@[toc]

# 调度的基本概念

* 当有一堆任务要处理，但由于资源有限，这些事情没法同时处理。这就需要确定某种规则来决定处理这些任务的顺序，这就是“调度”研究的问题。
* 在多道程序系统中，进程的数量往往是多于处理机的个数的，这样不可能同时并行地处理各个进程。处理机调度，就是从就绪队列中按照一定的算法选择一个进程并将处理机分配给它运行，以实现进程的并发执行。

## 高级调度

调度的三个层次

* 由于内存空间有限，有时无法将用户提交的作业全部放入内存，因此就需要确定某种规则来决定将作业调入内存的顺序。
* 高级调度（作业调度）.按一定的原则从外存上处于后备队列的作业中挑选一个（或多个）作业，给他们分配内存等必要资源，并建立相应的进程（建立PCB）、以使它（们）获得竞争处理机的权力。

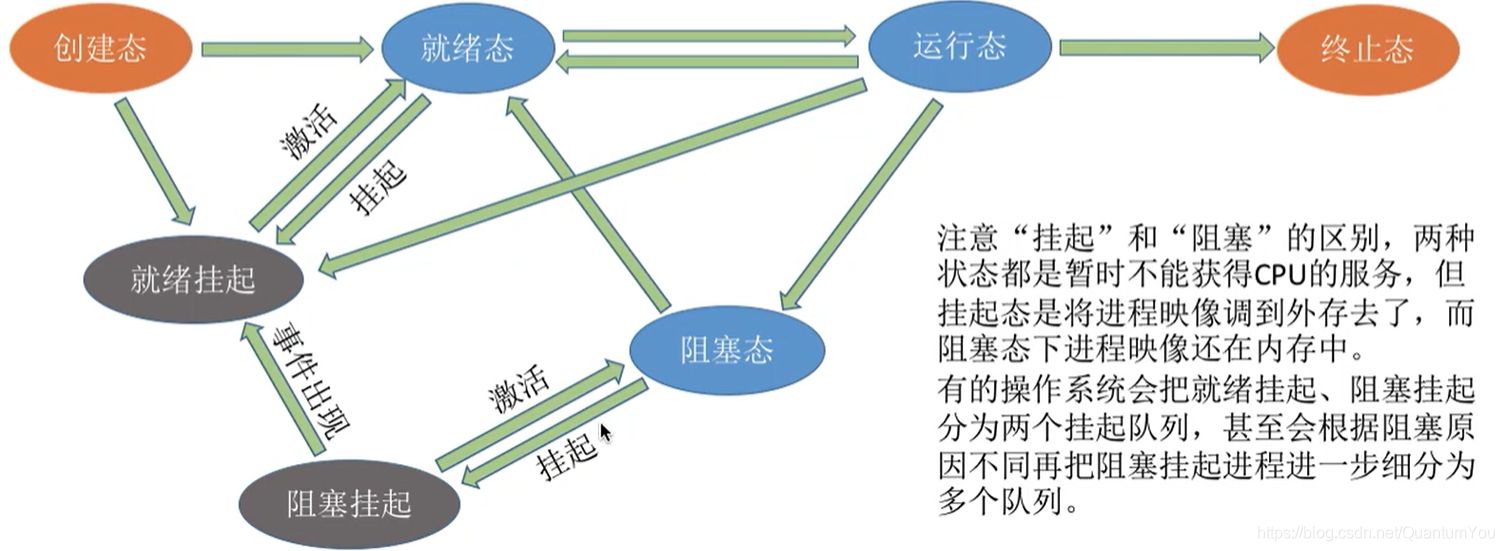
## 中级调度

* 引入了虚拟存储技术之后，可将暂时不能运行的进程调至外存等待。等它重新具备了运行条件且内存又稍有空闲时，再重新调入内存。这么做的目的是为了提高内存利用率和系统吞吐量。
* 暂时调到外存等待的进程状态为挂起状态。值得注意的是，PCB并不会一起调到外存，而是会常驻内存。PCB中会记录进程数据在外存中的存放位置，进程状态等信息，操作系统通过内存中的PCB来保持对各个进程的监控、管理。被挂起的进程PCB会被放到的挂起队列中
* 中级调度（内存调度），就是要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出、调入内存，因此中级调度发生的频率要比高级调度更高。

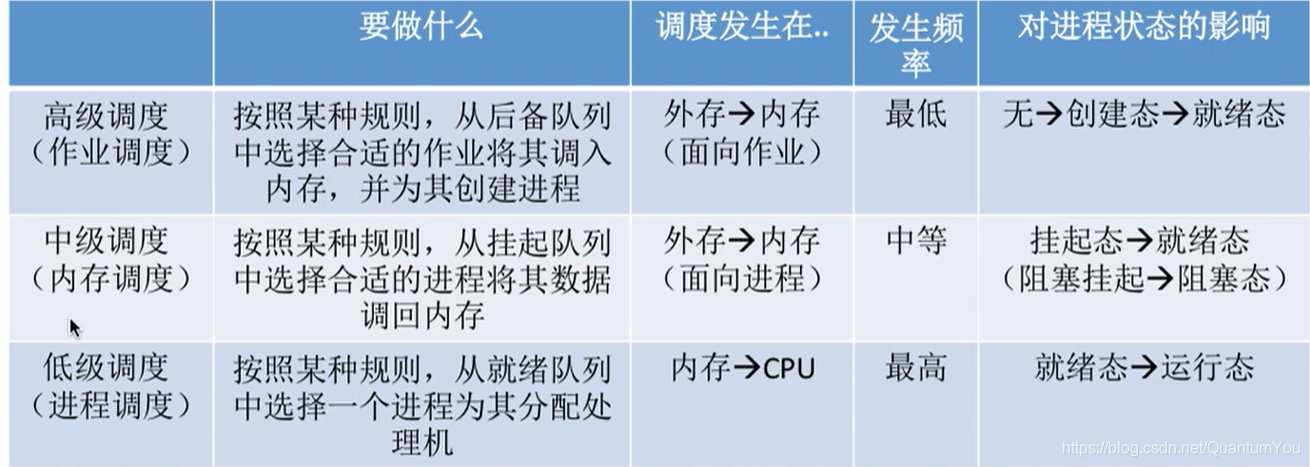
## 低级调度

* 低级调度（进程调度），其主要任务是按照某种方法和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。

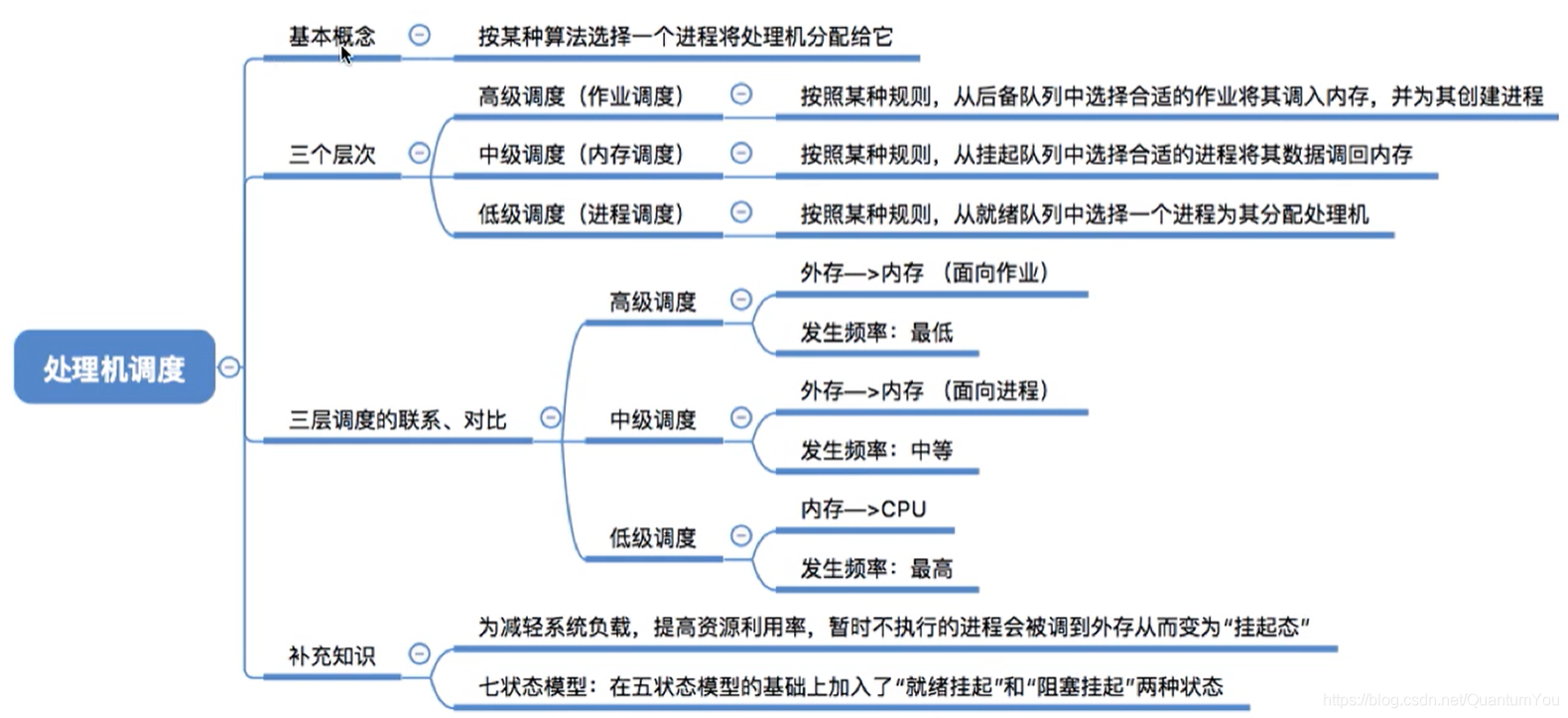
## 进程的挂起状态与七状态模型



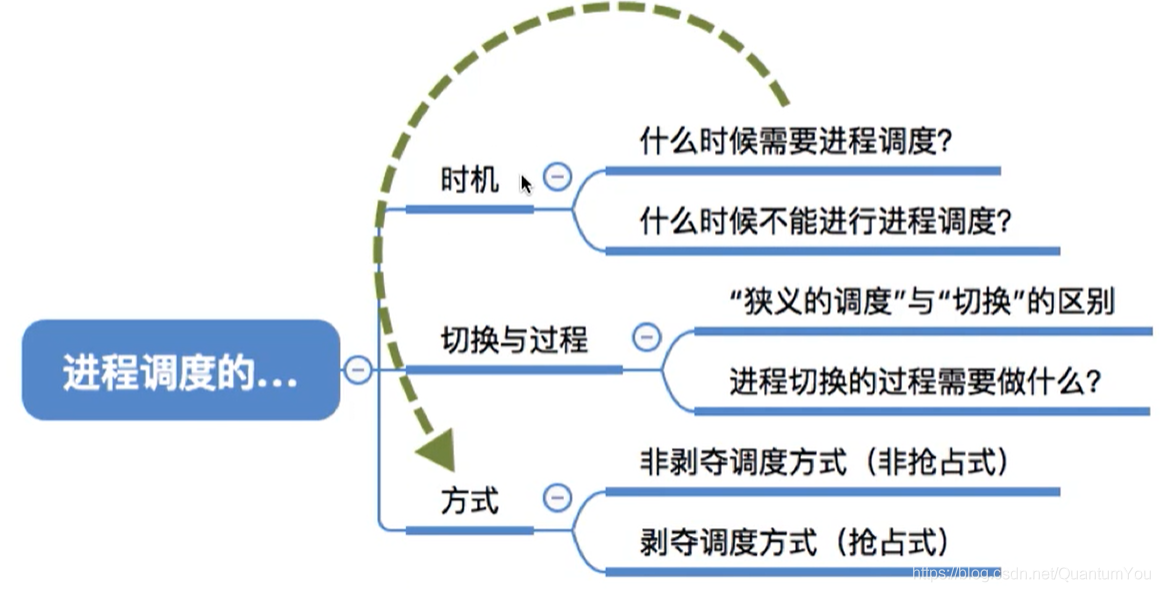
## 三层调度对比



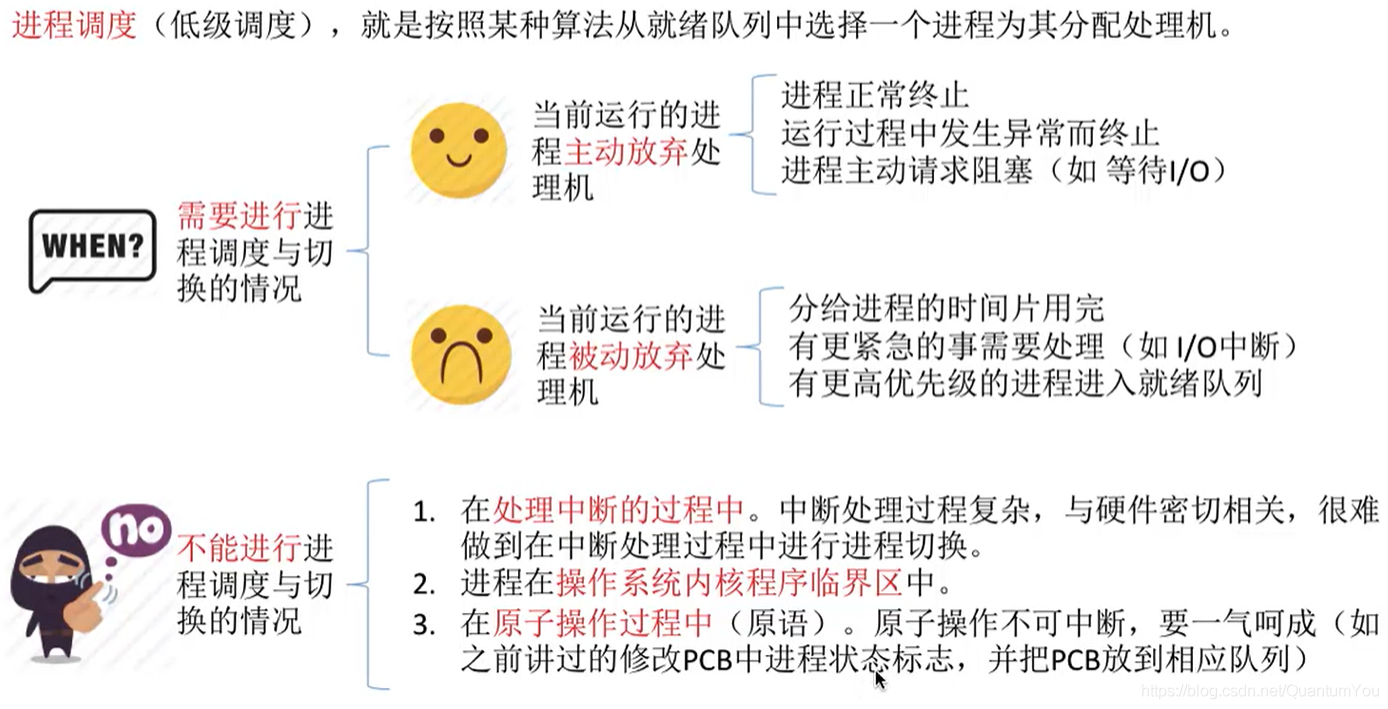
# 总结



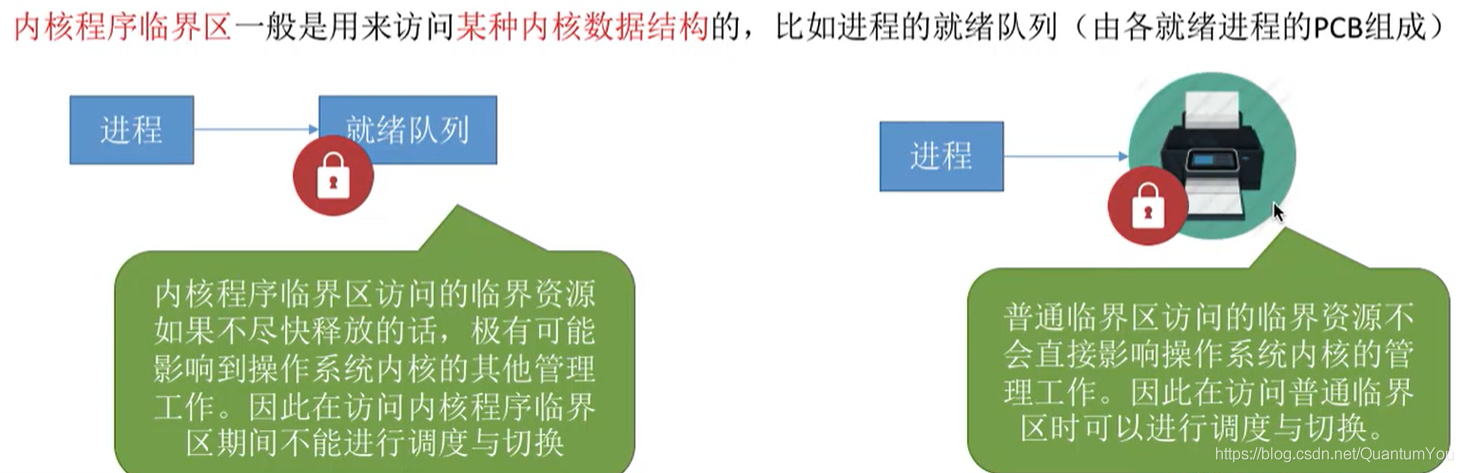
# 进程调度的时机过程与切换



## 进程调度的时机

  
注意：

* 进程在操作系统内核程序临界区中不能进行调度与切换（对）
* 进程处于临界区时不能进行处理机调度 （错）
* **临界资源**：一个时间段内只允许一个进程使用的资源。各进程需要互斥地访问临界资源。
* **临界区**：访问临界资源的那段代码。
* 内核程序临界区一般是用来访问某种内核数据结构的，比如进程的就绪队列（由各就绪进程的PCB组成）



* 进程是否主动放弃处理机（强行剥离处理机资源） ---> 进程调度的方式

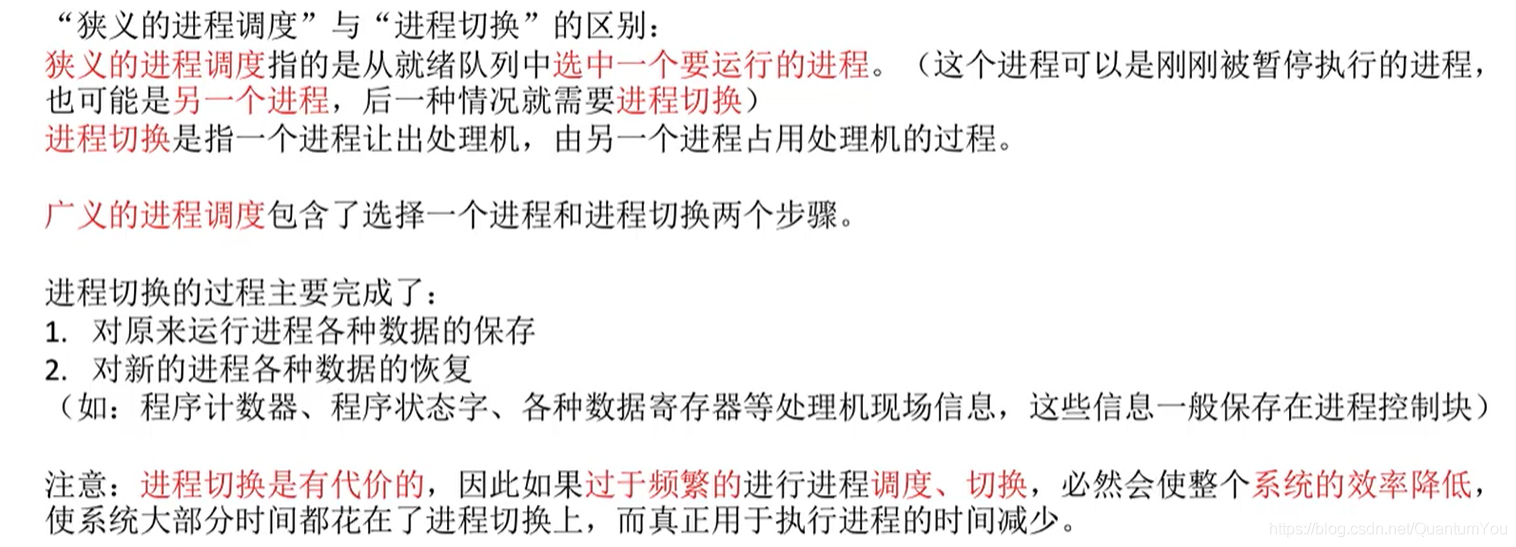
### 非剥夺调度方式

* 非剥夺调度方式，又称非抢占方式。即，只允许进程主动放弃处理机。在运行过程中即便有更紧迫的任务到达，当前进程依然会继续使用处理机，直到该进程终止或主动要求进入阻塞态。
* 优缺点：实现简单，系统开销小但是无法及时处理紧急任务，适合于早期的批处理系统

### 剥夺调度方式

* 剥夺调度方式，又称抢占方式。当一个进程正在处理机上执行时，如果有一个更重要或更紧迫的进程需要使用处理机，则立即暂停正在执行的进程，将处理机分配给更重要紧迫的那个进程。
* 优缺点： 可以优先处理更紧急的进程，也可实现让各进程按时间片轮流执行的功能（通过时钟中断）.适合于分时操作素统、实时操作系统

## 进程的切换与过程



# 小结

