



华中农业大学
HUAZHONG AGRICULTURAL UNIVERSITY

计算机系统

倪福川

fcni_cn@mail.hzau.edu.cn

华中农业大学 信息学院





第二章 进程的描述与控制

2.1 前趋图和程序执行

2.2 进程的描述

2.3 进程控制

2.4 进程同步

2.5 经典进程的同步问题

2.6 进程通信

2.7 线程的基本概念

2.8 线程的实现





2.1 前趋图和程序执行

未配置OS的系统和单道批处理

顺序执行，仅装入一道程序且独占所有资源

缺点：浪费资源、系统运行效率低等

多道批处理系统

同时装入多道程序，共享系统资源，并发执行。



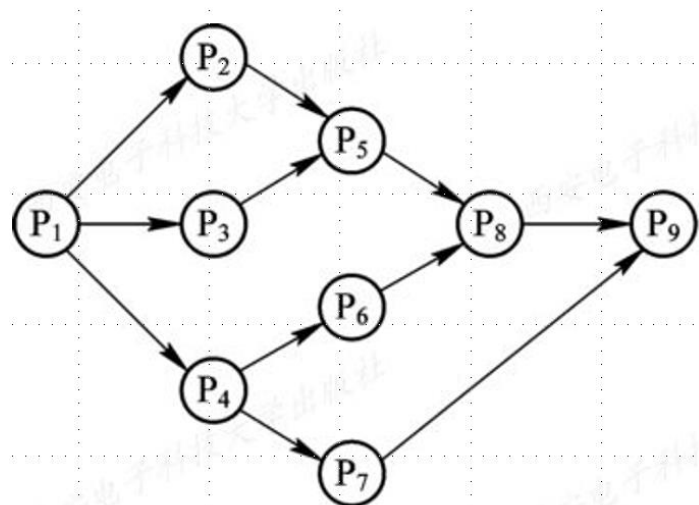


2.1.1 前趋图

前趋图：有向无循环图，描述进程间执行的先后顺序。

结点：一个进程或程序段，乃至一条语句；

有向边：两个结点之间存在的偏序或前趋关系



(a) 具有九个结点的前趋图





进程间的**前趋关系**表示

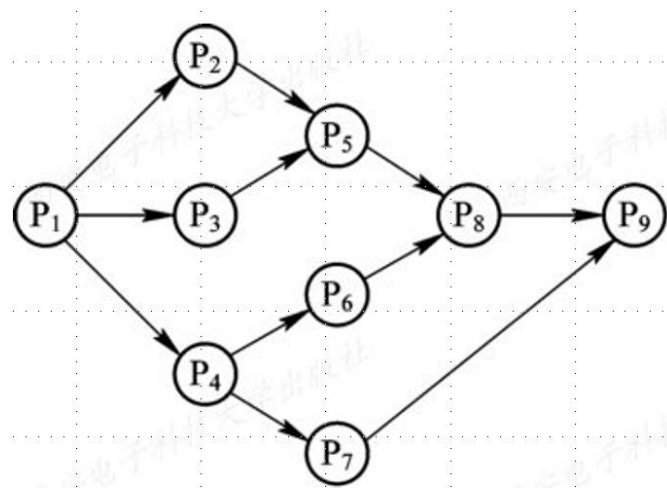
如果 P_j 开始执行之前 P_i 必须完成，则进程 P_i 和 P_j 存在前趋关系： $(P_i, P_j) \in \rightarrow$ ，或 $P_i \rightarrow P_j$ 。

P_i 是 P_j 的**直接前趋**， P_j 是 P_i 的**直接后继**。

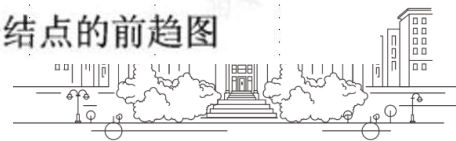
初始结点：没有前趋的结点；

终止结点：没有后继的结点；

结点的权：该结点所含有的程序量或程序的执行时间。

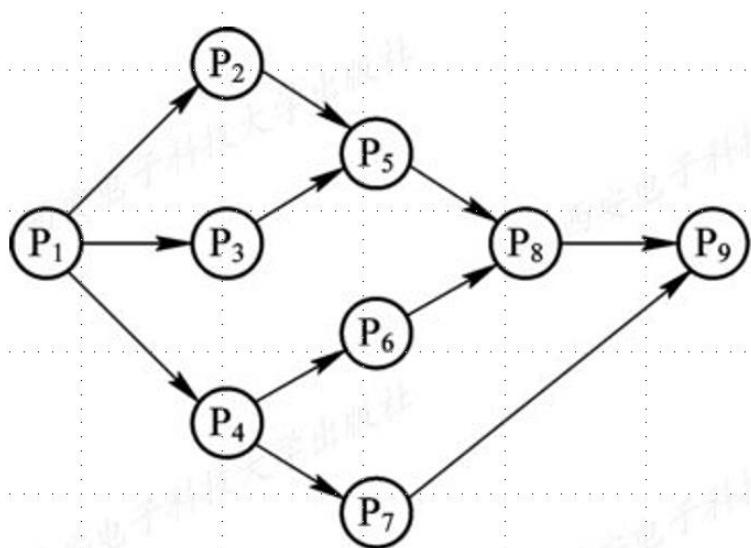


(a) 具有九个结点的前趋图





进程之间的前趋关系表示



(a) 具有九个结点的前趋图

图中存在前趋关系：

$P_1 \rightarrow P_2, P_1 \rightarrow P_3, P_1 \rightarrow P_4, P_2 \rightarrow P_5,$
 $P_3 \rightarrow P_5, P_4 \rightarrow P_6, P_4 \rightarrow P_7, P_5 \rightarrow P_8,$
 $P_6 \rightarrow P_8, P_7 \rightarrow P_9, P_8 \rightarrow P_9$

或： $P = \{V, E\}$

$V = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9\}$

$E = \{(P_1, P_2), (P_1, P_3), (P_1, P_4), (P_2, P_5), (P_3, P_5), (P_4, P_6), (P_4, P_7), (P_5, P_8), (P_6, P_8), (P_7, P_9), (P_8, P_9)\}$

前趋图中是不允许有循环的，否则必然会产生不可能实现的前趋关系

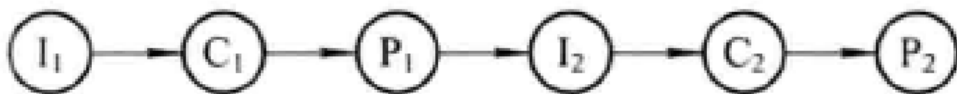




2.1.2 程序顺序执行

1. 程序的顺序执行

程序的若干个程序段，先后次序顺序执行



(a) 程序的顺序执行

前趋关系:

$$I_i \rightarrow C_i \rightarrow P_i$$

其中I代表输入操作，C代表计算操作，P为打印操作，用箭头指示操作的先后次序。





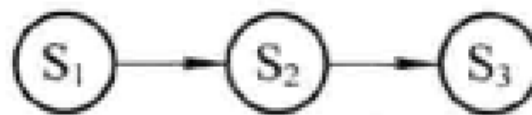
程序段也存在执行顺序问题。

包含三条语句的程序段：

$S_1: a := x + y;$

$S_2: b := a - 5;$

$S_3: c := b + 1;$



(b) 三条语句的顺序执行

前趋关系： $S_1 \rightarrow S_2 \rightarrow S_3$

语句 S_2 必须在语句 S_1 后(a 被赋值)才能执行, 语句 S_3 也只能在 b 被赋值后才能执行。





2. 程序顺序执行时的特征

① 顺序性：

严格地按规定顺序执行

② 封闭性：

封闭环境，独占资源，只有本程序能改变资源状态（除初始状态外），执行结果不受外界因素影响；

③ 可再现性：

执行环境和初始条件相同，重复执行获得相同结果。

方便检测和校正错误

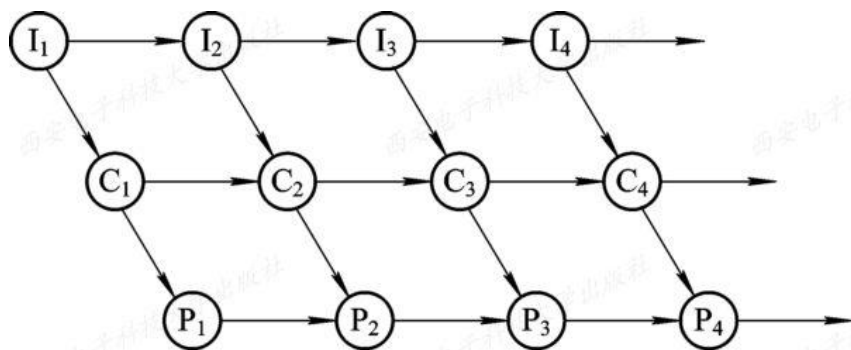




2.1.3 程序并发执行

1. 程序的并发执行

多道程序并发执行



作业的 I_i , C_i , P_i 三个程序段必须顺序执行

前趋关系 $I_i \rightarrow C_i$, $I_i \rightarrow I_{i+1}$,
 $C_i \rightarrow P_i$, $C_i \rightarrow C_{i+1}$, $P_i \rightarrow P_{i+1}$

I_{i+1} 和 C_i 及 P_{i-1} 是重叠的:

在 P_{i-1} 和 C_i 以及 I_{i+1} 之间, 不存在前趋关系, 才可以并发执行





华中农业大学

HUAZHONG AGRICULTURAL UNIVERSITY

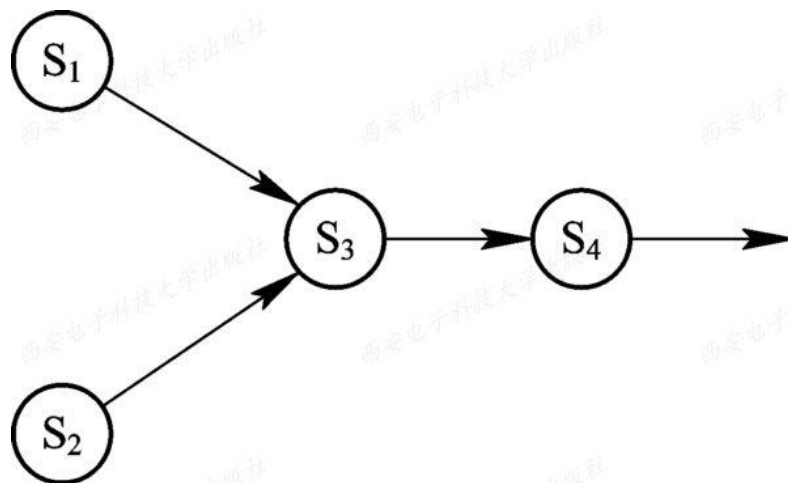
对具有四条语句的程序段：

$S_1: a := x + 2$

$S_2: b := y + 4$

$S_3: c := a + b$

$S_4: d := c + b$



S_4 必须在 S_3 之后执行；

但 S_1 和 S_2 可并发执行，彼此互不依赖





2. 程序并发执行时的特征

并发程序形成相互制约的关系

{ 程序并发，共享资源
相互合作完成同一任务

- (1) 间断性。相互制约导致执行间断
- (2) 失去封闭性。资源状态受多程序影响
- (3) 不可再现性。相同环境和初始条件，
重复执行结果不同





程序A

L1:

N:=N+1;

goto L1

程序B

L2:

PRINT (N) ;

N:=0;

goto L2

设共享变量N初值为5，并发执行可能的结果：

6, 6, 0

5, 0, 1

5, 6, 0





2.2 进程的描述

2.2.1 进程的定义和特征

1. 进程的定义

进程控制块PCB，描述进程的基本情况和活动过程的数据结构。

进程实体，由程序段、数据段
和PCB构成。
CPU 寄存器





典型的进程定义：

(1) 进程是程序的一次执行。

(2) 进程是一个程序及其数据在处理机上顺序执行时所发生的活动。

(3) 进程是具有独立功能的程序在一个数据集合上运行的过程，它是系统进行资源分配和调度的一个独立单位。



南京大学
NANJING UNIVERSITY



骆斌

学堂在线

<https://next.xuetangx.com/course/NJU08091000228/15157>

中国大学MOOC

<https://www.icourse163.org/course/NJU-1001571004>





进程定义:

进程是进程实体的运行过程，是系统进行资源分配和调度的一个独立的单位。

PCB、程序段、
相关数据段
CPU寄存器





程序与进
程的关系？



2. 进程的特征

除了具有的PCB结构外，还具有特征：

- (1) 动态性 生命期，创建 调度 撤销
- (2) 并发性 在内存 共存，同一段时间运行
- (3) 独立性 独立运行、获得资源、调度
- (4) 异步性 以不可预知速度推进，不可再现性





进程和程序的关系

- (1) 进程是动态的，程序是静态的；
- (2) 进程具有并行特征，程序没有；
- (3) 进程是竞争资源的基本单位；
- (4) 一个程序对应多个进程，一个进程为多个程序服务。





2.2.2 进程的基本状态及转换

1. 进程的三种基本状态

进程并发执行，共享资源，间断性运行，可能具有多种状态。三种基本状态：

- (1) 就绪(Ready)状态。 准备好，除了CPU
- (2) 执行(Running)状态。 正在执行，占有CPU
- (3) 阻塞(Block)状态。 受阻暂停，释放CPU

就绪队列： 多个就绪进程排成一个队列

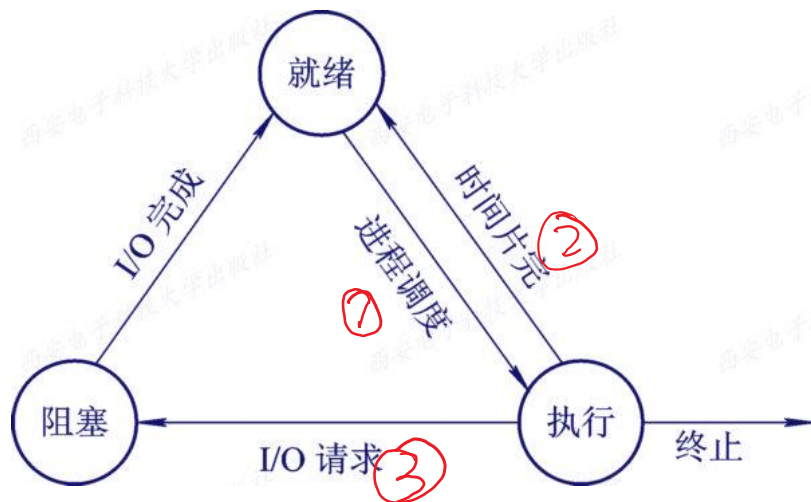




2. 三种基本状态的转换

进程三种基本状态以及转换

进程由调度而执行



①就绪态转执行态：就绪态的进程，分配了处理机

②执行态转为就绪态：正在执行的进程，时间片已完而被剥夺处理机，暂停执行

③执行态转为阻塞态：因某事件，执行受阻。





3. 创建状态和终止状态

1) 创建状态

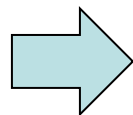
进程由创建而生

创建过程:

申请空白PCB, 并填写
控制和管理的信息;

为进程分配运行所必须
的资源;

把进程转入就绪状态, 插
入就绪队列之中。



创建状态

创建进程所需的
资源不能得到满足,
进程不能被调度运
行时所处的状态

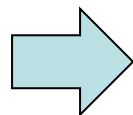




2) 终止状态

进程终止:

自然结束;
错误;
被操作系统/
其他有终止权的
进程所终结;



终止状态:

不能再执行, 保留状态码和计时统计数据;
信息被提取后, OS将其PCB清零, 删除PCB, 将空PCB返还系统。





进程五种状态及转换

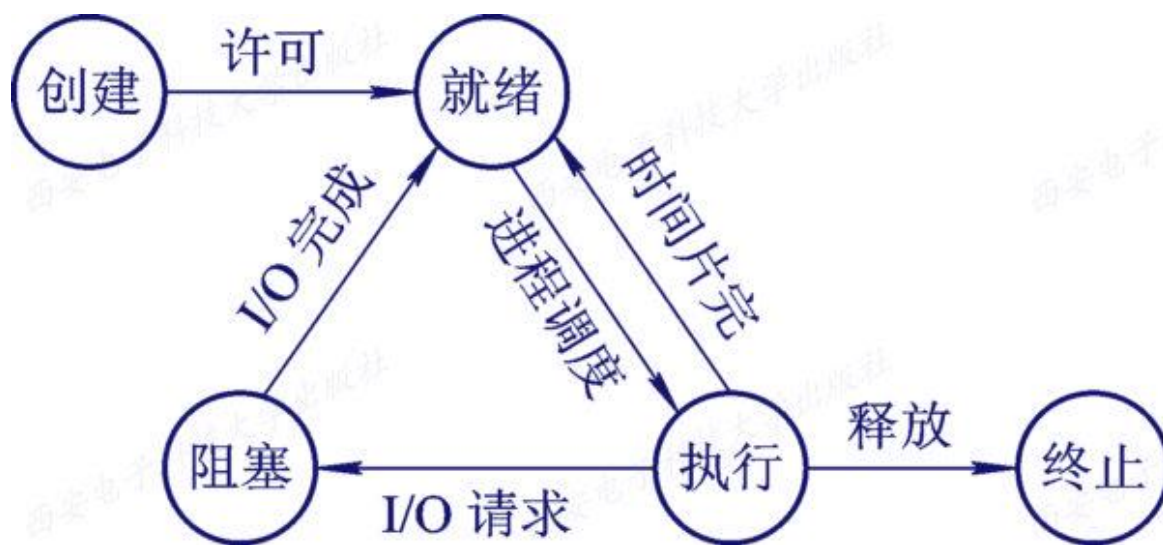


图2-6 进程的五种基本状态及转换





2.2.3 挂起操作和进程状态的转换

1. 挂起操作

引入原因：

- (1) 终端用户的需要。
- (2) 父进程请求。
- (3) 负荷调节的需要。
- (4) 操作系统的需要。

进程挂起：对换区，资源被剥夺
进程阻塞：占有资源，等待





2. 引入挂起操作的进程状态转换

在引入和激活后，进程可能发生状态转换：

- (1) 活动就绪→静止就绪。
- (2) 活动阻塞→静止阻塞。
- (3) 静止就绪→活动就绪。
- (4) 静止阻塞→活动阻塞。

挂起Suspend

激活Active





进程五状态转换

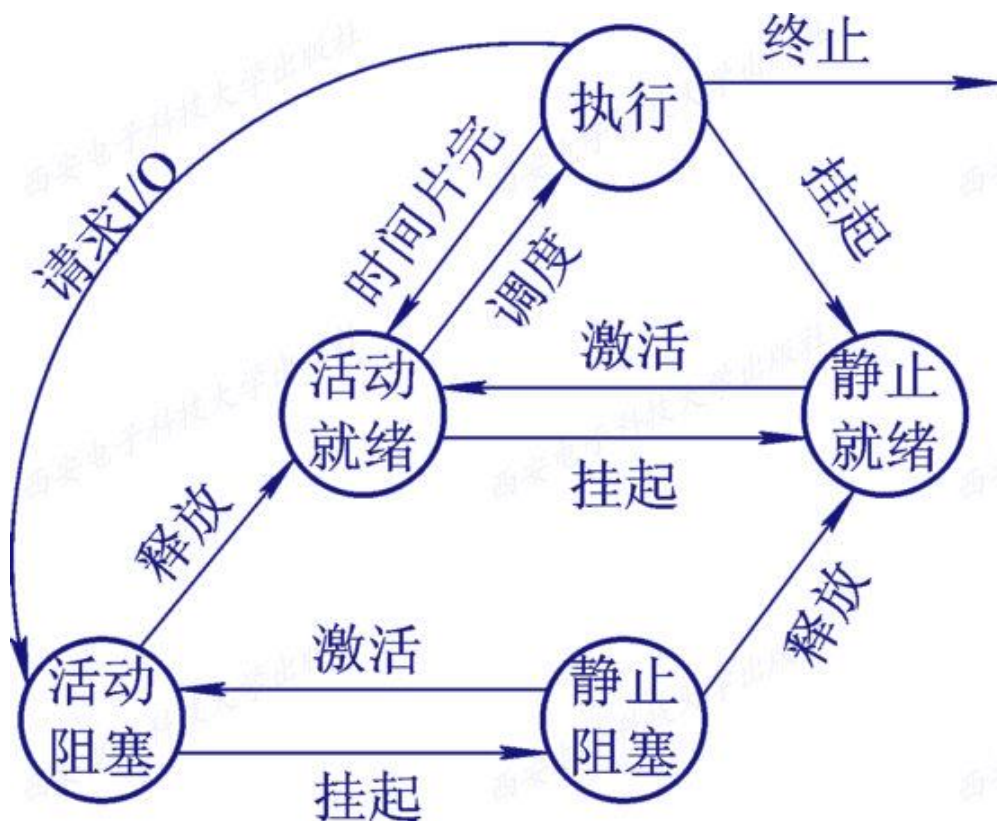
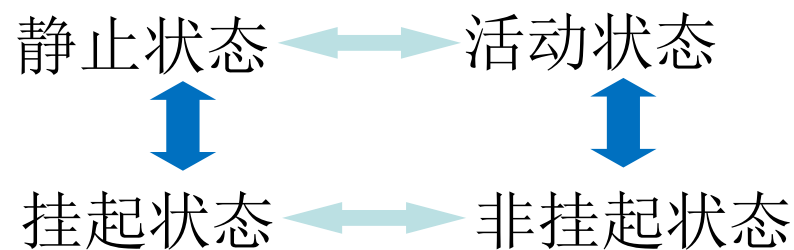


图2-7 具有挂起状态的进程状态图





3. 引入挂起操作后五个进程状态的转换

增加了创建、终止状态后具有挂起状态的进程状态及转换图。考虑下面的情况：

- (1) NULL→创建：
- (2) 创建→活动就绪：
- (3) 创建→静止就绪：
- (4) 执行→终止：



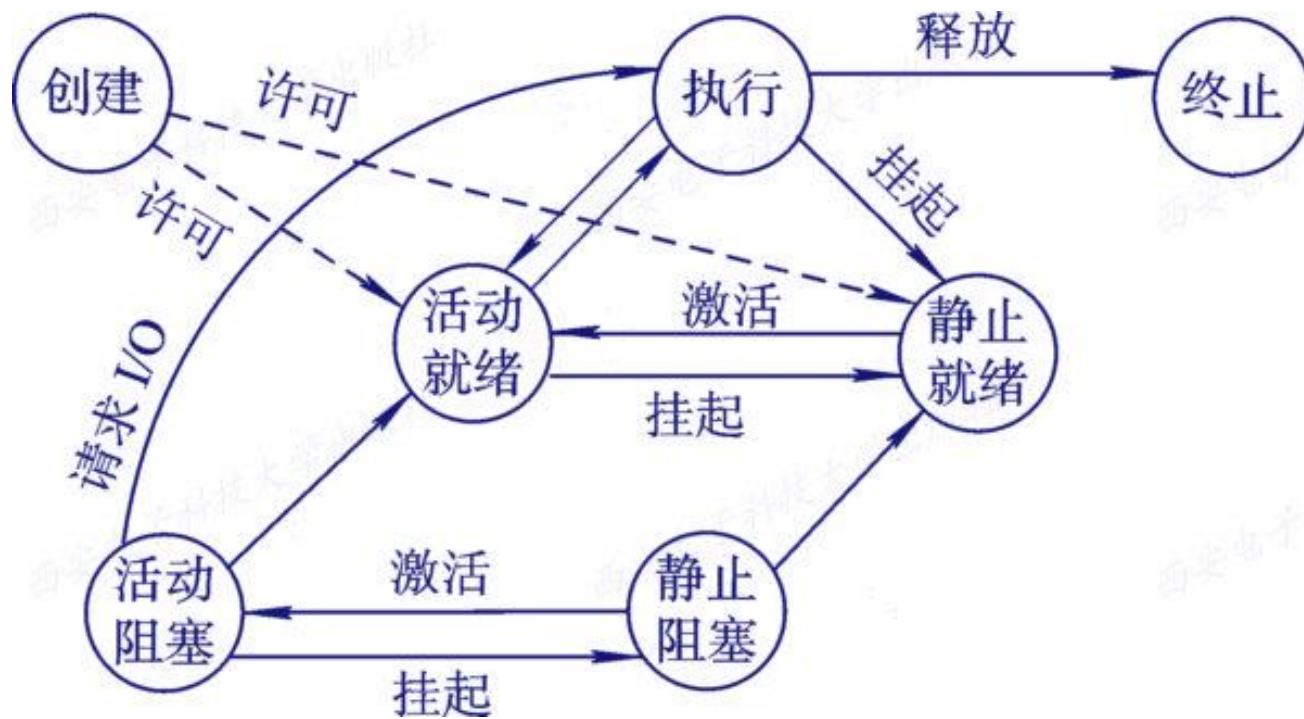


图2-8 具有创建、终止和挂起状态的进程状态图





2.2.4 进程管理中的数据结构

1. 操作系统中用于管理控制的数据结构

对于资源或进程用数据结构表征，包含资源或进程的标识、描述、状态等信息以及一批指针。

OS管理的这些数据结构分为四类：

{ 内存表、
设备表、
文件表、
进程表





操作系统控制表

OS专门开辟PCB区将所有的PCB组织成若干个链表或队列

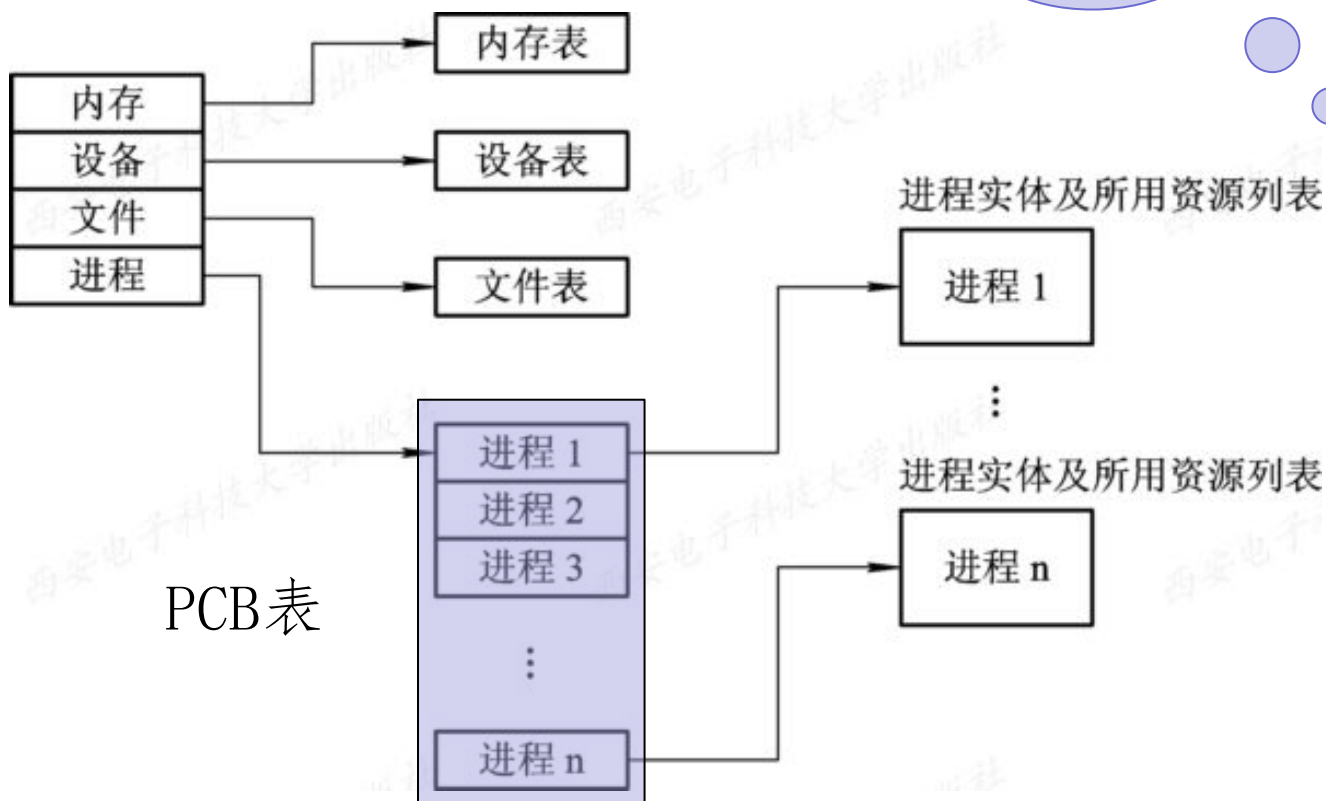


图2-9 操作系统控制表的一般结构





2. 进程控制块PCB的作用 P₄₀

(1) 作为独立运行基本单位的标志。

通过PCB感知进程，存在的唯一标志

(2) 能实现间断性运行方式。

CPU现场信息保存在PCB中，保存/恢复现场

(3) 提供进程管理所需要的信息。

根据PCB对进程控制与管理：访问程序、数据
文件和I/O 设备；资源清单





2. 进程控制块PCB的作用 P_{40}

(4) 提供进程调度所需要的信息。

状态信息 优先级 ; 等待时间/执行时间

(5) 实现与其它进程的同步与通信。

同步信号量 通信队列





3. 进程控制块中的信息

进程控制块主要包括四方面的信息

- 1) 进程标识符
- 2) 处理机状态
- 3) 进程调度信息
- 4) 进程控制信息





3. 进程控制块中的信息

1) 进程标识符

进程标识符用于唯一地标识一个进程。

- (1) 外部标识符。由字母、数字组成，给用户使用
- (2) 内部标识符。进程唯一的数字编号，给OS使用



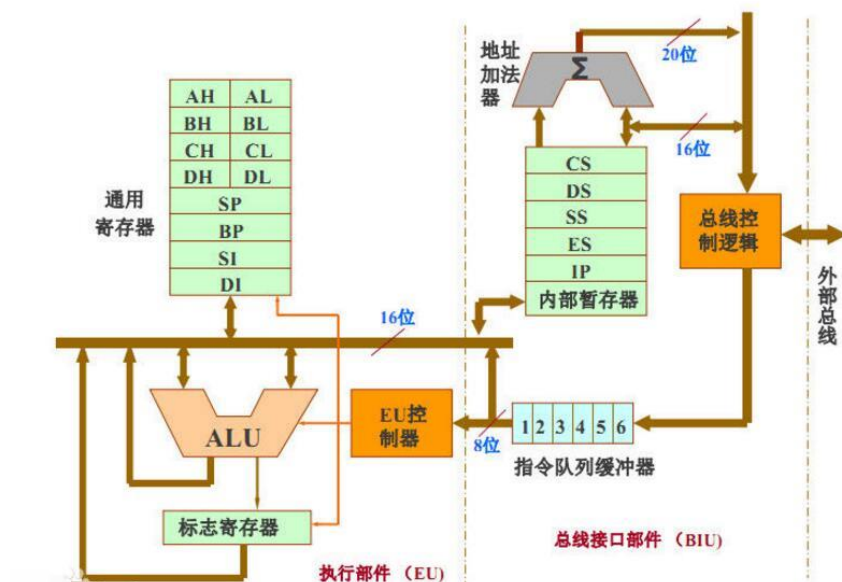


2) 处理机状态

P₄₁

处理机的上下文，主要是由处理机的各种寄存器中的内容组成的，保存在相应的PCB中。

通用寄存器
指令计数器
程序状态字
栈指针



系统栈

存放过程或系统调用参数及调用地址





3) 进程调度信息

- ① 进程状态，进程调度和对换依据；
- ② 进程优先级，优先级高优先获得处理机；
- ③ 调度所需其它信息，已等待时间总和、
已执行时间总和等；
- ④ 事件，阻塞原因。





4) 进程控制信息

- ① 程序和数据的地址，程序和数据地址
- ② 同步和通信机制，消息队列指针、信号量等，
- ③ 资源清单，所需的全部资源(除CPU以外)，
已分配的资源；
- ④ 链接指针，所在队列的下一个PCB首地址。





4. 进程控制块的组织方式

- (1) 线性方式
- (2) 链接方式
- (3) 索引方式





4. 进程控制块的组织方式

(1) 线性方式 所有PCB都存在一张线性表中，
表首址存放在内存专用区域。

PCB1
PCB2
PCB3
⋮
PCBn

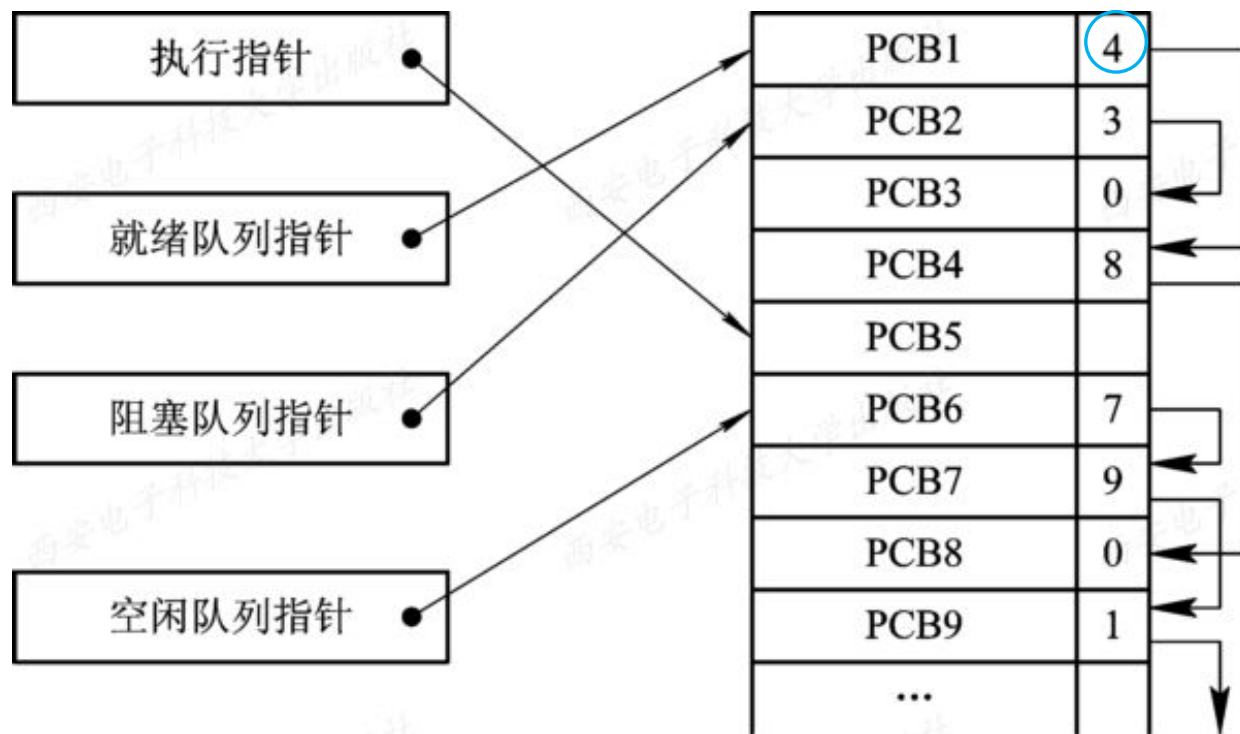
适合进程数目不多的系统：
实现简单、开销小；
每次查找需要扫描整张表





(2) 链接方式

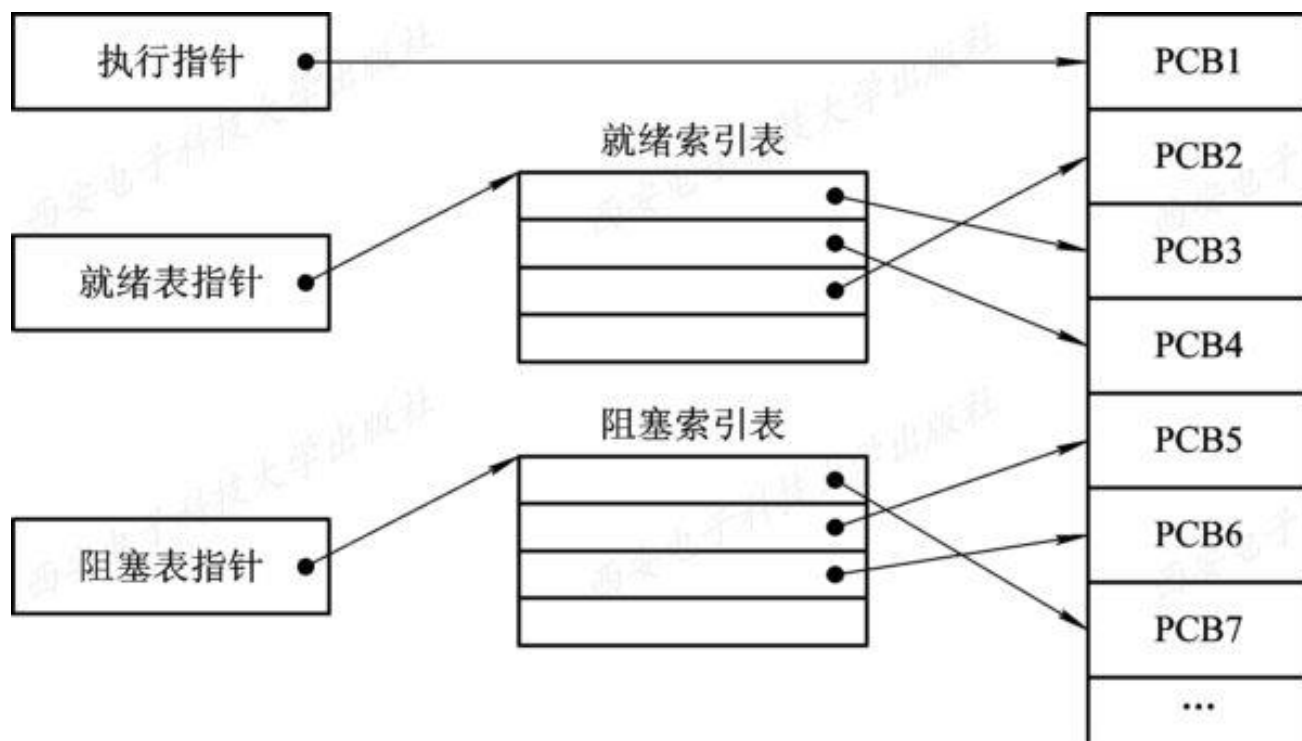
相同状态进程的PCB通过PCB链接字链接成队列。





(3) 索引方式

建立就绪索引表、阻塞索引表等，各索引表在内存的首地址记录在内存的专用单元中。





2.3 进程控制

进程管理最基本的功能：进程控制

{ 创建新进程、
终止进程、
进程状态转换等。

由OS内核的原语来实现。

原语：不允许
被中断、系统
态下执行





2.3.1 操作系统内核

OS内核 运行在系统态；用户程序运行在用户态

1. 支撑功能

(1) 中断处理。内核最基本功能

(2) 时钟管理。时钟中断

(3) 原语操作。不可分割 不允许被中断





2. 资源管理功能

(1) 进程管理。进程控制、调度 同步 通信

(2) 存储器管理。分配 保护 地址变换 对换

(3) 设备管理。分配、独立性 缓冲 驱动





2.3.2 进程的创建

1. 进程的层次结构

进程创建另一个进程，被创建的进程称为子进程，创建进程的进程称为父进程。

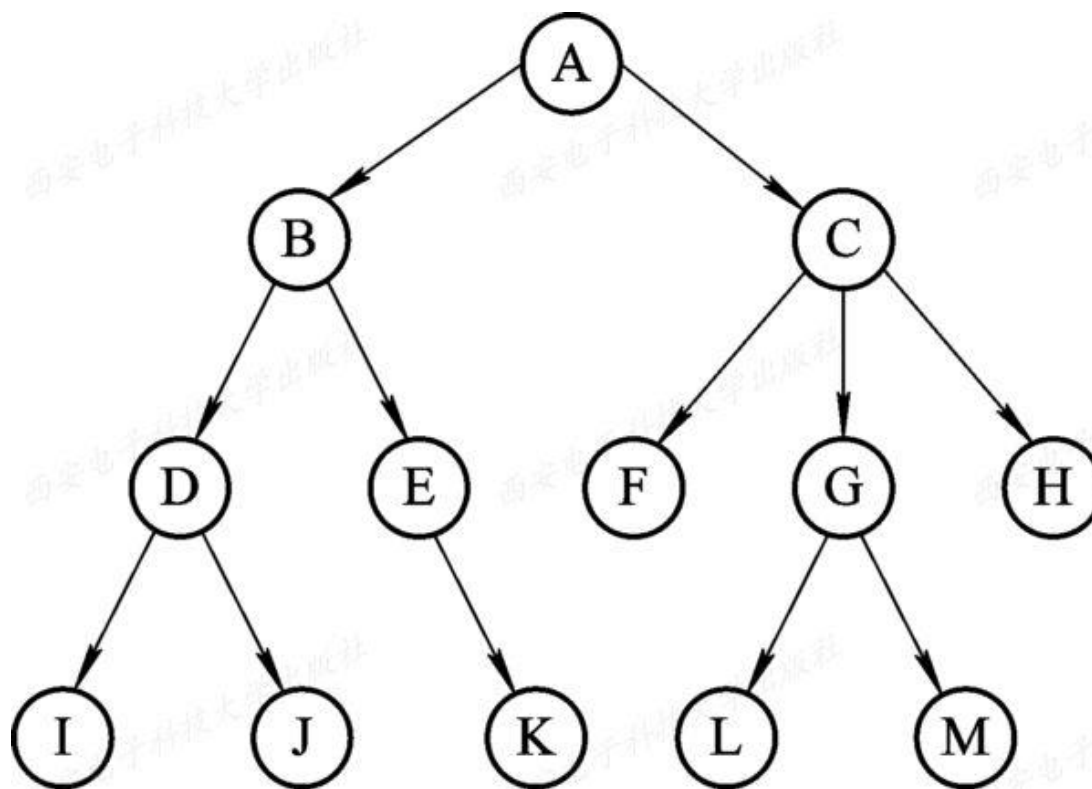
进程与其子孙进程组成一个次结构的进程家族(组)。





2. 进程图

描述进程的家族关系的有向树





3. 引起创建进程的事件

P₄₄

- (1) 用户登录。
- (2) 作业调度。
- (3) 提供服务。
- (4) 应用请求。

OS内核创建

用户进程创建





4. 进程的创建

进程创建原语Creat创建新进程：

- (1) 申请空白PCB，申请唯一标识符，从PCB集中获取一空白PCB；
- (2) 分配所需的资源，包括内存、文件、I/O设备和CPU时间等；
- (3) 初始化 PCB；
- (4) 若就绪队列能接纳，新进程插入就绪队列。

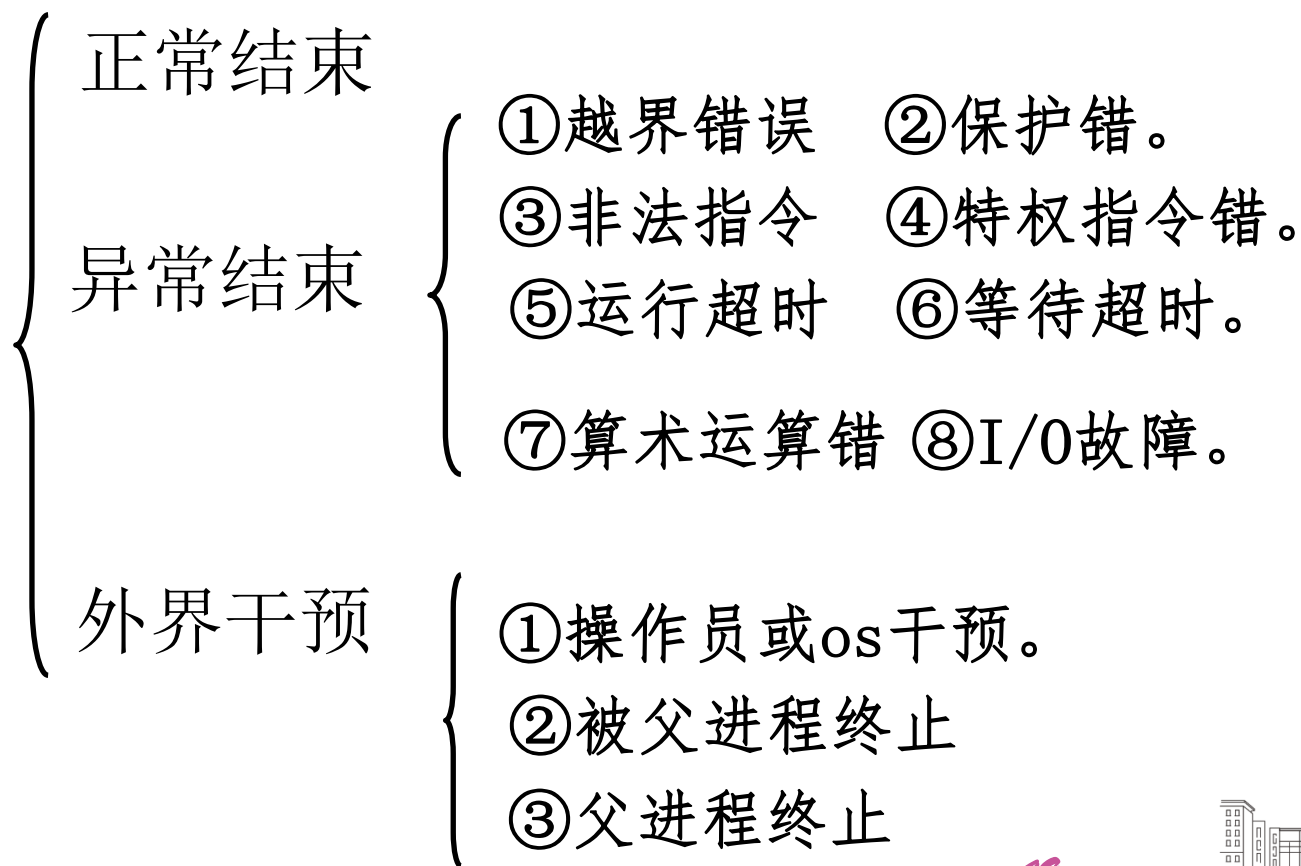




2.3.3 进程的终止

1. 引起进程终止的事件

P₄₅





2. 进程的终止过程

调用进程终止原语终止指定进程：

(1) 根据标识符，从PCB集合中检索出进程的PCB，读出该进程的状态；

(2) 若处于执行状态，终止该进程执行，置调度标志为真，该进程被终止后应重新进行调度；

(3) 若有子孙进程，将所有子孙进程终止，以防它们成为不可控进程；





(4) 将进程所拥有全部资源或者归还给其父进程或者系统；

(5) 将进程PCB从所在队列(或链表)中移出，等待其它程序来搜集信息。





2.3.4 进程的阻塞与唤醒

1. 引起进程阻塞和唤醒的事件

引起进程阻塞或被唤醒事件：

- (1) 向系统请求共享资源失败。
- (2) 等待某种操作的完成。
- (3) 新数据尚未到达。
- (4) 等待新任务的到达。





2. 进程阻塞过程

通过调用阻塞原语block将自己阻塞。 主动行为

- ①停止执行，把PCB现行状态改为阻塞，
- ②将PCB插入阻塞队列。
- ③调度程序重新调度，处理机分配给另外就绪进程，并进行切换

保留被阻塞进程的处理机状态，按新进程的PCB中的处理机状态设置CPU的环境



不同事件设置多个阻塞队列





3. 进程唤醒过程

所期待事件发生，I/O操作完成，或数据到达，
由有关进程调用唤醒原语wakeup，唤醒等待进程

唤醒过程：

- ①从阻塞队列中移出，
- ②将其PCB现行状态改为就绪，
- ③将PCB插入到就绪队列中。





2.3.5 进程的挂起与激活

1. 进程的挂起

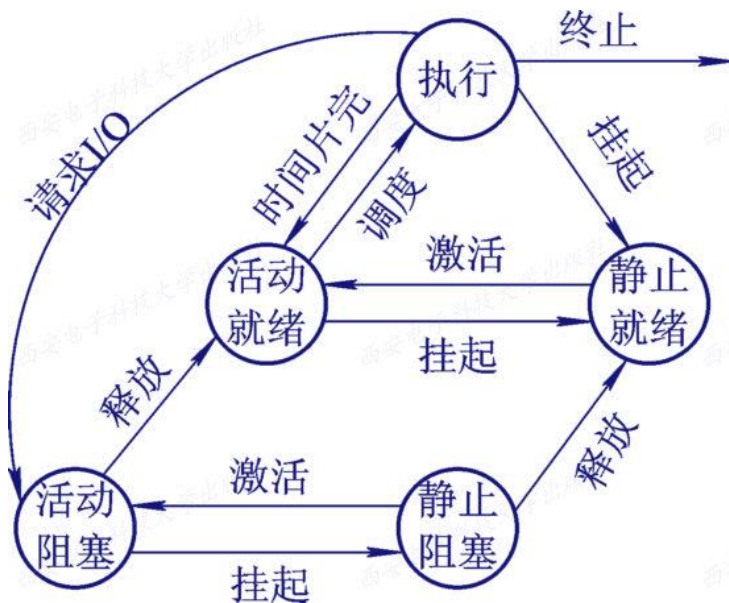
活动就绪 \rightarrow 静止就绪；

活动阻塞 \rightarrow 静止阻塞

2. 进程的激活过程

静止就绪 \rightarrow 活动就绪；

静止阻塞 \rightarrow 活动阻塞



? 可抢占
方式



对换 : 内存 \rightarrow 外存





进程切换

处理机从一个进程运行转到另一个进程运行，运行环境产生实质性变化

进程切换的过程：

保存处
理机上
下文

更新
PCB
信息

PCB移
入相应
的队列

选择另
一进程
执行，
更新其
PCB

更新内
存管理
的数据
结构

恢复新
进程处
理机上
下文

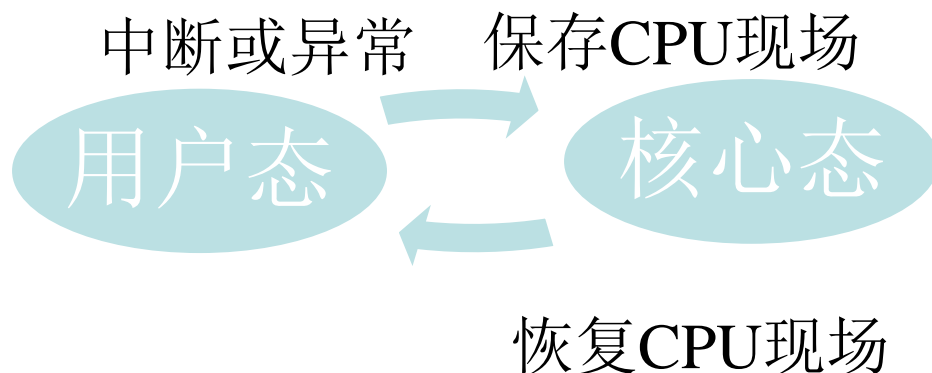
处理机的状态信息都保存在相应的PCB中，以便能从断点继续执行





进程切换与处理机模式切换

模式切换： 处理机逻辑可能还在同一进程中运行



进程切换： 当前运行进程改变，当前进程环境信息需要改变。

