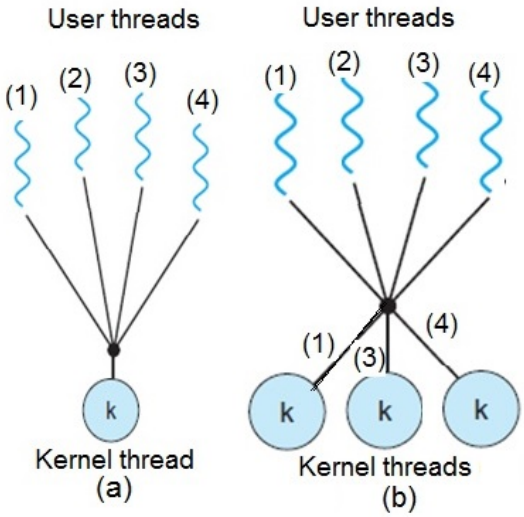
Week 4 Questions:

1. Is it possible to have concurrency but not parallelism? Explain.
2. Using Amdahl's Law, calculate an application's speedup gain with a 60 percent parallel component for (a) two processing cores and (b four processing cores.
3. What are the 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. Can a multithreaded solution using multiple user-level threads perform better on a multiprocessor system than on a single processor? Explain.
6. Answer the following based on the two **user-level to kernel-level threads mapping** scenarios shown in the **Figure below**:



6.1. Why is thread mapping important?

6.2. Identify the thread mapping scheme suitable for multiprogramming and multiprocessing.

6.3. Describe the impact on the process when the **user thread3** in (a) causes a **system call**.

6.4. Describe the impact on the process when the **user thread1** in (b) causes a system call.

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:

    a. The number of kernel threads allocated to the program is less than the number of processing cores.

    b. The number of kernel threads allocated to the program equals the number of processing cores.

    c. 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.