## 操作系统（李志军）

### L 1 什么是操作系统

#### 1. 计算机构成

![1](D:\Offer\操作系统\笔记\操作系统（李治军）图片\1.jpg)

计算机操作系统需要管理的硬件包括CPU/内存/终端/磁盘/文件以及网络/电源和多核管理：

![2](D:\Offer\操作系统\笔记\操作系统（李治军）图片\2.jpg)

### L 2 L3揭开操作系统启动

图灵机加上冯诺依曼的存储程序思想，计算机通过取指执行

![3](D:\Offer\操作系统\笔记\操作系统（李治军）图片\3.jpg)

0磁道0扇区是操作系统的引导扇区，首先写入内存。这是操作系统的第一段代码。

引导扇区代码：bootsect.s(汇编代码)

![4](D:\Offer\操作系统\笔记\操作系统（李治军）图片\4.jpg)

1.首先将引导扇区读入内存，代码执行后然后读入setup模块（四个扇区），然后是system模块

![5](D:\Offer\操作系统\笔记\操作系统（李治军）图片\5.jpg)

2.然后执行setup.s程序，将操作系统写入内存0x7c00,最后进入保护模式，寻址模式发生变化，从16位机切换到32位机地址模式，通过cr0来进行转变。

![6](D:\Offer\操作系统\笔记\操作系统（李治军）图片\6.jpg)

保护模式下地址翻译：cs:选择子 gdt表，setup需要先初始化gdt表

![7](D:\Offer\操作系统\笔记\操作系统（李治军）图片\7.jpg)

![8](D:\Offer\操作系统\笔记\操作系统（李治军）图片\8.jpg)

system模块的第一部分代码是head.s

![9](D:\Offer\操作系统\笔记\操作系统（李治军）图片\9.jpg)

![10](D:\Offer\操作系统\笔记\操作系统（李治军）图片\10.jpg)

操作系统做了两件事：读入内存和初始化。

### L4 操作系统接口

![12](D:\Offer\操作系统\笔记\操作系统（李治军）图片\12.jpg)

操作系统提供了一些重要的函数：这就是操作系统接口，接口表现为函数调用，又由系统提供，所以称为系统调用。

### L5 系统调用的实现

![13](D:\Offer\操作系统\笔记\操作系统（李治军）图片\13.jpg)

![14](D:\Offer\操作系统\笔记\操作系统（李治军）图片\14.jpg)

int 0x80 int指令查询idt表

![15](D:\Offer\操作系统\笔记\操作系统（李治军）图片\15.jpg)

![16](D:\Offer\操作系统\笔记\操作系统（李治军）图片\16.jpg)

![17](D:\Offer\操作系统\笔记\操作系统（李治军）图片\17.jpg)

### L6 操作系统历史

![18](D:\Offer\操作系统\笔记\操作系统（李治军）图片\18.jpg)

![19](D:\Offer\操作系统\笔记\操作系统（李治军）图片\19.jpg)

![20](D:\Offer\操作系统\笔记\操作系统（李治军）图片\20.jpg)

### L7 L8 CPU管理直观想法

一个CPU上交替的执行多个程序被称为并发。并发的目的是提高CPU利用率

### L9 多进程图像

1. PCB（process control block）是用来记录进程信息的数据结构

2. 进程状态：正在执行，就绪队列中等待执行，磁盘等待队列等。

3.

![21](D:\Offer\操作系统\笔记\操作系统（李治军）图片\21.jpg)

4. 进程交替的三个部分：队列操作+调度+切换

5. 调度的基本思路：

FIFO: 先进先出，但是要考虑优先级

![22](D:\Offer\操作系统\笔记\操作系统（李治军）图片\22.jpg)

6. 进程需要内存映射表来进行访问，这里涉及到虚拟内存的概念

7. 生产者-消费者实例，生产者进程和消费者进程需要给counter上锁，检查counter锁才能执行。生产者给counter上锁后消费者必须检查counter锁，然后生产者结束后给counter开锁，消费者才能执行。

![23](D:\Offer\操作系统\笔记\操作系统（李治军）图片\23.jpg)

### L10 L11用户级线程/内核级线程

1. 进程=资源+指令执行序列，将资源和指令执行分开，一个资源+多个指令执行序列

2. 线程保留了并发的优点，避免进程切换代价，资源不切换而只有指令切换是线程thread,线程的切换实际上就是映射表不变而PC指针变

3. yield函数能完成切换，create就是制造出第一次切换时应该的样子

4. 每个函数需要有自己的栈区储存函数返回的位置，这样才能完成不同函数的切换。

![24](D:\Offer\操作系统\笔记\操作系统（李治军）图片\24.jpg)

![26](D:\Offer\操作系统\笔记\操作系统（李治军）图片\26.jpg)

![25](D:\Offer\操作系统\笔记\操作系统（李治军）图片\25.jpg)

5. 多核和多处理器概念，多核支持并行

### L10 用户级线程

1.

![27](D:\Offer\操作系统\笔记\操作系统（李治军）图片\27.jpg)

2. ![28](D:\Offer\操作系统\笔记\操作系统（李治军）图片\28.jpg)

![29](D:\Offer\操作系统\笔记\操作系统（李治军）图片\29.jpg)

![30](D:\Offer\操作系统\笔记\操作系统（李治军）图片\30.jpg)

![31](D:\Offer\操作系统\笔记\操作系统（李治军）图片\31.jpg)

### L12 内核级线程实现

1. 理解起来较为困难，不知道需要掌握到何种程度。

### L14 CPU调度策略

1. CPU调度的直观想法是FIFO，，但是要考虑到优先级的影响，比如任务短的可以适当优先；

2. 如何设计调度算法：面对用户满意，面对进程满意。如何让进程满意：尽快结束任务：周转时间（从任务进入到任务结束）短，用户操作尽快响应：响应时间（从操作发生到响应）短；系统内耗时间少：吞吐量（完成的任务量）。总原则：系统专注于任务执行又能合理调配任务。

3. 如何做到合理，需要折中，需要综合。存在的矛盾如下：吞吐量和响应时间有矛盾（响应时间小->切换次数多->系统内耗大->吞吐量小）；前台任务和后台任务的关注点不同：前台任务关注响应时间，后台任务关注周转时间；IO约束型任务和CPU约束型任务有各自特点

4. 各种CPU调度算法：First Come,First Served(FCFS):周转时间过长；SJF:短作业优先缩短周转时间

5. 响应时间如何优化：RR（按时间片轮转调度），时间片大：响应时间太长；时间片小：吞吐量小。折衷思想：时间片10-100ms，切换时间0.1-1ms(1%)

6. 响应时间和周转时间同时存在，则前台任务利用RR，后台任务利用SJF，这两个任务队列优先级前台高，但是存在问题，所以需要后台任务优先级动态升高，同时

### L15 一个实际的schedule函数

1. ![32](D:\Offer\操作系统\笔记\操作系统（李治军）图片\32.jpg)

### L16 进程同步和信号量

1. ![33](D:\Offer\操作系统\笔记\操作系统（李治军）图片\33.jpg)

![34](D:\Offer\操作系统\笔记\操作系统（李治军）图片\34.jpg)

![35](D:\Offer\操作系统\笔记\操作系统（李治军）图片\35.jpg)

![36](D:\Offer\操作系统\笔记\操作系统（李治军）图片\36.jpg)

![37](D:\Offer\操作系统\笔记\操作系统（李治军）图片\37.jpg)

### L17 信号量临界区保护

1. ![38](D:\Offer\操作系统\笔记\操作系统（李治军）图片\38.png)

![39](D:\Offer\操作系统\笔记\操作系统（李治军）图片\39.jpg)

![40](D:\Offer\操作系统\笔记\操作系统（李治军）图片\40.jpg)

![41](D:\Offer\操作系统\笔记\操作系统（李治军）图片\41.jpg)

![42](D:\Offer\操作系统\笔记\操作系统（李治军）图片\42.jpg)

![43](D:\Offer\操作系统\笔记\操作系统（李治军）图片\43.jpg)

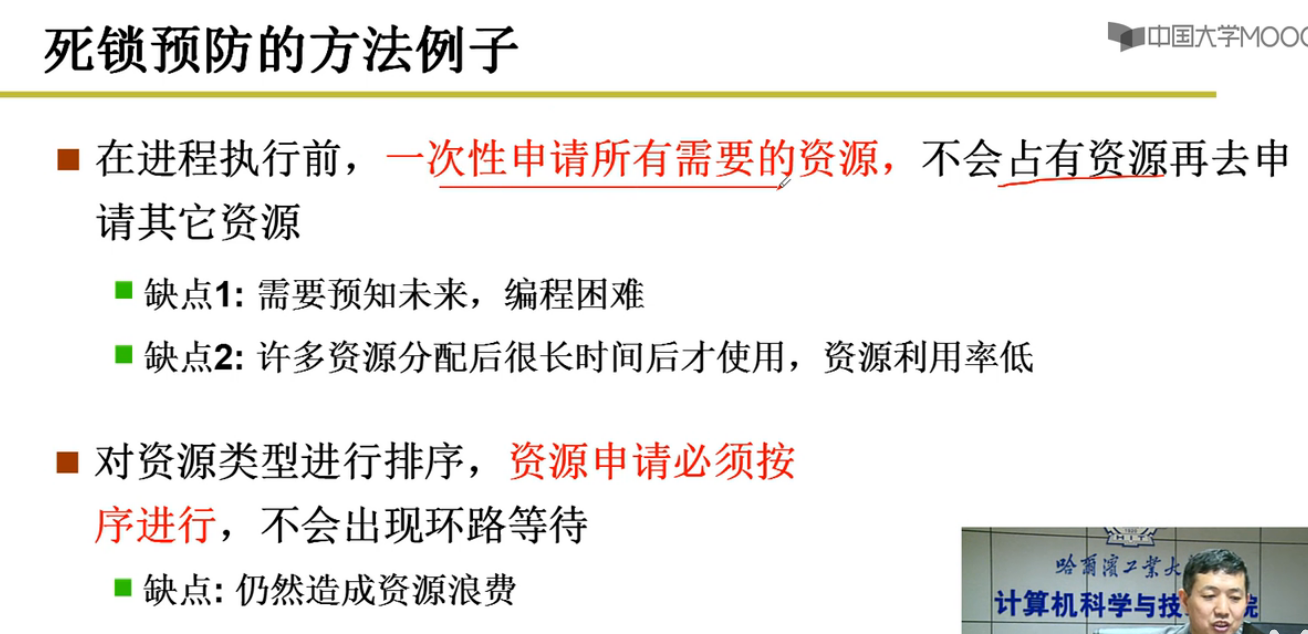
![44](D:\Offer\操作系统\笔记\操作系统（李治军）图片\44.jpg)

![45](D:\Offer\操作系统\笔记\操作系统（李治军）图片\45.jpg)

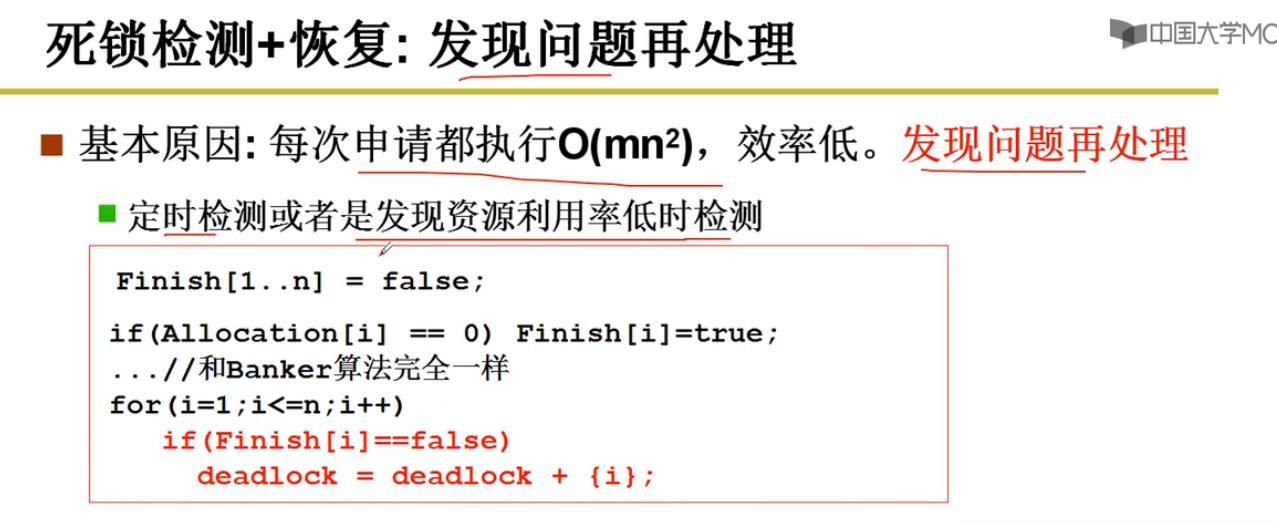
L18 信号量的代码实现

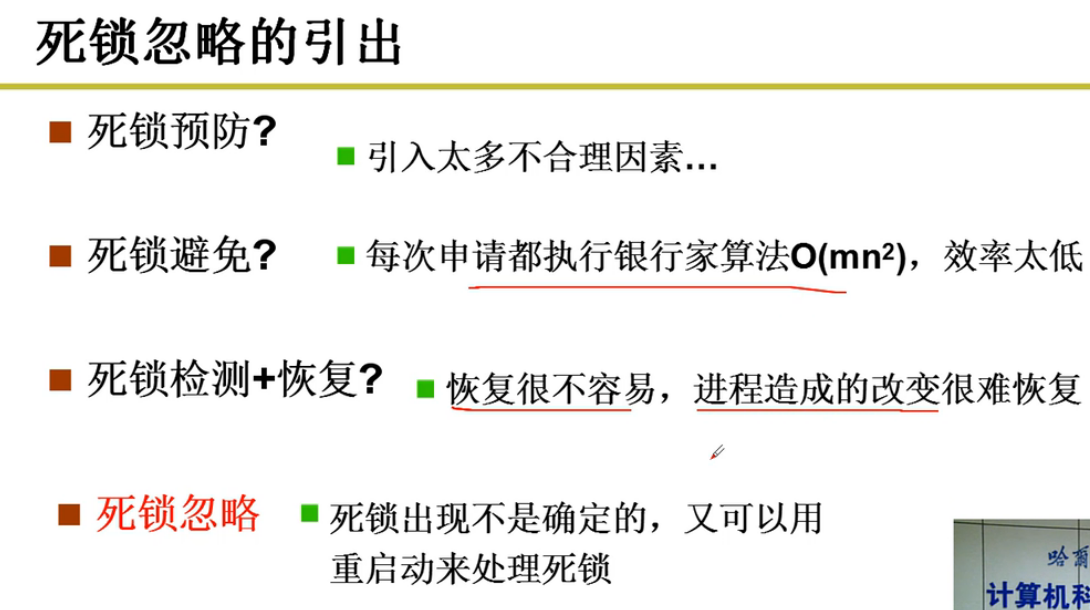
L19 死锁处理

1. 我们将这种多个进程由于相互等待对方持有的资源而造成的谁都无法执行的情况叫做死锁，四种方式处理死锁：死锁预防（破环死锁出现的条件）/死锁避免（检测每个资源请求，如果造成死锁就拒绝）/死锁检测+恢复（检测到死锁出现时就让一些进行回滚，让出资源）/死锁忽略（就好像没有出现死锁一样）

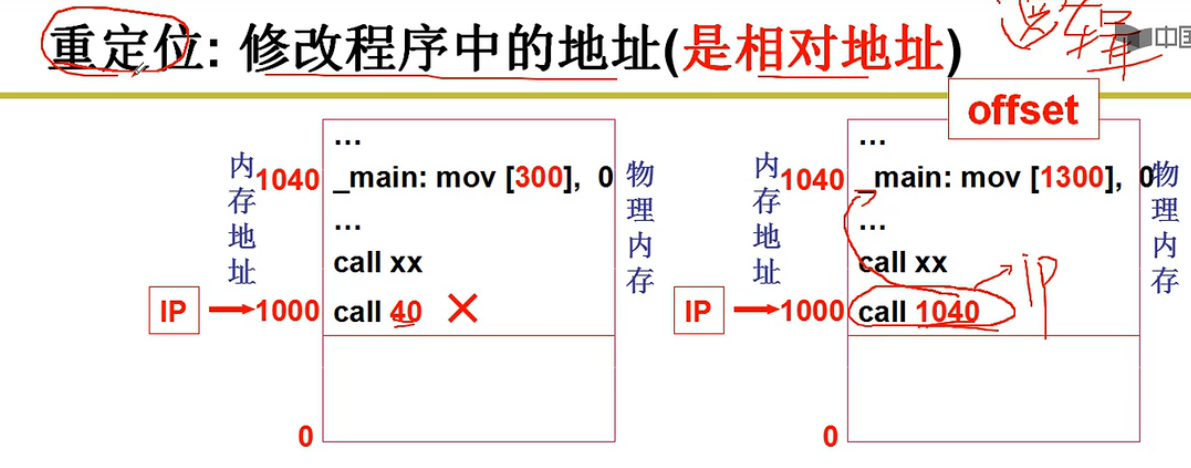


银行家算法：找安全序列，请求出现是首先假装分配，然后调用银行家算法。





L20 内存使用和分段



载入时进行重定位？但是由于程序载入后还需要移动所以最合适的时机是运行时重定位。

