2023年5月16日：

**知识点：**页式机制**+**简单段式存储管理过程+段页式+第二次作业布置

**思考题**：

1. 实现页式管理过程的数据结构是页表，位图等数据结构，那么，进程页表存放在哪里？有多少个这样的页表？什么样子的？有多少项？内容是什么？首地址存放在哪里？
2. 对于内存空闲空间的管理，往往采用位图这个数据结构，那么，位图存放在哪里？什么样子的？内容是什么？多大？系统有几个这个位图？这种管理方法存在的不足是什么？
3. 页式管理机制中，会存在碎片么？什么碎片？最大为多大？段式机制中，存在碎片么？什么碎片？
4. 逻辑地址和物理地址映射过程的图会示意画；
5. 引入两级页表甚至三级页表的目的是什么？优缺点？
6. 实现页式机制中，需要的硬件支持是哪些？快表是什么？为什么要引入快表？
7. 简单页式管理中的优缺点？进而在两级页表基础上，理解为什么要引入带有中断机制的内存管理机制（即虚拟存储器）？
8. 为什么要引入段式管理机制？其基本思路是什么？和页式管理思想的不同在哪里？
9. 段式存储管理实现过程？即数据结构段表，和具体管理过程。现在系统常倾向于段式还是页式？
10. 段式机制的优缺点？
11. X386CPU内存管理过程（见Linux内存管理案例PPT），理解段机制是这种CPU中必不可少的阶段，是CPU技术自身的延续。而运行在其上的操作系统，必然要为CPU进行段机制提供数据，如通过段描述符表，通过其中的段描述符这些数据结构，告知CPU这个段的基地址以及段偏移，然后CPU在获得这些值后，进行地址运算，这就是段机制的过程。而页机制，是在这个基础上进行，比较特别的是，CPU反而对页机制是可选择的，可以支持页机制，也可以不支持，而我们操作系统往往喜欢用页机制，这就要求CPU最好能支持页机制。
12. 画出段页式内存管理机制管理过程的示意图，并解释其进行过程。其中，PPT中示意图中举例中，那个最上面的逻辑地址，不要理解为48位，它就是一个段偏移之前的32地址，经过段偏移后，形成一个32位的线性地址。容易理解为48位，是因为，那幅图中是16为的选择器+32的段内偏移量，其实意思不一样的。
13. 段页式内存管理过程中，CPU在做什么呢？操作系统在做什么？

**拓展阅读**

1. **预习段页式机制，预习9.1/9/2，或**预习PPT的虚拟存储器内容。
2. 第二次项目提交时间：**5月31日**前