1. In a multiprogramming and time-sharing environment, several users share the system simultanesouly. The situation can result in various security problems. What are such problems? Can we ensure the same degree of security in a time-shared machine as in a dedicated machine? Explain your answer.

(1)多到程序设计可能延长某些程序的运行时间； 并行工作道数与系统效率不成正比 ，在分配资源时可能出现冲突情况，需合理进行资源分配；多道程序之间的切换也会耗费一定的时间，造成时间上的浪费。

(2) 分时系统需要设计好的调度算法，不能因为响应时间影响到用户体验；且分时系统的专用型以及安全性相比专用机器没有那么强,用户资料可能被窃取

(3) 不能保证，因为相比于专用机器，分时系统机器需要处理调度与响应等，过于高度安全的检测可能将延长响应时间，进而影响到用户体验。且大多数的专用机器对安全性的要求很高，极小的差错可能带来难以挽回的损失

2. What is the purpose of interrupts? What are the differences between a trap and an interrupt? Can traps be generaged intentionally by a user program? If so, for what purpose?

（1）事件发生以终端形式来通知，中断是可用于系统调用以及避免某些不安全程序如无限循环带来的危害，os还可通过timer来定时中断﻿﻿﻿﻿﻿﻿，防止用户程序无限占用资源

（2）陷入是由error或用户请求(后者通常称为system call)引起的软件中断，如除0；狭义中断的产生与硬件内部的变化有关﻿﻿﻿﻿﻿﻿﻿，硬件通过系统总线发送给CPU﻿﻿﻿﻿﻿﻿﻿以触发中断。﻿﻿﻿﻿﻿

（3）有意义的产生用来进行系统调用以及处理一些异常﻿﻿

3. Linux is a monolithic kernel system, while some embedding systems are micro-kernel systems. Please briefly describe their differences and pros/cons.

(1) ﻿微内核删除了不必要的组件，便于扩展操作系统，所有新服务可在用户空间内增加，便于移植，有更好的安全性和可靠性；但缺点是增加的系统功能

         ﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿的开销可能会使微内核的性能受损﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿，且﻿与进程间的通信过度联系和﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿对OS消息传递功能的频繁使用﻿﻿﻿﻿﻿﻿

(2) ﻿单一﻿地址空间的操作系统可提供非常高效的性能﻿﻿﻿﻿﻿﻿，避免了有关消息传递过多的处理。﻿﻿﻿﻿﻿﻿﻿但在系统功能的扩展方面可能带来不便，缺少微内核系统扩展的灵活性，故实际上现代的linux系统虽然是单一内核的，但同时也吸收了许多微内核的优点﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿。