Due: 11.23 8:00

1. 微内核，84年，不太成功，早期版本的NT是微内核，linux（没有采用）也不是。抛弃windows。
2. 有各种说法，找国外。
3. 微内核
   1. Mach 3.0 79年-88年，90年代还在做，最终还是失败了，大部分软件的结局都是失败的。94年不再更新 ×
   2. plam-9 KThompson做的，有名，评价很高，不管用，40后，还在google，GO语言，做过Unix，plam-9该网站宣布不再更新；×
   3. Minux bios是minux操作系统上写的，license是BSD，DELL用的minux写的bios，bios是基本输入输出系统；✔
   4. QNX，汽车上用的系统，过不了10年？现在是统治地位。智能汽车用的什么？但作为微内核操作系统，它被认为是唯一一个成功的案例。Minux只是做了一个bios，也不是所有bios都是minux做的。给了微内核主义者极大支持✔
   5. Exokernel 外核，95年不再更新，MIT， 欣赏，idea很对；
   6. Ameaba，欧洲航天器啥的，百节虫；
   7. Chorus，法国，好多进程大家一起跑，一起演奏。设计一个一个进程一起跑，软件-考虑多少个进程，可能使用各种语言，用库函数。。。。。都没有时，才需要自己写。也像ameaba，已经面向多处理器的os，据说很好。Java最初做嵌入式，就用chorus微内核，跑java虚拟机，**sin？**公司，被oracle收购;
   8. L4, 德国，和外核属于一类，微内核失败以后，有些人不服，95年开始，第二代微内核开始，idea很对，欣赏；
4. 设计微内核，微内核还是宏内核，没标准答案，说出点什么，看看思路；
5. 核心态xxxxxx。奔腾里有四个状态，00，01，10，11，window&linux只用了两个状态，设备管理层次，minux结构，不同的层次放在不同的模式下，不同态权限不一样。
6. 系统功能调用和库函数的概念和区别：操作系统最重要的是安全，safety & security & protection，并不完全一样，计算机是一个执行程序的机器，保证系统的安全、控制指令的运行，公安局保证社会安全，控制指令的运行，安全指令分开（特权privilege指令-能抢银行、能干坏事），普通指令什么都能做，核心态（无法无天）&用户态（只能执行非特权指令），只有核心态可以执行特权指令，CPU内；os执行，CPU跃迁到核心态，app执行，CPU就是用户态，格式化硬盘是程序不是指令，危险工作放核里，要验证各类，验证才需要算法。先有指令才有算法。 怎么样由用户态进入核心态，库函数-两个状态是一样的，系统功能调用-两个状态不一样，进入核心态，返回时又回到用户态，是安全状态的基石；需要指令。Call 函数调用，RET 子程序返回， INT软中断，IRET中断返回。内核检查怎么检查，主动的是进程，被动的是文件，所有都是对文件的访问，有权限就能访问，没权限就不能访问，linux的**安全？**机制，进程-用户-访问。安全机制主要看人
7. 算法：多级反馈队列进程调度算法，动态调整时间片大小，minux不是动态返回队列，但也分层，有底层的高层就没办法运行。
8. 死锁，概念，必要条件，银行家算法适用条件，教科书，算法过程，书上转述。
9. PAE模式，存储扩充。虚空间很大，物理空间很小？真的吗？虚地址空间32位，物理地址空间52位，总线地址，进程地址32位。对奔腾地址方案有看法，while some said挺好，岔出一个地址……虚空间小于物理空间，每个进程有个虚空间，物理空间整个计算机只有一个，总线空间，创建的进程足够多，空间就会很大，因此创建不了这么多进程，物理空间用不完，内存条用得完，10G就能用光。
10. 为什么，英文题，翻译，阿姆斯特丹，多个读写请求里选择一个，选择一个处于运行态。只能说算法不好，不能说他不行，想应用场景，而不是第一部干啥….
11. 系统功能调用
    1. C语言main函数完整格式
    2. 入口参数和exec()的关系，专业词and专业句式
    3. 软件是由很多进程构成的，微内核思想也是这个思想，没必要像微内核这样切这么碎。有些功能切碎了反而会写的很乱，Myth man month。
12. 遇到的问题
13. 体会