



操作系统课程 重点解析

2022~2023第二学期
郑州轻工业大学 软件学院

为之则易
不为之则难



● 重点1：操作系统的目标、发展过程和基本特性

● 操作系统的目标（P2）

- 1. 方便性的含义
- 2. 有效性的含义
- 3. 可扩充性的含义
- 4. 开放性的含义

● 操作系统的基本特性（P15~P18）

- 1. 并发的含义
- 2. 共享的含义
- 3. 虚拟的含义
- 4. 异步的含义

● 操作系统的发展过程（P5~P15）

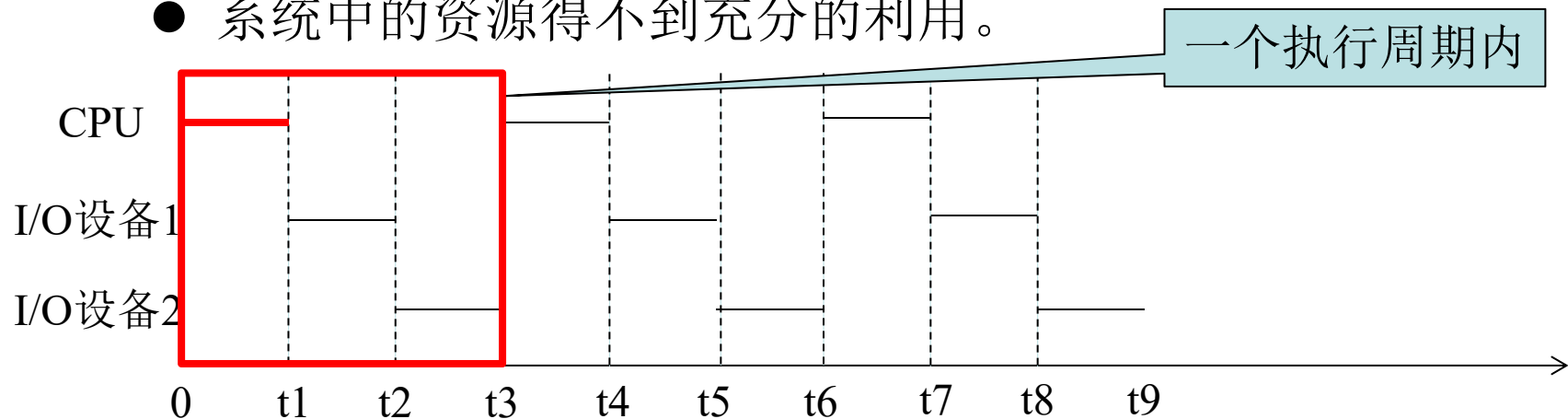
- 1. 未配置操作系统的计算机系统
- 2. 单道批处理系统（基本特征,下同）
- 3. 多道批处理系统
- 4. 分时系统
- 5. 实时系统
- 6. 微机操作系统
- 6. 嵌入式系统
- 7. 网络 and 分布式操作系统



● 重点2：单道批处理系统和多道批处理系统

● 单道批处理系统的运行情况及缺点（P7）

- 内存中仅能装入一道程序；
- 仅当当前作业执行完毕之后才能执行下一道作业；
- 难以发挥系统中各类型资源的并行处理能力；
- 系统中的资源得不到充分的利用。



● 计算系统中某资源的利用率

$$\text{CPU 的利用率} = \frac{\text{CPU 有效工作时间}}{\text{CPU 有效工作时间} + \text{CPU 空闲等待时间}} = \frac{t_1}{t_3} = \frac{t_4 - t_3}{t_6 - t_3}$$

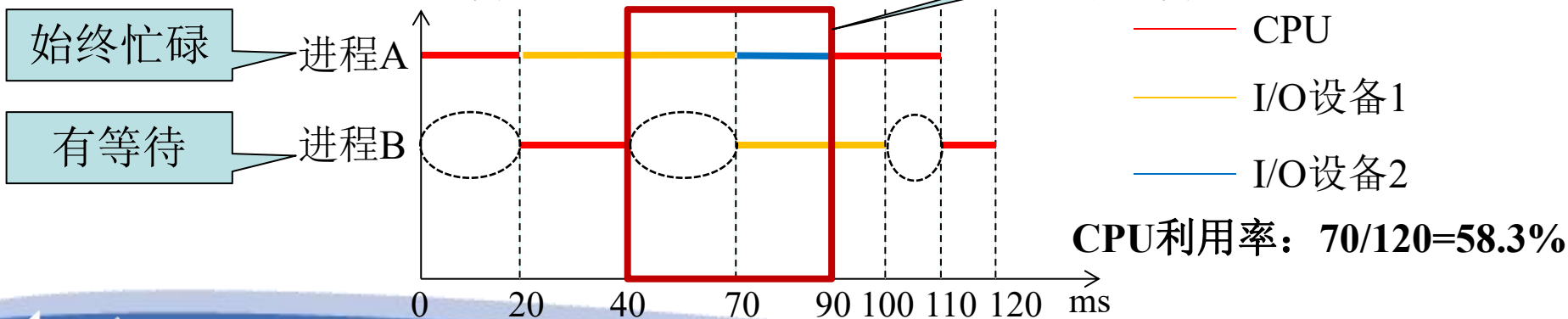
● 重点2：单道批处理系统和多道批处理系统

● 多道批处理系统的运行情况（P8）

- 内存中可同时装入多道程序，共享CPU和系统的各种资源；
- 能够充分发挥系统中各类型资源的并行处理能力；
- 多道程序交替运行，保持CPU处于忙碌状态。

● 能够在图中画出多道批处理系统的运行情况（P8）

- 设某计算机系统有CPU、I/O设备1、I/O设备2。现有两个进程同时进入就绪态，且进程A先运行，进程B后运行。
- 进程A：计算20ms，设备1运行50ms，设备2运行20ms，再计算20ms。
- 进程B：计算20ms，设备1运行30ms，再计算10ms。





- 重点3：操作系统的运行环境
 - 操作系统的启动过程（P19）
 - 操作系统内核（P19~P20）
 - 处理机的双重工作模式（P20~P21）
- 重点4：微内核操作系统
 - 微内核OS的基本概念（P30~P31）
 - 微内核的基本功能（P31）
 - 微内核OS的优点（P32）
 - 微内核OS存在的问题（P32~P33）

● 重点5：前趋图

● 根据程序代码画出前趋图（P42）

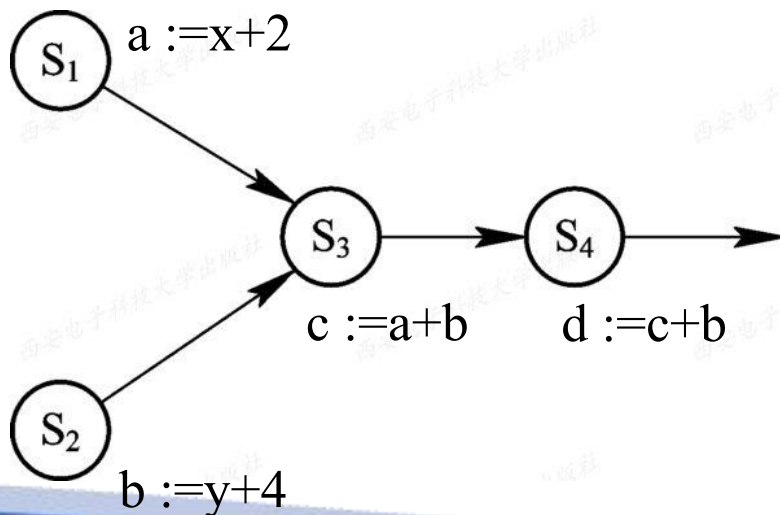
● 对于具有下述四条语句的程序段

$S_1: a := x + 2$ $S_2: b := y + 4$ $S_3: c := a + b$ $S_4: d := c + b$

如何使用前趋图表示它们的执行顺序？

● 解析：

- S_3 必须在 a 和 b 被赋值后方能执行；
- S_4 必须在 S_3 之后执行；
- S_1 和 S_2 则可以并发执行，因为它们彼此互不依赖。





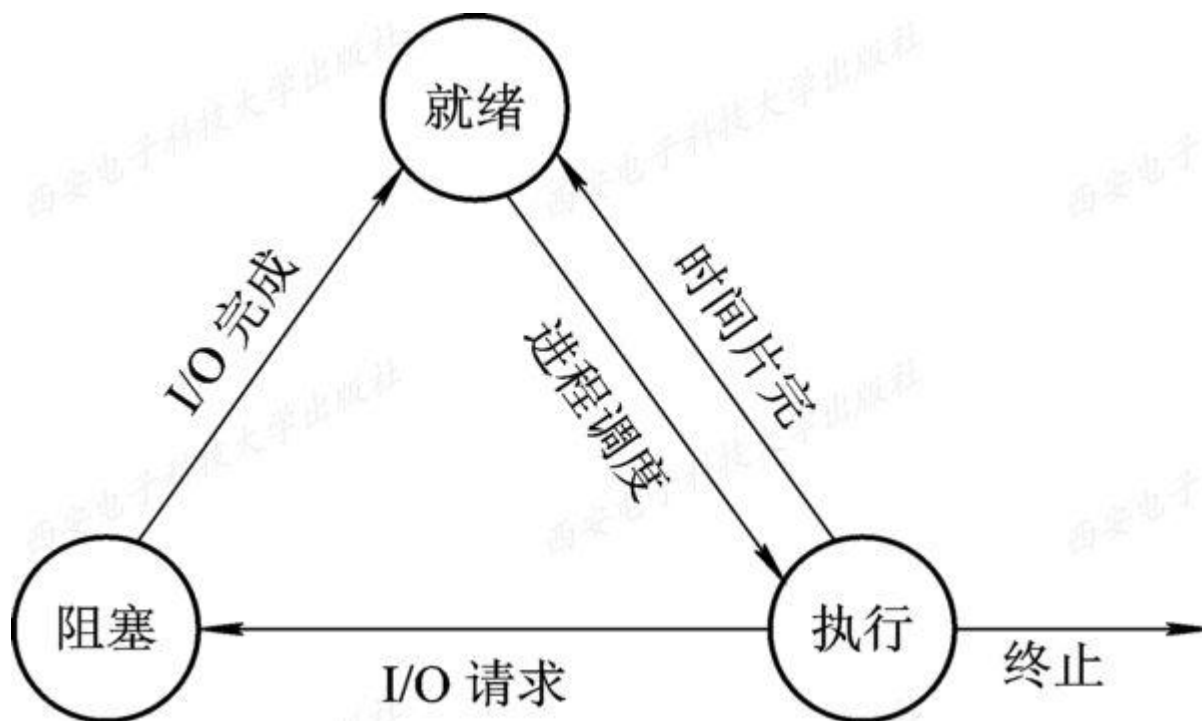
● 重点6：进程的描述

- 进程的定义（P43）
- 进程的特征（P43~P44）
- 进程控制块的作用（P48~P49）
- 进程控制块中的信息（P49）
- 进程控制块的组织方式（P49~P50）



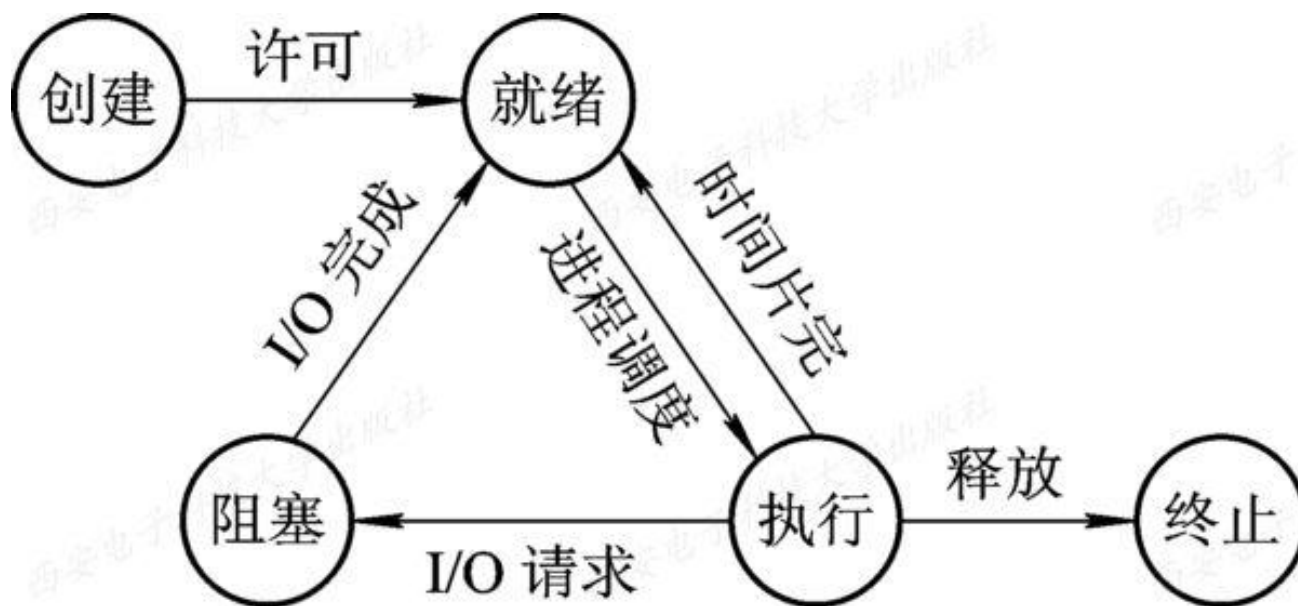
● 重点6：进程的描述

● 进程的三种基本状态及其转换（P45）



● 重点6：进程的描述

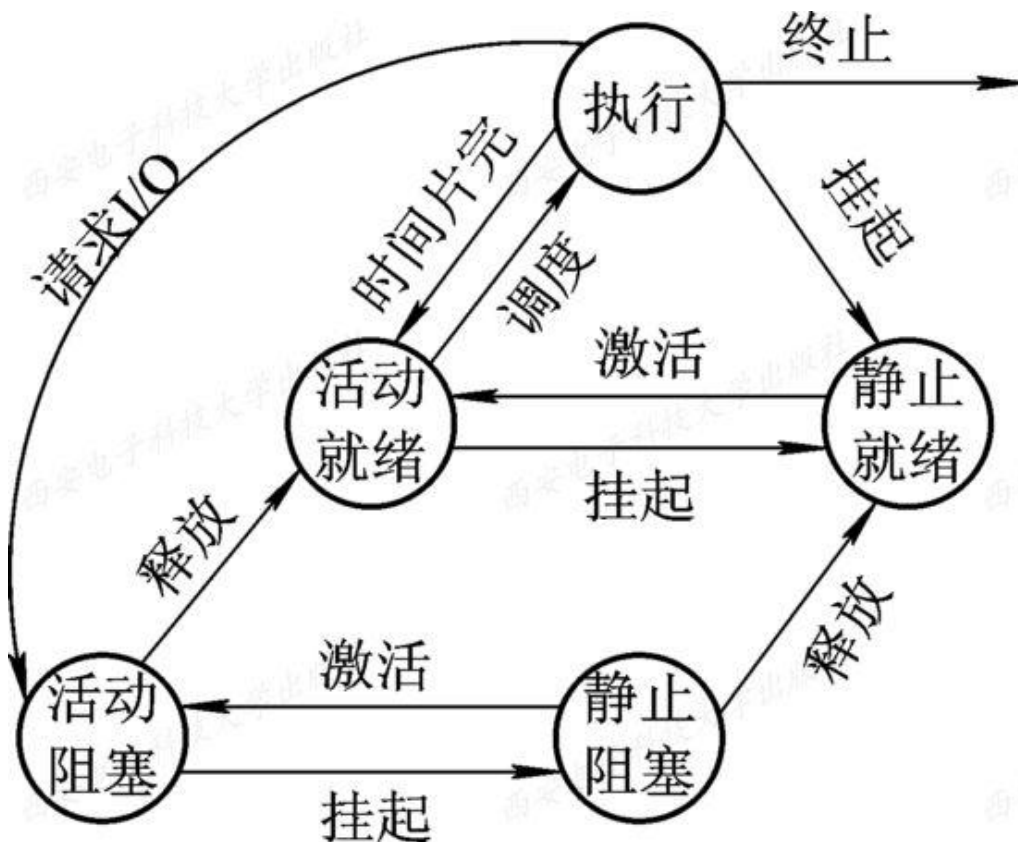
● 进程的五种基本状态及其转换（P45）



● 进程创建的步骤（P52）

● 重点6：进程的描述

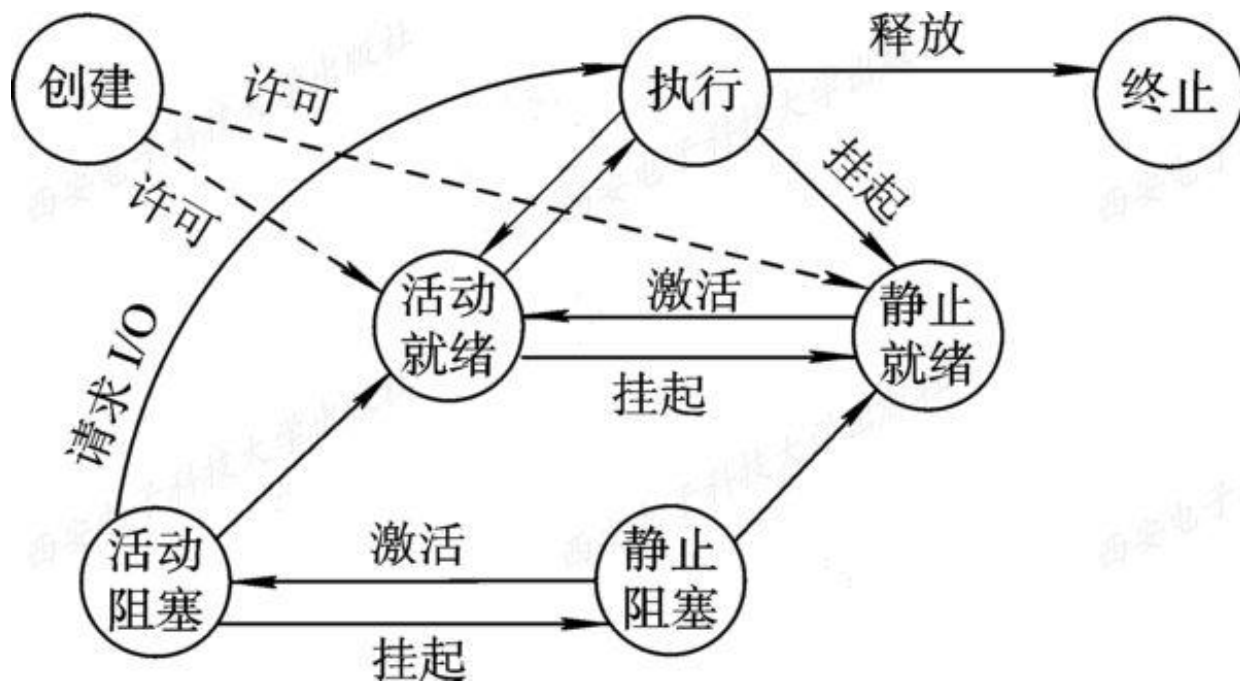
- 进程具有挂起操作的状态转换图（P46）



- 进程挂起的原因（P46）

● 重点6：进程的描述

- 进程具有创建、终止和挂起操作的状态转换图（P47）





● 重点7：进程调度（P75~P82）

- 进程调度的任务（P75~P76）
- 进程调度方式（P76~P77）
- 例题：4个任务P1，P2，P3，P4几乎同时到达，预期运行时间分别为4，6，8，10个时间单位。各个任务的优先级分别为2，1，3，4，数值越大，优先级越高。请按下列调度算法计算任务的平均周转时间（进程切换开销可忽略不计）。另外考虑带权周转时间
 - （1）先来先服务（按照P1，P2，P3，P4 的顺序）；
 - （2）时间片轮转算法，假定时间片大小为2个时间单位；
 - （3）高优先级优先调度算法。





● 重点7：进程调度（P75~P82）

● 先来先服务

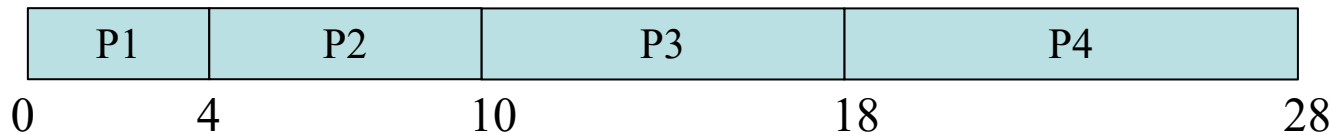
● 执行顺序、完成时间及周转时间

执行次序	运行时间	优先数	等待时间	周转时间
P1	4	2	0	4
P2	6	1	4	10
P3	8	3	10	18
P4	10	4	18	28

● 4个进程的平均周转时间T为：

$$T = (4 + 10 + 18 + 28) / 4 = 15$$

先来先服务调度算法的调度顺序





● 重点7：进程调度（P75~P82）

● 时间片轮转算法

● 执行顺序、完成时间及周转时间

执行次序	运行时间	优先数	周转时间
P1	4	2	10
P2	6	1	18
P3	8	3	24
P4	10	4	28

● 4个进程的平均周转时间T为：

$$T = (10 + 18 + 24 + 28) / 4 = 20$$

时间片轮转算法的调度顺序

P1	P2	P3	P4	P1	P2	P3	P4	P2	P3	P4	P3	P4	P4	
0	2	4	6	8	10	12	14	16	18	20	22	24	26	28



第三章处理机调度与死锁

● 重点7：进程调度（P75~P82）

● 高优先级优先调度算法

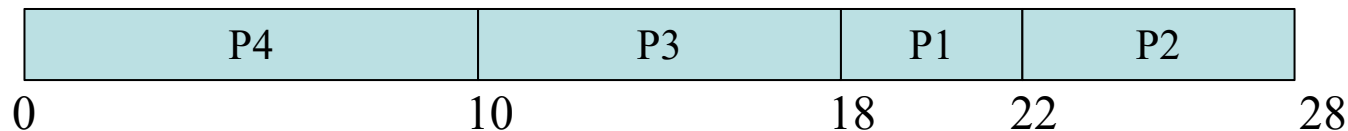
● 执行顺序、完成时间及周转时间

执行次序	运行时间	优先数	等待时间	周转时间
P4	10	4	0	10
P3	8	3	10	18
P1	4	2	18	22
P2	6	1	22	28

● 4个进程的平均周转时间T为：

$$T = (10 + 18 + 22 + 28) / 4 = 19.5$$

高优先级优先调度算法的调度顺序





第三章处理机调度与死锁

● 重点8：银行家算法（P100~P102）

- 设系统中有五个进程{P₀, P₁, P₂, P₃, P₄}和三类资源{A, B, C}，资源数量分别为10、5、7。

若在T₀时刻的资源分配情况如下：

	<u>Allocation</u>			<u>Max</u>			<u>Available</u>		
	A	B	C	A	B	C	A	B	C
P ₀	0	1	0	7	5	3	3	3	2
P ₁	2	0	0	3	2	2			
P ₂	3	0	2	9	0	2			
P ₃	2	1	1	2	2	2			
P ₄	0	0	2	4	3	3			

- 1) T₀时刻的状态是否安全？为什么？
- 2) P₁请求资源(1, 0, 2)，系统能够将资源分配给它？
- 3) 此后，P₀发出资源请求 (0, 2, 0)，系统能否为P₀分配资源？





第三章处理机调度与死锁

● 重点8：银行家算法（P100~P102）

- 1) 系统在T0时刻的资源分配情况

$$\text{Need} = \text{Max} - \text{Allocation}$$

资源 进程	Max A B C	Alloc. A B C	Need A B C	Avail.
P ₀	7 5 3	0 1 0	5th	10 5 7
P ₁	3 2 2	2 0 0	1st	
P ₂	9 0 2	3 0 2	4th	
P ₃	2 2 2	2 1 1	2nd	
P ₄	4 3 3	0 0 2	3rd	

- 系统处于安全状态，因存在安全序列 $\langle P_1, P_3, P_4, P_2, P_0 \rangle$

资源 进程	Work A B C	Need A B C	Alloc A B C	Work+Alloc A B 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



● 重点8：银行家算法（P100~P102）

2) P_1 发出资源请求 (1, 0, 2)时，系统按银行家算法进行检查：

- ① $Request_1(1, 0, 2) \leq Need_1(1, 2, 2)$;
- ② $Request_1(1, 0, 2) \leq Available(3, 3, 2)$;
- ③ 系统试探性地为 P_1 分配资源，修改数据为：

进程 \ 资源	Max	Alloc.	Need	Avail.
	A B C	A B C	A B C	
P_0	7 5 3	0 1 0	7 4 3	2 3 1
P_1	3 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	

- ④进行安全性检查
 - 可找到一个安全序列 $\langle P_1, P_3, P_4, P_2, P_0 \rangle$ ，所以系统可以将 P_1 申请的资源分配给它。

● 重点8：银行家算法（P100~P102）

3) P_0 发出资源请求 $(0, 2, 0)$ 时，系统按银行家算法进行检查：

- ◆ ① $Request_0(0, 2, 0) \leq Need_0(7, 4, 3)$;
- ◆ ② $Request_0(0, 2, 0) \leq Available(2, 3, 0)$;
- ◆ ③ 系统试探性地为 P_0 分配资源，修改数据为：

资源 情况 进 程	Allocation			Need			Available		
	A	B	C	A	B	C	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			

- ◆ ④进行安全性检查

- ◆ 可用资源 $Available(2, 1, 0)$ 已不能满足任何进程的需要，故系统进入不安全状态，此时系统不分配资源。



● 重点9：死锁定理（P103）

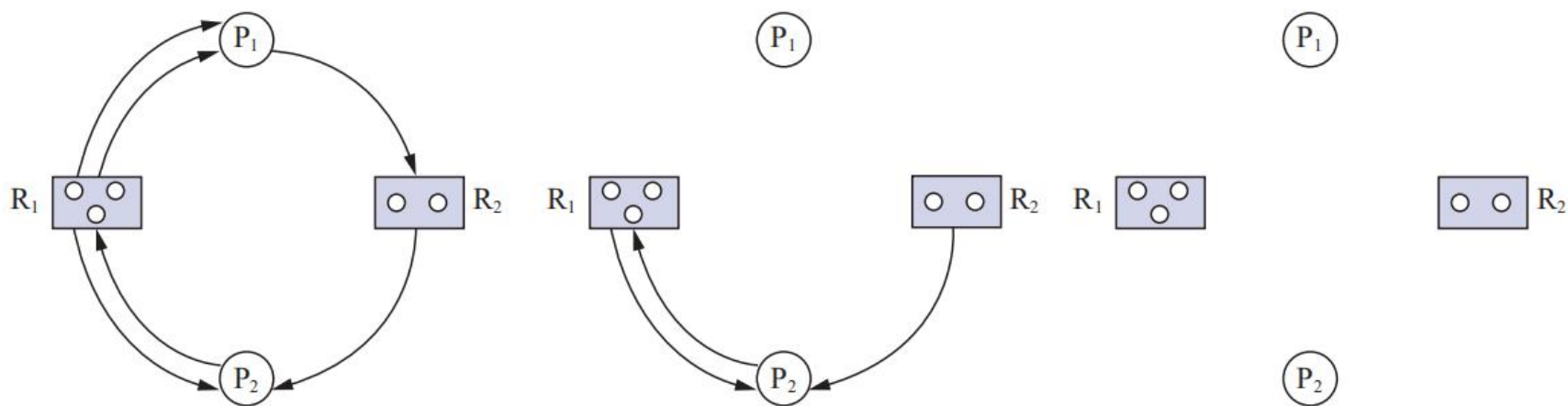
- 死锁定理：S为死锁状态的充分条件是当且仅当S状态的资源分配图是**不可完全简化**的。
- 可通过**简化资源分配图**的方式来判断当前系统是否处于死锁状态：
 - 在资源分配图中，**找出一个既不阻塞又非独立的进程结点 p_i** 。消去 p_i 所有的请求边和分配边，使之成为孤立的结点；这等于释放 p_i 占有的所有资源；
 - p_i 释放资源后，**便可使 p_2 获得资源而继续运行**，直到 p_2 完成后又释放出它所占有的全部资源；
 - 在进行一系列的简化后，**若能消去图中所有的边，使所有进程都成为孤立结点**，则称该图是**可完全简化的**；若不能通过任何过程使该图完全简化，则称该图是**不可完全简化的**。



● 重点9：死锁定理（P103）

● 资源分配图简化的例子

- 系统中共有3个资源 R_1 和2个资源 R_2 。
- 进程 P_1 占有两个资源 R_1 ，请求一个资源 R_2 ；进程 P_2 占有一个资源 R_1 和一个资源 R_2 ，请求一个资源 R_1 。



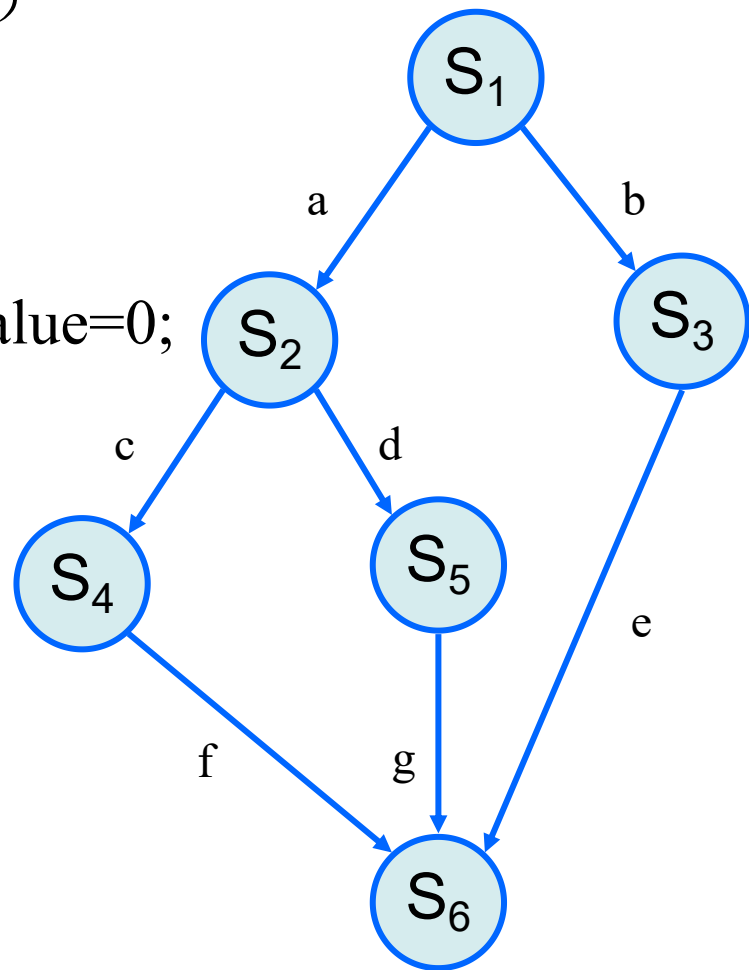
结论：资源分配图是可完全化简的，当前系统没有发生死锁。

● 重点10：记录型信号量机制

- 利用信号量实现前趋关系（P119）

main()

```
{ Semaphore a,b,c,d,e,f,g;  
  a.value=0;b.value=0;c.value=0;  
  d.value=0;e.value=0;f.value=0;g.value=0;  
  cobegin  
    { S1;signal(a);signal(b); }  
    { wait(a);S2;signal(c) ;signal(d);}  
    { wait(b);S3;signal(e); }  
    { wait(c);S4;signal(f); }  
    { wait(d);S5;signal(g); }  
    { wait(e);wait(f);wait(g);S6; }  
  coend
```



● 重点10：记录型信号量机制

● 共享浴室问题（P129）

- 现要设计一个共享公共浴室管理系统，该浴室每次只能有一个性别的人员进入使用，但允许同性别的多人同时使用。
- 请使用带有信号量的P、V操作（`wait()`、`signal()`操作）的伪代码描述任意数量的男士和女士使用该共享浴室的过程，并说明所用信号量及初值的含义。





● 重点10：记录型信号量机制

● 共享浴室问题（P129）

● 信号量和变量定义：

int nmen=0; //浴室中男士的数量

int nwomen=0; //浴室中女士的数量

semaphore Smen=1; //男士变量的互斥信号量

semaphore Swomen=1; //女士变量的互斥信号量

semaphore Sused=1; //浴室互斥信号量





● 重点10：记录型信号量机制

● 共享浴室问题（P129）

● 核心代码：

```
void Men() {  
    while(1){  
        wait(Smen);  
        if(nmen==0) wait(Sused);  
        nmen++;  
        signal(Smen);  
        男士进入洗浴;  
        wait(Smen);  
        nmen--;  
        if (nmen==0) signal(Sused);  
        signal(Smen)  
    }  
}
```

```
void Women() {  
    while(1){  
        wait(Swomen);  
        if(nwomen==0) wait(Sused);  
        nwomen++;  
        signal(Swomen);  
        女士进入洗浴;  
        wait(Swomen);  
        nwomen--;  
        if (nwomen ==0) signal(Sused);  
        signal(Swomen)  
    }  
}  
  
void main() {  
    cobegin{  
        Men();  
        Women();  
    }  
}
```





● 重点11：动态分区分配算法（P149-P151）

- 设某计算机采用动态分区分配的存储器管理方式，其内存大小为640KB，其中低地址区域的40KB用于存放操作系统。系统从低地址区开始为用户作业分配空间。针对以下作业请求序列，说明采用首次适应算法（思考：其他算法的情况——**最坏适应**）进行内存分配和回收后，内存的最终映像。
- 作业请求序列如下：
 - 作业1申请200KB，作业2申请70KB；
 - 作业3申请150KB，作业2释放70KB；
 - 作业4申请80KB，作业3释放150KB；
 - 作业5申请100KB，作业6申请60KB；
 - 作业7申请50KB，作业6释放60KB。





● 重点11：动态分区分配算法（P149-P151）

● 作业请求序列：

- 作业1申请200KB，作业2申请70KB；
- 作业3申请150KB，作业2释放70KB；
- 作业4申请80KB，作业3释放150KB；
- 作业5申请100KB，作业6申请60KB；
- 作业7申请50KB，作业6释放60KB。

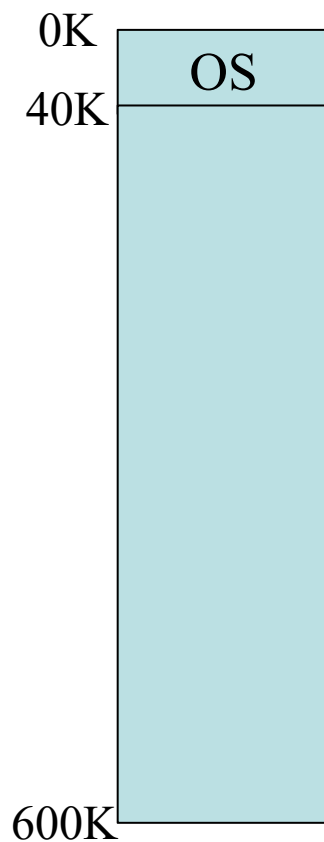
- 解析：首次适应算法的思想是把空闲分区按照地址递增的顺序组成一个链表，为进程分配内存时从链首开始查找，直至找到能容纳进程的分区。



● 重点11：动态分区分配算法（P149-P151）

内存具体分配和回收的过程：

- 初始时OS存放在0K~40K，40K~600K为空；



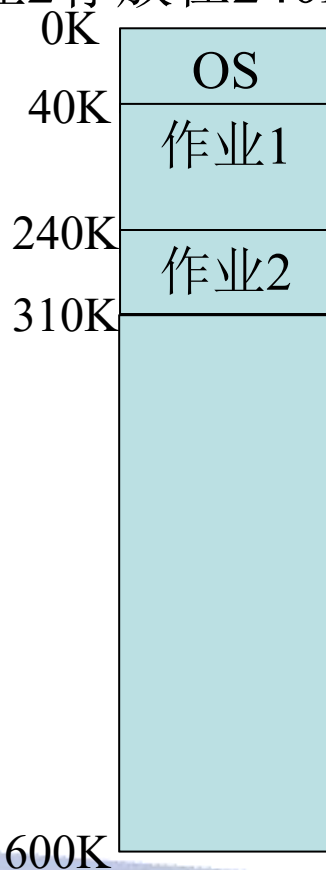


第五章存储器管理

● 重点11：动态分区分配算法（P149-P151）

内存具体分配和回收的过程：

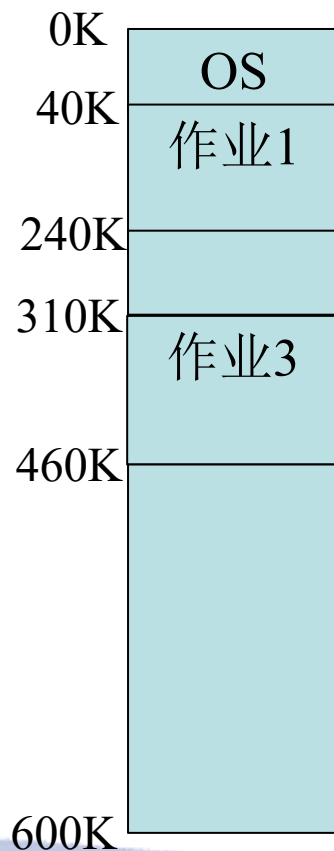
- 作业1申请200KB，作业2申请70KB：OS存放在0K~40K，作业1存放在40K~240K，作业2存放在240K~310K，310K~600K 为空；



● 重点11：动态分区分配算法（P149-P151）

内存具体分配和回收的过程：

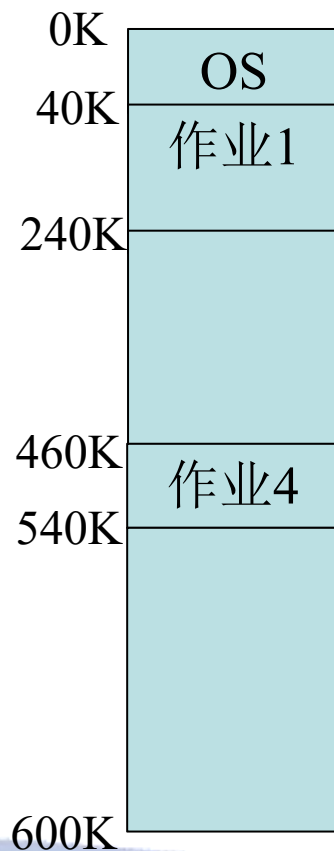
- 作业3申请150KB，作业2释放70KB：OS在0K~40K，作业1在40K~240K，240K~310K为空，作业3在310K~460K，460K~600K为空；



● 重点11：动态分区分配算法（P149-P151）

内存具体分配和回收的过程：

- 作业4申请80KB，作业3释放150KB：OS在0K~40K，作业1在40K~240K，240K~460K为空，作业4在460K~540K，540K~600K为空；

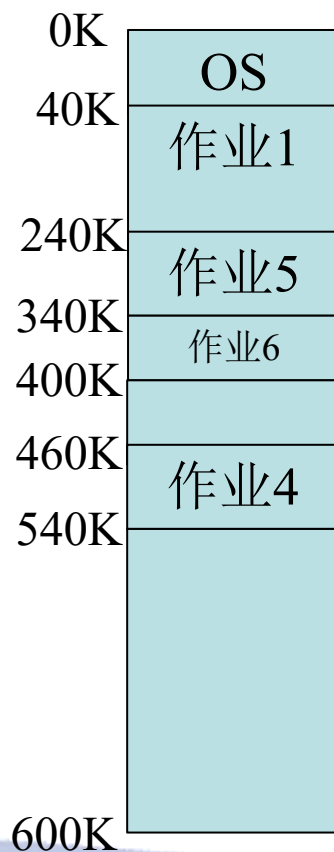




第五章存储器管理

● 重点11：动态分区分配算法（P149-P151）

- 作业5申请100KB，作业6申请60KB：OS在0K~40K，作业1在40K~240K，作业5在240K~340K，作业6在340K~400K，400K~460K为空，作业4在460K~540K，540K~600K 为空；

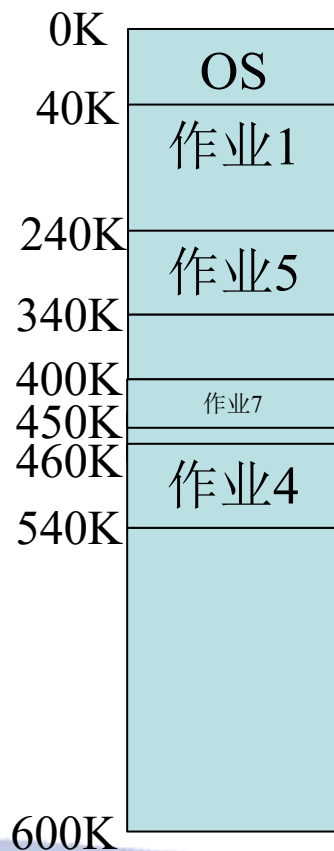




第五章存储器管理

● 重点11：动态分区分配算法（P149-P151）

- 作业7申请50KB，作业6释放60KB：OS在0K~40K，作业1在40K~240K，作业5在240K~340K，340K~400K为空，作业7在400K~450K，450K~460K 为空，作业4在460K~540K，540K~600K 为空。



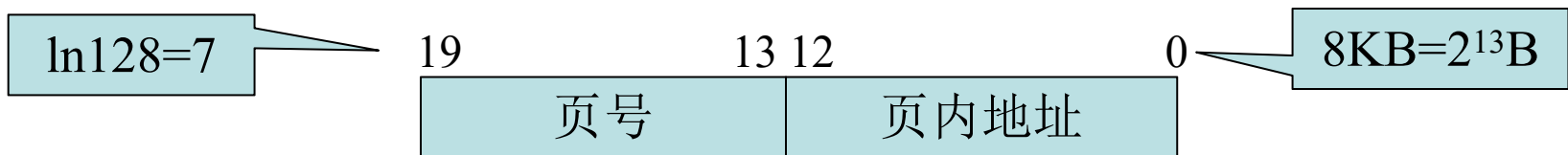


第五章存储器管理

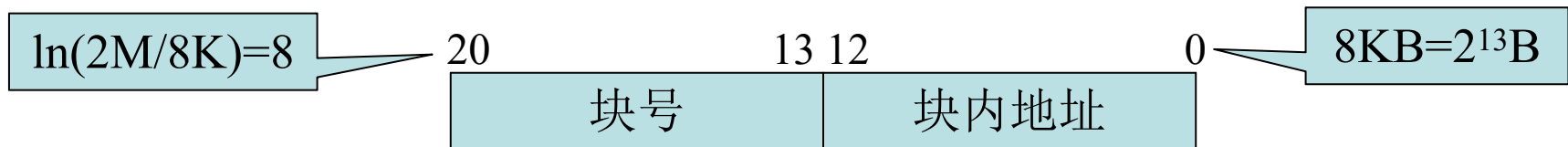
● 重点12：分页存储管理及其地址结构（P155）

- 例题：某计算机按字节编址，采用分页存储管理方式，拥有逻辑空间128页，每页8KB，拥有物理空间2MB。

(1) 设计逻辑地址格式。

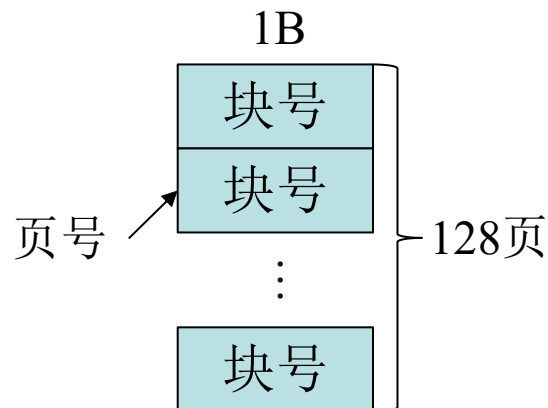


(2) 给出物理地址的格式。



(3) 求页表大小。

进程逻辑地址空间有128个页，存储器物理空间有 $2^8=256$ 个块，每个页表项大小为1B，即8位，页表大小为 $128 \times 1\text{B} = 128\text{B}$



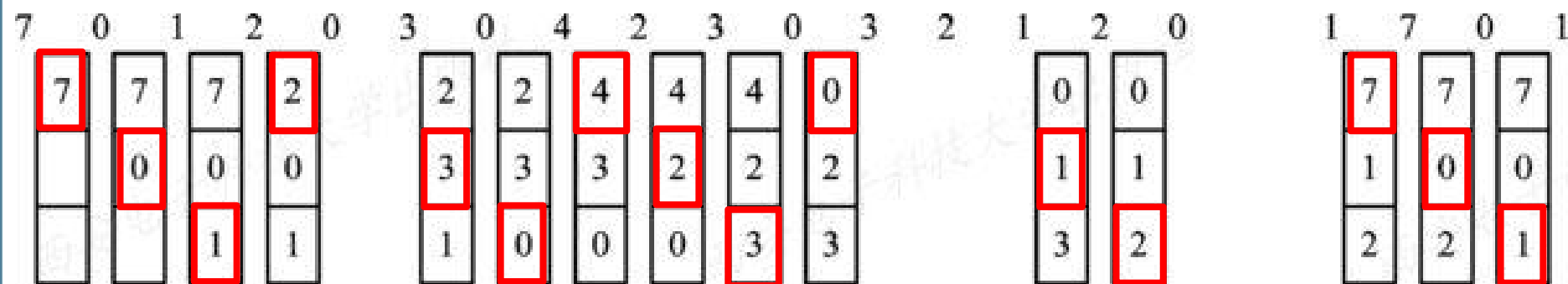


第六章虚拟存储器

● 重点13：页面置换算法（P182~P183）

● FIFO算法的例子

设系统为某进程分配3个物理块，考虑以下的页面号引用顺序：



试画出FIFO置换算法的页面置换图，并计算缺页率。

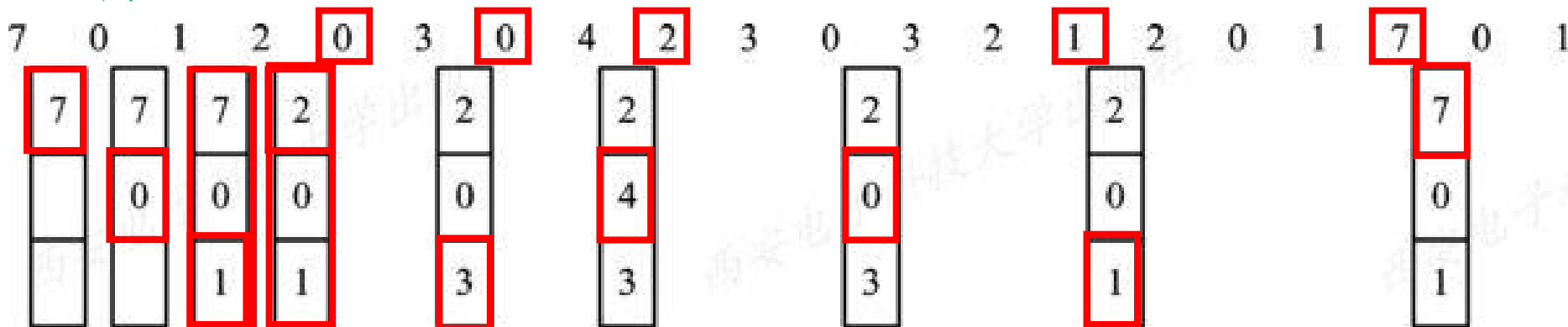
共发生15次缺页中断，缺页率 $15/20=75\%$



● 重点13：页面置换算法（P182~P183）

● 最佳(Optimal)置换算法的例子

设系统为某进程分配3个物理块，考虑以下的页面号引用顺序：



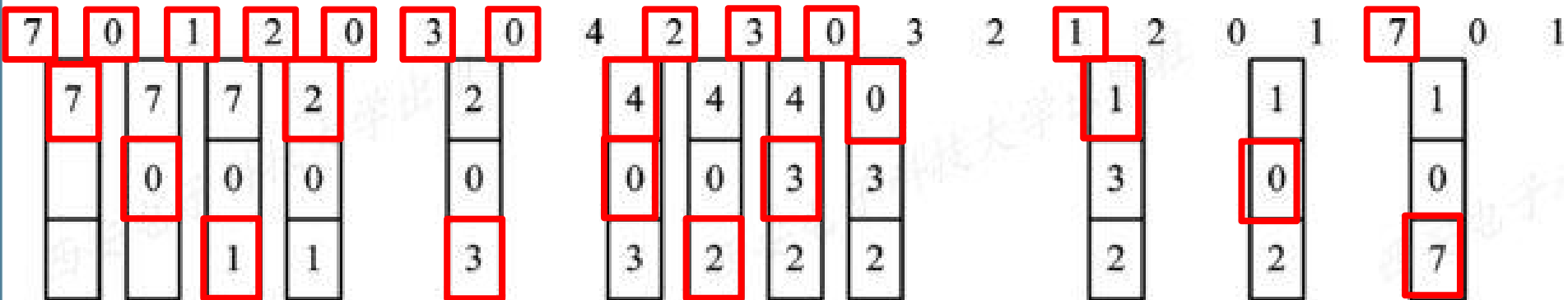
试画出最佳置换算法的页面置换图，并计算缺页率。

共发生9次缺页中断，缺页率 $9/20=45\%$

● 重点13：页面置换算法（P182~P183）

● LRU算法的例子

设系统为某进程分配3个物理块，考虑以下的页面号引用顺序：



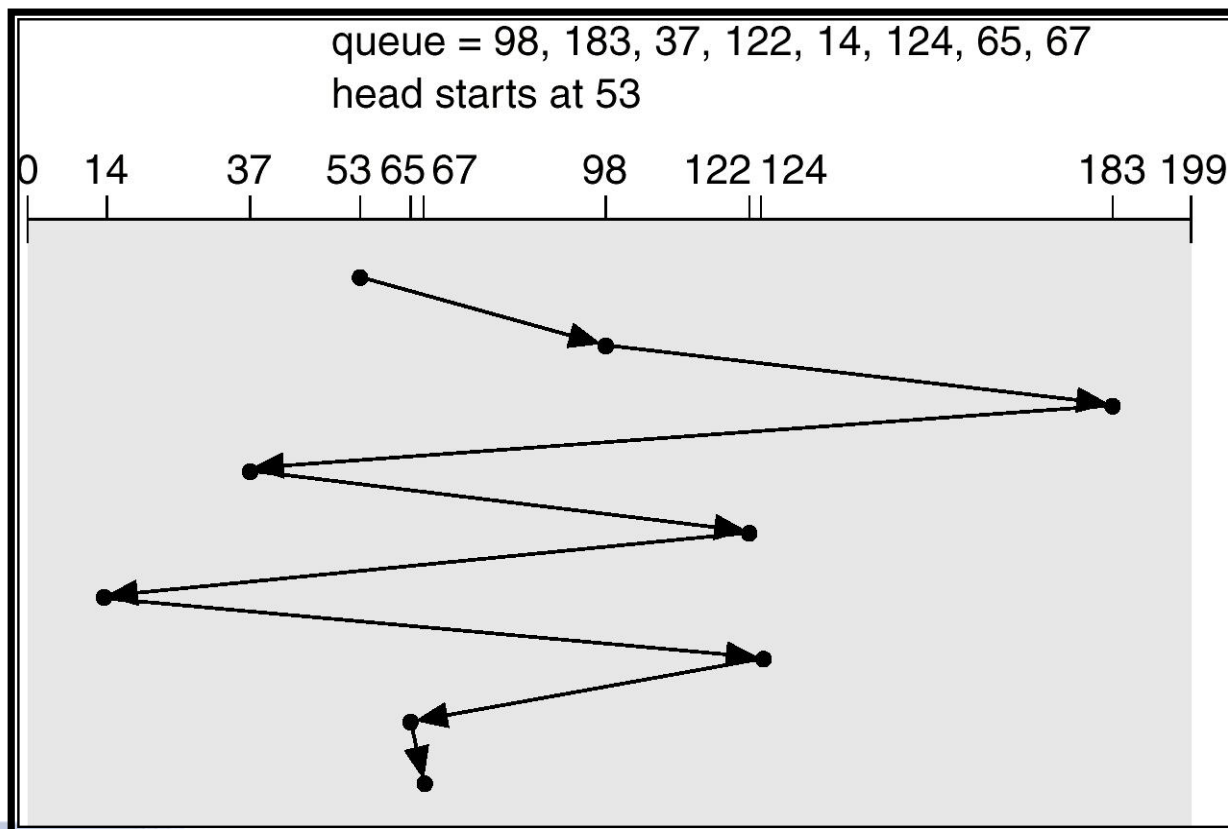
试画出LRU置换算法的页面置换图，并计算缺页率。

共发生12次缺页中断，缺页率 $12/20=60\%$

● 重点14：磁盘调度算法（P241~P243）

- 例题：给定磁道请求序列，求解FCFS、SSTF、SCAN三种算法的调度序列及其平均寻道长度。

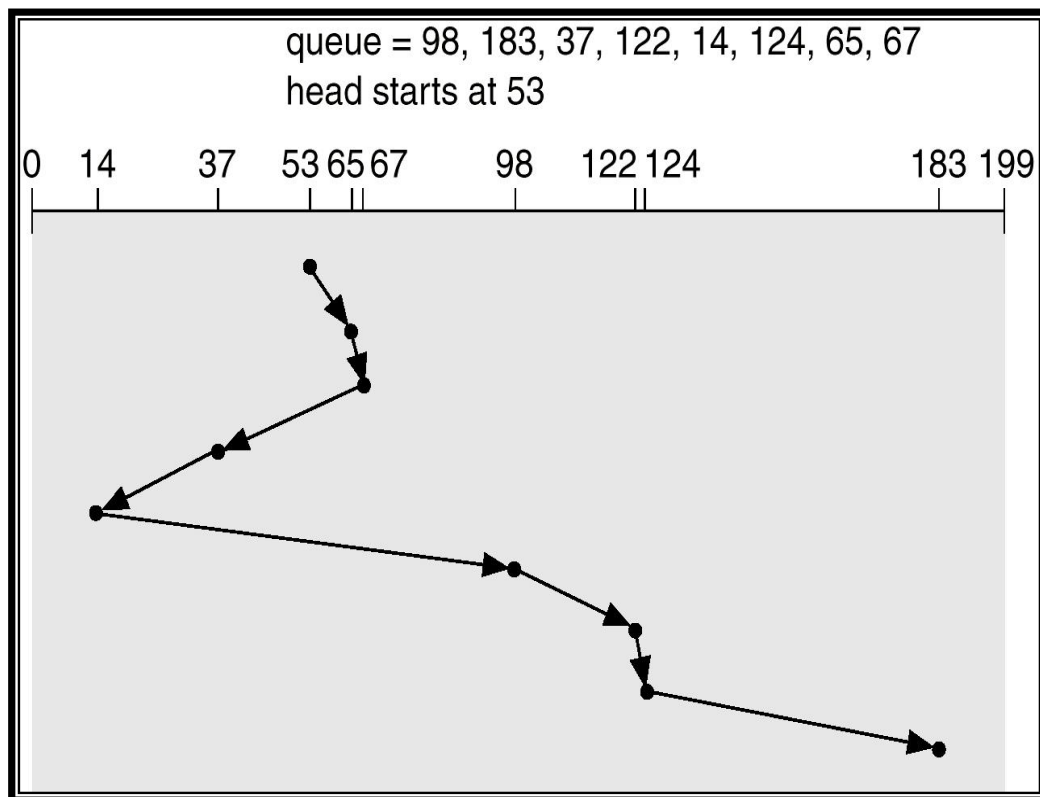
FCFS调度序列



总的磁头移动为640柱面，平均寻道长度 $640/8=80$

- 重点14：磁盘调度算法（P241~P243）
 - 例题：给定磁道请求序列，求解FCFS、SSTF、SCAN三种算法的调度序列及其平均寻道长度。

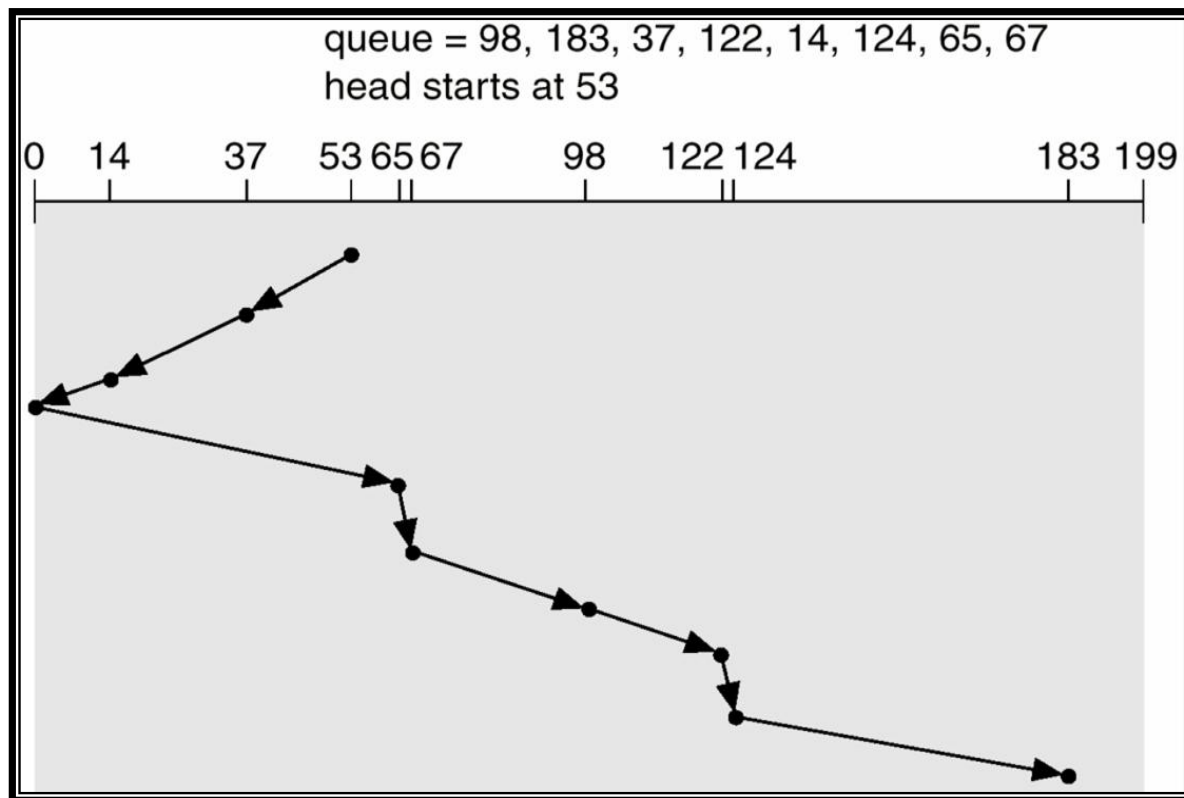
SSTF调度序列



总的磁头移动为236柱面，平均寻道长度 $236/8=29.5$

- 重点14：磁盘调度算法（P241~P243）
 - 例题：给定磁道请求序列，求解FCFS、SSTF、SCAN三种算法的调度序列及其平均寻道长度。

SCAN调度序列



总的磁头移动为236柱面，平均寻道长度 $236/8=29.5$



第八、九章文件管理及磁盘存储器的管理

- 重点15：文件目录和外存的组织方式
 - FCB和索引结点的异同（ P260~P261 ）；
 - 分别阐述连续、链式、索引3种文件的数据块组织方式的特点及优缺点（P281~P288）。

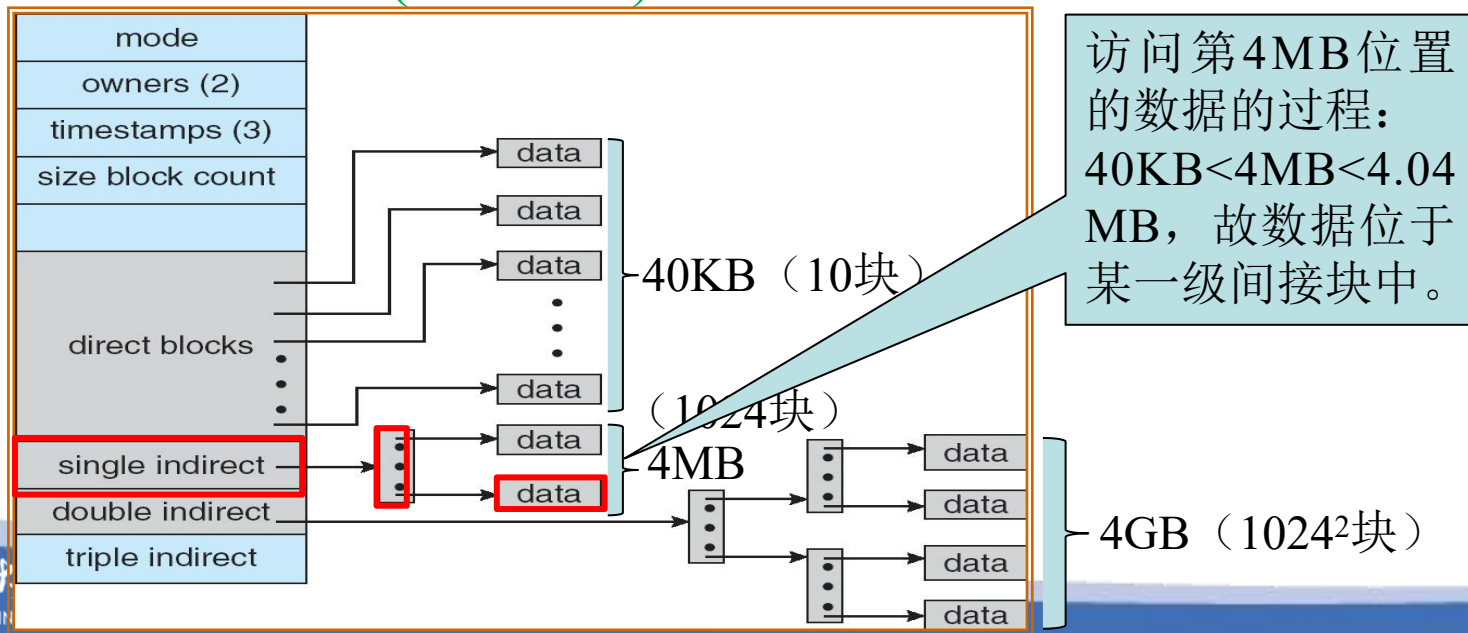




第八、九章文件管理及磁盘存储器的管理

● 重点16: 混合索引组织方式 (P287~P288)

- 例题: UNIX System V系统采用混合索引组织方式管理文件内容。设每个盘块大小为4KB, 每个索引表项为4B, 分别计算直接地址、一级间接地址和二级间接地址可寻址的文件最大长度。
- 直接地址: $10 \times 4KB = 40KB$;
- 一次间址: $40KB + (4KB/4B) \times 4KB = 4.04MB$
- 二次间址: $4.04MB + (4KB/4B)^2 \times 4KB = 4.00404GB$





第八、九章文件管理及磁盘存储器的管理

● 重点17：位示图法（P290~P291）

● 例题：若某操作系统的磁盘文件空间共有M块，若用字长为n位的位示图管理盘空间，试问：

- (1) 位示图需要多少个字？
- (2) 第i字第j位对应的块号是多少？
- (3) 给出申请、归还一块的工作流程。

- 答：(1) $\lceil M/n \rceil$ ；
- (2) $b = n \times (i-1) + j$
- (3) 流程参考课本P290~P291。

	n															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0
2	0	0	0	1	1	1	1	M 1	1	0	0	0	0	1	1	1
3	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
...	1	1														
16																





祝同学们顺利通过 考试！

郑州轻工业大学 软件学院
操作系统课程组

为之则易
不为之则难