Chapter 5

Multiple Choice Questions

1. The ready queue can be implemented as a _____.
A) FIFO queue
B) priority queue
C) tree
D) unordered linked list
E) all of the above

Ans: E
Section: 5.1.2
Difficulty: Easy

2. Which of the following circumstances can cooperative scheduling take place?
A) when a process switches from the running state to the waiting state
B) when a process switches from the running state to the ready state
C) when a process switches from the waiting state to the ready state
D) none of the above

Ans: A
Section: 5.1.3
Difficulty: Easy

3. Which of the following circumstances can preemptive scheduling take place?
A) when a process switches from the running state to the waiting state
B) when a process switches from the waiting state to the ready state
C) when a process terminates
D) none of the above

Ans: B
Section: 5.1.3
Difficulty: Easy

4. Which of the following items does not belong to the function of a dispatcher?
A) switching context from one process to another
B) selecting a process among the available ones in the ready queue
C) switching to user mode
D) jumping to the proper location in the user program to resume that program

Ans: B
Section: 5.1.4
Difficulty: Medium

5. Assume process P_0 and P_1 are the process before and after a context switch, and PCB_0 and PCB_1 are respectively their process control block. Which of the following time units are included inside the dispatch latency?
A) P_0 executing
B) save state into PCB_0, and restore state from PCB_1
C) P_1 executing
D) all of the above

Ans: B
Section: 5.1.4
Difficulty: Medium

6. Which of the following criteria is more important for an interactive system?
A) CPU utilization
B) Response time
C) Turnaround time
D) Throughput

Ans: B
Section: 5.2
Difficulty: Easy

7. Which of the following criteria is more important from the point of view of a particular process?
A) CPU utilization
B) Response time
C) Turnaround time
D) Throughput

Ans: C
Section: 5.2
Difficulty: Easy

8. For interactive systems, it is more important to minimize _____.
A) the average response time
B) the average turnaround time
C) the variance in response time
D) the waiting time

Ans: C
Section: 5.2
Difficulty: Easy

9. Which of the following scheduling algorithm may suffer from convoy effect?
A) SJF
B) FCFS
C) RR
D) Multilevel queue

Ans: B
Section: 5.3.1
Difficulty: Easy

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

Ans: C
Feedback: 5.3.1
Difficulty: Medium

11. I/O-bound program typically has many short _____ and a CPU-bound program might have a few long _____.
A) I/O burst, CPU burst
B) I/O burst, I/O burst
C) CPU burst, CPU burst
D) CPU burst, I/O burst

Ans: C
Section: 5.3.1
Difficulty: Easy

12. 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

Ans: C
Feedback: 5.3.1
Difficulty: Medium

13. Which of the following scheduling algorithms gives the minimum average waiting time for a given set of processes?
A) SJF
B) FCFS
C) RR
D) Multilevel queue

Ans: A
Section: 5.3.2
Difficulty: Easy

14. Shortest-remaining-time-first scheduling is the preemptive version of _____?
A) SJF
B) FCFS
C) RR
D) Multilevel queue

Ans: A
Section: 5.3.2
Difficulty: Easy

15. Which of the following scheduling algorithms gives the minimum average response time?
A) SJF
B) FCFS
C) RR
D) Multilevel queue

Ans: C
Section: 5.3.3
Difficulty: Easy

16. 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

Ans: B
Feedback: 5.3.3
Difficulty: Medium

17. If the time quantum get too large, RR scheduling degenerates to _____?
A) SJF
B) FCFS
C) Shortest-remaining-time-first
D) Multilevel queue

Ans: B
Section: 5.3.3
Difficulty: Medium

18. Which of the following can be a solution to the problem of indefinite blockage of low-priority processes?
A) Aging
B) Starvation
C) Multilevel queue
D) All of the above

Ans: A
Section: 5.3.4
Difficulty: Easy

19. Which of the following processes usually have highest priority?
A) real-time processes
B) system processes
C) interactive processes
D) batch processes

Ans: A
Section: 5.3.5
Difficulty: Easy

20. Which of the following is allowed on Linux and Mac OS X systems?
A) only PTHREAD_SCOPE_PROCESS
B) only PTHREAD_SCOPE_SYSTEM
C) Both PTHREAD_SCOPE_PROCESS and PTHREAD_SCOPE_SYSTEM
D) none of the above

Ans: B
Section: 5.4.2
Difficulty: Easy

21. Which of the following system architectures involves multiprocessor scheduling?
A) multicore CPUs
B) multithreaded cores
C) NUMA systems
D) heterogeneous multiprocessing
E) all of the above

Ans: E
Section: 5.5
Difficulty: Easy

22. Which of the following is preemptive?
A) rate-monotonic scheduling
B) earliest-deadline-first scheduling
C) both of the above
D) none of the above

Ans: C
Section: 5.6.3 and 5.6.4
Difficulty: Medium

23. Which of the following POSIX API sets the scheduling policy?
A) pthread_attr_getsched_policy
B) pthread_attr_setsched_policy
C) pthread_attr_getscope
D) pthread_attr_setscope

Ans: B
Section: 5.6.6
Difficulty: Easy

24. 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

Ans: B
Feedback: 5.7.1
Difficulty: Medium

25. 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

Ans: B
Feedback: 5.7.2
Difficulty: Easy

26. 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

Ans: A
Feedback: 5.7.2
Difficulty: Easy

27. In Solaris, if an interactive thread with priority 25 is waiting for I/O, what is its priority recalculated to when it is eligible to run again?
A) 15
B) 120
C) 52
D) It remains at 25

Ans: C
Feedback: 5.7.3
Difficulty: Easy

28. In Solaris, what is the time quantum (in milliseconds) of an interactive thread with priority 35?
A) 25
B) 54
C) 80
D) 35

Ans: C
Section: 5.7.3
Difficulty: Easy

29. The default scheduling class for a process in Solaris is ____.
A) time sharing
B) system
C) interactive
D) real-time

Ans: A
Feedback: 5.7.3
Difficulty: Easy

Essay Questions

1. 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.
Feedback: 5.1.1
Difficulty: Medium

2. 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.
Feedback: 5.1.4
Difficulty: Medium

3. Explain the difference between response time and 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.
Feedback: 5.2
Difficulty: Medium

4. 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 waited for

a long enough period of time.
Feedback: 5.3.3
Difficulty: Difficult

5. 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.
Feedback: 5.3.4
Difficulty: Medium

6. 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.
Feedback: 5.5.1
Difficulty: Medium

7. 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. Push and pull migration are often implemented in parallel on load-balancing systems.
Feedback: 5.5.3
Difficulty: Medium

8. Distinguish between coarse-grained and fine-grained multithreading.

Ans: There are two approaches to multithread a processor. (1) Coarse-grained multithreading allows a thread to run on a processor until a long-latency event, such as waiting for memory, to occur. When a long-latency event does occur, the processor switches to another thread. (2) Fine-grained multithreading switches between threads at a much finer-granularity, such as between instructions.
Feedback: 5.5.4
Difficulty: Medium

9. What are the two types of latency that affect the performance of real-time systems?
Ans: Interrupt latency refers to the period of time from the arrival of an interrupt at the CPU to the start of the routine that services the interrupt. Dispatch latency refers to the amount of time required for the scheduling dispatcher to stop one process and start another.
Section: 5.6.1
Difficulty: Medium

10. What are the advantages of the EDF scheduling algorithm over the rate-monotonic scheduling algorithm?
Ans: Unlike the rate-monotonic algorithm, EDF scheduling does not require that processes be periodic, nor must a process require a constant amount of CPU time per burst. The appeal of EDF scheduling is that it is theoretically optimal - theoretically, it can schedule processes so that each process can meet its deadline requirements and CPU utilization will be 100 percent.
Section: 5.6.4
Difficulty: Medium

11. In Windows, 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.
Feedback: 5.7.2
Difficulty: Difficult

12. 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 describing scheduling algorithms and providing examples to indicate trends.
Feedback: 5.8.1
Difficulty: Medium

True/False Questions

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

Ans: False
Feedback: 5.3.4
Difficulty: Medium

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

Ans: True
Feedback: 5.3.4
Difficulty: Easy

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

Ans: True
Feedback: 5.3.6
Difficulty: Medium

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

Ans: False
Feedback: 5.4.1
Difficulty: Easy

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

Ans: False
Feedback: 5.5.3
Difficulty: Medium

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

Ans: False
Feedback: 5.5.3
Difficulty: Medium

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

Ans: True
Feedback: 5.5.4
Difficulty: Easy

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

Ans: True
Feedback: 5.5.4
Difficulty: Easy

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

Ans: False
Section: 5.6
Difficulty: Difficult

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

Ans: True
Section: 5.6.1
Difficulty: Medium

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

Ans: False
Section: 5.6.6
Difficulty: Medium

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

Ans: False
Feedback: 5.7
Difficulty: Medium

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

Ans: True
Section: 5.7.1
Difficulty: Medium

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

Ans: False
Section: 5.7.1
Difficulty: Medium

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

Ans: True
Section: 5.7.1
Difficulty: Medium

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

Ans: True
Feedback: 5.7.2
Difficulty: Medium

17. A Solaris interactive thread with priority 15 has a higher relative priority than an interactive thread with priority 20

Ans: False
Feedback: 5.7.3
Difficulty: Easy

18. A Solaris interactive thread with a time quantum of 80 has a higher priority than an interactive thread with a time quantum of 120.

Ans: True
Feedback: 5.7.3
Difficulty: Easy