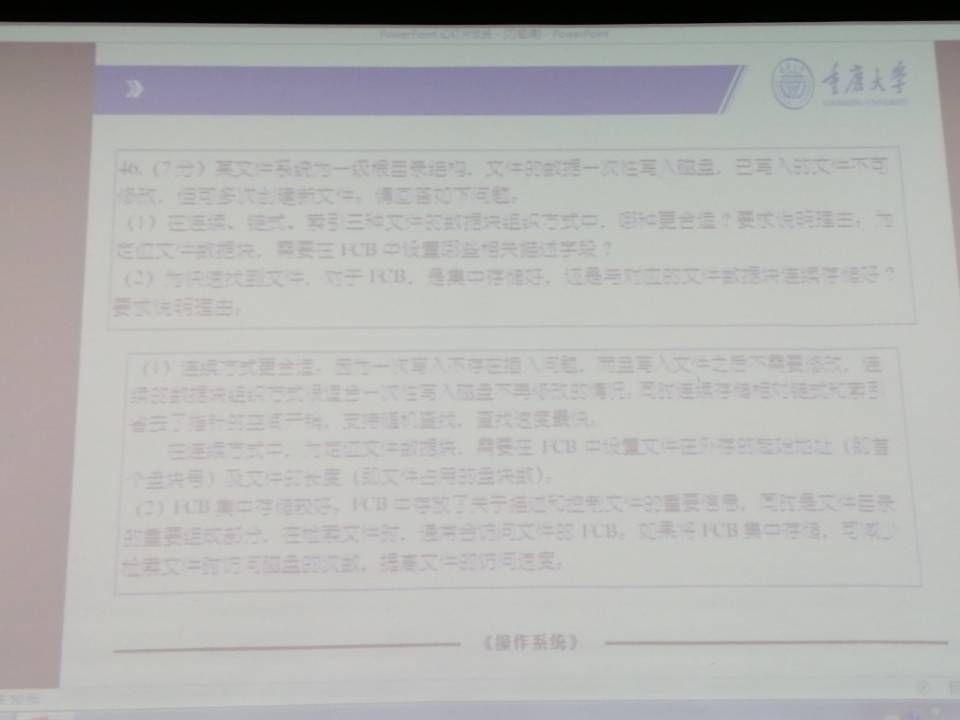
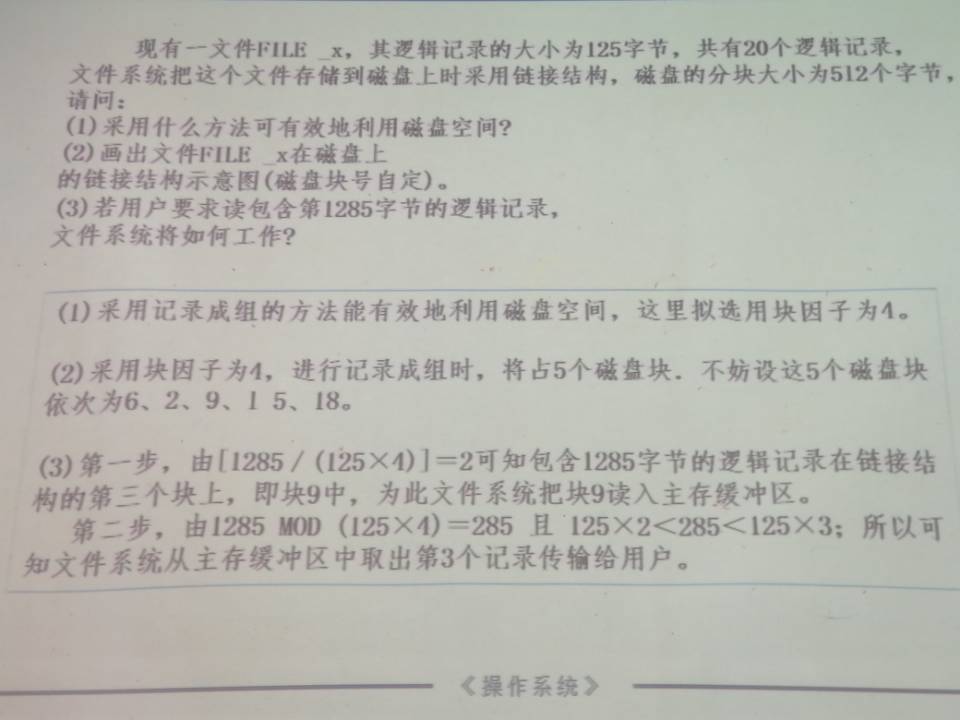
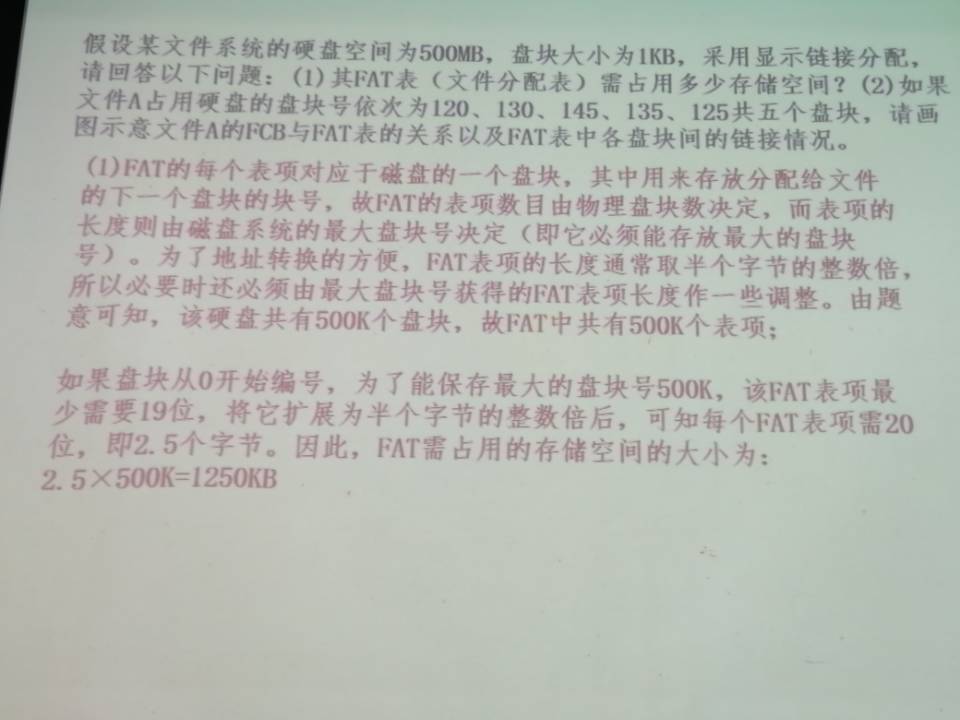
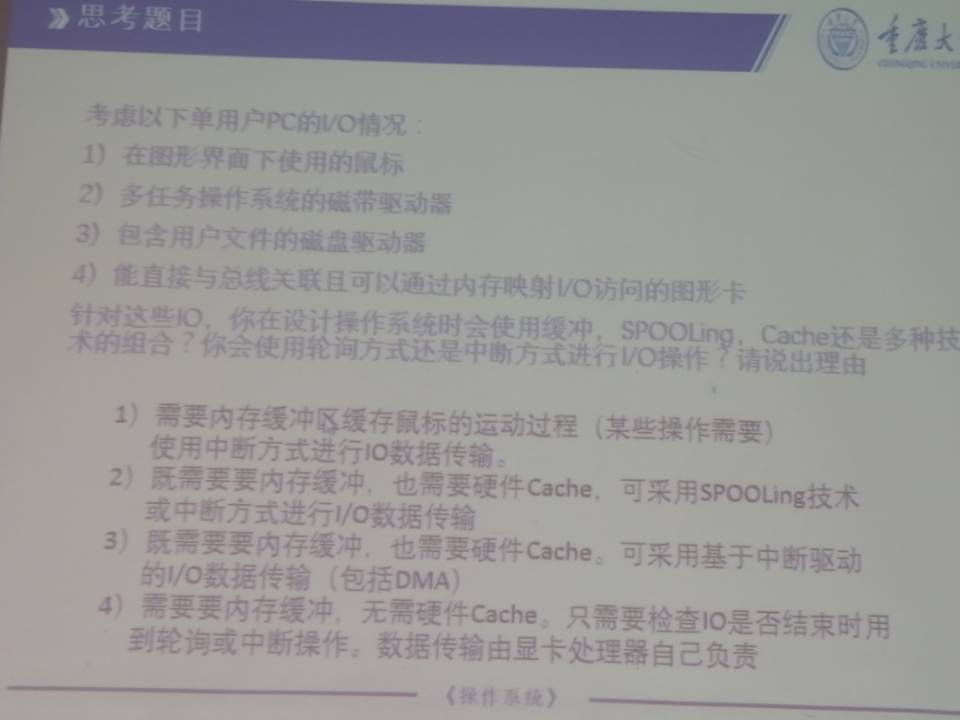
## 文件管理



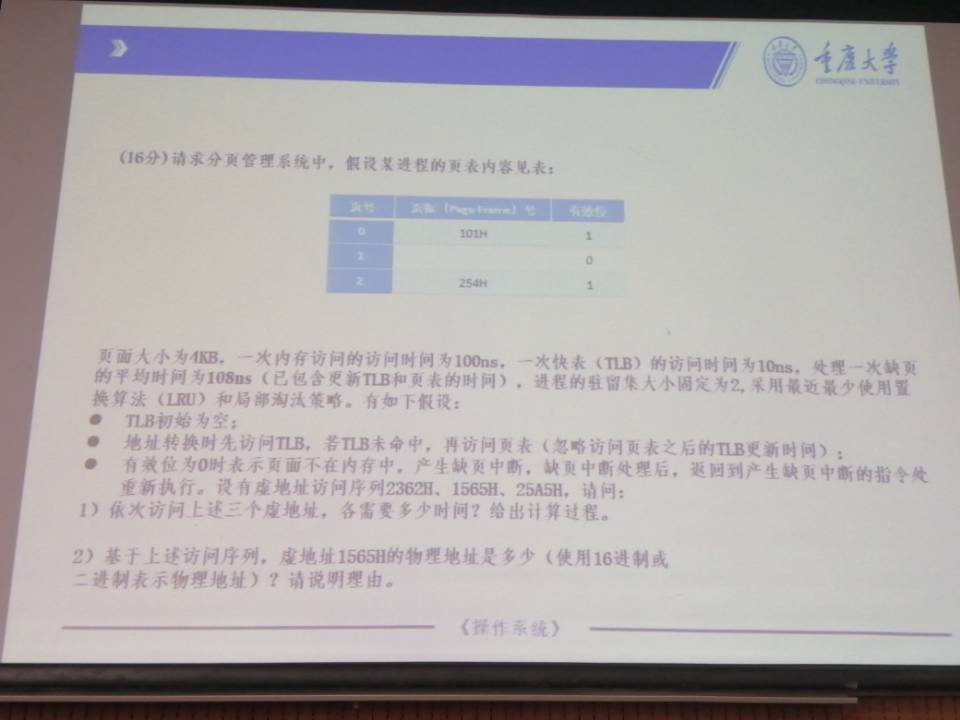


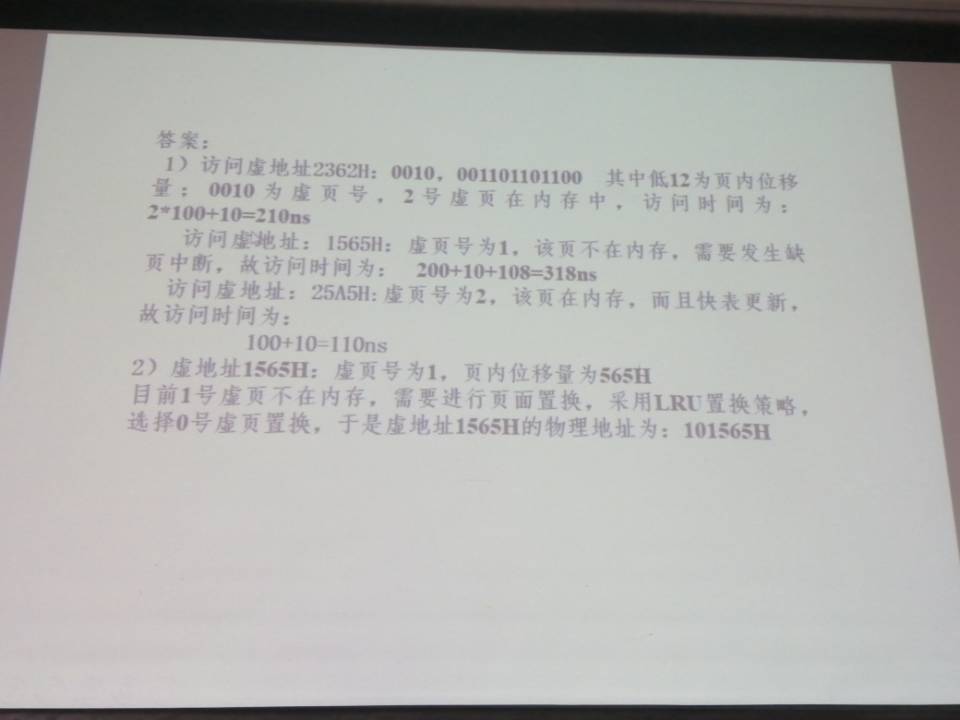


## IO

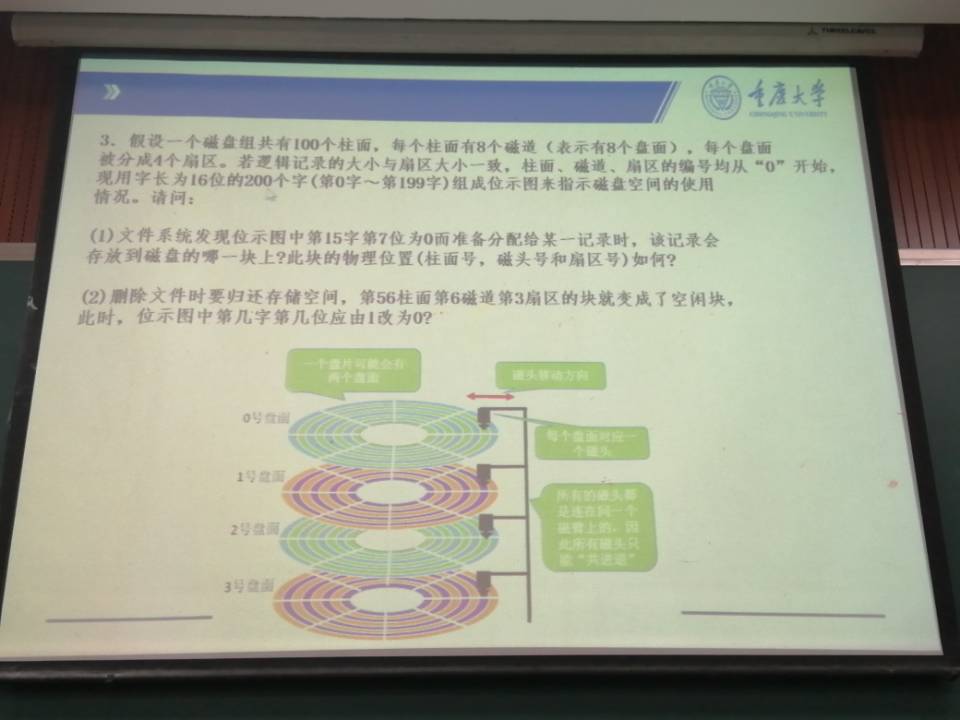


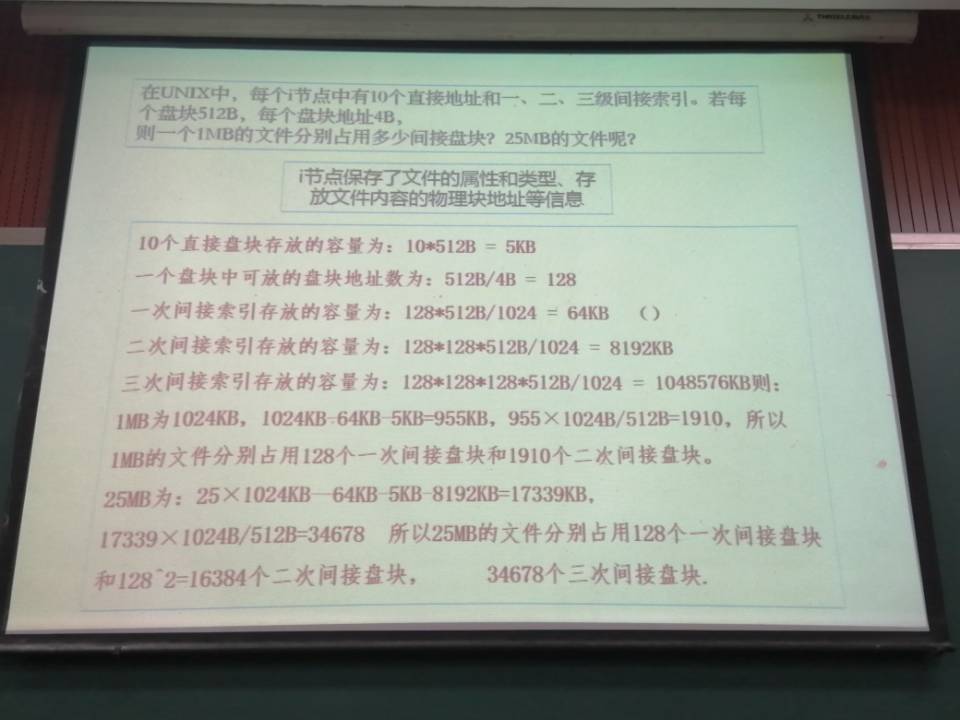
## 分页管理

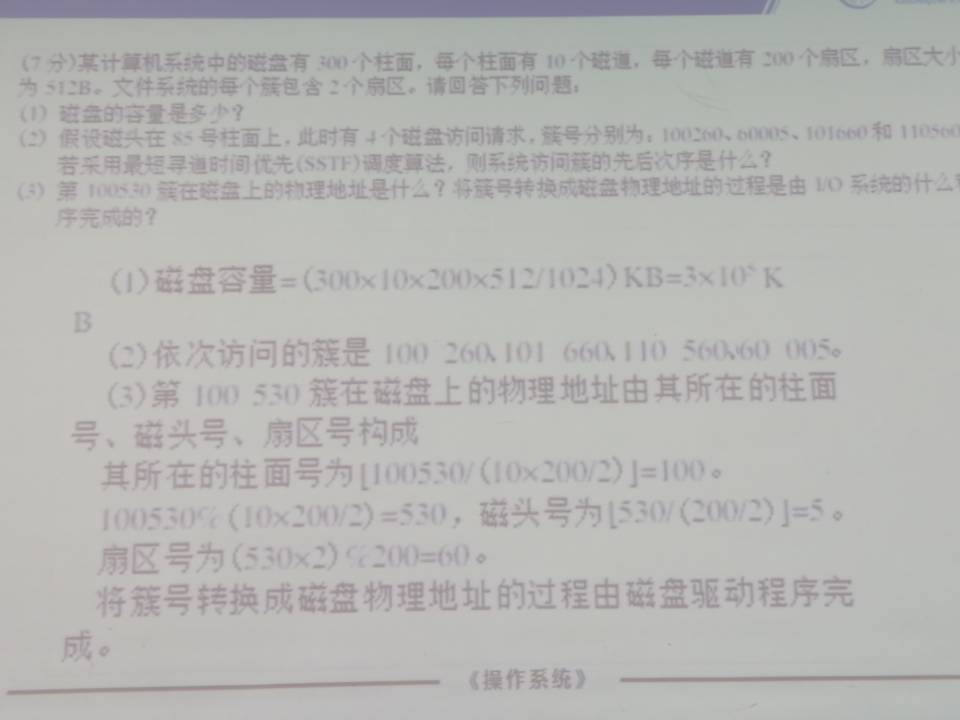


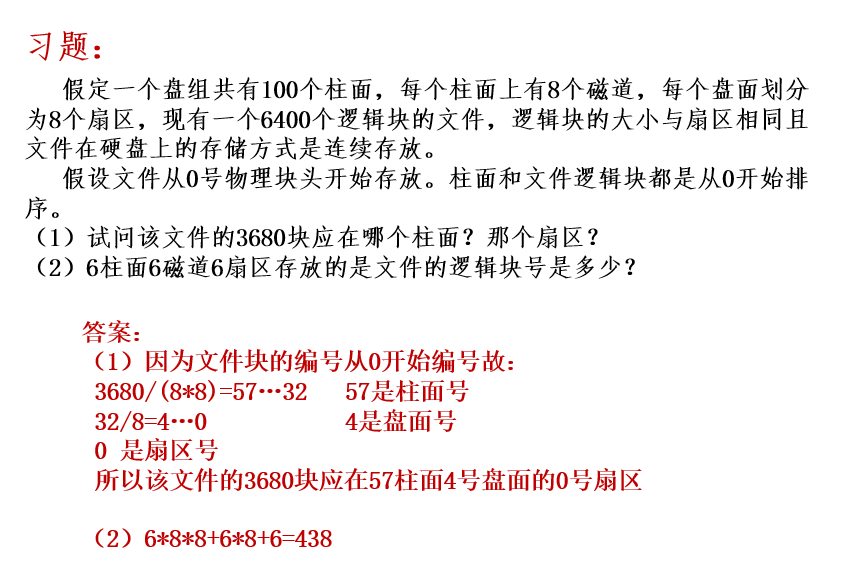


## 磁盘结构

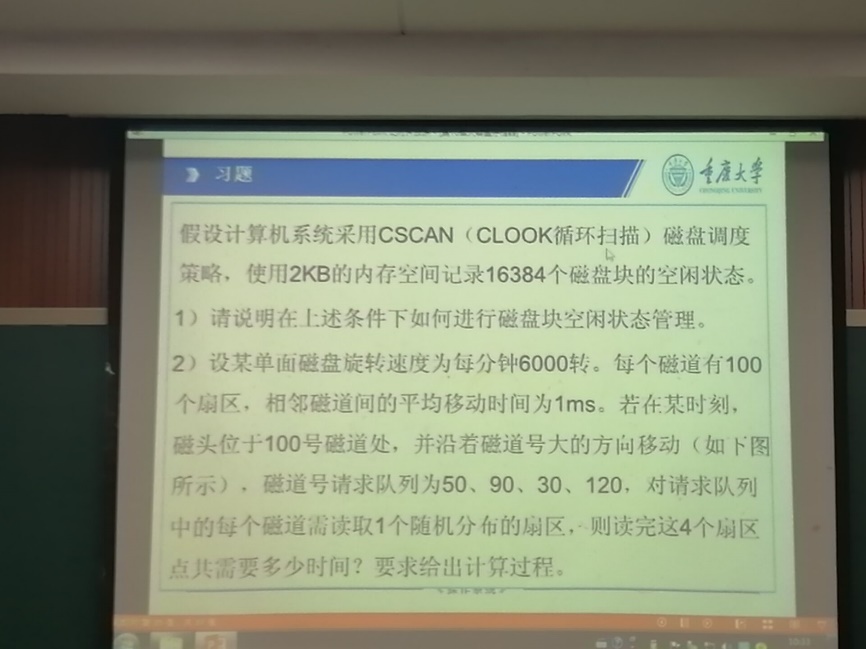


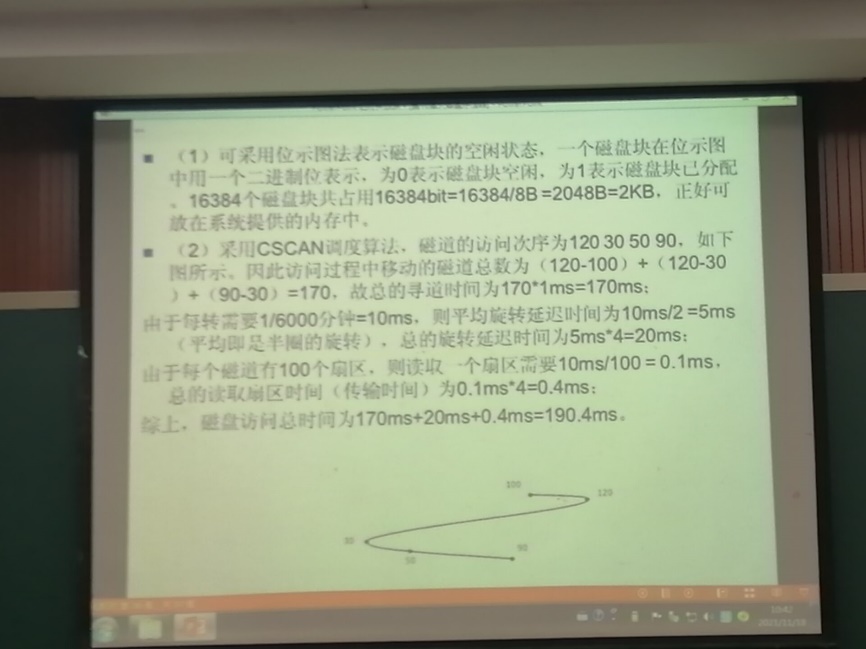




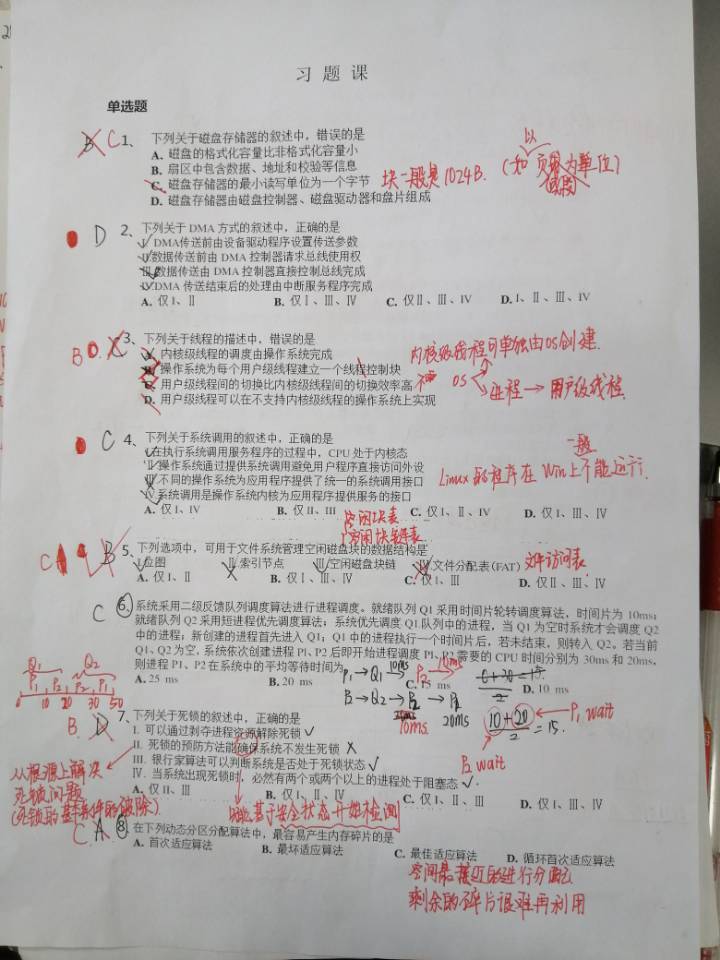


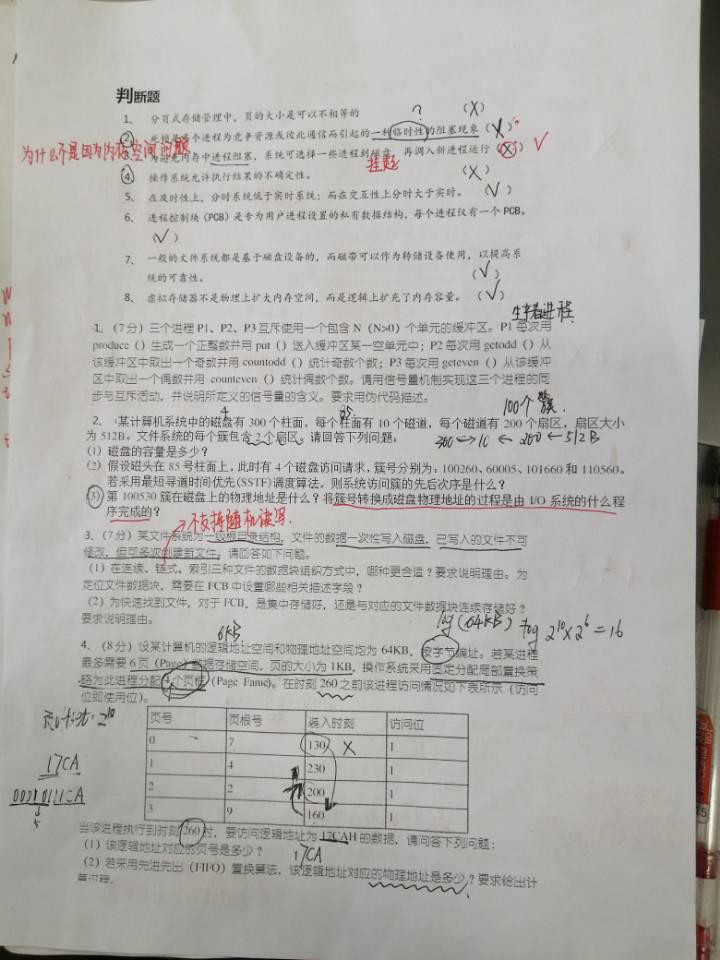
## 内存

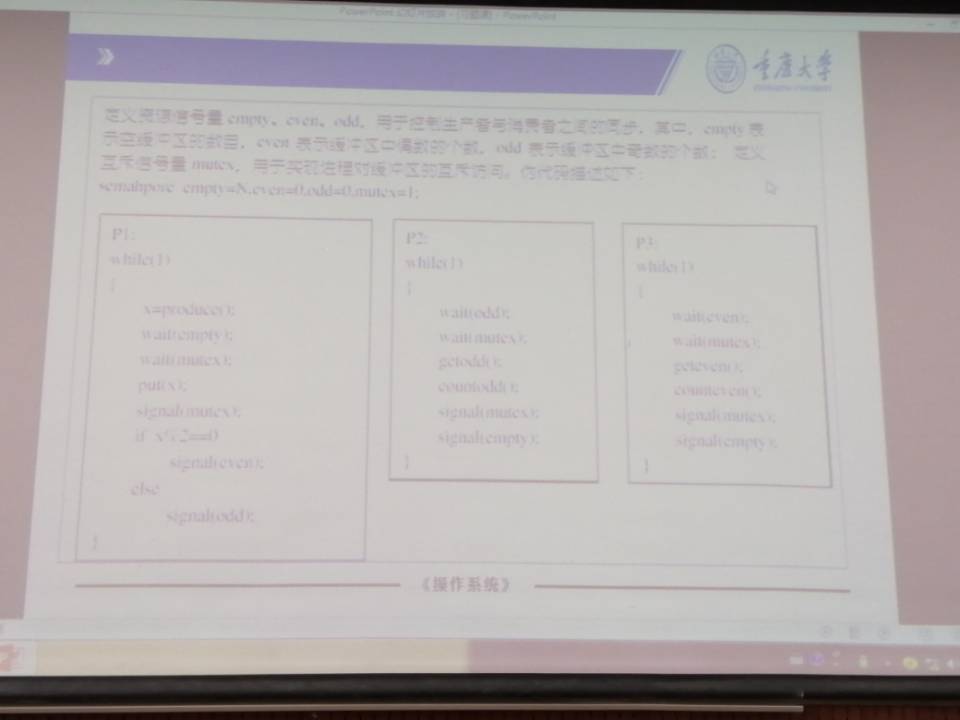




## 考题示例







# 期末试题集锦

## 2017-2018期中

Monitor怎么理解？

( F) The monitor can be used to control exclusive access of critical resources, but it cannot be used for complicated process synchronization.

系统调用和API的关系：

( F) The system calls provide a programming interface to the operating system, they are called API.

系统调用概念：

( D) which of following function is system call ()

A printk B pthread\_create C printf D wait

神奇的考点：

1. ( A ) refers to the POSIX standard(IEEE 1003.1c) defining an API for thread creation and synchronization.
2. Pthreads B. Threads C. Process D. Plib

判断

（） A relocation register is used to check for invalid memory addresses

generated by a CPU

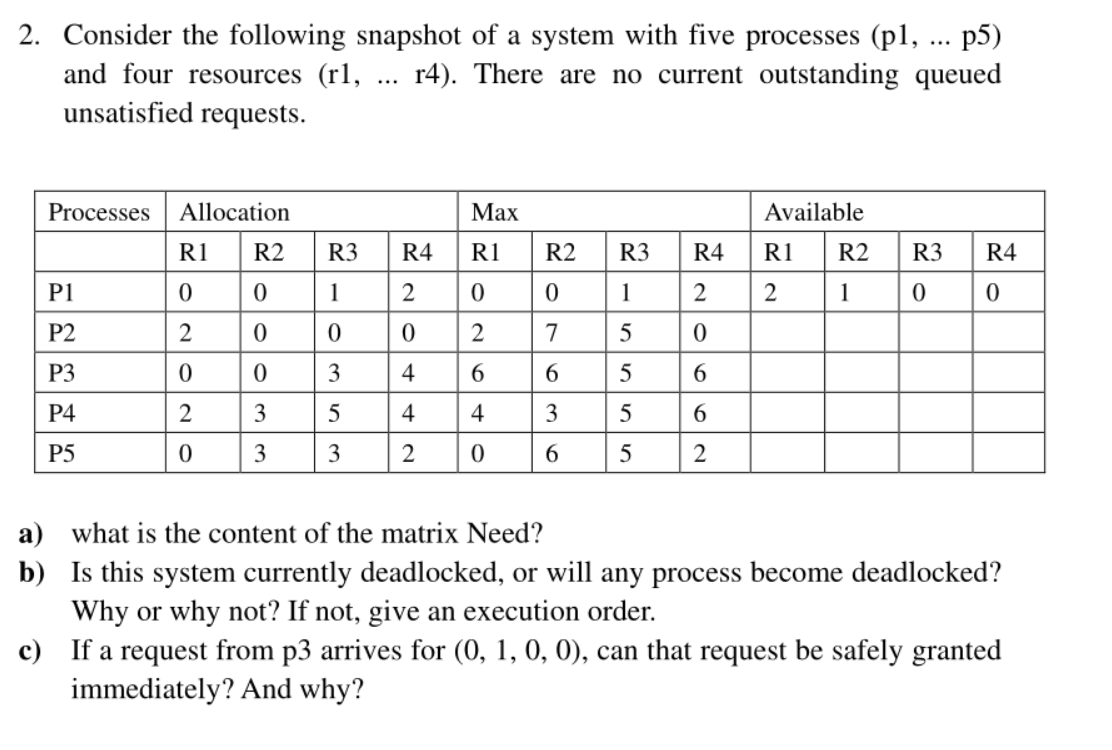
（）Monitors are a theoretical concept and are not practiced in modern

programming languages

为什么需要内核级线程和用户级线程两种模式？

死锁恢复的方案及其优缺点

这里的p3分配出去的还能收回吗？→ 能的，应该也是算在Need里面，分配出去后，Need[p3]相应的减少[0,1,0,0]



# 知识点集锦

进程调度算法必须是非抢占式的：FCFS

## 2021.12猜简答题

### 系统调用和API的关系：

区别：API是函数定义，描述函数功能，与程序员直接交互；系统调用，是内核接口，通过中断向内核发起请求，并实现内核提供的某些服务，比API更注重细节也更难懂，一般不与程序员直接交互。

联系：一个需要使用内核的API可能需要一个或多个系统调用来实现。

区别：

API是函数的定义，规定了函数的功能，与内核无直接关系，而系统调用是通过中断向内核发出请求，实现内核提供的某些服务

API能让程序的可移植性变高

系统调用比API更注重细节且更加难用

联系

API通常为应用程序员调用实际的系统调用

一个API可能会需要一个或多个系统调用来完成特定的功能。

### 进程与程序：

区别：

进程，为动态执行过程，可并发，可独立运行；

程序，是指令的有序集合，为静态实体，不可并发，需要在创建的进程中运行。

联系：

进程由PCB、程序段、数据段组成

程序需要在创建的进程中运行

### 进程与线程：㊗

在引入线程的系统中，进程与线程有以下区别与联系：

1. 调度上

线程是调度和分派的基本单位，进程是资源分配的基本单位，同一进程内的线程切换不会引起进程切换。

2. 并发上

不同进程之间、同一进程不同线程之间都可以并发执行，能更有效的使用系统资源，具有更好的并发性。

3. 系统开销上

进程是拥有资源的基本单位，线程只拥有少量资源，但它可以访问其隶属进程的资源。故线程的创建、终止、切换的系统开销都比进程小。

4. 通信效率上

同一进程内的线程共享内存和文件，处在统一地址空间里，它们之间的通信无需调用内核，从而提高了通信的效率。

### 请简述写时复制技术

父进程创建子进程时，父子进程共享页面，当任一进程修改数据时复制页面并在复制的页面上修改，未修改的页面仍然可以继续共享。

这是对程序性能的优化，可以延迟甚至是避免内存拷贝，当然目的就是**避免不必要的内存拷贝**

### 产生死锁的必要条件㊗

1. 互斥条件：每个资源每次只能分配给一个进程使用

2. 请求并保持条件：进程占有一部分资源后还允许继续申请新的资源

3. 不可剥夺条件：任一个进程不能从另一个进程那里抢占资源，只能由占用进程自己释放

4. 循环等待条件：存在一条循环的等待序列

### 死锁解除的三种方法及其优缺点分析：

书上的：

撤销所有死锁进程

把所有死锁进程恢复到前面的某个检查点

有选择地撤销死锁进程，直到不存在死锁

最小代价地剥夺资源，直到不存在死锁

1. 资源剥夺法：基于最小代价原则剥夺进程的资源，每剥夺一个，都要调用死锁检测算法。

优点：保证进程的原有资源尽可能少的被剥夺，尽可能少的重复再次申请资源

缺点：需要计算进程的重要性以确定要剥夺的资源

2. 撤销进程法：基于最小代价原则撤销进程，每撤销一个，都要调用死锁检测算法。

优点：简单直接，有效。

缺点：需要计算进程的重要性以确定要撤销的进程；撤销的进程需要重新运行，增大了系统的负担；

3. 进程回退法：所有死锁进程恢复到前期定义的某个检查点

优点：能最快的解除死锁局面，特别是也因为一些巧合因素叠加而导致的死锁局面

缺点：需要系统有构建重新运行和重新启动的机制；运行开销大；还可能再次发生死锁；

### 虚拟存储技术的基础：局部性原理

程序在执行过程中的一个较短时间内，所执行的指令地址或操作数地址分别局限于一定的内存区域中。可以只将部分指令（代码）和数据装入内存并允许它执行。

### TODO FAT㊗

FAT（显示链接文件）

• 画出FAT？

• FAT有多少项？每项多少位？

·FAT有多大

### 微内核结构和模块化结构㊗