Import Settings:
Base Settings: Brownstone Default
Highest Answer Letter: D
Multiple Keywords in Same Paragraph: No

Chapter: Chapter 4

Multiple Choice

1. ____ is a thread library for Solaris that maps many user-level threads to one kernel thread.
A) Pthreads
B) Green threads
C) Sthreads
D) Java threads

Ans: B
Feedback: 4.3.1
Difficulty: Medium

2. Pthreads refers to ____.
A) the POSIX standard.
B) an implementation for thread behavior.
C) a specification for thread behavior.
D) an API for process creation and synchronization.

Ans: C
Feedback: 4.4.1
Difficulty: Medium

3. The ____ multithreading model multiplexes many user-level threads to a smaller or equal number of kernel threads.
A) many-to-one model
B) one-to-one model
C) many-to-many model
D) many-to-some model

Ans: C
Feedback: 4.3.3
Difficulty: Easy

4. Cancellation points are associated with ____ cancellation.
A) asynchronous
B) deferred
C) synchronous
D) non-deferred

Ans: B
Feedback: 4.6.3
Difficulty: Medium

5. Which of the following would be an acceptable signal handling scheme for a multithreaded program?
A) Deliver the signal to the thread to which the signal applies.
B) Deliver the signal to every thread in the process.
C) Deliver the signal to only certain threads in the process.
D) All of the above

Ans: D
Feedback: 4.6.2
Difficulty: Medium

6. Signals can be emulated in windows through ____.
A) asynchronous procedure calls
B) local procedure calls
C) remote procedure calls
D) none of the above

Ans: A
Feedback: 4.6.2
Difficulty: Medium

7. Thread-local storage is data that ____.
A) is not associated with any process
B) has been modified by the thread, but not yet updated to the parent process

C) is generated by the thread independent of the thread's process
D) is unique to each thread

Ans: D
Feedback: 4.6.4
Difficulty: Medium

8. LWP is ____.
A) short for lightweight processor
B) placed between system and kernel threads
C) placed between user and kernel threads
D) common in systems implementing one-to-one multithreading models

Ans: C
Feedback: 4.6.5
Difficulty: Easy

9. Windows uses the ____.
A) one-to-one model
B) many-to-one model
C) one-to many-model
D) many-to-many model

Ans: A
Feedback: 4.7.1
Difficulty: Easy

10. In multithreaded programs, the kernel informs an application about certain events using a procedure known as a(n) ____.
A) signal
B) upcall
C) event handler
D) pool

Ans: B
Feedback: 4.6.5
Difficulty: Medium

11. _____ is not considered a challenge when designing applications for multicore systems.
A) Deciding which activities can be run in parallel
B) Ensuring there is a sufficient number of cores
C) Determining if data can be separated so that it is accessed on separate cores
D) Identifying data dependencies between tasks.

Ans: B
Feedback: 4.2.1
Difficulty: Medium

12. A ____ provides an API for creating and managing threads.
A) set of system calls
B) multicore system
C) thread library
D) multithreading model

Ans: C
Feedback: 4.4
Difficulty: Easy

13. The _____ model multiplexes many user-level threads to a smaller or equal number of kernel threads.
A) many-to-many
B) two-level
C) one-to-one
D) many-to-one

Ans: A
Feedback: 4.3.3
Difficulty: Easy

14. The _____ model maps many user-level threads to one kernel thread.
A) many-to-many
B) two-level
C) one-to-one
D) many-to-one

Ans: D
Feedback: 4.3.1
Difficulty: Easy

15. The _____ model maps each user-level thread to one kernel thread.
A) many-to-many
B) two-level
C) one-to-one
D) many-to-one

Ans: C
Feedback: 4.3.2
Difficulty: Easy

16. The _____ model allows a user-level thread to be bound to one kernel thread.
A) many-to-many
B) two-level
C) one-to-one
D) many-to-one

Ans: B
Feedback: 4.3.3
Difficulty: Easy

17. The most common technique for writing multithreaded Java programs is _____.
A) extending the `Thread` class and overriding the `run()` method
B) implementing the `Runnable` interface and defining its `run()` method
C) designing your own `Thread` class
D) using the `CreateThread()` function

Ans: B
Feedback: 4.4.3
Difficulty: Easy

18. In Pthreads, a parent uses the `pthread_join()` function to wait for its child thread to complete. What is the equivalent function in Win32?
A) `win32_join()`
B) `wait()`
C) `WaitForSingleObject()`
D) `join()`

Ans: C
Section 4.4.2
Difficulty: Medium

19. Which of the following statements regarding threads is false?
A) Sharing is automatically provided in Java threads.
B) Both Pthreads and Win32 threads share global data.
C) The `start()` method actually creates a thread in the Java virtual machine.
D) The Java method `join()` provides similar functionality as the `WaitForSingleObject` in Win32.

Ans: A
Feedback: 4.4.3
Difficulty: Medium

20. A _____ uses an existing thread — rather than creating a new one — to complete a task.
A) lightweight process
B) thread pool
C) scheduler activation
D) asynchronous procedure call

Ans: B
Feedback: 4.5.1
Difficulty: Easy

21. According to Amdahl's Law, what is the speedup gain for an application that is 60% parallel and we run it on a machine with 4 processing cores?
A) 1.82
B) .7
C) .55
D) 1.43

Ans: D
Feedback: 4.2
Difficulty: Medium

22. _____ involves distributing tasks across multiple computing cores.
A) Concurrency

B) Task parallelism
C) Data parallelism
D) Parallelism

Ans: B
Feedback: 4.2.2
Difficulty: Medium

23. _____ is a formula that identifies potential performance gains from adding additional computing cores to an application that has a parallel and serial component.
A) Task parallelism
B) Data parallelism
C) Data splitting
D) Amdahl's Law

Ans: D
Feedback: 4.2
Difficulty: Medium

24. When OpenMP encounters the `#pragma omp parallel` directive, it
A) constructs a parallel region
B) creates a new thread
C) creates as many threads as there are processing cores
D) parallelizes `for` loops

Ans: C
Feedback: 4.5.2
Difficulty: Medium

25. Grand Central Dispatch handles blocks by
A) placing them on a dispatch queue
B) creating a new thread
C) placing them on a dispatch stack
D) constructing a parallel region

Ans: A
Feedback: 4.5.3
Difficulty: Medium

Essay

26. Why should a web server not run as a single-threaded process?

Ans: For a web server that runs as a single-threaded process, only one client can be serviced at a time. This could result in potentially enormous wait times for a busy server.
Feedback: 4.1.1
Difficulty: Medium

27. List the four major categories of the benefits of multithreaded programming. Briefly explain each.

Ans: The benefits of multithreaded programming fall into the categories: responsiveness, resource sharing, economy, and utilization of multiprocessor architectures. Responsiveness means that a multithreaded program can allow a program to run even if part of it is blocked. Resource sharing occurs when an application has several different threads of activity within the same address space. Threads share the resources of the process to which they belong. As a result, it is more economical to create new threads than new processes. Finally, a single-threaded process can only execute on one processor regardless of the number of processors actually present. Multiple threads can run on multiple processors, thereby increasing efficiency.
Feedback: 4.1.2
Difficulty: Difficult

28. What are the two different ways in which a thread library could be implemented?

Ans: The first technique of implementing the library involves ensuring that all code and data structures for the library reside in user space with no kernel support. The other approach is to implement a kernel-level library supported directly by the operating system so that the code and data structures exist in kernel space.
Feedback: 4.4
Difficulty: Medium

29. Describe two techniques for creating `Thread` objects in Java.

Ans: One approach is to create a new class that is derived from the `Thread` class and to override its `run()` method. An alternative — and more commonly used — technique is to define a class that implements the `Runnable` interface. When a class implements `Runnable`, it must define a

`run()` method. The code implementing the `run()` method is what runs as a separate thread.
Feedback: 4.4.3
Difficulty: Medium

30. In Java, what two things does calling the `start()` method for a new `Thread` object accomplish?

Ans: Calling the `start()` method for a new `Thread` object first allocates memory and initializes a new thread in the JVM. Next, it calls the `run()` method, making the thread eligible to be run by the JVM. Note that the `run()` method is never called directly. Rather, the `start()` method is called, which then calls the `run()` method.
Feedback: 4.4.3
Difficulty: Medium

31. Some UNIX systems have two versions of `fork()`. Describe the function of each version, as well as how to decide which version to use.

Ans: One version of `fork()` duplicates all threads and the other duplicates only the thread that invoked the `fork()` system call. Which of the two versions of `fork()` to use depends on the application. If `exec()` is called immediately after forking, then duplicating all threads is unnecessary, as the program specified in the parameters to `exec()` will replace the process. If, however, the separate process does not call `exec()` after forking, the separate process should duplicate all threads.
Feedback: 4.6.1
Difficulty: Difficult

32. How can deferred cancellation ensure that thread termination occurs in an orderly manner as compared to asynchronous cancellation?

Ans: In asynchronous cancellation, the thread is immediately cancelled in response to a cancellation request. There is no insurance that it did not quit in the middle of a data update or other potentially dangerous situation. In deferred cancellation, the thread polls whether or not it should terminate. This way, the thread can be made to cancel at a convenient time.
Feedback: 4.6.3
Difficulty: Medium

33. What is a thread pool and why is it used?

Ans: A thread pool is a collection of threads, created at process startup, that sit and wait for work to be allocated to them. This allows one to place a bound on the number of concurrent threads associated with a process and reduce the overhead of creating new threads and destroying them at termination.
Feedback: 4.5.1
Difficulty: Medium

34. What are the general components of a thread in Windows?

Ans: The thread consists of a unique ID, a register set that represents the status of the processor, a user stack for user mode, a kernel stack for kernel mode, and a private storage area used by run-time libraries and dynamic link libraries.
Feedback: 4.4.2
Difficulty: Medium

35. Describe the difference between the `fork()` and `clone()` Linux system calls.

Ans: The `fork()` system call is used to duplicate a process. The `clone()` system call behaves similarly except that, instead of creating a copy of the process, it creates a separate process that shares the address space of the calling process.
Feedback: 4.7.2
Difficulty: Medium

36. Multicore systems present certain challenges for multithreaded programming. Briefly describe these challenges.

Ans: Multicore systems have placed more pressure on system programmers as well as application developers to make efficient use of the multiple computing cores. These challenges include determining how to divide applications into separate tasks that can run in parallel on the different cores. These tasks must be balanced such that each task is doing an equal amount of work. Just as tasks must be separated, data must also be divided so that it can be accessed by the tasks running on separate cores. So that data can safely be accessed, data dependencies must be identified and where such dependencies exist, data accesses must be synchronized to ensure the safety of the data. Once all such challenges have been met, there remains considerable challenges testing and debugging such applications.
Feedback: 4.2.1
Difficulty: Difficult

37. Distinguish between parallelism and concurrency.

Ans: A parallel system can perform more than one task simultaneously. A concurrent system supports more than one task by allowing multiple tasks to make progress.
Feedback: 4.2
Difficulty: Medium

38. Distinguish between data and task parallelism.

And: Data parallelism involves distributing subsets of the same data across multiple computing cores and performing the same operation on each core. Task parallelism involves distributing tasks across the different computing cores where each task is performing a unique operation.
Feedback: 4.2.2
Difficulty: Difficult

39. Describe how OpenMP is a form of implicit threading.

Ans: OpenMP provides a set of compiler directives that allows parallel programming on systems that support shared memory. Programmers identify regions of code that can run in parallel by placing them in a block of code that begins with the directive `#pragma omp parallel`. When the compiler encounters this parallel directive, it creates as many threads as there are processing cores in the system.
Feedback: 4.5.2
Difficulty: Difficult

40. Describe how Grand Central Dispatch is a form of implicit threading.

Ans: Grand Central Dispatch (GCD) is a technology for Mac OS X and iOS systems that is a combination of extensions to the C language, an API, and a runtime library that allows developers to construct "blocks" - regions of code that can run in parallel. GCD then manages the parallel execution of blocks in several dispatch queues.
Feedback: 4.5.3
Difficulty: Difficult

True/False

41. A traditional (or heavyweight) process has a single thread of control.

Ans: True
Feedback: 4.1
Difficulty: Easy

42. A thread is composed of a thread ID, program counter, register set, and heap.

Ans: False
Feedback: 4.1
Difficulty: Medium

43. Virtually all contemporary operating systems support kernel threads.

Ans: True
Feedback: 4.1.1
Difficulty: Easy

44. Linux distinguishes between processes and threads.

Ans: False
Feedback: 4.7.2
Difficulty: Easy

45. In Java, data shared between threads is simply declared globally.

Ans: False
Feedback: 4.4.3
Difficulty: Medium

46. Each thread has its own register set and stack.

Ans: True
Feedback: 4.1
Difficulty: Easy

47. Deferred cancellation is preferred over asynchronous cancellation.

Ans: True
Feedback: 4.6.3
Difficulty: Easy

48. The single benefit of a thread pool is to control the number of threads.

Ans: False
Feedback: 4.5.1
Difficulty: Easy

49. It is possible to create a thread library without any kernel-level support.

Ans: True
Feedback: 4.4
Difficulty: Medium

50. It is possible to have concurrency without parallelism.

And: True
Feedback: 4.2
Difficulty: Medium

51. Amdahl's Law describes performance gains for applications with both a serial and parallel component.

Ans: True
Feedback: 4.2
Difficulty: Medium

52. OpenMP only works for C, C++, and Fortran programs.

Ans: True
Feedback 4.5.2:

Difficulty: Medium

53. Grand Central Dispatch requires multiple threads.

Ans: False
Feedback: 4.5.3
Difficulty: Medium

54. The trend in developing parallel applications is to use implicit threading.

Ans: True
Feedback: 4.5
Difficulty: Medium

55. Task parallelism distributes threads and data across multiple computing cores.

Ans: False
Feedback: 4.2.2
Difficulty: Difficult