Chapter 5

Multiple Choice

1. Which of the following is true of nonpreemptive (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.

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

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

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

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

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.

7. Which of the following statements are false with regards to the Linux CFS scheduler?
A) Each task is assigned a proportion of CPU processing time.
B) Lower numeric values indicate higher relative priorities.

C) There is a single, system-wide value of `vruntime`.
D) The scheduler doesn't directly assign priorities.

8. The Linux CFS scheduler identifies _____ as the interval of time during which every runnable task should run at least once.
A) virtual run time
B) targeted latency
C) `nice` value
D) load balancing

9. In Little's formula, λ, 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

10. _____ allows a thread to run on only one processor.
A) Processor affinity
B) Processor set
C) NUMA
D) Load balancing

11. What is the numeric priority of a Windows thread in the NORMAL_PRIORITY_CLASS with HIGHEST relative priority?
A) 24
B) 10
C) 8
D) 13

12. What is the numeric priority of a Windows thread in the HIGH_PRIORITY_CLASS with ABOVE_NORMAL relative priority?
A) 24
B) 10
C) 8
D) 14

13. What is the numeric priority of a Windows thread in the BELOW_NORMAL_PRIORITY_CLASS with NORMAL relative priority?
A) 6
B) 7
C) 5
D) 8

14. _____ involves the decision of which kernel thread to schedule onto which CPU.
A) Process-contention scope
B) System-contention scope

C) Dispatcher
D) Round-robin scheduling

15. A significant problem with priority scheduling algorithms is _____.
A) complexity
B) starvation
C) determining the length of the next CPU burst
D) determining the length of the time quantum

16. The _____ occurs in first-come-first-served scheduling when a process with a long CPU burst occupies the CPU.
A) dispatch latency
B) waiting time
C) convoy effect
D) system-contention scope

17. The rate of a periodic task in a hard real-time system is ____, where p is a period and t is the processing time.
A) $1/p$
B) p/t
C) $1/t$
D) pt

18. Which of the following is true of the rate-monotonic scheduling algorithm?
A) The task with the shortest period will have the lowest priority.
B) It uses a dynamic priority policy.
C) CPU utilization is bounded when using this algorithm.
D) It is non-preemptive.

19. Which of the following is true of earliest-deadline-first (EDF) scheduling algorithm?
A) When a process becomes runnable, it must announce its deadline requirements to the system.
B) Deadlines are assigned as following: the earlier the deadline, the lower the priority; the later the deadline, the higher the priority.
C) Priorities are fixed; that is, they cannot be adjusted when a new process starts running.
D) It assigns priorities statically according to deadline.

20. The two general approaches to load balancing are _____ and _____.
A) soft affinity, hard affinity
B) coarse grained, fine grained
C) soft real-time, hard real-time
D) push migration, pull migration

True/False

21. In preemptive scheduling, the sections of code affected by interrupts must be guarded from

simultaneous use.

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

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

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

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

26. Solaris and Windows assign higher-priority threads/tasks longer time quantums and lower-priority tasks shorter time quantums.

27. SMP systems that use multicore processors typically run faster than SMP systems that place each processor on separate cores.

28. Windows 7 User-mode scheduling (UMS) allows applications to create and manage thread independently of the kernel

29. Round-robin (RR) scheduling degenerates to first-come-first-served (FCFS) scheduling if the time quantum is too long.

30. Load balancing algorithms have no impact on the benefits of processor affinity.

31. A multicore system allows two (or more) threads that are in compute cycles to execute at the same time.

32. Providing a preemptive, priority-based scheduler guarantees hard real-time functionality.

33. In hard real-time systems, interrupt latency must be bounded.

34. In Pthread real-time scheduling, the SCHED_FIFO class provides time slicing among threads of equal priority.

35. In the Linux CFS scheduler, the task with smallest value of `vruntime` is considered to have the highest priority.

36. The length of a time quantum assigned by the Linux CFS scheduler is strictly dependent upon the relative priority of a task.

37. The Completely Fair Scheduler (CFS) is the default scheduler for Linux systems.