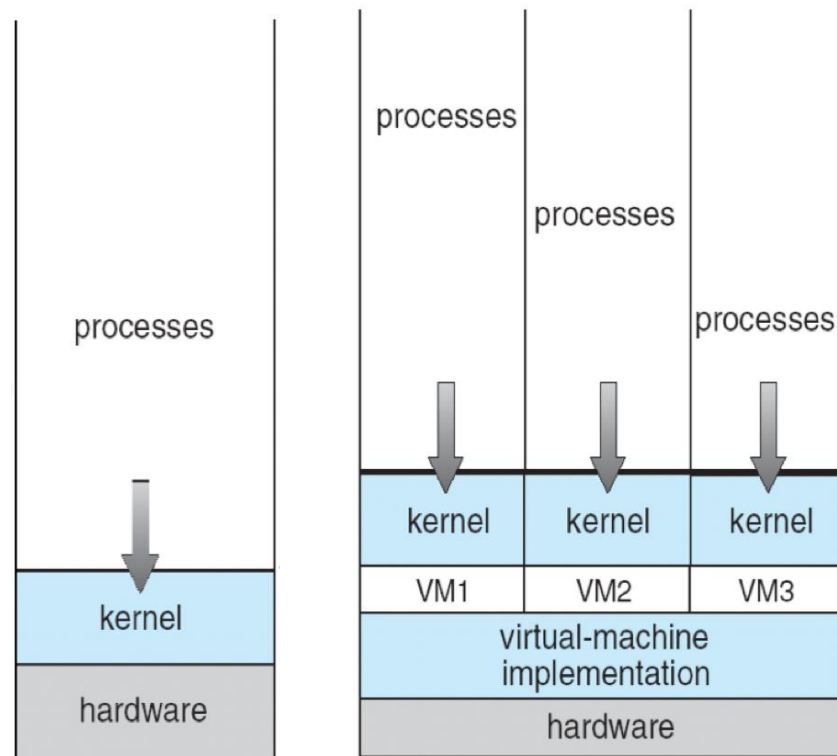


操作系统复习

课号： 102J05C

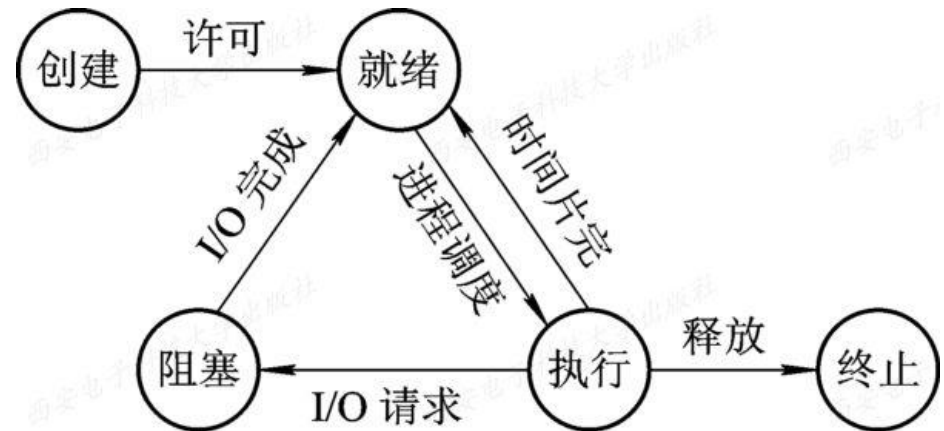
第一章

- 操作系统的目标和作用
- 操作系统的发展过程
- 操作系统的基本特性
- 操作系统的主要功能
- OS结构设计



第二章

- 前趋图和程序执行
- 进程的描述
- 进程控制
- 进程同步
- 经典进程的同步问题
- 进程通信
- 线程(Threads)的基本概念
- 线程的实现



PV: 面包师问题

- ✓ 面包师有很多面包，由 n 个销售人员推销。每个顾客进店后取一个号，并且等待叫号，当一个销售人员空闲下来时，就叫下一个号。试设计一个使销售人员和顾客同步的算法。

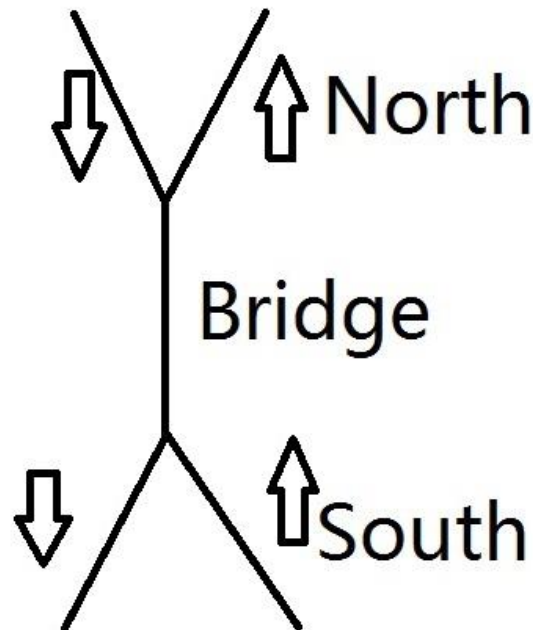
思路：顾客进店后按序取号，并等待叫号，销售人员空闲之后也是按序叫号，并销售面包。因此同步算法只要对顾客取号和销售人员进行合理的同步即可。我们使用两个变量 i 和 j 分别记录当前的取号值和叫号值，并各自使用一个互斥信号用于对 i 和 j 的访问和修改。

```
int i=0,j=0;
semaphore mutex_i=1,mutex_j=1;
Consumer() {
    //进入面包店
    p(mutex_i);
    //取号i
    i++;
    v(mutex_i);
    //等待叫号i并购买面包
}
```

```
Seller(){
    while(1){
        p(mutex_j);
        if(j<i){ //号j已有顾客取走并等待
            //叫号j
            j++;
            v(mutex_j);
            //销售面包
        }else{
            v(mutex_j);
            //休息片刻
        }
    }
}
```

PV: 独木桥问题

- ✓ 假设存在如下图的独木桥，车流方向如箭头所示，该桥上不允许两车交会，但允许同方向多个车一次通过(桥上可有多多个同方向行驶的车)。



```
int countSN=0;//表示从南到北的汽车数量
int countNS=0;//表示从北到南的汽车数量
semaphore mutexSN=1;//保护countSN
semaphore mutexNS=1;//保护countNS
semaphore bridge=1;//互斥访问桥
```

NtoS()

```
{
    P(mutexNS);
    if(countNS==0){
        P(bridge);
    }
    countNS++;
    V(mutexNS);
    //过桥
    P(mutexNS);
    countNS--;
    if(countNS==0){
        V(bridge);
    }
    V(mutexNS);
}
```

StoN()

```
{
    P(mutexSN);
    if(countSN==0){
        P(bridge);
    }
    countSN++;
    V(mutexSN);
    //过桥
    P(mutexSN);
    countSN--;
    if(countSN==0){
        V(bridge);
    }
    V(mutexSN);
}
```

PV: 课程考试问题

- ✓ 操作系统课程的期末考试即将举行，假设把学生和监考老师都看作进程，学生有 N 人，教师有1人。考场门口每次只能进出1个人，进考场的原则是先来先进。当 N 个学生都进入了考场后，教师才能发卷子。学生交卷后即可离开考场，而教师要等收上来全部卷子并封装卷子后才能离开考场。


```

semaphore S_Door; // 能否进出门, 初值1
semaphore S_StudentReady; // 学生是否到齐, 初值为0
semaphore S_ExamBegin; // 开始考试, 初值为0
semaphore S_ExamOver; // 考试结束, 初值为0
int nStudentNum = 0; // 学生数目
semaphore S_Mutex1 // 互斥信号量, 初值为1
int nPaperNum = 0; // 已交的卷子数目
semaphore S_Mutex2 // 互斥信号量, 初值为1

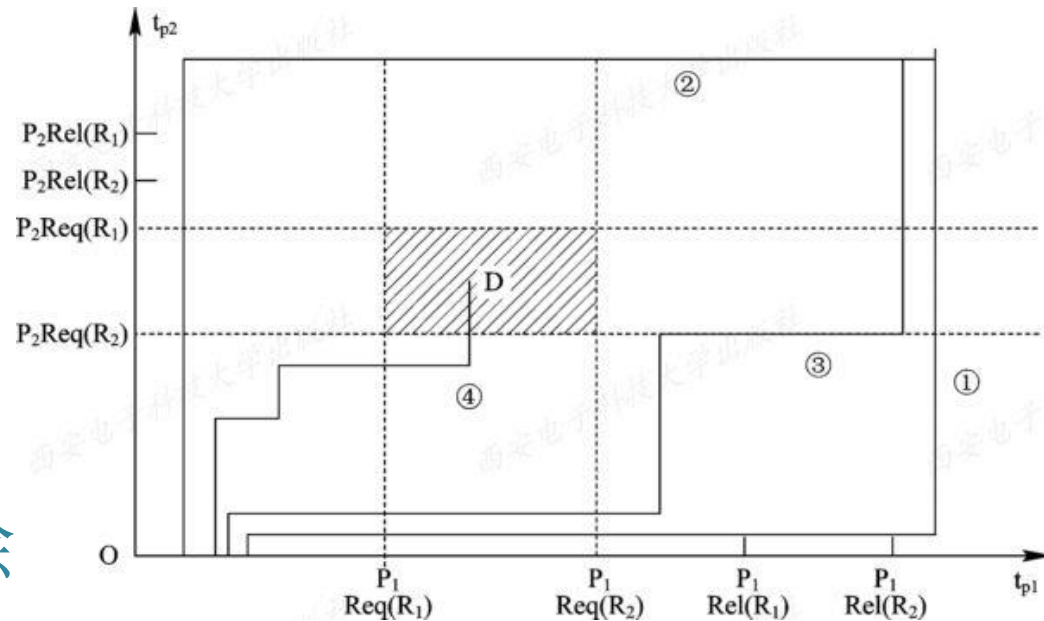
void teacher( )
{
    P(S_Door);
    进门;
    V(S_Door);
    P(S_StudentReady)
    发卷子;
    for(i = 1; i <= N; i++) V(S_ExamBegin);
    P(S_ExamOver);
    封装试卷;
    P(S_Door);
    出门;
    V(S_Door);
}

```

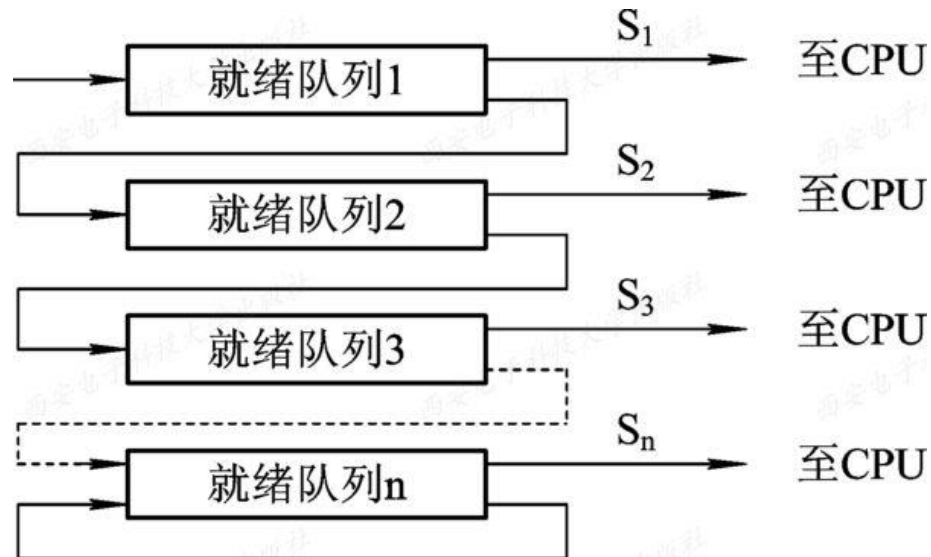
```
void student( )
{
    P(S_Door);
    进门;
    V(S_Door);
    P(S_Mutex1);
    nStudentNum ++;
    if(nStudentNum == N) V(S_StudentReady);
    V(S_Mutex1);
    P(S_ExamBegin);
    考试中...
    交卷;
    P(S_Mutex2);
    nPaperNum ++;
    if(nPaperNum == N) V(S_ExamOver);
    V(S_Mutex2);
    P(S_Door);
    出门;
    V(S_Door);
}
```

第三章

- 处理机调度的层次和调度算法的目标
- 作业与作业调度
- 进程调度
- 实时调度
- 死锁概述
- 预防死锁
- 避免死锁
- 死锁的检测与解除



<div> <div>作业情况</div> <div>时间片</div> </div>	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	4	2	4	
RR q=1	完成时间	15	12	16	9	17	
	周转时间	15	11	14	6	13	11.8
	带权周转时间	3.75	3.67	3.5	3	3.33	3.46
RR q=4	完成时间	4	7	11	13	17	
	周转时间	4	6	9	10	13	8.4
	带权周转时间	1	2	2.25	5	3.33	2.5



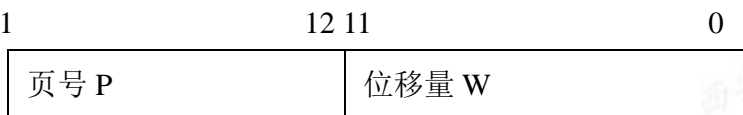
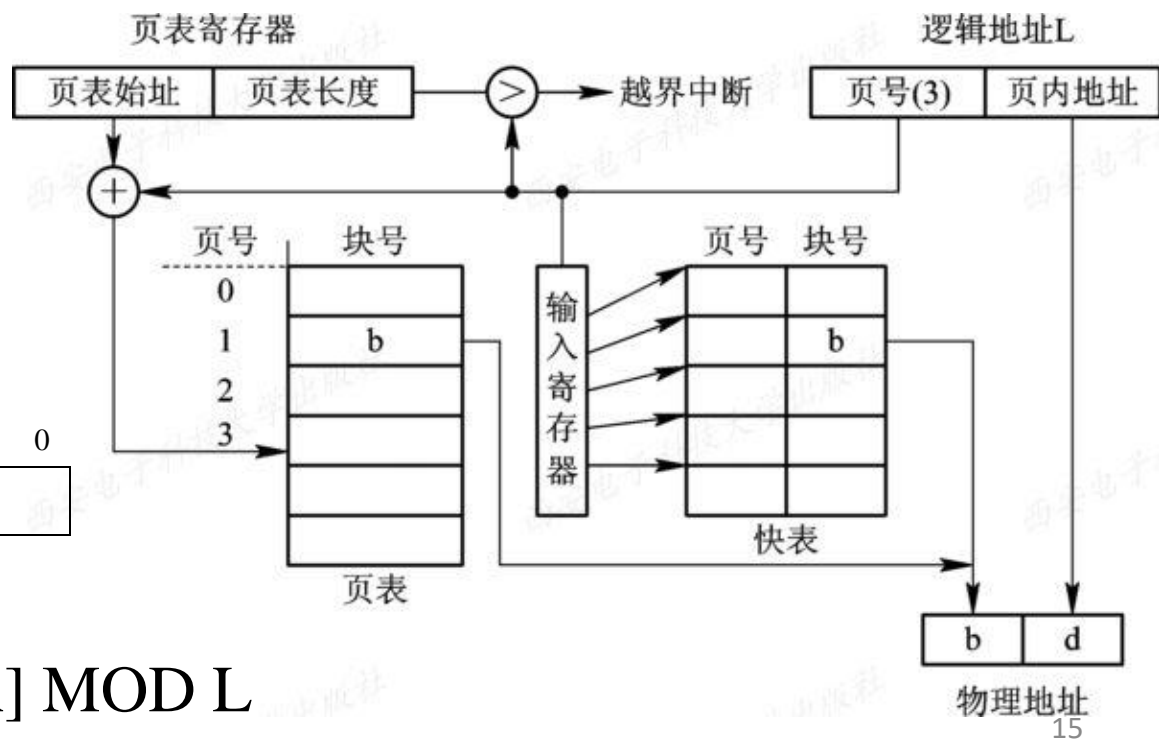
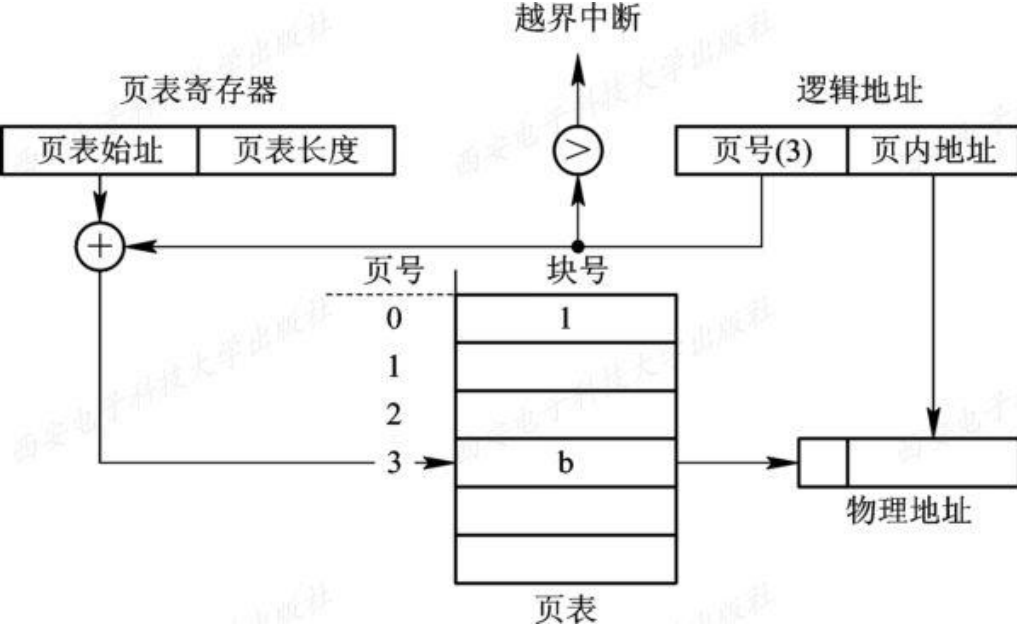
(时间片: $S_1 < S_2 < S_3 \cdots < S_n$)

资源 情况 进 程	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P ₀	7	5	3	0	1	0	7	4	3	3 (2	3 3	2) 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			

第四章

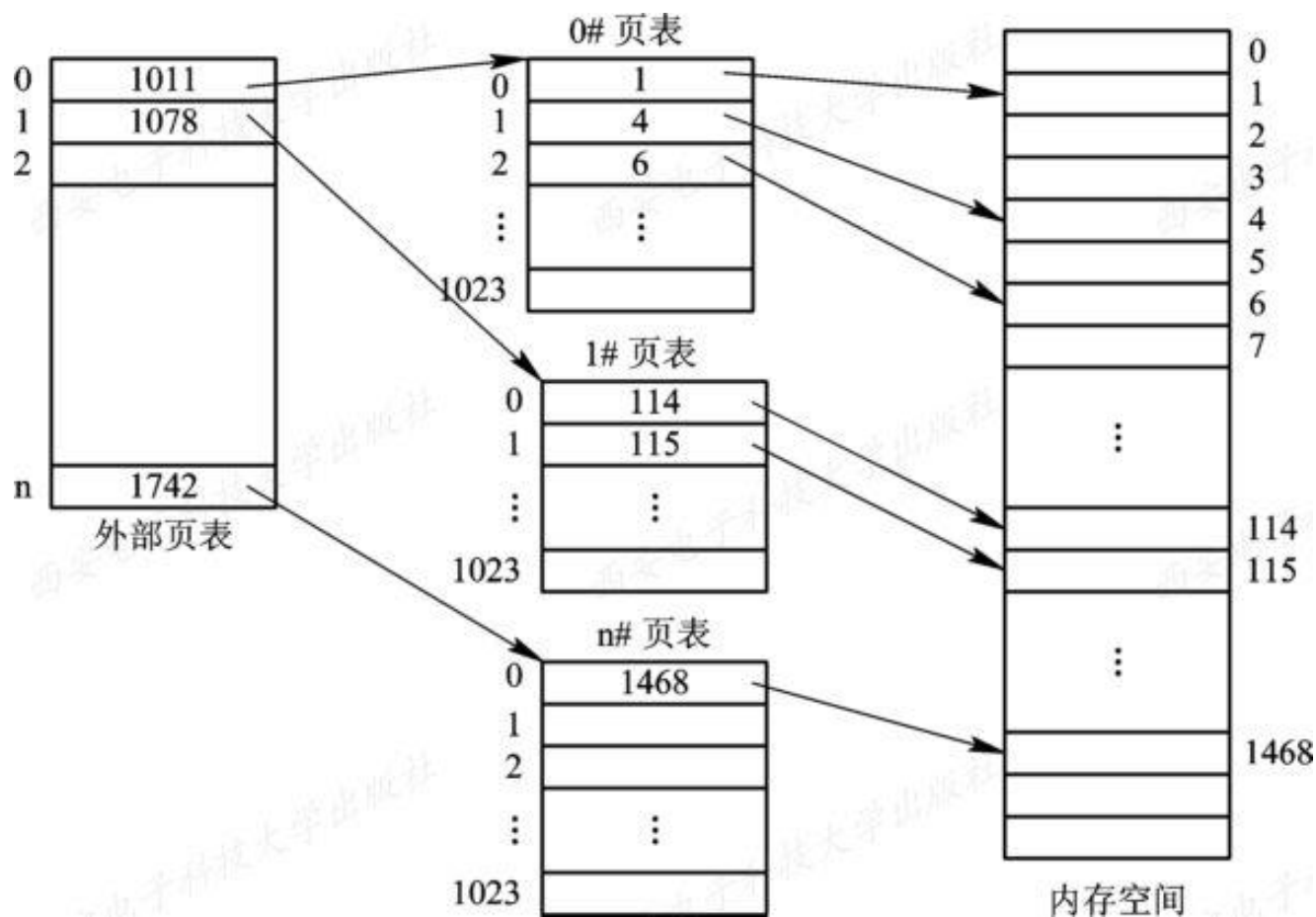
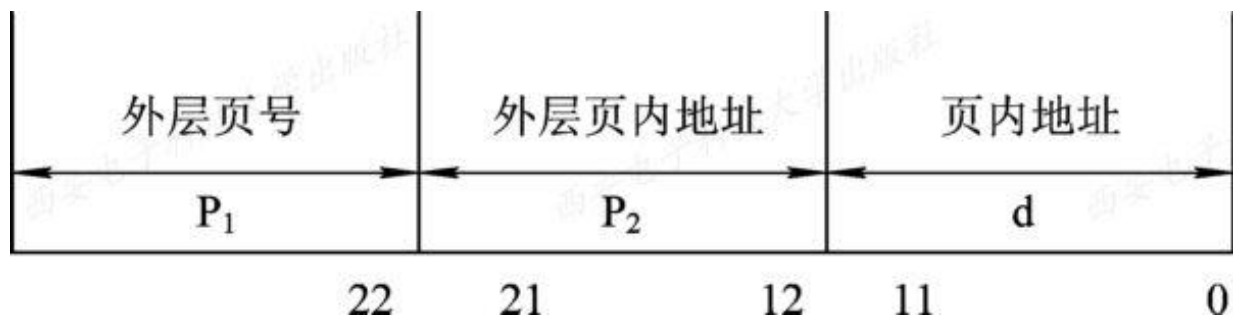
- 存储器的层次结构
- 程序的装入和链接
- 连续分配存储管理方式
- 对换(Swapping)
- 分页存储管理方式
- 分段存储管理方式

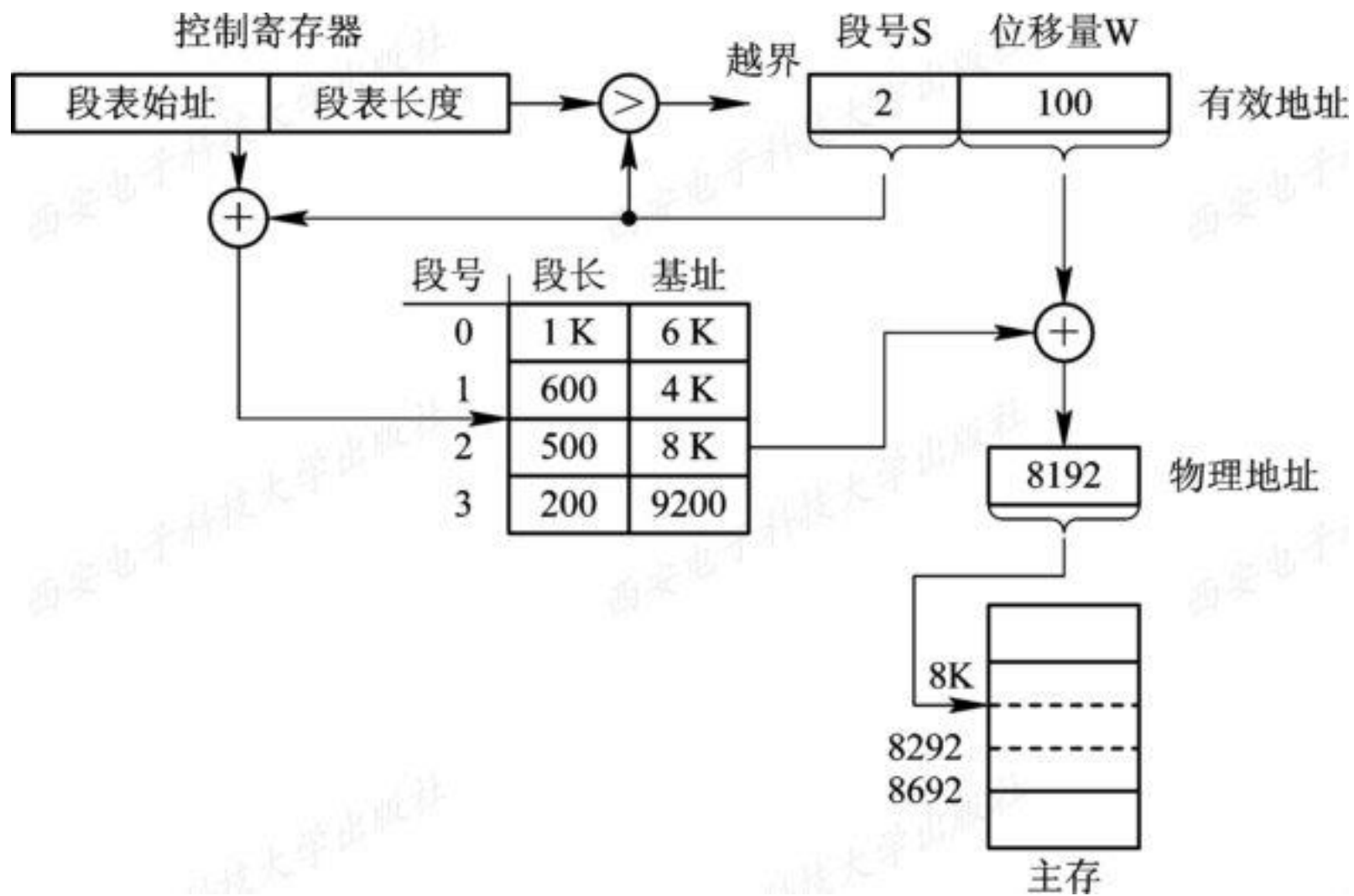
- ✓ 首次适应
- ✓ 循环首次适应
- ✓ 最佳适应
- ✓ 最坏适应
- ✓ 伙伴系统
- ✓ 紧凑
- ✓ 动态重定位
- ✓ 对换
- ✓ 有效访存时间计算
- ✓ 分页、分段

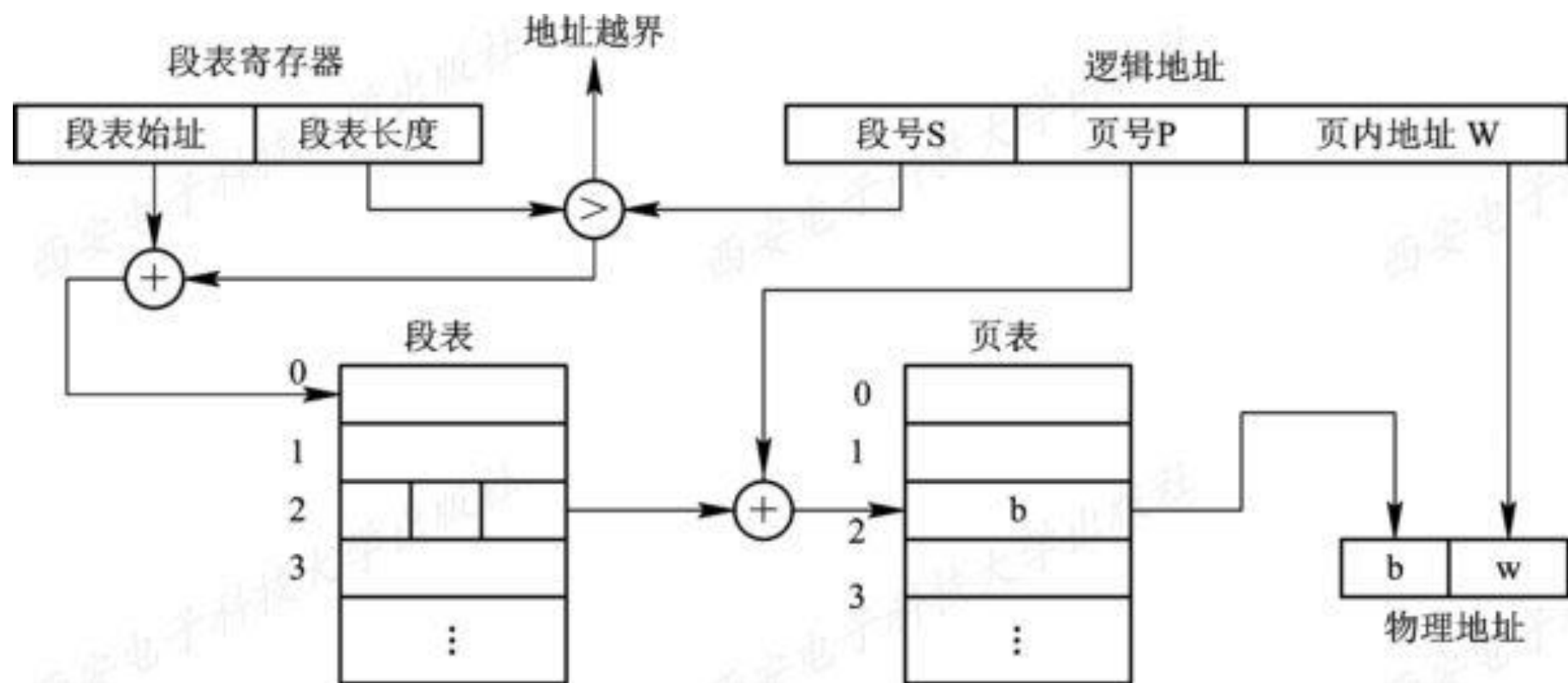


$$P = \text{INT} \left[\frac{A}{L} \right], \quad d = [A] \text{ MOD } L$$

- ✓ 假设某分页系统的主存容量为32K字节，页面大小为1K，对于4页的作业，其0, 1, 2, 3页分别被分配到主存的1, 3, 5, 7块中。请将十进制的逻辑地址1023, 2500, 3500, 4500转换为物理地址，并阐述地址转换过程。





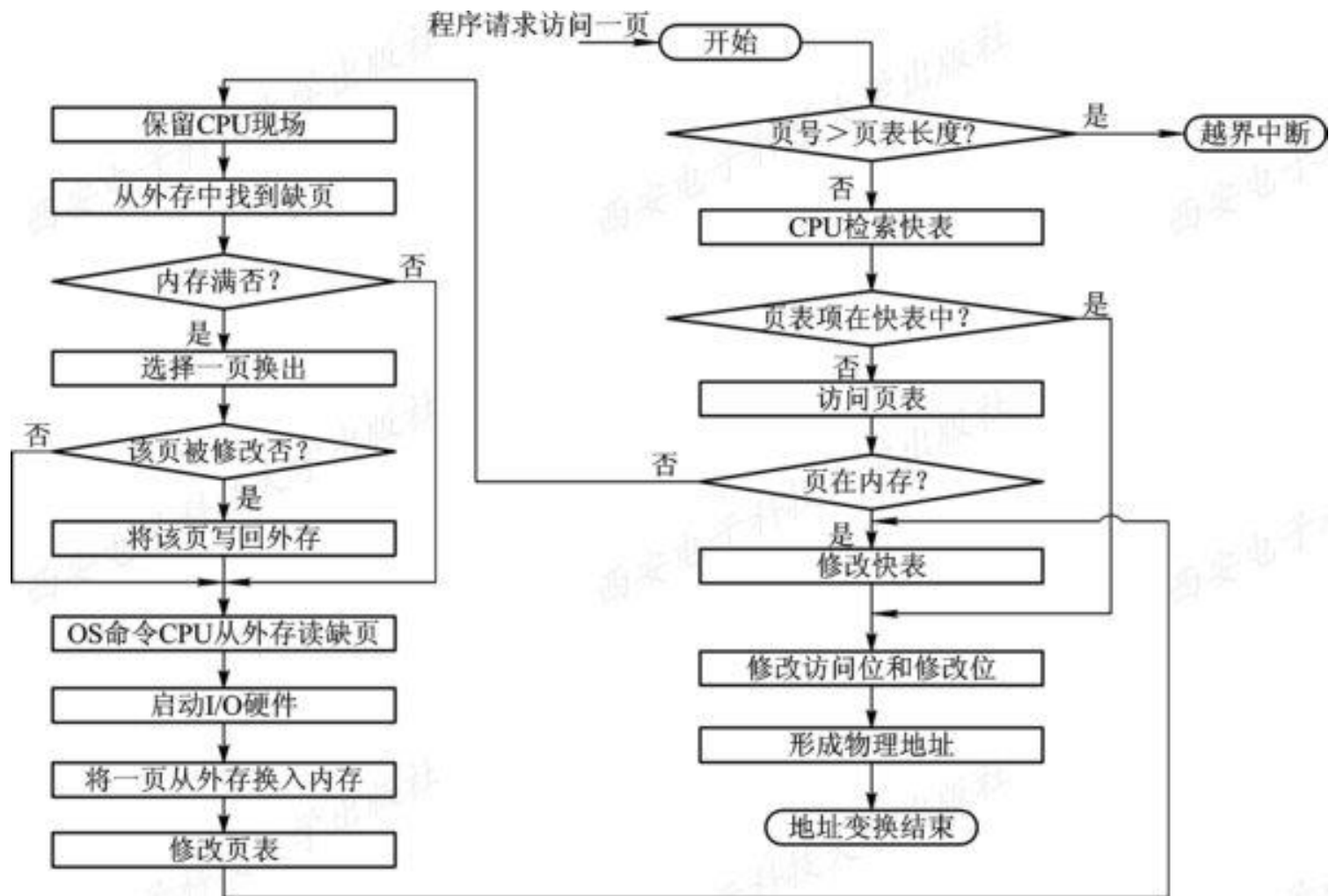


第五章

- 虚拟存储器概述
- 请求分页存储管理方式
- 页面置换算法
- “抖动” 与工作集
- 请求分段存储管理方式

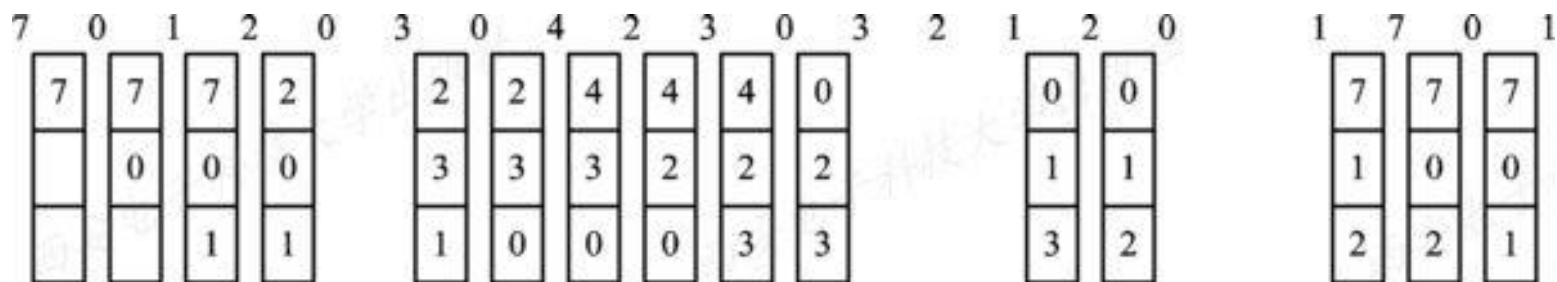
- ✓ 虚拟存储器概念特征
- ✓ 请求分页地址变换
- ✓ 物理块确定原则
- ✓ 缺页率计算
- ✓ 页面置换算法
- ✓ “抖动” 和工作集

页号	物理块号	状态位 P	访问字段 A	修改位 M	外存地址
----	------	-------	--------	-------	------

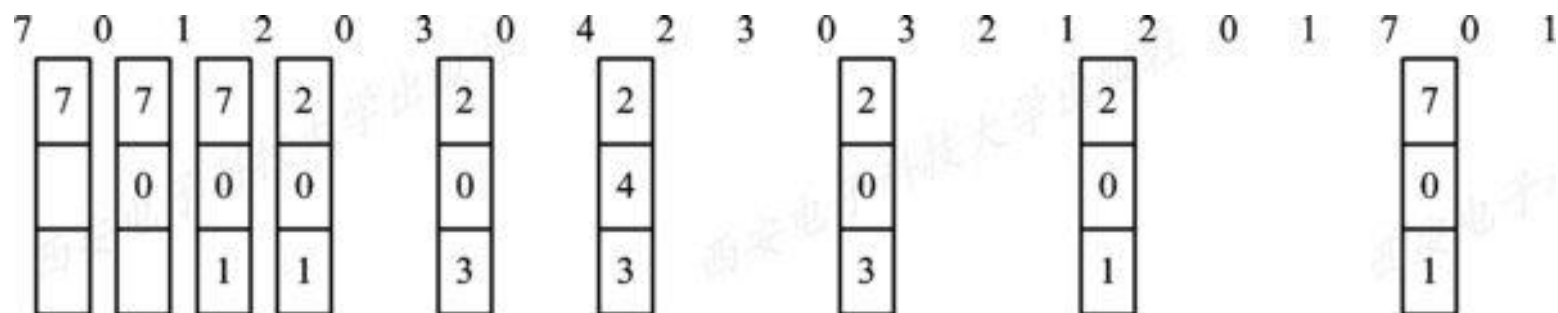


请求分页中的地址变换过程

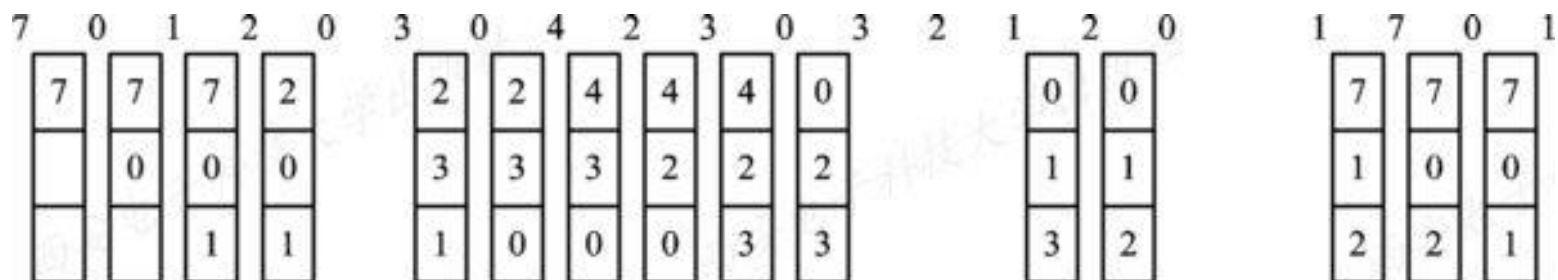
FIFO



Optimal



LRU



第六章

➤ I/O系统的功能、模型和接口

➤ I/O设备和设备控制器

➤ 中断机构和中断处理程序

➤ 设备驱动程序

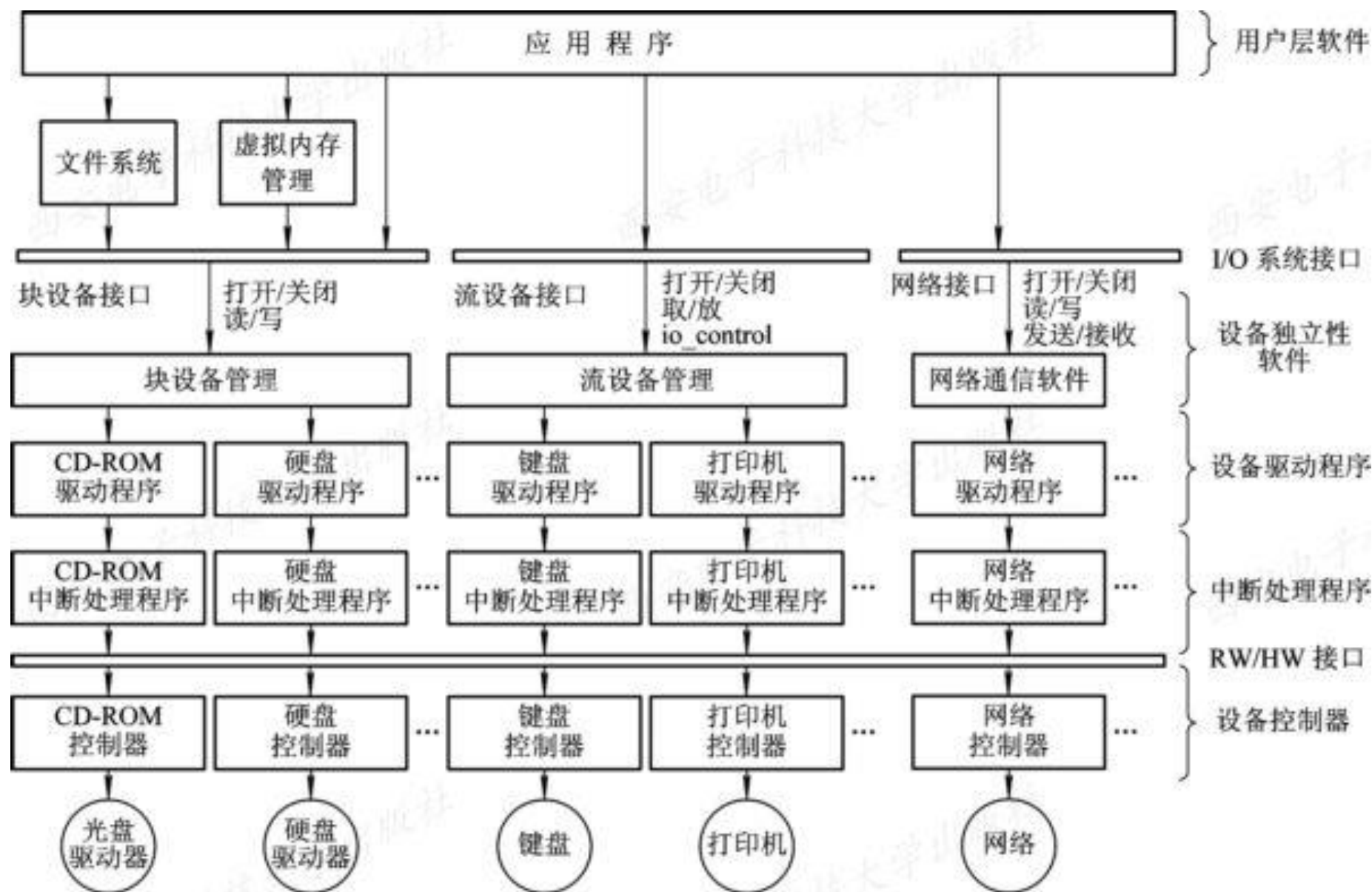
➤ 与设备无关的I/O软件

➤ 用户层的I/O软件

➤ 缓冲区管理

➤ 磁盘存储器的性能和调度

- ✓ 设备类型
- ✓ 设备控制器
- ✓ 通道类型
- ✓ 中断和中断处理
- ✓ I/O设备控制方式
- ✓ 逻辑设备、物理设备
- ✓ 系统调用
- ✓ 假脱机
- ✓ 缓冲区类型
- ✓ 磁盘调度算法



- ✓ 假设磁盘有300个磁道，磁盘请求队列中的请求随机产生，它们按照到达的次序分别处于168、230、265、23、60、70、15、155、170、30号磁道上，当前磁头在100号磁道上，并正由里向外移动。请给出FCFS、SSTF、SCAN及CSCAN算法进行磁盘调度时满足请求的次序，并计算出它们的平均寻道长度。

第七章

- 文件和文件系统
- 文件的逻辑结构
- 文件目录
- 文件共享
- 文件保护

第八章

- 外存的组织方式
- 文件存储空间的管理
- 提高磁盘I/O速度的途径
- 提高磁盘可靠性的技术
- 数据一致性控制

第九章

- 用户接口
- Shell 命令语言
- 联机命令接口的实现
- 系统调用的概念和类型
- UNIX系统调用
- 系统调用的实现