



操作系统 Operating System

总复习

沃天宇

woty@buaa.edu.cn

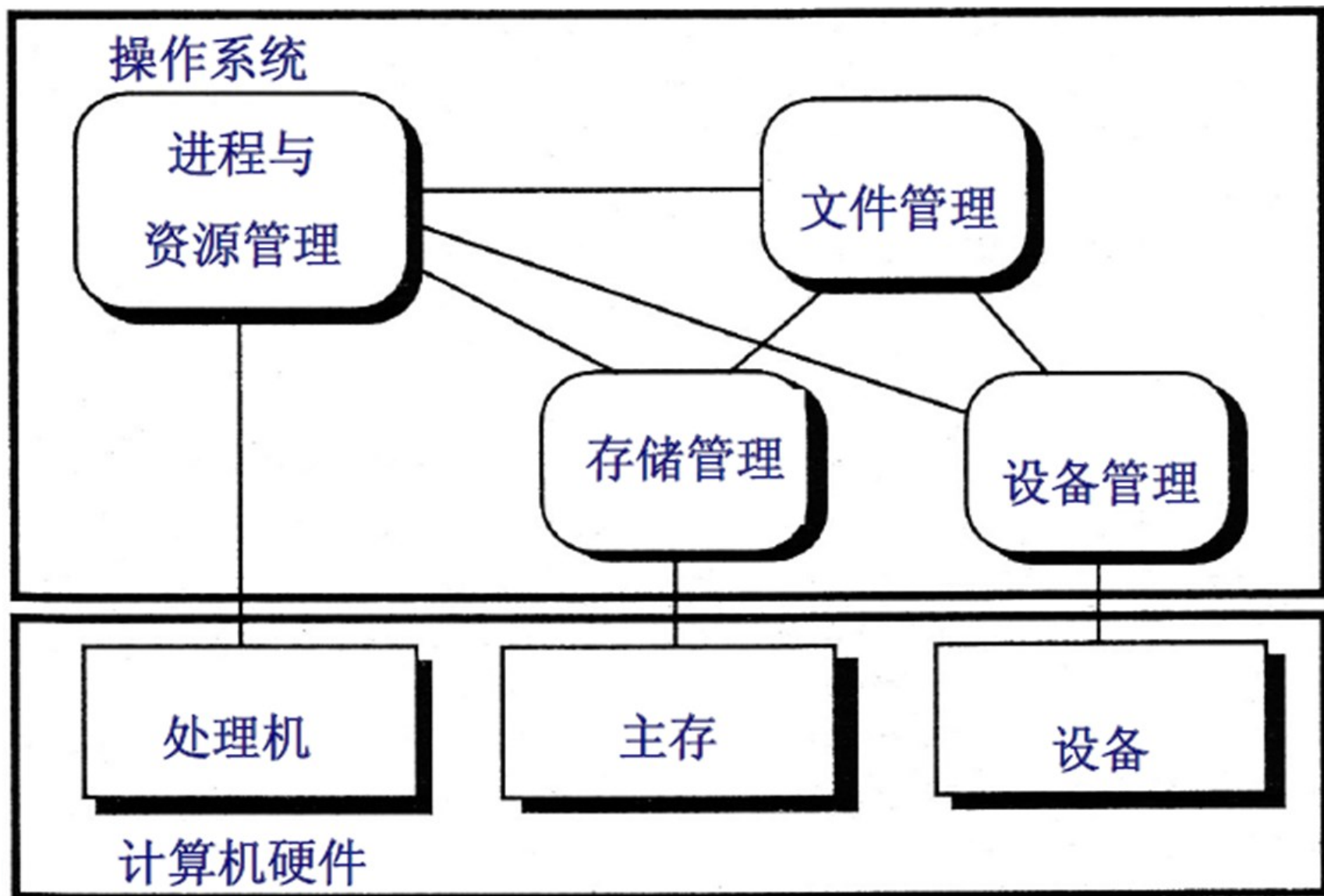
2022年5月26日



课程定位

- 操作系统的基本类型、特征和功能
- 了解操作系统各个组成部分的基本结构和原理
- 掌握在进程管理、存储管理、设备管理、磁盘管理及文件管理等涉及的概念和相关算法
- 了解多处理机操作系统和网络操作系统的基本原理
- 简单了解相关的硬件知识、有代表性的实际系统的设计与实现以及操作系统的一些新概念、新方法以及相关的国际标准







核心内容

- 第一章 操作系统概论
- 第二章 操作系统引导
- 第三章 存储管理
- 第四章 进程机制与并发程序设计
- 第五章 输入/输出系统
- 第六章 文件系统
- 其它：安全、分布式系统、移动OS【自学，不考】



核心内容

- 第一章 操作系统概论
- 第二章 操作系统引导
- 第三章 存储管理
- 第四章 进程机制与并发程序设计
- 第五章 输入/输出系统
- 第六章 文件系统
- 其它：安全、分布式系统



第一章 概论

- 操作系统的发展变化（辨析概念）
 - 批处理、分时；
 - 单道程序、多道程序；
 - 实时、网络、分布式。
- 冯诺依曼体系结构：存储程序式
 - 集中顺序过程控制

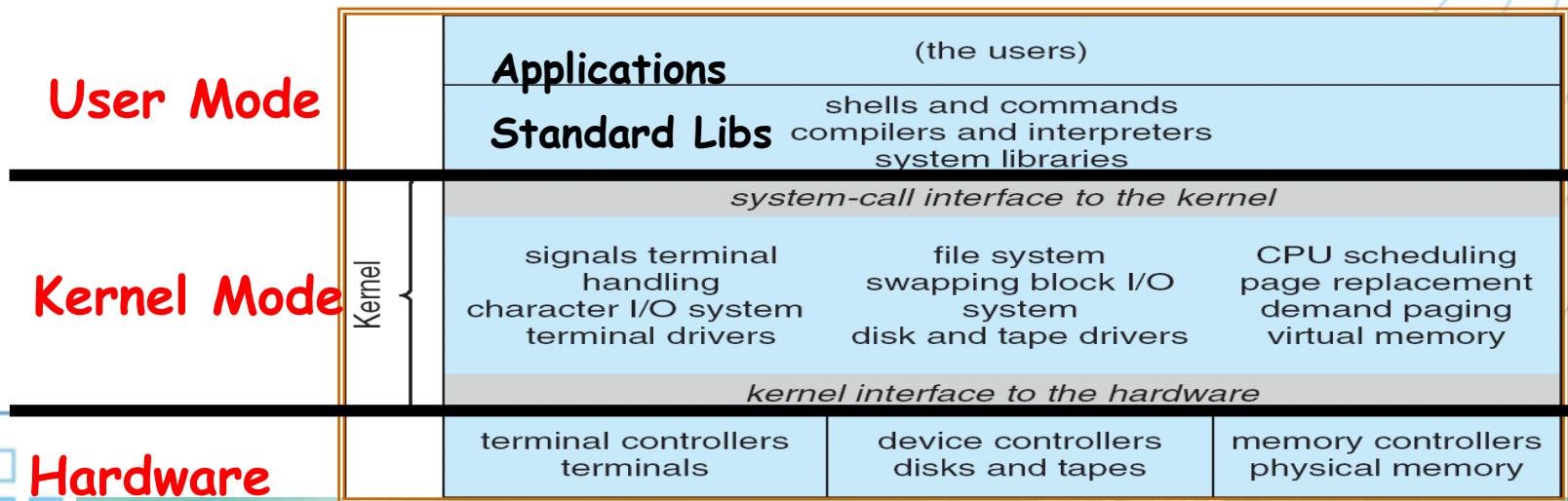


第一章 概论

- 操作系统的主要功能
 - 进程管理（处理机分配）
 - 存储管理
 - 设备管理（包括磁盘管理）
 - 文件系统
- 现代操作系统的基本特征
 - 并发执行
 - 资源共享（复用）
 - 虚拟化管理
 - 异步性（不确定性事件的处理）

第一章 概论

- 操作系统的工作模式
 - 内核态/管态
 - 用户态/目态
 - （切换过程？）
- 几个术语
 - 特权指令/保护指令
 - 异常：中断、陷入（作用、区别？）（系统调用过程）



第一章 概论

• 分析操作系统的几种观点

- 用户观点：操作系统为用户提供了哪些功能可以使用
- 资源管理：回答了整个操作系统是由哪几部分组成的
- 进程观点：指明了这些资源管理程序在什么时候开始起作用，以及它们在执行过程中是如何相互联系的
- 模块分层：从操作系统构建的角度，考虑操作系统的结构
- 抽象与权衡



第一章 概论

- 操作系统结构
 - 什么是操作系统内核？如何进入？
 - 微内核、整体内核（辨析，差异、优缺点）



第一章 概论

- 熟悉系统调用的概念与实现方法
 - 什么是系统调用？与函数调用的区别？
 - 用户态程序如何访问操作系统提供的系统功能（如创建新进程），描述过程，如何切换入内核态执行？
- 系统调用与shell命令的区别？
 - API vs CLI
- Shell命令与脚本
- Linux操作系统的接口

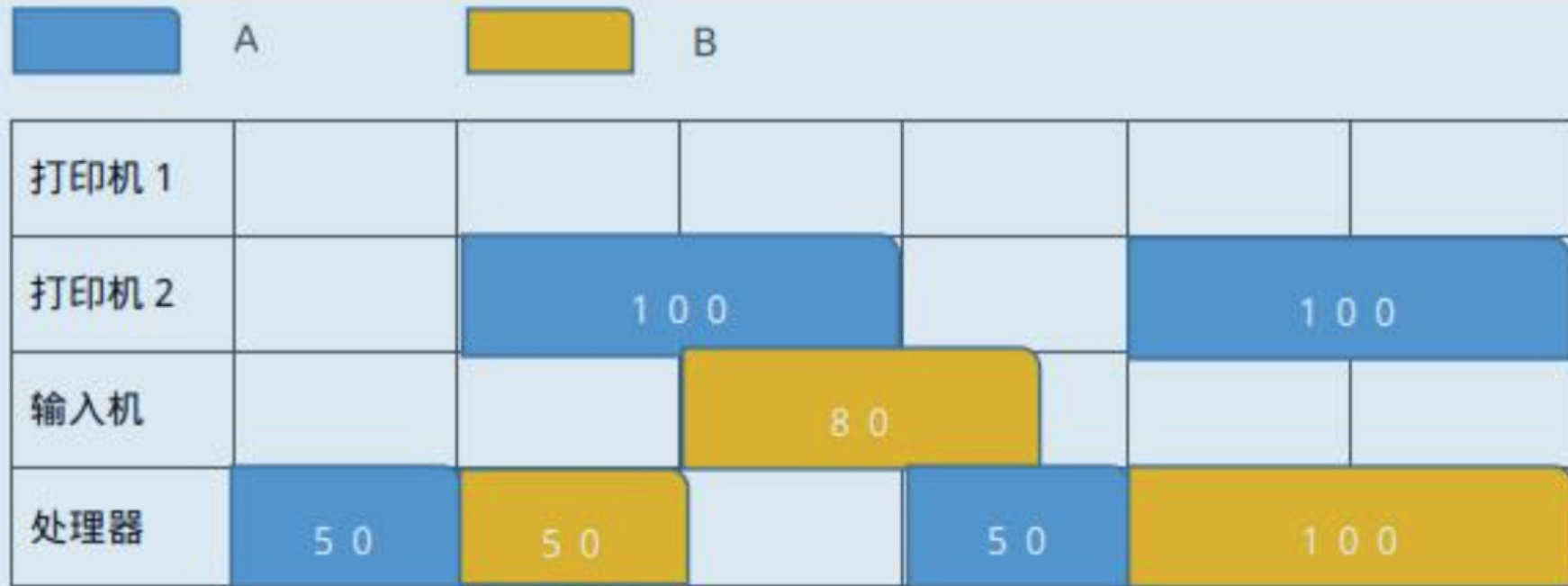
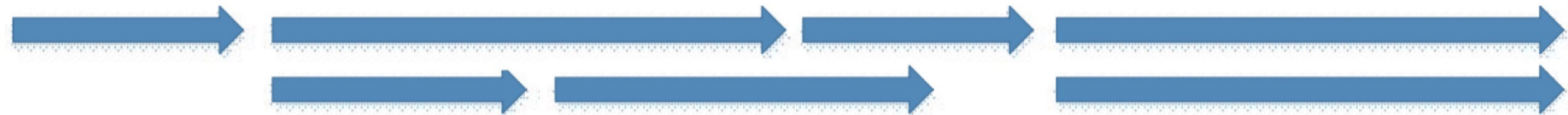


作业1.1

- 设一计算机系统有输入机一台、打印机两台，现有二道程序同时投入运行，且程序A先开始运行，程序B后运行。程序A的运行轨迹为：计算50ms，打印信息100ms，再计算50ms，打印信息100ms，结束。程序B运行的轨迹为：计算50ms，输入数据80ms，再计算100ms，结束。要求：
 - (1) 用图画这二道程序并发执行时的工作情况。
 - (2) 说明在二道程序运行时，CPU有无空闲等待？若有，在哪段时间内等待？为什么会空闲等待？
 - (3) 程序A、B运行时有无等待现象？在什么时候会发生等待现象？



作业1.1



- (2) CPU 有空闲等待, 当 A 在打印 B 在输入时 CPU 等待, 因为并发的程序不够多导致这段时间没有程序需求 CPU 资源
- (3) 有等待现象, CPU 资源不足时出现等待。

课后作业1.2

- 在单CPU和两台I/O设备(I_1, I_2)的多道程序设计环境下, 同时投入3个作业 J_1, J_2 和 J_3 运行, 其对CPU和I/O设备使用的顺序与时间如下:
 - J_1 : $I_2(30\text{ms}) \rightarrow \text{CPU}(10\text{ms}) \rightarrow I_1(30\text{ms}) \rightarrow \text{CPU}(10\text{ms}) \rightarrow I_2(20\text{ms})$
 - J_2 : $I_1(20\text{ms}) \rightarrow \text{CPU}(20\text{ms}) \rightarrow I_2(40\text{ms})$
 - J_3 : $\text{CPU}(30\text{ms}) \rightarrow I_1(20\text{ms}) \rightarrow \text{CPU}(10\text{ms}) \rightarrow I_1(10\text{ms})$
- 假定CPU和I/O设备能够并行, I_1 和 I_2 能够并行。作业优先级 $J_1 > J_2 > J_3$, 高优先级作业可抢占低优先级作业的CPU, 但不能抢占I/O设备。
- 问题:
 - 分别求出3个作业的turnaround时间(wall-clock时间)。
 - 计算CPU的利用率(计算时间/(计算时间+空闲时间))。
 - 计算I/O设备的利用率(工作时间/(工作时间+空闲时间))。



课后作业1.2

在单CPU和两台I/O设备(I_1 , I_2)的多道程序设计环境下, 同时投入3个作业 J_1 , J_2 和 J_3 运行, 其对CPU和I/O设备使用的顺序与时间如下:

J_1 : $I_2(30\text{ms}) \rightarrow \text{CPU}(10\text{ms}) \rightarrow I_1(30\text{ms}) \rightarrow \text{CPU}(10\text{ms}) \rightarrow I_2(20\text{ms})$

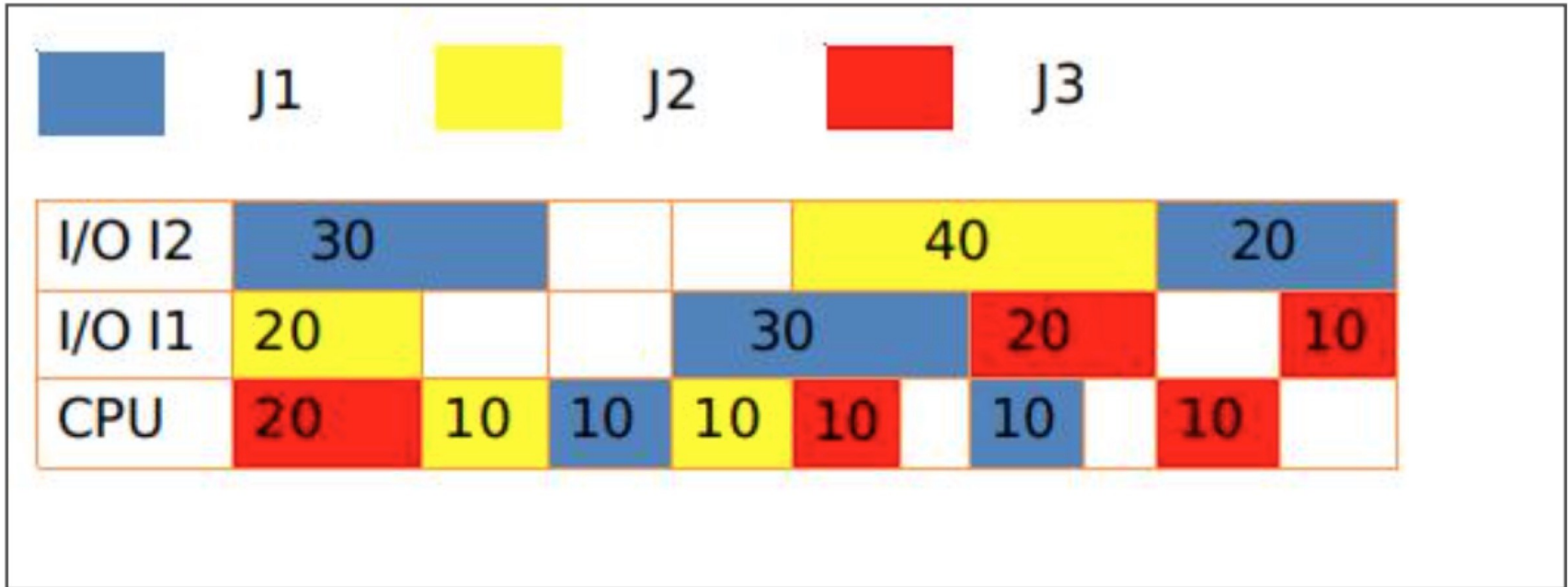
J_2 : $I_1(20\text{ms}) \rightarrow \text{CPU}(20\text{ms}) \rightarrow I_2(40\text{ms})$

J_3 : $\text{CPU}(30\text{ms}) \rightarrow I_1(20\text{ms}) \rightarrow \text{CPU}(10\text{ms}) \rightarrow I_1(10\text{ms})$

假定CPU和I/O设备能够并行, I_1 和 I_2 能够并行。作业优先级 $J_1 > J_2 > J_3$, 高优先级作业可抢占低优先级作业的CPU, 但不能抢占I/O设备。



课后作业1.2



- 分别求出3个作业的turnaround时间（wall-clock时间）。
- 计算CPU的利用率（计算时间/(计算时间+空闲时间)）。
- 计算I/O设备的利用率（工作时间/(工作时间+空闲时间)）。

(1) 110ms、90ms、110ms

(2) 截止至 3 个程序都运行完时 CPU 利用率 $(10+10+20+30+10)/110=72.73\%$ (3)

截止至 3 个程序都运行完时 I01 利用率 $(30+20+20+10)/110=72.73\%$

截止至 3 个程序都运行完时 I02 利用率 $(30+20+40)/110=81.82\%$

问题：可移植的操作系统可以从一个系统架构移植到另外一个系统架构而**无需修改**。

- (1) 请解释为什么构建完全可移植的OS是不可能的？
- (2) 如果需要你设计一个高度可移植的OS，那么请描述你需要设计的两个层次？



(1) 请解释为什么构建完全可移植的OS是不可能的？

每个系统架构都有自己独特的一套指令集。因此，奔腾架构的设备无法执行SPARC程序，反之亦然。而且，不同架构的总线架构也不同，（例如 VME, ISA, PCI, MCA, SBus, ...），CPU的字长度也不同（32/64 bit）。由于这些硬件的差异，构建完全可移植的操作系统并不可能。



(2)如果需要你设计一个高度可移植的OS，那么请描述你需要设计的两个层次？

一个高度可移植的操作系统包含两个层次：——机器相关层和机器无关层。机器相关层处理硬件的特性，并且需要为每个架构单独实现。这层提供一个统一的接口，供机器无关层使用。而设备无关层仅仅需要实现一次。要想实现高可移植性，机器相关层的规模应该设计的越小越好。





核心内容

- 第一章 操作系统概论
- **第二章 操作系统引导**
- 第三章 存储管理
- 第四章 进程机制与并发程序设计
- 第五章 输入/输出系统
- 第六章 文件系统
- 其它：安全、分布式系统



第二章 操作系统引导

- OS分阶段引导过程
- 各个阶段的职责
- 各个阶段引导程序的存储与装载过程
- 简述Linux内核在x86平台上的加载过程



作业2

- OS启动为什么慢？如何提高OS启动效率？
 - 硬件角度
 - 软件角度
 - 软硬件协同角度





核心内容

- 第一章 操作系统概论
- 第二章 操作系统引导
- **第三章 存储管理**
- 第四章 进程机制与并发程序设计
- 第五章 输入/输出系统
- 第六章 文件系统
- 其它：安全、分布式系统



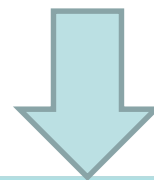
存储管理的主要内容

1. 存储管理的功能
2. 分区存储管理
3. 页式存储管理
4. 段式存储管理
5. 虚拟存储原理与算法

1. 存储管理的功能

- 内存的分配与回收
- 存储保护
- 地址转换
- 静态重定位
- 动态重定位
- 存储共享
- “扩充”内存容量

地址空间



存储空间

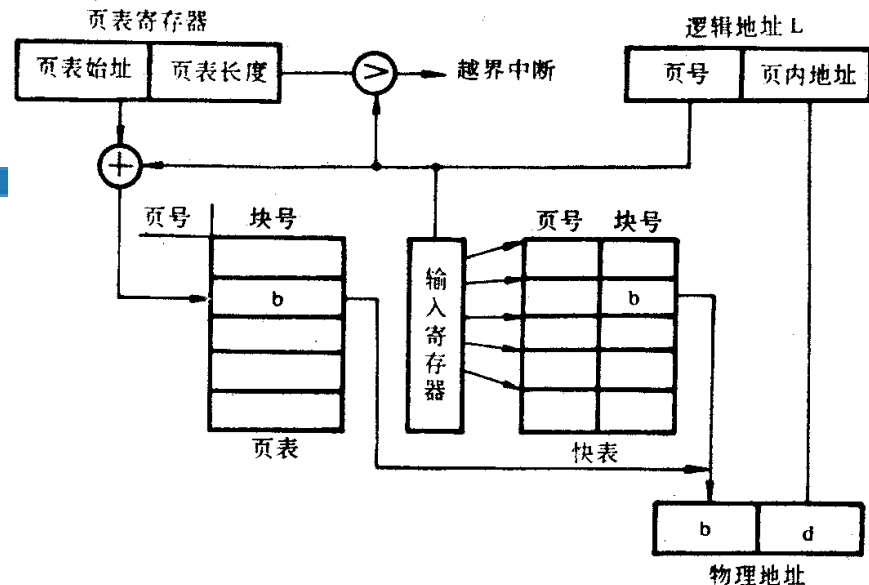
2. 分区存储管理

- 固定分区
 - 基本思想
 - 优缺点
- 可变分区
 - 基本思想
 - 优缺点
 - 主要算法: BestFit, WorstFit, FirstFit, NextFit
- 覆盖与交换

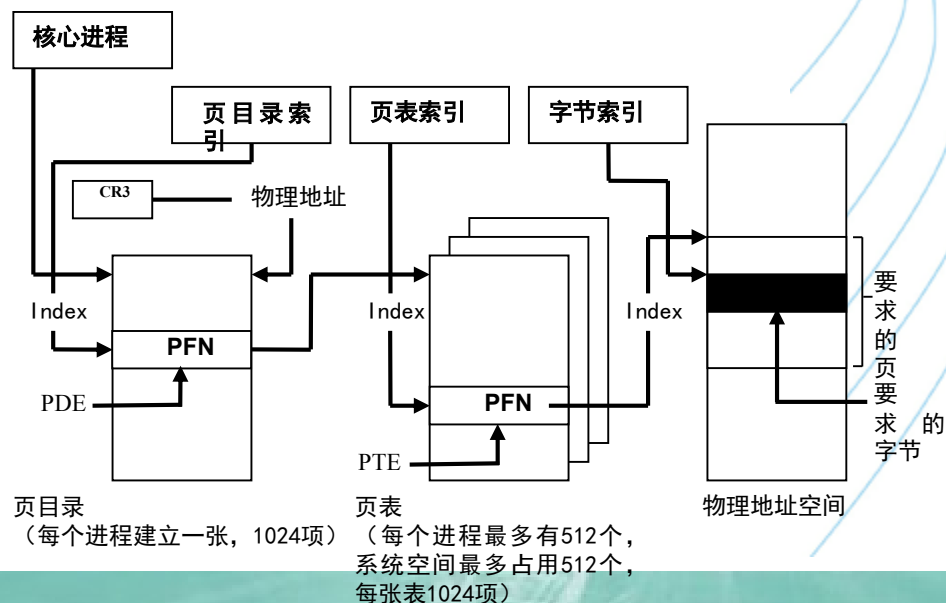


3.页式存储管理

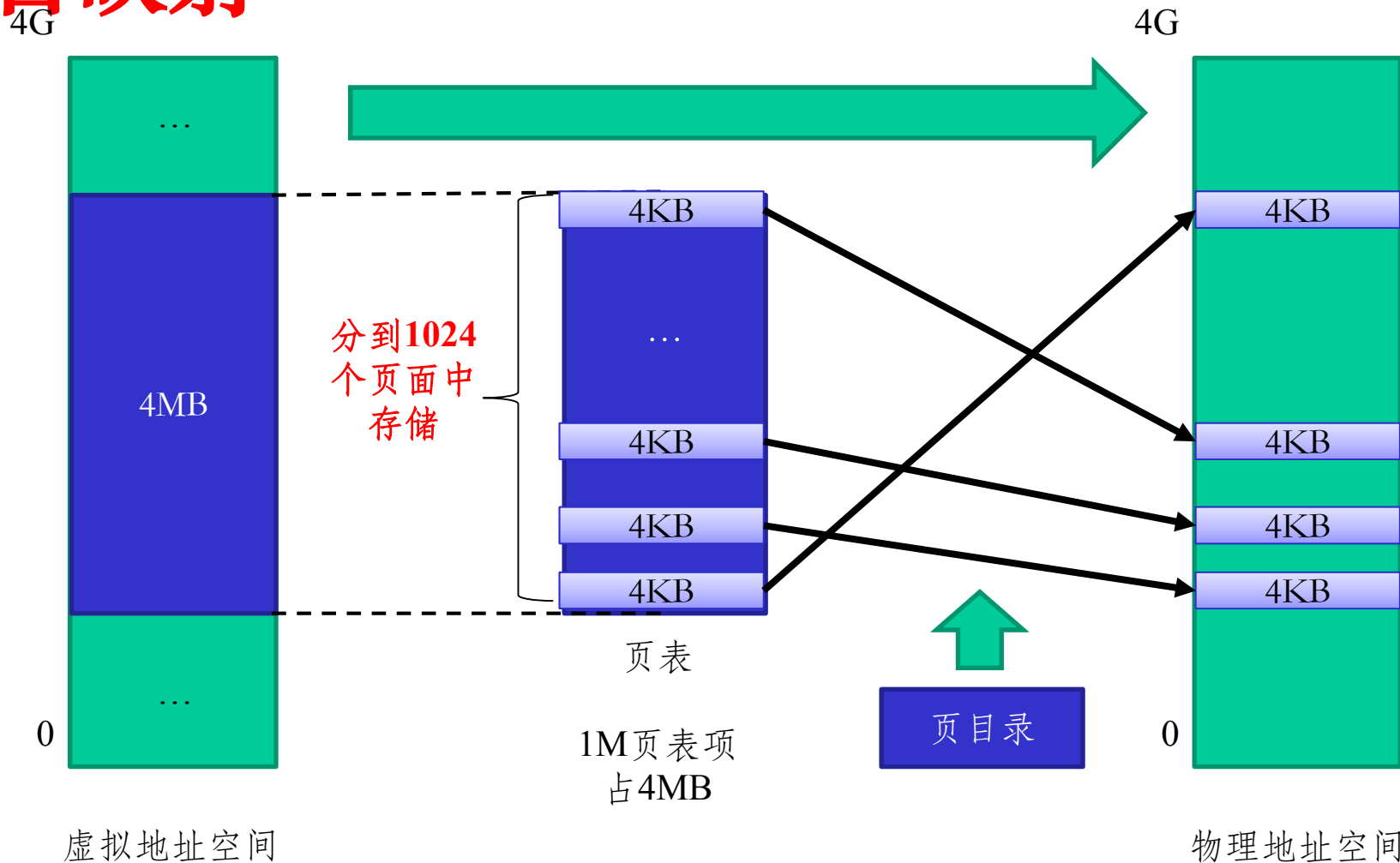
- 页面、页框
- 地址转换
 - MMU
 - 页表：多级页表、杂凑页表、反置页表
 - 快表TLB
 - 页面的大小



具有快表的地址变换机构



自映射

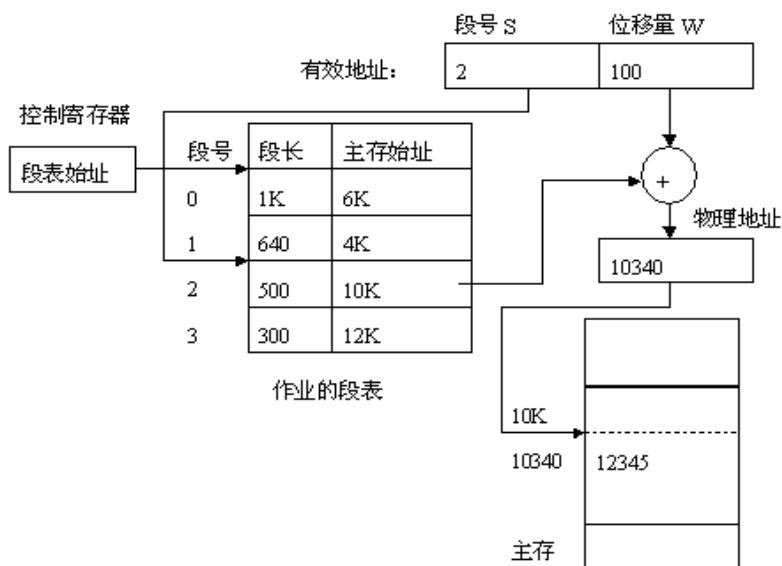


1100 0000 0011 0000 0000 1100 0000 0000



4.段式内存管理

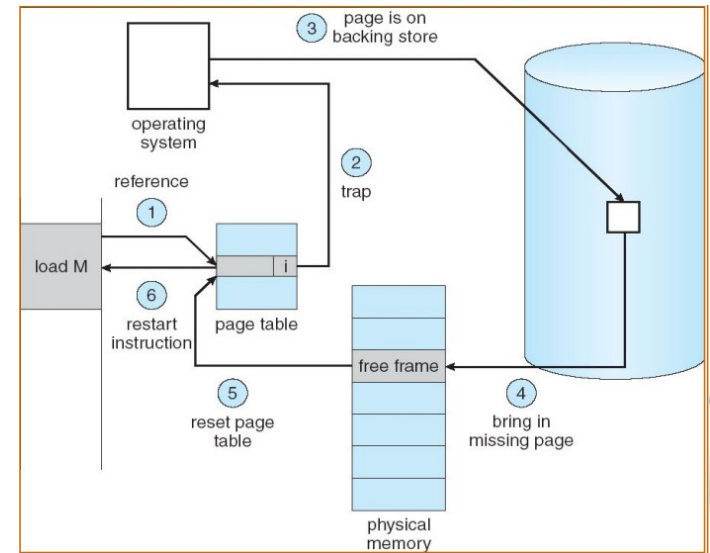
- 基本思想
- 地址变换
- 分页与分段的比较（辨析异同）
- 段页式内存管理



	页式存储管理	段式存储管理
目的	实现非连续分配, 解决碎片问题	更好地满足用户需要
信息单位	页 (物理单位)	段 (逻辑单位)
大小	固定 (由系统定)	不定 (由用户程序定)
内存分配单位	页	段
作业地址空间	一维	二维
优点	有效解决了碎片问题 (没有外碎片, 每个内碎片不超过页大小); 有效提高内存的利用率; 程序不必连续存放。	更好地实现数据共享与保护; 段长可动态增长; 便于动态链接

5. 虚拟存储：原理与算法

- 内存访问的局部性原理
- 页面置换策略
 - 最优算法：OPT
 - 先进先出：FIFO
 - 最近最少使用：LRU
 - 最不频繁使用：LFU
 - 第二次机会算法、时钟算法：Clock
 - 工作集策略
- 缺页中断率



其他内容

• 内存抖动

- 刚被置换出去的页，很快又要访问，因而要把它重新调入；可调入不久又再次被置换出去，这样再访问、再调入，如此反复，使得整个系统的页面替换非常频繁，以致大部分的机器时间都花在来回进行的页面调度上，只有一小部分时间用于进程的实际运算。
- 解决办法：局部置换策略、工作集算法、预留部分页面、挂起若干进程

- 写时复制技术
- 内存映射文件
- 存储保护



作业3

1. 动态内存分配需要对内存分区进行管理，一般使用位图和空闲链表两种方法。128MB的内存以n字节为单元分配，对于链表，假设内存中数据段和空闲区交替排列，长度均为64KB。并假设链表中的每个节点需要记录32位的内存地址信息、16位长度信息和16位下一节点域信息。这两种方法分别需要多少字节的存储空间？那种方法更好？
- 答：128MB= 2^{27} ，n字节为单元，所以 $2^{27}/n$ 个单元，使用位图需 $2^{27}/n$ 位，即 $2^{24}/n$ 字节，使用链表需 $128\text{MB}/64\text{KB}=2\text{K}$ 个节点，每个节点8（64位）字节，所以16KB= 2^{14} 字节。当n小于1KB时，链表较好；反之，位图更好。



作业3

- 2. 在一个交换系统中，按内存地址排列的空闲区大小是：10KB、4KB、20KB、18KB、7KB、9KB、12KB和15KB。对于连续的段请求：12KB、10KB、9KB。使用FirstFit、BestFit、WorstFit和NextFit将找出哪些空闲区？
- 答：
 - FirstFit-20KB, 10KB, 18KB
 - BestFit-12KB, 10KB, 9KB
 - WorstFit-20KB, 18KB, 15KB
 - NextFit-20KB, 18KB, 9KB



作业3

- 3. 解释逻辑地址、物理地址、地址映射，并举例说明。
 - 逻辑地址：程序产生的地址
 - 物理地址：在存储器里以字节为单位存储信息，为了正确地存放或取得信息，每一个字节单元给以一个唯一的存储器地址，称为物理地址
 - 地址映射：将逻辑地址转化为物理地址的过程

作业3

- 4. 解释页式（段式）存储管理中为什么要设置页（段）表和快表，简述页式（段式）地址转换过程。
 - 页表是为了保存页面号和物理块的映射关系，而段表保存了段号和相应段基址的映射关系，以便进行地址转换。
 - 快表：是一个高速缓存，用以存放一部分页表，加快转换速度转换过程：给出一个逻辑地址，将其页号与快表和页表中的逻辑页号同时进行比较，命中后用物理页号替换逻辑页号，和页内偏移拼接，得到物理地址

作业3

• 5.叙述缺页中断的处理流程。

- 缺页中断处理过程：当某逻辑地址不在页表中时，产生缺页中断，如果地址有效，操作系统从外存将对应的一页数据读入内存中空闲的页中(如果没有空闲页先淘汰出一页)，并更新页表。回到中断处继续执行。

作业4

- 6. 假设一个机器有38位的虚拟地址和32位的物理地址。
 - (1) 与一级页表相比，多级页表的主要优点是什么？
 - (2) 如果使用二级页表，页面大小为16KB，每个页表项有4个字节。应该为虚拟地址中的第一级和第二级页表域各分配多少位？
- 答：页面16KB，14 位。二级页表大小和页面大小相同，页表项4 字节需要2 位，所以二级页表域 $14-2=12$ 位。一级页表 $38-12-14=12$ 位。一级和二级页表域分别需要12 位，偏移量需要14 位。（画结构）



作业3

- 假设页面的访问存在一定的周期性循环，但周期之间会随机出现一些页面的访问。例如：
0,1,2...,511,431,0,1,2...511,332,0,1,2,...,511等。
请思考：
 - (1) LRU、FIFO和Clock算法的效果如何？
 - (2) 如果有500个页框，能否设计一个优于LRU、FIFO和Clock的算法？
- 答：
- (1) 不符合局部性原理的访问。三种算法产生的缺页中断是一样的。
- (2) 尽量把工作集装入内存。将0-498 页面映射到固定的页框，每次只置换第499 个页面。



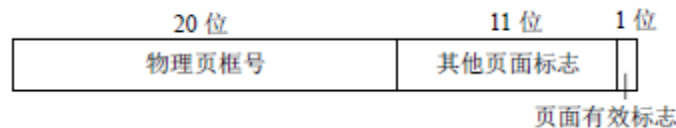
作业3

- 14. 一个32位的虚拟存储系统有两级页表，其逻辑地址中，第22到31位是第一级页表（页目录）的索引，第12位到21位是第二级页表的索引，页内偏移占第0到11位。每个页表（目录）项包含20位物理页框号和12位标志位，其中最后1位为页有效位。

- 逻辑地址格式如下：

10 位	10 位	12 位
页目录号	二级页表号	页内偏移量

- 页目录项、页表项格式如下：





作业3

- (1) 请问进程整个的地址空间有多少字节？一页有多少字节？
– 进程整个的地址空间为 $2^{32}=4\text{G}$ 字节，一页为 $2^{12}=4\text{K}$ 字节。
- (2) 如果当前进程的页目录物理基地址、页目录和相应页表内容如图下所示，请描述访问以下虚拟地址时系统进行地址转换的过程，如可行给出最终访存获取到的数据。虚拟地址：0x0、0x00803004、0x00402001
- (3) 要想访问物理地址0x326028，需要使用哪个虚拟地址？

页目录物理基地址

0x1000

页目录
物理地址：0x1000

0	0x0
1	0x1001
2	0x5001
3	0x20001
4	0x0
	...
1023	0x0

页表
物理地址：0x5000

0	0x0
1	0x4e001
2	0x67001
3	0x20001
4	0x0
	...
1023	0x0

页表
物理地址：0x20000

0	0x9000
1	0x326001
2	0x41001
3	0x0
4	0x0
	...
1023	0x0



作业3

- a)虚拟地址0x0：虚拟页目录号0，查询对应0号页目录项，得到页目录项0x0，由于有效标志位为0，表示该页面无效，因此页表中不存在该虚拟地址到物理地址的映射，产生缺页中断；
- b)虚拟地址0x00803004：虚拟页目录索引号为2，二级页表索引号为3，页内偏移为4，查询对应2号页目录项，得到0x5001，有效标志位为1，在物理地址0x5000查找二级页表，找到对应的3号页表项，0x20001，由于有效标志位为1，对应物理页框基地址为0x20000，根据页内偏移4，查找偏移地址为4的内存（第5个字节），得到数据0x326001（按字寻址，1字=4字节）（如果回答按字节寻址得到0x00或者0x01也算对，分别对应不同尾端）。



作业3

- c)虚拟地址0x00402001：虚拟页目录索引号为1，二级页表索引号为2，页内偏移为1，查询对应1号页目录项，得到0x1001，有效标志位为1，在物理地址0x1000查找二级页表，即为页目录自身，找到对应的2号页表项，0x5001，由于有效标志位为1，对应物理页框基地址为0x5000，根据页内偏移1（第二个字节），查找对应内存，得到数据0x00（按字节寻址）。
- 3.要想访问物理地址0x326028，需要使用虚拟地址0x00C01028，其中虚拟页目录索引号为3，二级页表索引号为1，页内偏移为0x028



核心内容

- 第一章 操作系统概论
- 第二章 操作系统引导
- 第三章 存储管理
- **第四章 进程机制与并发程序设计**
- 第五章 输入/输出系统
- 第六章 文件系统
- 其它：安全、分布式系统



主要内容

1. 进程与线程的基本概念
2. 进程调度算法
3. 进程通信
4. 死锁问题

1. 进程与线程的基本概念

- 并发与并行的区别
- 进程与程序的区别
- 进程的三个基本状态、转换条件
- 进程的控制原语
- 进程的组成：程序、数据、PCB
- 进程与线程的区别（辨析异同）
 - 共享地址空间、轻量级、切换性能



2. 进程调度算法

- 调度的三个类型：高级、中级、低级
- 进程调度算法
 - 总体上：非抢占、抢占
 - 评价指标：
 - 周转时间、平均周转时间、带权平均周转时间
 - 吞吐量
 - 响应时间（响应时间）
 - CPU利用率
 - 常见调度算法：FCFS, SJF, 轮转调度, 优先级调度
 - 甘特图

3. 进程通信：同步与互斥

- 临界资源、临界区
 - 资源角度：一段时间内只允许一个进程使用
 - 代码角度：由于共享临界资源，必须互斥执行的程序段是临界区
- 同步、互斥
 - 概念
- 信号量、P、V操作
 - 信号量取值的含义（ $S=0$ 表示什么含义？）
 - 信号量是只能由P、V操作修改的数据结构
- 管程的基本概念



同步与互斥的原则（正确、公平、效率）

- 空闲让进
 - 临界资源处于空闲状态，允许进程进入临界区
 - 临界区内仅有一个进程执行
- 忙则等待
 - 临界区有进程正在执行其中的代码，所有其他进程则不可以进入临界区
- 有限等待
 - 对要求访问临界区的进程，应在保证在有限时间内进入自己的临界区，避免死等。
- 让权等待
 - 当进程不能进入自己的临界区时，应立即释放处理机，避免忙等。

经典的同步与互斥问题（编程）

- 生产者—消费者问题
 - 同步、互斥
- 读者—写者问题
 - 互斥
- 哲学家就餐问题
 - 互斥
- 睡觉的理发师问题
 - 同步、互斥

练习（2015）

- 1个仓库最多可以容纳100件产品（不分产品类型），每次只允许一个产品进出仓库。甲乙两个车间分别生产A、B两种产品并共用上述仓库。如果仓库满了则不能进行新的生产。有2个需要A产品的客户和2个需要B产品的客户，分别从仓库提取A、B产品。请用P、V操作来实现上述甲、乙车间以及A、B产品的客户之间的同步与互斥关系。



4. 死锁

- 基本概念：死锁、活锁、饥饿
- 产生死锁的必要条件
 - 互斥、不可剥夺、请求和保持、环路等待
- 进程-资源图/资源分配图
- 处理死锁的方法
 - 允许死锁发生：无作为、检测与解除死锁
 - 不允许死锁发生：预防死锁、避免死锁

死锁处理方法

- 预防死锁（静态）：破坏死锁产生的四个条件
- 避免死锁（动态）：安全性、银行家算法
- 检测与解除死锁：
 - 发现死锁：基于进程-资源图的化简
 - 解除死锁：资源剥夺、撤销进程



作业5

- 1. 有五个进程P1、P2、P3、P4、P5，它们同时依次进入就绪队列，它们的优先数和需要的处理器时间如下表：

进程	处理器时间	优先级 (数小优先级高)
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

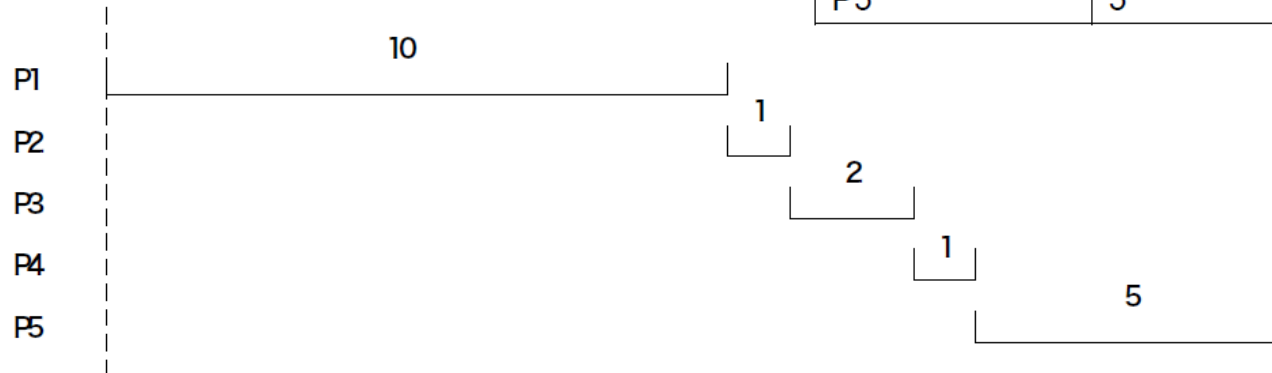
- 忽略进行调度等所花费的时间，回答下列问题：
- a. 写出采用“先来先服务”、“短作业（进程）优先”、“非抢占式的优先数”和“轮转法”等调度
- 算法，进程执行的次序。（其中轮转法的时间片为2）
- b. 分别计算上述算法中各进程的周转时间和等待时间，以及平均周转时间。



ACT

进程	处理器时间	优先级 (数小优先级高)
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

1 先来先服务

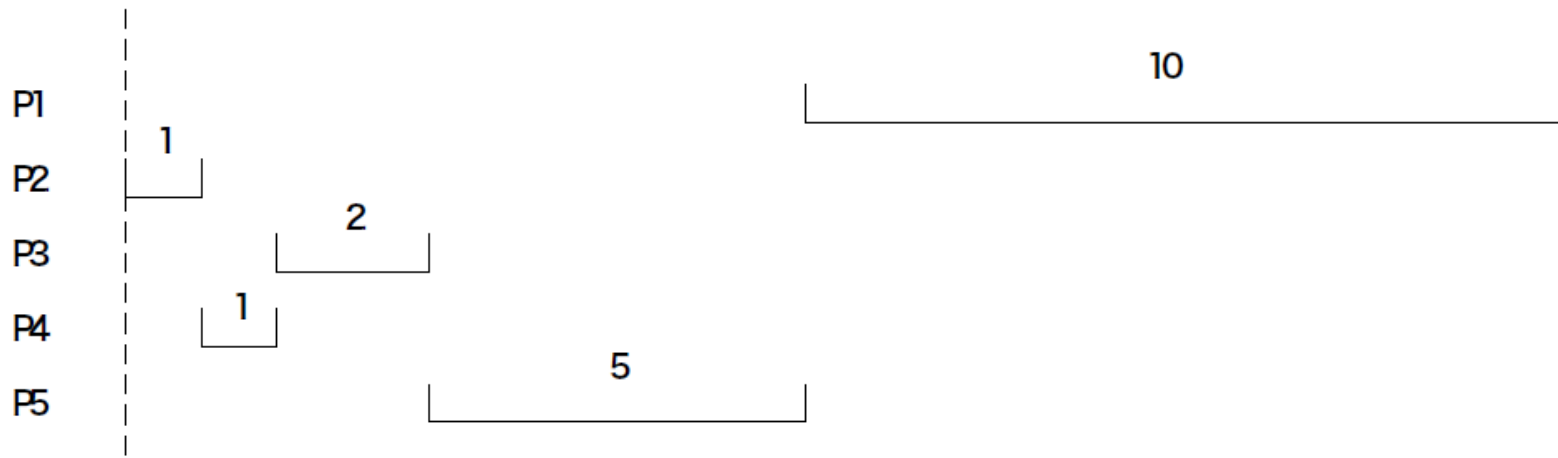


	P1	P2	P3	P4	P5	平均
周转时间	10	11	13	14	19	13.4
等待时间	0	10	11	13	14	



短作业优先

(同等长度的作业按进入就绪队列的先后顺序)

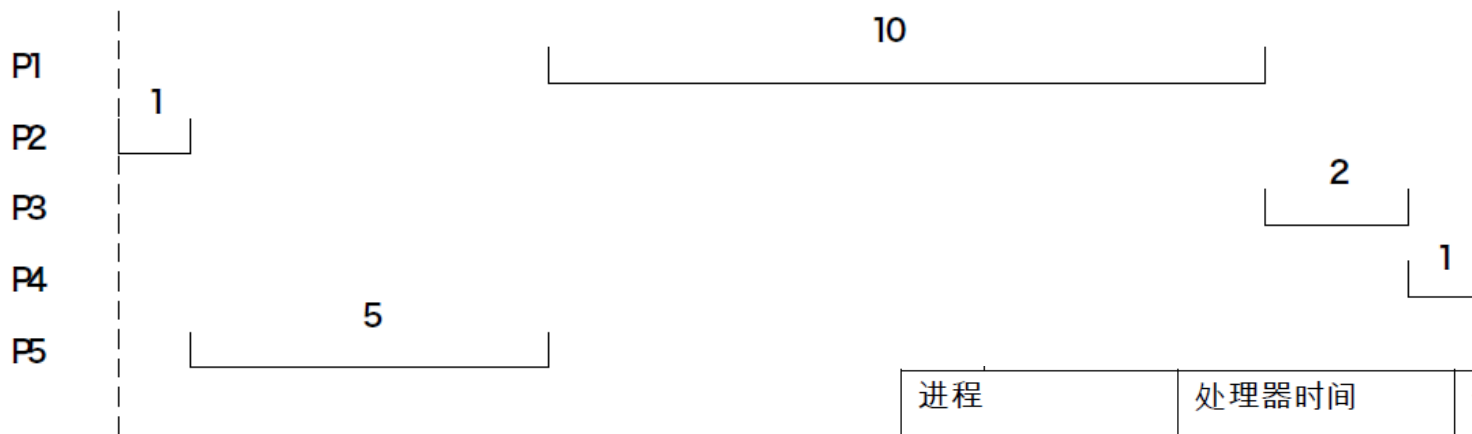


进程	处理器时间	优先级 (数小优先级高)
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2



非抢占式优先数

1) 优先级数小的优先级高



进程	处理器时间	优先级 (数小优先级高)
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2



作业5

- 3.线程的基本概念是什么？引入线程的好处是什么？
 - 减小进程切换的开销
 - 提高进程内的并发程度
 - 共享资源



作业5

- 4. 一个系统有4个进程和5个可分配资源，当前分配和最大需求如下：

	已分配资源	最大需求量	可用资源
进程A	10211	11213	00x12
进程B	20110	22210	
进程C	11010	21310	
进程D	11110	11221	

若保持该状态是安全状态，那么x的最小值是多少？

- 答：需求矩阵如下：
- A 01002
- B 02100
- C 10300
- D 00111
- 如果x为0，我们会立刻死锁。如果x是1，进程D可以分配资源运行，当它结束时，可用资源向量为 1 1 2 2 2，这时候进程A可以分配资源执行，当A结束时，可用资源向量为21433。这时候进程C可以被满足，当C执行结束后，可用资源向量为32443。最后进程B运行完毕。所以x的最小值为1。



核心内容

- 第一章 操作系统概论
- 第二章 操作系统引导
- 第三章 存储管理
- 第四章 进程机制与并发程序设计
- **第五章 输入/输出系统**
- 第六章 文件系统
- 其它：安全、分布式系统



I/O设备及控制

- I/O设备：字符设备、块设备、网络设备
- I/O控制技术（比较异同）
 - 程序控制
 - 中断驱动
 - 直接内存访问：DMA
 - 通道技术：Channel
- I/O软件的组成与分层设计
 - 设备无关性（独立性）
- 缓冲技术
- SPOOLing技术



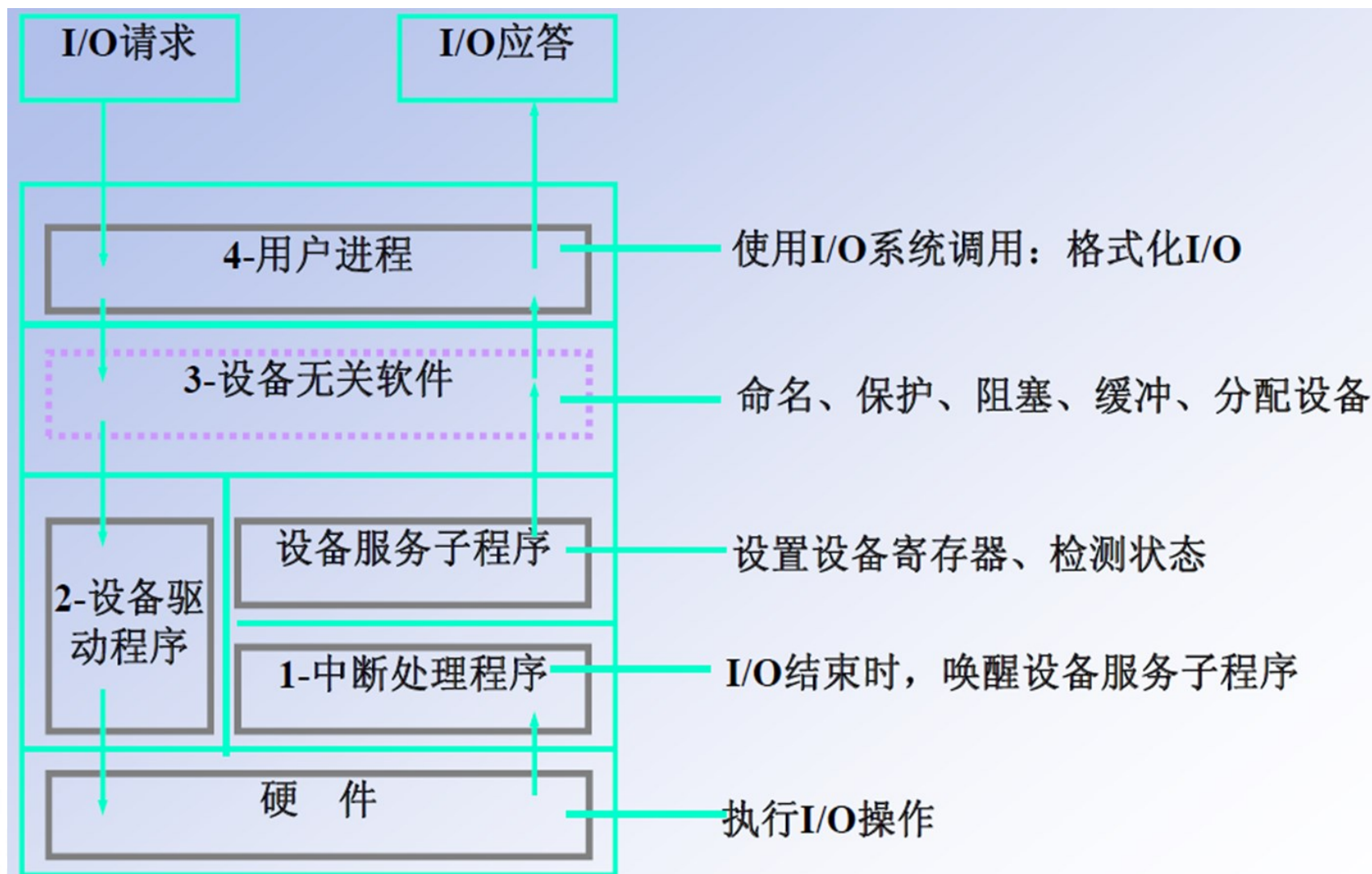
缓冲区的管理

- 缓冲区作用？
- 单缓冲区
- 双缓冲区
- 环形缓冲区
- 缓冲池



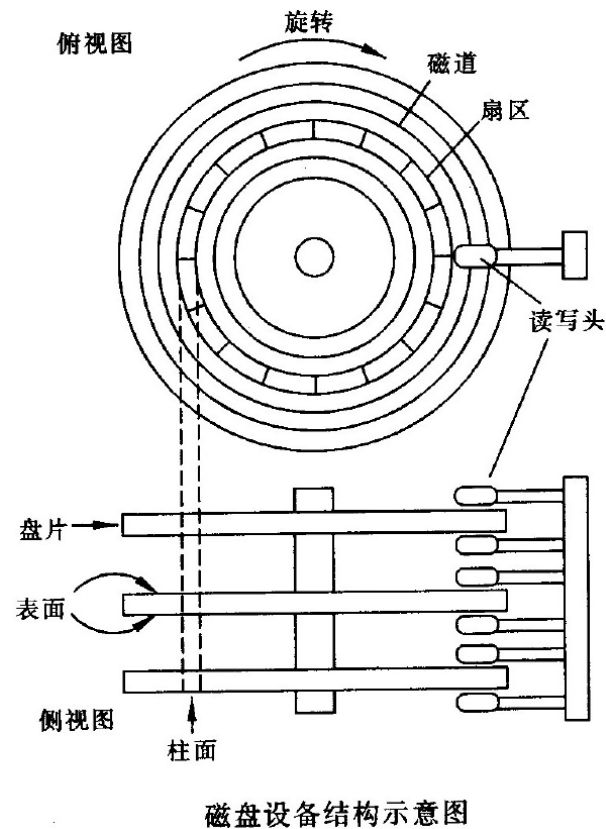


I/O管理软件



磁盘存储管理

- 磁盘的工作原理
- 磁盘访问时间
 - 寻道时间 + 旋转延迟时间 + 传输时间
- 磁盘调度算法
 - 先来先服务、最短寻道时间优先、扫描算法、循环扫描算法
- 提高I/O速度的主要途径
- 磁盘冗余阵列: RAID





核心内容

- 第一章 操作系统概论
- 第二章 操作系统引导
- 第三章 存储管理
- 第四章 进程机制与并发程序设计
- 第五章 输入/输出系统
- **第六章 文件系统**
- 其它：安全、分布式系统



基本概念

- 文件的概念
- 逻辑上，文件包括两种形式：有结构（记录式）和无结构（流式）。
- 目录的概念
- 文件系统的功能

文件系统实现技术

- 文件控制块
- 文件的逻辑结构：记录、流式
- 文件的物理结构
 - 连续文件
 - 串联文件
 - 索引文件：一级索引、多级索引
- 磁盘空间的管理
 - 空闲表、空闲链表、位示图、成组链接



文件系统

- FAT、ext2文件系统
 - 软链接（符号链接）与硬链接区别
 - 权限位操作
- E.g.假设有一个文件x存在磁盘当前目录中，依次执行：
 - `ln -s x xx`
 - `ln -s xx xxx`
 - `ln x xxxx`
 - `rm x`
 - `cat x xx xxx xxxx`
 - 会出现几个错误？
 - 在执行`touch x`，然后再`cat`呢？

```
98041871 -rw-r--r--  2 root root    0 May 25 23:20 x
98041872 lrwxrwxrwx  1 root root    1 May 25 23:21 xx -> x
98041873 lrwxrwxrwx  1 root root    2 May 25 23:21 xxx -> xx
98041871 -rw-r--r--  2 root root    0 May 25 23:20 xxxx
```



IACT

问题？

