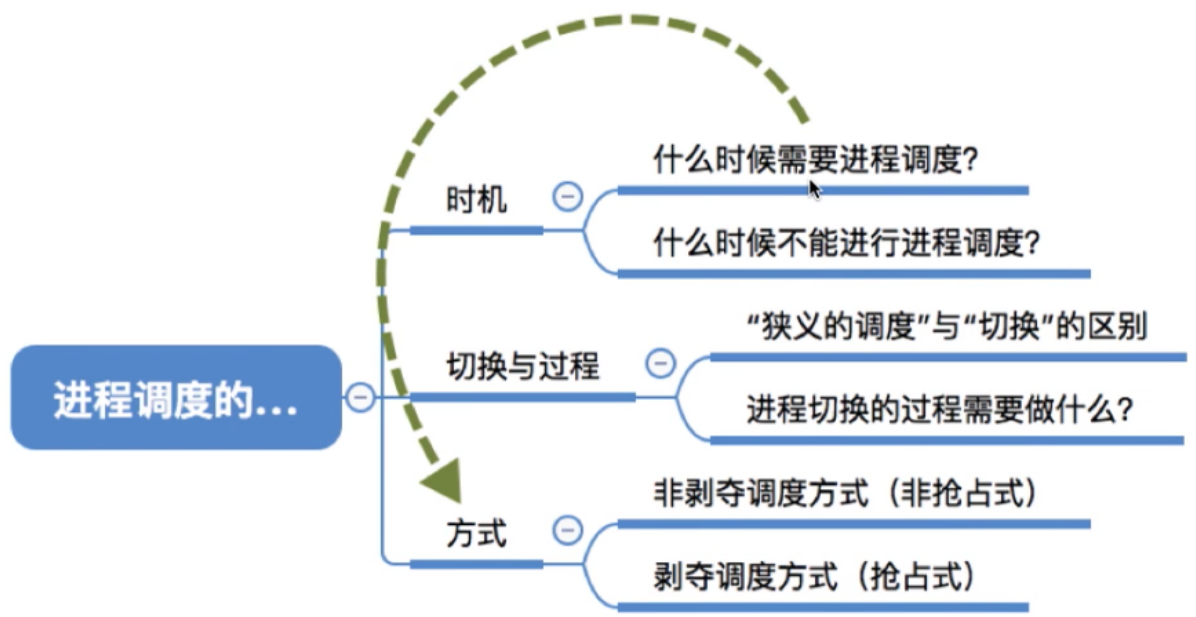
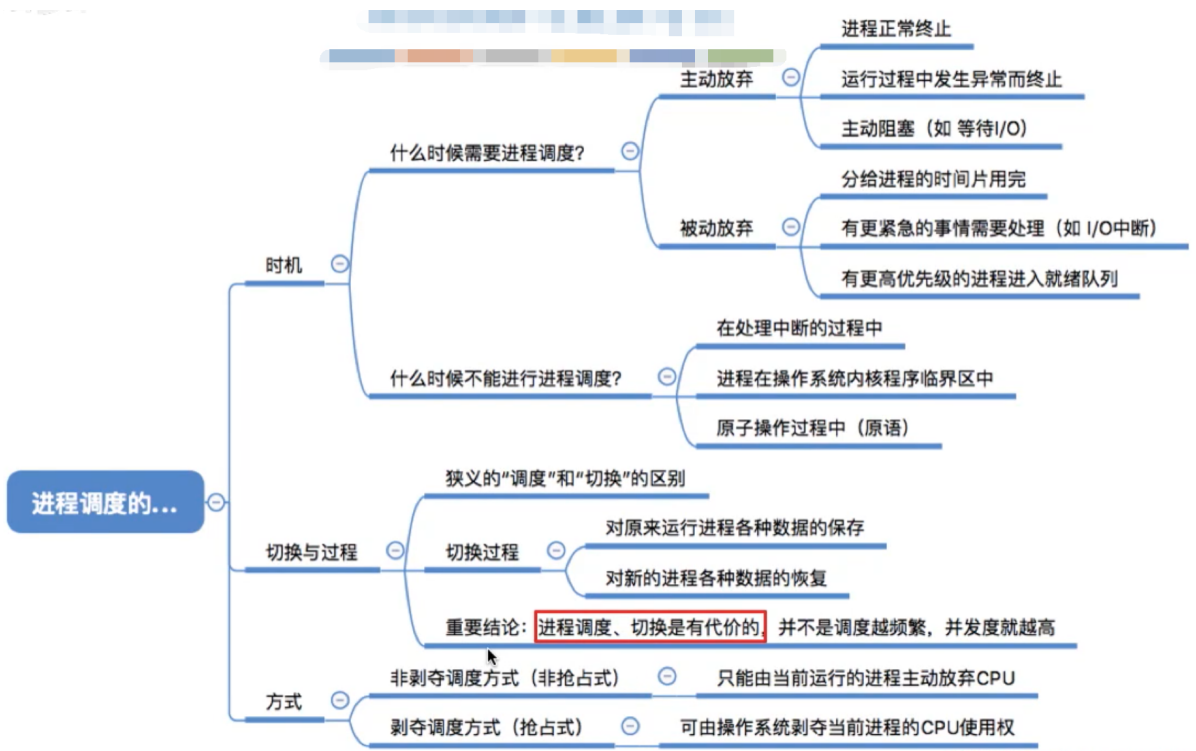


# 调度的实现



## 进程调度的时机

进程调度（低级调度），按照某种算法从就绪队列中选择一个进程为其分配处理机。

有的系统中，只允许进程主动放弃处理机

有的系统中，进程可以主动放弃处理机，当有更紧急的任务需要处理时，也会强行剥夺处理机（被动放弃）

但是进程在普通临界区中是可以进行调度、切换的。



进程在操作系统内核程序临界区中不能进行调度与切换

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）

如果还没退出临界区（还没解锁）就进行进程调度，但是进程调度相关的程序也需要访问就绪队列，但此时就绪队列被锁住了，因此又无法顺利进行进程调度

内核程序临界区访问的临界资源如果不尽快释放的话，极有可能影响到操作系统内核的其他管理工作。因此在访问内核程序临界区期间不能进行调度与切换

在打印机打印完成之前，进程一直处于临界区内，临界资源不会解锁。但打印机又是慢速设备，此时如果一直不允许进程调度的话就会导致CPU一直空闲

普通临界区访问的临界资源不会直接影响到操作系统内核的管理工作。因此在访问普通临界区时可以进行调度与切换。

## 进程调度的方式

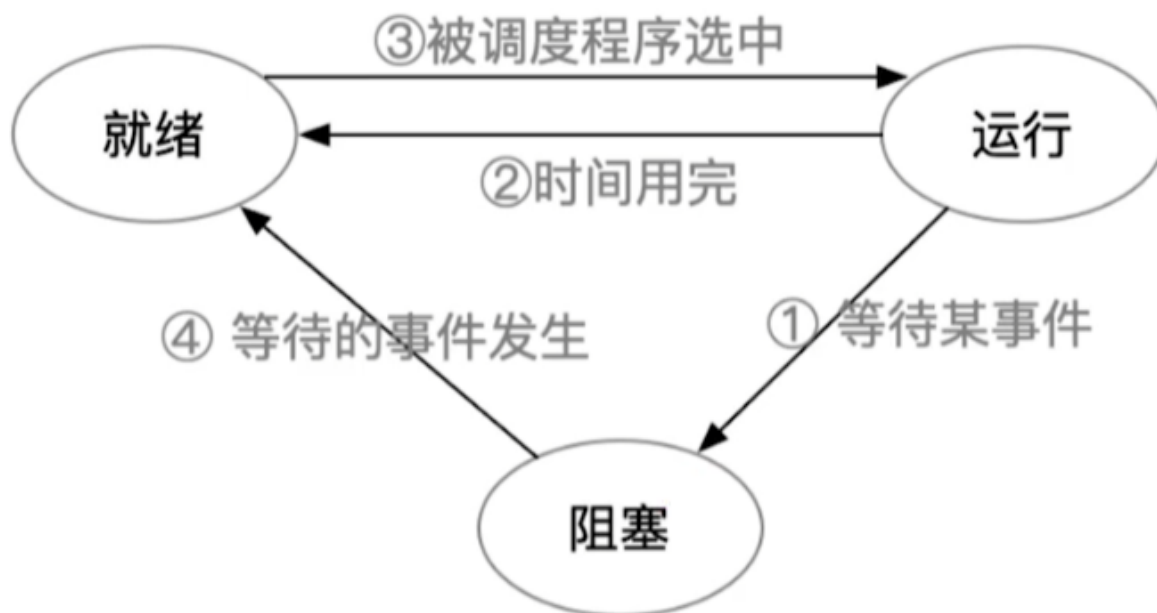
非剥夺调度方式，又称非抢占式。即，只允许进程主动放弃处理机。在运行过程中即便有更紧迫的任务到达，当前进程依然会继续使用处理机，直到该进程终止或主动要求进入阻塞态。

实现简单，系统开销小但是无法及时处理紧急任务，适用于早期的批处理系统

剥夺调度方式，又称抢占式。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。

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转执行的功能（通过时钟中断）。适合于分时操作系统、实时操作系统

## 调度器/调度程序(scheduler)



2、3由调度程序引起，调度程序决定：

让谁运行？——调度算法

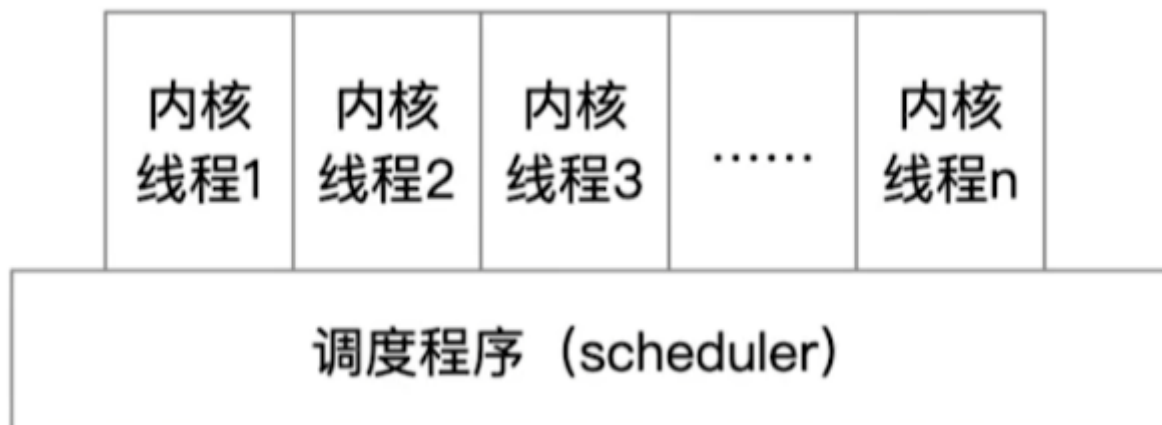
运行多长时间？——时间片大小

调度时机——什么事件会触发“调度程序”？

- 创建新进程
- 进程退出
- 运行进程阻塞
- I/O中断发生（可能唤醒某些阻塞进程）
- 非抢占式调度策略，只有运行进程阻塞或退出才触发调度程序工作
- 抢占式调度策略，每个时钟中断或k个时钟中断会触发调度程序工作



不支持内核级线程的操作系统，调度程序的处理对象是进程



支持内核级线程的操作系统，调度程序的处理对象是内核线程

## 闲逛进程

---

没有其他就绪进程时，运行闲逛进程（idle）

闲逛进程的特性：

- 优先级最低
- 可以是0地址指令，占一个完整的指令周期（指令周期末尾例行检查中断）
- 能耗低