

第二章

并发进程

主要内容

2.1 进程基本概念

2.2 处理机调度与死锁

2.3 UNIX的进程

2.4 中断的基本概念及UNIX中断处理

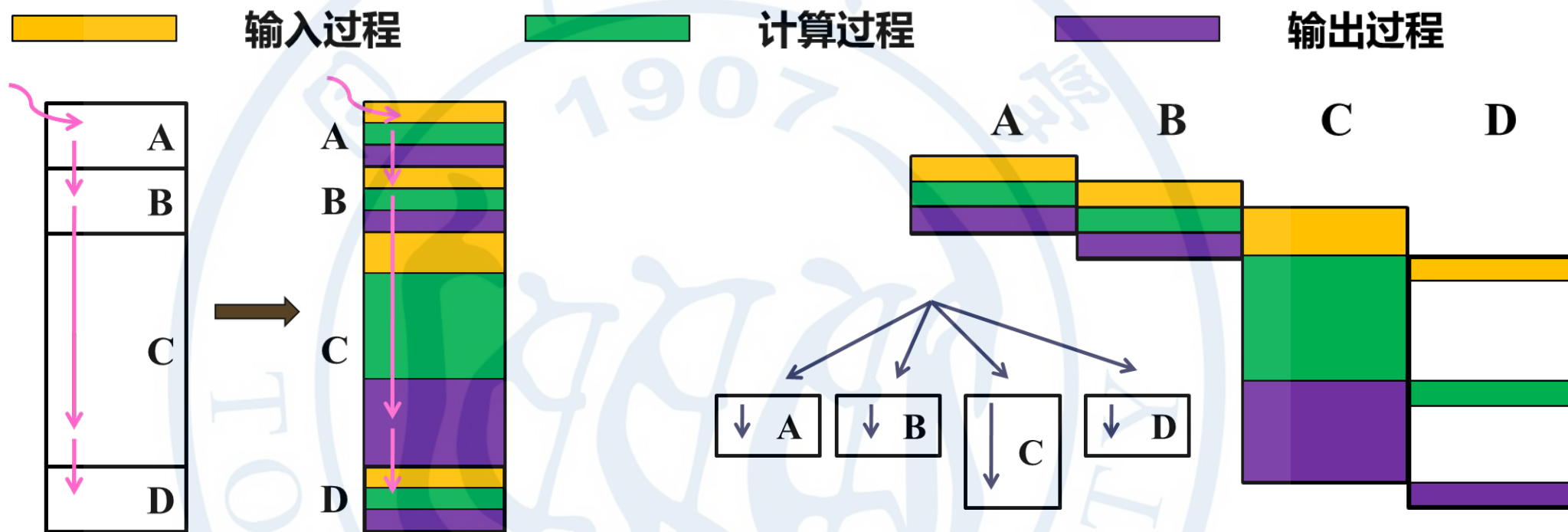
2.5 进程通信



单道批处理系统

VS.

多道批处理系统



封闭：程序执行时独占全机，结果不受外界影响。

可再现：只要执行时的环境和初始条件相同，结果即相同。

间断：相互制约导致并发程序具有“**执行—暂停—执行**”这种间断性的活动规律。

开放：多个程序共享系统中的资源。

不可再现：结果与并发程序的执行速度有关。



进程的基本概念



 资源共享  各种程序活动的相互依赖与制约

为了解决程序并发执行带来的问题：



一组数据与指令代码的集合

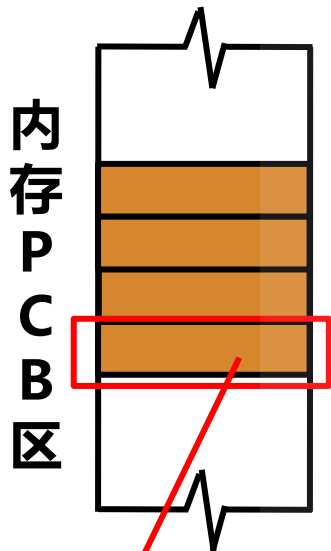
结构特征
代码段、数据段、堆
栈段、**进程控制块**



进程的基本概念



Process Control Block, PCB



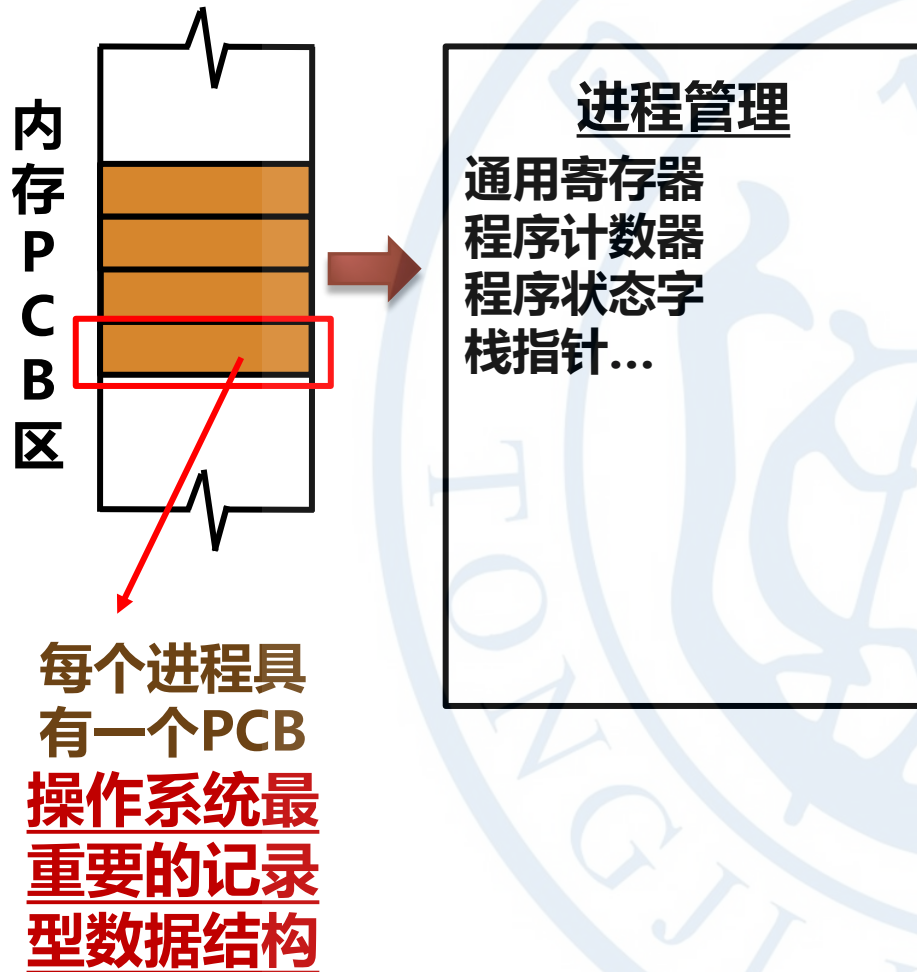
每个进程具
有一个PCB
操作系统最
重要的记录
型数据结构



进程的基本概念



Process Control Block, PCB

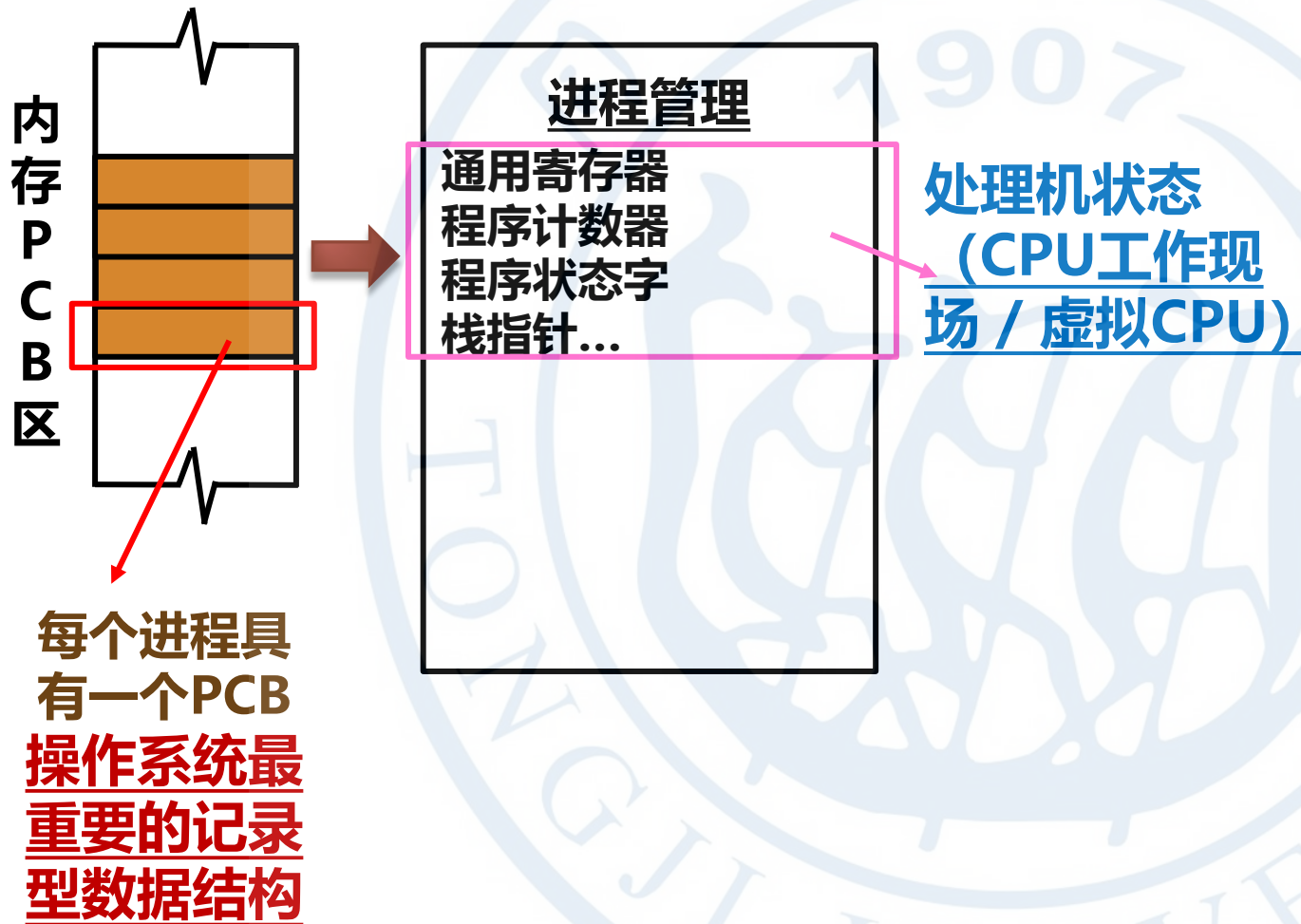




进程的基本概念



Process Control Block, PCB

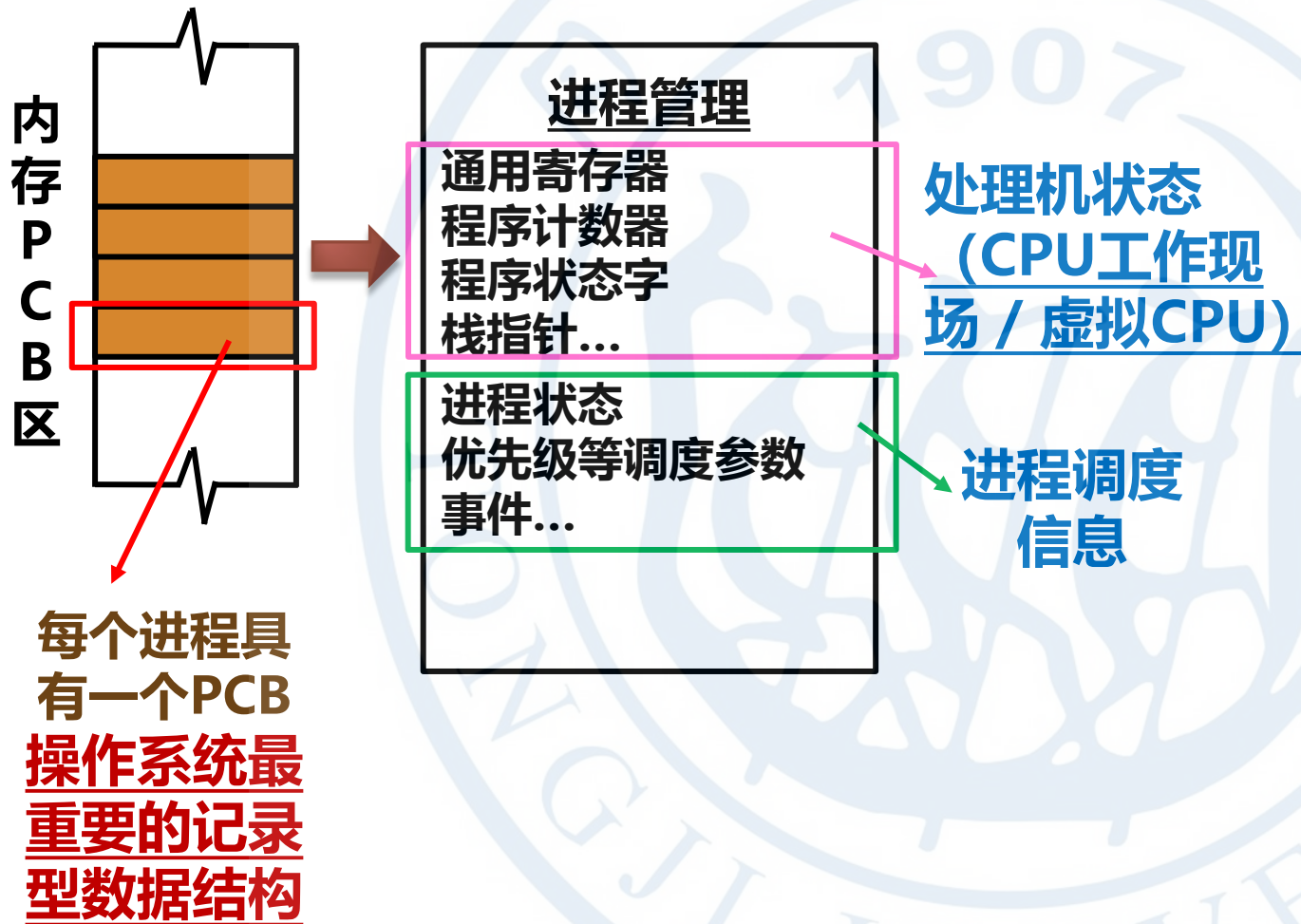




进程的基本概念



Process Control Block, PCB

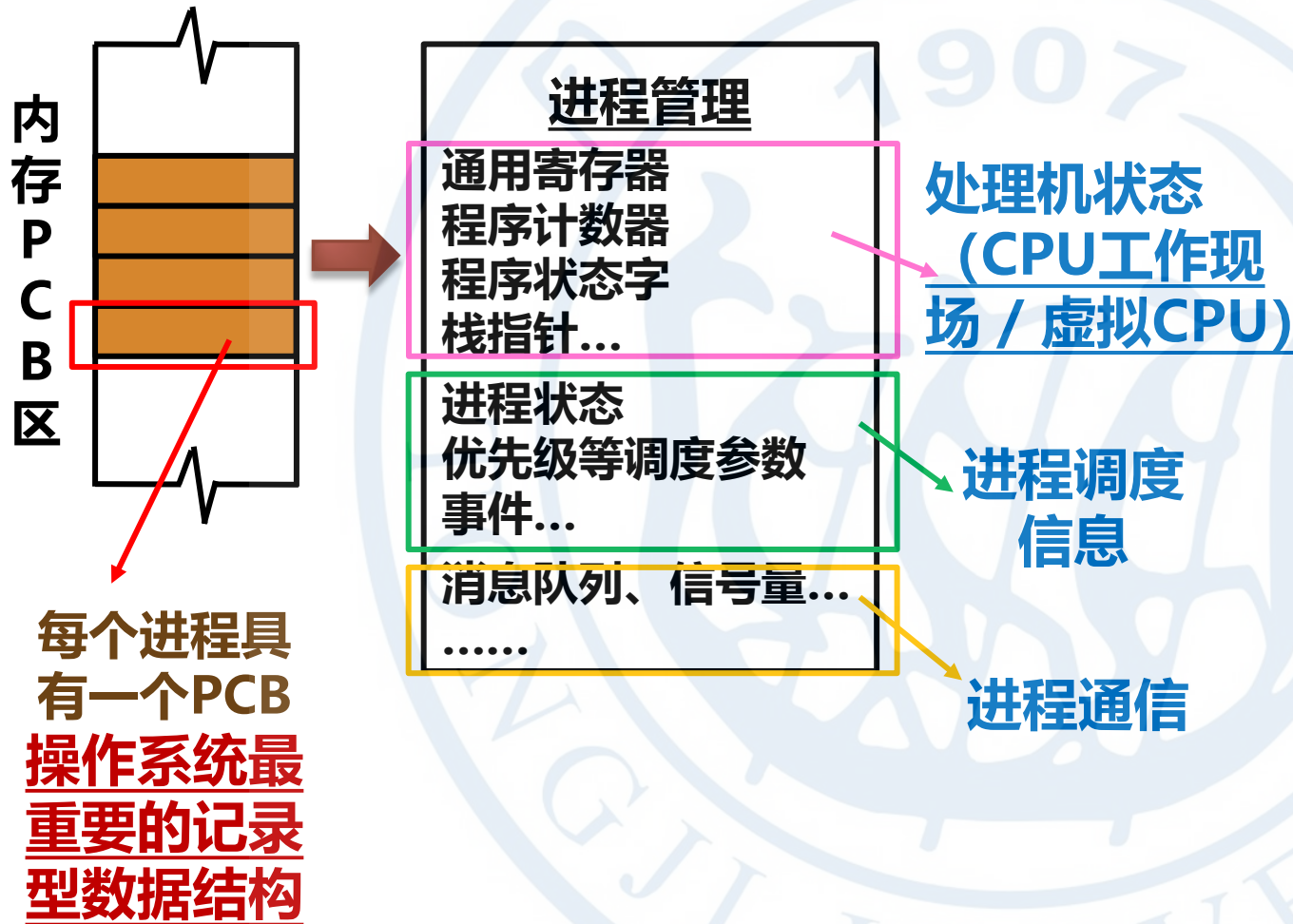




进程的基本概念



Process Control Block, PCB

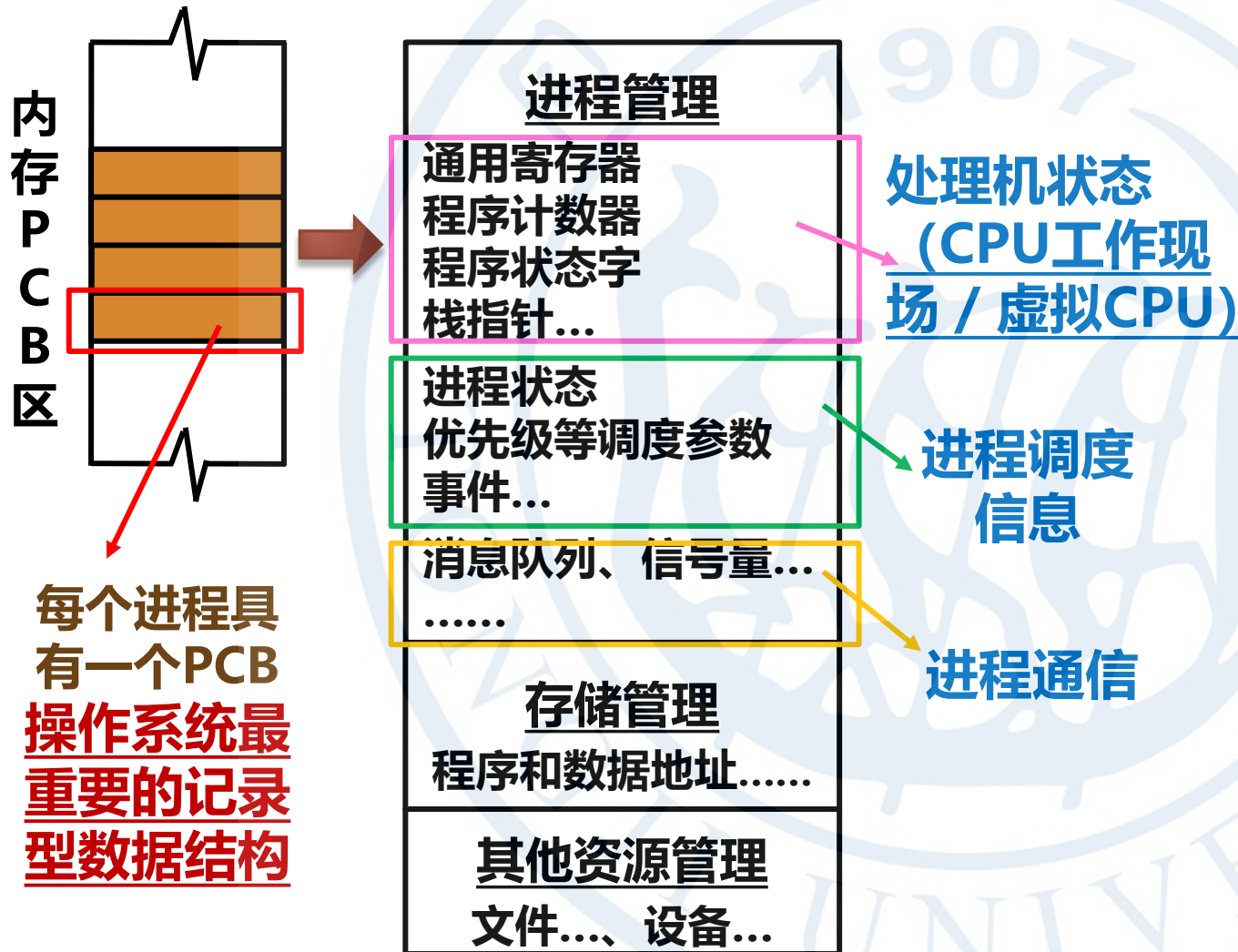




进程的基本概念



Process Control Block, PCB



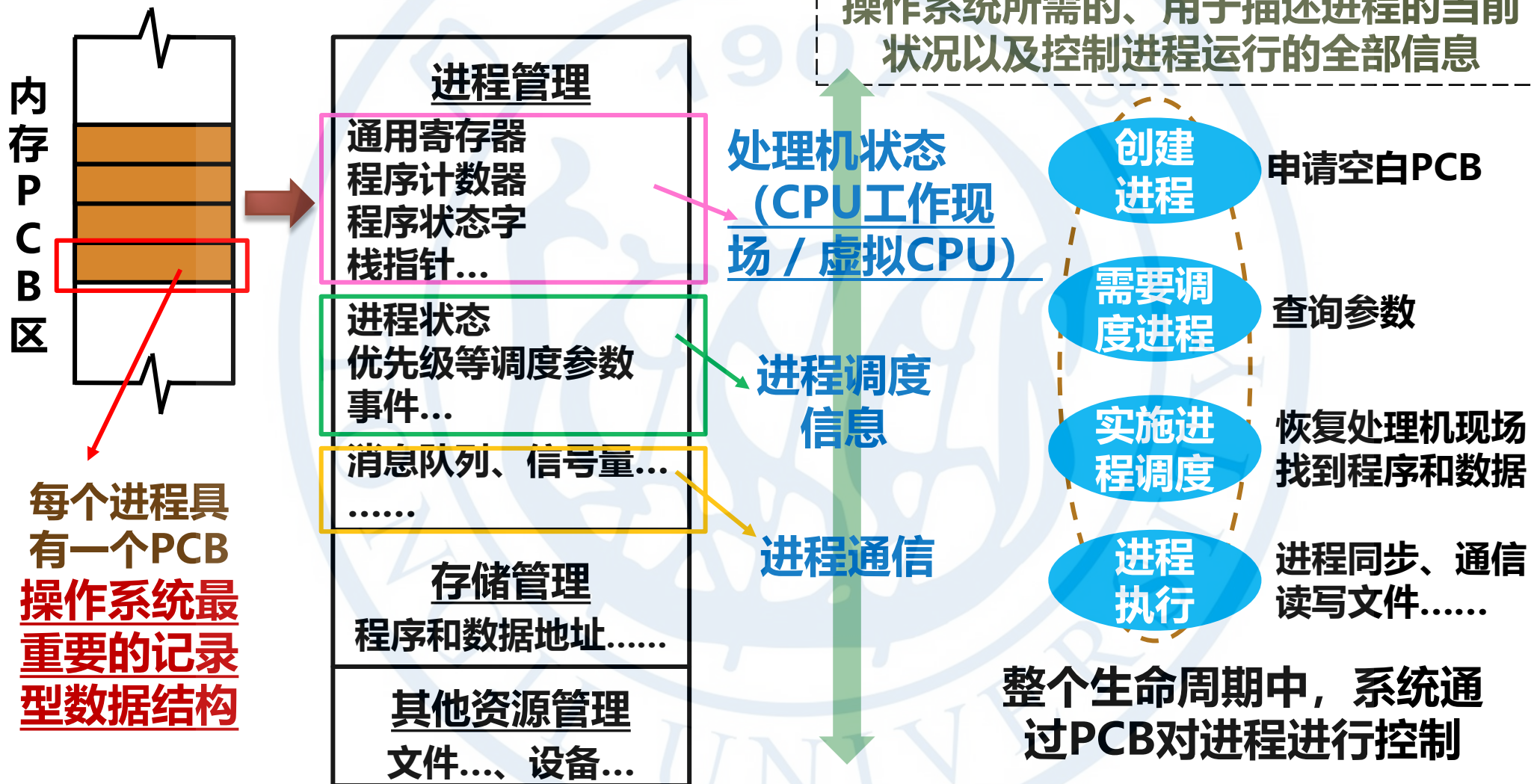


进程的基本概念



Process Control Block, PCB

进程控制块



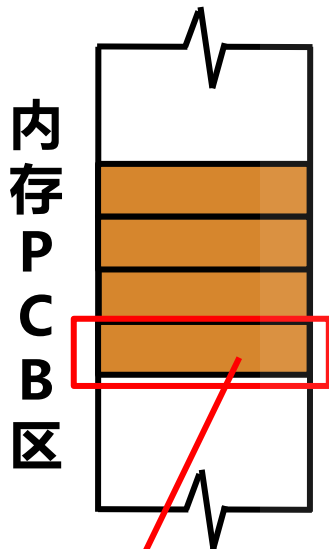


进程的基本概念



Process Control Block, PCB

进程控制块

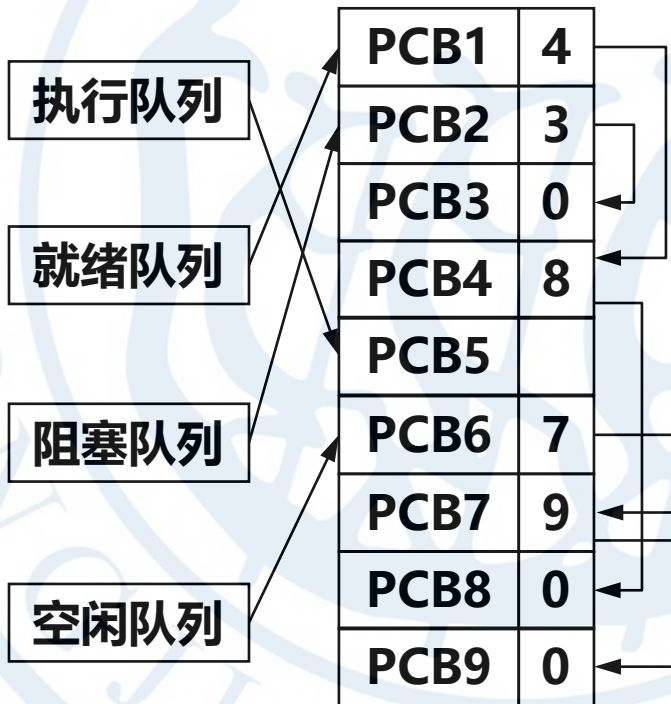


每个进程具有一个PCB
操作系统最重要的记录
型数据结构

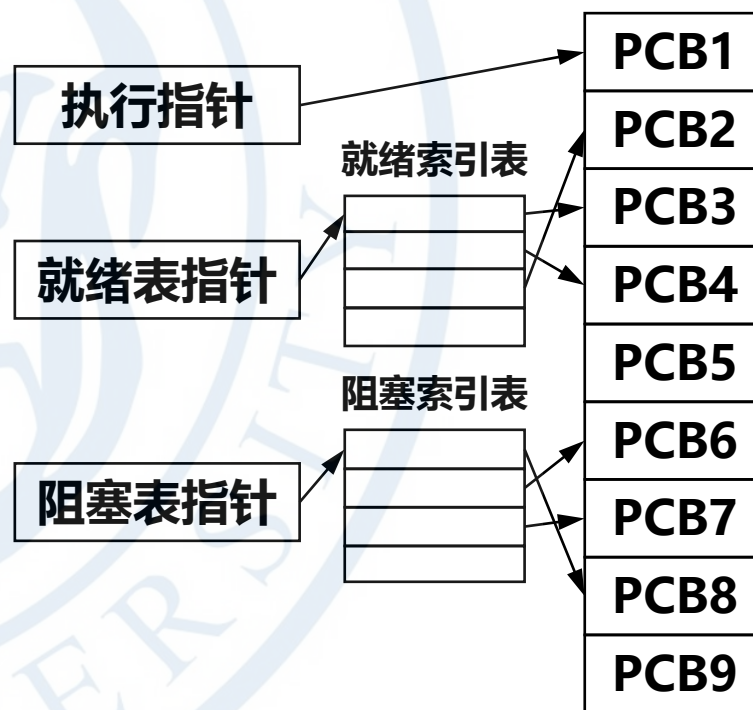
重要性高、访问频率高、数量巨大

高效的数据组织方式

PCB组织方式



按队列方式组织PCB



按索引方式组织PCB



进程的基本概念



资源共享



各种程序活动的相互依赖与制约

为了解决程序并发执行带来的问题：



程序



进程

一组数据与指令代码的集合

结构特征

代码段、数据段、堆
栈段、**进程控制块**

静态的
存放在某种介质上

动态性，具有生命周期

“由创建而产生，由调度而
执行，由撤销而消亡”

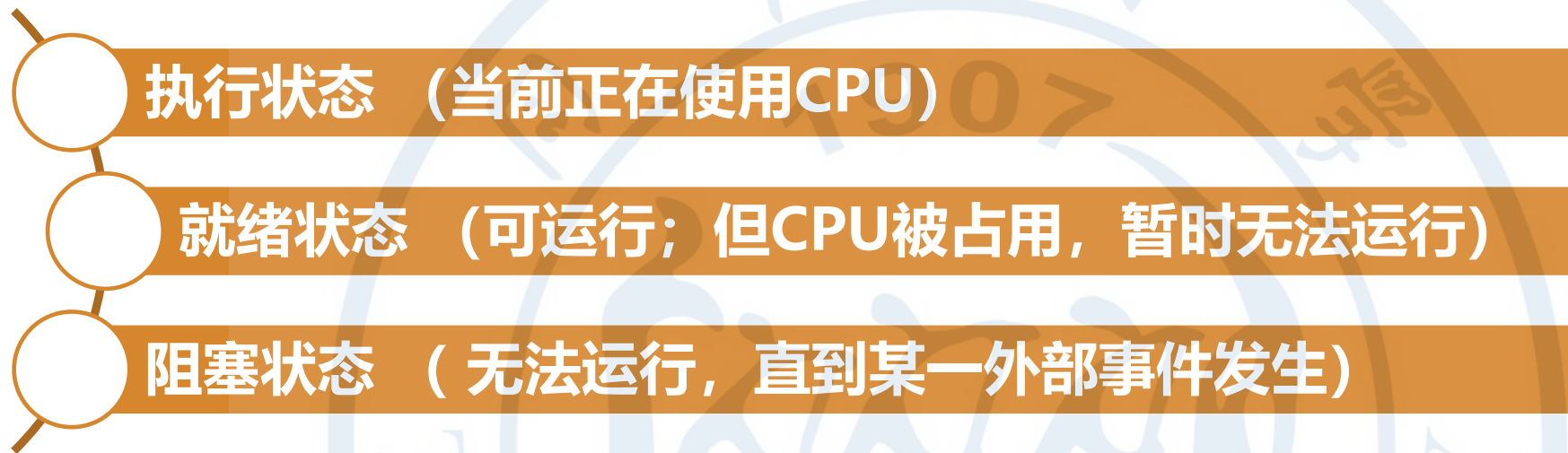
多道程序并发执行带来的问题



进程的基本概念



进程的三种调度状态



1. 进程被调度
2. 进程由于等待某种外部事件被阻塞
3. 等待的外部事件发生被唤醒
4. 将CPU让给另一个进程



进程的基本概念



排队等待叫号
(就绪状态, 等待调度)



就绪状态



执行状态



进程的三种调度状态



进程的基本概念



进程的三种调度状态

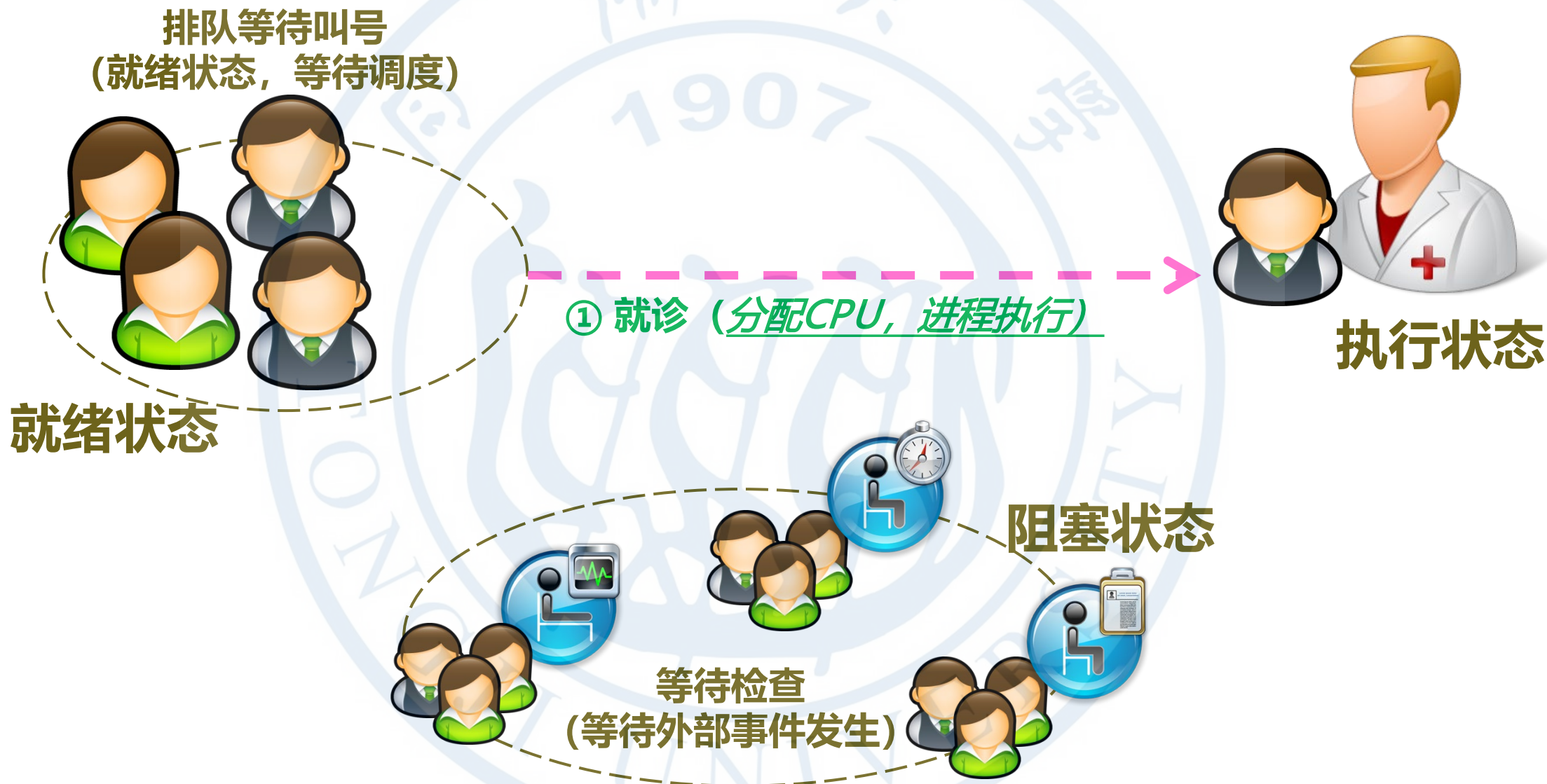




进程的基本概念



进程的三种调度状态

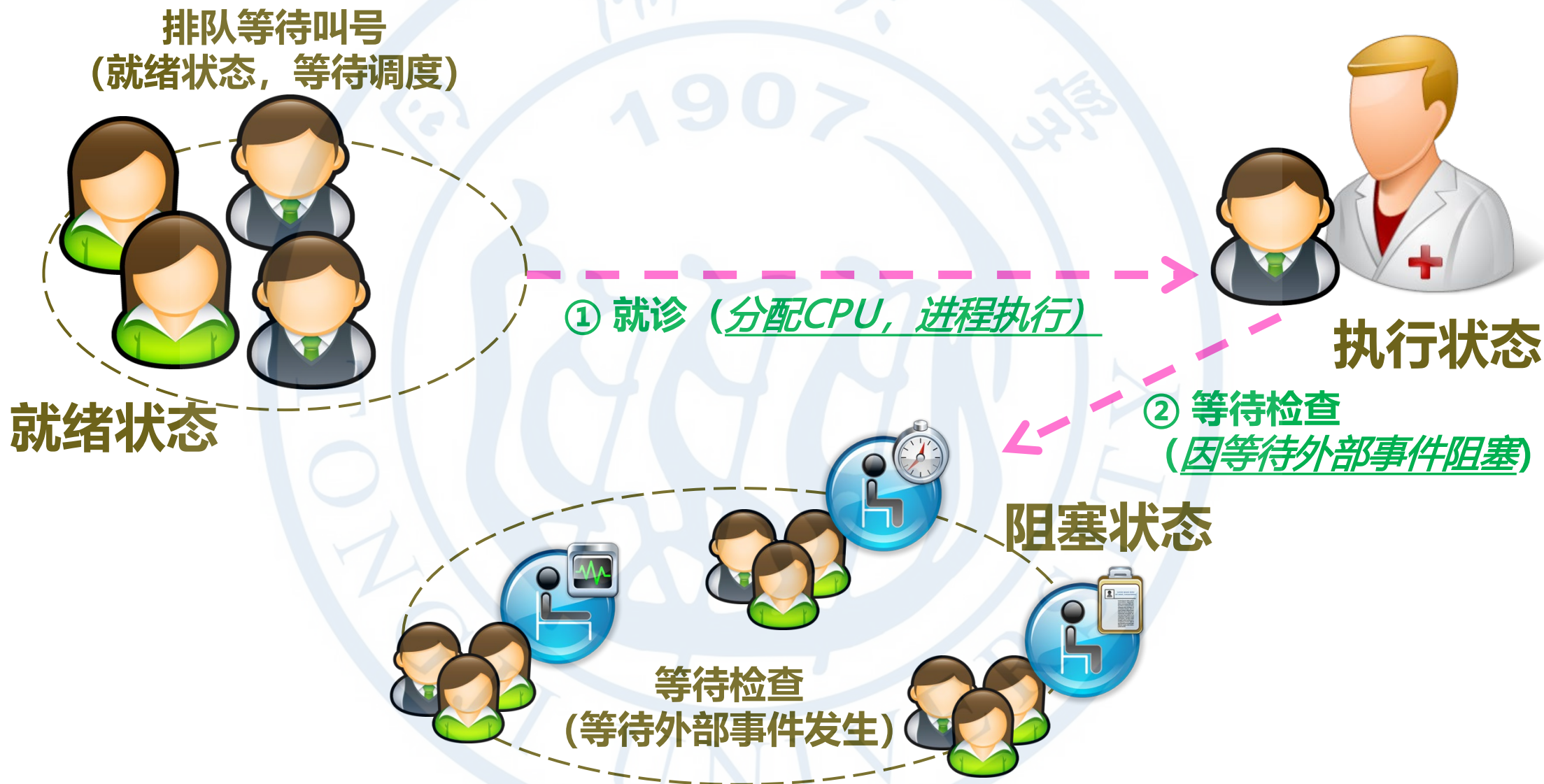




进程的基本概念



进程的三种调度状态

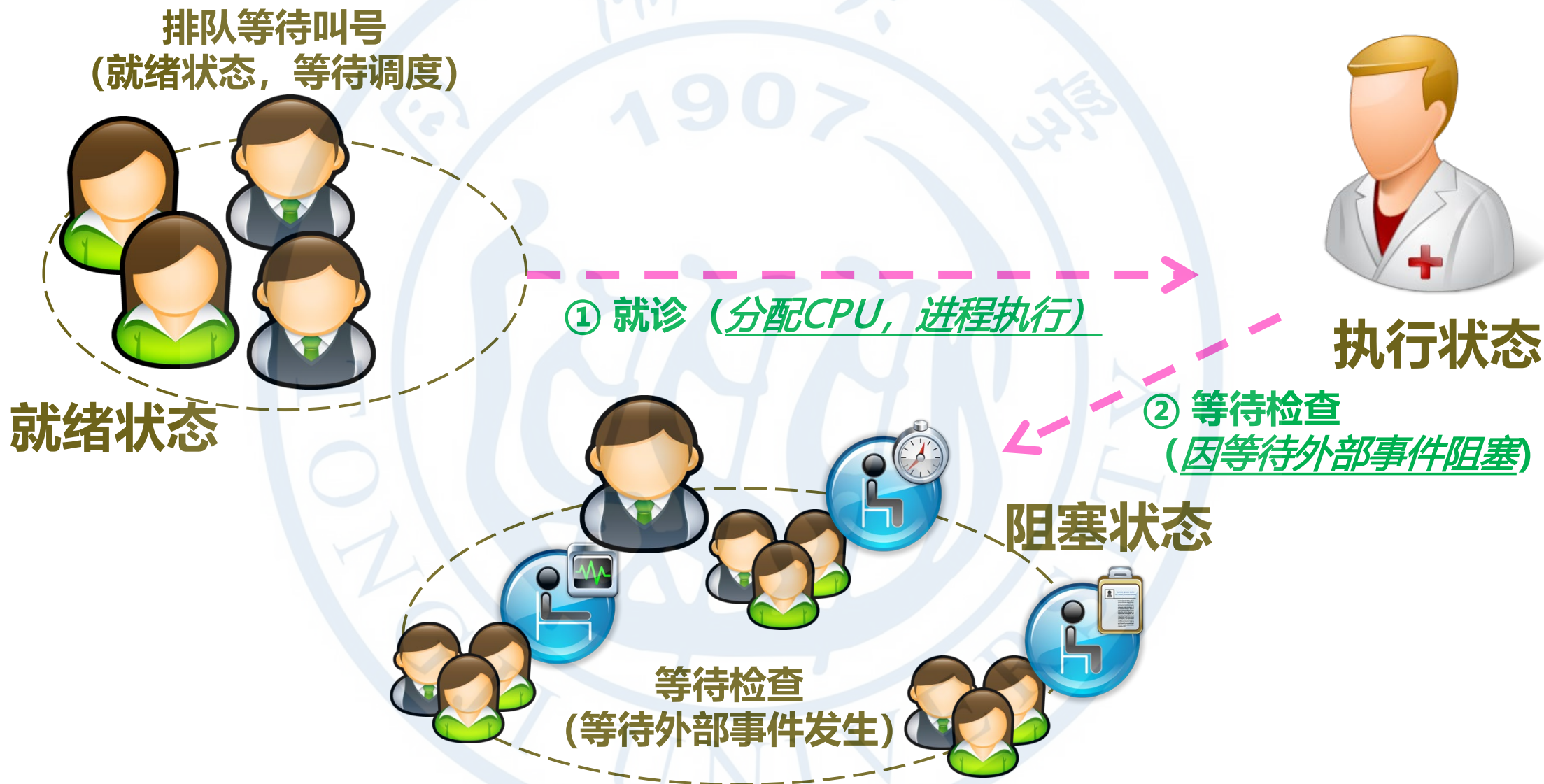




进程的基本概念



进程的三种调度状态

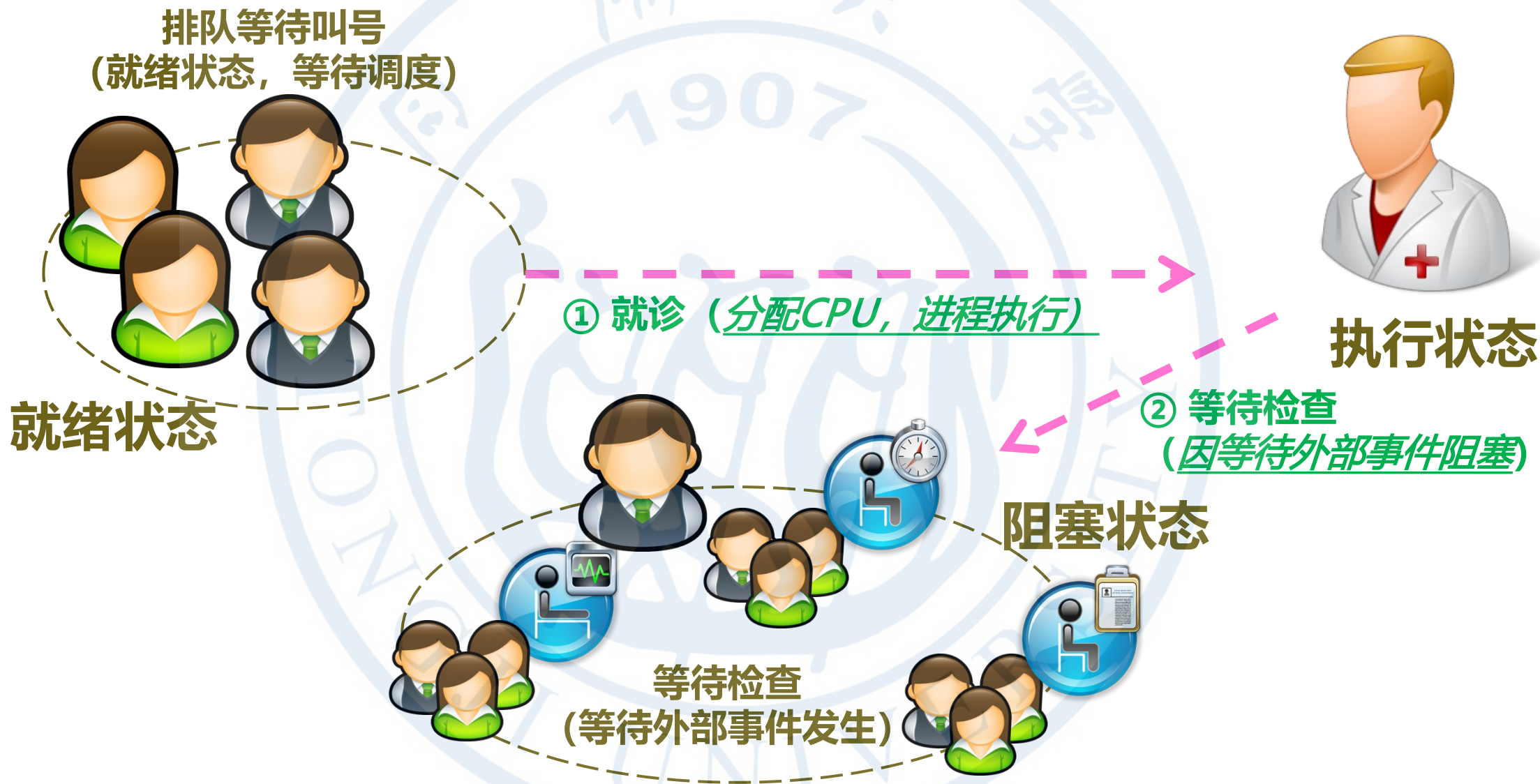




进程的基本概念



进程的三种调度状态

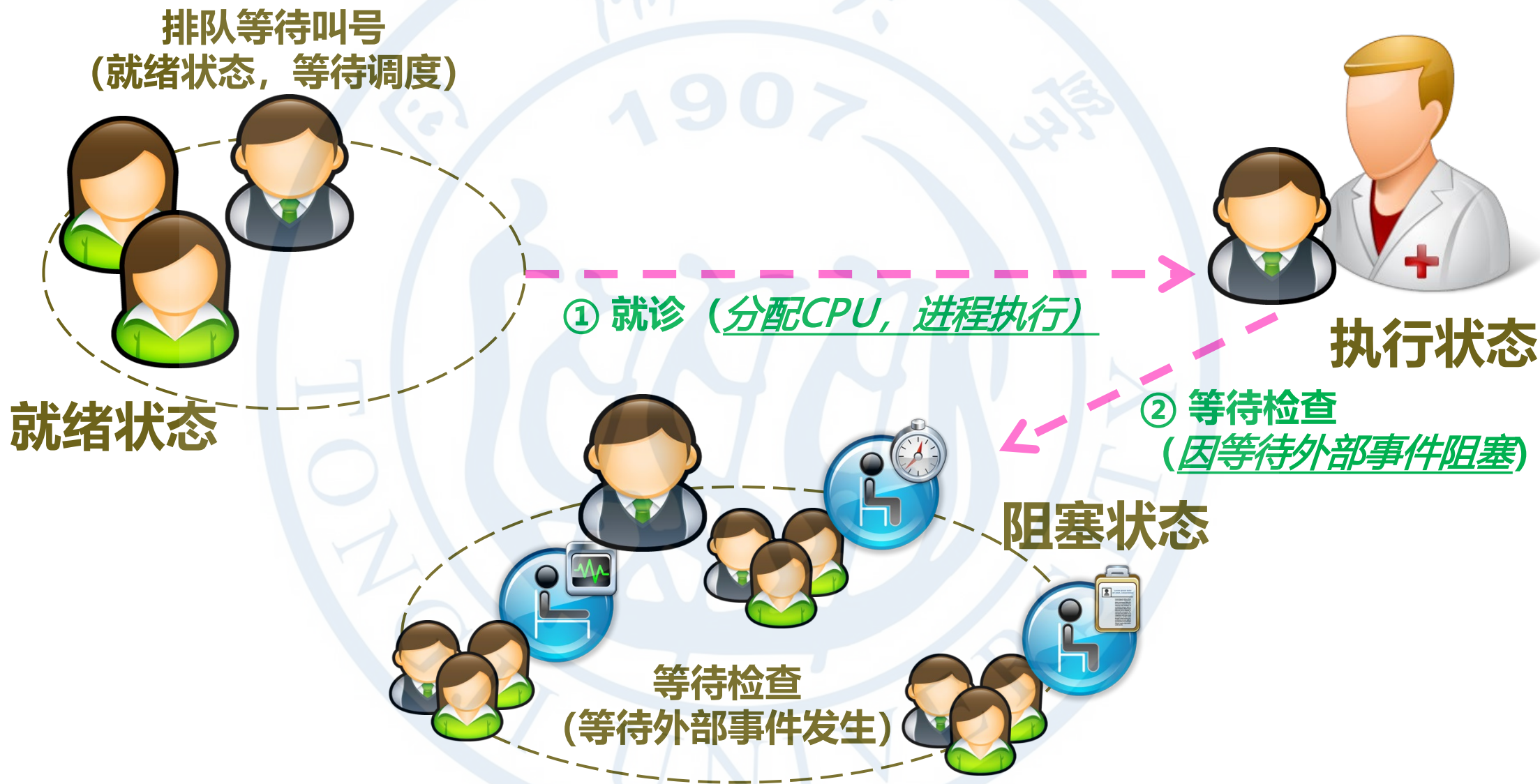




进程的基本概念



进程的三种调度状态

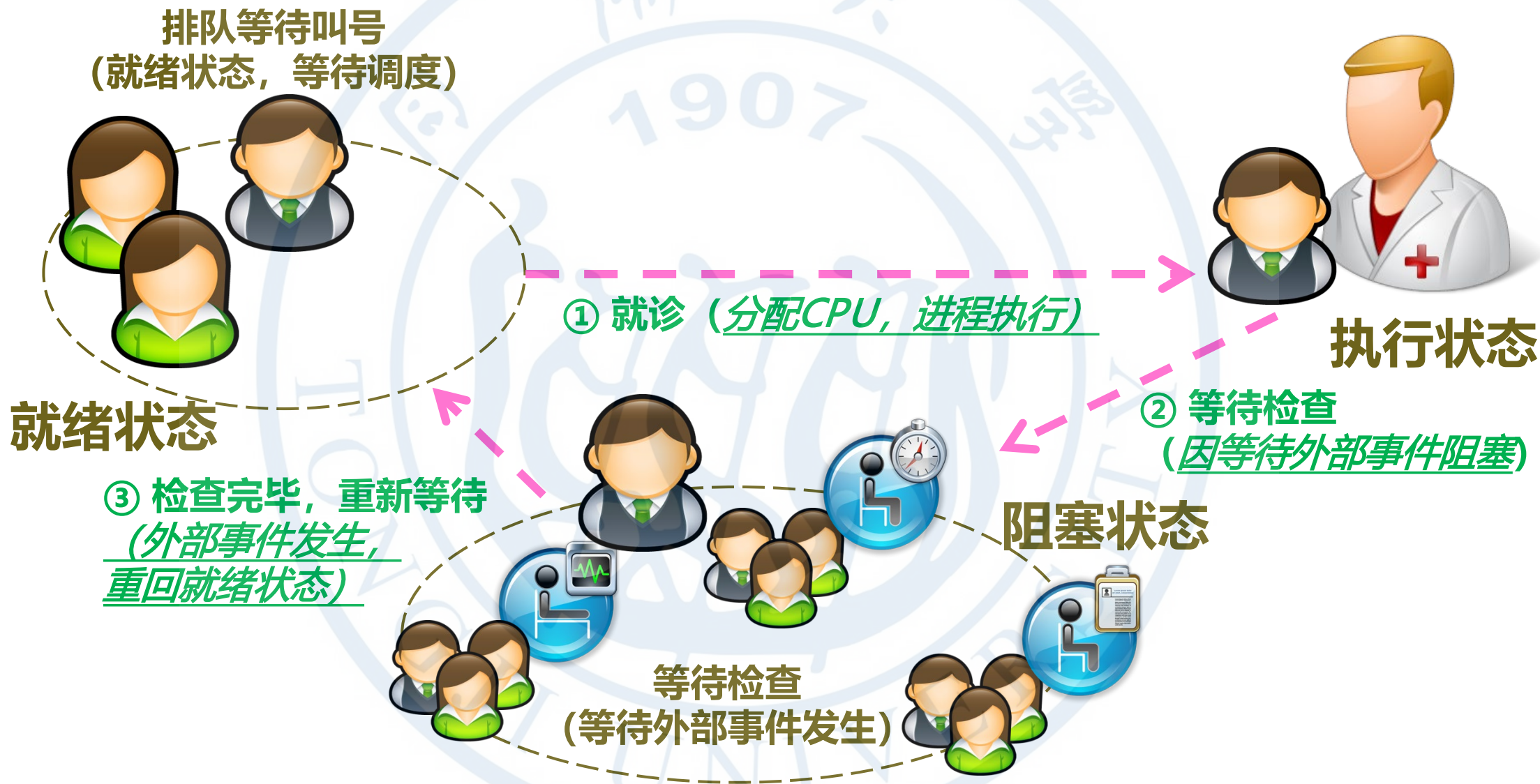




进程的基本概念



进程的三种调度状态

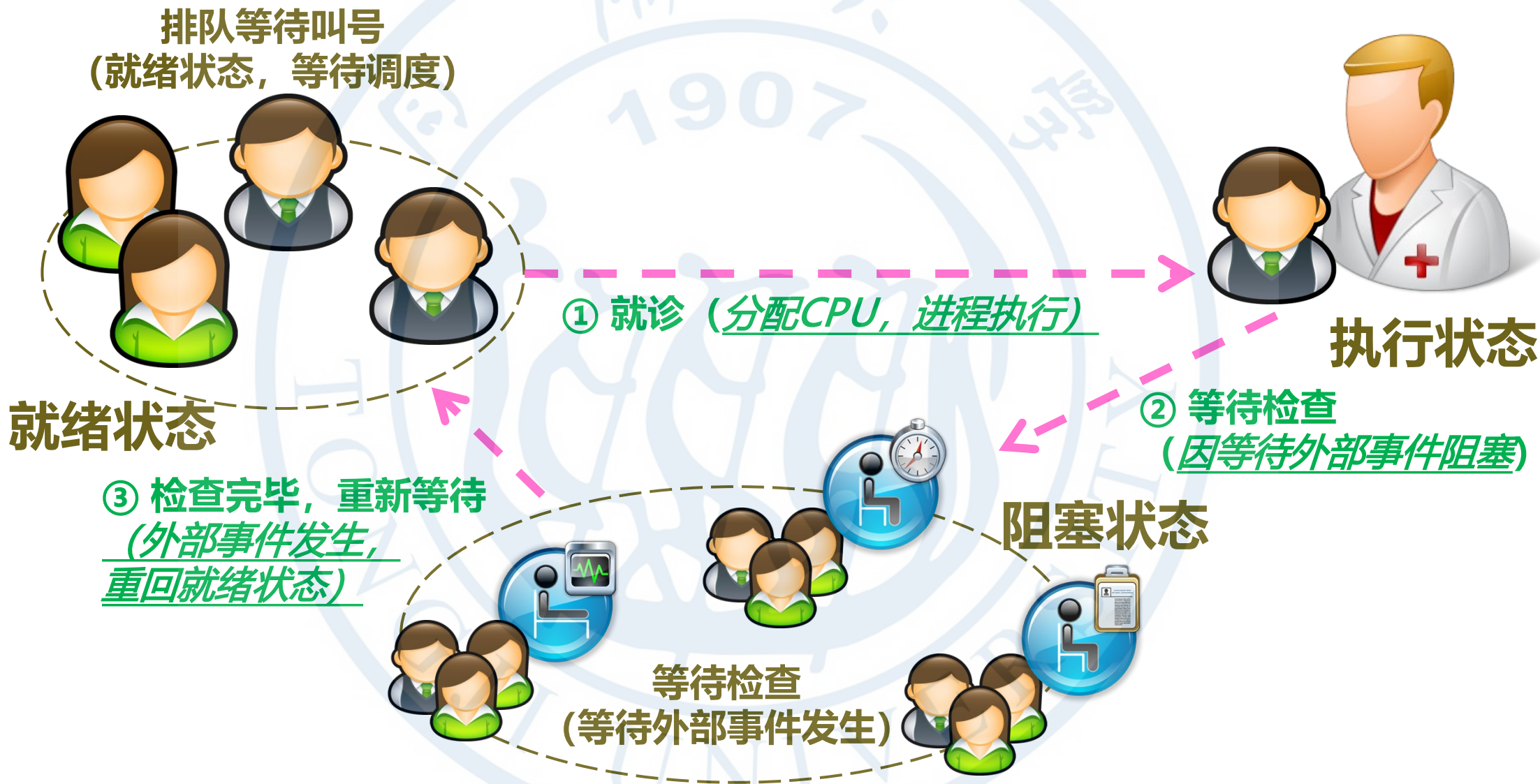




进程的基本概念



进程的三种调度状态

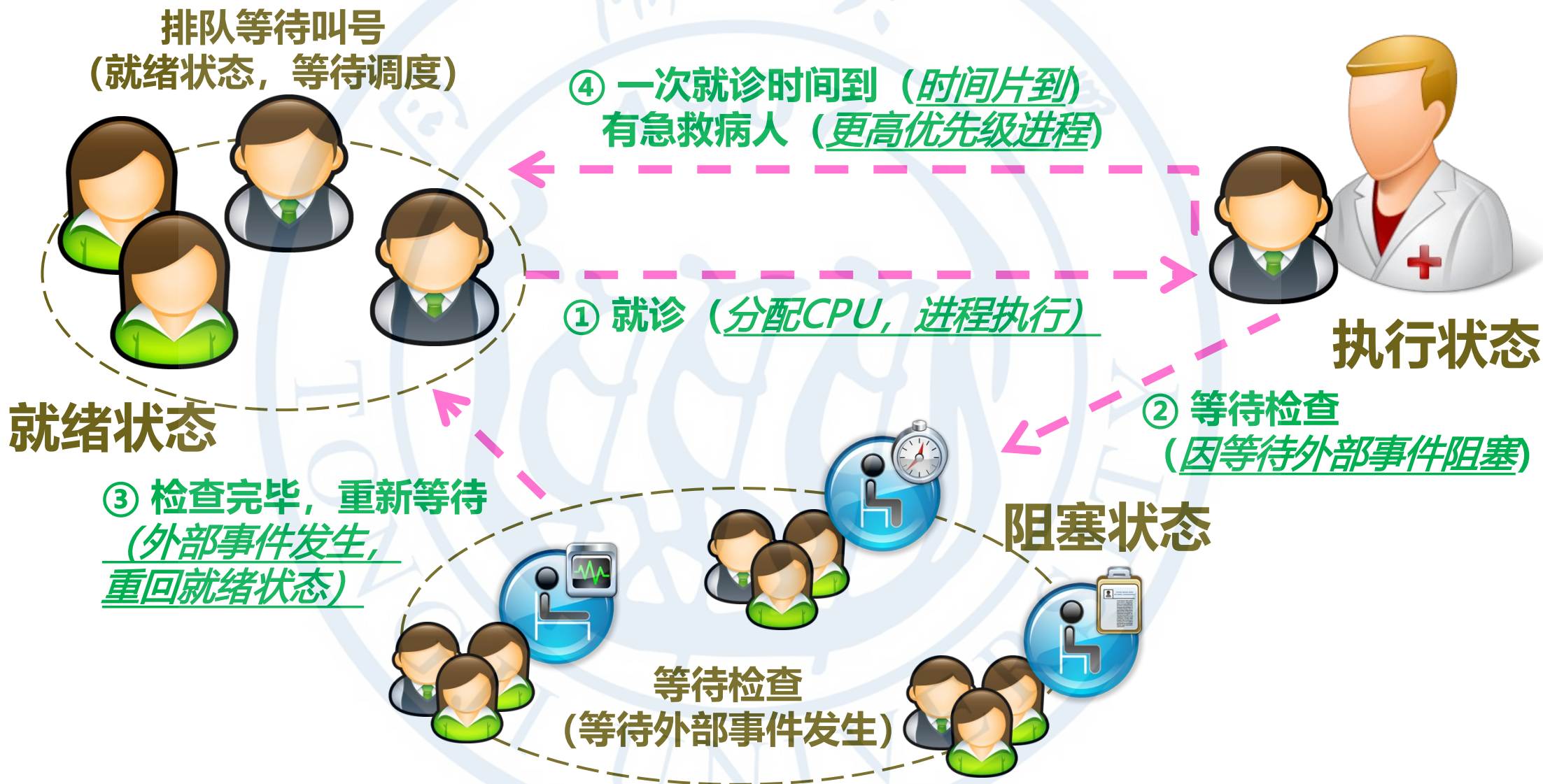




进程的基本概念



进程的三种调度状态

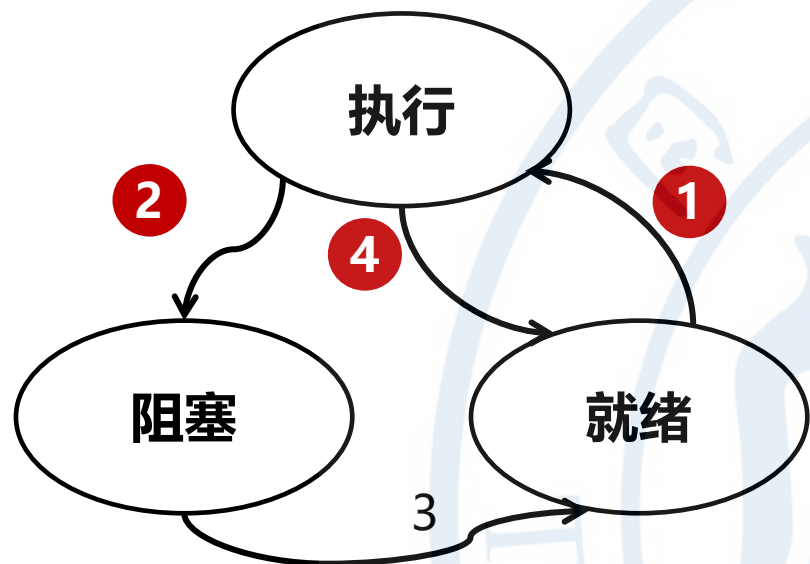




进程的基本概念



进程的调度控制



进程“下台” / “上台”

引起进程切换调度的事件：

(不同的调度方式会不同)

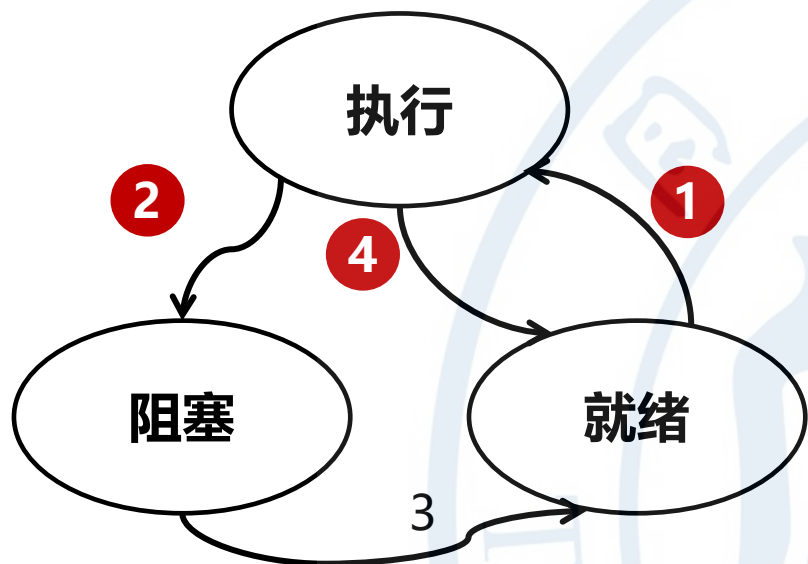
1. 进程时间片到
2. 有更高优先级的进程就绪
3. 进程阻塞，无法继续执行



进程的基本概念



进程的调度控制



进程“下台” / “上台”

引起进程切换调度的事件：
(不同的调度方式会不同)

1. 进程时间片到
2. 有更高优先级的进程就绪
3. 进程阻塞，无法继续执行

抢占式/剥夺式调度

现运行进程暂停，PCB中的调度状态

4 “执行” → “就绪”

非抢占式/进程主动放弃

现运行进程暂停，PCB中的调度状态

2 “执行” → “阻塞”

不同的调度方式

实时性高
但开销大

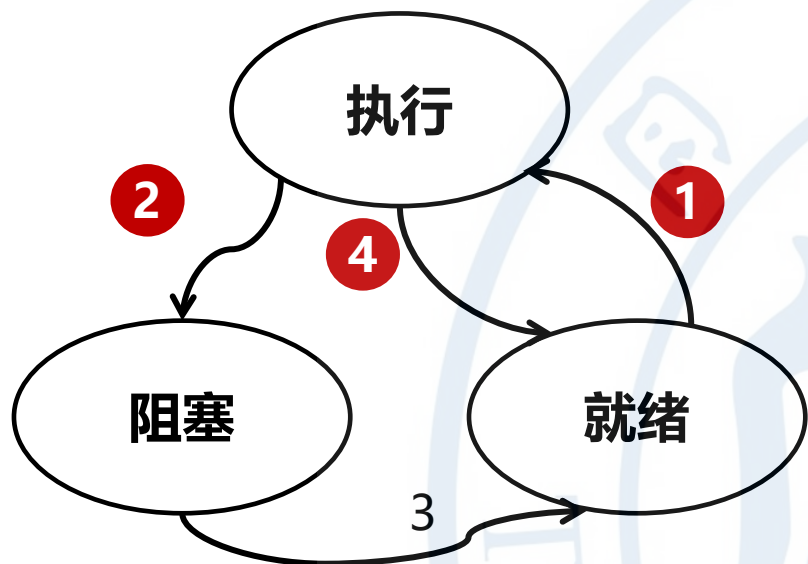
实现简单，开销小
难于满足紧急任务的
需求



进程的基本概念



进程的调度控制



进程“下台” / “上台”

引起进程切换调度的事件：
(不同的调度方式会不同)

1. 进程时间片到
2. 有更高优先级的进程就绪
3. 进程阻塞，无法继续执行

执行进程切换调度（由**调度程序**完成）：

1. **保存**现执行进程**工作现场**信息在其PCB中
2. **选择**另一个**就绪进程**，状态“**就绪**” → “**执行**” ①
3. 用该进程PCB中的工作现场信息**恢复现场**

进程的上下文切换

“下台” 进程未来某时刻会被调度程序重新选中而“上台”

抢占式/剥夺式调度

现运行进程暂停，PCB中的调度状态

④ “**执行**” → “**就绪**”

非抢占式/进程主动放弃

现运行进程暂停，PCB中的调度状态

② “**执行**” → “**阻塞**”



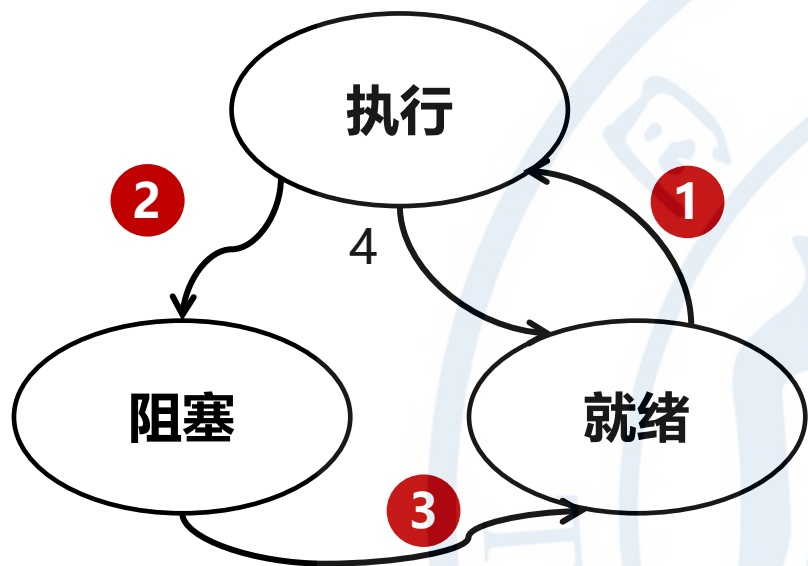
- 调度算法
- 调度时机
- 调度过程
- 进程死锁



进程的基本概念



进程的调度控制



进程的阻塞与唤醒

引起进程阻塞的事件:

1. 请求系统服务
 2. 启动某个操作
 3. 无新工作可做
 4.
- (进程无法再继续执行下去)

进程**阻塞过程** (由**阻塞程序**完成) :

1. 立即停止执行
2. PCB中的进程状态 “**执行**” → “**阻塞**” ②
3. PCB进入阻塞队列
4. 由**调度程序**完成进程**切换调度** ①



进程不能永远“睡觉”，必须在某个时间被唤醒，两个过程必须成对出现

进程**唤醒过程** (由**唤醒程序**完成) :

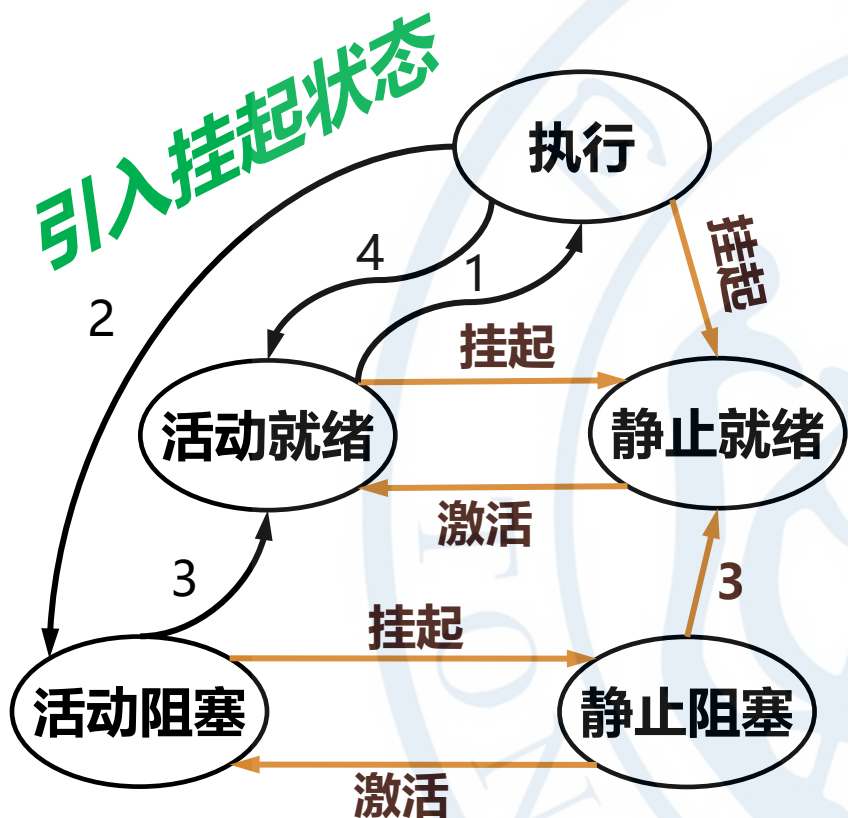
1. 将PCB从阻塞队列中移出
2. PCB中的进程调度状态 “**阻塞**” → “**就绪**” ③
3. 由调度算法决定是否**切换调度**



进程的基本概念



进程的调度控制



引起进程挂起的事件：
终端用户请求 父进程请求
操作系统负荷调节

进程挂起过程：

1. 若为当前执行进程：立即停止执行，PCB中的进程状态“执行”→“静止就绪”，调度程序进行**切换调度**
2. 若非当前执行进程：PCB中的进程状态“活动就绪”→“静止就绪” / “活动阻塞”→“静止阻塞”

进程激活过程：

1. PCB中的进程状态“静止就绪”→“活动就绪” / “静止阻塞”→“活动阻塞”
2. 若转入“活动就绪”，则PCB进入就绪队列，由调度算法决定是否**切换调度**

两个过程也必须成对出现



进程的基本概念



进程的调度控制



引起进程创建的事件:

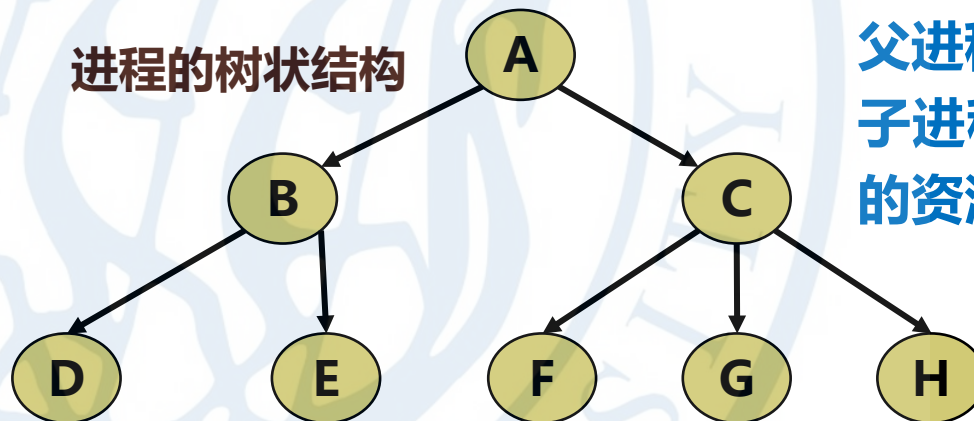
用户登录
提供服务

作业调度
应用请求

进程创建过程:

1. 申请空白PCB
2. 为进程分配资源 (内存空间)
3. PCB初始化 (标识、处理机状态、进程调度信息)
4. 进入就绪队列 (活动? 静止?)

进程的树状结构



父进程创建子进程
子进程拥有父进程的
资源

子进程撤销时, 资源归还父进程
父进程撤销时, 撤销所有子进程



进程的基本概念



进程的调度控制



引起进程终止的事件:

正常结束

异常结束

外界干预 (人为、父进程)

进程**终止过程**:

1. 从PCB中读出该进程的状态
2. 立即终止该进程的执行
3. 终止其所有子孙进程
4. 释放全部资源
5. 移除该进程PCB
6. 进程切换调度

只有当删除进程PCB后，进程才彻底消亡



进程

动态性，具有生命周期
“由创建而产生，由调度而执行，由撤销而消亡”

Very Important!



总结



为了解决程序并发执行带来的问题：



程序



进程

一组数据与指令代码的集合

结构特征

代码段、数据段、堆
栈段、**进程控制块**

静态的
存放在某种介质上

动态性，具有生命周期
“由创建而产生，由调度而
执行，由撤销而消亡”

- 多个进程实体可同时存在于内存中**并发执行**
- 独立运行、独立分配资源和独立接受调度的**基本单位**
- 按**不可预知（异步）**的速度向前推进

进程是程序的一次运行过程!!!



本节小结



- 1 程序与进程的区别与联系
- 2 进程的调度状态及状态转换

阅读教材：30页 ~ 41页



E02：并发进程（进程基本概念）