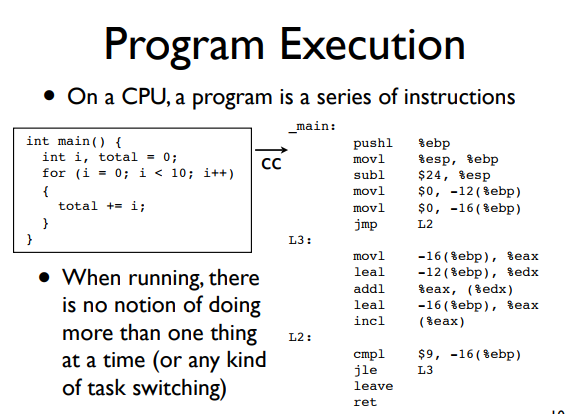
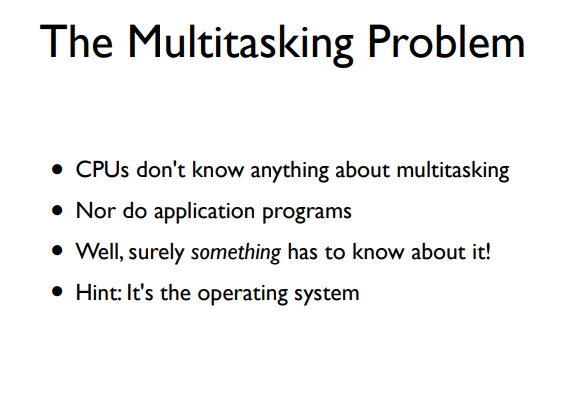
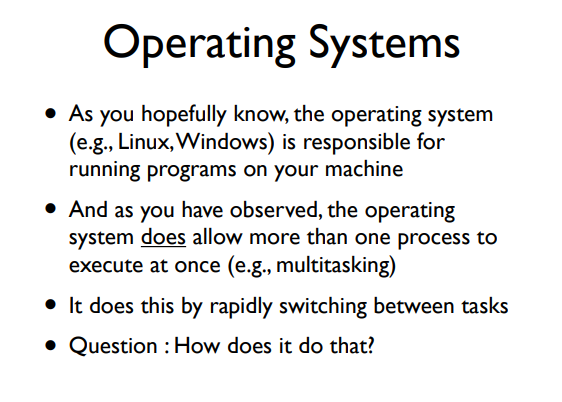
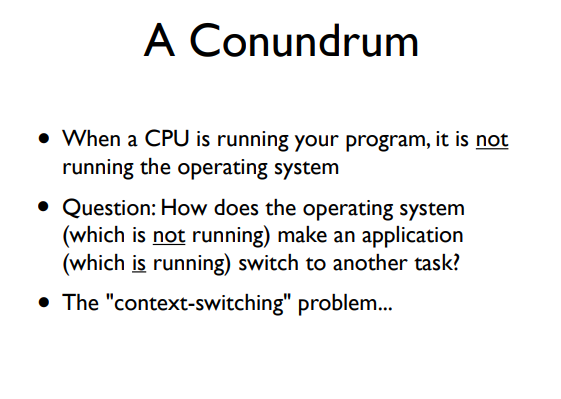
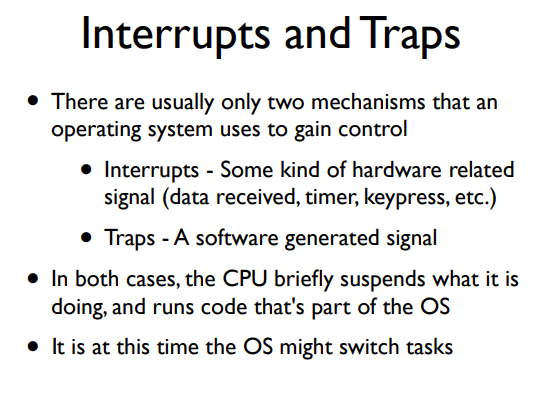
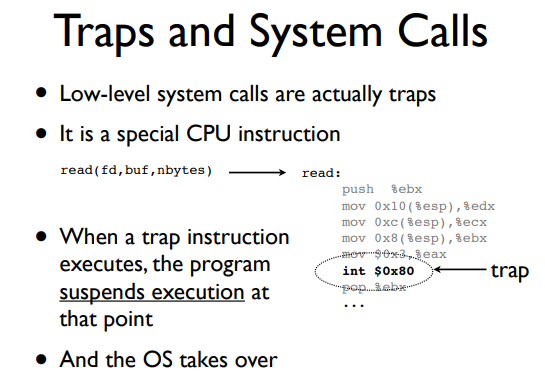
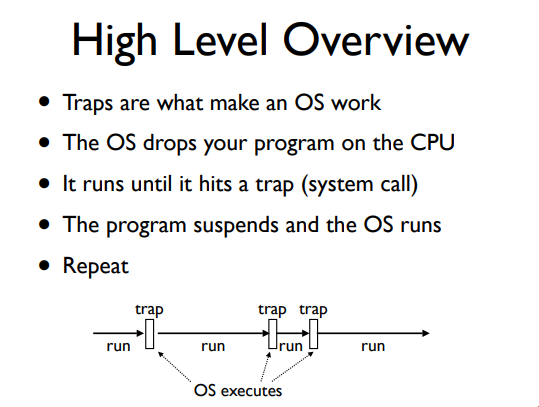
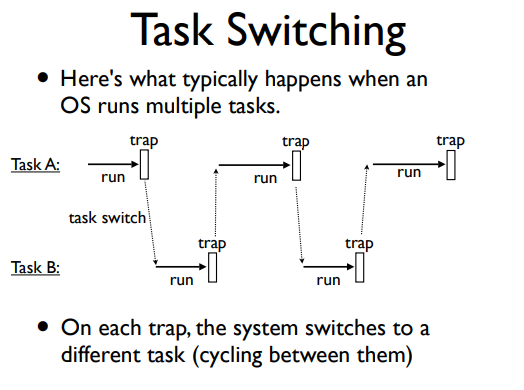
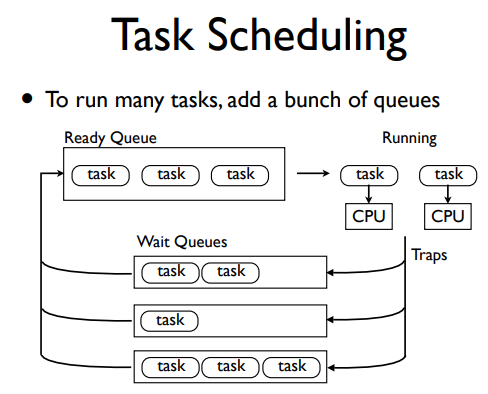
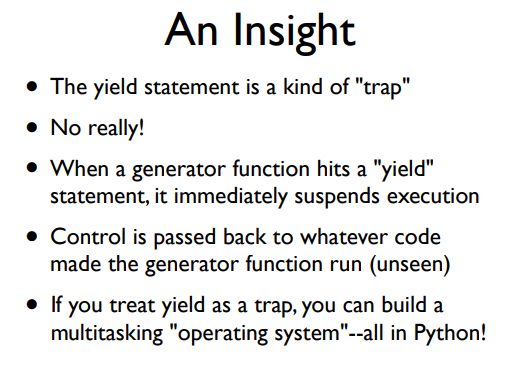
1. 对于CPU而言，并没有任务切换的概念，仅仅是执行执行  
     
   
2. 操作系统实现了多任务切换，保存上下文等功能  
     
   
3. 操作系统通过中断机制(Interrupts,硬件产生)和陷阱机制(traps,软件产生)这两种信号机制来控制CPU执行代码与切换任务  
   
4. 实际上大部分系统调用(SystemCall)都是通过traps机制实现用户态->内核态的切换，即用户程序暂停，执行系统函数的任务切换  
   
5. 操作系统程序与用户程序的任务切换主要是用过traps机制，首先操作系统控制我们的程序在CPU上运行，当用户程序发生系统调用时，此时CPU运行操作系统程序的指令，操作系统指令执行完毕后，继续执行用户程序指令，用户指令再发生系统调用时，又切换到操作系统程序指令执行，不断重复。  
   
6. 操作系统通过traps机制不断使得多个用户任务’一起’执行  
   
7. 操作系统当前不会仅仅像上图一般管理多任务，更多可能如下图一般实现一个任务编排程序来管理任务切换、暂停、运行..  
   
8. Python当中的traps机制（yield语句类似traps中断，当执行yield语句时，当前函数上下文环境保存，切换到另一个函数中执行..）  
   
9. 使用Python实现一个多任务的’OS’，我们需要实现如下几点  
   