**Week 4 Questions:**

1. Is it possible to have concurrency but not parallelism? Explain.
2. Using Amdahl’s Law, calculate the speedup gain of an application that has a 60 percent parallel component for (a) two processing cores and (b four processing cores.
3. What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other?
4. Describe the actions taken by a kernel to context-switch between kernel-level threads.
5. Assume that an operating system maps user-level threads to the kernel using the many-to-many model and that the mapping is done through LWPs. Furthermore, the system allows developers to create real-time threads for use in real-time systems. Is it necessary to bind a real-time thread to an LWP? Explain.
6. Can a multithreaded solution using multiple user-level threads achieve better performance on a multiprocessor system than on a single processor system? Explain.
7. Consider a multicore system and a multithreaded program written using the many-to-many threading model. Let the number of user-level threads in the program be greater than the number of processing cores in the system. Discuss the performance implications of the following scenarios:
8. The number of kernel threads allocated to the program is less than the number of processing cores.
9. The number of kernel threads allocated to the program is equal to the number of processing cores.
10. The number of kernel threads allocated to the program is greater than the number of processing cores but less than the number of user-level threads.