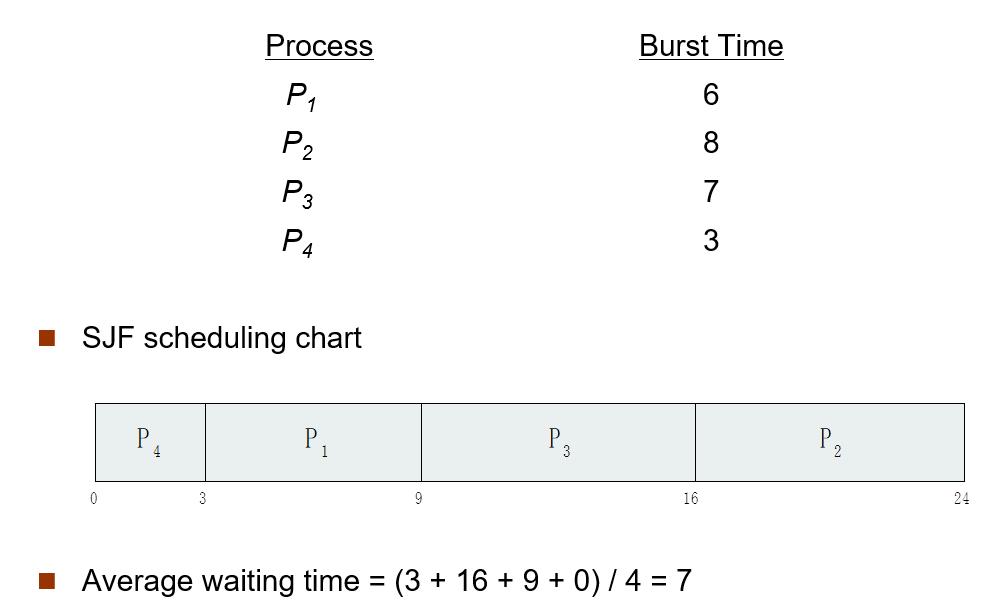
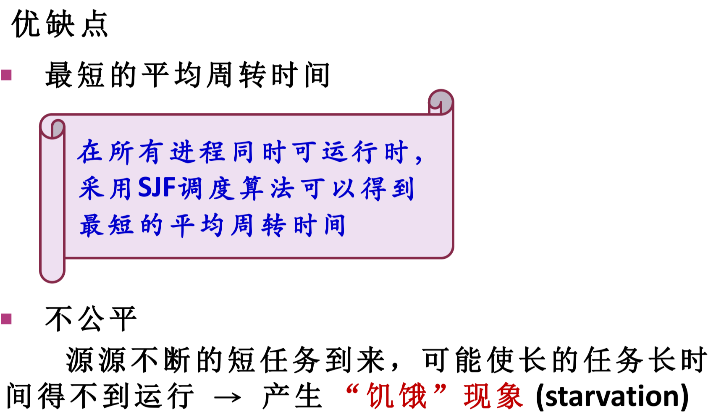


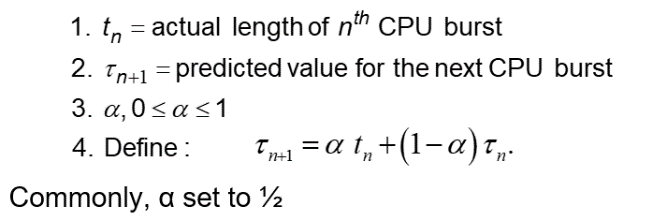
**#调度算法**

**\*FCFS先到先服务**

如同字面意识，先到的进程优先调入CPU

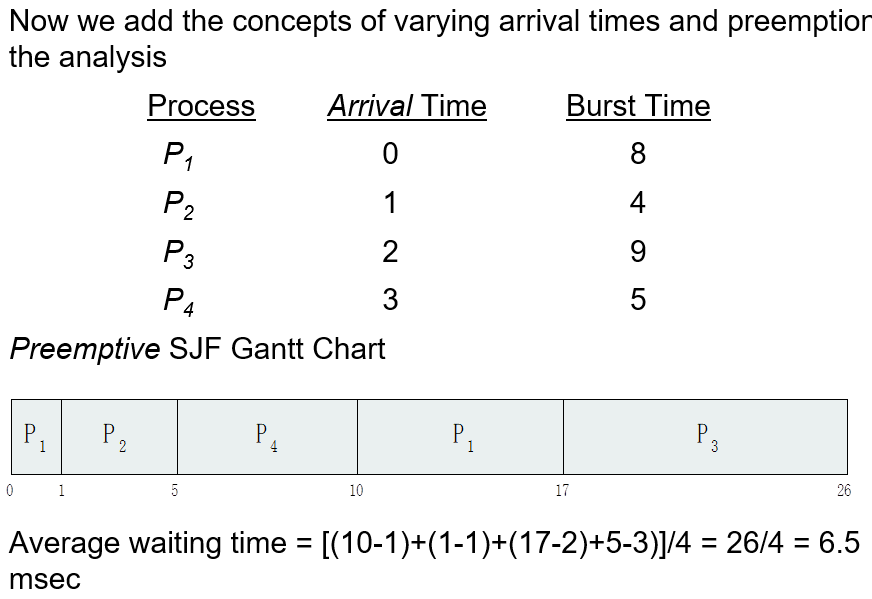
**\*SJF短作业优先**

很理想的算法，平均等待时间第二少，但具体无法实现。

注：其中真正困难是如何知道下一个CPU区间的长度。

其中之一的算法：----------------------🡪🡪

**\*SRTF最短剩余时间优先调度**

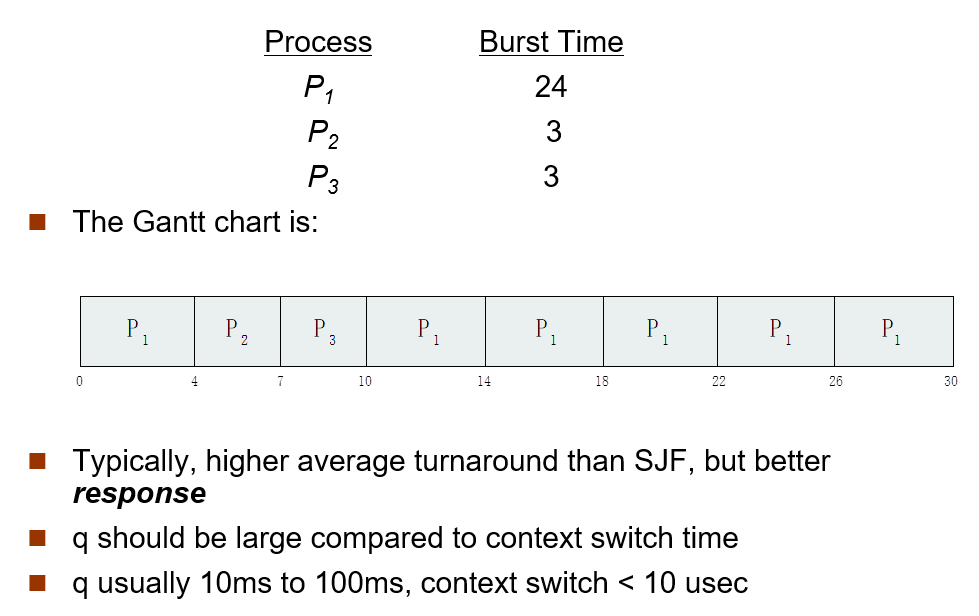
这种算法是抢占式的SJF算法，结合到达时间和区间时间，来抢占CPU，让当前区间时间最小的进程进入CPU。

**\*优先级调度**

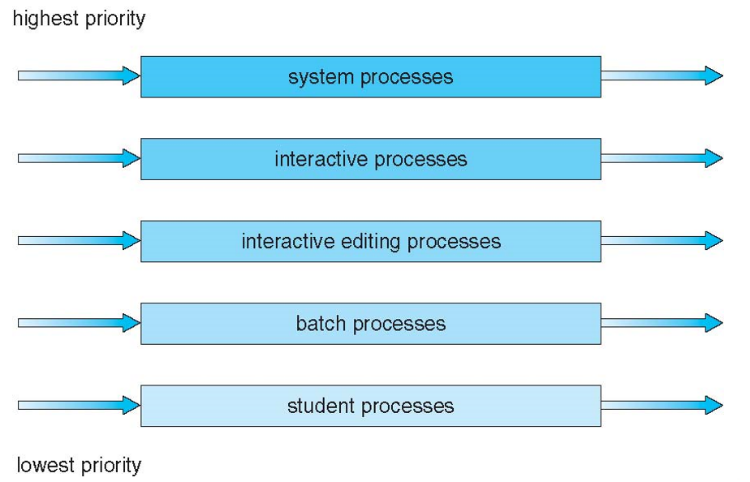
具有高优先级对的进程享有优先进入CPU的权利。

但是也有一些问题：**无穷阻塞**和**饥饿————低优先度的进程不能运行**

**\*RR轮转法调度**

在FCFS基础上，定义一个时间片time quantum，一个进程只能在一个时间片内运行，不论进程是否结束，在时间片结束后都将切换进程，没执行完的进程将添加至等待队列末尾。

**\*多级队列调度**

在进程可容易地分为不同组的情况下，可以建立另一类调度算法。

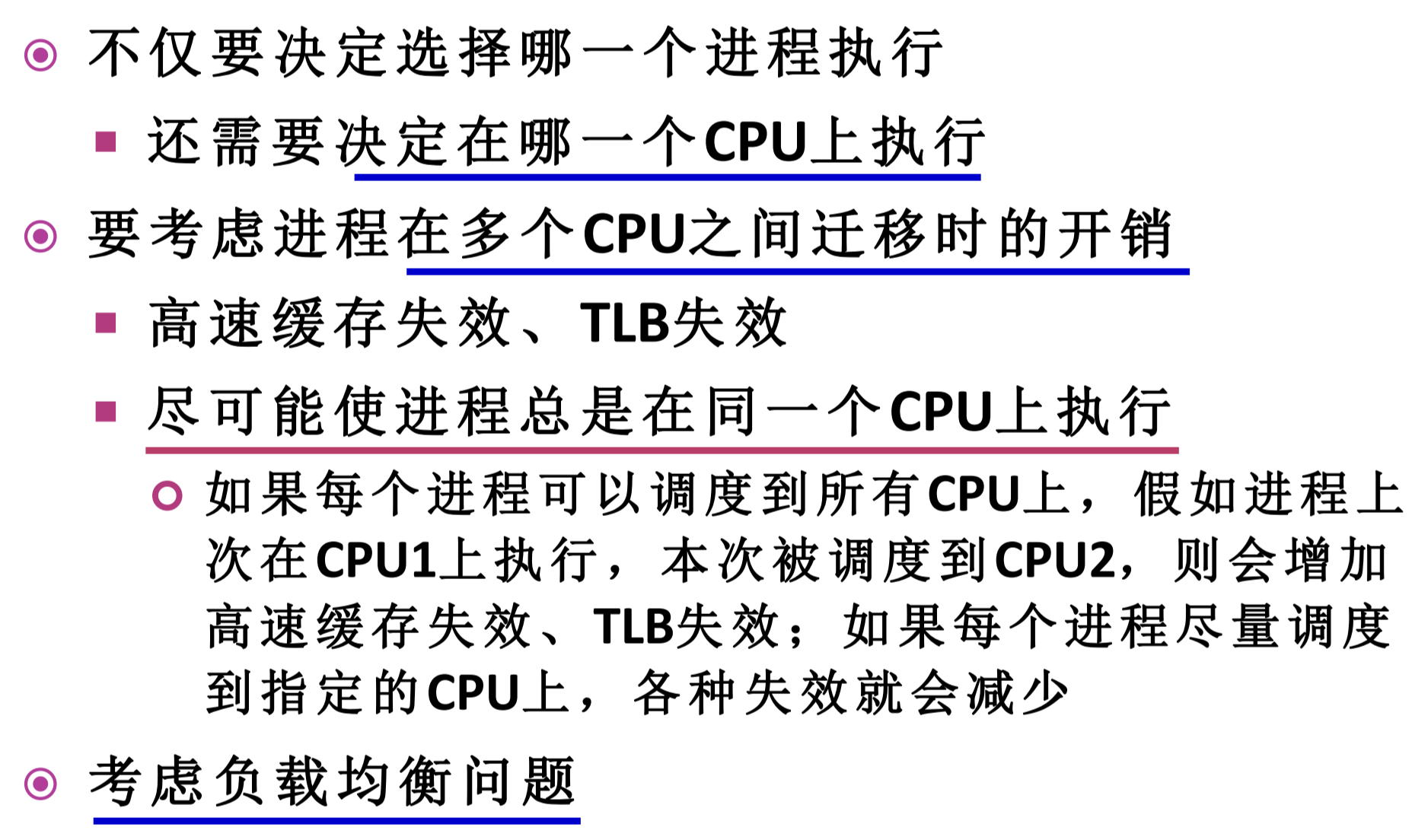
**\*多级反馈队列调度**

允许进程在队列之间移动。

#线程调度

？？？？

**#多处理器调度**



**Symmetric multiprocessing(SMP)对称多处理**：每个处理器自我调度。

**Processor affinity处理器亲和性**：避免一个进程从一个处理器移至另一个处理器，努力使一个进程在同一个处理器上运行。

**Load Balancing负载平衡**：设法将工作负载平均地分配到SMP系统中的所有处理器上。方法有push migration, pull migration.

#Real-time scheduling

???