

# 操作系统

Operating system

徐子川

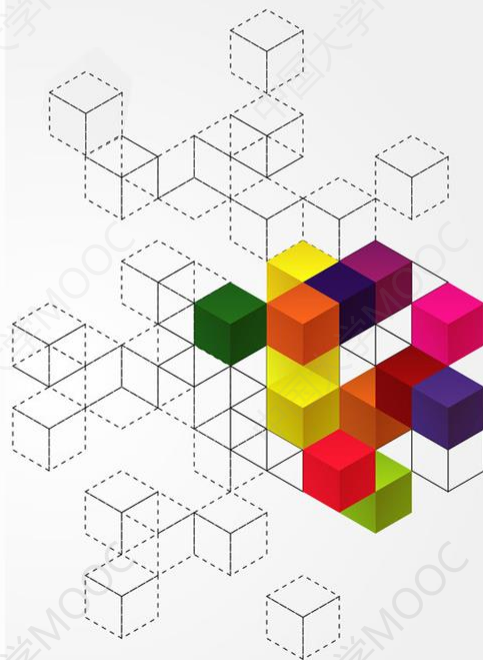
大连理工大学

一、调度基本概念

二、调度与进程队列

三、上下文切换

四、调度算法层次



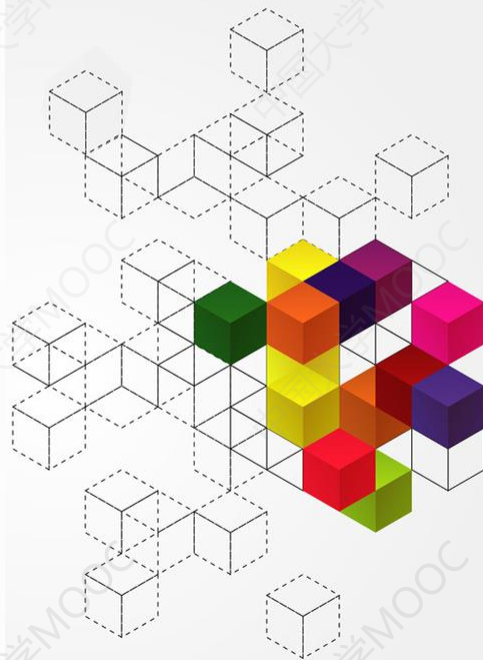
# 一、调度基本概念

## • 为什么要进行调度

- 多任务竞争使用CPU资源，需要对它们进行协调
- 进程不断在变换状态，操作系统需要对不同状态的任务进行管理

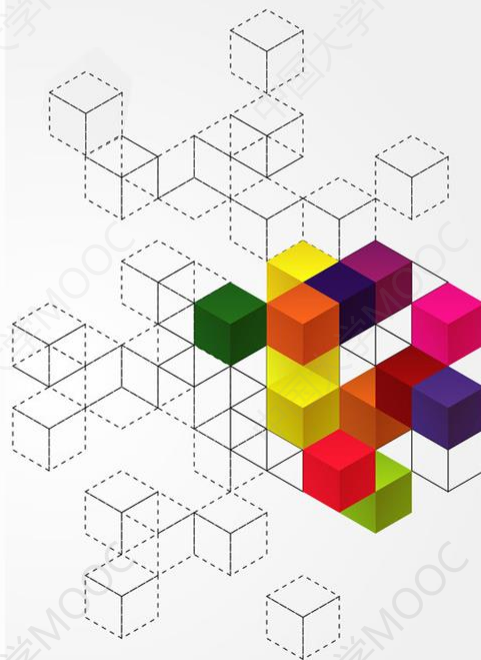
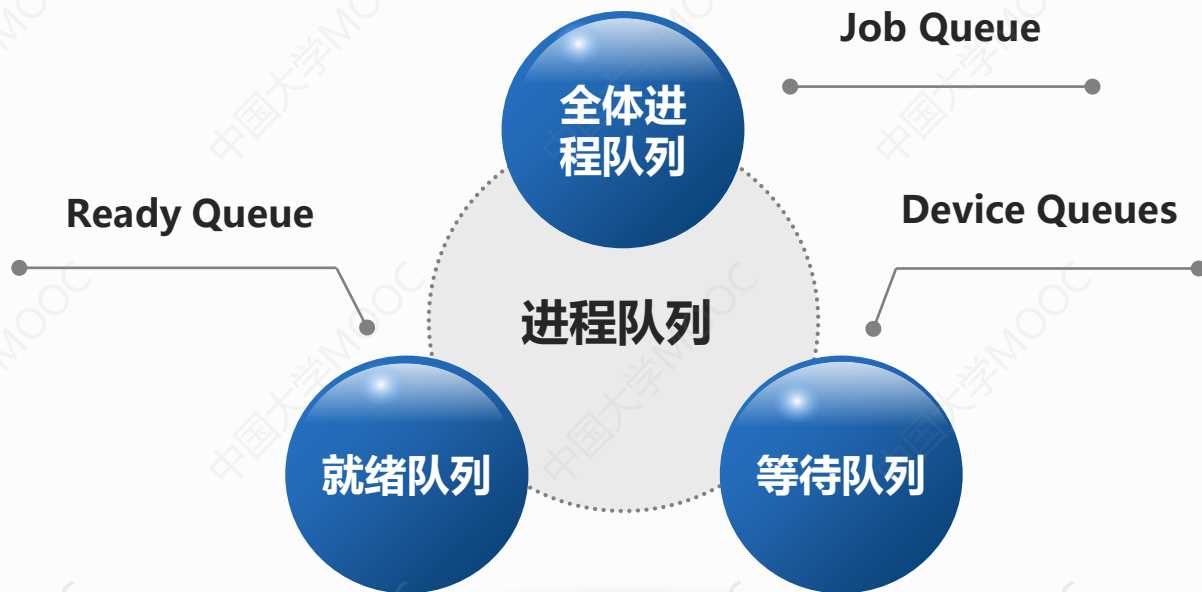
## • 调度的基本要求

- 主要目标：提高CPU利用率 (Maximize CPU Usage)
- 在不同进程间快速切换，使得多个进程可以分时共享CPU资源
- 调度器要在合适的时间点，从就绪的进程中选择下一个在CPU上执行的进程



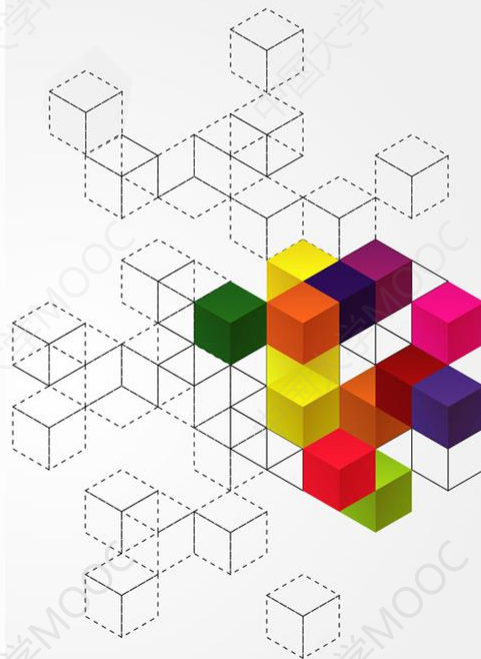
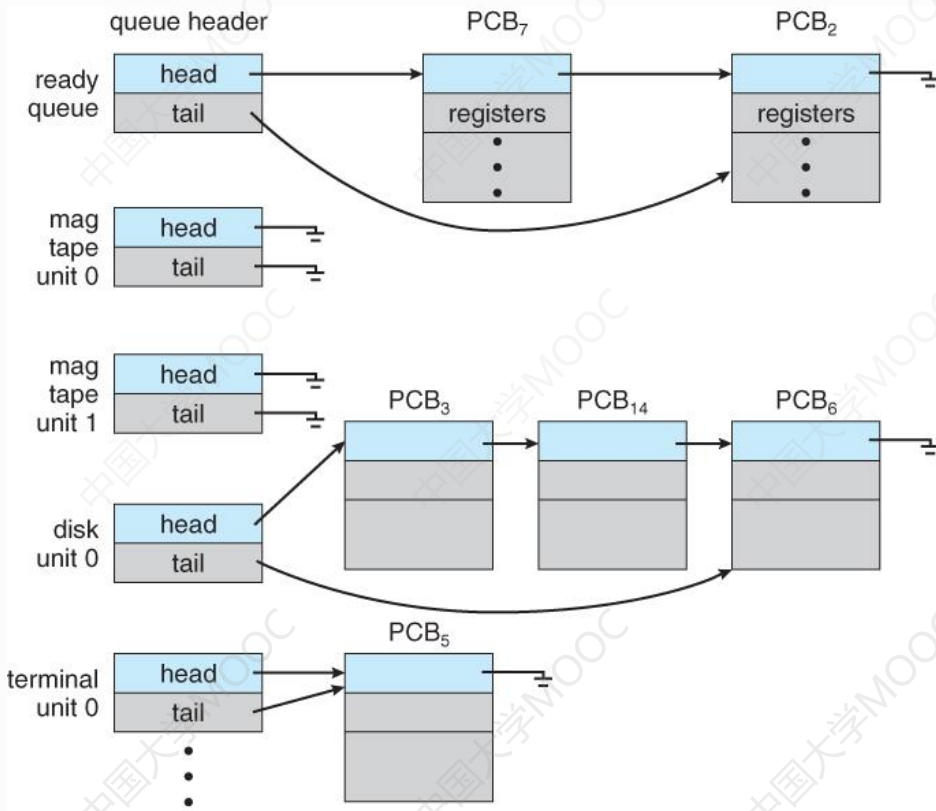
## 二、调度与进程队列

- 调度器需要操作和维护的进程队列包括



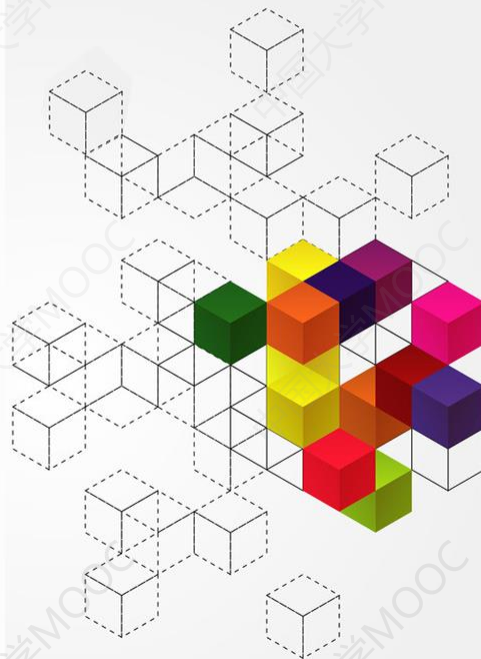
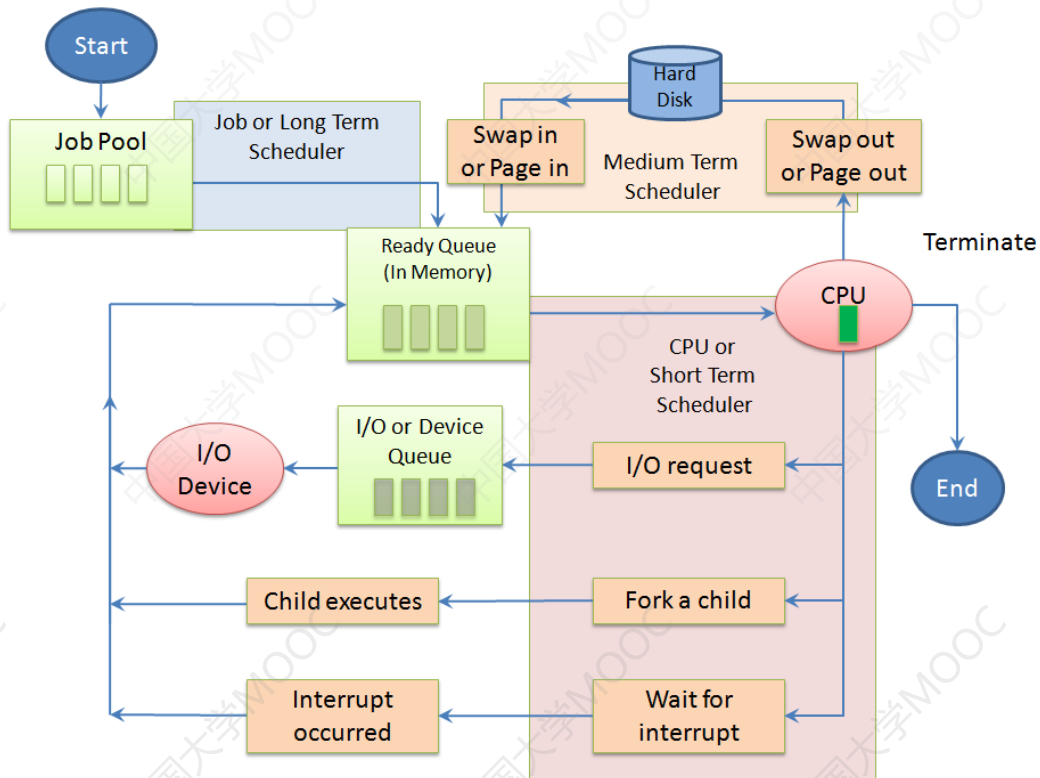
## 二、调度与进程队列

### • 就绪队列与等待队列示意图



## 二、调度与进程队列

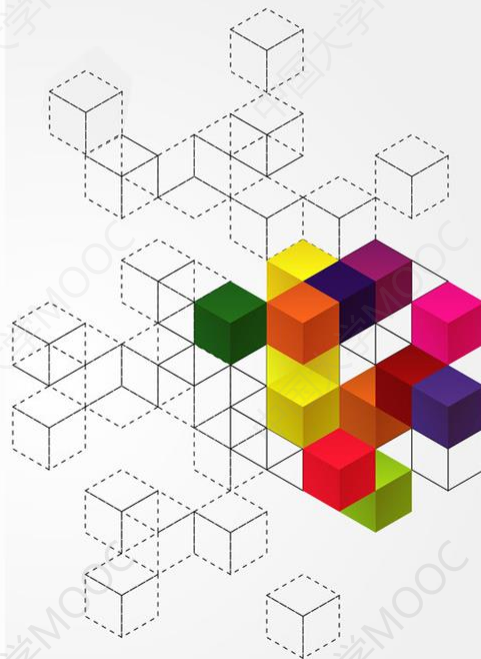
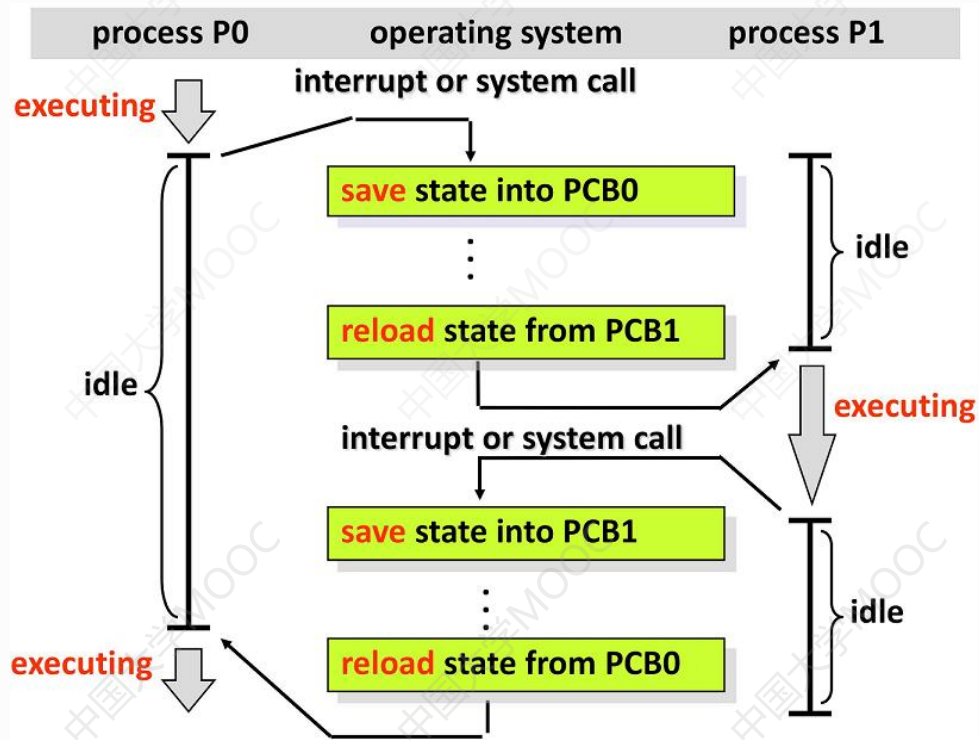
### • 在调度过程中，进程在不同队列之间的迁移图





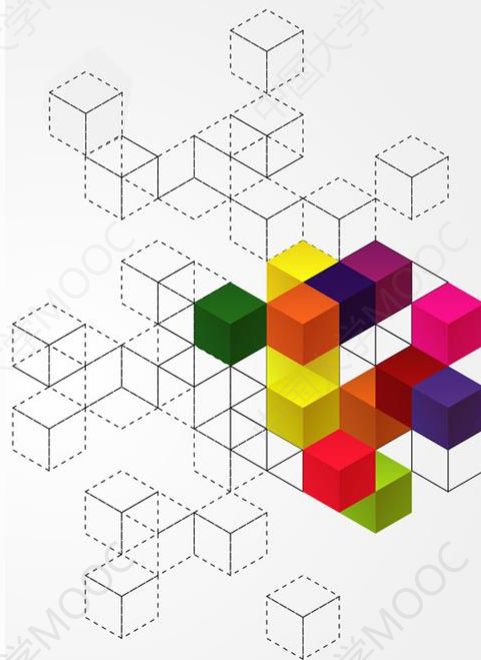
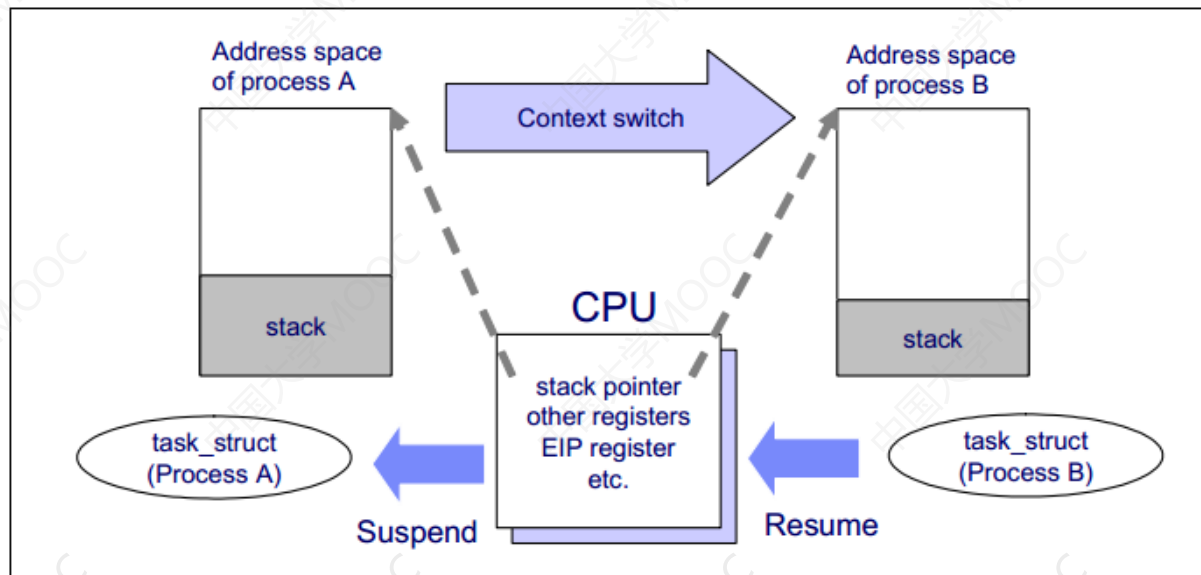
### 三、上下文切换

- 当进程获取对CPU控制，以及进程退出对CPU控制时，要保存进程执行的现场（又称**上下文**）



### 三、上下文切换

#### • 上下文切换细节

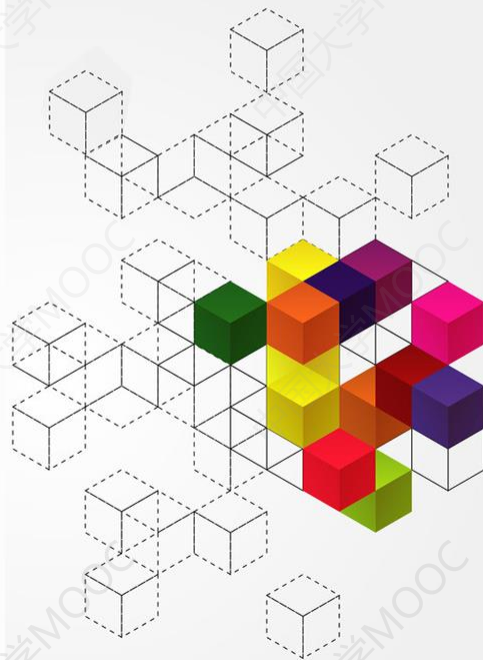




## 四、调度算法层次

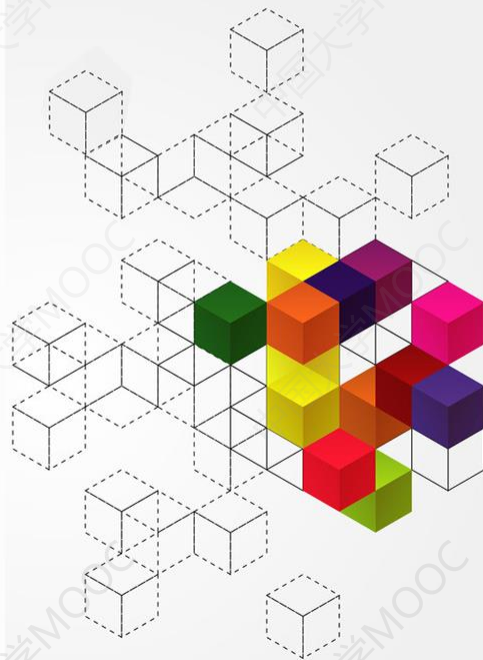
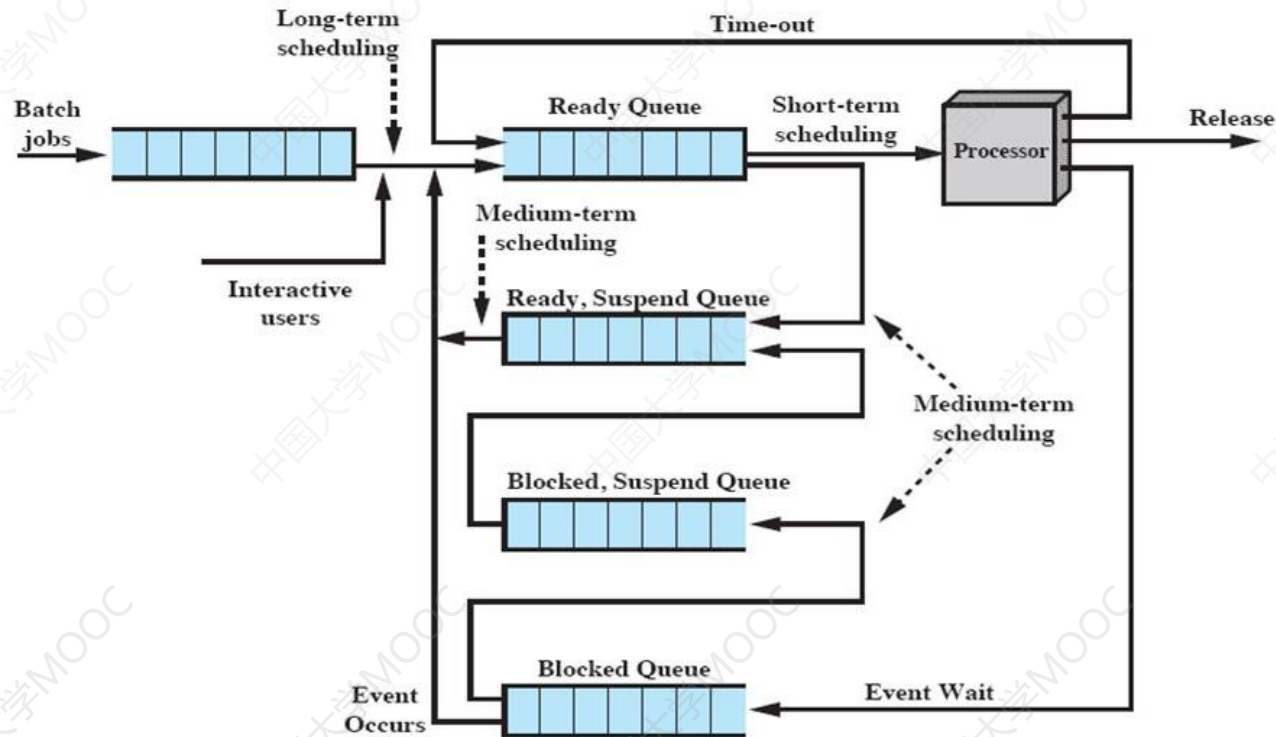
### • 调度分为几个层次

- **长程调度 (Long-Term Scheduling)** : 控制如何将外存中的作业加载到内存
- **中程调度 (Medium-Term Scheduling)** : 控制如何对进程实施换入/换出操作, 以便控制内存使用率
- **短程调度 (Short-Term Scheduling)** : 从就绪队列中挑选下个在CPU上执行的进程



## 四、调度算法层次

### • 调度层次与进程队列关系示意图



# 本讲小结

- 调度基本概念
- 调度与进程队列
- 上下文切换
- 调度算法层次

