

1. 多道程序设计是指在计算机系统中，允许多个程序同时进入内存并运行。即同时把多个程序放入内存中，并允许它们交替在 CPU 中运行，它们共享系统中的各种硬、软件资源。当一道程序因 I/O 请求而暂停运行时，CPU 便立即转去运行另一道。

分时操作系统将 CPU 处理时间分割为多个时间片，将时间片分给不同程序，达到多个程序“同时”运行的效果。时间片用完则产生时钟中断，控制权专职操作系统重新调度，未完成程序将挂起，等待再次分配。

两者的区别在于：

目标不同：批处理操作系统以提高系统资源率和作业吞吐量为目标，分时操作系统强调公平性；适应作业的性质不同：批处理操作系统适应已经调试好的大型作业，分时操作系统适应正在调试的小型作业；资源使用率不同：批处理操作系统合理安排不同负载作业，分时操作系统终端作业使用相同编译系统，调度开销小。

2. 器件快速更新换代、计算机体系结构不断发展、让用户使用计算机越来越方便的需要、满足用户新要求并提供给用户新服务的需求和图形化、网络化、开源化进程的发展。
3. 陷阱是指内核把控制权返回给程序后可继续执行而不失连续性，只有不必重新执行已终止指令时才触发陷阱，如调试程序，`eip` 保存随后要执行的指令地址。

区别在于：陷阱是同步的，中断是异步的；陷阱是程序内部有意设计的，中断是来自外部设备的信号。

操作系统的功能通过一组称为系统调用的接口呈现给用户，内核提供一系列实现预定功能的内核函数，系统调用把应用程序的请求传达给内核，内核调用对应的内核函数完成请求所需处理后，再将处理结果返回给应用程序。

4. (1) 每个系统架构都有自己独特的一套指令集。因此，不同架构的设备无法执行相同的程序，反之亦然。而且，不同架构的总线架构也不同，CPU 的字长度也不同。由于这些硬件的差异，构建完全可移植的操作系统并不可能。
(2) 一个高度可移植的操作系统包含两个层次：---机器相关层和机器无关层。机器相关层处理硬件的特性，并且需要为每个架构单独实现。这层提供一个统一的接口，供机器无关层使用。而设备无关层仅仅需要实现一次。要想实现高可移植性，机器相关层的规模应该设计的越小越好。
5. 一个设计实例是资源利用率和响应时间之间的关系。增加并行性会提高资源利用率，但可能会降低每个任务的响应时间。在一个操作系统中，如果为了提高资源利用率，设计选择了多道程序设计技术，允许多个程序同时在内存中运行，这样可以提高 CPU 和其他资源的利用率。然而，如果并发度过高，可能会导致每个程序的响应时间变长，因为 CPU 需要在多个程序之间切换，造成任务的等待时间增加。

6. (1)

打印机1						
打印机2		A	A		A	A
输入			B	B(30ms)		
处理器	A	B		A	B	B
	50ms	50ms	50ms	50ms	50ms	50ms

(2) 有空闲，在 B 程序第一次计算后的 50ms，因为程序 A 在输出，B 在等待输入。

(3) 程序 A 没有等待，程序 B 在输入结束时发生等待，因为此时程序 A 在占用处理器。