

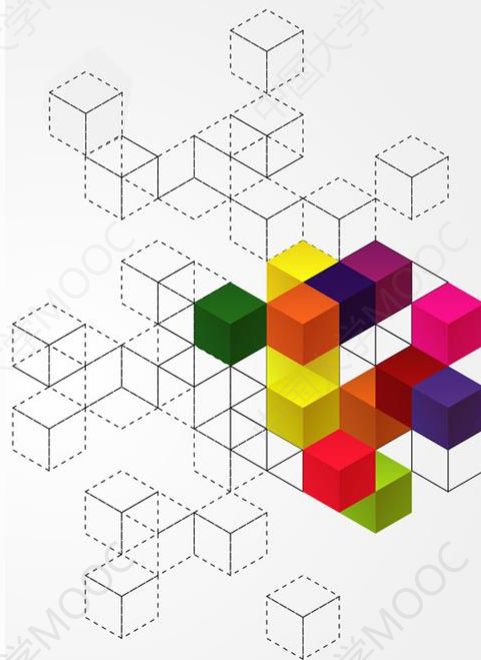
# 操作系统

Operating system

徐子川

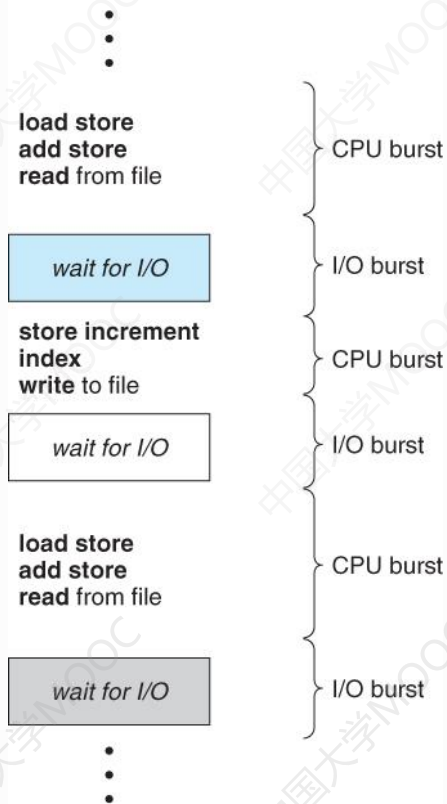
大连理工大学

- 一、调度基本概念
- 二、并行与并发
- 三、调度时机
- 四、CPU调度器构成

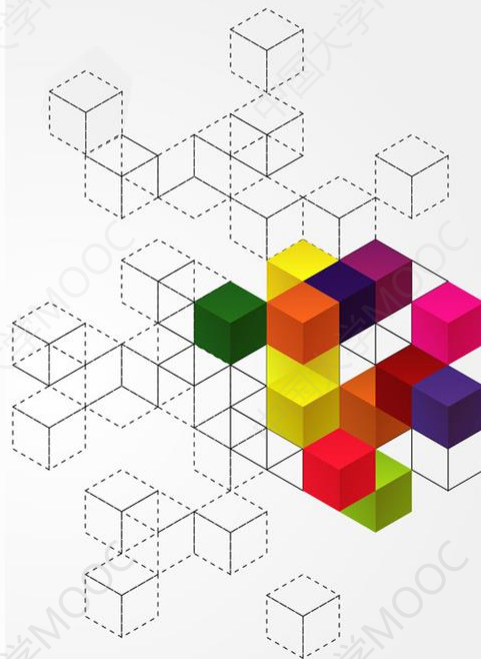


# 一、调度基本概念

## 为什么要进行调度（基本场景）



- 引入multitasking的根本原因
  - 程序执行场景：CPU burst与IO burst交替
  - 若仍采用**串行执行**的方式，则IO burst期间，CPU被限制（**浪费CPU资源**）
- **需要更为精细的调度方法**，使得多任务能够以高效的并发形式运作



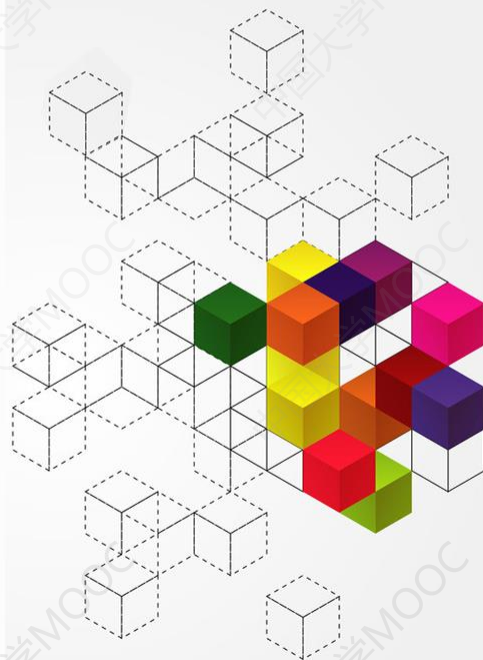
## 二、并行与并发

### ● 并行与并发的比较

- 并行 (Parallel)
- 并发 (Concurrency)



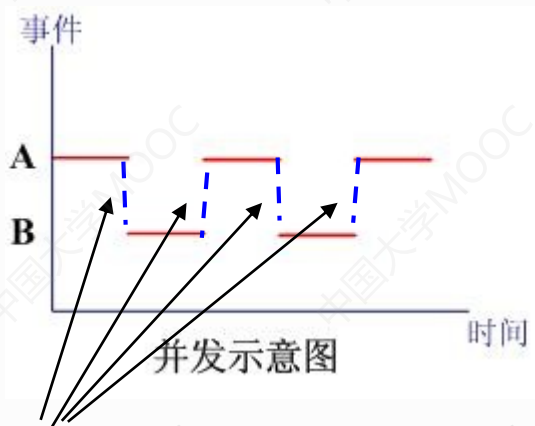
- 并发的概念其实包含并行的概念
- 右侧的图说明的是并发概念中除了并行情况之外的另一种例外情况：在单处理机的机器上，也可以通过任务调度与切换来造成多任务同时执行的假象



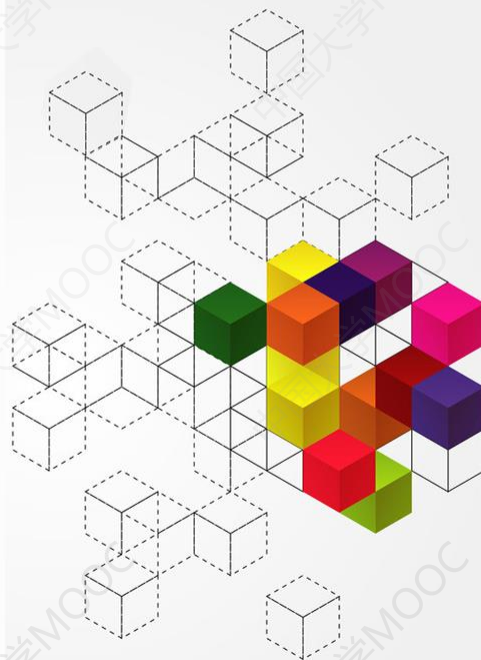
## 二、并行与并发

### ● 并发概念下的上下文切换

- 并发 (Concurrency)

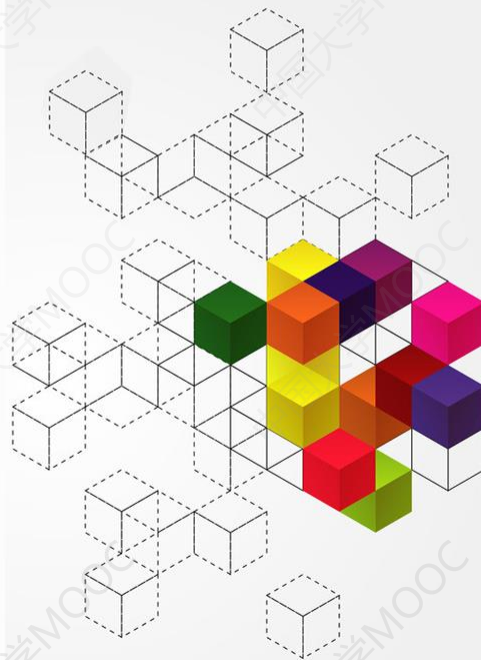
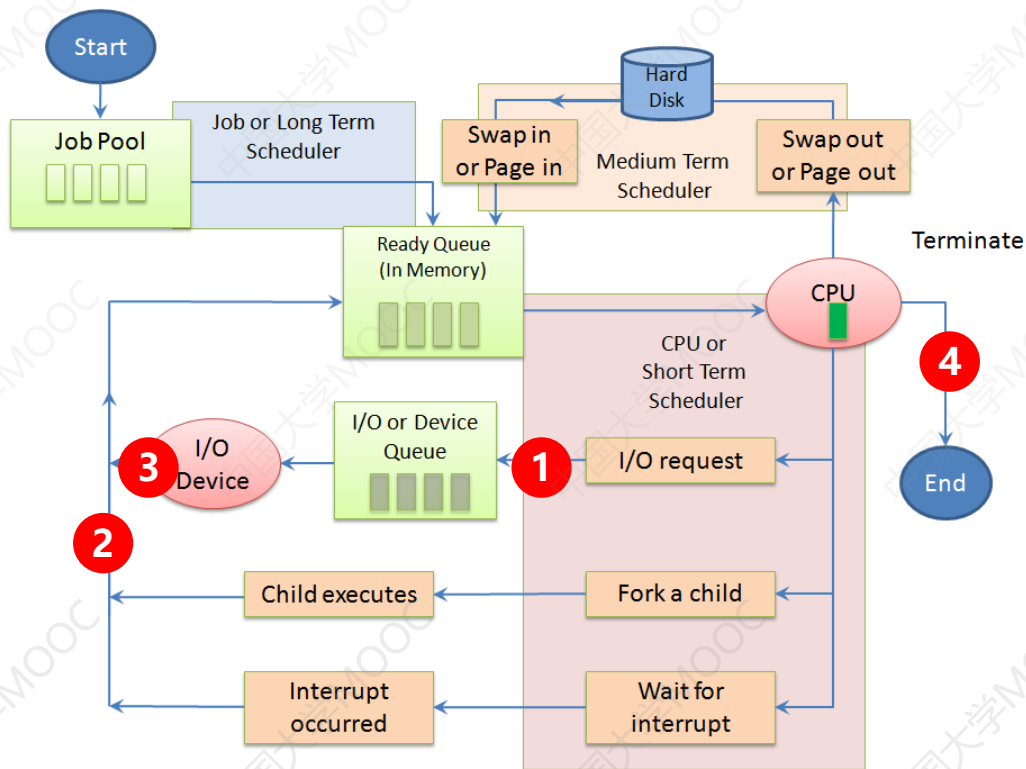


上下文切换，调度发生的地方



### 三、调度时机

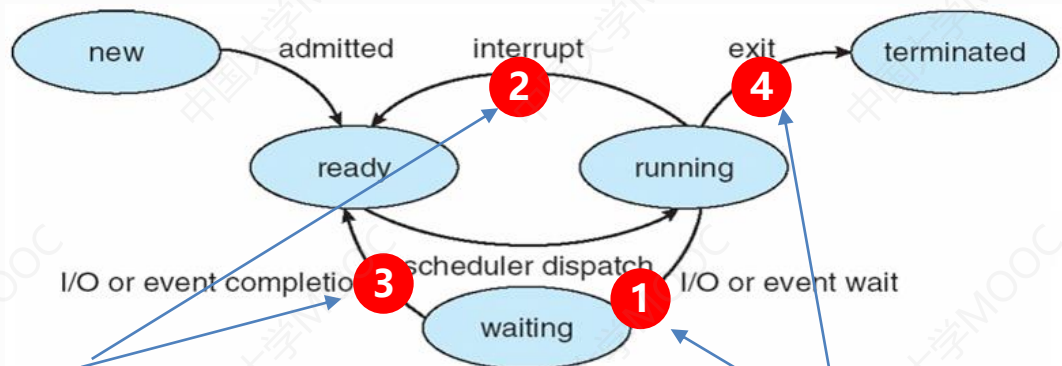
- 何时需要调度程序选择新的进程运行？





### 三、调度时机

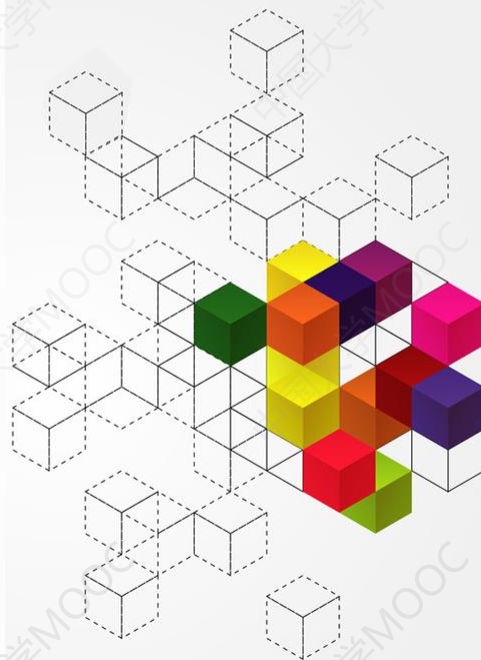
- 何时需要调度程序选择新的进程运行？
  - 从进程状态变换的视角看调度时机



**可抢占调度点**  
(preemptive)

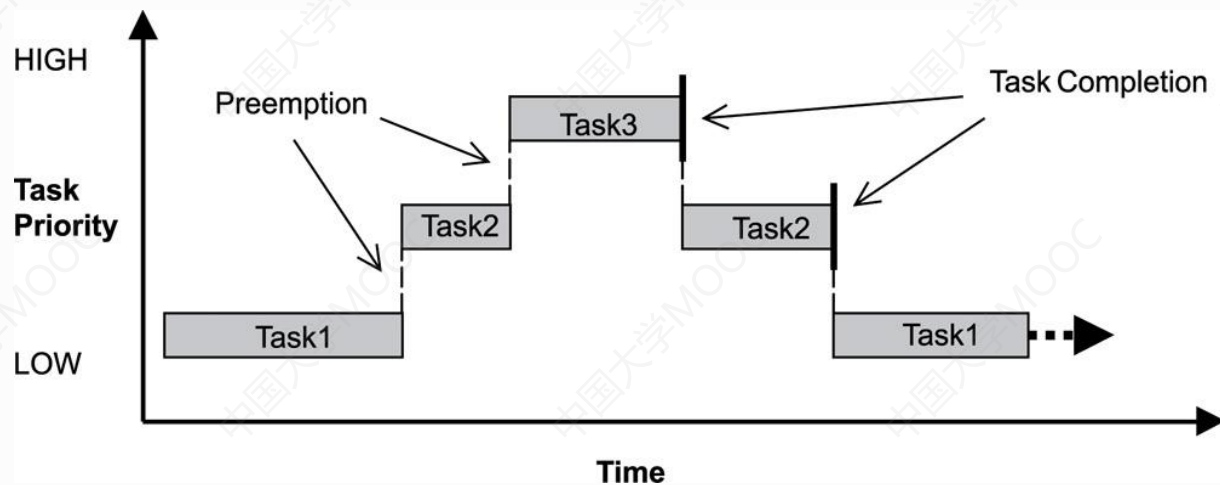
-允许新进程抢占当前进程执行

**非抢占调度点**  
(non-preemptive)

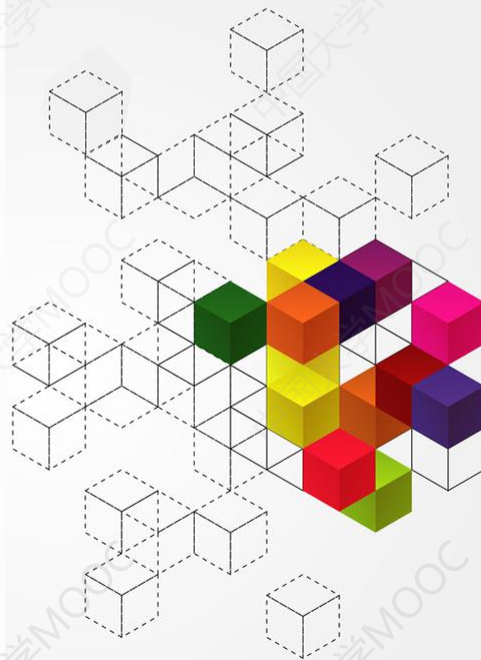


### 三、调度时机

- 何时需要调度程序选择新的进程运行？
  - Preemption in realtime OS



高优先级任务抢占低优先级任务的时间  
(preemption)

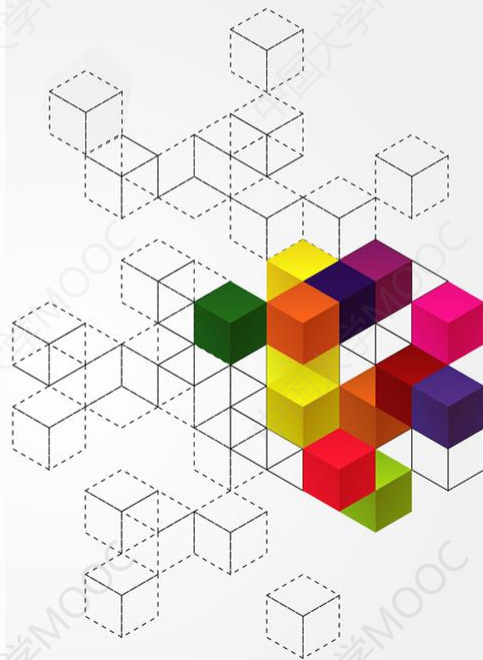
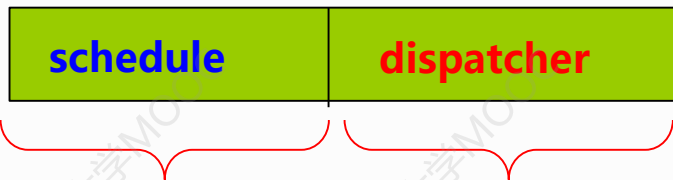




## 四、CPU调度器构成

● 调度器由两个主要部件组成：

- 调度(schedule)：在调度点，从就绪队列选择新的进程
- 派遣(dispatch)：将新的进程安排到CPU上运行
- 操作系统通过派遣程序将CPU的控制权交给新选中的进程



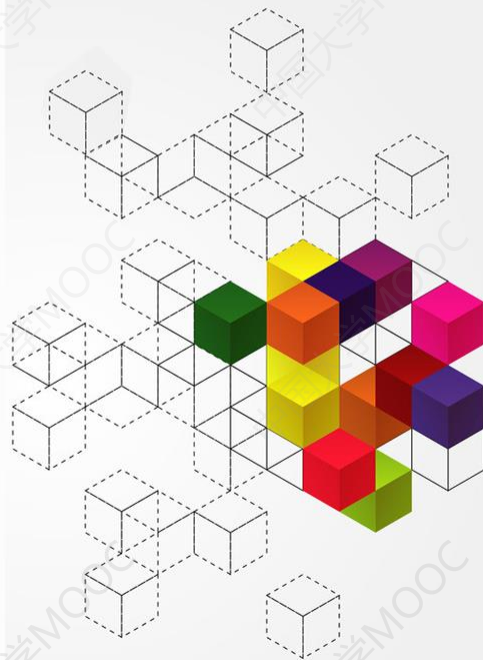
## 四、CPU调度器构成

### ● 派遣 (Dispatch) 的功能

- 1. 切换上下文: 切换到新被选中的进程的上下文
- 2. 将进程从内核态切换到用户态
- 3. 跳转到进程上次执行到的PC位置, 重新开始执行

派遣延迟: 派遣过程所耗费的时间

调度器执行一次调度的开销  
= 调度算法执行时间 + 派遣延迟



# 本讲小结

- 调度基本概念
- 并行与并发
- 调度时机
- CPU调度器构成

