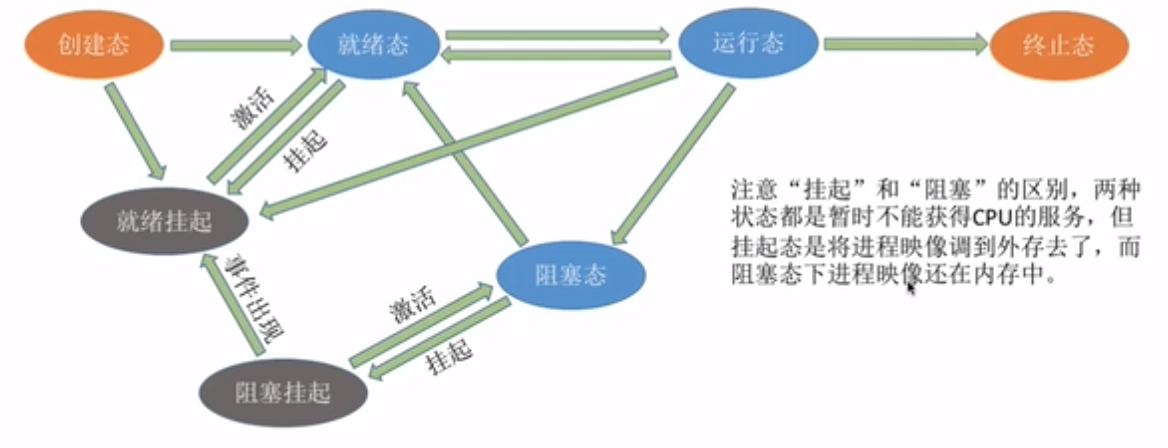
# 处理机调度

高级调度：（作业调度）按照一定的原则从外村上的后被队列种选一个或者多个作业，给他们分配内存等必要资源，并且建立相应的PCB 使他们获得竞争处理机的权限（指调入过程）

中级调度：（内存调度）为了提高内存的利用率和吞吐量，操作系统会将暂时不用的进程调到外存（PCB不动，仅仅是其资源代码等载入外存Swap换出）这些进程出于挂起态，操作系统会将这些PCB放到挂起队列中，并且等有需求的时候通过中级调度决定哪个被挂起的进程换入内存。

（可能发生多次中级调度）



低级调度：（进程调度）按照某种方法从就绪队列中选取一个进程，将处理机分配给它（最基本的一种调度，每当进程切换的时候就进行调度，正常在几十毫秒就进行）

# 进程调度（低级调度）的时机

需要发生进程调度与切换的情况

1. 进程主动放弃处理机
   1. 进程正常结束
   2. 进程出现异常
   3. 进程发生IO请求，主动进入阻塞态
2. 进程被动放弃处理机
   1. 时间片结束
   2. 有更高优先级的进程进入就绪队列
   3. 有更紧急的事件需要处理，比如IO中断

（不能进行进程调度和切换的情况）

1. 中断处理的过程中2.进程在操作系统内核程序临界区中3.原子操作

这里的内核程序临界区是用来访问某种内核数据结构的，比如说进程的就绪队列

在访问的过程中，一般会先对就绪队列进行上锁，这时候进程调度就无法访问就绪队列，就无法进行进程调度和切换，但是对于一些非内核程序临界区的程序，比如访问打印机的程序，是可以被中途切换的，这里也不涉及对于就绪队列的上锁。

所以进程处于临界区是不能被处理及调度是不对的

进程调度方式

非剥夺调度方式（非抢占式调度方式 ）&剥夺调度方式（抢占式调度方式）

非抢占式调度处理方式：只允许进程主动放弃处理机，即只能在进程正常退出，发生异常，IO阻塞的时候放弃，一般适用于早期的调度系统

抢占式调度处理方式，除了进程主动放弃处理机，当有更紧急或者优先级更高的任务来的时候，或者时间片切换的时候，也可以放弃处理机

进程切换一般完成：

1. 对原来的数据进行保存
2. 对新的进程的数据进行恢复

一般将程序状态字，程序计数器，各种寄存器信息和现场信息保存在PCB

（进程切换是由代价的，所以不是进程切换越频繁效率越高）

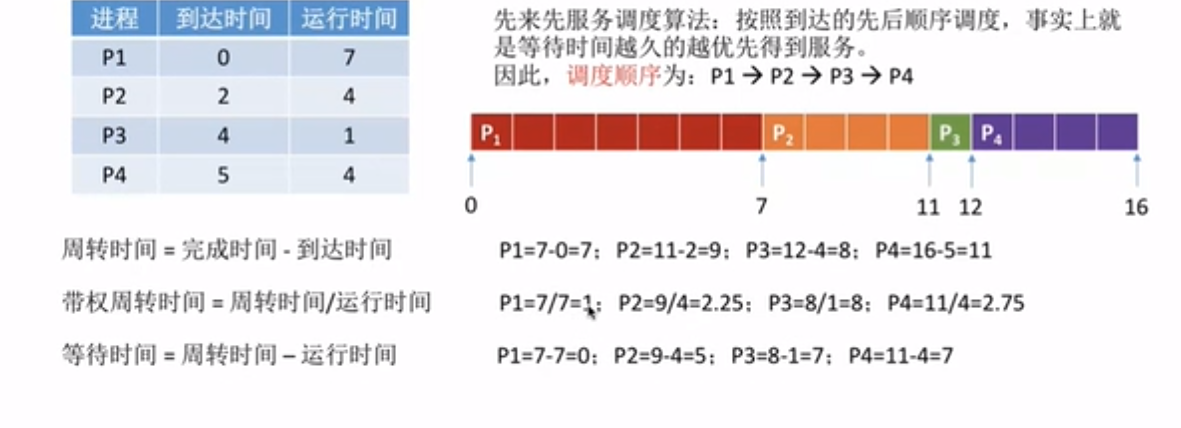
评价调度算法的指标

1. **cpu利用率=cpu忙碌时间/总时间**
2. **系统吞吐量 单位时间内完成作业的数量 = 总共完成了多少作业/总时间**
3. **周转时间 从作业被提交给系统开始到作业完成=完成时间-提交时间**
4. **平均周转时间 各个作业周转时间之和/作业数**
5. **带权周转时间 作业周转时间/实际运行时间 运行时间越长 带权周转时间越短**
6. **平均带权周转时间 各个作业的带权周转时间/作业数**
7. **等待时间 进程被建立起之后的等待时间之和 （对于作业的等待时间还要加上其在外存中的等待时间）周转时间-运行时间（纯计算进程），如果有IO操作，还要减去IO时间**
8. **平均等待时间**
9. **响应时间 从提出请求到首次相应的时间**

**调度算法**

# 先到先服务FCFS

按照作业/进程到达的先后顺序进行服务

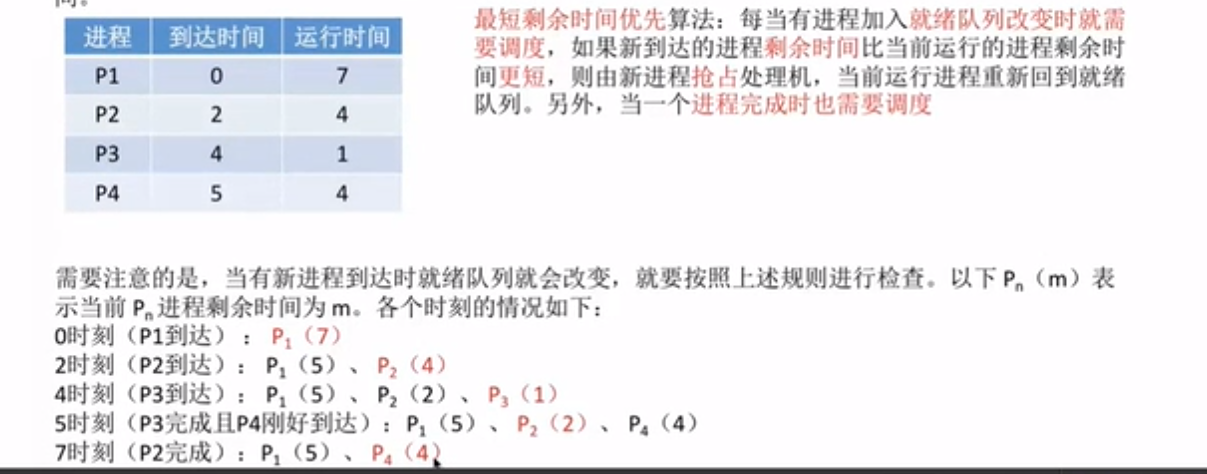


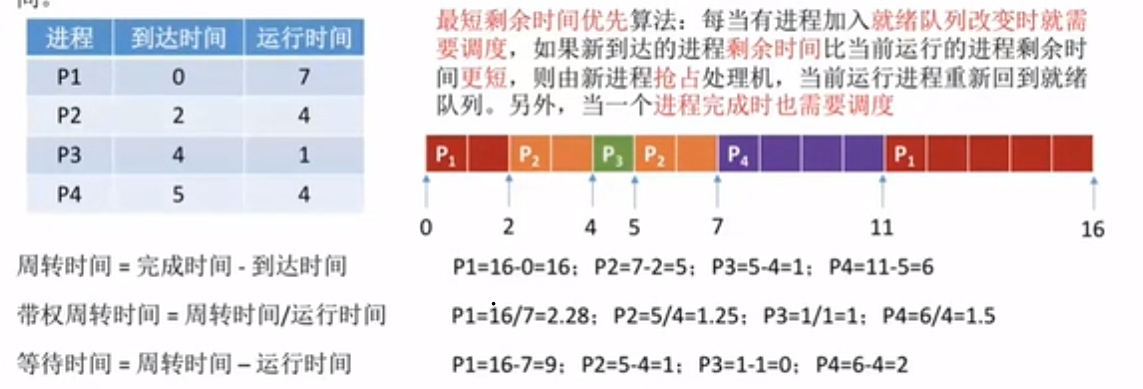
对长作业有利，对短作业不利，但是不会产生饥饿

# 最短作业优先 SJF Shortest Job First

每次调度的时候都只选择已经准备调度的最短的作业进行调度

抢占式短作业优先STRN 最短剩余时间优先算法 也就是有新进程载入之后，判断其需要用的时间是否小于正在运行进程的剩余运行时间，如果小于则抢占运行。





所有进程都是同时到达时SJF算法的平均等待时间，平均周转时间是比较短的

缺点：对于短作业有利，长作业不利，导致饥饿

# **高响应比优先算法**

要综合考虑作业/进程的等待时间和要求服务的时间

每次调度的时候都计算各个作业/进程的相应比，选择相应比最高的作业/进程并且为其服务

**相应比=（等待时间+要求服务时间）/要求服务时间 >1 服务时间即运行时间**

**非抢占式算法**

**综合考虑了等待时间和要求服务时间**

**当要求服务时间相同的时候，等待时间长的优先 FCFS优点**

**当等待时间相同的时候，要求服务时间短的优先 SJF优点**

**兼顾了 FCFS 和 SJF算法的优点**

**//没有优先级和紧急程度**

**//适用于早期的批处理系统**

# **时间片轮转RR Round-Robin**

按照各个进程到达就绪队列的顺序，轮流执行一个时间片的时间长度

是一种抢占式的算法

由时钟中断来切换 主要要求系统的响应时间

**如果时间片太大，使得每个进程都在时间片内执行完成，那么就会退化为FCFS算法，所以时间片不能太大。**

**如果时间片太小，那么系统会用大部分时间进行时间切换。**

**所以时间片大小要适中，不能太大太小**

**（不区分紧急程度，并且不会产生饥饿）**

# **优先级调度算法**

抢占式/非抢占式优先级调度算法

抢占式的，要检查当前到达的进程是否优先级要更高

静态优先级：创建时确定，之后不发生改变

动态优先级：创建时有初值，运行过程中会视情况而改变

系统进程优先级高于用户进程优先级

前台进程优先级高于后台进程优先级

操作系统更加偏好IO繁忙进程 让IO设备提早工作，因为IO比较慢，对整体的性能提升有帮助，相比之下计算型进程优先级低