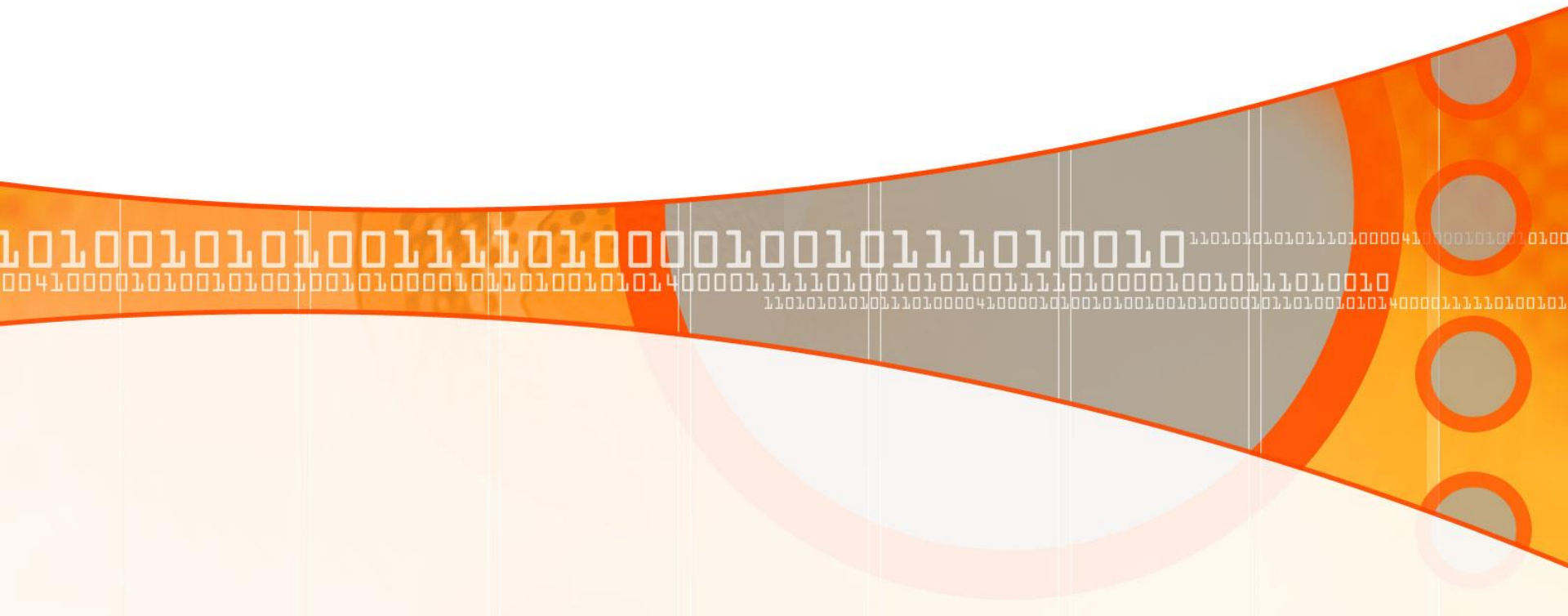
第3章 处理机调度与死锁



本章重点

1、掌握处理机调度的基本概念和调度算法。

2、掌握银行家算法避免死锁的方法。

3.1 处理机调度的层次和调度算法的目标

作业和进程

①作业是用户向计算机提交任务的任务实体。

进程是完成用户任务的执行实体，是资源分配的基本单位。没有作业任务，进程无事可干；没有进程，作业任务没法完成。

②作业建立完毕后，是放在外存等待运行。

进程一经创建，总由相应的部分存于内存。

③一个作业可由多个进程组成，且必须至少由一个进程组成，反之则不然。

④作业的概念更多地用在批处理系统中。

进程的概念几乎可以用在所有的多道程序系统中。

批处理作业的调度

- **作业调度：使作业进入主存储器。**
- **处于后备状态的作业在系统资源满足的前提下可以被作业调度选中进入内存计算。**
- **进程调度：使作业进程占用处理器。**
- **只有处于执行状态的作业才真正构成进程获得计算的机会。**

3.1 处理机调度的层次和调度算法的目标

一个批处理型作业，从进入系统并驻留在外存的后备队列上开始，直至作业运行完毕，可能要经历下述三级调度。

1. 高级调度（High Scheduling）
2. 中级调度（Intermediate-Level Scheduling）
3. 低级调度（Low Level Scheduling）

3.1 处理机调度的层次和调度算法的目标

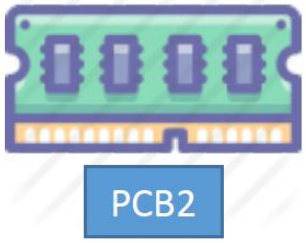
- 调度的概念：在多道程序环境下，进程数目往往多于处理机数目。要求系统能按某种算法，动态地将处理机分配给就绪队列中的一个进程，使之执行。
- 现代OS的运行性能，如吞吐量、作业平均周转时间、响应的及时性等，在很大程度上取决于调度，因而，调度策略成了OS的一个关键问题。

三个层次
- 高级调度（作业调度）
- 中级调度（内存调度）
- 低级调度（进程调度）

- **高级调度**
  - **又称作业调度或长期调度（long-term scheduling）。**根据某种算法，决定把外存上处于后备队列的哪些作业调入内存
    - 作业：（用户）利用计算机进行一次运行所需工作的集合。要完成一个工作，用户必须先提交一个作业。如一个作业可能由多个程序构成。
    - 在PC机或普通工作站和服务器上几乎没有作业的概念
    - 在巨型机和大型服务器上，主机的速度高且造价高，要保证其利用率和效率，用专门的作业控制软件作为前端机向主机提交作业的唯一入口。用户以批文件将作业提交到前端机是用户启动程序的主要方式。

3.1.1 处理机调度的层次

高级调度



内存空间有限，有时无法将用户提交的作业全部放入内存

作业后备队列

作业：一个具体的任务

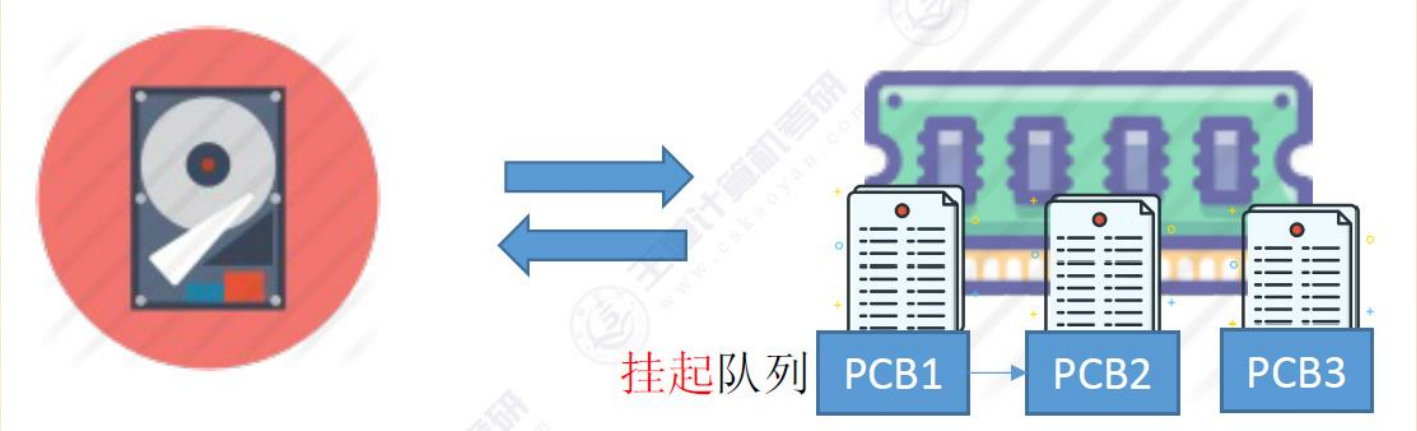
用户向系统提交一个作业≈用户让操作系统启动一个程序（来处理一个具体的任务）

简化理解：好几个程序需要启动，到底先启动哪个

高级调度（作业调度）。按一定的原则从外存的作业后备队列中挑选一个作业调入内存，并创建进程。每个作业只调入一次，调出一次。作业调入时会建立PCB，调出时才撤销PCB。

- 中级调度
  - 在条件允许下，在外存挂起的进程集合中选择哪些进程激活并调回内存。
  - 为提高效率，加快进程运行，调节系统的负荷，提高吞吐量。
  - 有时需要在选择内存中阻塞或就绪的进程暂时放到外存（一般是硬盘），即所谓的挂起。
  - 当这些进程又具备了运行条件、且内存又稍有空闲时，中级调度把外存上的就绪进程调入内存，放入就绪队列。这种内外存的数据交换称为对换。

- 中级调度



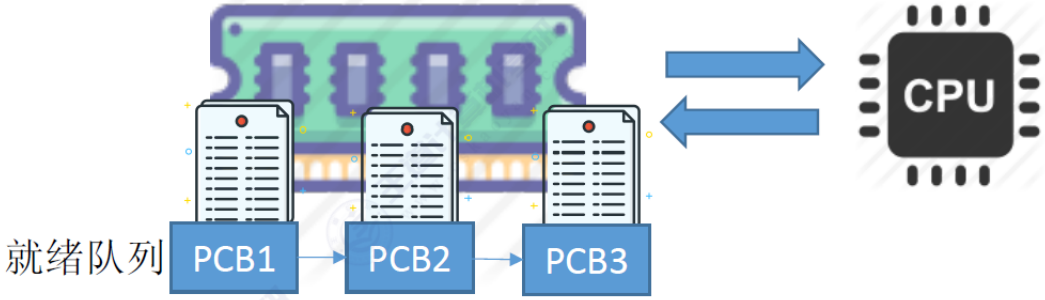
内存不够时，可将某些进程的数据调出外存。等内存空闲或者进程需要运行时再重新调入内存。

暂时调到外存等待的进程状态为挂起状态。被挂起的进程PCB会被组织成挂起队列。

中级调度(内存调度)：按照某种策略决定将哪个处于挂起状态的进程重新调入内存。一个进程可能会被多次调入、调出内存。因此，中级调度发生的频率要比高级调度更高。

3.1.1 处理机调度的层次

- **低级调度**
  - **也称进程调度或短期调度。用于决定就绪队列中哪个进程获得处理机，之后派发程序（dispatcher）将处理机分配给该进程。**

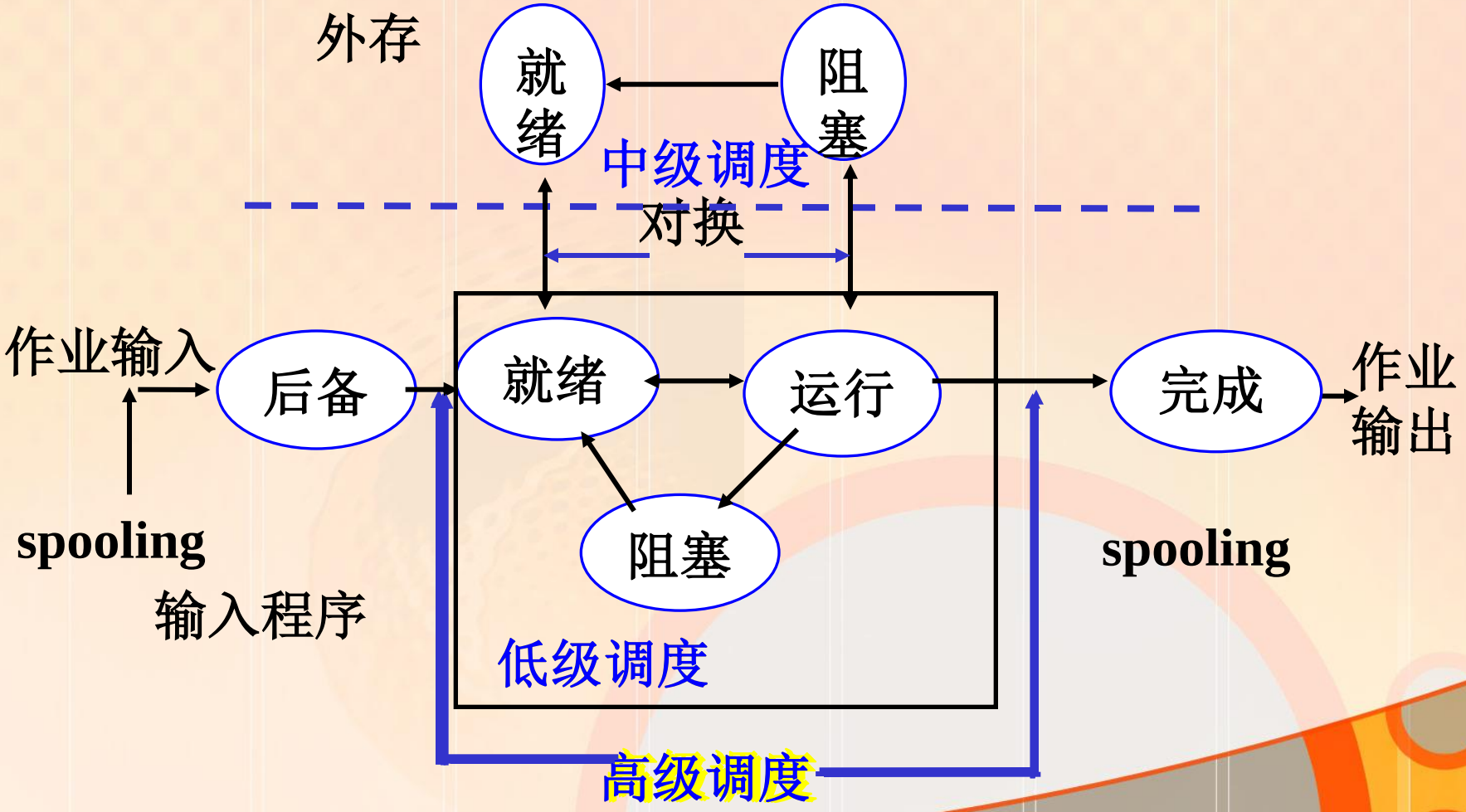


低级调度（进程调度/处理机调度）—— 按照某种策略从就绪队列中选取一个进程，将处理机分配给它。

进程调度是操作系统中最基本的一种调度，在一般的操作系统中都必须配置进程调度。
进程调度的频率很高，一般几十毫秒一次。

- 进程调度可采用下面两种方式
  - 1）非抢先式调度（Non-preemptive Mode）
    - 一旦把处理机分配给某进程后，便让该进程一直执行，直至该进程完成或阻塞时，才再把处理机分配给其他进程。
      - *正在执行的进程正常结束或由于某种错误而终止运行；*
      - *执行中的进程提出I/O请求，在等待I/O完成前，进程阻塞，转进程调度；*
      - *在进程通讯中，执行中的进程执行了某种原语操作，如P操作、阻塞原语和唤醒原语。*
  - 2）抢先式调度（preemptive mode）
    - 允许暂停某个正在执行的进程，将已分配给该进程的处理机重新分配给另一进程。
      - *时间片原则：在分时系统中，按照时间片轮转，分给进程的时间片用完*
      - *优先权原则：按照优先级调度，有更高优先级进程变为就绪*
      - *短作业优先原则*

三种调度之间的关系

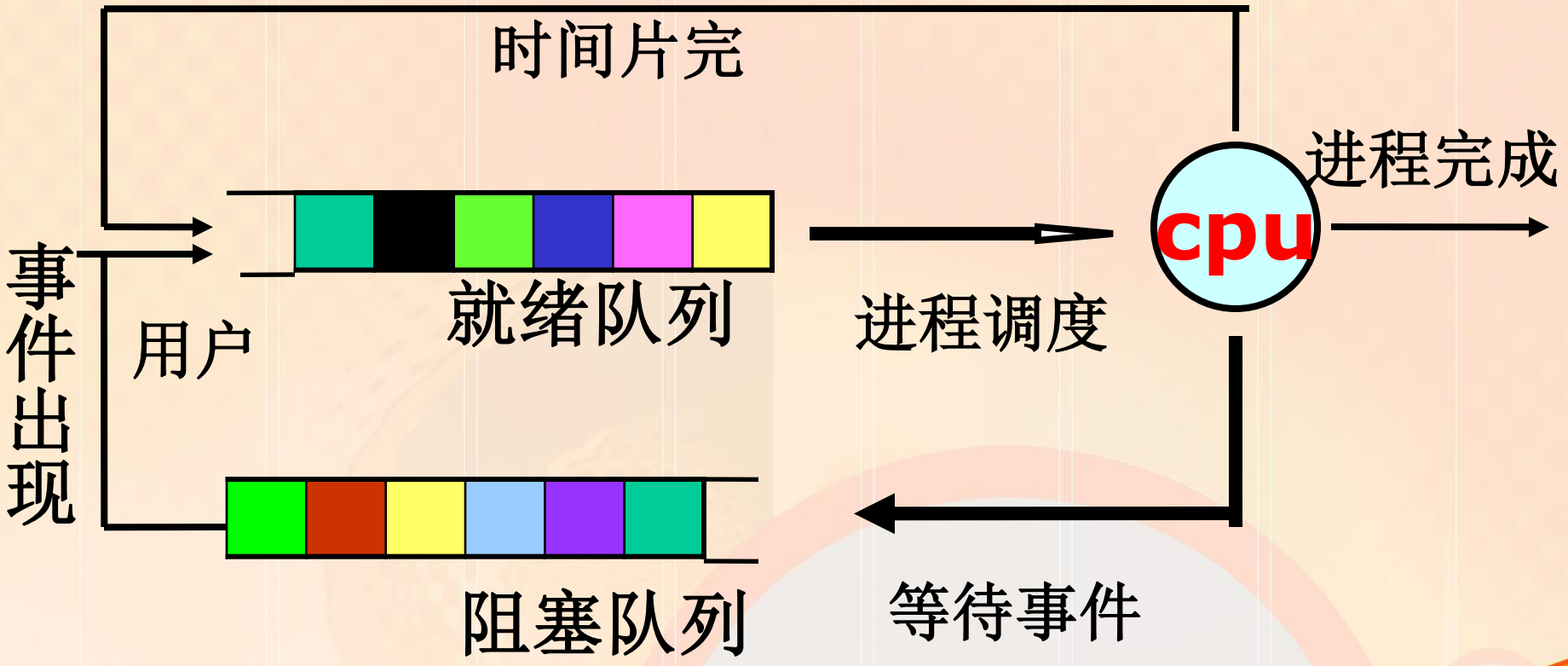

外存
就绪
阻塞
中级调度
对换
作业输入
后备
就绪
运行
完成
作业输出
spooling
输入程序
阻塞
spooling
低级调度
高级调度

3.1.1 处理机调度的层次

	要做什么	调度发生在..	发生频率	对进程状态的影响
高级调度（作业调度）	按照某种规则，从后备队列中选择合适的作业将其调入内存，并为其创建进程	外存→内存（面向作业）	最低	无→创建态→就绪态
中级调度（内存调度）	按照某种规则，从挂起队列中选择合适的进程将其数据调回内存	外存→内存（面向进程）	中等	挂起态→就绪态（阻塞挂起→阻塞态）
低级调度（进程调度）	按照某种规则，从就绪队列中选择一个进程为其分配处理机	内存→CPU	最高	就绪态→运行态

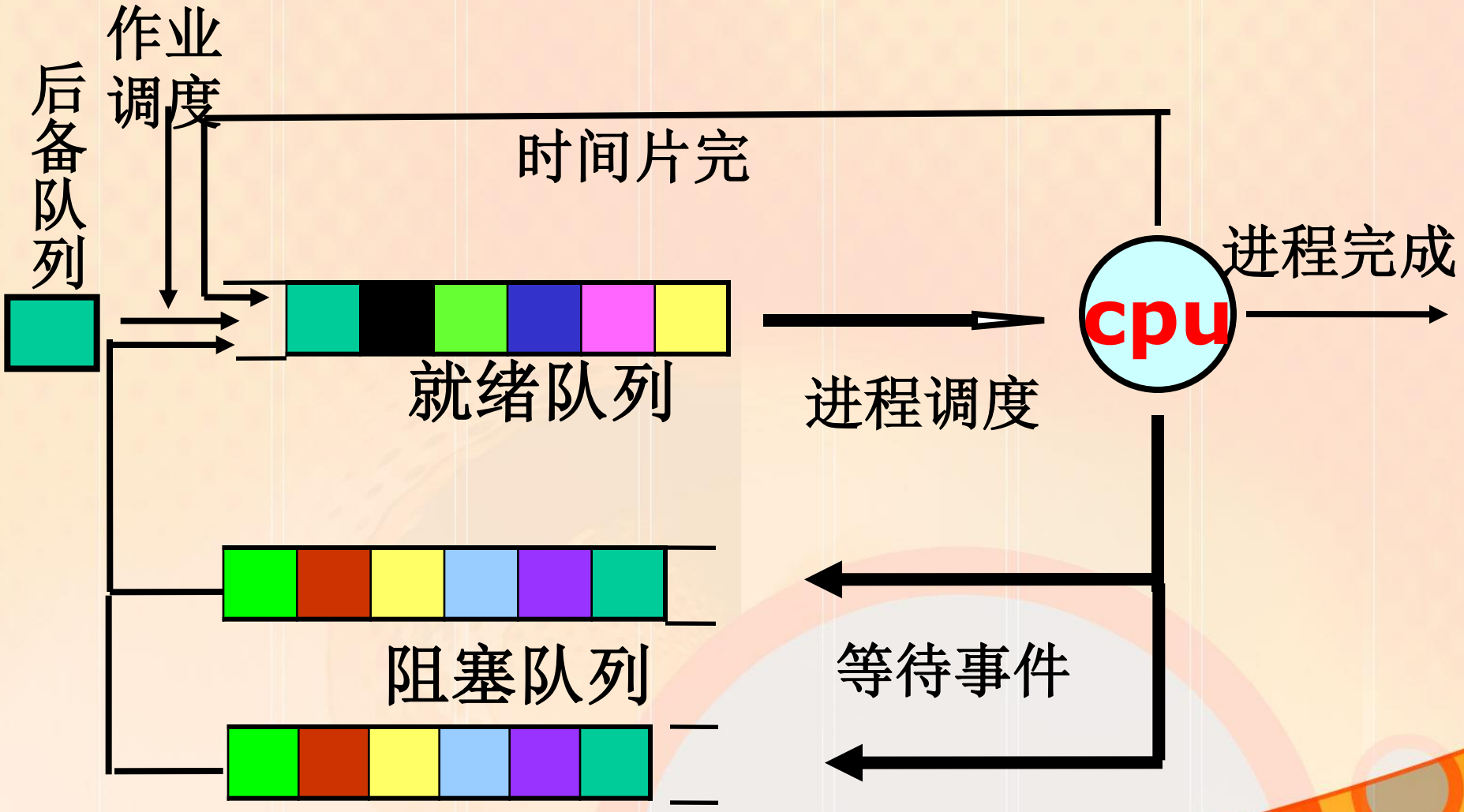
3.1.2 调度队列模型

1. 仅有进程调度的调度队列模型



特点：单就绪、单阻塞队列

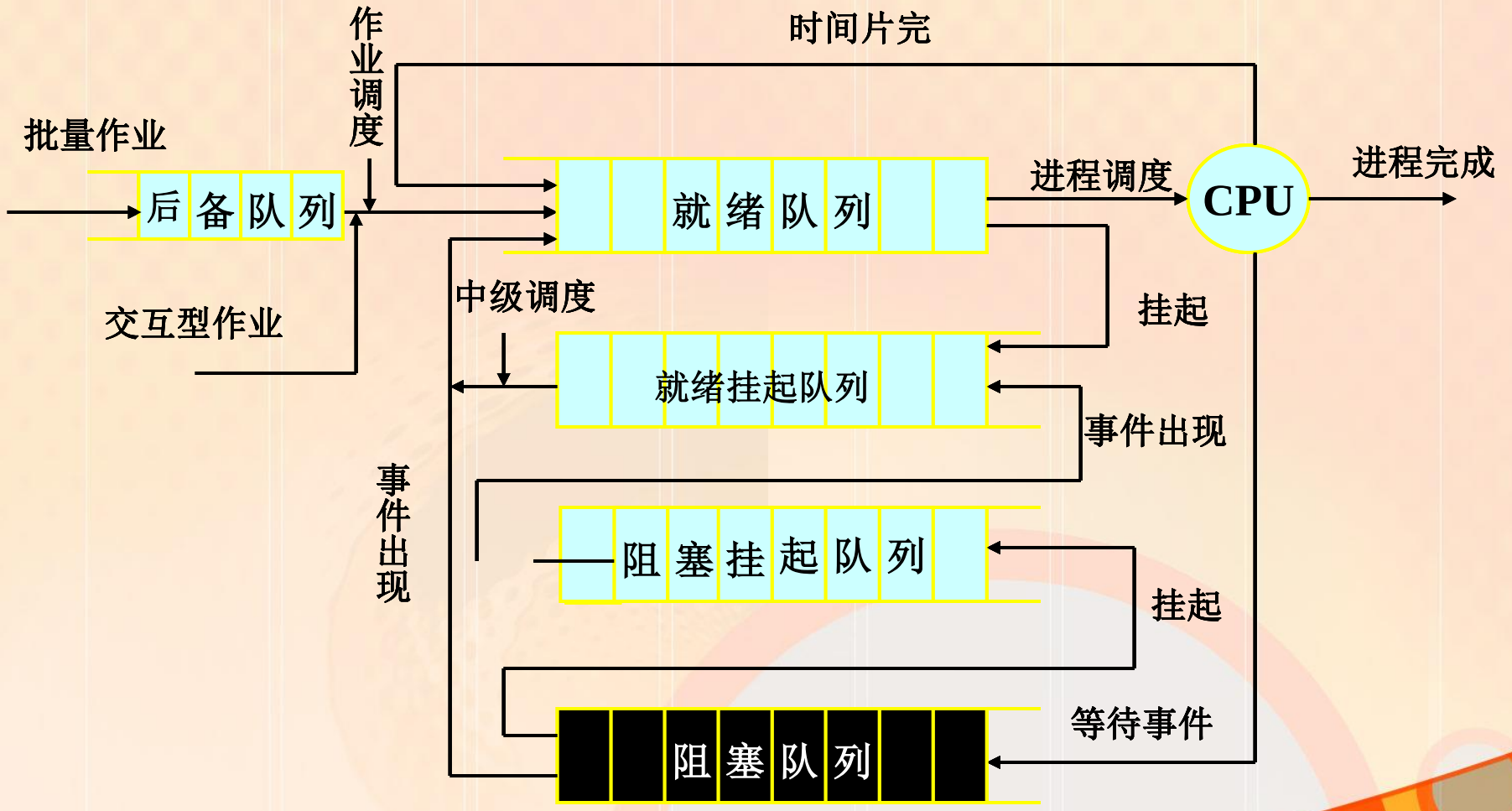
2. 具有高级和低级的调度队列模型



特点 ： 1) 具有进程调度、作业调度

2) 根据阻塞原因设置了多个阻塞队列

3. 同时具有三级调度的调度队列模型



选择哪种模型应根据系统的规模及目标制定

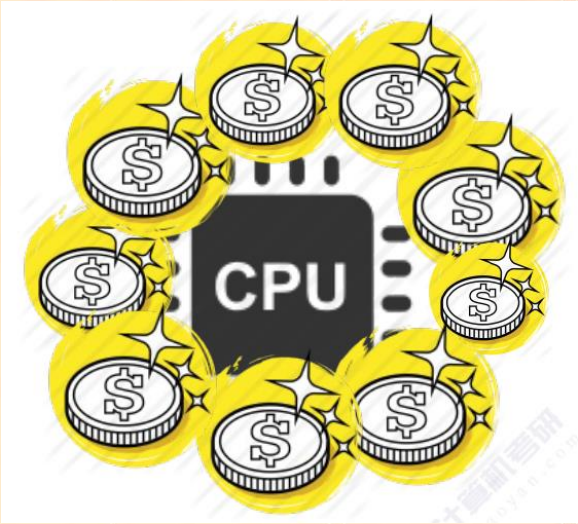
3.1.3 选择调度方式和算法的若干准则

- 我们可从不同的角度来判断处理机调度算法的性能。实际的处理机调度算法选择是一个综合的判断结果。

- 面向用户的准则
  - 周转时间短
  - 响应时间快
  - 截止时间的保证
  - 优先权准则

- 面向系统的准则
  - 系统吞吐量高
  - 处理机利用率好
  - 资源的平衡利用

CPU利用率



由于早期的CPU造价极其昂贵，因此人们会希望CPU尽可能多地工作。

CPU利用率：指CPU“忙碌的”时间占总时间的比例

利用率 = 忙碌的时间/总时间

e.g. 某计算机只支持单道程序，某个作业刚开始只需要在CPU上运行5秒，再用打印机打印输出5秒，之后再执行5秒，才能结束。在此过程中，CPU利用率、打印机利用率分别是多少？

CPU利用率 = (5+5)/(5+5+5) = 66.66%

打印机利用率 = 5/(5+5+5) = 33.33%

通常会考察多道程序并发运行的情况，可以用“甘特图”来辅助计算。

系统吞吐量

对计算机来说，希望能用尽可能少的时间处理完尽可能多的工作。

系统吞吐量：单位时间内完成的作业数量。

系统吞吐量 = 总共完成了多少道作业/总共花了多少时间

E.g., 某计算机系统处理完10道作业，共花费100秒，则系统吞吐量为：

10/100 = 0.1 道/秒

周转时间

- 批处理系统的重要指标。
- 作业从提交到完成（得到结果）所经历的时间为周转时间。
- 包括：在外存后备队列中等待，CPU上执行，就绪队列和阻塞队列中等待，结果输出等待。
- 平均周转时间T和平均带权周转时间（带权周转时间W是 T(周转)/ (CPU执行)）
- 平均周转时间： $T = \frac{1}{n}\sum_{i=1}^{n} T_i$
- 带权周转时间： $W = \frac{1}{n}\sum_{i=1}^{n} \frac{T_i}{T_{si}}$

$$\text{带权周转时间} = \frac{\text{作业周转时间}}{\text{作业实际运行的时间}} = \frac{\text{作业完成时间} - \text{作业提交时间}}{\text{作业实际运行的时间}}$$

$$\text{平均带权周转时间} = \frac{\text{各作业带权周转时间之和}}{\text{作业数}}$$

例：有如下三道作业。系统为它们服务的顺序是：1、2、3。求平均周转时间和平均带权周转时间。

作业	提交时间/时	运行时间/h
1	10.00	2
2	10.10	1
3	10.25	0.25

解：

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	10.00	2	10	12	2	1
2	10.10	1	12	13	2.9	2.9
3	10.25	0.25	13	13.25	3	12

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	10.00	2	10	12.00	2	2/2
2	10.10	1	12	13.00	2.9	2.9/1
3	10.25	0.25	13	13.25	3	3/0.25

平均周转时间：

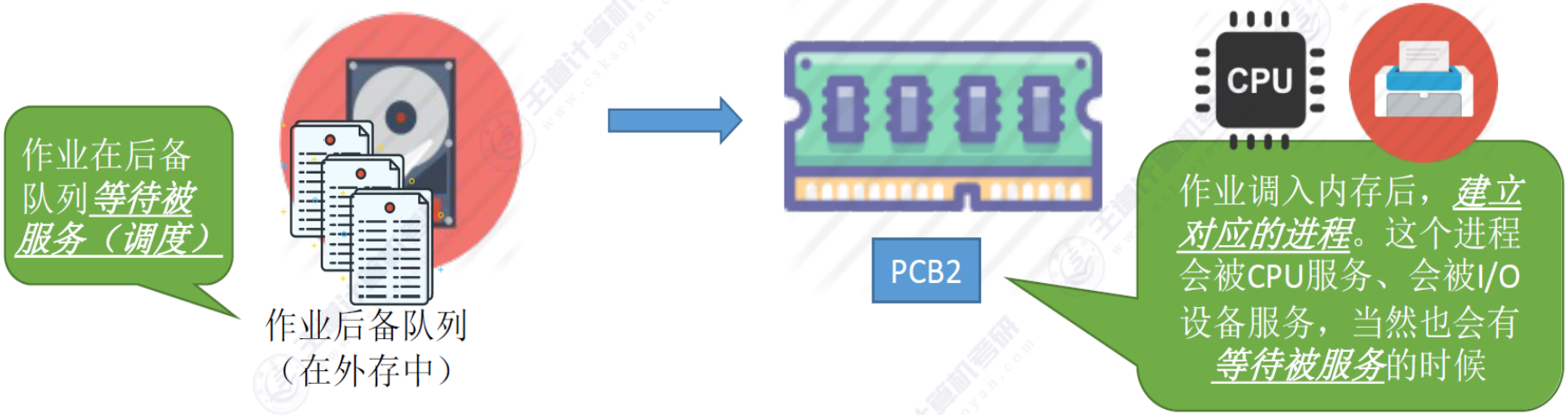
T=(2+2.9+3)/3=2.63h

平均带权周转时间：

W=(2+2.9+12)/3=5.3h。

等待时间

计算机的用户希望自己的作业尽可能少的等待处理机
等待时间，指进程/作业处于等待处理机状态时间之和，等待时间越长，用户满意度越低。



对于进程来说，等待时间就是指进程建立后等待被服务的时间之和，在等待I/O完成的期间其实进程也是在被服务的，所以不计入等待时间。
对于作业来说，不仅要考虑建立进程后的等待时间，还要加上作业在外存后备队列中等待的时间。

一个作业总共需要被CPU服务多久，被I/O设备服务多久一般是确定不变的，因此调度算法其实只会影响作业/进程的等待时间。当然，与前面指标类似，也有“平均等待时间”来评价整体性能。

响应时间

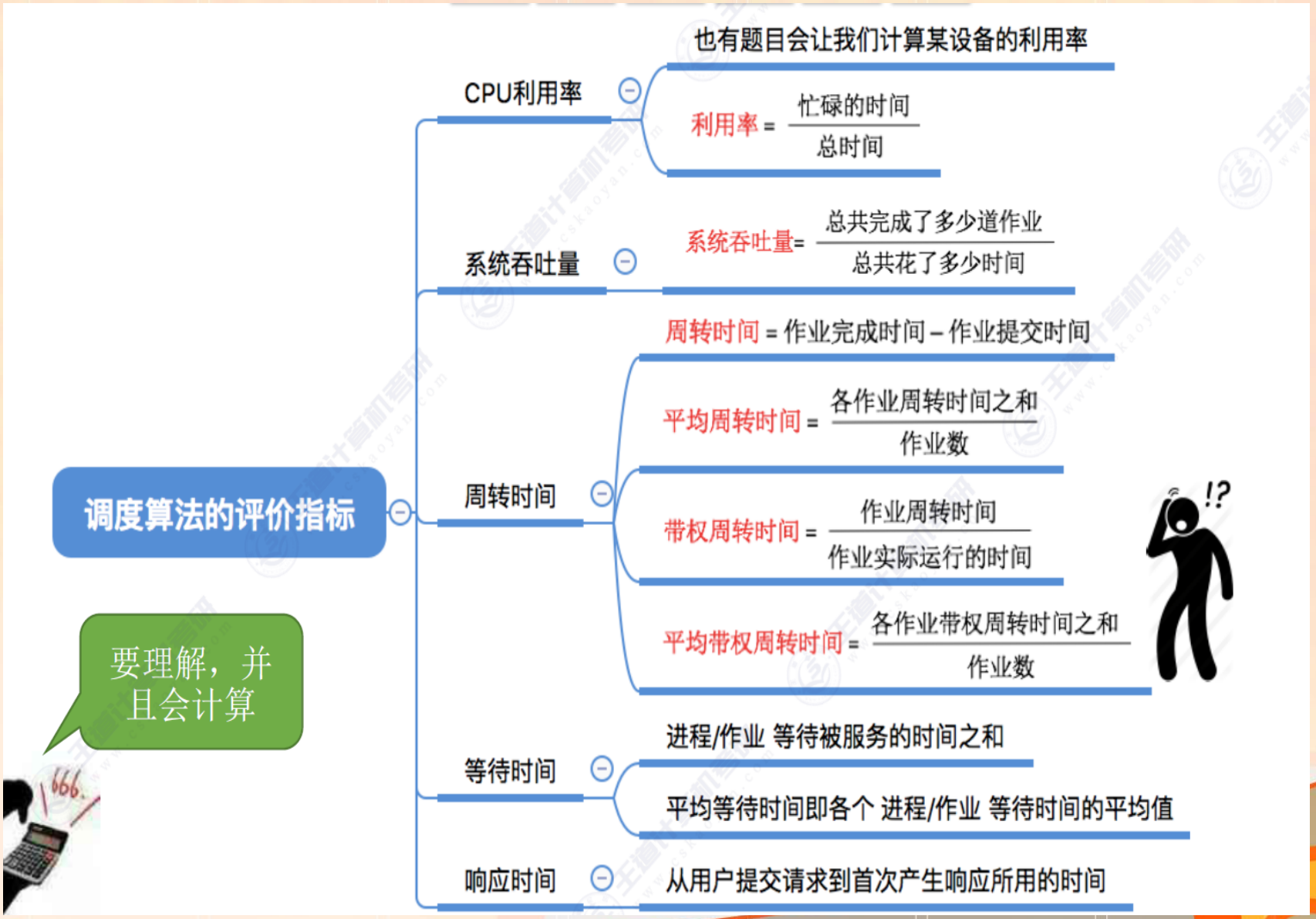
- 分时系统的重要指标。
- 用户输入一个请求（如击键）到系统给出首次响应（如屏幕显示）的时间。
- 包括：从终端的键盘输入的一个请求信息传送到处理机的时间；处理机对请求的处理时间；处理结果送到终端显示器的时间。

截止时间

- 实时系统的重要指标。
- 开始截止时间和完成截止时间
- 某任务必须开始执行的最迟时间，或必须完成的最迟时间。

优先权原则

- 批处理、分时、实时系统都可遵循
- 可以使关键任务达到更好的指标。
- 公平性：不因作业或进程本身的特性而使上述指标过分恶化。如长作业等待很长时间。


调度算法的评价指标
CPU利用率
也有题目会让我们计算某设备的利用率
利用率 = 忙碌的时间 / 总时间
系统吞吐量
系统吞吐量= 总共完成了多少道作业 / 总共花了多少时间
周转时间
周转时间 = 作业完成时间 – 作业提交时间
平均周转时间 = 各作业周转时间之和 / 作业数
带权周转时间 = 作业周转时间 / 作业实际运行的时间
平均带权周转时间 = 各作业带权周转时间之和 / 作业数
!?
要理解，并且会计算
666.
等待时间
进程/作业 等待被服务的时间之和
平均等待时间即各个 进程/作业 等待时间的平均值
响应时间
从用户提交请求到首次产生响应所用的时间

面向系统准则

- 吞吐量
  - 批处理系统的重要指标。
  - 吞吐量指单位时间内所完成的作业数，跟作业本身特性和调度算法都有关系。
- 处理机利用率高
  - 大中型主机多用户系统性能指标，系统价格昂贵。PC一般不考虑这个指标。
- 各种资源的均衡利用
  - 大中型主机多用户系统性能指标。如CPU繁忙的作业和I/O繁忙（指次数多，每次时间短）的作业搭配。对PC及实时系统该指标并不重要。

调度算法本身的调度性能准则

- 易于实现
- 执行开销比