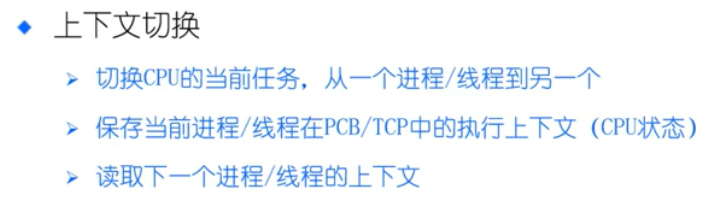
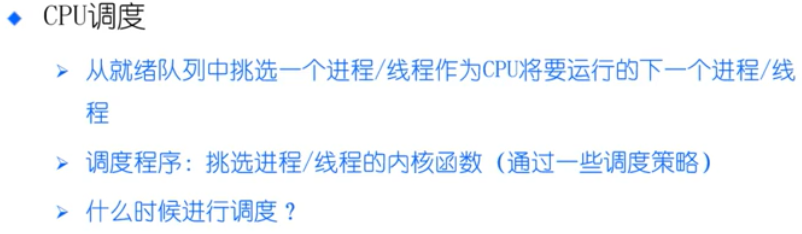
背景

CPU调度

CPU调度时间

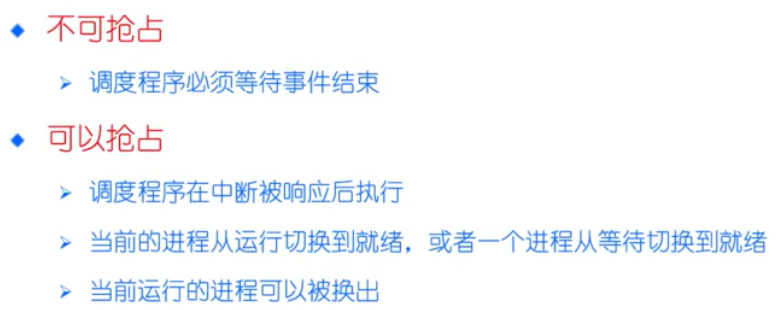






在生命周期中当一个状态切换到另一个状态时就会发生调度，特别是和一些运行相关的状态变化：ready-running、running-waiting(blocked)、running-done





内核的不可抢占？？

调度准则

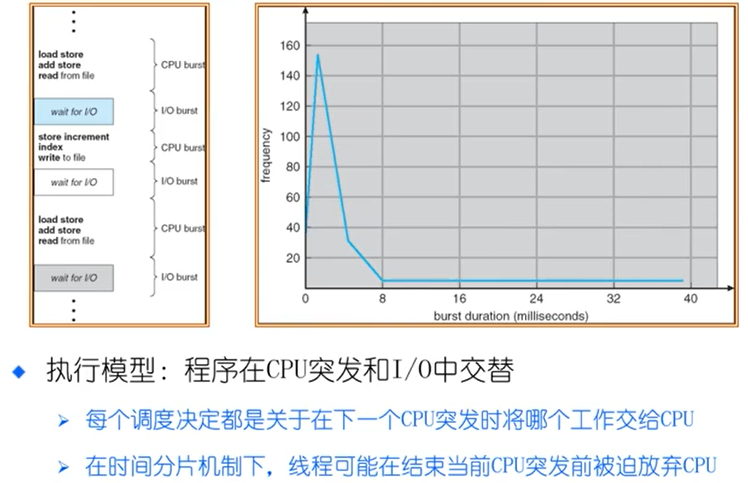
调度策略

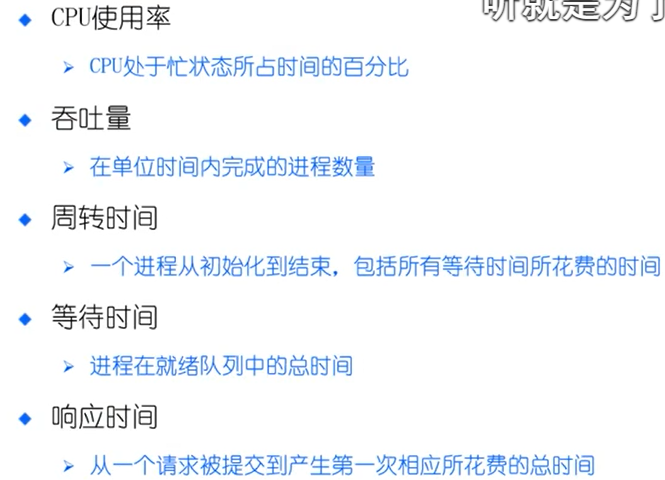
程序执行模型

比较调度算法的准则

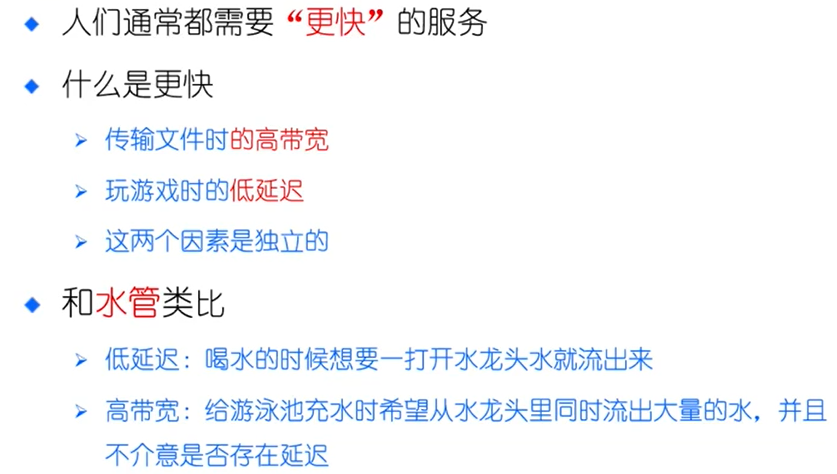
吞吐量vs延迟

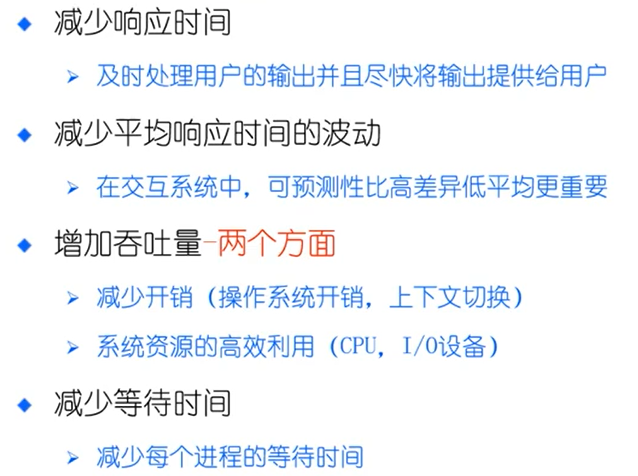
公平的目标

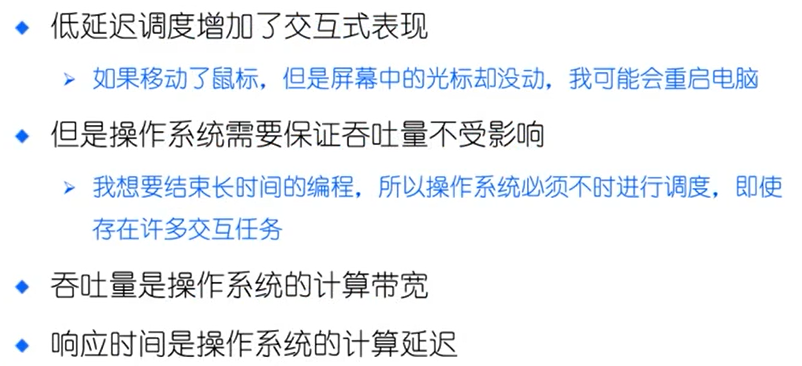


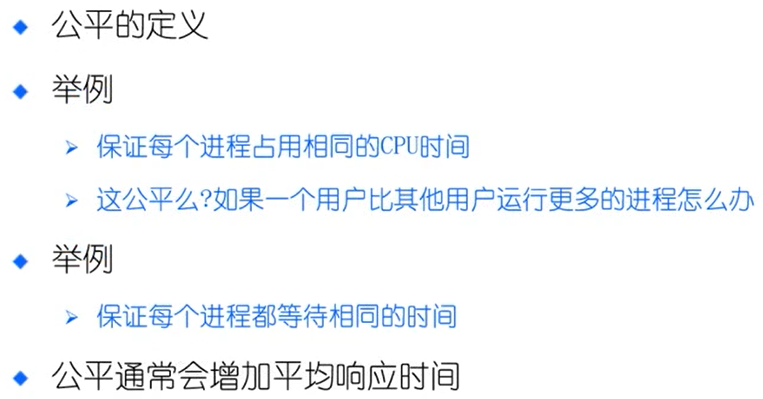
《评价指标》

周转时间 = 服务时间+等待时间



希望算法的指标有点矛盾

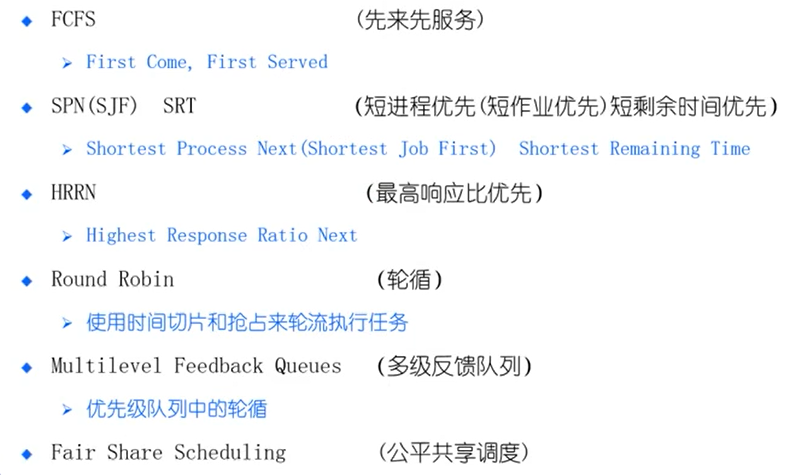




公平：尽量让每个进程分到的CPU资源都多

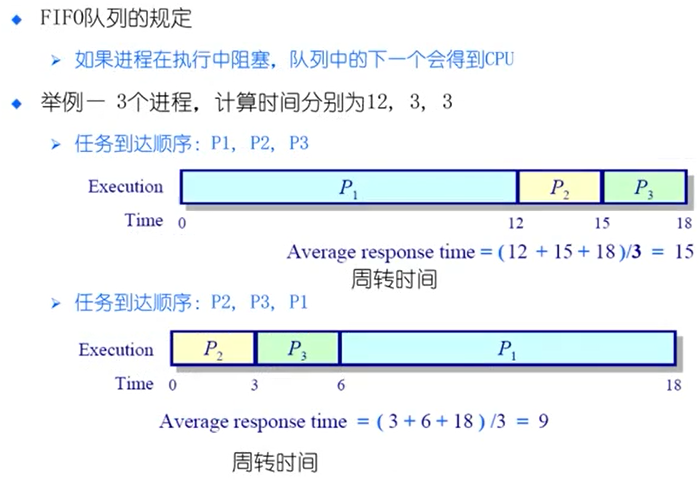
调度算法

常用的算法如下

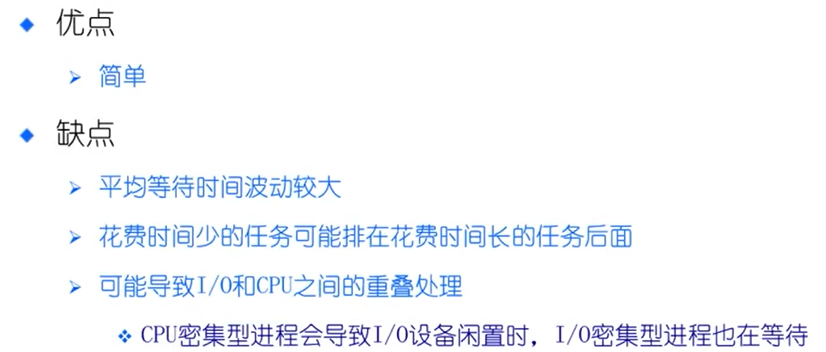


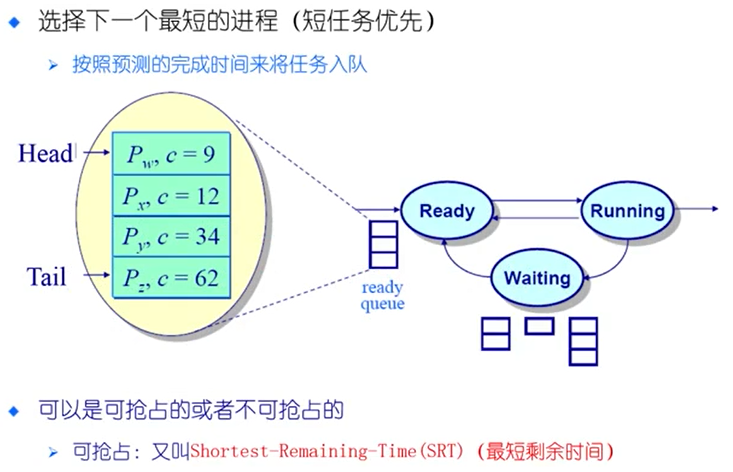
FCFS，first come first served

这里打错了，Average response time是平均响应时间而不是周转时间，下面公式中文是对的，英文是错的，公式对应的是周转时间



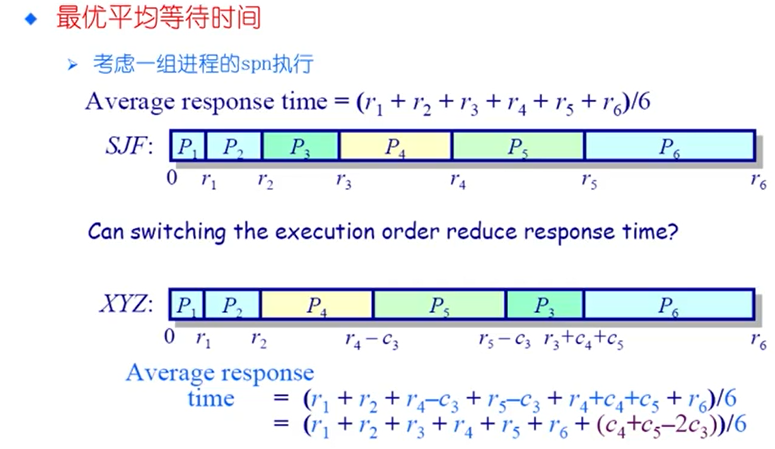
周转时间=等待时间+处理时间



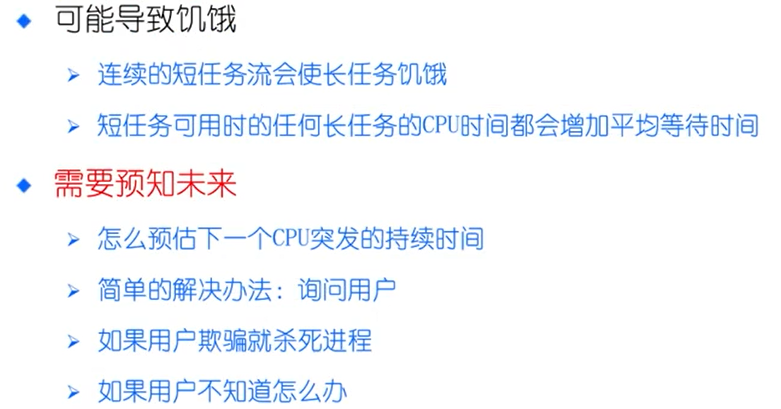


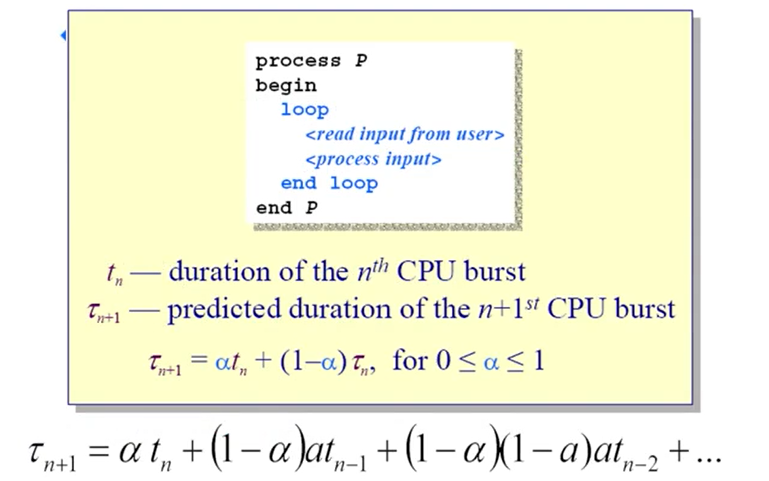
这个是进程的执行时间决定了它的优先级

SRT和SPN(SJF)的区别在于是否考虑了抢占

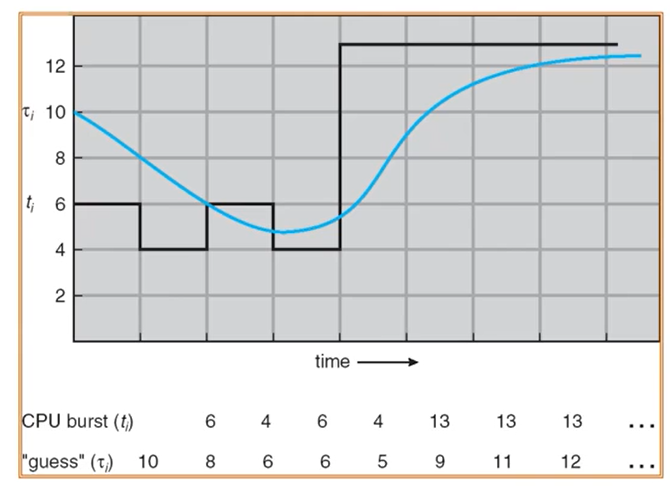
C是进程执行时间

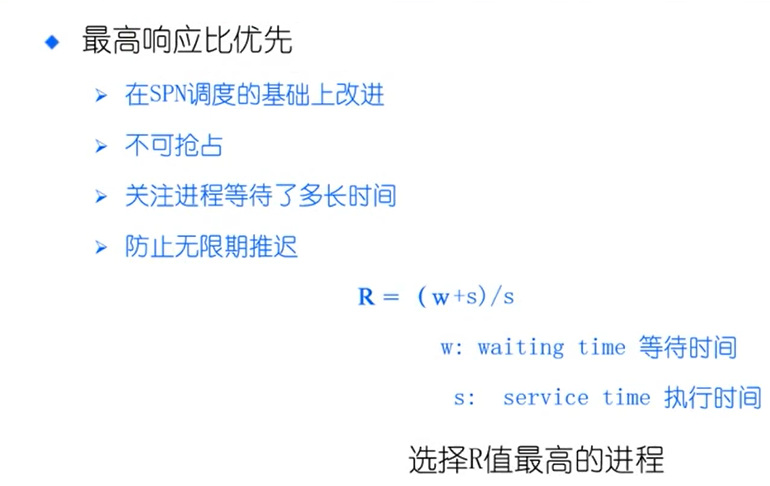
缺点：：





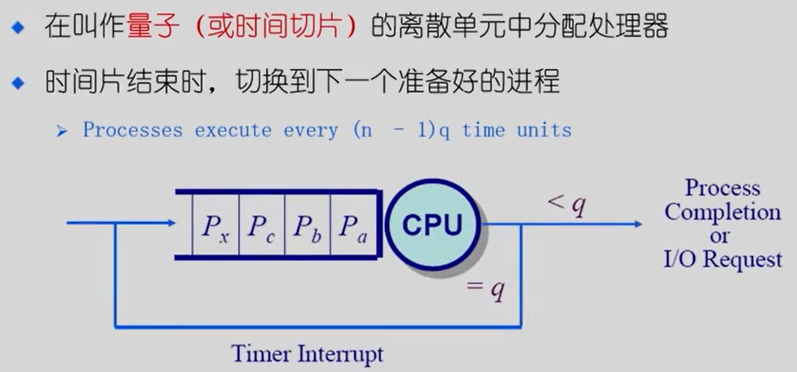
预估进程的执行时间

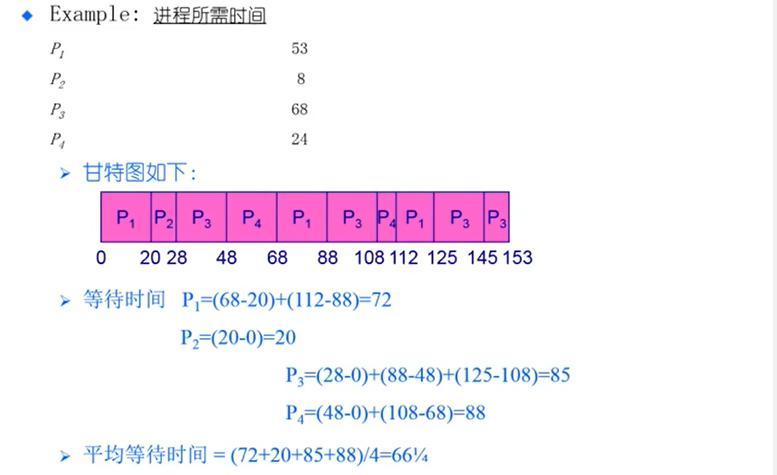




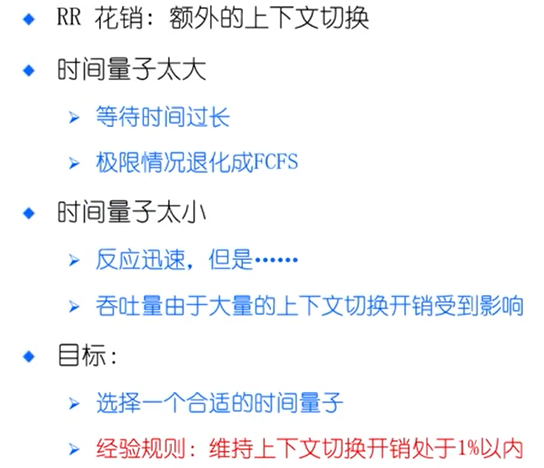
不考虑抢占的。并且要知道进程的执行时间（要预估）

轮询调度算法：让各个进程轮流去占用CPU





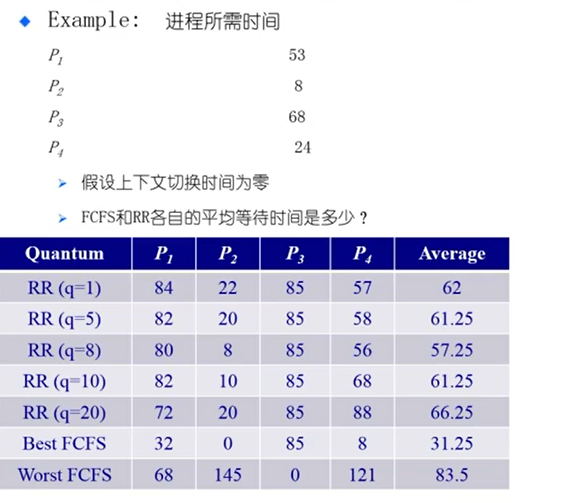
时间片的设置很重要，这里的平均等待时间略大



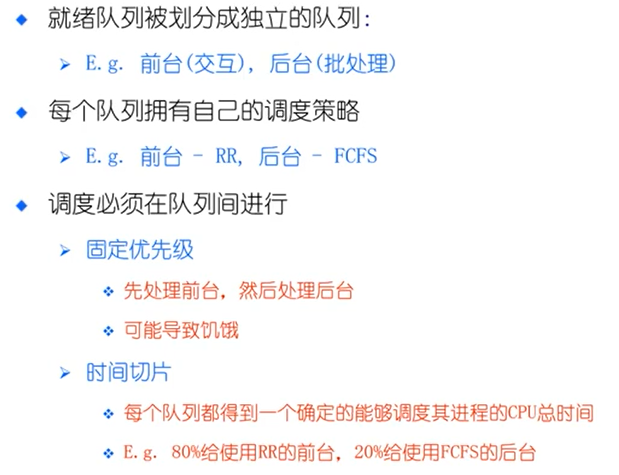
切换开销大

时间片的设置主要靠经验，一般把上下文开销维持在1%内

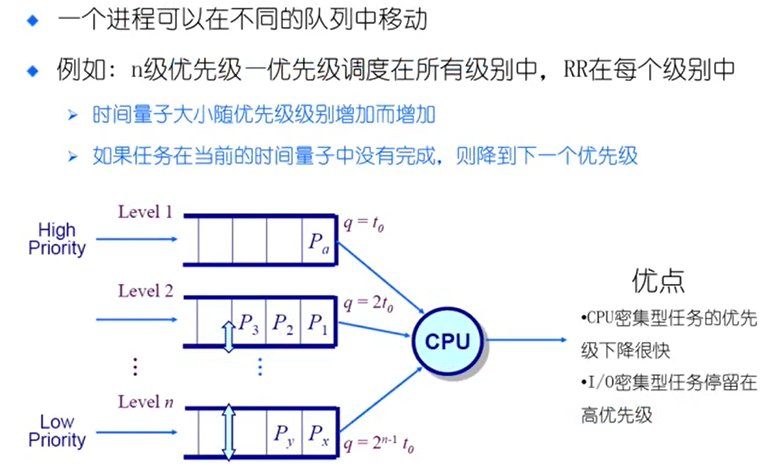
例子：：



有时候FCFS好一些的原因是它没有频繁的上下文切换，但是它没有RR这么公平



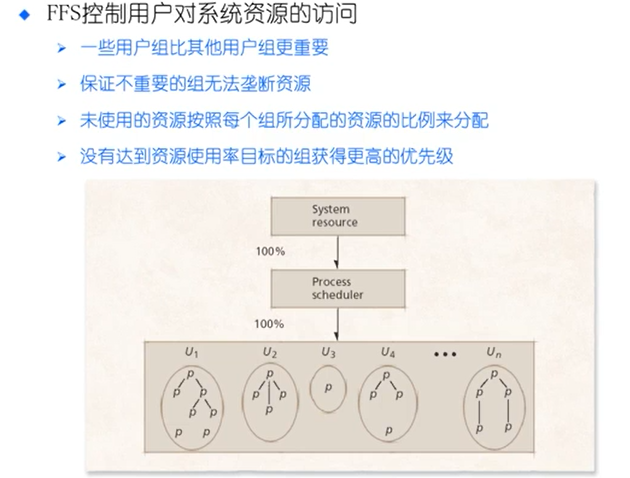
多级队列，但没有考虑进程的动态性



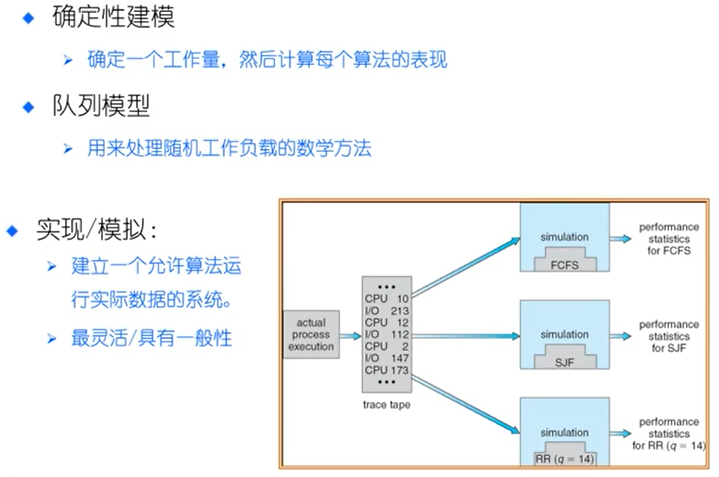
多级反馈队列

高优先级需要大量的交互，时间片要短一些；

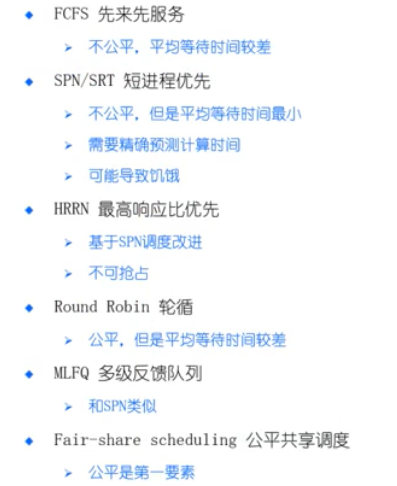
公平共享调度



公平共享调度

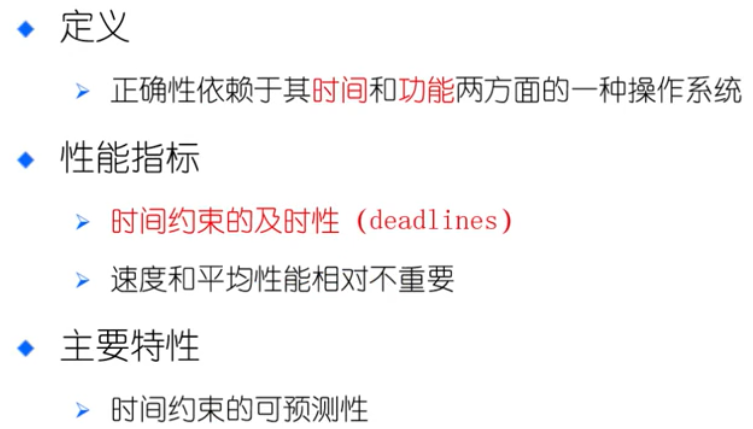


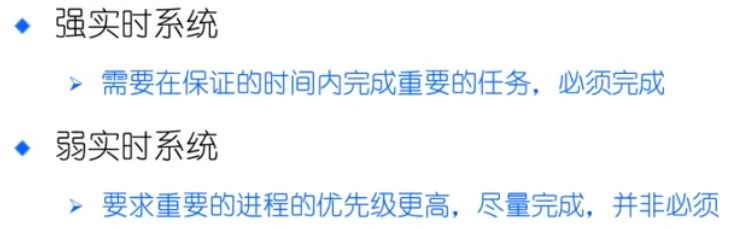
评价调度算法，还与硬件有关

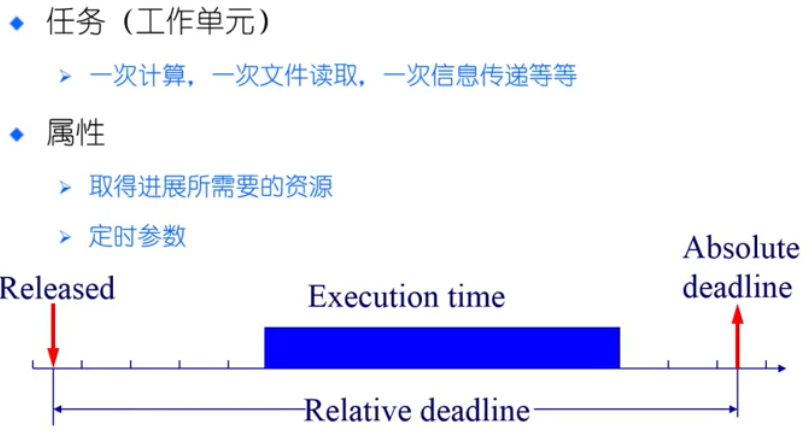
《总结》

实时调度

主要针对实时系统：确定性和可预测性





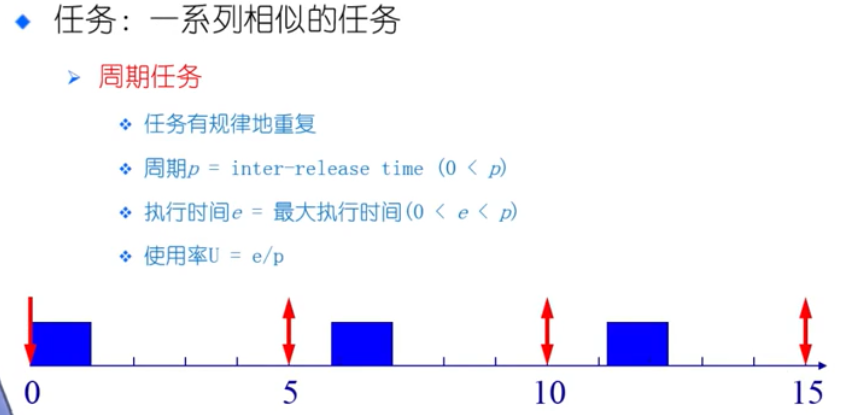


用一些术语来描述进程的实时性性质

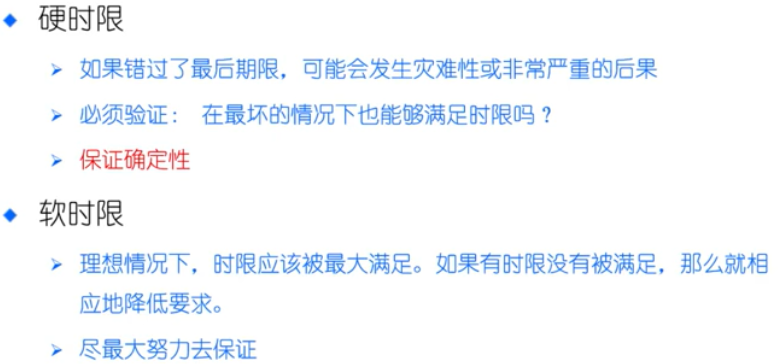
Released：发起进程，进程就绪

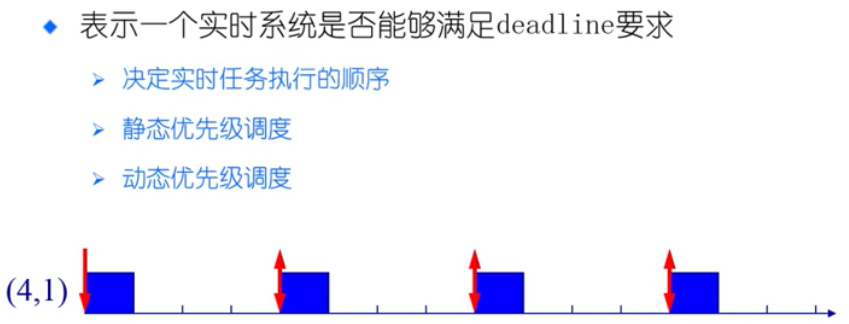
Relative deadline：相对deadline，范围性质

Absolute deadline：最终deadline



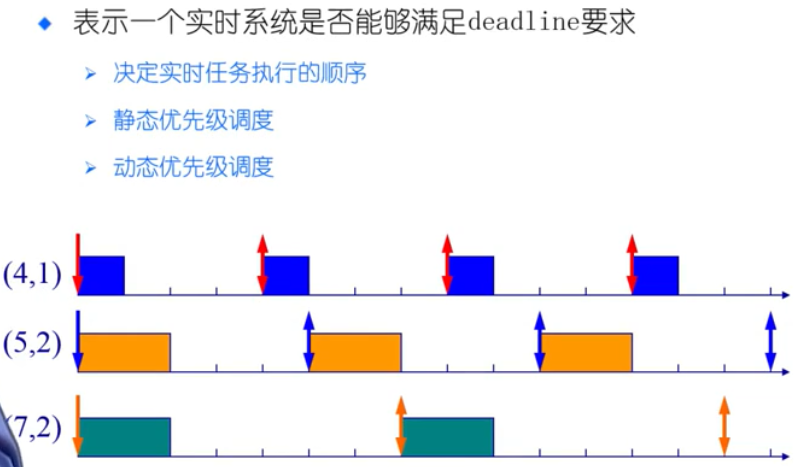
例子，执行时间一定要在一个周期内完成

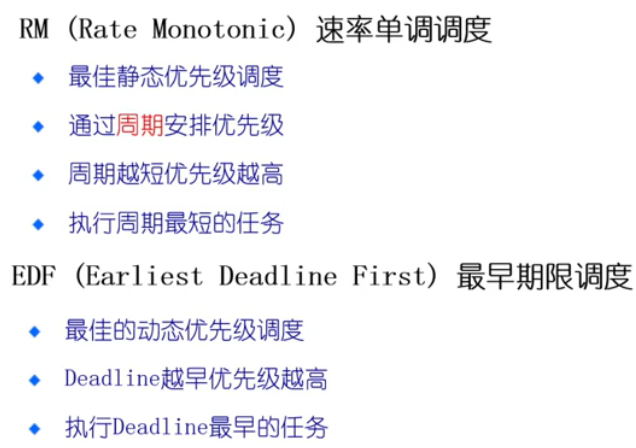




为了满足一个进程的硬实时和软实时的要求，算法可以分为静态优先级调度和动态优先级调度。静态是指在任务执行之前，优先级就已经确定了，按优先级来选择任务，再按照规定时间内去完成任务。动态是指随着执行的过程，优先级会发生动态变化，在不同时刻的优先级有所不一样。

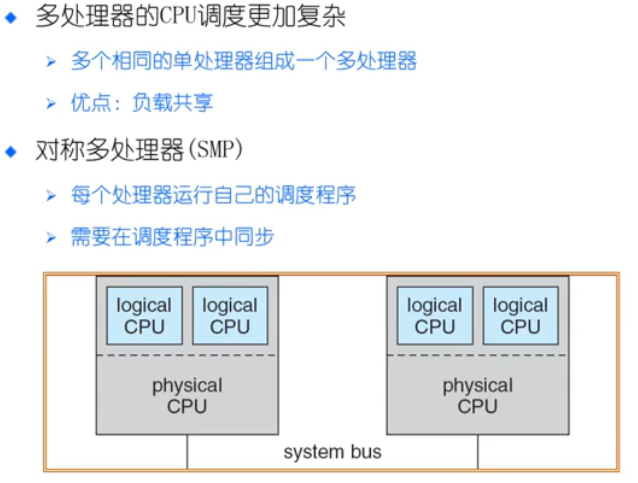
FCFS就是静态调度。RR就是动态调度。

《例子》



RM是静态调度算法，EDF是动态调度算法。

多处理器调度



同步是比较难的？

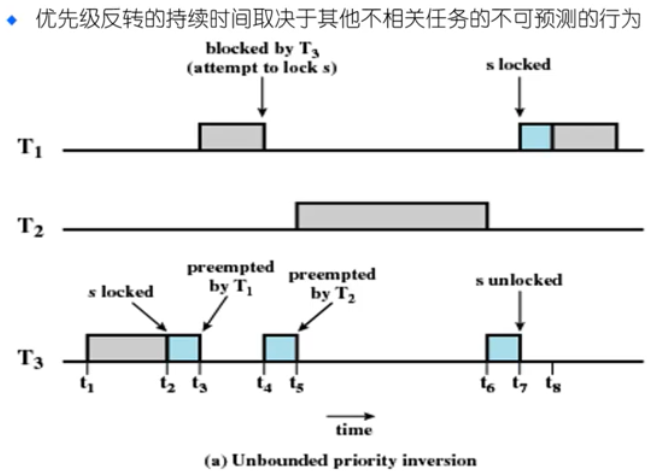
现在有多个CPU，来了一个进程，那么这个进程要在那个CPU上面执行呢？

多核负载平衡

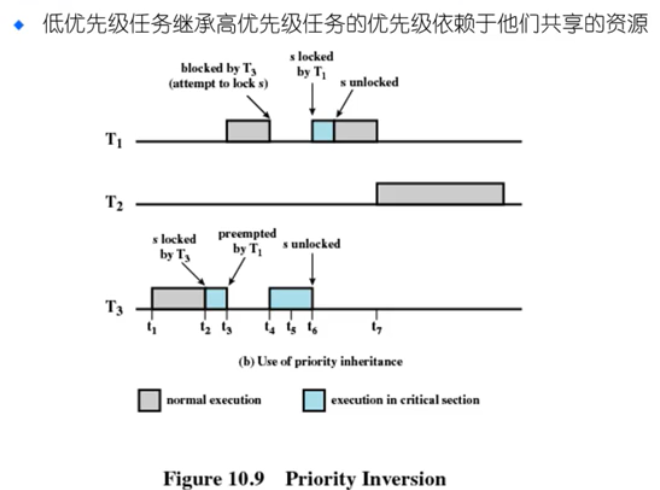
优先级反转



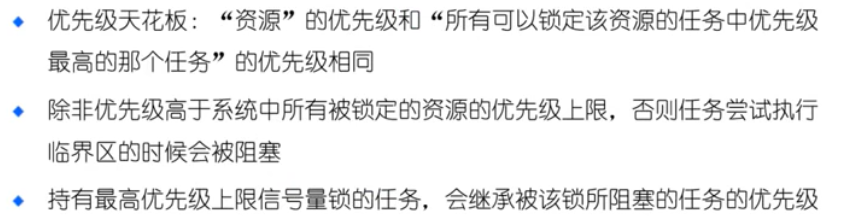
第二句的“强制”



T3先执行到t2时刻，访问共享资源到t3，T1在t3时刻出现了，于是优先级较高的T1执行到t4时刻，此时T1想要访问共享资源，但是共享资源被T3占有了，于是T1进入等待，于是T3重新由t4执行到t5，这时候T2来了，T3进入等待让优先级更高的T2开始执行，T2执行到t6时刻时才执行完，然后又回到T3继续访问T3想要的共享资源，等T3释放后才能给T1。这里我们看到T2竟然比T1先执行了。



就是使得T3在t4时刻的优先级提升到和T1的优先级一样，这样即使T2再来也无法抢占T3的执行资源。



优先级天花板要提前统计