高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

计算机操作系统

liuyong@ustc.edu.cn

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

第三章 处理机调度与死锁

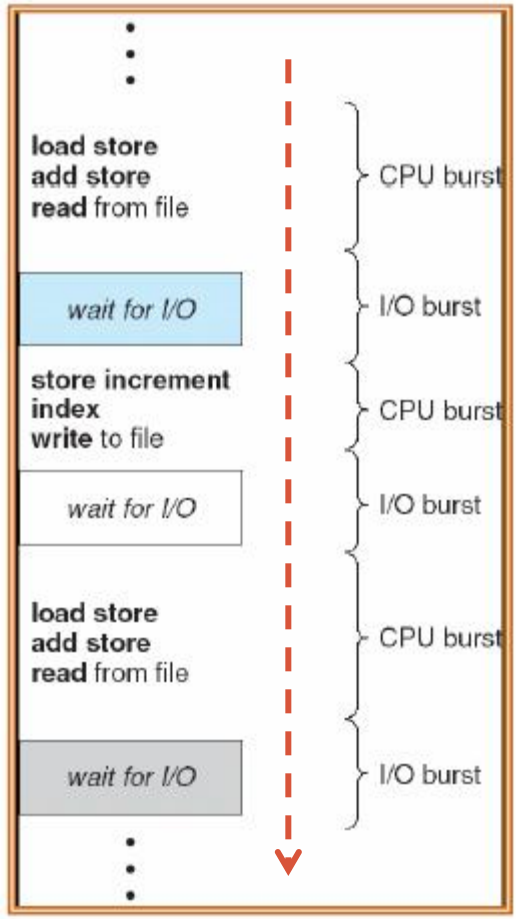
3.1 处理机调度的基本概念
3.2 调度算法
3.3 实时调度
3.4 多处理机系统中的调度
3.5 产生死锁的原因和必要条件
3.6 预防死锁的方法
3.7 死锁的检测与解除

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.1 处理机调度的基本概念

- 调度是OS必需的基本功能，几乎所有计算机资源都是经过调度而使用；
- CPU是计算机最重要的资源，通过在进程之间切换进行调度；
- 多道程序设计目标— CPU利用率的最大化；
- CPU – I/O 区间周期；
  - 进程执行由CPU执行周期和I/O周期组成，这种属性 是CPU调度的基础

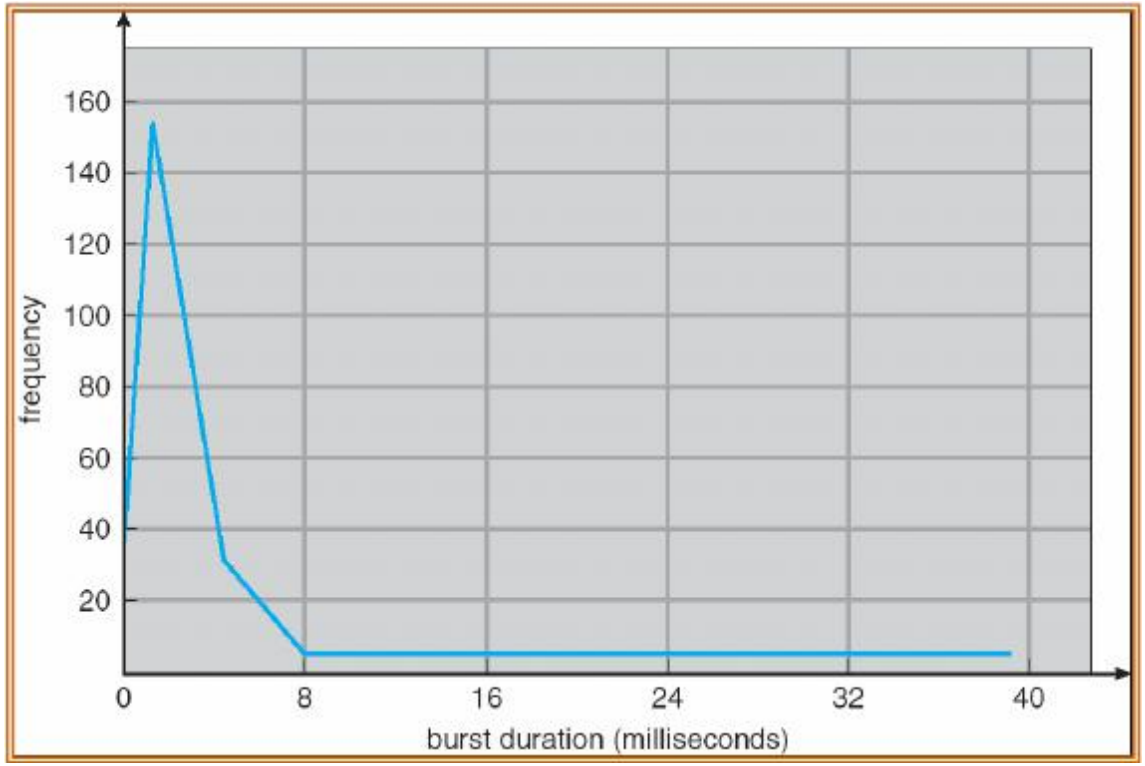


图：进程的观测属性：
CPU区间—I/O区间—CPU区间…

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

- CPU区间时间分布情况有助于选择合适的CPU调度算法。大量统计显示，虽然不同计算机区别较大，但都呈现类似的频率曲线 ——**具有大量短CPU区间和少量长CPU区间**。



CPU区间时间统计图

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.1.1 高级、中级和低级调度

- CPU调度程序：当CPU空闲时选择一个进程来执行。
- CPU调度又称**短程调度/低级调度**。
- 相应的，还有高级和中级调度。

1. 高级调度(High level Scheduling)
---作业调度、接纳调度、长程调度

- 接纳外存上的后备队列中的作业。
- 分时系统中不需要作业调度机制。

- 在每次执行作业调度时，都须做出以下两个决定。囍

1. 接纳多少个作业
2. 接纳哪些作业

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 低级调度/短程调度/CPU调度
(Low Level Scheduling / Short-term Scheduling)

从内存中的就绪进程队列中选择一个进程来执行。

CPU 调度决策发生在4种情况下：

1. 当一个进程终止时
2. 当一个进程从运行状态切换到等待状态时（例如 I/O或wait）
3. 当一个进程从运行状态切换到就绪状态时（例如出现中断）
4. 当一个进程从等待状态切换到就绪状态时（例如 I/O完成）

当调度仅发生在1、2 两种情况时称作非抢占的（ Non-preemptive ）；否则称为可抢占的（ Preemptive ）。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 低级调度/短程调度 （续）

1) 非抢占方式(Non-preemptive Mode)囍

引起进程非抢占调度调度的因素：

① 正在执行的进程执行完毕， 或因发生某事件而不能再继续执行；

② 执行中的进程因提出I/O请求而暂停执行；

③ 在进程通信或同步过程中执行了某种原语操作，如P操作(wait操作)、Block原语、Wakeup原语等。

这种调度方式的优点是实现简单、系统开销小 。但它难以满足紧急任务的要求——立即执行。实时系统中，不宜采用这种调度方式。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 低级调度/短程调度 （2）
(Low Level Scheduling / Short-term Scheduling)

2) 抢占方式(Preemptive Mode)

允许暂停某个进程的执行重新分配处理机

抢占的原则有：囍

((1))优先权原则。

(2) 短作业(进程)优先原则。

(3) 时间片原则。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3. 中级调度(Intermediate-Level Scheduling)

•又称中程调度(Medium-Term Scheduling)。

•引入目的:为了提高内存利用率和系统吞吐量。

•方法：对换（swap）-把暂时不能运行的进程调至外存上去等待，把此时的进程状态称为就绪驻外存状态或**挂起状态**。中级调度决定把外存上的哪些又具备运行条件的就绪进程，重新调入内存，挂在就绪队列上等待进程调度。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.1.2 调度队列模型

1. 仅有进程调度的调度队列模型

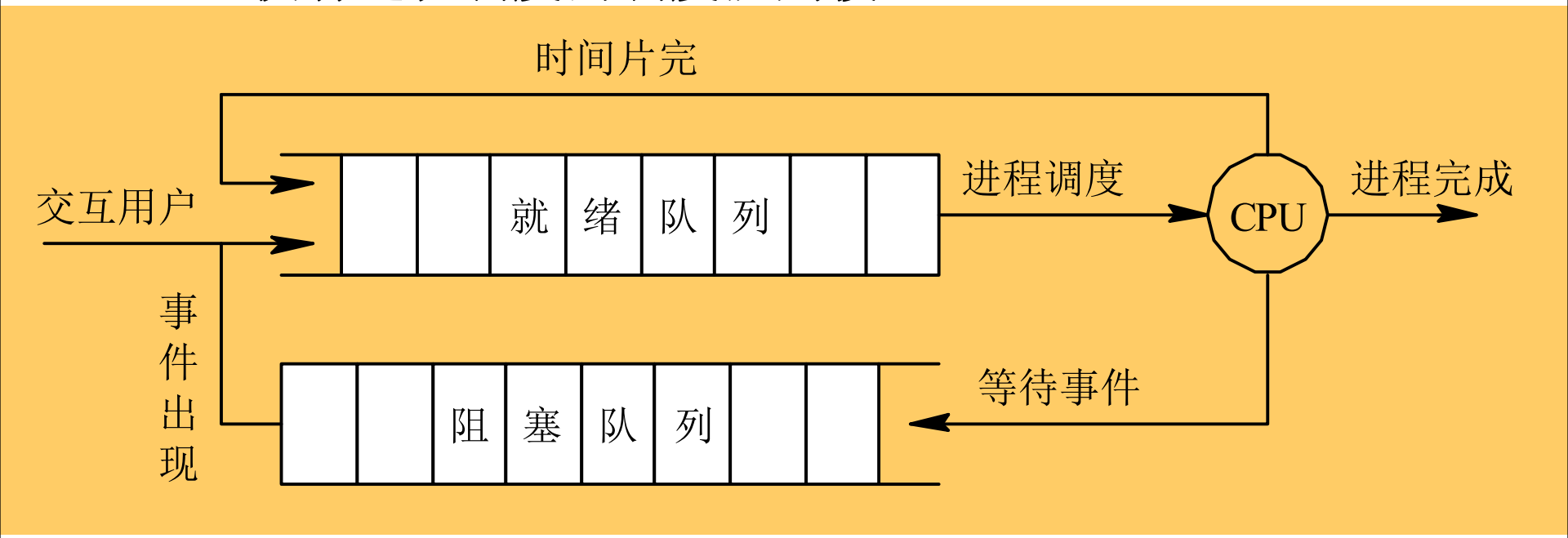


图 3 - 1 仅具有进程调度的调度队列模型

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 具有高级和低级调度的调度队列模型

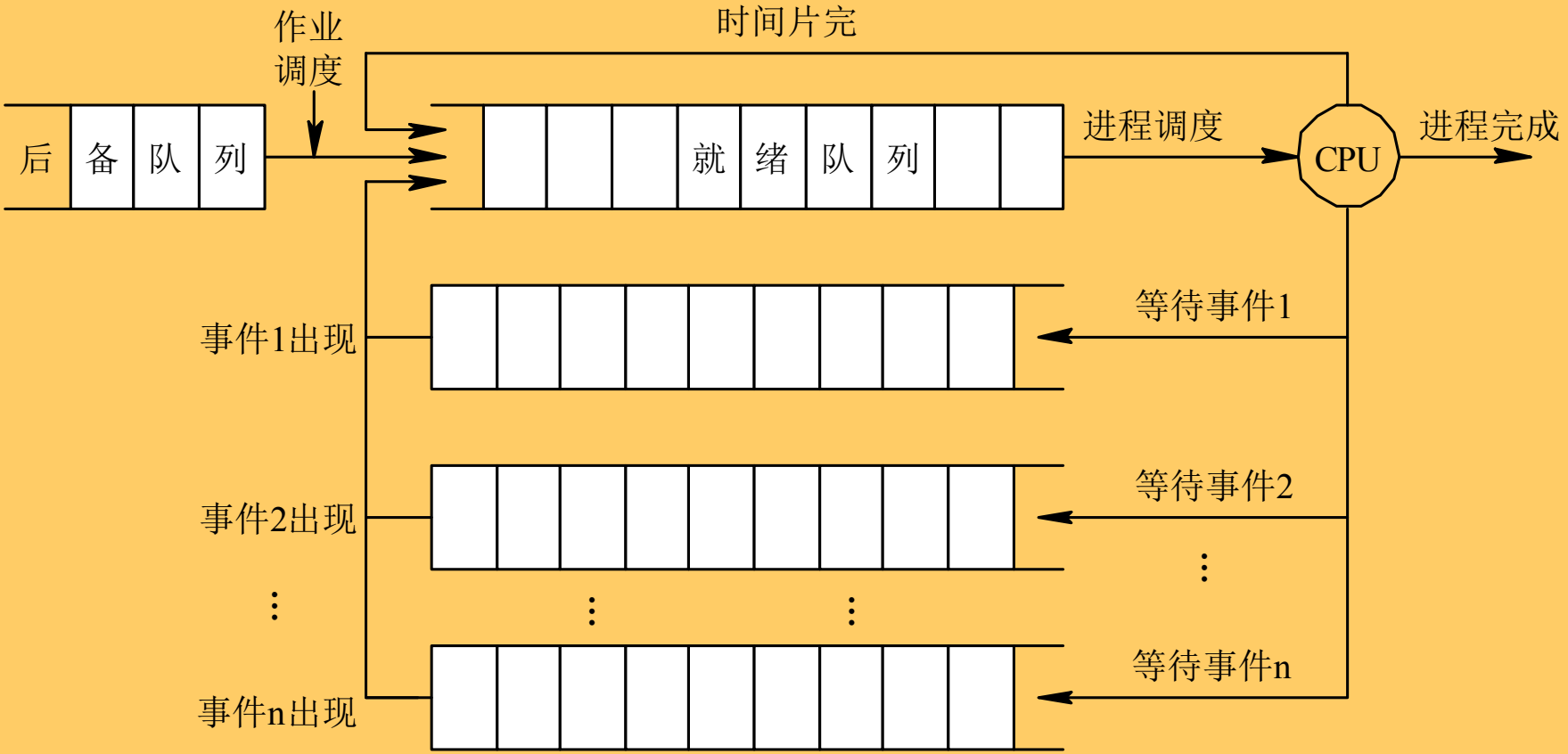


图 3-2 具有高、低两级调度的调度队列模型

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

具有高、低两级调度的调度队列模型 与上一模型的主要区别在于如下两个方面：

((1))就绪队列的形式。

(2) 设置多个阻塞队列。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3. 同时具有三级调度的调度队列模型

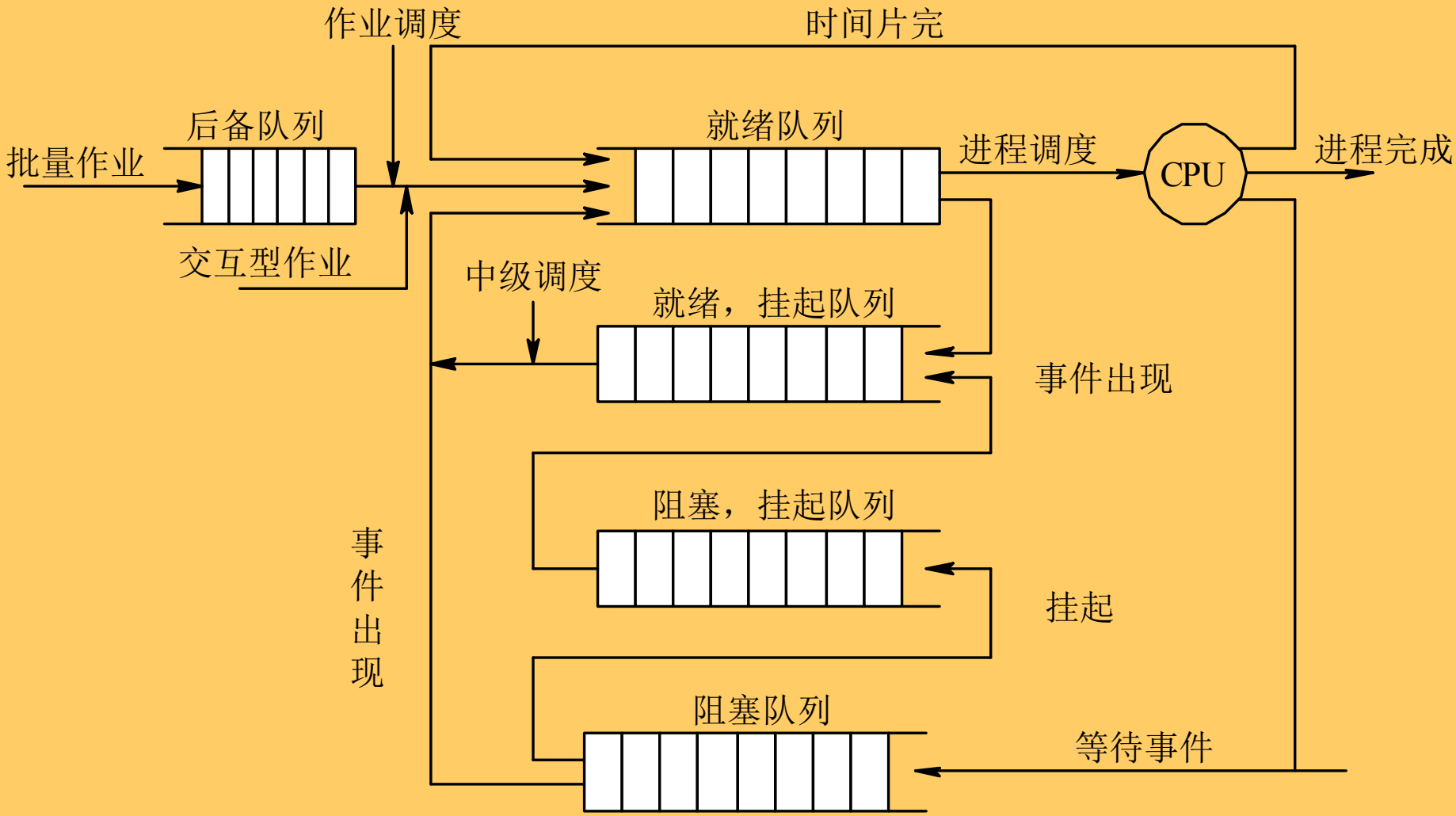


图 3-3 具有三级调度时的调度队列模型

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.1.3 选择调度方式和调度算法的若干准则

1. 面向用户的准则

((1))周转时间短。 周转时间- 作业从提交到完成的时间长度

可把平均周转时间描述为：$$T=\frac{1}{n}\left[\sum_{i=1}^{i}T_i\right]$$

带权周转时间：作业的周转时间T与系统为它提供服务的时间T_S之比，即$W=T/T_S$，称为带权周转时间，而平均带权周转时间则可表示为:

$$W=\frac{1}{n}\left[\sum_{i=1}^{n}\frac{T_i}{T_{Si}}\right]$$

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

（续）

(2) 响应时间快。（交互式系统中，响应时间比周转时间更有意义）

(3) 截止时间的保证。（实时系统调度）

(4) 优先权准则。（保证紧急重要作业及时处理）

等待时间：----在就绪队列中等待的时间之和。

CPU调度算法并不影响进程运行和I/O时间，只影响等待时间。

所以等待时间是衡量调度算法性能的较好指标。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 面向系统的准则囍

((1))系统吞吐量高。

(2) 处理机利用率好。 40% (轻负荷) --- 90%（重负荷）

(3) 各类资源的平衡利用。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.2 调 度 算 法

3.2.1 先来先服务和短作业(进程)优先调度算法

1. 先来先服务调度算法 （最简单的调度算法）

进程名	到达时间	服务时间	开始执行时间	完成时间	周转时间	带权周转时间
A	0	1	0	1	1	1
B	1	100	1	101	100	1
C	2	1	101	102	100	100
D	3	100	102	202	199	1.99

- 护航效应(*Convoy effect*): *短进程跟在长进程后面*

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 短作业(进程)优先调度算法

- 短作业(进程)优先调度算法SJ(P)F，是指对短作业或短进程优先调度的算法。它们可以分别用于作业调度和进程调度。
- 短作业优先(SJF)的调度算法，是从后备队列中选择一个或若干个估计运行时间最短的作业，将它们调入内存运行。
- 而短进程优先(SPF)调度算法，则是从就绪队列中选出一估计运行时间最短的进程，将处理机分配给它：
  - 非抢占：一直执行到完成，或被阻塞放弃处理机时，再重新调度。
  - 可抢占：新到达的进程时间比当前运行进程的剩余时间短，则抢占。——称作最短剩余时间优先调度算法。

SJF算法可证明为最佳，因为给定一组进程，该算法的平均等待时间最小。

短进程移到长进程之前，[短进程减少的等待时间> 长进程增加的等待时间]

所以平均等待时间减少了。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

调度算法 \ 作业情况	进程名	A	B	C	D	E	平均
	到达时间	0	1	2	3	4	
	服务时间	4	3	5	2	4	
FCFS (a)	完成时间	4	7	12	14	18	
	周转时间	4	6	10	11	14	9
	带权周转时间	1	2	2	5.5	3.5	2.8
SJF (b)	完成时间	4	9	18	6	13	
	周转时间	4	8	16	3	9	8
	带权周转时间	1	2.67	3.1	1.5	2.25	2.1

图 3-4 FCFS和SJF调度算法的性能

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

SJF 调度算法也存在不容忽视的缺点：

1. 该算法对长作业不利，如作业C的周转时间由10增至16，其带权周转时间由2增至3.1。更严重的是，如果有一长作业(进程)进入系统的后备队列(就绪队列)，由于调度程序总是优先调度那些(即使是后进来的)短作业(进程)，将导致长作业(进程)长期不被调度。
2. 该算法完全未考虑作业的紧迫程度，因而不能保证紧迫性作业(进程)会被及时处理。
3. 由于作业(进程)的长短只是根据用户所提供的估计执行时间而定的，而用户又可能会有意或无意地缩短其作业的估计运行时间，致使该算法不一定能真正做到短作业优先调度。 对短期CPU调度的层次无法真正实现SJF。

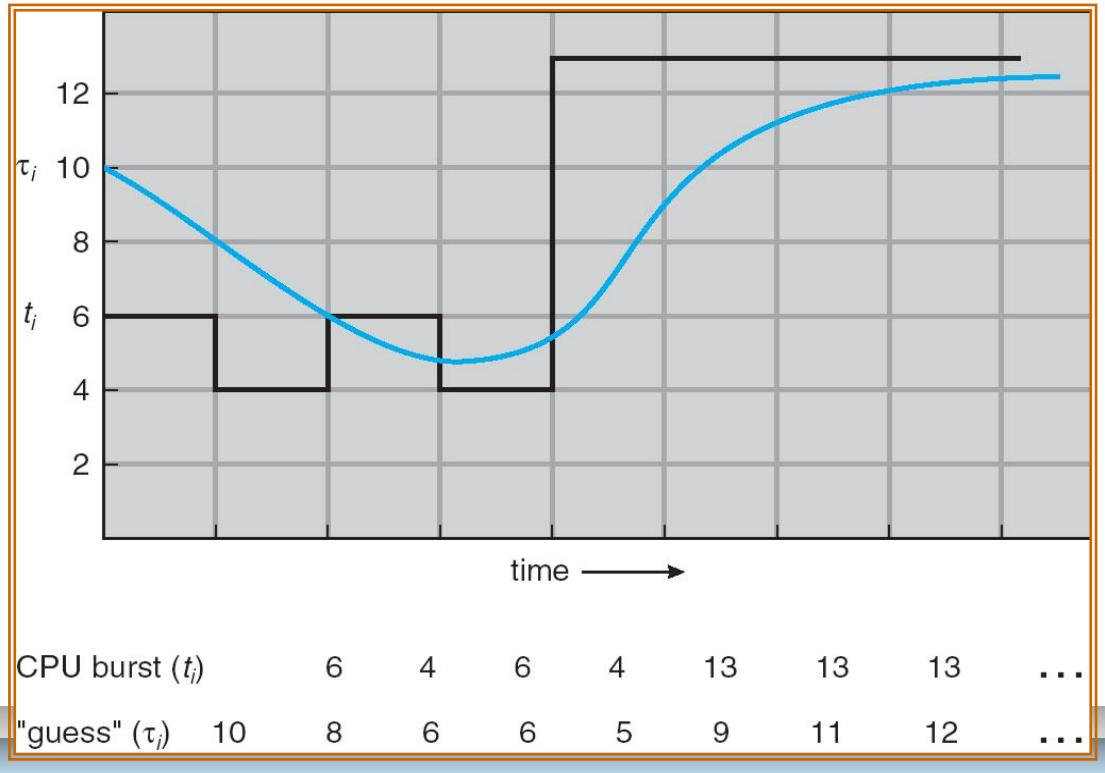
- SJF的真正困难是如何知道下一个CPU请求的长度。
- SJF不能在短程CPU调度上实现- 无法预知下一个CPU区间的长度。它可以在长程调度上实现（用户提交作业时指定预计时间）。
- 可利用一些预测方法实现近似SJF调度。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

下一个CPU区间长度的预测

1. t_n = actual length of nth CPU burst
2. τ_{n+1} = predicted value for the next CPU burst
3. $\alpha, 0 \le \alpha \le 1$
4. Define: $\tau_{n+1} = \alpha\, t_n + (1-\alpha)\tau_n.$



完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.2.2 优先权调度

- 每个进程关联一个优先权(整数);
- CPU被分配给优先权最高的(整数最小的)进程
- 【SJF也可看作此类调度的一个特例】

1. 优先权调度算法的类型

1) 非抢占式优先权算法

在这种方式下，系统一旦把处理机分配给就绪队列中优先权最高的进程后，该进程便一直执行下去，直至完成； 或因发生某事件使该进程放弃处理机时，系统方可再将处理机重新分配给另一优先权最高的进程。这种调度算法主要用于批处理系统中；也可用于某些对实时性要求不严的实时系统中。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2) 抢占式优先权调度算法囍

------把处理机分配给优先权最高的进程，使之执行。但在其执行期间，只要又出现了另一个其优先权更高的进程，进程调度程序就立即停止当前进程(原优先权最高的进程)的执行，重新将处理机分配给新到的优先权最高的进程。

------在采用这种调度算法时，是每当系统中出现一个新的就绪进程i时，就将其优先权P_i与正在执行的进程j的优先权P_j进行比较。如果$P_i \leq P_j$，原进程P_j便继续执行；但如果是$P_i > P_j$, 则立即停止P_j的执行，做进程切换，使i进程投入执行。

------ 这种抢占式的优先权调度算法，能更好地满足紧迫作业的要求，故而常用于要求比较严格的实时系统中以及对性能要求较高的批处理和分时系统中。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 优先权的类型

1) 静态优先权囍

静态优先权是在创建进程时确定的，且在进程的整个运行期间保持不变。

2) 动态优先权囍

动态优先权是指，在创建进程时所赋予的优先权，是可以随进程的推进或随其等待时间的增加而改变的，以便获得更好的调度性能。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

优先权定义方式:

1. 内部方式: 使用一些可测量数据来计算进程优先权. 例如时限, 内存要求,打开文件数量,平均I/O时间等等. (进程类型，资源需求)
2. 外部方式: OS之外的准则.进程重要性,付费类型和数量,人为因素等. （用户要求）

➢优先权调度算法的一个**主要问题**: ------**饥饿（starvation）/或称无限阻塞**（indefinite blocking） -- 低优先权进程长期得不到处理机。

- 一种解决方案: **老化(aging)/**动态优先权；
  即 逐渐增加在系统中等待很长时间的进程的优先权.

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3. 动态优先权-- 高响应比优先调度算法

优先权的变化规律可描述为：

$$\text{优先权} = \frac{\text{等待时间} + \text{要求服务时间}}{\text{要求服务时间}}$$

由于等待时间与服务时间之和，就是系统对该作业的响应时间，故该优先权又相当于**响应比R_P**。据此，又可表示为：

$$\text{优先权} = \frac{\text{等待时间} + \text{要求服务时间}}{\text{要求服务时间}} = \frac{\text{响应时间}}{\text{要求服务时间}}$$

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

优点:

(1) 如果作业的等待时间相同，则要求服务的时间愈短，其优先权愈高，因而该算法有利于短作业。囍

(2) 当要求服务的时间相同时，作业的优先权决定于其等待时间，等待时间愈长，其优先权愈高，因而它实现的是先来先服务。囍 (3) 对于长作业，作业的优先级可以随等待时间的增加而提高，当其等待时间足够长时，其优先级便可升到很高， 从而也可获得处理机。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.2.3 基于时间片的轮转调度算法

1. 时间片轮转法Round Robin (RR)

在早期的时间片轮转法中，系统将所有的就绪进程按**FCFS**的原则，排成一个队列，每次调度时，把CPU分配给队首进程，并令其执行一个时间片(通常10ms -- 100ms)。当执行的时间片用完时，由一个计时器发出时钟中断请求，调度程序便据此信号来停止该进程的执行，(被**抢占**) 并将它送往就绪队列的末尾；然后，再把处理机分配给就绪队列中新的队首进程，同时也让它执行一个时间片。这样就可以保证就绪队列中的所有进程，在一给定的时间内，均能获得一时间片的处理机执行时间。

- RR算法的性能很大程度上取决于时间片的大小.
- 设计时间片必须长度大于上下文切换时间长度. 如果上下文切换时间约为时间片长度的10%, 则大概有10%的CPU时间花费在上下文切换上.

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 多级反馈队列调度算法

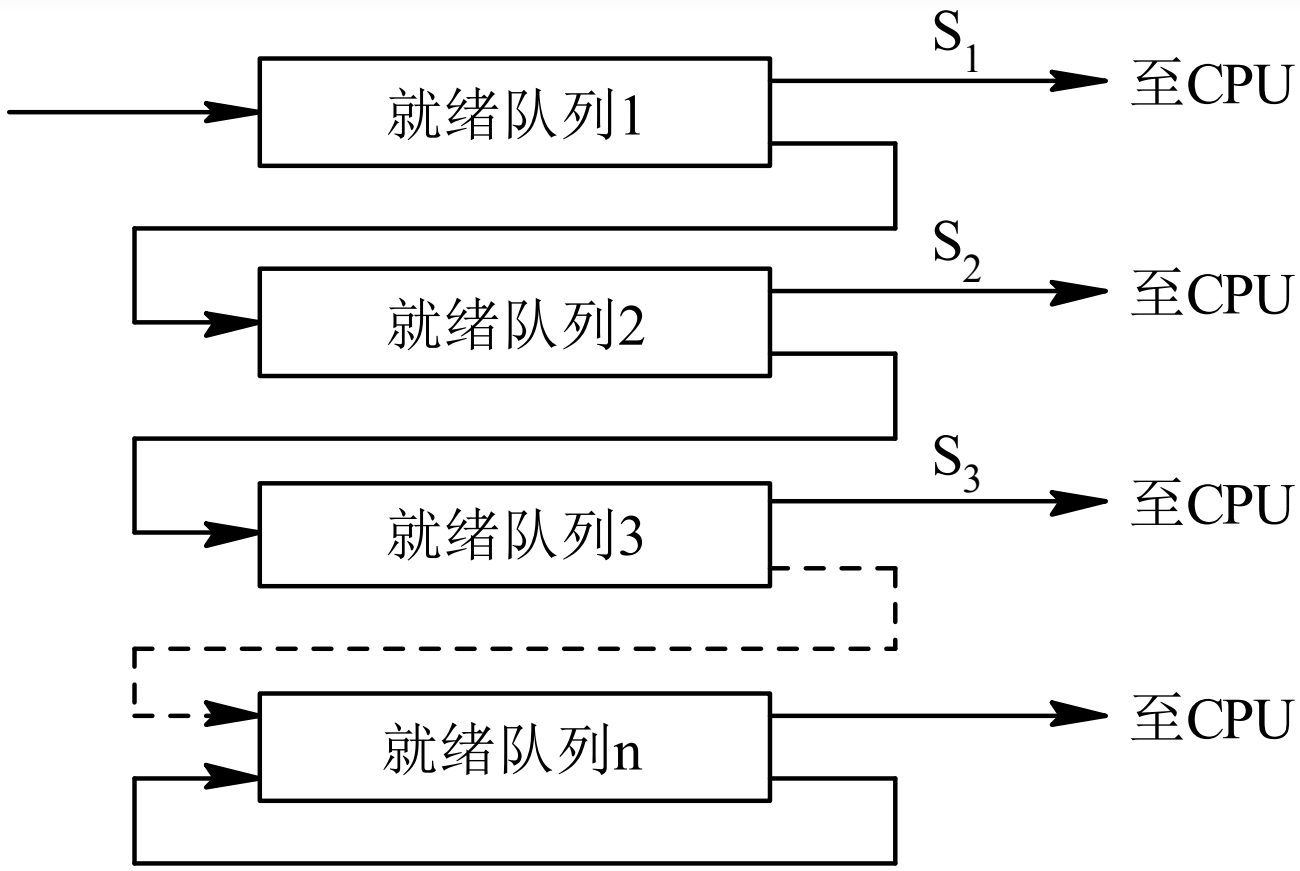
(1) 设置多个就绪队列，并依次为各个队列赋予不同的优先级。

✓第一个队列的优先级最高 。

✓不同队列中进程获得执行时间片大小也不同，优先权愈高时间片就愈小。　例如，第二个队列的时间片要比第一个队列的时间片长一倍，.....，第i+1个队列的时间片要比第i个队列的时间片长一倍。【如图 3-5】

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net



（时间片：$S_1 < S_2 < S_3$）

图 3-5 多级反馈队列调度算法

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(2)多级排队：

✓新进程进入内存后，首先将它放入第一队列的末尾，按FCFS原则排队等待调度。

✓进程执行时，如它能在该时间片内完成，便可准备撤离系统；如果它在一个时间片结束时尚未完成，调度程序便将该进程转入第二队列的末尾，再同样地按FCFS原则等待调度执行；如果它在第二队列中运行一个时间片后仍未完成，再依次将它放入第三队列，……，如此下去

✓当一个长作业(进程)从第一队列依次降到第n队列后，在第n队列中便采取按时间片轮转的方式运行。囍

✓老化（aging）在低优先级等待过长的进程会被移到较高优先队列，防止饥饿现象。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(3)先调度高优先级队列：

✓仅当第一队列空闲时，调度程序才调度第二队列中的进程运行；

✓仅当第 1～(i-1) 队列均空时，才会调度第i队列中的进程运行。

✓如果处理机正在第i队列中为某进程服务时，又有新进程进入优先权较高的队列(第1~(i-1)中的任何一个队列)，则此时新进程将**抢占**正在运行进程的处理机，即由调度程序把正在运行的进程放回到第i队列的末尾，把处理机分配给新到的高优先权进程。鍏

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

多级反馈队列调度算法的参数：

- 队列数量；
- 每个队列的调度算法；
- 决定何时升级到高优先权队列的算法；
- 决定何时降级到低优先权队列的算法；
- 决定进程在需要服务时进入哪个队列的算法。

3. 多级反馈队列调度算法的性能

((1)) 终端型作业用户。

(2) 短批处理作业用户。

(3) 长批处理作业用户。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.3 实 时 调 度

- 主要介绍通用计算机系统中用于支持实时计算的调度方法

3.3.1 实现实时调度的基本条件

- **1.** 提供必要的信息
  - ① 就绪时间。
  - ② 开始截止时间和完成截止时间。
  - ③ 处理时间。
  - ④ 资源要求。
  - ⑤ 优先级。

“资源预约” –

- 提交进程时有一条语句告诉系统上述时间信息，调度程序要么允许进程并保证按时完成，要么因为不可能完成而拒绝请求。这称作资源预约。
- 对于具有外存或虚拟内存的系统，无法保证按时完成，因为这些子系统会必然引起难以预计的时间量用于特定进程。故硬实时系统通常由带有硬件支持的专用软件系统构成，通常不具有现代计算机和操作系统的全部功能。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 系统处理能力强

若处理机的处理能力不够强，则有可能因处理机忙不过来而使某些实时任务不能得到及时处理， 从而导致发生难以预料的后果。假定系统中有m个周期性的硬实时任务，它们的处理时间可表示为C_i，周期时间表示为P_i，则在单处理机情况下，必须满足下面的限制条件：

$$\sum_{i=1}^{m}\frac{C_i}{P_i}\leq 1$$

假如系统中有6个硬实时任务，它们的周期时间都是 50 ms，而每次的处理时间为 10 ms，则不难算出，此时是不能满足上式的，因而系统是不可调度的。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

解决的方法是提高系统的处理能力，其途径有二：其一仍是采用单处理机系统， 但须增强其处理能力， 以显著地减少对每一个任务的处理时间；其二是采用多处理机系统。假定系统中的处理机数为N，则应将上述的限制条件改为：

$$\sum_{i=1}^{m} \frac{C_i}{P_i} \leq N$$

注： 上述条件尚未考虑任务切换时间等系统开销。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3. 采用抢占式调度机制

- 硬实时任务为了满足对截止时间的要求广泛采用可抢占调度。调度机制相对复杂。囂
- 对于一些小的(软)实时系统可简化调度程序 ，如果能预知任务的开始截止时间，则对实时任务的调度可采用非抢占调度机制，以减少调度时所花费的系统开销。
- 总的来说都必须具有**优先权调度**，实时进程必须具有最高优先权。为保证这一属性成立，可以不允许进程老化，使其优先权不变。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

4. 具有快速切换机制

分派延迟必须小。应具有如下两方面的能力：

(1) **对外部中断的快速响应能力**。为使在紧迫的外部事件请求中断时系统能及时响应，要求系统具有**快速硬件中断机构**，还应使禁止中断的时间间隔尽量短， 以免耽误时机(其它紧迫任务)。

(2) **快速的任务分派能力**。为提高分派程序进行任务切换时的速度， 应使系统中的每个运行功能单位适当的小，以减少任务切换的时间开销。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

两种方法降低分派延迟

① 允许系统调用是可抢占的，可在需要长时间的系统调用内插入**抢占点**，以检查高优先级进程是否需要运行，抢占点设在对内核“安全”的位置(当内核数据不是正在被修改时)。实际可增加的抢占点可能很有限。

① 允许整个内核可抢占，内核数据必须通过各种同步机制加以保护，不会受到高优先级进程的修改。

- 第二种方法最有效（也更复杂），应用最广。例如Solaris 2.

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.3.2 实时调度算法的分类

1. 非抢占式调度算法
（小型软实时系统）

((1))非抢占式轮转调度算法。

如工业生产中的群控系统。

(2) 非抢占式优先调度算法。

可用于实时要求稍高的系统。

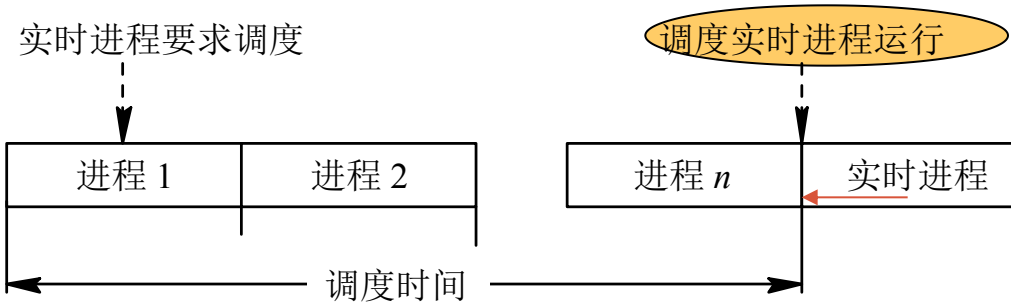
完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

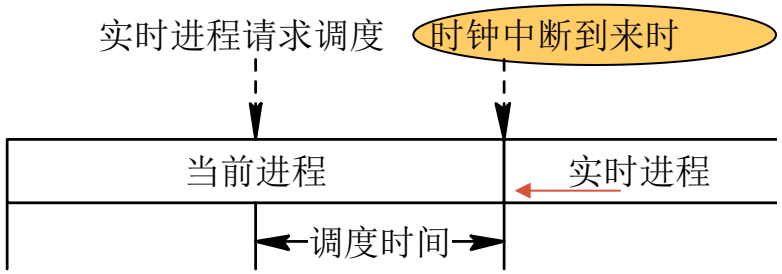
2. 抢占式调度算法

((1))基于时钟中断的抢占式优先权调度算法。（抢占发生在时钟中断到来时）

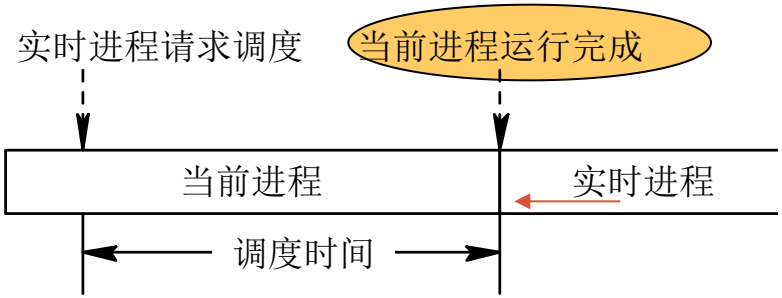
(2) 立即抢占(Immediate Preemption)的优先权调度算法。



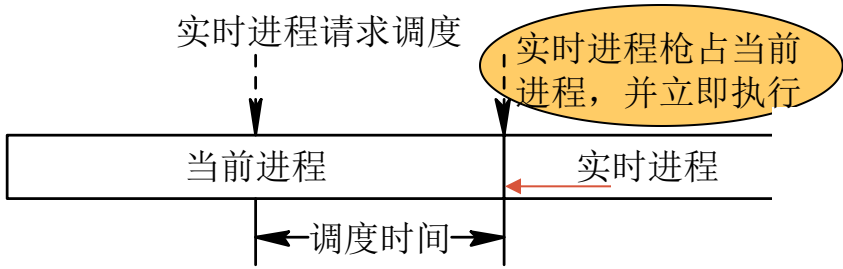
(*a*) 非抢占轮转调度



(*c*) 基于时钟中断抢占的优先权抢占调度



(*b*) 非抢占优先权调度



(*d*) 立即抢占的优先权调度

图 3-6 实时进程调度

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.3.3 常用的几种实时调度算法

1. 最早截止时间优先即EDF(Earliest Deadline First)算法

—— 截止时间越早，优先级越高。

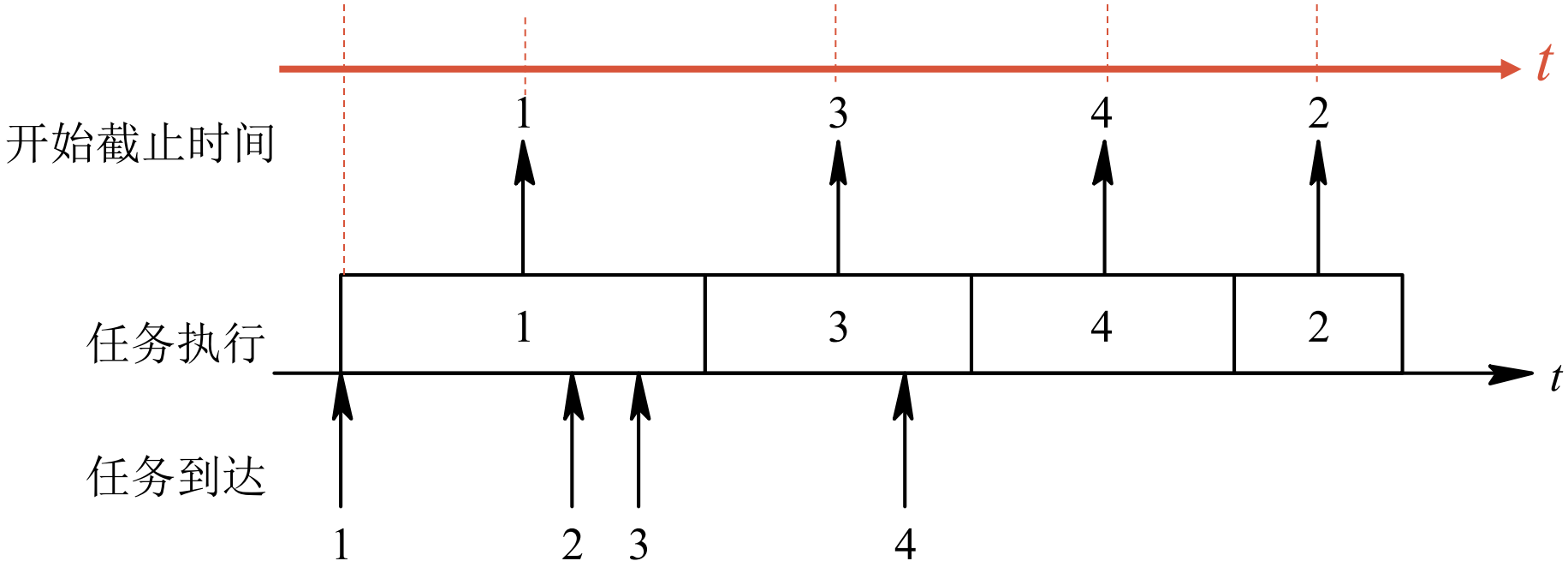


图 3-7 EDF算法用于非抢占调度方式

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 最低松弛度优先LLF(Least Laxity First)算法

任务的紧急程度愈高（松弛度越低），为该任务所赋予的优先级就愈高，优先执行。

松弛度＝ 必须完成时间－当前时间－本身运行时间

•在实现该算法时要求系统中有一个按松弛度排序的实时任务就绪队列，松弛度最低的任务排在队列最前面，调度程序总是选择就绪队列中的队首任务执行。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

•**例：** 在一个实时系统中，有两个周期性实时任务A和B，任务A要求每 20 ms执行一次，执行时间为 10 ms；任务B只要求每50 ms执行一次，执行时间为 25 ms。

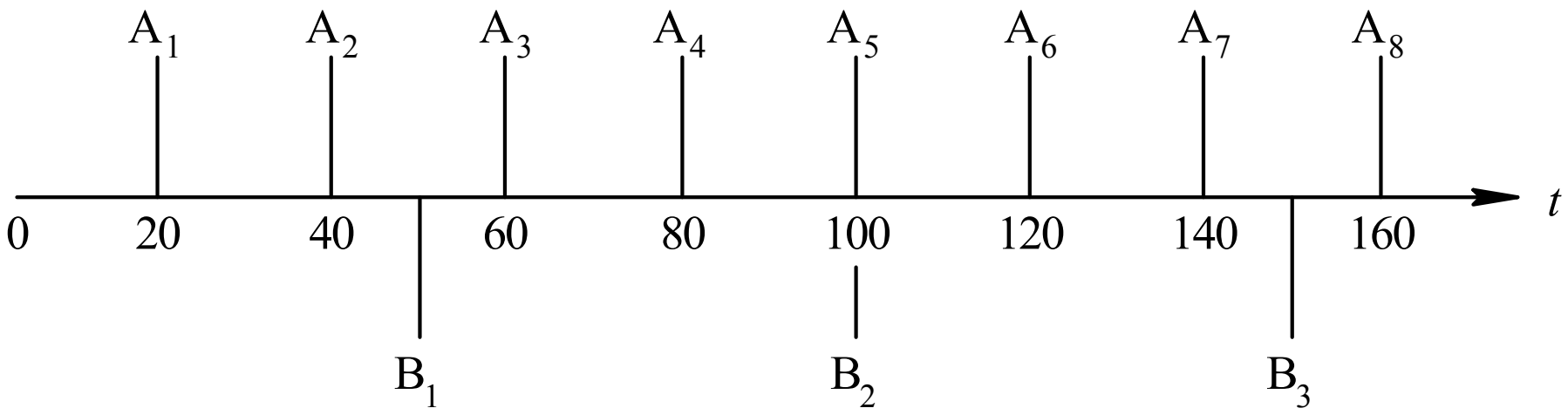


图 3-8 A和B任务每周期的截止时间

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

- 在刚开始时(t_1=0)：

  A_1必须在20ms时完成，而它本身运行又需 10 ms，可算出A_1的松弛度为10ms；B_1必须在50ms时完成， 而它本身运行就需25 ms，可算出B_1的松弛度为25 ms，故调度程序应先调度A_1执行。

- 在t_2=10 ms时：A_2的松弛度可按下式算出：

琥 A_2的松弛度=必须完成时间-其本身的运行时间-当前时间

=40 ms-10 ms-10 ms=20 ms

类似地，可算出B1的松弛度为15ms，故调度程序应选择B2运行。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

- 在t_3=30 ms时，A_2的松弛度已减为0(即40-10-30)，而B_1的松弛度为15 ms(即50-5-30)，于是调度程序应抢占B_1的处理机而调度A_2运行。
- 在t_4=40 ms时，A_3的松弛度为10 ms(即60-10-40)，而B_1的松弛度仅为5 ms(即50-5-40)，故又应重新调度B_1执行。
- 在t_5=45 ms时，B_1执行完成，而此时A_3的松弛度已减为5 ms(即60-10-45)，而B_2的松弛度为30 ms(即100-25-45)，于是又应调度A_3执行。
- 在t_6=55ms时，任务A尚未进入第4周期，而任务B已进入第2周期，故再调度B_2执行。
- 在t_7=70 ms时，A_4的松弛度已减至0 ms(即80-10-70)，而B_2的松弛度为20 ms(即100-10-70)，故此时调度又应抢占B_2的处理机而调度A_4执行。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

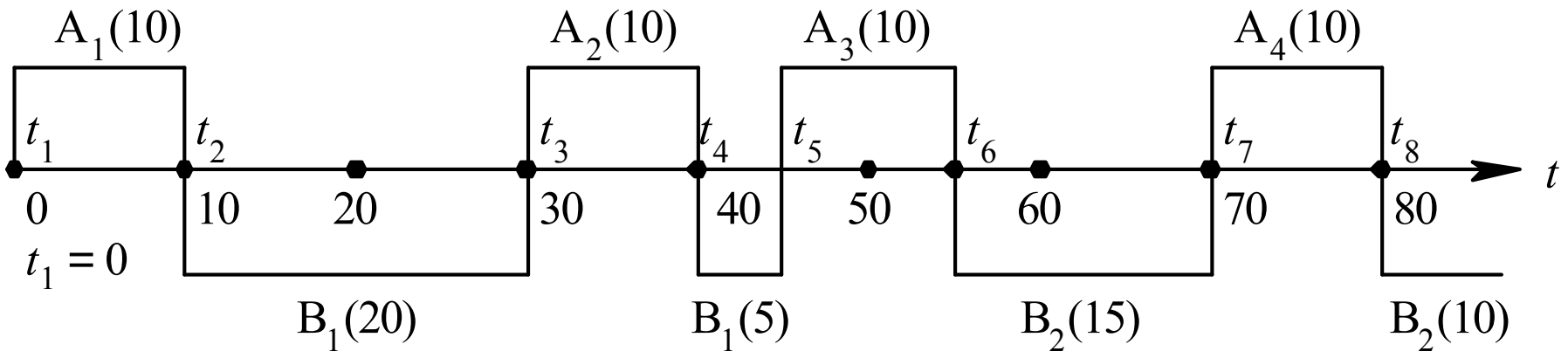


图 3-9 利用ELLF算法进行调度的情况

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.4 多处理机系统中的调度

3.4.1 多处理器系统的类型

1 紧密耦合和松散耦合系统

(1) 紧密耦合(Tightly Coupted)MPS。

这通常是通过高速总线或高速交叉开关，来实现多个处理器之间的互连的。它们共享主存储器系统和I/O设备，并要求将主存储器划分为若干个能独立访问的存储器模块，以便多个处理机能同时对主存进行访问。系统中的所有资源和进程，都由操作系统实施统一的控制和管理。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(2) 松散耦合(Loosely Coupled)MPS。

在松散耦合MPS中，通常是通过通道或通信线路，来实现多台计算机之间的互连。每台计算机都有自己的存储器和I/O设备，并配置了OS来管理本地资源和在本地运行的进程。因此，每一台计算机都能独立地工作， 必要时可通过通信线路与其它计算机交换信息，以及协调它们之间的工作。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 对称多处理器系统和非对称多处理器系统

(1) 对称多处理器系统SMPS(Symmetric MultiProcessor System)。 在系统中所包含的各处理器单元，在功能和结构上都是相同的（**同构**），当前的绝大多数MPS都属于SMP系统。例如，IBM公司的SR/6000 Model F50, 便是利用4片Power PC处理器构成的。囍

(2) 非对称多处理器系统。在系统中有多种类型的处理单元， 它们的功能和结构各不相同(**异构**)，其中只有一个主处理器，有多个从处理器。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.4.2 进程分配方式（负载分配load sharing）

1. 对称多处理器系统中的进程分配方式

在SMP系统中，处理器都是同构的，理论上可以将任何一个进程放在任何一个处理器上运行（实际可能的例外比如，一个处理器与某I/O设备之间通过私有总线相连，则希望使用该设备的进程必须被调度到该处理器上运行，否则无法使用该设备。）

可把所有的处理器作为一个处理器池(Processor pool)，由调度程序或基于处理器的请求， 将任何一个进程分配给池中的任何一个处理器去处理。

在进行进程分配时，可采用以下两种方式之一。

1) 静态分配(Static Assigenment)方式 ： 各处理器专用就绪队列

2) 动态分配(Dynamic Assgement)方式：公共就绪队列。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 非对称MPS中的进程分配方式囍

对于非对称MPS， 其OS大多采用主–从(Master-Slave)式OS， 即OS的核心部分驻留在一台主机上(Master)， 而从机(Slave)上只是用户程序，进程调度只由主机执行。 每当从机空闲时， 便向主机发送一索求进程的信号， 然后， 便等待主机为它分配进程。 在主机中保持有一个就绪队列， 只要就绪队列不空， 主机便从其队首摘下一进程分配给请求的从机。从机接收到分配的进程后便运行该进程， 该进程结束后从机又向主机发出请求。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.4.3 进程(线程)调度方式

1. 自调度(Self-Scheduling)方式

1) 自调度机制囍-- 每个处理器都检查公共就绪队列，选择一个进程来执行。

在多处理器系统中，自调度方式是最简单的一种调度方式。它是直接由单处理机环境下的调度方式演变而来的。 在系统中设置有一个公共的进程或线程就绪队列，所有的处理器在空闲时，都可自己到该队列中取得一进程(或线程)来运行。

在自调度方式中，可采用在单处理机环境下所用的调度算法，如先来先服务(FCFS)调度算法、最高优先权优先(FPF)调度算法和抢占式最高优先权优先调度算法等。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2) 自调度方式的优点囍

自调度方式的主要优点表现为：

首先，系统中的公共就绪队列可按照单处理机系统中所采用的各种方式加以组织； 其调度算法也可沿用单处理机系统所用的算法，亦即，很容易将单处理机环境下的调度机制移植到多处理机系统中，故它仍然是当前多处理机系统中较常用的调度方式。

其次， 只要系统中有任务，或者说只要公共就绪队列不空，就不会出现处理机空闲的情况，也不会发生处理器忙闲不均的现象，因而有利于提高单个处理器的利用率。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3) 自调度方式的缺点

((1))瓶颈问题。

(2) 低效性。

(3) 线程切换频繁。

应考虑的问题：当多个处理器试图访问和更新一个共同数据结构，则每个处理器必须仔细编程，确保两个处理器不会选择同一个进程，且进程不会从队列中丢失。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 成组调度(Gang Scheduling)方式囍

避免同一进程中的线程被调度时处理器分配过于离散，减少频繁切换。

在成组调度时，如何为应用程序分配处理器时间：

1) 面向所有应用程序平均分配处理器时间

2) 面向所有线程平均分配处理器时间

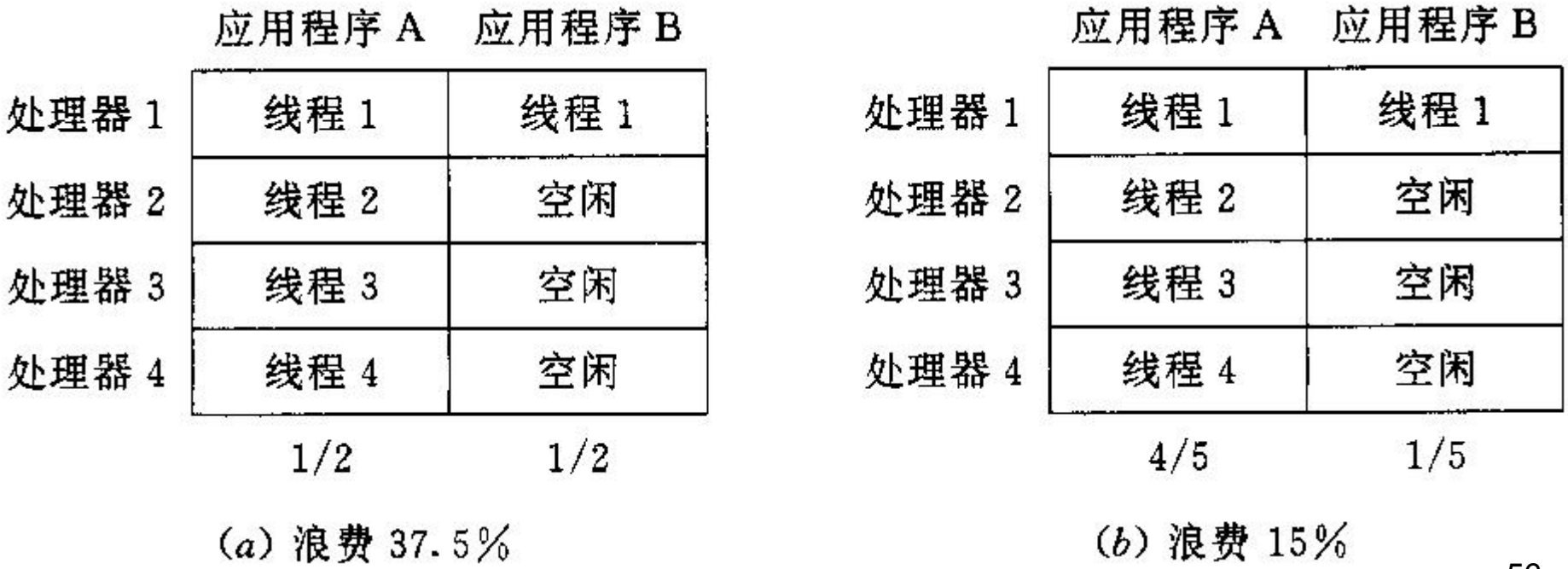


图 3 - 10 两种分配处理器时间的方法

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

成组调度的优点

- 对一组合作的进程或线程可并行执行，则能有效减少进程阻塞的发生，减少线程的切换；
- 每次调度都可以解决一组线程的处理器分配问题，减少调度频率。降低系统开销。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3. 专用处理器分配(Dedicated Processor Assigement)方式

◆实验证明，高并发度的多处理器环境，采用专用处理器分配方式可加速用用程序执行速度。

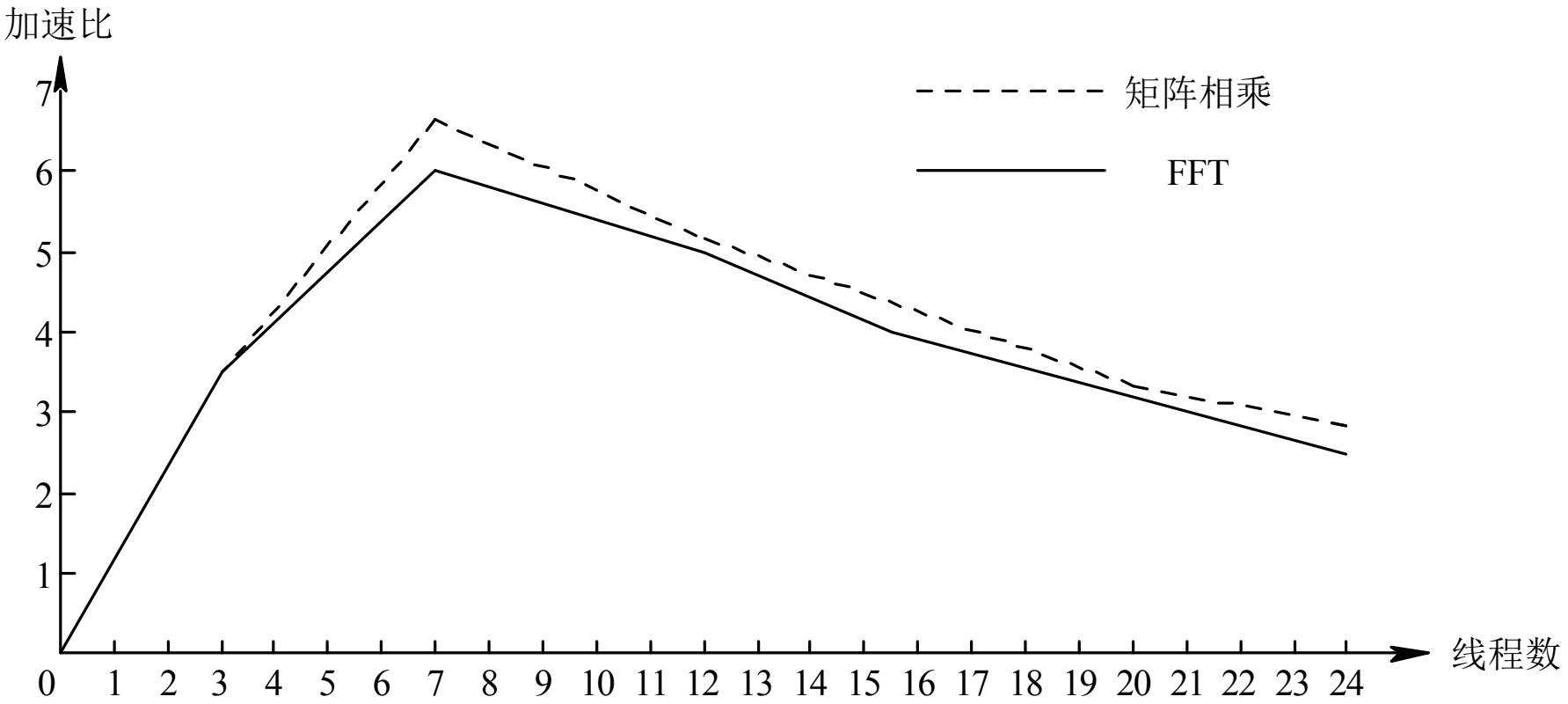


图 3-11 线程数对加速比的影响

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.5 产生死锁的原因和必要条件

死锁（ deadlock ）：多个进程在运行过程中因竞争资源造成系统无法推进的僵持状态。

----当一组进程的每一个进程都等待一个事件，而这一事件只能由这组进程的另一个进程引起，则这组进程就处于死锁状态。

3.5.1 产生死锁的原因

((1))竞争资源。

(2) 进程间推进顺序非法。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

- 一、竞争资源引起死锁。
  - 1)．可剥夺（CPU、内存，）和非剥夺性（打印机，磁带机）资源
  - 2)．竞争非剥夺性资源——可造成死锁

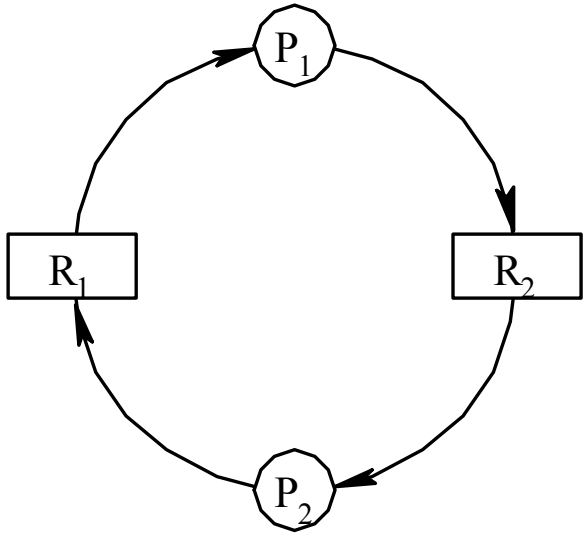


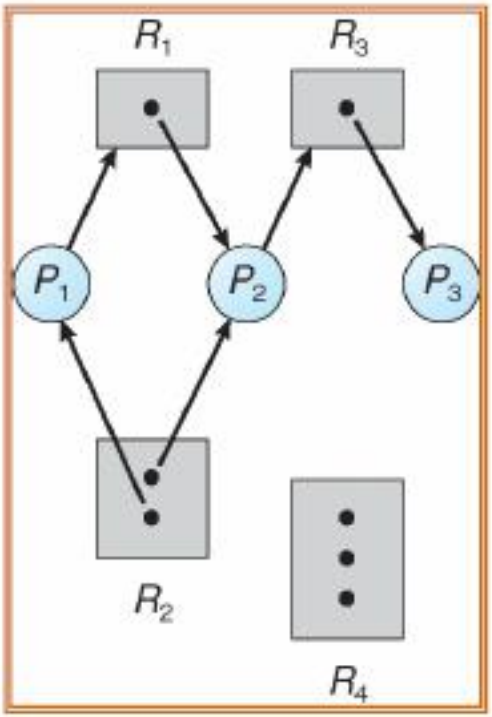
图 3-12　I/O设备共享时的死锁情况

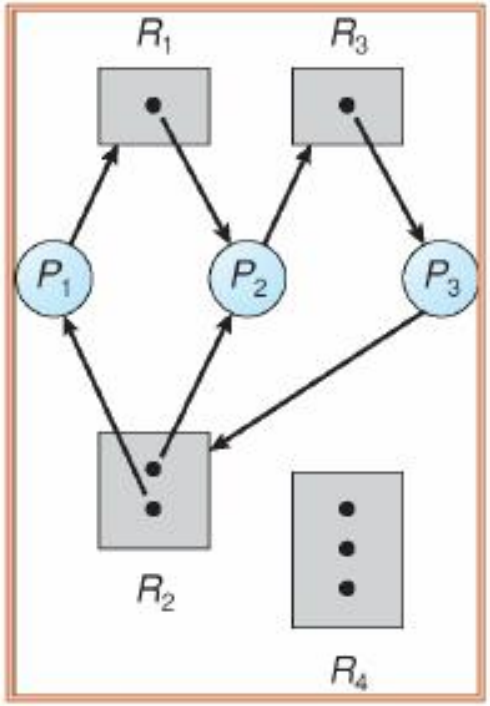
完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

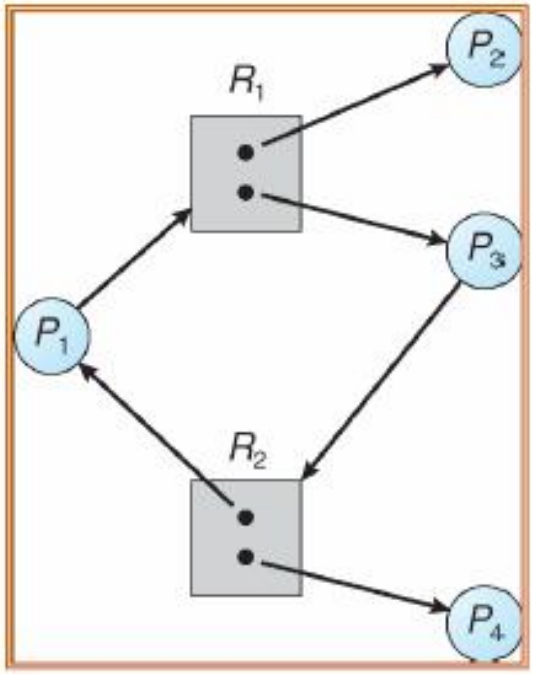
资源分配图

- 有向图G={V, E}
- 节点集V包括 P(活动进程集合),R(资源类型集合),
- 边集E可以分为请求边P→R 和分配边R→P





有死锁的资源分配图



有环但无死锁的资源分配图

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

- 3)．竞争临时性资源
  - 临时性资源是指由一个进程产生，被另一个进程使用一段时间后便无用的资源。

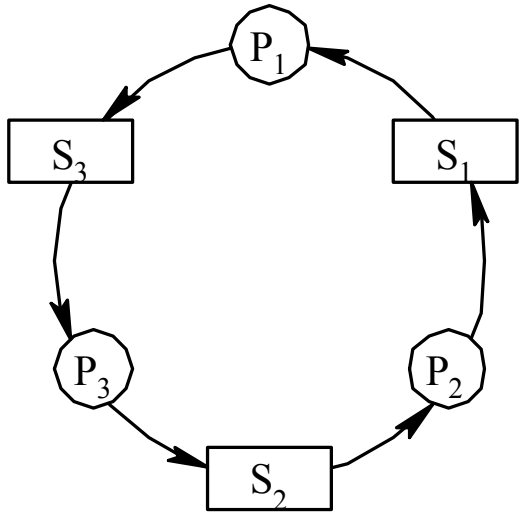


图 3-13 进程之间通信时的死锁

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 进程推进顺序不当引起死锁

1) ①②③合法的进程推进顺序

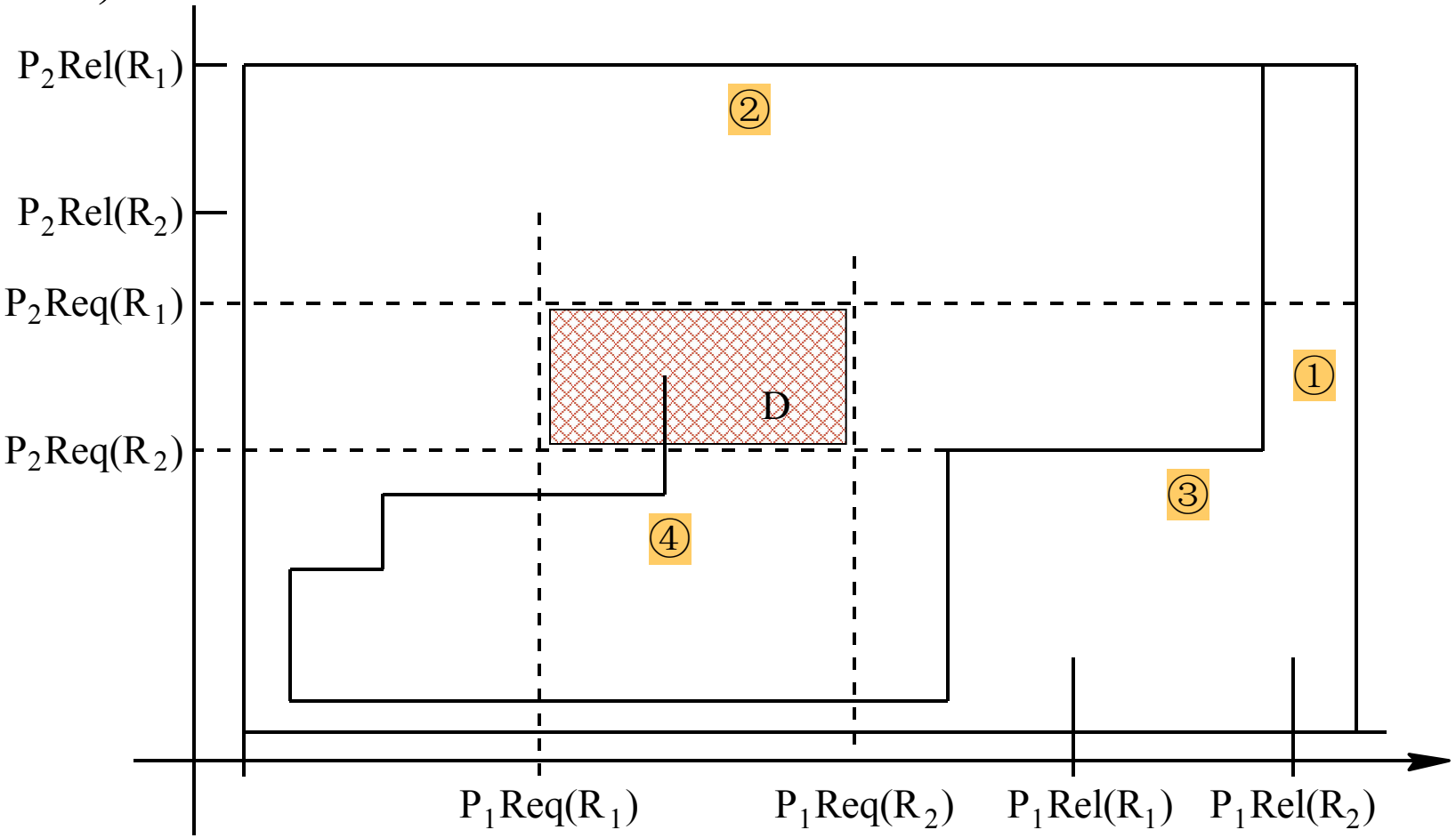


图 3-14 进程推进顺序对死锁的影响

完整版，请访问www.kaoyancas.net-科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2) ④为非法推进顺序囍

若并发进程P_1和P_2按曲线④所示的顺序推进，它们将进入不安全区D内。此时P_1保持了资源R_1, P_2保持了资源R_2, 系统处于不安全状态。因为，这时两进程再向前推进，便可能发生死锁。例如，当P_1运行到P_1:Request(R_2)时，将因R_2已被P_2占用而阻塞；当P_2运行到P_2: Request(R_1)时，也将因R_1已被P_1占用而阻塞，于是发生了进程死锁。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.5.2 产生死锁的必要条件：【若死锁，一定具有以下条件】

((1))互斥条件

(2) 请求和保持条件

(3) 不剥夺条件

(4) 环路等待条件

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

关于死锁的一些结论

- 参与死锁的进程最少是两个
  （两个以上进程才会出现死锁）
- 参与死锁的进程至少有两个已经占有资源
- 参与死锁的所有进程都在等待资源

-- 如果死锁发生，会浪费大量系统资源，甚至导致系统崩溃

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.5.3 处理死锁的基本方法

((1))预防死锁。

(2) 避免死锁。

(3) 检测死锁。

(4) 解除死锁。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.6 预防死锁的方法

- 3.6.1 死锁预防
  - 一、互斥条件是资源固有属性，不能避免。
  - 二、摒弃请求和保持条件

    全分配，全释放（AND）

    缺点：（1）延迟进程运行，”饥饿”

    （2）资源利用率低
  - 三、摒弃“不剥夺”条件 （限“可剥夺资源”）

    增加系统开销，且进程前段工作可能失效。(通常用于其状态可以保存和恢复的资源,不适用于其他如打印机磁带机等资源.)

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

预防死锁(续)

- 四、摒弃“环路”条件
  - 有序资源分配法：为资源编号，申请时需按编号进行。
  - 缺点：
  - （1）新增资源不便，（原序号已排定）
  - （2）用户不自由
  - （3）资源与进程使用顺序不同造成浪费

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.6.2 系统安全状态

- 在“避免死锁”方法中的判断条件
- 1. 安全状态
  - 按某种顺序并发进程都能达到获得最大资源而顺序完成的序列为安全序列。
  - 能找到安全序列的状态为安全状态。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

预防死锁(续)

2. 安全状态例题

我们通过一个例子来说明安全性。假定系统中有三个进程P_1、P_2和P_3，共有12台磁带机。进程P_1总共要求10台磁带机，P_2和P_3分别要求4台和9台。假设在T_0时刻，进程P_1、P_2和P_3已分别获得5台、2台和2台磁带机，尚有3台空闲未分配，如下表所示：

进 程	最 大 需 求	已 分 配	可 用
P_1	10	5	3
P_2	4	2	
P_3	9	2	

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

预防死锁(续)

- 3 安全——不安全状态的转换
  - 上例中，若P3再申请一台，则不安全

进程	最大需求	已分配	可用
P1	10	5	2
P2	4	2	
P3	9	3	

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.6.3 利用银行家算法避免死锁

1. 银行家算法中的数据结构

((1)) 可利用资源向量Available。 available[j]=k: 系统现有Rj类资源k个；

((2)) 最大需求矩阵Max。 max[i,j]=k: 进程i需要Rj的最大数k个；

(3) 分配矩阵Allocation。 alloc[i,j]=k: 进程i已得到Rj类资源k个； 囍

(4) 需求矩阵Need。need[i,j]=k: 进程i需要Rj类资源k个.有：

$$Need[i,j] = Max[i,j] - Allocation[i,j]$$

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 银行家算法

设$Request_i$是进程P_i的请求向量，如果$Request_i$［j］=*K*，表示进程P_i需要*K*个R_j类型的资源。当P_i发出资源请求后，系统按下述步骤进行检查：

囍

(1) 如果$Request_i$［j］≤Need［i,j］，便转向步骤2；否则认为出错，因为它所需要的资源数已超过它所宣布的最大值。

(2) 如果$Request_i$［j］≤Available［j］，便转向步骤(3)；否则， 表示尚无足够资源，P_i须等待。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(3) 系统试探着把资源分配给进程P_i，并修改下面数据结构中的数值：

Available［j］：=Available［j］-$Request_i$［j］;囍

Allocation［i,j］：=Allocation［i,j］+$Request_i$［j］;囍

Need［i,j］：=Need［i,j］- $Request_i$［j］;囍

(4) 系统执行安全性算法，**检查此次资源分配后，系统是否处于安全状态**。若安全，才正式将资源分配给进程P_i，以完成本次分配；否则，将本次的试探分配作废，恢复原来的资源分配状态，让进程P_i等待。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3. 安全性算法

(1) 设置两个向量：

① 工作向量Work: 它表示系统可提供给进程继续运行所需的各类资源数目，它含有*m*个元素，在执行安全算法开始时，Work：=Available;

② Finish:布尔向量，它表示系统是否有足够的资源分配给进程，使之运行完成。开始时先做Finish［i］：=false; 当有足够资源分配给进程时，再令Finish［i］：=true。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(2) 从进程集合中找到一个能满足下述条件的进程： 囍

① Finish［i］=false;

② Need［i,j］≤Work［j］（所有j）； 若找到， 执行步骤(3)， 否则，执行步骤(4)。琇囍

(3) 当进程P_i获得资源后，可顺利执行，直至完成，并释放出分配给它的资源，故应执行：囍

Work［j］：=琇Work［i］+Allocation［i,j］;囍

Finish［i］：=琇true;囍

go to step 2;

(4) 如果所有进程的Finish［i］=true都满足， 则表示系统处于安全状态；否则，系统处于不安全状态。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

4. 银行家算法例

假定系统中有五个进程{P_0, P_1, P_2, P_3, P_4}和三类资源{A, B, C}，各种资源的数量分别为*10*、*5*、*7*，在T_0时刻的资源分配情况如图 3-15 所示。

进程 \ 资源情况	Max A	Max B	Max C	Allocation A	Allocation B	Allocation C	Need A	Need B	Need C	Available A	Available B	Available C
P_0	7	5	3	0	1	0	7	4	3	3	3	2
										(2	3	0)
P_1	3	2	2	2	0	0	1	2	2			
				(3	0	2)	(0	2	0)			
P_2	9	0	2	3	0	2	6	0	0			
P_3	2	2	2	2	1	1	0	1	1			
P_4	4	3	3	0	0	2	4	3	1			

图 3-15　T_0时刻的资源分配表

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(1) 鋍T_0时刻的安全性：

进程 \ 资源情况	Work A	Work B	Work C	Need A	Need B	Need C	Allocation A	Allocation B	Allocation C	Work+Allocation A	Work+Allocation B	Work+Allocation C	Finish
P_1	3	3	2	1	2	2	2	0	0	5	3	2	true
P_3	5	3	2	0	1	1	2	1	1	7	4	3	true
P_4	7	4	3	4	3	1	0	0	2	7	4	5	true
P_2	7	4	5	6	0	0	3	0	2	10	4	7	true
P_0	10	4	7	7	4	3	0	1	0	10	5	7	true

图 3-16 T_0时刻的安全序列

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(2) P_1请求资源：P_1发出请求向量$Request_1(1，0，2)$，系统按银行家算法进行检查：

① $Request_1(1, 0, 2) \leq Need_1(1, 2, 2)$囍

② $Request_1(1, 0, 2) \leq Available_1(3, 3, 2)$囍

③ 系统先假定可为P_1分配资源，并修改Available, $Allocation_1$和$Need_1$向量，由此形成的资源变化情况如图 3-15 中的圆括号所示。

④ 再利用安全性算法检查此时系统是否安全。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

进程 \ 资源情况	Work A	Work B	Work C	Need A	Need B	Need C	Allocation A	Allocation B	Allocation C	Work+Allocation A	Work+Allocation B	Work+Allocation C	Finish
P_1	2	3	0	0	2	0	3	0	2	5	3	2	true
P_3	5	3	2	0	1	1	2	1	1	7	4	3	true
P_4	7	4	3	4	3	1	0	0	2	7	4	5	true
P_0	7	4	5	7	4	3	0	1	0	7	5	5	true
P_2	7	5	5	6	0	0	3	0	2	10	5	7	true

图 3-17 P_1申请资源时的安全性检查

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(3) P_4请求资源：P_4发出请求向量$Request_4(3，3，0)$，系统按银行家算法进行检查：囍

① $Request_4(3, 3, 0) \leq Need_4(4, 3, 1)$;囍

② $Request_4(3, 3, 0) <$ $Available(2, 3, 0)$，让P_4等待。囍

(4) P_0请求资源： P_0发出请求向量$Requst_0(0，2，0)$，系统按银行家算法进行检查：囍

① $Request_0(0, 2, 0) \leq Need_0(7, 4, 3)$;囍

② $Request_0(0, 2, 0) \leq Available(2, 3, 0)$;囍

③ 系统暂时先假定可为P_0分配资源，并修改有关数据，如图 3-18 所示。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

进程 \ 资源情况	Allocation A B C	Need A B C	Available A B C
P_0	0 3 0	7 2 3	2 1 0
P_1	3 0 2	0 2 0	
P_2	3 0 2	6 0 0	
P_3	2 1 1	0 1 1	
P_4	0 0 2	4 3 1	

图 3-18 为P_0分配资源后的有关资源数据

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.7 死锁的检测与解除

3.7.1 死锁的检测

1. 资源分配图(Resource Allocation Graph)

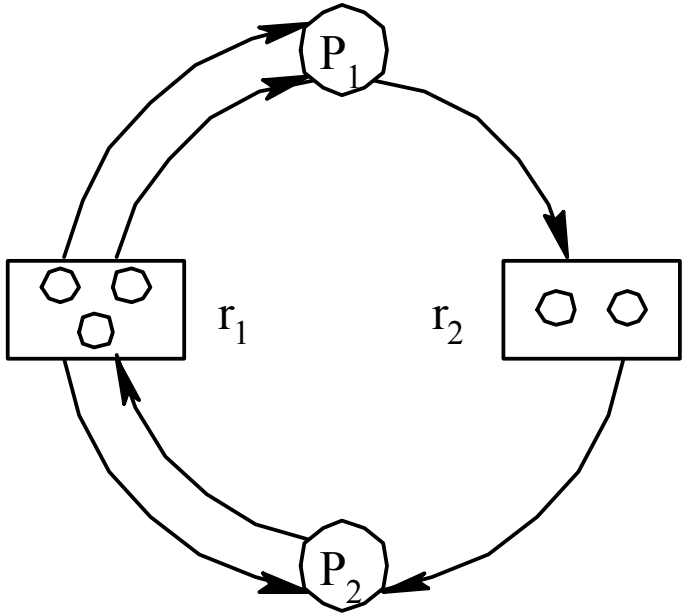


图 3-19 每类资源有多个时的情况

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(2) 凡属于E中的一个边e∈E，都连接着P中的一个结点和R中的一个结点，$e=\{p_i, r_j\}$是资源请求边，由进程p_i指向资源r_j，它表示进程p_i请求一个单位的r_j资源。$e=\{r_j, p_i\}$是资源分配边，由资源r_j指向进程p_i，它表示把一个单位的资源r_j分配给进程p_i。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

2. 死锁定理： 状态S为死锁状态的充分条件是，当且仅当S状态的资源分配图是不可完全化简的。

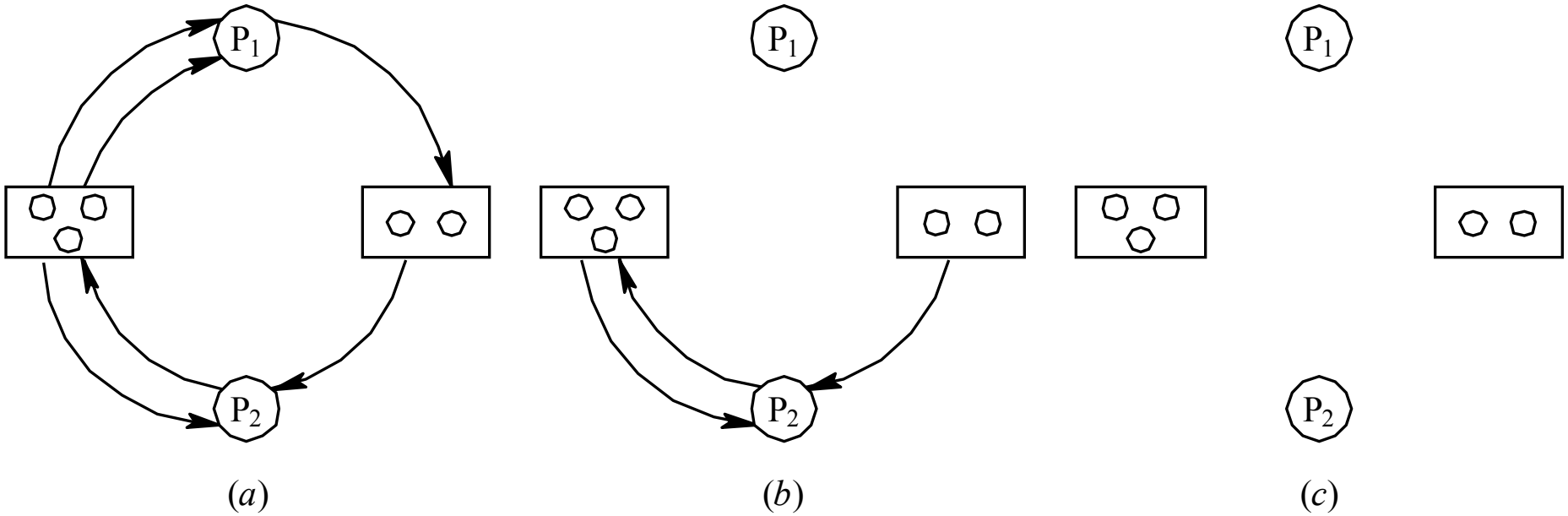


图 3-20 资源分配图的化简

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3. 死锁检测中的数据结构

(1) 可利用资源向量Available，它表示了*m*类资源中每一类资源的可用数目。囂

(2) 把不占用资源的进程(向量Allocation=0)记入L表中， 即$L_i \cup L$。

(3) 重复从进程集合中找到一个$Request_i \leq Work$的进程，做如下处理：

① 将其资源分配图简化，释放出资源，增加工作向量Work：=$Work+Allocation_i$。

② 将它记入L表中。

(4) 若不能把所有进程都记入L表中， 便表明系统状态S的资源分配图是不可完全简化的。 因此，该系统状态将发生死锁。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

算法描述：

Work： =Available;囍

L坱： =珎{L_i|$Allocation_i$=0∩$Request_i$=0}囍

```
for all Li 笑∉ L do囍
  begin囍
     for all Requesti≤Work do囍
        begin囍
            Work坱： =Work+Allocationi;囍
            Li∪L;囍
        end囍
    end囍
```

deadlock： = (L={p_1, p_2, …, p_n});

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

3.7.2 死锁的解除

((1))剥夺资源。

(2) 撤消进程。

- 撤销所有进程 ——简单有效，代价过高。
- 一次仅终止一个进程直到消除死锁循环。

- 撤消进程的策略：
  a) **树的宽度优先搜索算法**。这种算法不太实际。
  b) 一个比较有效的方法为**最短路径的优先策略搜索算法**。

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

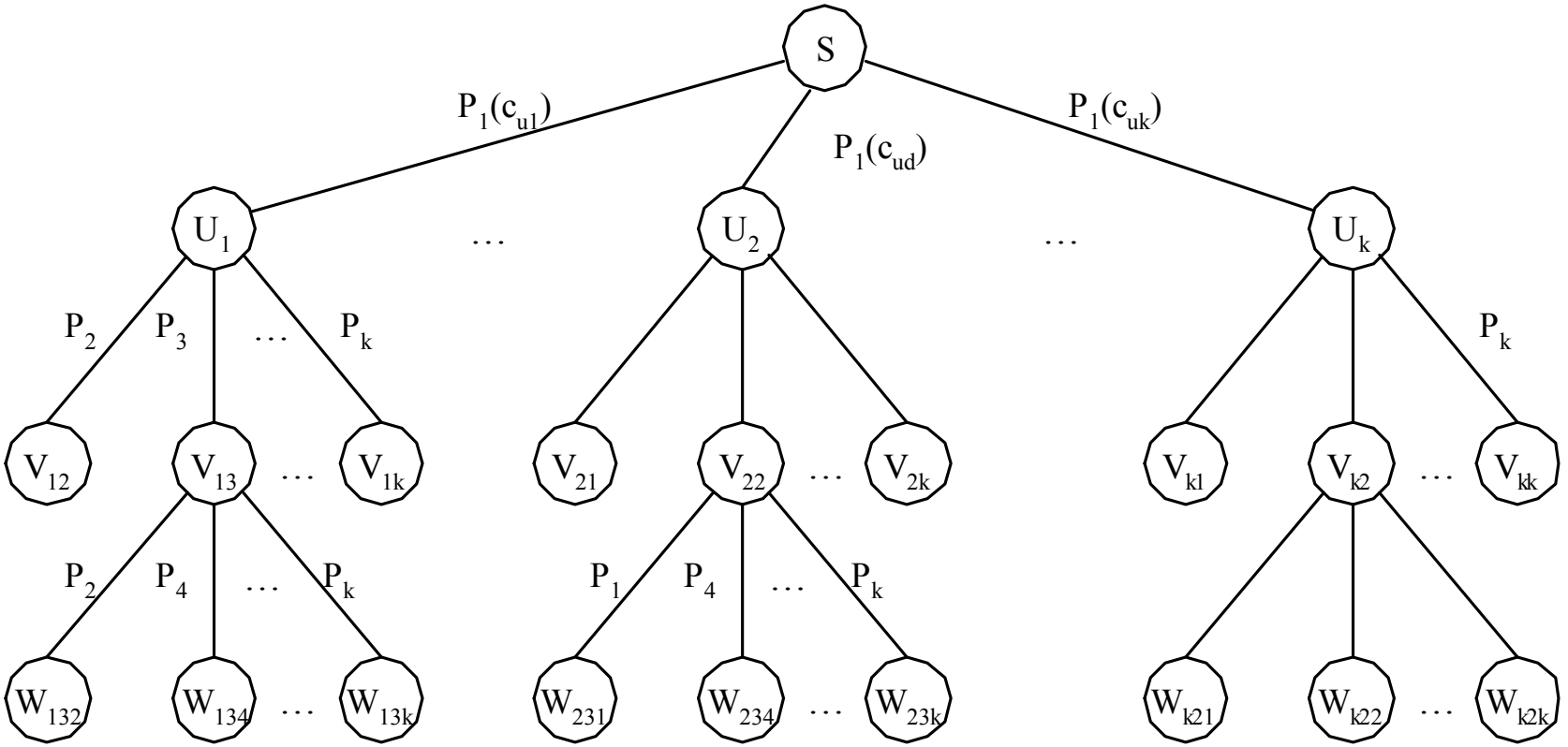


图 3-21 付出代价最小的死锁解除方法

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

- **练习：**有三类资源A(17)、B(5)、C(20)。有5个进程P1—P5。T0时刻系统状态如下：

	最大需求	已分配
P1	5 5 9	2 1 2
P2	5 3 6	4 0 2
P3	4 0 11	4 0 5
P4	4 2 5	2 0 4
P5	4 2 4	3 1 4

问(1)、T0时刻是否为安全状态，给出安全系列。

(2)、T0时刻，P2：Request(0,3,4)，能否分配，为什么？

(3)、在(2)的基础上P4：Request(2,0,1)，能否分配，为什么？

(4)、 在(3)的基础上P1：Request(0,2,0)，能否分配，为什么？

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

解：(1) T0时刻的出安全系列

先求出Need和Work

	最大需求	已分配	Need
P1	5 5 9	2 1 2	3 4 7
P2	5 3 6	4 0 2	1 3 4
P3	4 0 11	4 0 5	0 0 6
P4	4 2 5	2 0 4	2 2 1
P5	4 2 4	3 1 4	1 1 0

A(17)、B(5)、C(20)

Work=2 3 3

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

Work=2 3 3

	Work	Allocation	Need	W+A	Finish
P5	2 3 3	3 1 4	1 1 0	5 4 7	T
P4	5 4 7	2 0 4	2 2 1	7 4 11	T
P3	7 4 11	4 0 5	0 0 6	11 4 16	T
P2	11 4 16	4 0 2	1 3 4	15 4 18	T
P1	15 4 18	2 1 2	3 4 7	17 5 20	T

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(2) P2: Request(0,3,4)

- 因（ Available =2 3 3）< Request(0,3,4) 所以不能分配

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(3) P4：Request(2,0,1) Available =2 3 3

	Allocation	Need	Available
P1	2 1 2	3 4 7	0 3 2
P2	4 0 2	1 3 4	
P3	4 0 5	0 0 6	
P4	4 0 5	0 2 0	
P5	3 1 4	1 1 0	

有安全序列**P4 P5 P3 P2 P1**可以分配

	Work	Allocation	Need	W+A	Finish
P4	0 3 2	4 0 5	0 2 0	4 3 7	T
P5	4 3 7	3 1 4	1 1 0	7 4 11	T
P3	7 4 11	4 0 5	0 0 6	11 4 16	T
P2	11 4 16	4 0 2	1 3 4	15 4 18	T
P1	15 4 18	2 1 2	3 4 7	17 5 20	T

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

(4) P1：Request(0,2,0)

	Allocation	Need	Available
P1	2 3 2	3 2 7	0 1 2
P2	4 0 2	1 3 4	
P3	4 0 5	0 0 6	
P4	4 0 5	0 2 0	
P5	3 1 4	1 1 0	

0 1 2 已不能满足任何进程的需要，不能分配

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研

高参考价值的真题、答案、学长笔记、辅导班课程，访问：www.kaoyancas.net

第三章 作业

- page101
  - 3. 6. 16. 20

完整版，请访问www.kaoyancas.net 科大科院考研网，专注于中科大、中科院考研