1. **What are special machine instructions?** — Using special atomic machine instruction to implement a lock

2. **What are strong semaphores?** — Semaphores that use a queue

3. **What are the advantages and disadvantages of disable interrupts?** — They are easy to implement, but it restricts the efficiency of the processor, doesn't work on multiprocessor systems, and the user cannot be trusted

4. **What are the three characteristics of a monitor for mutual exclusion?** — Local data private to the monitor; a process enters it using its functions; only one process can be in the monitor at a time

5. **What are the three operations a semaphore can do?** — Initialize itself to a non-negative value; semWait (decrement the value, if the new value is negative then block the calling process); semSignal (increment the value, if the new value is <= to 0 then unblock a process blocked by semWait)

6. **What are the two problems that can be created with enforcement of mutual exclusion?** — Starvation and deadlock

7. **What are three advantages of the machine instruction approach?** — Works for any number of threads on any number of processors; simple; works for as many critical sections as needed

8. **What are three disadvantages of the machine instruction approach?** — Busy waiting is used, starvation can occur, and deadlock can occur

9. **What are two methods of mutual exclusion enforcement?** — Disable interrupt and special machine instructions

10. **What are two of the more common types of special machine instructions?** — Compare & swap and exchange instructions

11. **What are weak semaphores?** — Semaphores that use a list w/ processes removed in an order determined by the OS

12. **What does a compare and swap special machine instruction do?** — Compares the shared variable's value and a test value; if the values are the same, a swap occurs. The shared variable's value is always returned

13. **What does an exchange special machine instruction do?** — Exchanges the contents of a register with a memory location

14. **What does it mean when the semaphore's value is less than 0?** — The number of processes in the semaphore queue

15. **What do semWait and semSignal do in a binary semaphore?** — semWait (if it is 0, then put the calling process in the semaphore queue and block it; otherwise run the calling process) and semSignal (if the semaphore queue is empty set the value to 1, otherwise take the next process off the queue and run it)

16. **What is a binary semaphore?** — A semaphore that can only take on the values 0 and 1

17. **What is a critical section?** — Code that requests access to shared data - cannot be executed by more than one thread at a time

18. **What is a disable interrupt?** — If a process enters a critical section, then make the process un-interruptible until it leaves the critical section

19. **What is a monitor for mutual exclusion?** — Software module which consists of one or more functions, an initialization sequence, and local data

20. **What is a race condition?** — When multiple threads read & write a shared data item & the results depend on their relative timing

21. **What is deadlock?** — When two threads both need the same resources and won't release until they get the others' resources, so neither can continue

22. **What is mutual exclusion?** — When one process is in a critical section that accesses a set of resources, no other thread can be in a critical section that accesses any of these resources

23.	**What is starvation?**	When one process is blocked indefinitely because resources are being switched between other processes
24.	**What is the disadvantage of using a monitor?**	Program overhead
25.	**What is the downside of using semaphores for mutual exclusion?**	Inherent difficulty - semaphore operations can be scattered all over, making it hard to know the overall effect they have
26.	**What is the initial value of a semaphore?**	The number of processes allowed in the critical section at a time
27.	**What is the main idea of a semaphore, and what are semaphores?**	Two processes can cooperate by means of simple signals; a semaphore is a variable whose sole purpose is signalling b/w threads/processes