**(1) What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other?**

**a. 用户级线程是无需内核感知的线程, 内核级线程需要操作系统内核感知.**

**b. 系统可以采用映射 M:1 or M:N,用户级线程与内核级线程就联系起来了.用户级线程的上下文切换与内核级线程的上下文切换类似，但是用户级线程的切换需要依赖于线程库与用户级线程与内核级线程之间的映射关系。**

**c. 内核级线程无需与进程相关联,而每个用户级线程需要与一个进程相关联**

**d. 内核级线程比用户级线程更难维护，因为它们必须用一个内核固定的数据结构表示**

**(2) What resources are used when a thread is created? How do they differ from those used when a process is created?**

* **创建一个线程需要分配一小段数据结构(TCB)来维护寄存器组，栈与优先级、调度等信息。**
* **进程创建的时候需要分配PCB，这是一个相当大的数据结构。PCB中包含了内存的映射，打开文件的列表与环境变量。分配和管理内存映射通常是最费时的。**
* **因此，由于线程比进程小很多，所以创建线程需要的资源会比创建进程的小很多。**

**(3) Which of the following components of program state are shared across threads in a multithreaded process?**

**a. Register values**

**b. Heap memory**

**c. Global variables**

**d. Stack memory**

**a) Heap memory and Global variables are shared**

**b) Register values and Stack memory can not be shared**

**(4) Compare the cost of thread creation and process creation (with evidence!)‏**

**创建线程的所需要的时间比创建进程少很多**

**进程：当进程开始时，主要开销在:** **allocate address space（分配地址单元空间）、establish numerous data tables（创建多个数据表）**

**线程：由于进程中的线程可以互相通用地址空间，因此他们开销只有: share most of the data （共享数据）。So it takes less space to start a thread than process（比进程需要的空间少）**

**(5) Describe the advantages and disadvantages of various threading model.**

**Many-to-One model：**

* **Advantages：线程管理在用户空间完成，所以它的效率比较高**
* **Disadvantages：但是如果一个线程调用了导致阻塞的系统调用的话，那么将阻塞整个进程。而且，因为一次只有一个线程可以访问内核，所以在多处理机环境中多个线程不能够并发执行。**

**One-to-One model**

* **Advantages：它允许在一个线程在调用某系统调用的情况下持续运行其它的线程，即使这个系统调用会导致阻塞。从而提供了比多对一模型更好的并发性；它也允许多个线程在多处理机环境中并行执行。**
* **Disadvantages：创建一个用户线程就需要创建一个相应的内核线程，加重应用程序的负担**

**Many-to-Many model**

* **Advantages：开发者能够创建所需的用户线程，而且相应的内核线程能够在多处理机环境中并行运行；当一个线程执行导致阻塞的系统调用时，内核能够调度其它的线程执行。**
* **Disadvantages：线程的管理机制变得复杂**