1. Thread is a fundamental unit of …….
2. What do/don’t threads share?
3. Which one is process/thread and what it means?
   1. Heavy-weight
   2. Light-weight
4. Name three benefits of using threads?
5. Name three challenges facing programmers when using multicore or multiprocessor systems?
6. Concurrency vs. parallelism
7. Data parallelism vs. task parallelism
8. If application is 60% parallel, you decided to go from 1 to 2 cores. What is the speedup you will get?
9. Multithreading models?
10. Which one:
    1. One thread blocking causes all to block
    2. Multiple threads run in parallel on multiprocessors
    3. More concurrency
    4. Overhead of creating many kernel threads
    5. 5 users threads mapped to 2 kernel threads.
    6. Unable to run in parallel on multicores
11. Thread library can be implemented entirely in user space (user thread library) or as kernel-level library:
    1. Invoking a function in the library results in a local function call and not a system call
    2. Invoking a function in the API for the library results in a system call
    3. Supported directly by the OS
12. Name the three main thread libraries?
13. Asynchronous vs. synchronous threading:
    1. Parent resumes its execution, no waiting for child.
    2. Parent waits for all children to finish
    3. Uses fork-join strategy
    4. Significant data sharing
    5. Little data sharing
14. Why exec() replaces the running process including all threads (why not only the thread that calls exec())?
15. What are signals in UNIX? What is signal handler? Two types of signal handlers?
16. Asynchronous vs. synchronous signals? examples?
17. Where should a signal be delivered for multi-threaded?
18. Thread to be canceled is called ….
19. Two types of approaches for thread cancellation?
20. What is TLS? Is it same as local variables?