南京大学
NANJING UNIVERSITY

计算机与操作系统

第四讲 进程管理

南京大学软件学院

15:58

南京大学
NANJING UNIVERSITY

本主题教学目标

1. 掌握进程的概念，可再入过程
2. 掌握进程的状态、进程的挂起，以及队列实现模型
3. 掌握操作系统的控制结构
4. 掌握进程描述与控制的数据结构
5. 掌握处理机模式的概念
6. 掌握进程创建、模式切换、进程切换、进程队列、进程原语等进程实现的原理
7. 了解操作系统的执行模型

南京大学
NANJING UNIVERSITY

第四讲 进程管理

4.1 什么是进程

4.2 进程的状态

4.3 进程的描述与管理

南京大學
NANJING UNIVERSITY

4.1 什么是进程

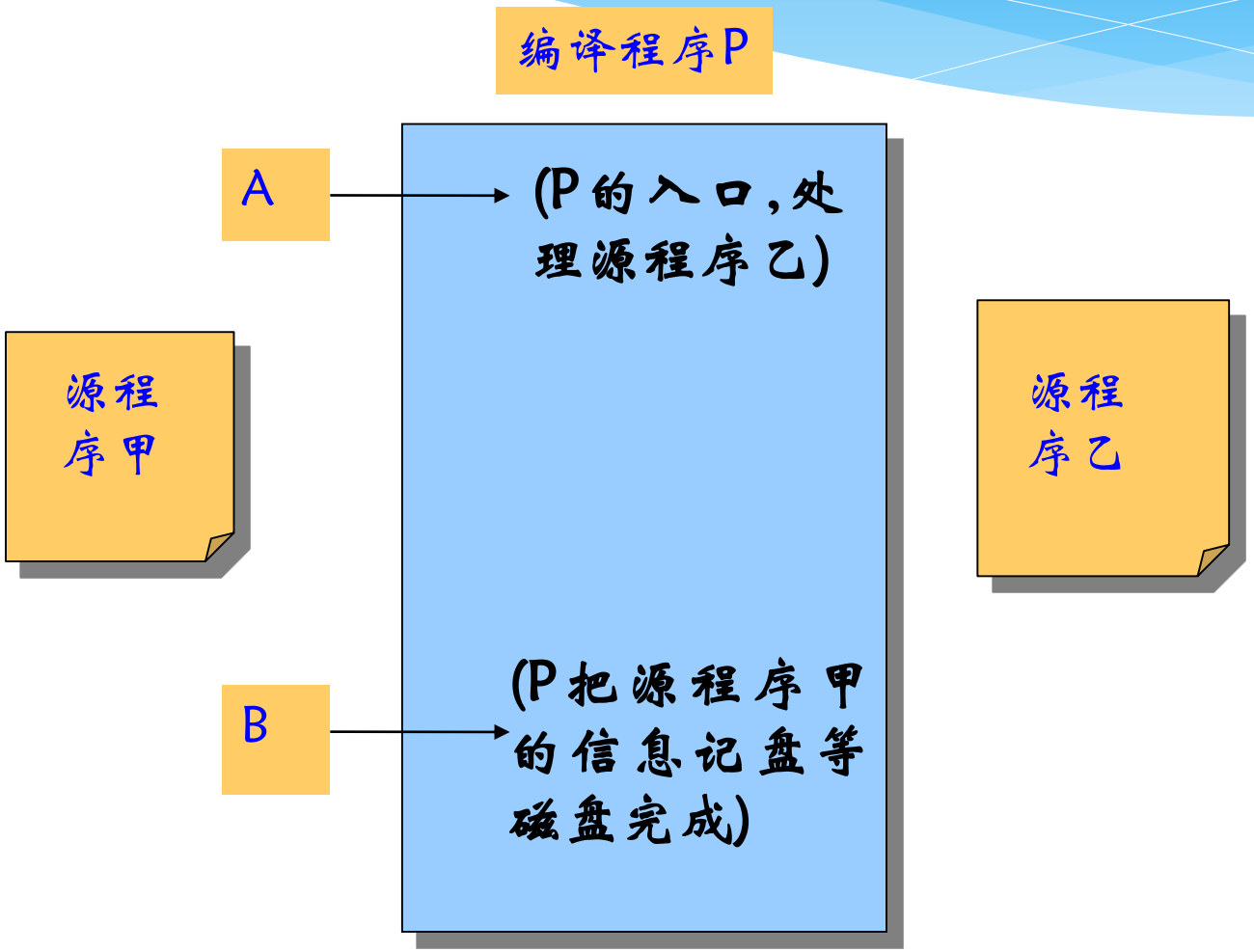
操作系统为什么要引入进程概念?(1)

* 原因1-刻画系统的动态性，发挥系统的并发性，提高资源利用率。
* 程序是并发执行的，即不是连续而是走走停停的。程序的并发执行引起资源共享和竞争问题，执行的程序不再处在封闭环境中。
* “程序”自身只是计算任务的指令和数据的描述，是静态概念无法刻画程序的并发特性，系统需要寻找一个能描述程序动态执行过程的概念，这就是进程。

操作系统为什么要引入进程概念?(2)

* 原因2-它能解决系统的“共享性”，正确描述程序的执行状态。
* “可再用”程序
* “可再入”程序
* “可再入”程序具有的性质

“可再入”程序举例



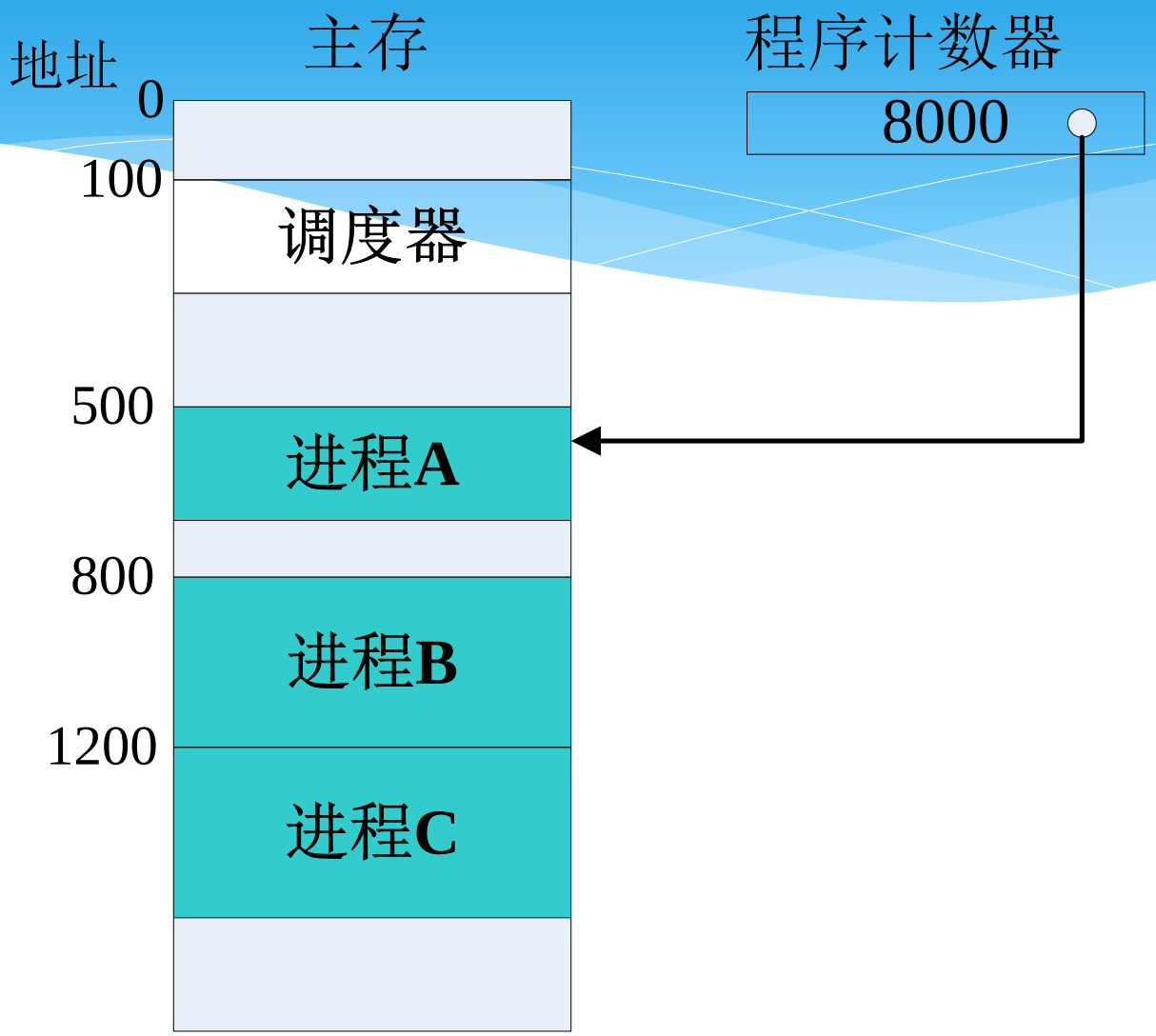
南京大学
NANJING UNIVERSITY

进程的定义

* 进程是一个具有一定独立功能的程序关于某个数据集合的一次运行活动
* A process is a program in execution, including:
  * Process id and Data structure for process management and execution
  * Code: program in memory
  * Data: data in memory and its values
  * Register values
  * Program status word

进程的定义

- **$Process_i$: (P_i, C_i, D_i, R_i, Psw_i)**
  - **Program(data) <> Program(data) in memory**
- **(P_1, C_1, D_1, R_1, Psw_1) and (P_2, C_2, D_2, R_2, Psw_2)**
  - **C_1 and C_2(different program) , but D_1 and D_2(different data)**
  - **C_1 and C_2(different program) , but D_1 and D_2(same data)**
  - **C_1 and C_2(same program), but D_1 and D_2(different data)**
  - **C_1 and C_2(same program), but D_1 and D_2(same data)**
- **(P_1, C_1, D_1, R_1, Psw_1) and (P_2, C_2, D_1, R_2, Psw_2)**
  - **Concurrent programming(in fact, only share some data)**
- **(P_1, C_1, D_1, R_1, Psw_1) and (P_2, C_1, D_2, R_2, Psw_2)**
  - **Share code(Reentrant program, Reentrant Procedure)**


地址
主存
程序计数器
8000
0
100
调度器
500
进程A
800
进程B
1200
进程C

5000	8000	12000
5001	8001	12001
5002	8002	12002
5003	8003	12003
5004		12004
5005		12005
5006		12006
5007		12007
5008		12008
5009		12009
5010		12010
5011		12011
(a) 进程A的轨迹	(b) 进程B的轨迹	(c) 进程C的轨迹

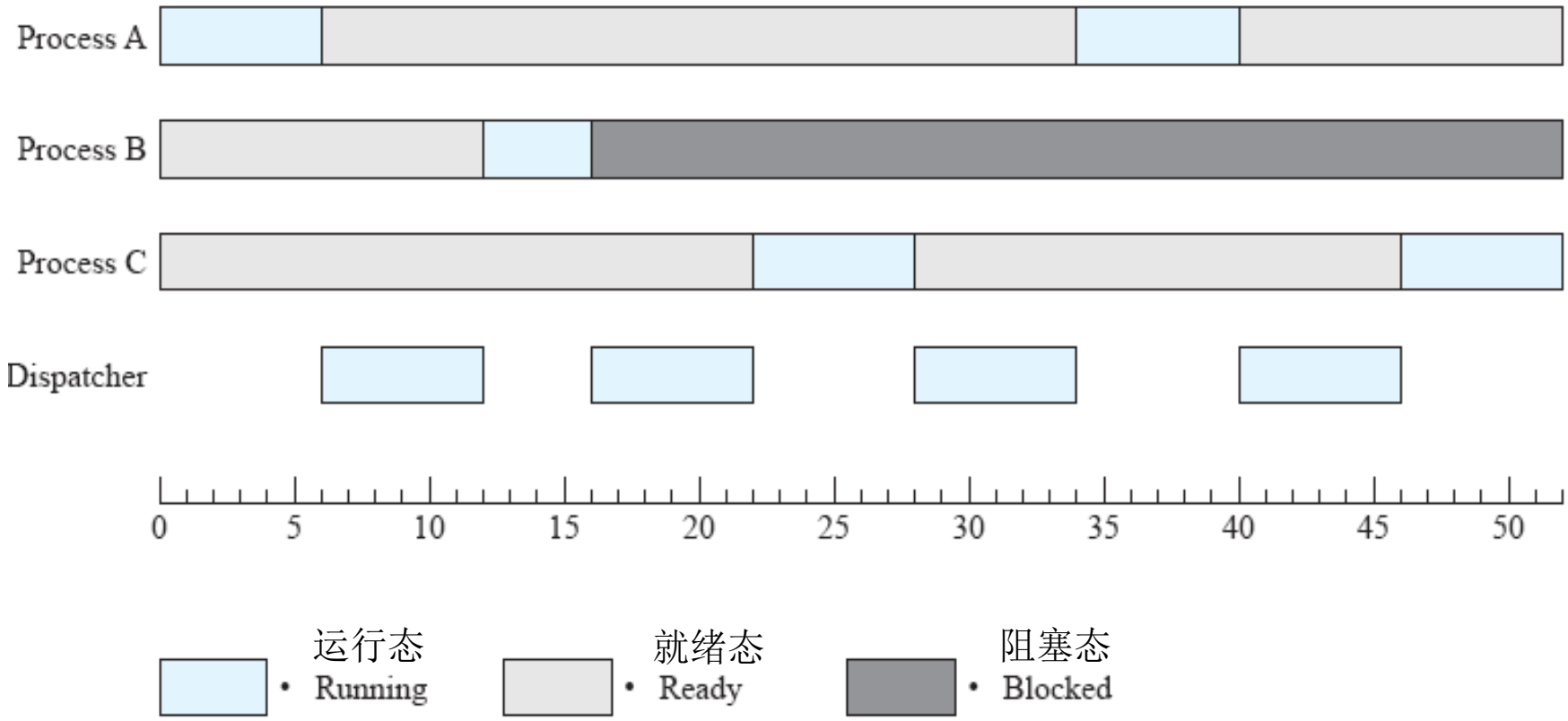
5000表示进程A的程序起始地址

8000表示进程B的程序起始地址

12000表示进程C的程序起始地址

- 100表示调度器程序的起始地址
- 阴影表示调度器程序的执行
- 第1列和第3列是指令周期计数
- 第2列和第4列给出了被执行的指令地址

1	5000	
2	5001	
3	5002	进程A
4	5003	
5	5004	
6	5005	超时
7	100	
8	101	
9	102	
10	103	
11	104	
12	105	
13	8000	
14	8001	进程B
15	8002	
16	8003	I/O请求
17	100	
18	101	
19	102	
20	103	
21	104	
22	105	
23	12000	
24	12001	
25	12002	进程C
26	12003	
27	12005	进程C
28	12006	超时
29	100	
30	101	
31	102	
32	103	
33	104	
34	105	
35	5006	
36	5007	
37	5008	进程A
38	5009	
39	5010	
40	5011	超时
41	100	
42	101	
43	102	
44	103	
45	104	
46	105	
47	12006	
48	12007	
49	12008	进程C
50	12009	
51	12010	
52	12011	超时


Process A
Process B
Process C
Dispatcher
0
5
10
15
20
25
30
35
40
45
50
运行态
• Running
就绪态
• Ready
阻塞态
• Blocked

4.2 进程的状态

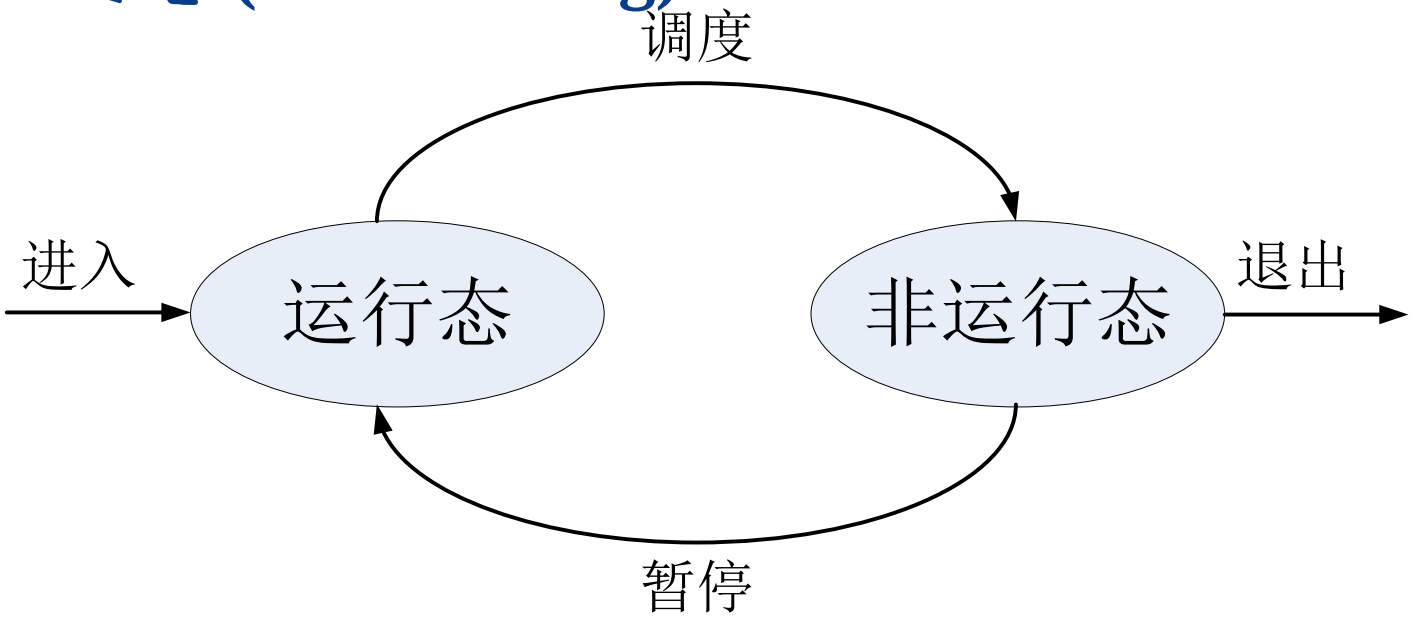
南京大学
NANJING UNIVERSITY

4.2 进程的状态

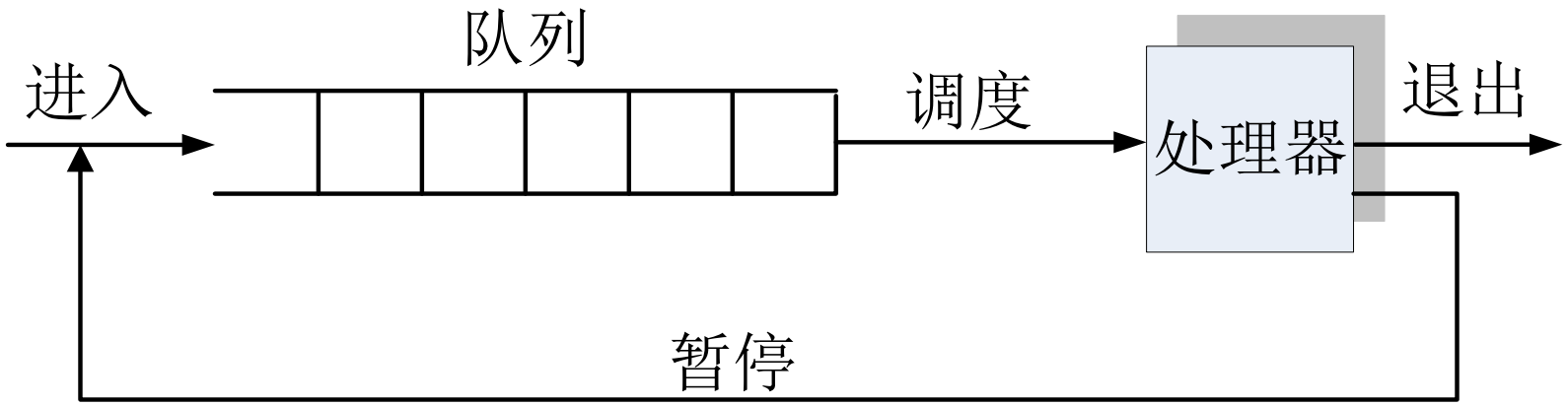
* 4.2.1 两状态模型
* 4.2.2 三状态模型
* 4.2.3 五状态模型

4.2.1 两状态模型

* 两个状态
  * 运行态 (running)
  * 非运行态 (Not-running)



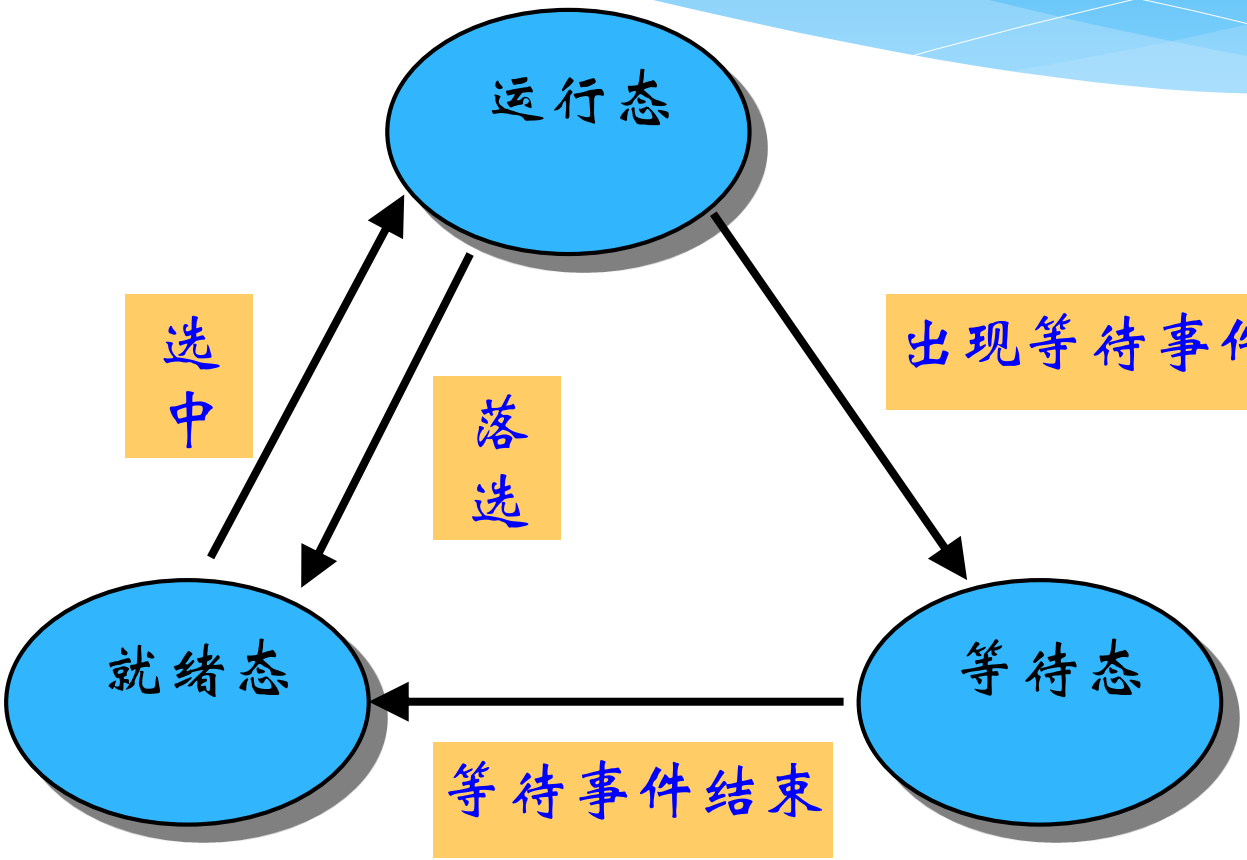
两状态模型的队列图



4.2.2 三状态模型

* 就绪态 (Ready)
  * 准备执行
* 阻塞态 (Blocked)，又叫或者等待态 (Waiting)
  * 在等待I/O
* 调度器只能选择就绪态进程进行调度，不能选择阻塞态进程

三状态模型及其转换



三状态模型

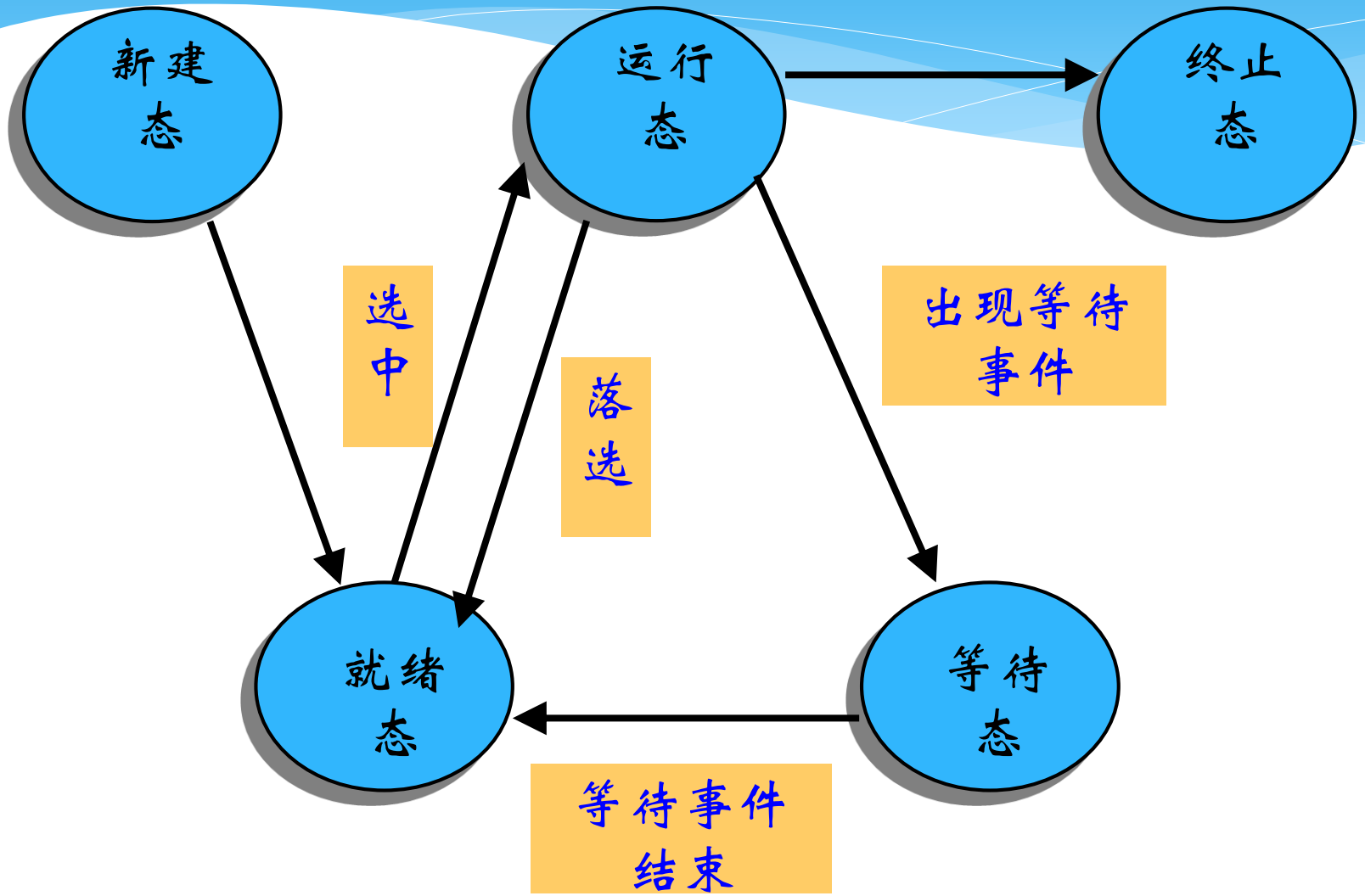
* 运行态→等待态：等待使用资源；等待外设传输；等待人工干预/信号
* 等待态→就绪态：资源得到满足；外设传输结束；人工干预/信号完成
* 运行态→就绪态：运行时间到；出现有更高优先权进程
* 就绪态→运行态：CPU空闲时选择一个就绪进程

南京大学
NANJING UNIVERSITY

4.2.3 五状态模型

* 运行态 (Running)
* 就绪态 (Ready)
* 阻塞态 (Blocked)
* 新建 (New)
* 退出 (Exit)

五状态模型及其转换

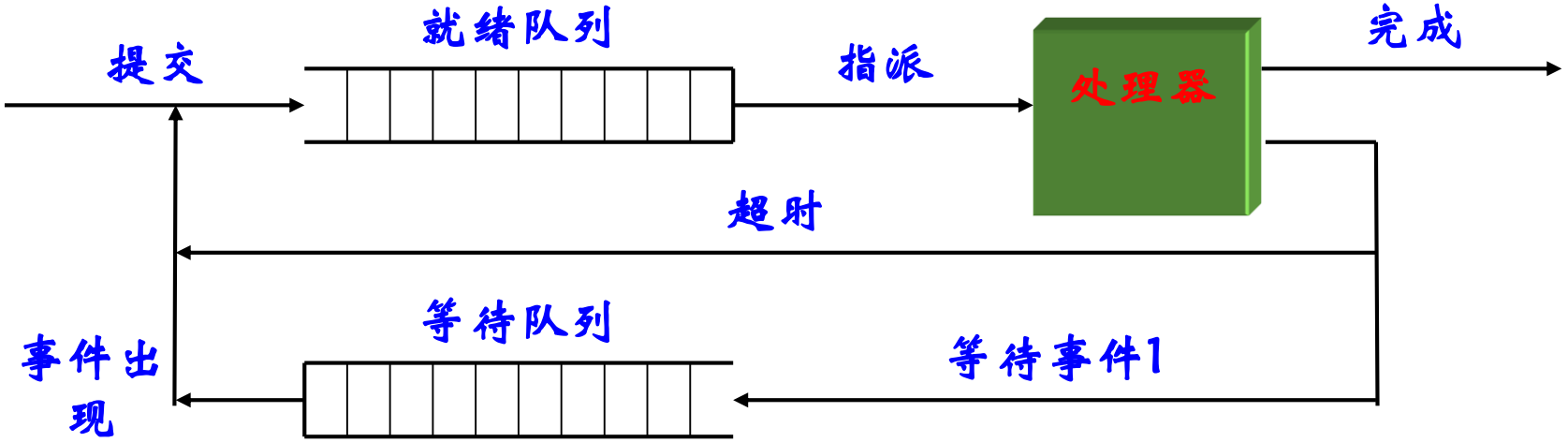


南京大学
NANJING UNIVERSITY

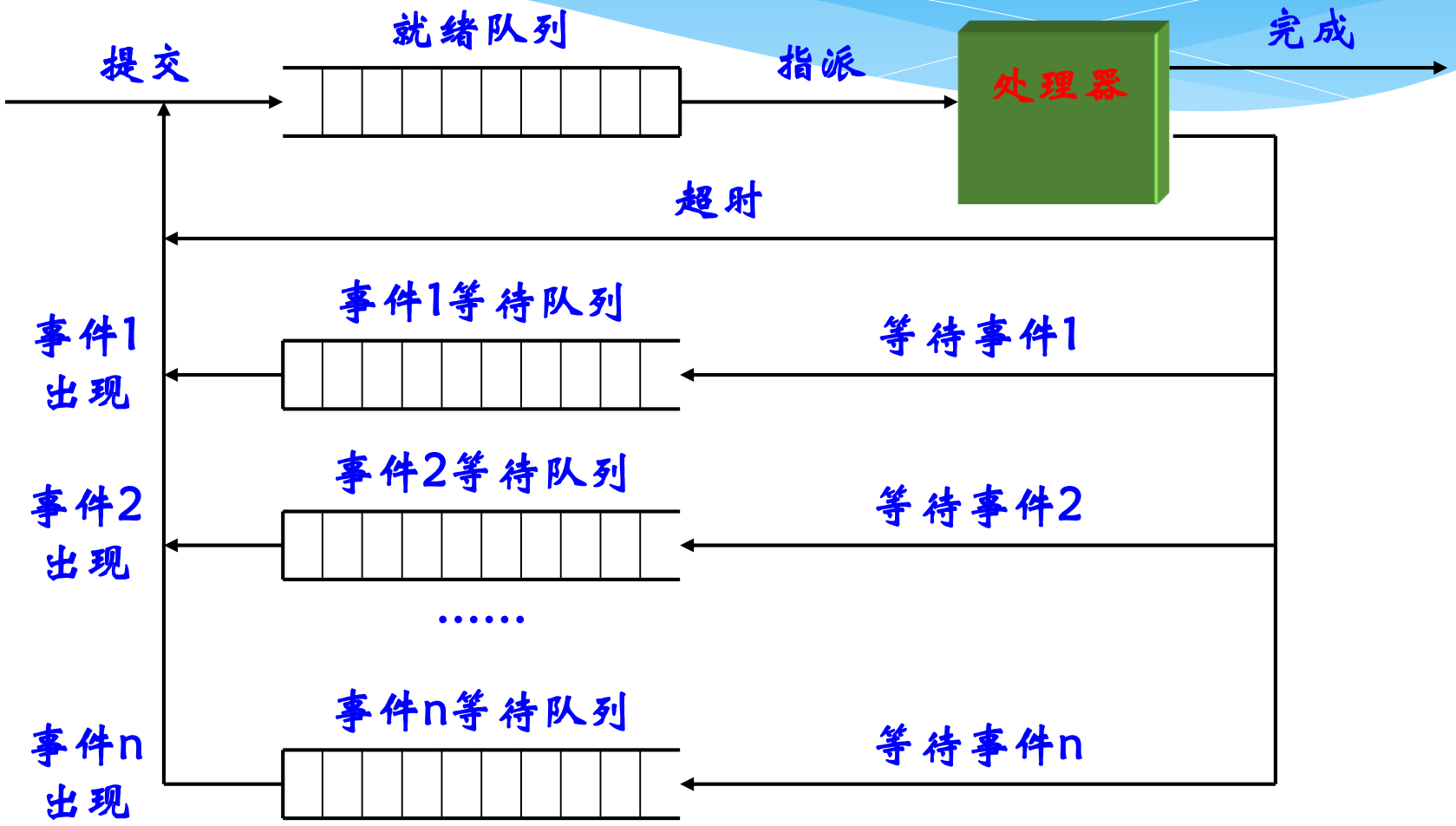
五状态模型

* NULL→新建态：执行一个程序/创建一个子进程
* 新建态→就绪态：当OS完成了进程创建的必要操作，且当前系统的性能和虚拟内存的容量均允许
* 运行态→终止态：当一个进程到达了自然结束点，或出现了无法克服的错误，或被操作系统所终结，或被其他有终止权的进程所终结
* 终止态→NULL：完成善后操作
* 就绪态→终止态：未在状态转换图中显示，但某些操作系统允许父进程终结子进程
* 等待态→终止态：未在状态转换图中显示，但某些操作系统允许父进程终结子进程

队列管理和状态转换示意图
使用两个队列



队列管理和状态转换示意图
使用多个队列



4.3 进程的描述与管理

南京大学
NANJING UNIVERSITY

4.3 进程的描述与管理

4.3.1 进程与资源

4.3.2 操作系统的控制结构

4.3.3 进程控制结构

4.3.4 执行模式

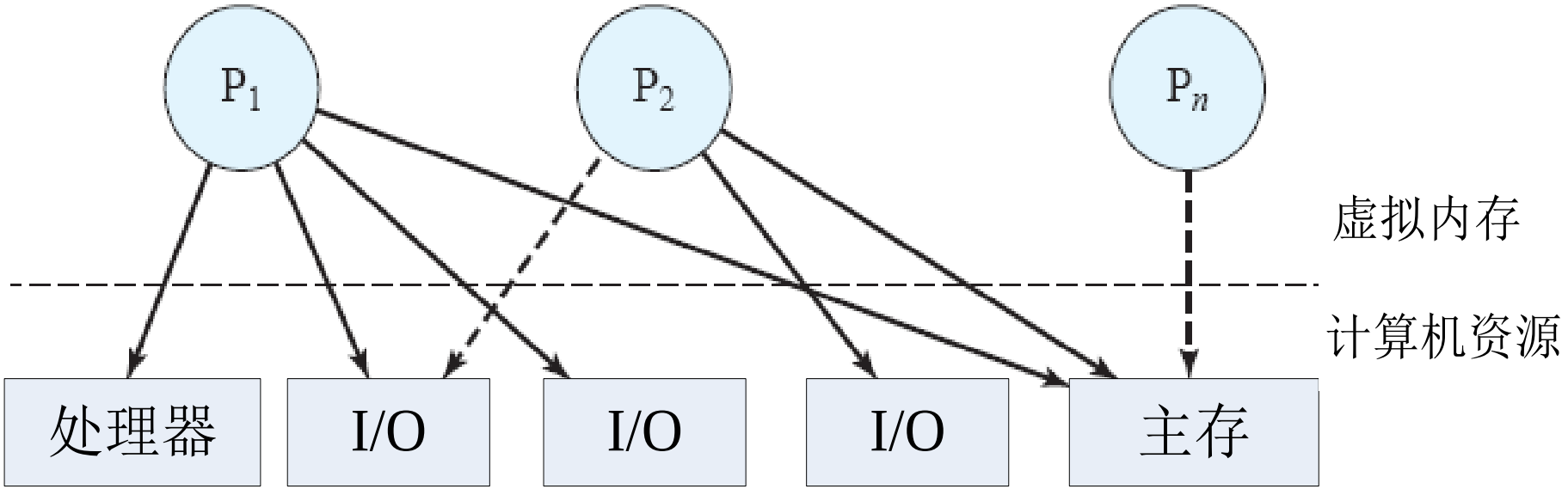
4.3.5 进程创建

4.3.6 进程切换

4.3.7 OS的执行方式

进程描述

进程和资源



南京大学
NANJING UNIVERSITY

4.3.2 操作系统的控制结构

* 必须掌握关于每个进程和资源当前状态的信息
* 操作系统构造并维护它所管理的每个实体的信息表

内存表

* 分配给进程的主存
* 分配给进程的辅存
* 主存块或虚拟内存块的任何保护属性，如哪些进程可以访问某些共享内存区域
* 管理虚拟内存所需要的任何信息

I/O 表

* I/O 是可用或者是已经分配
* I/O操作的状态
* 作为I/O传送的源和目标的主存单元

南京大学
NANJING UNIVERSITY

文件表

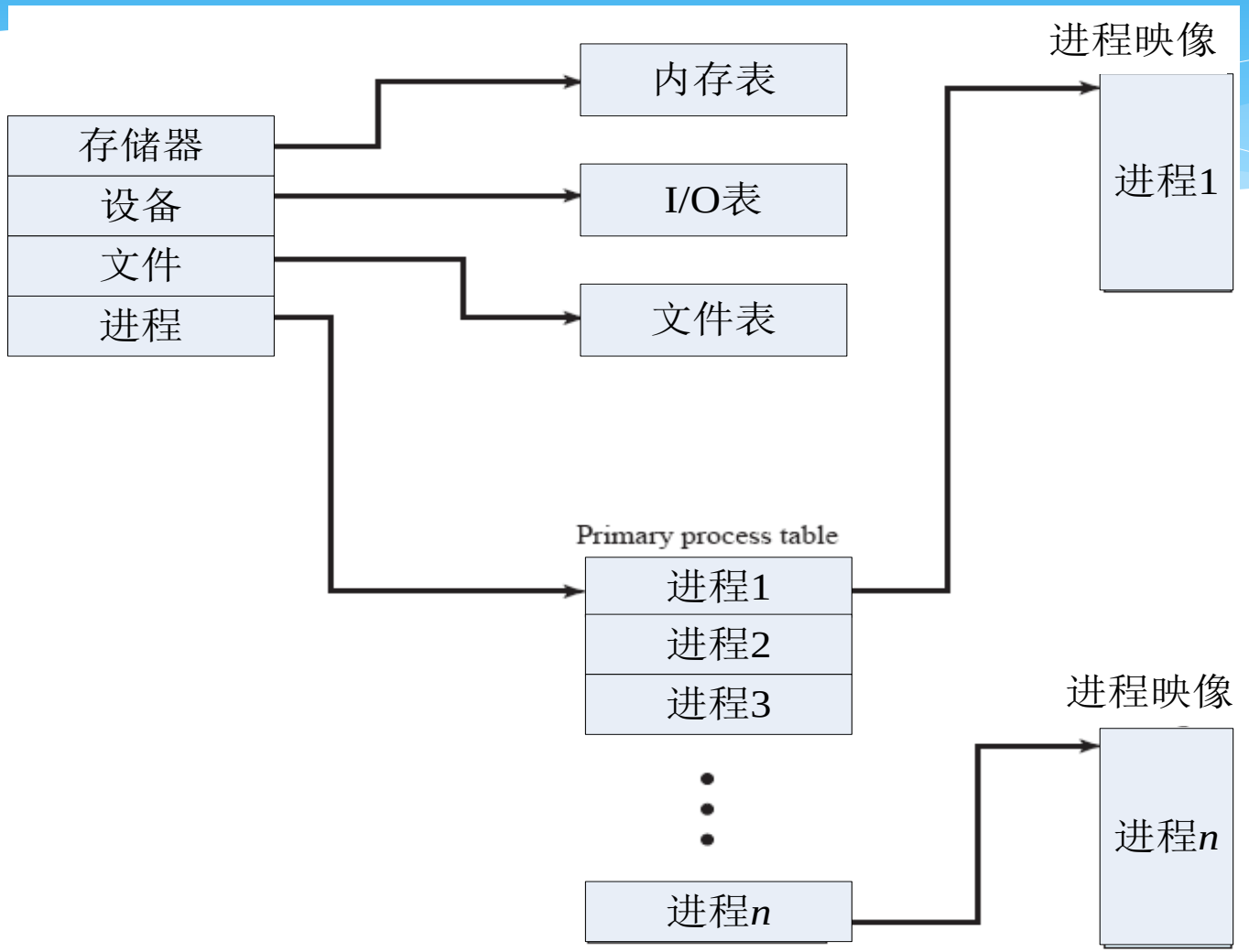
* 文件是否存在
* 文件在辅存中的位置
* 当前状态
* 属性
* 大部分信息可能由文件管理系统维护和使用

南京大学
NANJING UNIVERSITY

进程表

* 进程的位置
* 为管理而需要的属性
  * 进程 ID
  * 进程的状态state
  * 进程在内存中的位置

4.3.2 操作系统的控制结构



南京大学
NANJING UNIVERSITY

4.3.3 进程控制结构

* 进程映像
* 进程属性
* 处理器状态信息

南京大学
NANJING UNIVERSITY

进程映像

* 进程包括程序集合和数据
  * 程序：将被执行的程序
  * 用户数据：用户空间的可修改部分，可以包括程序数据、用户栈和可修改的程序。
  * 系统栈：每个进程有一个或多个后进先出的系统栈，栈用于保存参数、过程调用地址和系统
  * 进程控制块：操作系统控制进程所需要的数据
* 进程映像
  * 程序、数据、栈和属性的集合称为进程映像

南京大学
NANJING UNIVERSITY

进程控制块 (Process Control Block, PCB)

* 进程标识
  * 标识号
    * 此进程的标识号
    * 创建此进程的进程 (父进程) 的标识号
    * 用户标识号(User ID)

进程控制块

* 处理器状态信息
  * 用户可见寄存器
    * 用户可见寄存器是处于用户模式的处理器执行的机器语言可以访问的寄存器。通常有8到32个此类寄存器，而在一些RISC实现中有超过100个此类寄存器

南京大学
NANJING UNIVERSITY

进程控制块

* 处理器状态信息
  * 控制和状态寄存器，用于控制寄存器操作的各种处理器寄存器，包括：
    * 程序计数器：包含将要取的下一条指令的地址
    * 条件码：最近的算术或逻辑运算的结果（例如符号、零、进位、溢出）
    * 状态信息：包括中断允许/禁止标志，异常模式

南京大学
NANJING UNIVERSITY

进程控制块

* 处理器状态信息
  * 栈指针
    * 每个进程有一个或多个与之相关联的后进先出LIFO系统栈，栈用于保存参数和过程调用或系统调用的地址，栈指针指向栈顶

南京大学
NANJING UNIVERSITY

进程控制块

* 进程控制信息
  * 调度和状态信息，即操作系统执行其调度功能所需要的信息，典型的信息项包括：
    * 进程状态；
    * 优先级；
    * 调度相关信息：这取决于所使用的调度算法。例如进程等待地时间总量和进程在上一次运行时执行时间总量。
    * 事件：进程在继续执行前等待地事件标识

南京大学
NANJING UNIVERSITY

进程控制块

* 进程控制信息
  * 数据结构
    * 进程可以以队列、环或某些别的结构形式与其他进程进行链接。例如，所有具有某一特定优先级且处于等待状态的进程可链接在一个队列中；进程还可以表示出与另一个进程的父子(创建-被创建)关系。进程控制块为支持这些结构需要包括指向其他进程的指针

进程控制块

* 进程控制信息
  * 进程通信 (InterProcess Communication, IPC)
    * 与两个独立进程间的通信相关联的有各种标记、信号和消息。进程控制块中维护着某些或全部此类信息。

进程控制块

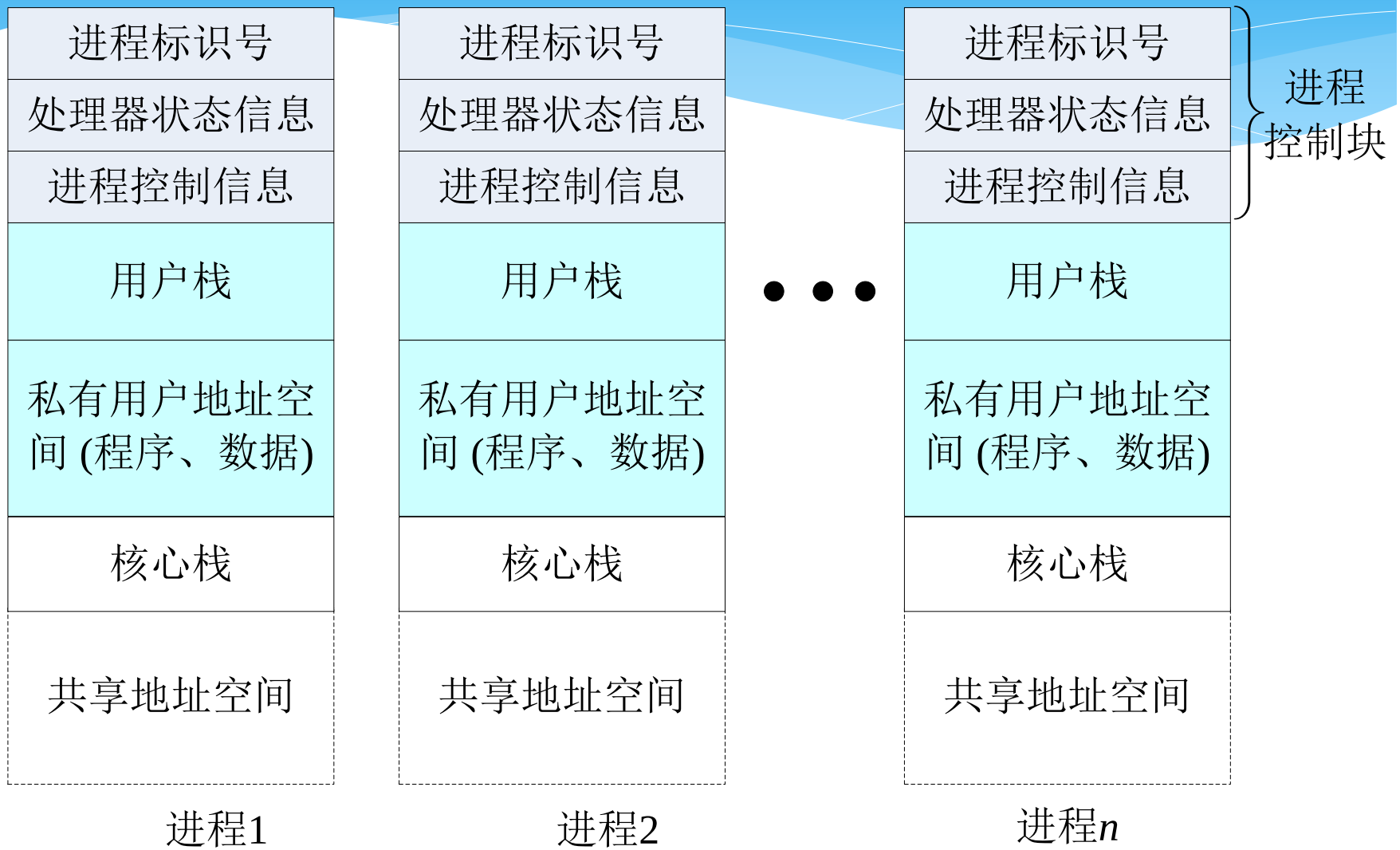
* 进程控制信息
  * 进程特权
    * 进程根据其可以访问的内存空间以及可以执行的指令类型被赋予各种特权。此外，特权还用于系统实现程序和服务地使用。

进程控制块

* 进程控制信息
  * 存储管理
    * 这一部分包括指向描述分配给该进程的虚拟内存空间地段表或页表地指针。
  * 资源地所有权和使用情况
    * 进程控制的资源可以表示成诸如一个打开的文件，还可能包括处理器或其他资源的使用历史；调度器会需要这些信息。

南京大学
NANJING UNIVERSITY

虚拟内存中的用户进程



南京大学 NANJING UNIVERSITY

4.3.4 执行模式

处理器状态信息

* 处理器寄存器
  * 用户可见寄存器
  * 控制和状态寄存器
  * 栈指针
* 程序状态字
  * 包括状态信息

南京大学
NANJING UNIVERSITY

进程上下文

* 用户级上下文：进程正文，进程数据，用户栈，共享内存区
* 寄存器上下文：PSW，栈指针，通用寄存器
* 系统级上下文：PCB，内存区表，内核栈

执行模式

* 用户模式
  * 非特权模式
  * 用户程序通常在该模式下运行
* 系统模式、控制模式或内核模式
  * 特权模式
  * 内核模式指的是操作系统的内核，这是操作系统中包含重要系统功能的部分

南京大学
NANJING UNIVERSITY

操作系统内核的典型功能

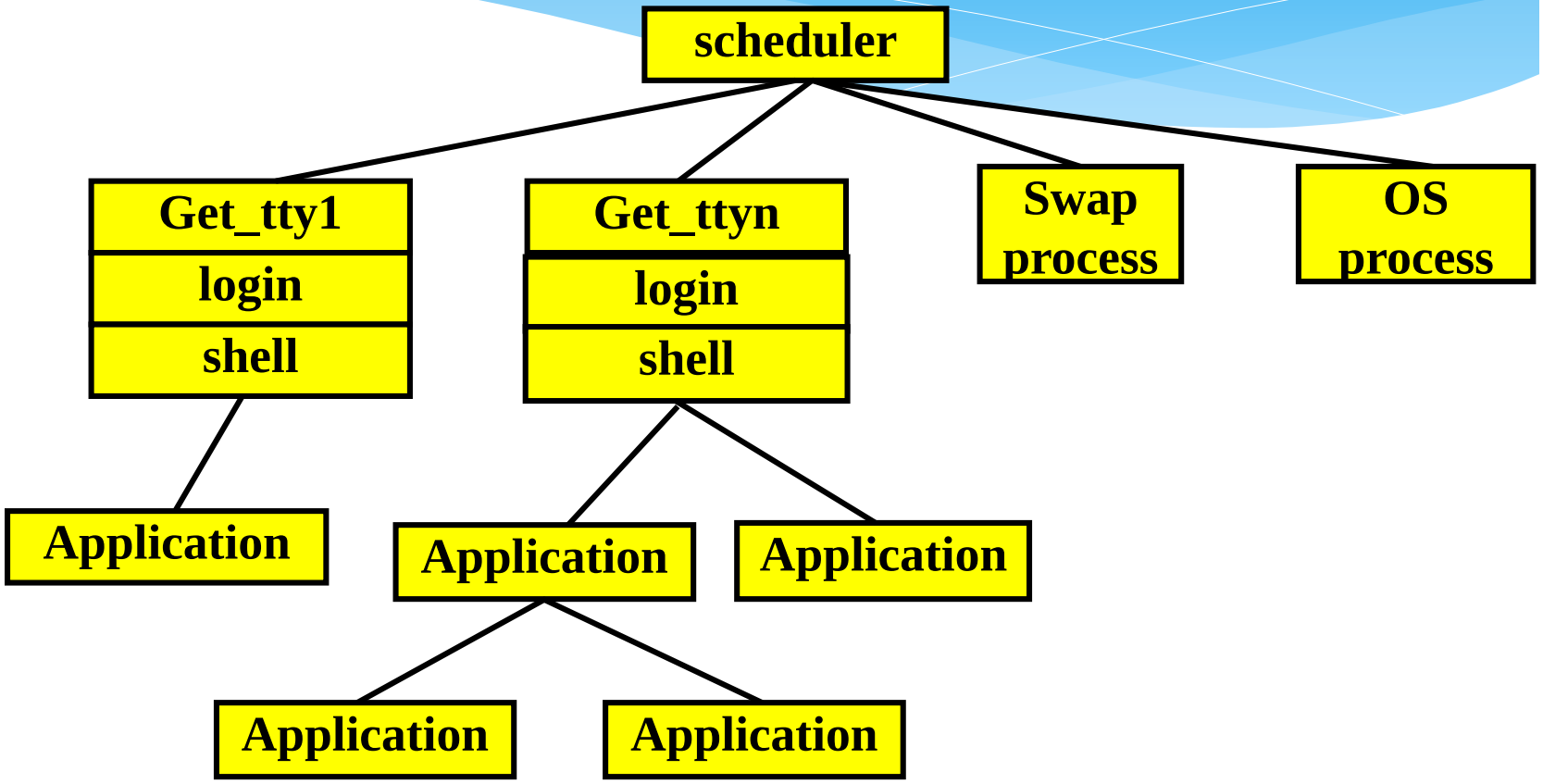
* 进程管理
  * 进程的创建和终止
  * 进程的调度和分析
  * 进程切换
  * 进程同步以及对进程间通信的支持
  * 进程控制块的管理
* 内存管理
  * 给进程分配地址空间
  * 交换
  * 页和段管理
* I/O管理
  * 缓冲区管理
  * 给进程分配I/O通道和设备
* 支持功能
  * 中断处理
  * 审计
  * 监视

南京大学
NANJING UNIVERSITY

4.3.5 进程创建

* 给新进程分配一个唯一的进程标识号
* 给进程分配空间。这包括进程映像中的所有元素
* 初始化进程控制块
* 设置正确的连接
  * 例如:新进程必须放置在就绪或就绪/挂起链表中
* 创建或扩充其他数据结构
  * 例如:操作系统可能为每个进程保存着一个审计文件，可用于编制帐单和/或进行性能评估

进程创建


scheduler
Get_tty1
login
shell
Get_ttyn
login
shell
Swap process
OS process
Application
Application
Application
Application
Application

南京大学
NANJING UNIVERSITY

4.3.6 进程撤销

* 通过进程标识在某个队列中找到该进程的PCB
* 去配(释放) 该进程的所有资源
* 撤销该进程的子进程
* 将PCB返还给PCB池

南京大学
NANJING UNIVERSITY

4.3.6 进程切换

* 时钟中断
  * 操作系统确定当前正在运行的进程的执行时间是否已经超过了最大允许时间段
* I/O中断
* 内存失效
  * 处理器访问一虚拟内存地址，且此地址单元不在主存中时，操作系统必须从辅存中把包含这个地址单元的内存块(页或段)调入主存中
* 陷阱
  * 确定错误或异常条件是致命的
  * 当前正在运行的进程被转换到退出态
* 系统调用
  * 例如打开某个文件

模式切换

* 保存当前进程的现场
* 把程序计数器置成中断处理程序的开始地址
* 把处理器模式从用户模式切换到内核模式，使得中断处理代码可以执行有特权的指令

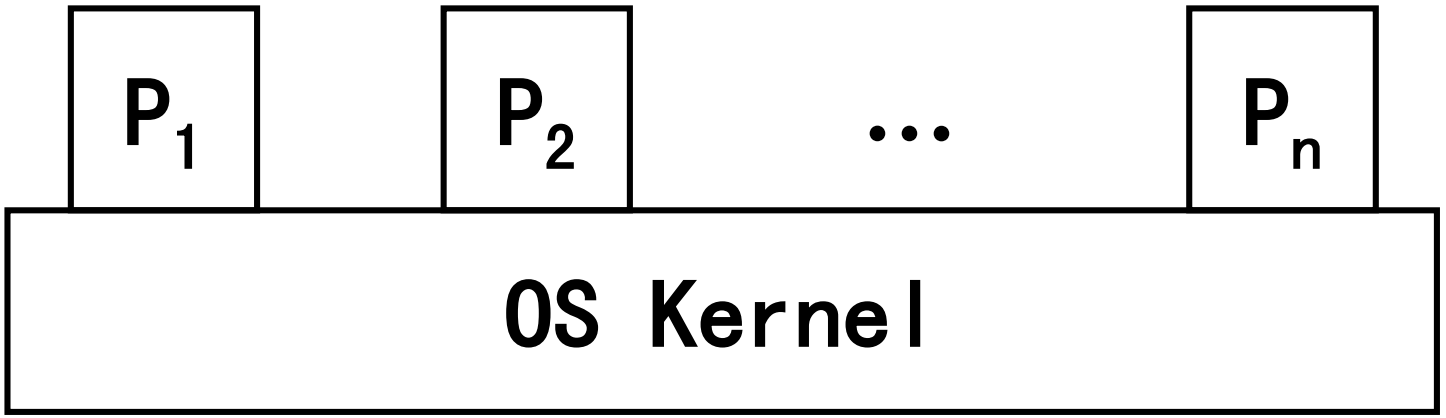
南京大学
NANJING UNIVERSITY

进程状态的变化

* 保存处理器上下文，包括程序计数器和其他寄存器
* 更新当前处于运行态的进程的进程控制块
* 还必须更新其他相关域，包括离开运行态的原因和审计信息
* 把进程的进程控制块移到相应的队列(就绪、在事件i处阻塞、就绪/挂起)
* 选择另一个进程执行 (即调度)
* 更新所选择进程的进程控制块
* 更新内存管理的数据结构
* 恢复处理器在被选择的进程最近切换出运行态时的上下文

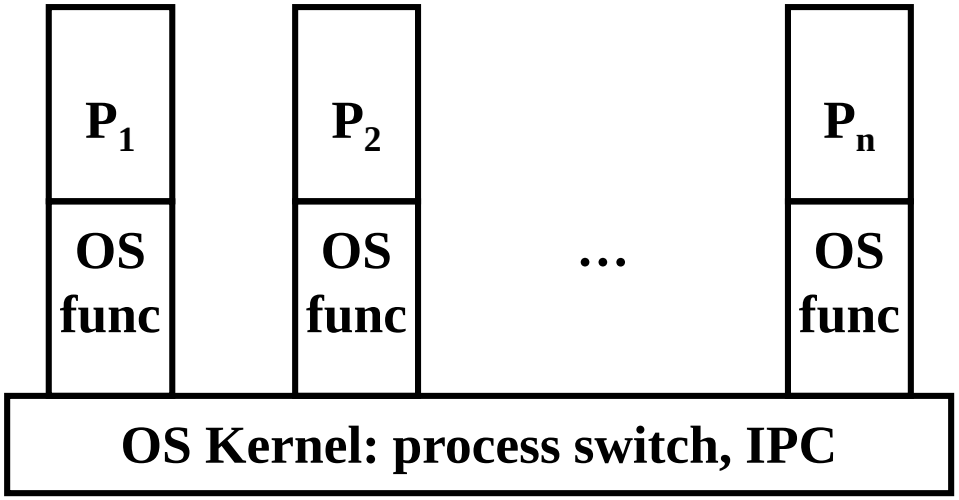
4.3.7 OS的执行方式

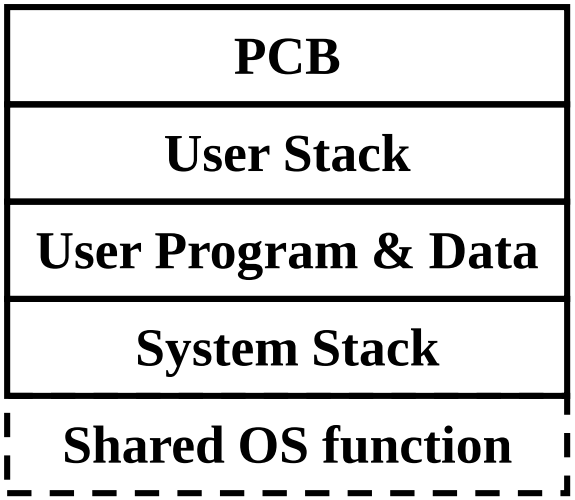
* 无进程的内核
  * 非常传统和通用的一种方法是在所有的进程之外执行操作系统内核
  * 操作系统代码作为一个在特权模式下工作的独立实体被执行



OS的执行方式

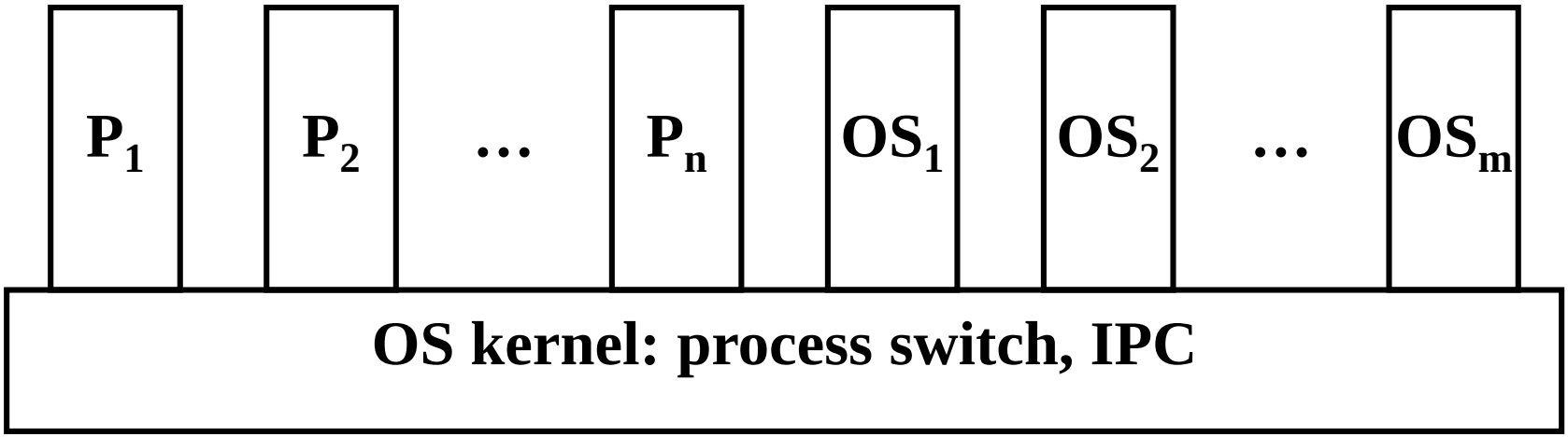
* 在用户进程中执行
  * 在用户进程的上下文中执行所有操作系统软件，其观点是操作系统从根本上说是用户调用的一组例程，在用户进程环境中执行，用于实现各种功能
  * 当发生一个中断、陷阱或系统调用时，处理器被置于内核模式，控制权转交给操作系统





OS的执行方式

* 基于进程的操作系统
  * 把操作系统作为一组进程实现
  * 在多处理器或多机环境中都是十分有用的，这时一些操作系统服务可以传送到专用处理器中执行，以提高性能



本主题小结

1. 掌握进程的概念，可再入过程
2. 掌握进程的状态、进程的挂起，以及队列实现模型
3. 掌握操作系统的控制结构
4. 掌握进程描述与控制的数据结构
5. 掌握处理机模式的概念
6. 掌握进程创建、模式切换、进程切换、进程队列、进程原语等进程实现的原理
7. 了解操作系统的执行模型