1. Which of the following is true of cooperative scheduling?
   A) It requires a timer.
   B) A process keeps the CPU until it releases the CPU either by terminating or by switching to the waiting state.
   C) It incurs a cost associated with access to shared data.
   D) A process switches from the running state to the ready state when an interrupt occurs.
   Ans: B

2. ____ is the number of processes that are completed per time unit.
   A) CPU utilization   B) Response time   C) Turnaround time   D) Throughput
   Throughput
   Ans: D

3. ____ scheduling is approximated by predicting the next CPU burst with an exponential average of the measured lengths of previous CPU bursts.
   A) Multilevel queue   B) RR   C) FCFS   D) SJF
   Ans: D

4. The ____ scheduling algorithm is designed especially for time-sharing systems.
   A) SJF   B) FCFS   C) RR   D) Multilevel queue
   Ans: C

5. Which of the following scheduling algorithms must be nonpreemptive?
   A) SJF   B) RR   C) FCFS   D) priority algorithms
   Ans: C

6. Which of the following is true of multilevel queue scheduling?
   A) Processes can move between queues.
   B) Each queue has its own scheduling algorithm.
   C) A queue cannot have absolute priority over lower-priority queues.
   D) It is the most general CPU-scheduling algorithm.
   Ans: B

7. The idea behind ____ is to create multiple logical processors on the same physical processor, presenting a view of several logical processors to the operating system.
   A) SMT   B) SMP   C) PCB   D) PCP
   Ans: A

8. The default scheduling class for a process in Solaris is ____.
   A) time sharing   B) system   C) interactive   D) real time
   Ans: A
9. In Linux, tasks that are not real-time have dynamic priorities that are based on their ____ values plus or minus the value 5.
   A) share    B) active    C) nice    D) runqueue
   Ans: C

10. In Little's formula, \( \eta = \lambda \times W \), \( \lambda \) represents the ____.
    A) average waiting time in the queue
    B) average arrival rate for new processes in the queue
    C) average queue length
    D) average CPU utilization
    Ans: B

11. Explain the concept of a CPU - I/O burst cycle.
    Ans: The lifecycle of a process can be considered to consist of a number of bursts belonging to two different states. All processes consist of CPU cycles and I/O operations. Therefore, a process can be modeled as switching between bursts of CPU execution and I/O wait.

12. What role does the dispatcher play in CPU scheduling?
    Ans: The dispatcher gives control of the CPU to the process selected by the short-term scheduler. To perform this task, a context switch, a switch to user mode, and a jump to the proper location in the user program are all required. The dispatch should be made as fast as possible. The time lost to the dispatcher is termed dispatch latency.

13. Explain the difference between a response time and a turnaround time. These times are both used to measure the effectiveness of scheduling schemes.
    Ans: Turnaround time is the sum of the periods that a process is spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O. Turnaround time essentially measures the amount of time it takes to execute a process. Response time, on the other hand, is a measure of the time that elapses between a request and the first response produced.

14. What effect does the size of the time quantum have on the performance of an RR algorithm?
    Ans: At one extreme, if the time quantum is extremely large, the RR policy is the same as the FCFS policy. If the time quantum is extremely small, the RR approach is called processor sharing and creates the appearance that each of \( n \) processes has its own processor running at \( 1/n \) the speed of the real processor.
15. Explain the process of starvation and how aging can be used to prevent it.
   Ans: Starvation occurs when a process is ready to run but is stuck waiting indefinitely for the CPU. This can be caused, for example, when higher priority processes prevent low-priority processes from ever getting the CPU. Aging involves gradually increasing the priority of a process so that a process will eventually achieve a high enough priority to execute if it does not execute for a long enough period of time.

16. Explain the fundamental difference between asymmetric and symmetric multiprocessing.
   Ans: In asymmetric multiprocessing, all scheduling decisions, I/O, and other system activities are handled by a single processor whereas in SMP, each processor is self-scheduling.

17. Describe two general approaches to load balancing.
   Ans: With push migration, a specific task periodically checks the load on each processor and - if it finds an imbalance - evenly distributes the load by moving processes from overloaded to idle or less-busy processors. Pull migration occurs when an idle processor pulls a waiting task from a busy processor. Pull and pull migration are often implemented in parallel on load-balancing systems.

18. In Windows XP, how does the dispatcher determine the order of thread execution?
   Ans: The dispatcher uses a 32-level priority scheme to determine the execution order. Priorities are divided into two classes. The variable class contains threads having priorities from 1 to 15, and the real-time class contains threads having priorities from 16 to 31. The dispatcher uses a queue for each scheduling priority and traverses the set of queues from highest to lowest until it finds a thread that is ready to run. The dispatcher executes an idle thread if no ready thread is found.

19. What is deterministic modeling and when is it useful in evaluating an algorithm?
   Ans: Deterministic modeling takes a particular predetermined workload and defines the performance of each algorithm for that workload. Deterministic modeling is simple, fast, and gives exact numbers for comparison of algorithms. However, it requires exact numbers for input, and its answers apply only in those cases. The main uses of deterministic modeling are in describing scheduling algorithms and providing examples to indicate trends.

20. Describe how trace tapes are used in distribution-driven simulations.
   Ans: In a distribution-driven simulation, the frequency distribution indicates only how many instances of each event occur; it does not indicate anything about the order of their occurrence. Trace tapes can correct this problem. A trace tape is created to monitor the real system and record the sequence of actual events. This sequence then drives the simulation. Trace tapes provide an excellent way to compare two algorithms on exactly the same set of real inputs.
21. In preemptive scheduling, the sections of code affected by interrupts must be guarded from simultaneous use.
   Ans: True

22. In RR scheduling, the time quantum should be small with respect to the context-switch time.
    Ans: False

23. The most complex scheduling algorithm is the multilevel feedback-queue algorithm.
    Ans: True

24. Load balancing is typically only necessary on systems with a common run queue.
    Ans: False

25. Systems using a one-to-one model (such as Windows XP, Solaris 9, and Linux) schedule threads using process-contention scope (PCS).
    Ans: False