

1. Semaphore: A synchronization primitive used in computer science to control access to resources in a multi-threaded or multi-process environment.
2. Dispatch Latency: The time delay between a task or process being scheduled for execution by an operating system and its actual execution on a CPU core.
3. IPC can stand for "Inter-Process Communication," which refers to the methods and mechanisms used by various processes to communicate and exchange data in a computer system.
4. Shared Memory: A technique in computer science where multiple processes or threads can access and manipulate the same region of memory concurrently, enabling communication and data sharing between them.
5. Message Passing: A method of inter-process communication where processes or threads communicate by sending and receiving messages, allowing them to exchange data and synchronize their actions in a distributed or parallel computing environment.
6. Short-Term Scheduler (CPU Scheduler): Manages the execution of processes in the ready queue, selecting which process to run next on the CPU.
7. Medium-Term Scheduler: Handles processes that are swapped in and out of main memory, helping to balance system load and memory usage.
8. Long-Term Scheduler (Admission Scheduler): Selects processes from the pool of new processes and decides which ones to allocate resources to and bring into the system.
9. Time Quantum: Also known as a time slice or time slot, it's the fixed amount of time allocated to a process for execution in a round-robin scheduling algorithm. After a process uses up its time quantum, it's moved to the back of the queue to allow other processes a chance to run.
10. Turnaround Time: The total time taken to execute a process from the moment it enters the system until it completes and exits, including both execution time and waiting time.
11. Response Time: The time elapsed between submitting a request or initiating a process and receiving the first output or response from the system. It measures how quickly a system reacts to a stimulus.
12. FCFS stands for "First-Come, First-Served." It's a scheduling algorithm where processes are executed in the order they arrive, with the earliest arriving process being executed first.
13. Round Robin: A preemptive scheduling algorithm where each process is assigned a fixed time quantum to execute. Processes are executed in a cyclic manner, with each process getting a turn to execute for a specific time period, after which it's moved to the end of the queue.
14. Multilevel Feedback Queue Scheduling: A scheduling algorithm that uses multiple queues with different priorities to manage processes. Processes start in the highest-priority queue and can move to lower-priority queues based on their behavior. This allows for dynamic adjustment of priorities based on the process's CPU usage and behavior over time.

15. **IO Queue:** Also known as the I/O request queue, it's a queue that holds I/O requests from processes waiting for input/output operations to be completed by devices such as disks, printers, or network interfaces. The operating system schedules and manages these requests to ensure efficient device utilization.
16. **Mutual Exclusion:** A synchronization concept that ensures only one process or thread at a time can access a shared resource, preventing concurrent access and potential conflicts that might lead to incorrect results or data corruption.
17. **Thread:** A basic unit of execution within a process. Threads share the same memory space and resources of the process they belong to, allowing for parallel execution of tasks within the same program.
18. **Critical Section:** A section of a program or code where shared resources are accessed and manipulated. To avoid data corruption, race conditions, and conflicts, critical sections must be executed by only one thread or process at a time using synchronization mechanisms like locks or semaphores.
19. **Race Condition:** A situation in which the behavior of a program or system depends on the relative timing or sequence of events, particularly when multiple threads or processes access shared resources concurrently. Race conditions can lead to unexpected and incorrect results due to unpredictable execution order.
20. **Job Queue:** A queue that holds processes or jobs that are waiting to be executed by the operating system. These jobs might include various types of tasks, programs, or processes submitted by users or applications for execution on the system.
21. **PCB Queue:** A queue that holds Process Control Blocks (PCBs) in an operating system. A PCB contains information about a process, including its state, priority, program counter, register values, and more. The PCB queue helps the operating system manage and schedule processes efficiently.
22. **Device Queue:** A queue that holds requests from processes for access to specific I/O devices, such as disks or printers. These queues are managed by the operating system to ensure proper handling of device requests and efficient resource utilization.
23. **Bounded Buffer Problem:** A classic synchronization problem involving multiple producer and consumer threads sharing a fixed-size buffer. Producers place items into the buffer, while consumers remove items. The challenge is to ensure that producers don't overflow the buffer and consumers don't underflow it, requiring synchronization mechanisms like semaphores or mutexes to coordinate their actions properly.
24. **Aging:** In the context of scheduling or resource allocation, aging refers to the gradual increase of priority given to a process or task that has been waiting for a while. This prevents processes from waiting indefinitely and ensures that lower-priority processes eventually get a chance to execute, promoting fairness and preventing starvation.
25. **Program Counter:** Also known as the "instruction pointer" or "instruction address register," the program counter is a CPU register that indicates the memory address of the next instruction to be executed in a program. It keeps track of the current position in the program's execution sequence.

- 26. Communication link: The link between 2 process P and Q to send and receive messages
- 27. One-to-One: Each thread corresponds to one kernel-level thread.
- 28. Many-to-Many: Many user-level threads mapped to a smaller or equal number of kernel-level threads.
- 29. Many-to-One: Many user-level threads mapped to a single kernel-level thread.
- 30. Child Process: A process that is created by another process, known as the parent process. Child processes inherit certain attributes from their parent, such as environment variables and file descriptors. They can execute independently and perform tasks concurrently with other processes, while still being related to the parent process in terms of their creation and resources.