1. 在理解的基础上，必须掌握的基本概念（涉及到填空、选择、简答，最好做做近五年考研题）
2. 内核、强内核、微内核；

内核：内核，是一个操作系统的核心。是基于硬件的第一层软件扩充，提供操作系统的最基本的功能，是操作系统工作的基础，它负责管理系统的进程、内存、设备驱动程序、文件和网络系统，决定着系统的性能和稳定性。

强内核：将操作系统的的主要功能模块都作为一个紧密联系的整体运行在核心态，从而为应用提供高性能的系统服务。缺点：维护成本高。

微内核：将内核中最基本的功能保留在内核，而将那些不需要在核心态执行的功能移到用户态执行，从而降低了内核的设计复杂性。缺点：效率较低。

1. 中断（异常与中断的区别）、软中断与硬中断实现；

中断：也称外中断，指来自CPU执行指令以外的事件的发生。（如IO中断、时钟中断）

异常：也称内中断，指源自CPU执行指令内部的事件。（如程序的非法操作码、地址越界、算术溢出、虚存系统的缺页以及专门的陷入指令等引起的事件）

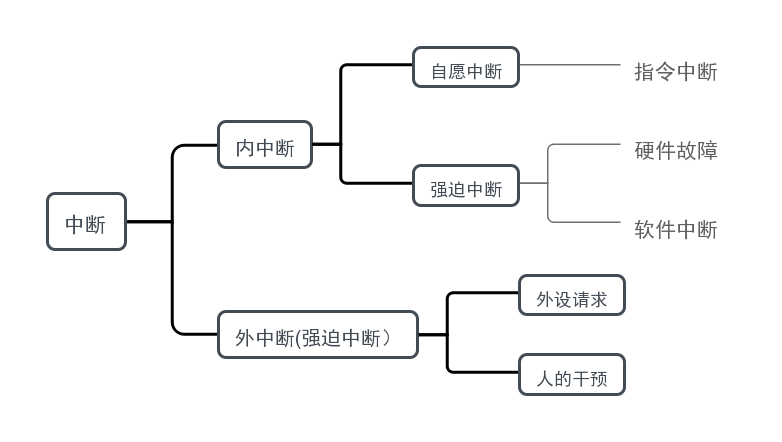
中断和异常的区别：CPU处理中断的过程中会屏蔽中断，不接受新的中断直到此次中断处理结束。而异常的发生并不屏蔽中断，可以接受新的中断。

异常会引起中断，中断未必由异常引起

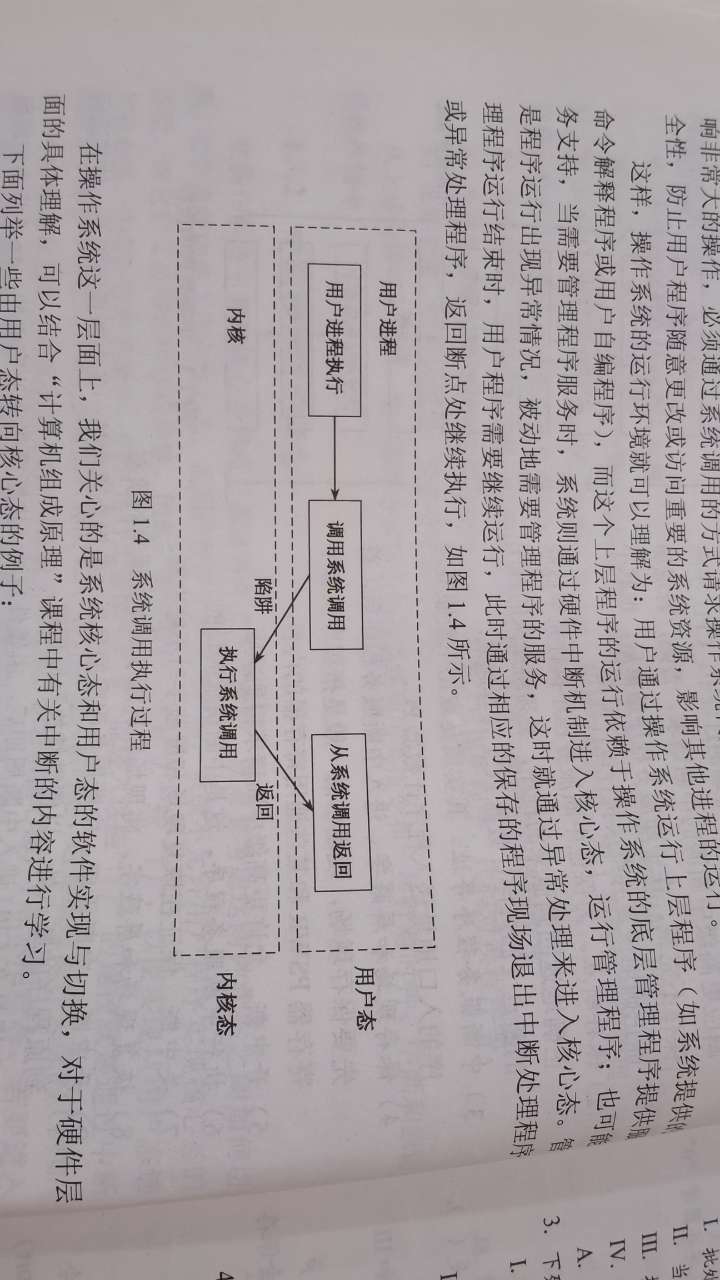
软中断是系统内部为了满足实时系统的需要，放到中断之后来完成

硬中断是由外设产生的。

陷阱指令可以使执行流程从用户态陷入内核，并把控制权转移给操作系统，使得用户程序可以调用内核函数和使用硬件从而获得操作系统所提供的服务。



1. 系统调用与库函数的区别、系统调用的过程；



系统调用是是操作系统提供的用户接口之一。

系统调用把应用程序的请求传给内核，调用相应的内核函数完成所需的处理并将处理结果返回给应用程序。

系统调用与库函数的区别：

1. 库函数调用函数库中的一段程序，系统调用是操作系统的一个入口点；
2. 库函数在用户地址空间执行，系统调用在内核地址空间执行；
3. 库函数的运行时间属于“用户时间”，系统调用的运行时间属于“系统时间”；
4. 库函数属于过程调用开销小，系统调用需要在用户空间和内核上下文环境间切换，开销较大；
5. 库函数是有缓冲的，系统调用是无缓冲的；
6. 系统调用依赖于平台，库函数并不依赖
7. 操作系统的四种结构；

四种结构：简单结构、分层结构、微内核结构、模块化结构

1. 进程、进程与线程的联系与区别、进程与程序的区别；

进程是具有独立功能的程序关于某个数据集合上的一次运行活动，是资源分配的基本单位，是CPU调度的一个单位。一个进程包括多个线程。

线程是程序执行流的最小单元，是系统独立调度和分配CPU（独立运行）的基本单位。

进程与线程的区别：

(1) 调度：线程作为调度的基本单位，同进程中线程切换不引起进程切换，当不同进程的线程切换才引起进程切换；进程作为拥有资源的基本单位。

(2) 并发性：一个进程间的多个线程可并发。

(3) 拥有资源：线程仅拥有隶属进程的资源；进程是拥有资源的独立单位。

(4) 系统开销：进程大；线程小。

进程与程序的主要区别：

(1) 程序是永存的，进程是暂时的

(2) 程序是静态的观念，进程是动态的观念

(3) 进程由三部分组成：程序+数据+进程控制块（描述进程活动情况的数据结构）

(4) 进程和程序不是一一对应的，一个程序可对应多个进程即多个进程可执行同一程序，一个进程可以执行一个或几个程序

(5) 进程特征：动态性、并发性、调度性、异步性、结构性

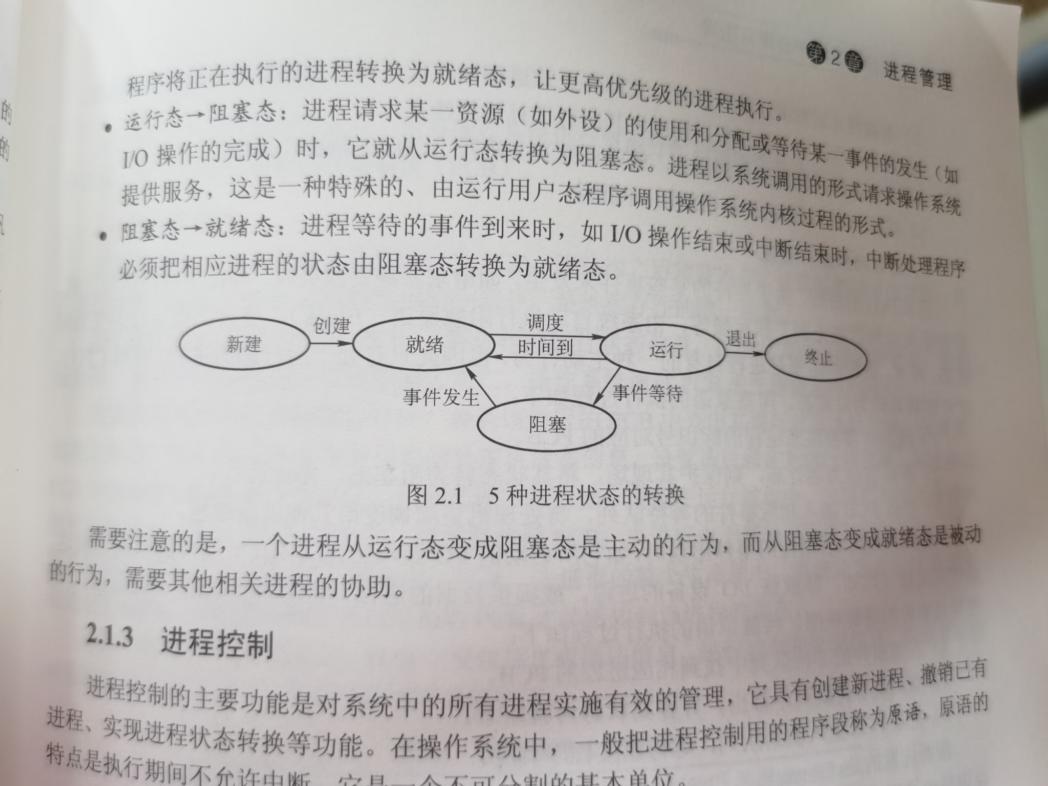
(6) 程序执行具有固定时序、顺序性、封闭性、可再现性的特点。

1. PCB的构成、进程的创建过程、进程的五种基本状态相互转换过程（进程长期调度、中期调度、短期调度）、fork()函数的返回值、父进程与子进程关系；

PCB名为进程控制块，每个进程都有一个PCB，是进程存在的唯一标志，其中一般会有进程标识符PID，进程当前状态，队列指针等

进程的创建过程：新进程的创建，首先在内存中为新进程创建一个task\_struct结构，然后将父进程的task\_struct内容复制其中，再修改部分数据。分配新的内核堆栈、新的PID、再将task\_struct 这个node添加到链表中。所谓创建，实际上是“复制”。子进程刚开始，内核并没有为它分配物理内存，而是以只读的方式共享父进程内存，只有当子进程写时，才复制。（通过fork()函数系统调用，可创建新进程）

进程的五种基本状态：新建、就绪、运行、阻塞、终止



长期调度：又称为作业调度或高级调度，这种调度将已进入系统并处于后备状态的作业按算法选择一个或一批，为其建立进程，并进入主机，当该作业执行完毕时，还负责回收系统资源，在批处理系统中，需要有作业调度的过程，以便将它们分批地装入内存，在分时系统和实时系统中，通常不需要长期调度。它的频率比较低，主要用来控制内存进程的数量。**（从缓冲期到内存）**

中期调度：又称为交换调度。它的核心思想是能将进程从内存或从CPU竞争中移出，从而降低多道程序设计的程度，之后进程能被重新调入内存，并从中断处继续执行，这种交换的操作可以调整进程在内存中的存在数量和时机。其主要任务是按照给定的原则和策略，将处于外存交换区中的就绪状态或等待状态的进程调入内存，或把处于内存就绪状态或内存等待状态的进程交换到外存交换区。**（从内存移到磁盘）**

短期调度：又称为进程调度、低级调度或微观调度。这也是通常所说的调度，一般情况下使用最多的就是短期调度。它的主要任务是按照某种策略和算法将处理机分配给一个处于就绪状态的进程，分为抢占式和非抢占式。**（就绪队列选中到CPU）**

fork函数在父进程中返回子进程的ID，在子进程中返回0，负数则是出错

1. 多线程的三种模型、多线程编程的优点；

多线程的三种模型：

①多对一模型：将许多用户级线程映射到一个内核线程；

②一对一模型：将每个用户线程映射到一个内核线程；

③多对多模型：将多个用户线程映射到同样数量或更小数量的内核线程上。

多线程编程的优点：

①资源利用率提升，程序处理效率提高

②代码相对简单

③软件运行速度提升

1. 进程同步与互斥、信号量的三种原子操作（初始化、P操作、V操作）的含义、临界区、临界资源、互斥信号量、如何解决临界区问题；

同步：是进程间共同完成一项任务时直接发生相互作用的关系。

互斥：排它性访问，即竞争同一个物理资源而相互制约。

P操作：(1) P操作一次，S值减１（请求分配资源）；

1. 如果S≥0，则该进程继续执行；如果S＜0表示无资源，则该进程的状态置为阻塞态。

V操作：(1) V操作一次，S值加1

(2) 如果S＞0，表示有资源，则该进程继续执行；如果S≤0，则释放信号量队列上的第一个PCB所对应的进程（阻塞态改为就绪态），执行V操作的进程继续执行。

临界区：每个进程中访问临界资源的那段程序。

临界资源：一次仅允许一个进程使用的共享资源。

信号量物理意义：

(1) 信号量的值大于0：表示当前资源可用数量

　　 　　　 小于0：其绝对值表示等待使用该资源的进程个数

(2) 信号量初值为非负的整数变量，代表资源数。

(3) 信号量值可变，但仅能由Ｐ、Ｖ操作来改变。

如何解决临界区问题：

①Peterson算法：适用于两个进程在临界区和剩余区间交替执行，该算法需要在两个进程之间共享数据项；只解决互斥问题，不解决同步问题

②硬件同步

解决进程临界区问题的条件：互斥、前进、有限等待

1. 死锁、死锁产生的原因、死锁产生的四个必要条件、如何避免死锁、如何预防死锁、如何判断死锁；

死锁：多个进程因循环等待资源而造成无法执行的现象。

死锁产生的原因：系统资源不足和进程推进顺序非法

死锁产生的四个必要条件：互斥、不剥夺、请求并保持、循环等待

如何避免死锁：系统安全状态、安全性算法、银行家算法

如何预防死锁：打破产生死锁的四个必要条件的一个或几个

如何判断死锁：资源分配图、死锁产生的条件

死锁恢复：进程终止、资源抢占（选择一个牺牲品、回滚、饥饿）

资源分配图：如果分配图没有环，系统没有进程死锁；若有环，可能存在死锁。如果每个资源类型只有一个实例，则环是死锁存在的充要条件。

1. 连续内存分配会产生外部碎片、如何解决外部碎片、分页管理与分段管理各自的优缺点、抖动、Belady’s异常、文件的绝对路径和相对路径、设置当前目录的作用。

连续内存分配会产生外部碎片

如何解决外部碎片：紧缩

分页管理：

优点：1.内存利用率高 2.实现了离散分配 3.便于存储控制访问 4.无外部碎片

缺点：1.需要硬件支持 2.内存访问效率下降 3.共享困难 4.内部碎片

分段管理：

优点：1.便于程序模块化处理和处理变换的数据结构 2.便于动态链接和共享 3.无内部碎片

缺点：1.需要硬件支持 2.需要采用拼接技术 3.分段的最大尺寸受到主存可用空间的限制 4.有外部碎片

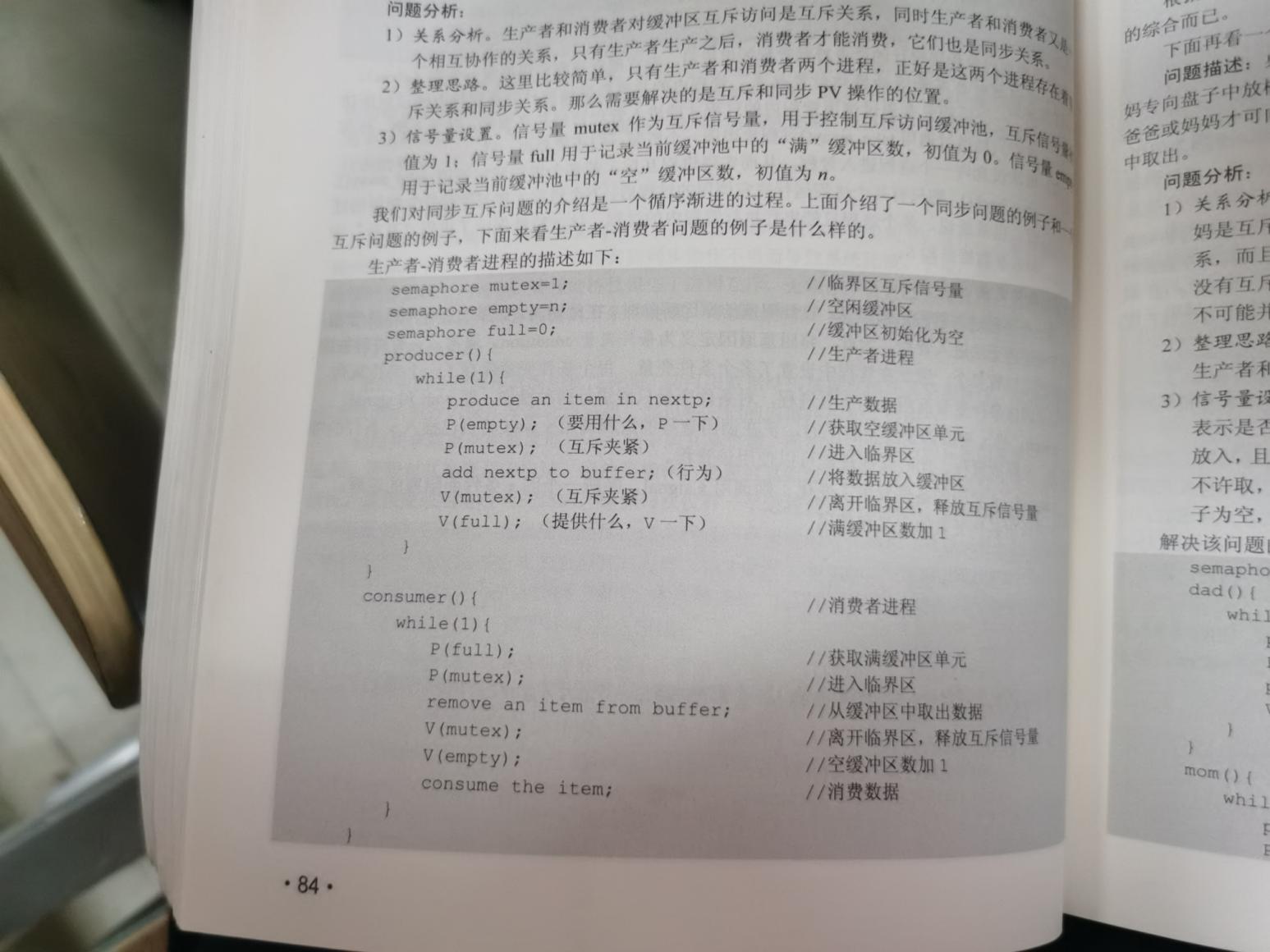
抖动：在页面置换过程中，刚刚换出的页面马上又要换入主存，刚刚换入的页面马上又要换出主存，这种频繁的页面调度行为称为抖动。

Belady’s异常：FIFO置换算法的缺页率可能会随着所分配的物理块数的增加而增加的奇怪现象。

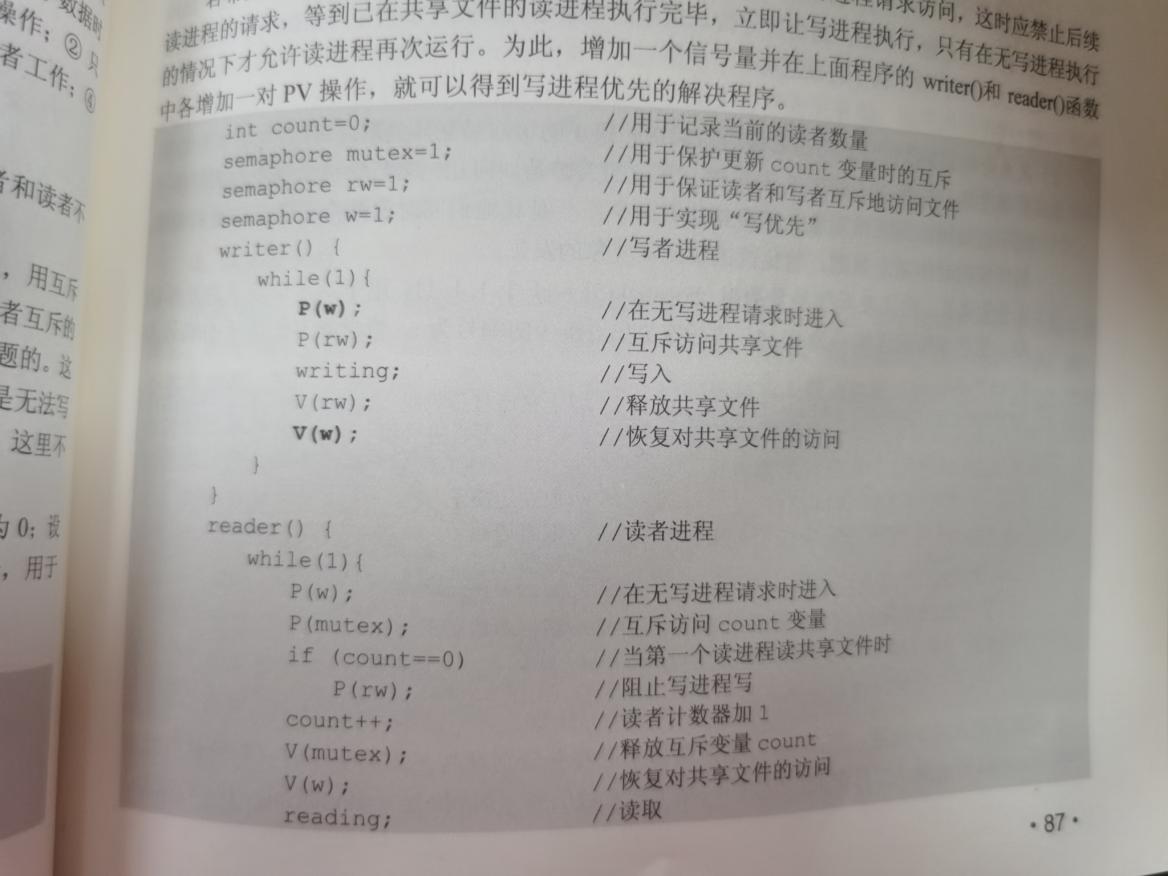
设置当前目录的作用：加快文件的检索速度

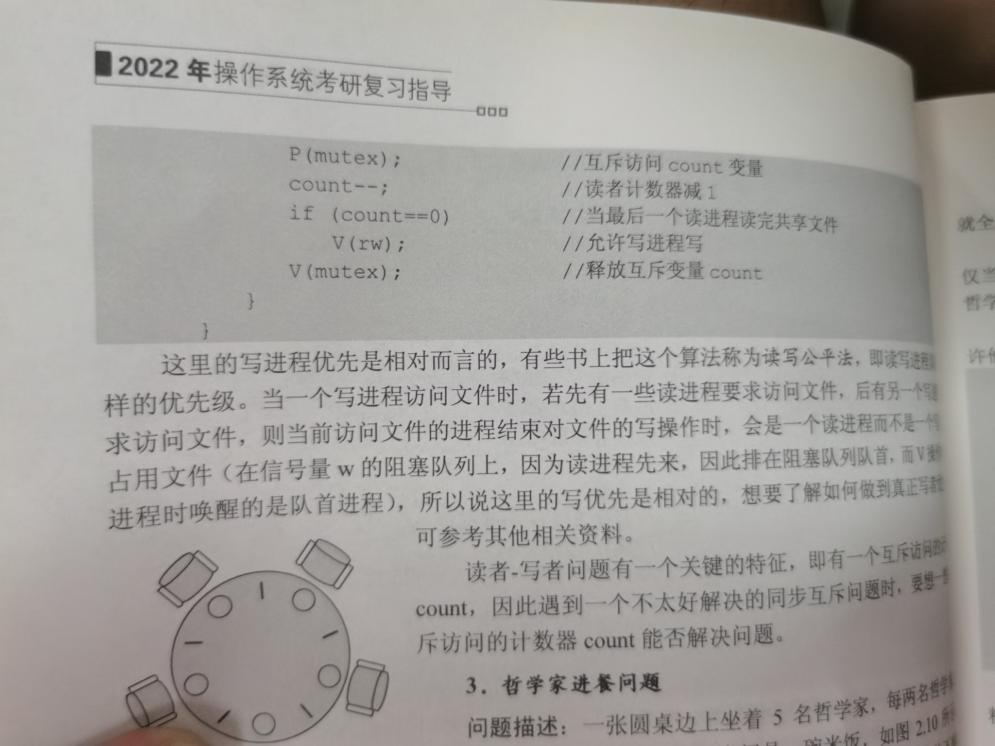
1. 必须掌握的基本原理（计算题）
2. CPU调度算法（SJF\SRTF\RR\FCFS）;
3. 银行家算法（安全系列算法，资源请求算法）；
4. 逻辑地址与物理地址的相互转化计算；
5. 有关快表（TLB）计算有效访问时间；
6. 页面替换算法（FIFO\OPT\LRU）
7. PV操作（伪代码编程）：能够利用进程同步的三个经典算法完成进程同步PV操作
8. 生产者-消费者问题（95%的概念会涉及各种考试题中）；

注意：先申请非互斥资源 P(empty) ，再申请互斥资源 P(mutex)，如果将这两个写反了一定会产生死锁，请思考。

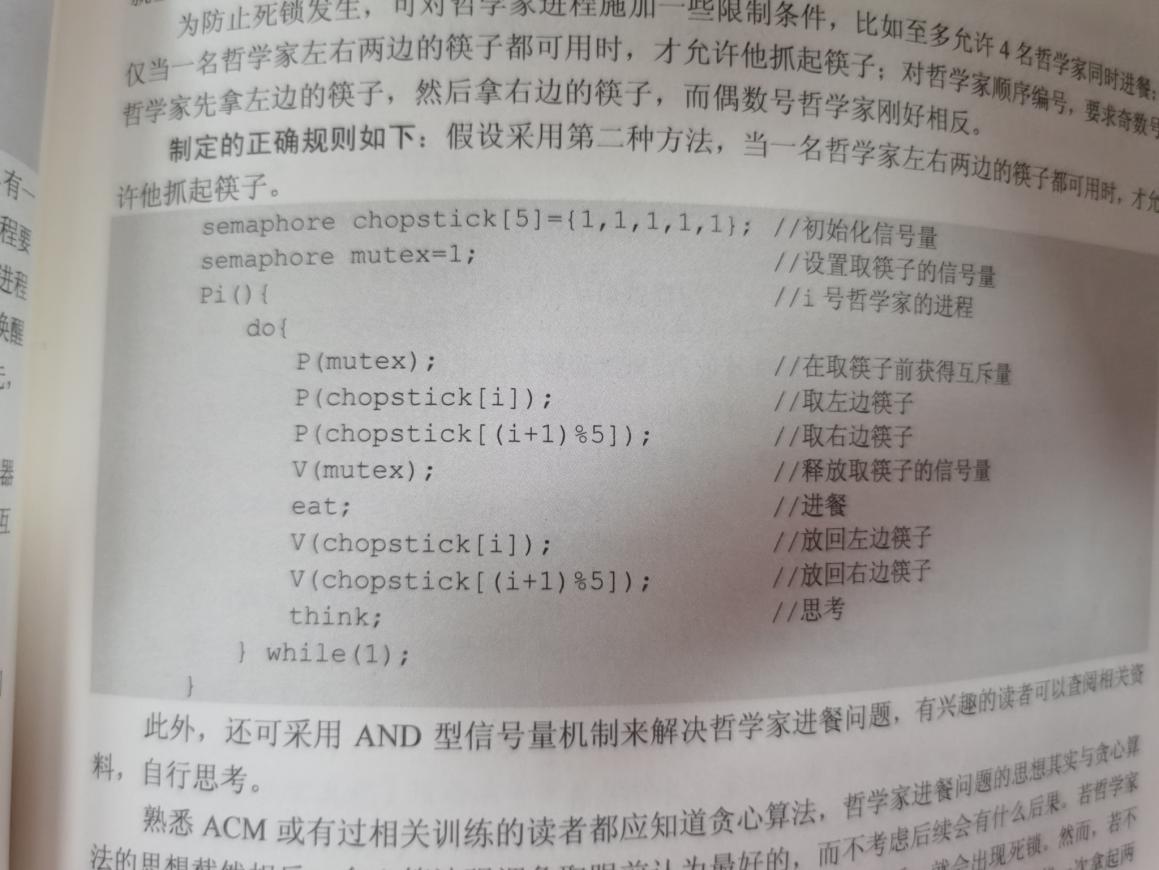


1. 读者-写者问题（比喻成过独木桥问题）；注意：车上桥时需要给对方发信号P(mutex),最后一辆车下桥时页需要给对方发信号V(mutex)。每一方自己内部排队用于控制计数页需要用到互斥信号量mutex1=1,mutex2=1.





1. 哲学家进餐问题:需要用到数组定义信号量，当同时用餐人数筷子总数-1时不会产生死锁；或将哲学家座位进行编号，分成奇数位和偶数位，处于奇数位先取左边，再取右边筷子。而偶数位先取右边，再取左边筷子不会发生死锁。



1. 下周上完最后一章内容：文件管理，课堂上继续复习完没有讲完的内容。
2. 请各位同学认真复习，祝愿大家都能考出自己理想的成绩！