# Operating System #IT241

## Chapter 1

### Multiple Choice

1. In what way is an operating system like a government?
A) It seldom functions correctly.
B) It creates an environment within which other programs can do useful work.
C) It performs most useful functions by itself.
D) It is always concerned primarily with the individual's needs.
Ans: B

2. ____ operating systems are designed primarily to maximize resource utilization.
A) PC B) Handheld computer
C) Mainframe D) Network
Ans: C

3. The most common secondary storage device is ____.
A) random access memory B) solid state disks
C) tape drives D) magnetic disk
Ans: D

4. Which of the following would lead you to believe that a given system is an SMP-type system?
A) Each processor is assigned a specific task.
B) There is a boss–worker relationship between the processors.
C) Each processor performs all tasks within the operating system.
D) None of the above
Ans: C

5. A ____ can be used to prevent a user program from never returning control to the operating system.
A) portal
B) program counter
C) firewall
D) timer
Ans: D

6. Embedded computers typically run on a ____ operating system.
A) real-time
B) Windows XP
C) network
D) clustered
Ans: A

7. Bluetooth and 802.11 devices use wireless technology to communicate over several feet, in essence creating a ____.
A) local-area network
B) wide-area network
C) small-area network
D) metropolitan-area network
Ans: C

8. A clustered system ____.
A) gathers together multiple CPUs to accomplish computational work
B) is an operating system that provides file sharing across a network
C) is used when rigid time requirements are present
D) can only operate one application at a time
Ans: A

9. Which of the following is a property of peer-to-peer systems?
A) Clients and servers are not distinguished from one another.
B) Separate machines act as either the client of the server but not both.
C) They do not offer any advantages over traditional client-server systems.
D) They suffer from the server acting as the bottleneck in performance.
Ans: A

10. Two important design issues for cache memory are ____.
A) speed and volatility
B) size and replacement policy
C) power consumption and reusability
D) size and access privileges
Ans: B

11. What are some  other terms for kernel mode?
A) supervisor mode
B) system mode
C) privileged mode
D) All of the above
Ans: D

12. Which of the following statements concerning open source operating systems is true?
A) Solaris is open source.
B) Source code is freely available.
C) They are always more secure than commercial, closed systems.
D) All open source operating systems share the same set of goals.
Ans: B

13. Which of the following operating systems is not open source?
A) Windows
B) BSD UNIX
C) Linux
D) PCLinuxOS
Ans: A

14. A _____ provides a file-system interface which allows clients to create and modify files.
A) compute-server system
B) file-server system
C) wireless network
D) network computer
Ans: B

15. A ____ is a custom build of the Linux operating system
A) LiveCD
B) installation
C) distribution
D) VMWare Player
Ans: C

16. _____ is a set of software frameworks that provide additional services to application developers.
A) System programs
B) Virtualization
C) Cloud computing
D) Middleware
Ans: D

17. What statement concerning privileged instructions is considered false?
A) They may cause harm to the system.
B) They can only be executed in kernel mode.
C) They cannot be attempted from user mode.
D) They are used to manage interrupts.
Ans: C

18. Which of the following statements is false?
A) Mobile devices must be concerned with power consumption.
B) Mobile devices can provide features that are unavailable on desktop or laptop computers.
C) The difference in storage capacity between a mobile device and laptop is shrinking.
D) Mobile devices usually have fewer processing cores than a standard desktop computer.
Ans: C

19. A(n) _____ is the unit of work in a system.
A) process
B) operating system
C) timer
D) mode bit
Ans: A

20. The two separate modes of operating in a system are
A) supervisor mode and system mode
B) kernel mode and privileged mode
C) physical mode and logical mode
D) user mode and kernel mode
Ans: D

## Essay

21. Explain why an operating system can be viewed as a resource allocator.
Ans: A computer system has many resources that may be required to solve a problem: CPU time, memory space, file-storage space, I/O devices, and so on. The operating system acts as the manager of these resources. Facing numerous and possibly conflicting requests for resources, the operating system must decide how to allocate them to specific programs and users so that it can operate the computer system efficiently and fairly.

22. Explain the purpose of an interrupt vector.
Ans: The interrupt vector is merely a table of pointers to specific interrupt-handling routines. Because there are a fixed number of interrupts, this table allows for more efficient handling of the interrupts than with a general-purpose, interrupt-processing routine.

23. What is a bootstrap program, and where is it stored?
Ans: A bootstrap program is the initial program that the computer runs when it is powered up or rebooted. It initializes all aspects of the system, from CPU registers to device controllers to memory contents. Typically, it is stored in read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM), known by the general term firmware, within the computer hardware.

24. What role do device controllers and device drivers play in a computer system?
Ans: A general-purpose computer system consists of CPUs and multiple device controllers that are connected through a common bus. Each device controller is in charge of a specific type of device. The device controller is responsible for moving the data between the peripheral devices that it controls and its local buffer storage. Typically, operating systems have a device driver for each device controller. This device driver understands the device controller and presents a uniform interface for the device to the rest of the operating system.

25. Why are clustered systems considered to provide high-availability service?
Ans: Clustered systems are considered high-availability in that these types of systems have redundancies capable of taking over a specific process or task in the case of a failure. The redundancies are inherent due to the fact that clustered systems are composed of two or more individual systems coupled together.

26. Describe the differences between physical, virtual, and logical memory.
Ans: Physical memory is the memory available for machines to execute operations (i.e., cache, random access memory, etc.). Virtual memory is a method through which programs can be executed that requires space larger than that available in physical memory by using disk memory as a backing store for main memory. Logical memory is an abstraction of the computer's different types of memory that allows programmers and applications a simplified view of memory and frees them from concern over memory-storage limitations.

27. Describe the operating system's two modes of operation.
Ans: In order to ensure the proper execution of the operating system, most computer systems provide hardware support to distinguish between user mode and kernel mode. A mode bit is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1). When the computer system is executing on behalf of a user application, the system is in user mode. However, when a user application requests a service from the operating system (via a system call), it must transition from user to kernel mode to fulfill the request.

28. Explain cache coherency.
Ans: In multiprocessor environments, two copies of the same data may reside in the local cache of each CPU. Whenever one CPU alters the data, the cache of the other CPU must receive an updated version of this data. Cache coherency involves ensuring that multiple caches store the most updated version of the stored data.

29. Why is main memory not suitable for permanent program storage or backup purposes? Furthermore, what is the main disadvantage to storing information on a magnetic disk drive as opposed to main memory?
Ans: Main memory is a volatile memory in that any power loss to the system will result in erasure of the data stored within that memory. While disk drives can store more information permanently than main memory, disk drives are significantly slower.

30. Describe the compute-server and file-server types of server systems.
Ans: The compute-server system provides an interface to which a client can send a request to perform an action (for example, read data from a database); in response, the server executes the action and sends back results to the client. The file-server system provides a file-system interface where clients can create, update, read, and delete files. An example of such a system is a Web server that delivers files to clients running Web browsers.

31. Computer systems can be divided into four approximate components. What are they?
Ans: Hardware, operating system, application programs, and users.

32. Distinguish between system and application programs.
Ans: System programs are not part of the kernel, but still are associated with the operating system. Application programs are not associated with the operating of the system.

33. Describe why direct memory access (DMA) is considered an efficient mechanism for performing I/O.
Ans: DMA is efficient for moving large amounts of data between I/O devices and main memory. It is considered efficient because it removes the CPU from being responsible for transferring data. DMA instructs the device controller to move data between the devices and main memory.

34. Describe why multi-core processing is more efficient than placing each processor on its own chip.
Ans: A large reason why it is more efficient is that communication between processors on the same chip is faster than processors on separate chips.

35. Distinguish between uniform memory access (UMA) and non-uniform memory access (NUMA) systems.
Ans: On UMA systems, accessing RAM takes the same amount of time from any CPU. On NUMA systems, accessing some parts of memory may take longer than accessing other parts of memory, thus creating a performance penalty for certain memory accesses.

36. Explain the difference between singly, doubly, and circularly linked lists.
Ans: A singly linked list is where each item points to its successor. A doubly linked
linked list allows an item to point to its predecessor or successor. A circularly linked
list is the where the last element points back to the first.

37. What two operating systems currently dominate mobile computing?
Ans: Apple's iOS and Google's Android

38. Explain the difference between protection and security.
Ans: Protection is concerned with controlling the access of processes or users to the resources of the computer system. The role of security is to defend the system from internal or external attacks.

39. Distinguish mobile computing from traditional desktop computing.
Ans: Mobile computing takes place on handheld devices and tablets. Because these devices are portable and lightweight, they typically do not have the processing power and storage capacity of desktop systems. However, features such as GPS and accelerometers have allowed mobile devices to provide functionality that is unavailable to desktop systems.

40. Describe cloud computing.
Ans: Cloud computing is a type of computing that delivers computing,storage, and application services across a network. Cloud computing often uses virtualization to provide its functionality. There are many different types of cloud environments, as well as services offered. Cloud computing may be either public, private, or a hybrid of the two. Additionally, cloud computing may offer applications, platforms, or system infrastructures.

## True/False

41. The operating system kernel consists of all system and application programs in a computer. **False**

42. Flash memory is slower than DRAM but needs no power to retain its contents. **True**

43. A system call is triggered by hardware. **False**

44. UNIX does not allow users to escalate privileges to gain extra permissions for a restricted activity. **False**

45. Processors for most mobile devices run at a slower speed than a processor in a desktop PC. **True**

46. Interrupts may be triggered by either hardware of software. **True**

47. A dual-core system requires each core has its own cache memory. **False**

48. Virtually all modern operating systems provide support for SMP. **True**

49. All computer systems have some sort of user interaction. **False**

50. Solid state disks are generally faster than magnetic disks. **True**

51. Solid state disks are considered volatile storage. **False**

52. There is no universally accepted definition of an operating system. **True**

# Chapter 2

## Multiple Choice

1. A _____ is an example of a systems program.
A) command interpreter
B) Web browser
C) text formatter
D) database system
Ans: A

2. If a program terminates abnormally, a dump of memory may be examined by a ____ to determine the cause of the problem.
A) module
B) debugger
C) shell
D) control card
Ans: B

3. A message-passing model is ____.
A) easier to implement than a shared memory model for intercomputer communication
B) faster than the shared memory model
C) a network protocol, and does not apply to operating systems
D) only useful for small simple operating systems
Ans: A

4. Policy ____.
A) determines how to do something
B) determines what will be done
C) is not likely to change across places
D) is not likely to change over time
Ans: B

5. The major difficulty in designing a layered operating system approach is ____.
A) appropriately defining the various layers
B) making sure that each layer hides certain data structures, hardware, and operations from higher-level layers
C) debugging a particular layer
D) making sure each layer is easily converted to modules
Ans: A

6. A microkernel is a kernel ____.
A) containing many components that are optimized to reduce resident memory size
B) that is compressed before loading in order to reduce its resident memory size
C) that is compiled to produce the smallest size possible when stored to disk
D) that is stripped of all nonessential components
Ans: D

7. To the SYSGEN program of an operating system, the least useful piece of information is _____.
A) the CPU being used
B) amount of memory available
C) what applications to install
D) operating-system options such as buffer sizes or CPU scheduling algorithms
Ans: C

8. A boot block ____.
A) typically only knows the location and length of the rest of the bootstrap program
B) typically is sophisticated enough to load the operating system and begin its execution
C) is composed of multiple disk blocks
D) is composed of multiple disk cylinders
Ans: A

9. _____ provide(s) an interface to the services provided by an operating system.
A) Shared memory
B) System calls
C) Simulators
D) Communication
Ans: B

10. _____ is not one of the major categories of system calls.
A) Process control
B) Communications
C) Protection
D) Security
Ans: D

11. _____ allow operating system services to be loaded dynamically.
A) Virtual machines
B) Modules
C) File systems
D) Graphical user interfaces
Ans: B

12. Microkernels use _____ for communication.
A) message passing
B) shared memory
C) system calls
D) virtualization
Ans: A

13. The Windows `CreateProcess()` system call creates a new process. What is the equivalent system call in UNIX:
A) `NTCreateProcess()`
B) `process()`
C) `fork()`
D) `getpid()`
Ans: C

14. The `close()` system call in UNIX is used to close a file. What is the equivalent system call in Windows:
A) `CloseHandle()`
B) `close()`
C) `CloseFile()`
D) `Exit()`
Ans: A

15. The Windows `CreateFile()` system call is used to create a file. What is the equivalent system call in UNIX:
A) `ioctl()`
B) `open()`
C) `fork()`
D) `createfile()`
Ans: B

16. Android runs Java programs _____
A) in the Dalvik virtual machine.
B) natively.
C) in the Java virtual machine.
D) Android does not run Java programs.
Ans: A

17. _____ is a mobile operating system designed for the iPhone and iPad.
A) Mac OS X B) Android
C) UNIX D) iOS
Ans: D

18. The _____ provides a portion of the system call interface for UNIX and Linux.
A) POSIX
B) Java
C) Standard C library
D) Standard API
Ans: C

19. Which of the following statements is incorrect?
A) An operating system provides an environment for the execution of programs.
B) An operating system manages system resources.
C) Operating systems provide both command line as well as graphical user interfaces.
D) Operating systems must provide both protection and security.
Ans: C

20. _____ is/are not a technique for passing parameters from an application to a system call.
A) Cache memory
B) Registers
C) Stack
D) Special block in memory
Ans: A

## Essay

21. There are two different ways that commands can be processed by a command interpreter. One way is to allow the command interpreter to contain the code needed to execute the command. The other way is to implement the commands through system programs. Compare and contrast the two approaches.
Ans: In the first approach, upon the user issuing a command, the interpreter jumps to the appropriate section of code, executes the command, and returns control back to the user. In the second approach, the interpreter loads the appropriate program into memory along with the appropriate arguments. The advantage of the first method is speed and overall simplicity. The disadvantage to this

technique is that new commands require rewriting the interpreter program which, after a number of modifications, may get complicated, messy, or too large. The advantage to the second method is that new commands can be added without altering the command interpreter. The disadvantage is reduced speed and the clumsiness of passing parameters from the interpreter to the system program.

22. Describe the relationship between an API, the system-call interface, and the operating system.
Ans: The system-call interface of a programming language serves as a link to system calls made available by the operating system. This interface intercepts function calls in the API and invokes the necessary system call within the operating system. Thus, most of the details of the operating-system interface are hidden from the programmer by the API and are managed by the run-time support library.
Feedback: 2.3
Difficulty: Hard

23. Describe three general methods used to pass parameters to the operating system during system calls.
Ans: The simplest approach is to pass the parameters in registers. In some cases, there may be more parameters than registers. In these cases, the parameters are generally stored in a block, or table, of memory, and the address of the block is passed as a parameter in a register. Parameters can also be placed, or pushed, onto the stack by the program and popped off the stack by the operating system.

24. What are the advantages of using a higher-level language to implement an operating system?
Ans: The code can be written faster, is more compact, and is easier to understand and debug. In addition, improvements in compiler technology will improve the generated code for the entire operating system by simple recompilation. Finally, an operating system is far easier to port — to move to some other hardware — if it is written in a higher-level language.

25. Describe some requirements, or goals, when designing an operating system.
Ans: Requirements can be divided into user and system goals. Users desire a system that is convenient to use, easy to learn, and to use, reliable, safe, and fast. System goals are defined by those people who must design, create, maintain, and operate the system: The system should be easy to design, implement, and maintain; it should be flexible, reliable, error-free, and efficient.

26. What are the advantages and disadvantages of using a microkernel approach?
Ans: One benefit of the microkernel approach is ease of extending the operating system. All new services are added to user space and consequently do not require modification of the kernel. The microkernel also provides more security and reliability, since most services are running as user — rather than kernel — processes. Unfortunately, microkernels can suffer from performance decreases due to increased system function overhead.

27. Explain why a modular kernel may be the best of the current operating system design techniques.

Ans: The modular approach combines the benefits of both the layered and microkernel design techniques. In a modular design, the kernel needs only to have the capability to perform the required functions and know how to communicate between modules. However, if more functionality is required in the kernel, then the user can dynamically load modules into the kernel. The kernel can have sections with well-defined, protected interfaces, a desirable property found in layered systems. More flexibility can be achieved by allowing the modules to communicate with one another.

28. Describe how Mac OS X is considered a hybrid system.
Ans: Primarily because he kernel environment is a blend of the Mach microkernel and BSD UNIX (which is closer to a monolithic kernel.)

29. Describe how Android uses a unique virtual machine for running Java programs.
Ans: The Dalvik virtual machine is designed specifically for Android and has been optimized for mobile devices with limited memory and CPU processing capabilities.

## True/False

30. KDE and GNOME desktops are available under open-source licenses. **True**

31. Many operating system merge I/O devices and files into a combined file because of the similarity of system calls for each. **True**

32. An initial bootstrap program is in the form of random-access memory (RAM). **False**

33. System calls can be run in either user mode or kernel mode. **False**

34. Application programmers typically use an API rather than directory invoking system calls. **True**

35. In general, Windows system calls have longer, more descriptive names and UNIX system calls use shorter, less descriptive names. **True**

36. Mac OS X is a hybrid system consisting of both the Mach microkernel and BSD UNIX. **True**

37. iOS is open source, Android is closed source. **False**

# Chapter 3

## Multiple Choice

1. The ____ of a process contains temporary data such as function parameters, return addresses, and local variables.
A) text section B) data section
C) program counter D) stack
Ans: D

2. A process control block ____.
A) includes information on the process's state
B) stores the address of the next instruction to be processed by a different process
C) determines which process is to be executed next
D) is an example of a process queue
Ans: A

3. The list of processes waiting for a particular I/O device is called a(n) ____.
A) standby queue
B) device queue
C) ready queue
D) interrupt queue
Ans: B

4. The _____ refers to the number of processes in memory.
A) process count
B) long-term scheduler
C) degree of multiprogramming
D) CPU scheduler
Ans: C

5. When a child process is created, which of the following is a possibility in terms of the execution or address space of the child process?
A) The child process runs concurrently with the parent.
B) The child process has a new program loaded into it.
C) The child is a duplicate of the parent.
D) All of the above
Ans: D

6. A _____ saves the state of the currently running process and restores the state of the next process to run.
A) save-and-restore
B) state switch
C) context switch
D) none of the above
Ans: C

7. A process may transition to the Ready state by which of the following actions?
A) Completion of an I/O event
B) Awaiting its turn on the CPU
C) Newly-admitted process
D) All of the above
Ans: D

8. In a(n) ____ temporary queue, the sender must always block until the recipient receives the message.
A) zero capacity B) variable capacity
C) bounded capacity D) unbounded capacity
Ans: A

9. A blocking `send()` and blocking `receive()` is known as a(n) _____
A) synchronized message
B) rendezvous
C) blocked message
D) asynchronous message
Ans: B

10 . Which of the following is true in a Mach operating system?
A) All messages have the same priority.
B) Multiple messages from the same sender are guaranteed an absolute ordering.
C) The sending thread must return immediately if a mailbox is full.
D) It is not designed for distributed systems.
Ans: A

11. When communicating with sockets, a client process initiates a request for a connection and is assigned a port by the host computer. Which of the following would be a valid port assignment for the host computer?
A) 21
B) 23
C) 80
D) 1625
Ans: D

12. A(n) _____ allows several unrelated processes to use the pipe for communication.
A) named pipe
B) anonymous pipe
C) LIFO
D) ordinary pipe
Ans: B

13. Which of the following statements is true?
A) Shared memory is typically faster than message passing.
B) Message passing is typically faster than shared memory.
C) Message passing is most useful for exchanging large amounts of data.
D) Shared memory is far more common in operating systems than message passing.
Ans:A

14. Imagine that a host with IP address 150.55.66.77 wishes to download a file from the web server at IP address 202.28.15.123. Select a valid socket pair for a connection between this pair of hosts.
A) 150.55.66.77:80 and 202.28.15.123:80
B) 150.55.66.77:150 and 202.28.15.123:80
C) 150.55.66.77:2000 and 202.28.15.123:80
D) 150.55.66.77:80 and 202.28.15.123:3500
Ans:C

15. Child processes inherit UNIX ordinary pipes from their parent process because:
A) The pipe is part of the code and children inherit code from their parents.
B) A pipe is treated as a file descriptor and child processes inherit open file descriptors from their parents.
C) The STARTUPINFO structure establishes this sharing.
D) All IPC facilities are shared between the parent and child processes.
Ans:B

16. Which of the following statements is true?
A) Named pipes do not allow bi-directional communication.
B) Only the parent and child processes can use named pipes for communication.
C) Reading and writing to ordinary pipes on both UNIX and Windows systems can be performed like ordinary file I/O.
D) Named pipes can only be used by communicating processes on the same machine.
Ans: C

17. Which of the following is not a process type in the Chrome browser?
A) Plug-in
B) Renderer
C) Sandbox
D) Browser
Ans: C

18. The _____ application is the application appearing on the display screen of a mobile device.
A) main
B) background
C) display
D) foreground
Ans: D

19. A process that has terminated, but whose parent has not yet called wait(), is known as a _____ process.
A) zombie B) orphan
C) terminated D) init
Ans: A

20. The _____ process is assigned as the parent to orphan processes.
A) zombie
B) init
C) main
D) renderer
Ans: B

## Short Answer

21. Name and describe the different states that a process can exist in at any given time.
Ans: The possible states of a process are: new, running, waiting, ready, and terminated. The process is created while in the new state. In the running or waiting state, the process is executing or waiting for an event to occur, respectively. The ready state occurs when the process is ready and waiting to be assigned to a processor and should not be confused with the waiting state mentioned earlier. After the process is finished executing its code, it enters the termination state.

22. Explain the main differences between a short-term and long-term scheduler.
Ans: The primary distinction between the two schedulers lies in the frequency of execution. The short-term scheduler is designed to frequently select a new process for the CPU, at least once every 100 milliseconds. Because of the short time between executions, the short-term scheduler must be fast. The long-term scheduler executes much less frequently; minutes may separate the creation of one new process and the next. The long-term scheduler controls the degree of multiprogramming. Because of the longer interval between executions, the long-term scheduler can afford to take more time to decide which process should be selected for execution.

23. Explain the difference between an I/O-bound process and a CPU-bound process.
Ans: The differences between the two types of processes stem from the number of I/O requests that the process generates. An I/O-bound process spends more of its time seeking I/O operations than doing computational work. The CPU-bound process infrequently requests I/O operations and spends more of its time performing computational work.

24. Explain the concept of a context switch.
Ans: Whenever the CPU starts executing a new process, the old process's state must be preserved. The context of a process is represented by its process control block. Switching the CPU to another process requires performing a state save of the current process and a state restore of a different process. This task is known as a context switch. When a context switch occurs, the kernel saves the context of the old process in its PCB and loads the saves context of the new process scheduled to run.

25. Explain the fundamental differences between the UNIX `fork()` and Windows `CreateProcess()` functions.
Ans: Each function is used to create a child process. However, `fork()` has no parameters; `CreateProcess()` has ten. Furthermore, whereas the child process created with `fork()` inherits a copy of the address space of its parent, the `CreateProcess()` function requires specifying the address space of the child process.

26. Name the three types of sockets used in Java and the classes that implement them.
Ans: Connection-oriented (TCP) sockets are implemented with the `Socket` class. Connectionless (UDP) sockets use the `DatagramSocket` class. Finally, the `MulticastSocket` class is a subclass of the `DatagramSocket` class. A multicast socket allows data to be sent to multiple recipients.

27. What is a loopback and when is it used?
Ans: A loopback is a special IP address: 127.0.0.1. When a computer refers to IP address 127.0.0.1, it is referring to itself. When using sockets for client/server communication, this mechanism allows a client and server on the same host to communicate using the TCP/IP protocol.

28. Explain the purpose of external data representation (XDR).
Ans: Data can be represented differently on different machine architectures (e.g., *little-endian* vs. *big-endian*). XDR represents data independently of machine architecture. XDR is used when transmitting data between different machines using an RPC.

29. Explain the term marshalling.
Ans: Marshalling involves the packaging of parameters into a form that can be transmitted over the network. When the client invokes a remote procedure, the RPC system calls the appropriate stub, passing it the parameters provided to the remote procedure. This stub locates the port on the server and marshals the parameters. If necessary, return values are passed back to the client using the same technique.

30. Explain the terms "at most once" and "exactly once" and indicate how they relate to remote procedure calls.
Ans: Because a remote procedure call can fail in any number of ways, it is important to be able to handle such errors in the messaging system. The term "at most once" refers to ensuring that the server processes a particular message sent by the client only once and not multiple times. This is implemented by merely checking the timestamp of the message. The term "exactly once" refers to making sure that the message is executed on the server once and only once so that there is a guarantee that the server received and processed the message.

31. Describe two approaches to the binding of client and server ports during RPC calls.
Ans: First, the binding information may be predetermined, in the form of fixed port addresses. At compile time, an RPC call has a fixed port number associated with it. Second, binding can be done dynamically by a rendezvous mechanism. Typically, an operating system provides a rendezvous daemon on a fixed RPC port. A client then sends a message containing the name of the RPC to the rendezvous daemon requesting the port address of the RPC it needs to execute. The port number is returned, and the RPC calls can be sent to that port until the process terminates (or the server crashes).

32. Ordinarily the `exec()` system call follows the `fork()`. Explain what would happen if a programmer were to inadvertently place the call to `exec()` before the call to `fork()`.
Ans: Because `exec()` overwrites the process, we would never reach the call to `fork()` and hence, no new processes would be created. Rather, the program specified in the parameter to `exec()` would be run instead.

33. Explain why Google Chrome uses multiple processes.
Ans: Each website opens up in a separate tab and is represented with a separate renderer process. If that webpage were to crash, only the process representing that the tab would be affected, all other sites (represented as separate tabs/processes) would be unaffected.

34. Describe how UNIX and Linux manage orphan processes.
Ans: If a parent terminates without first calling `wait()`, its children are considered orphan processes. Linux and UNIX assign the init process as the new parent of orphan processes and init periodically calls `wait()` which allows any resources allocated to terminated processes to be reclaimed by the operating system.

## True/False

35. All processes in UNIX first translate to a zombie process upon termination. **True**

36. The difference between a program and a process is that a program is an active entity while a process is a passive entity. **False**

37. The `exec()` system call creates a new process. **False**

38. All access to POSIX shared memory requires a system call. **False**

39. Local Procedure Calls in Windows XP are similar to Remote Procedure Calls. **True**

40. For a single-processor system, there will never be more than one process in the Running state. **True**

41. Shared memory is a more appropriate IPC mechanism than message passing for distributed systems. **False**

42. Ordinary pipes in UNIX require a parent-child relationship between the communicating processes. **True**

43. Ordinary pipes in Windows require a parent-child relationship between the communicating processes. **True**

44. Using a section object to pass messages over a connection port avoids data copying. **True**

45. A socket is identified by an IP address concatenated with a port number. **True**

46. Sockets are considered a high-level communications scheme. **False**

47. The Mach operating system treats system calls with message passing. **True**

48. Named pipes continue to exist in the system after the creating process has terminated. **True**

49. A new browser process is create by the Chrome browser for every new website that is visited. **False**

50. The iOS mobile operating system only supports a limited form of multitasking. **True**

# 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

22. _____ involves distributing tasks across multiple computing cores.
A) Concurrency
B) Task parallelism
C) Data parallelism
D) Parallelism
Ans: B

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

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

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

## Short Answer

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

## True/False

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

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

43. Virtually all contemporary operating systems support kernel threads. **True**

44. Linux distinguishes between processes and threads. **False**

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

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

47. Deferred cancellation is preferred over asynchronous cancellation. **True**

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

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

50. It is possible to have concurrency without parallelism. **True**

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

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

53. Grand Central Dispatch requires multiple threads. **False**

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

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

# Chapter 5

## Multiple Choice

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

8. 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.
Ans: C

9. 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

10. 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
Ans: B

11. 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

12. In Solaris, if an interactive thread with priority 15 uses its entire time quantum, what is its priority recalculated to?
A) 51
B) 5
C) 160
D) It remains at 15
Ans: B

13. 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

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

15. 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

16. 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
Ans: D

17. 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

18. _____ 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
Ans: B

19. With _____ a thread executes on a processor until a long-latency event (i.e. a memory stall) occurs.
A) coarse-grained multithreading
B) fine-grained multithreading
C) virtualization
D) multicore processors
Ans: A

20. 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

21. 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

22. 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*
Ans: A

23. 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.
Ans: C

24. 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.
Ans: A

25. 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
Ans: D

## Short Answer

26. 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.

27. 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.

28. 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.

29. 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.
30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

## True/False

38. In preemptive scheduling, the sections of code affected by interrupts must be guarded from simultaneous use. **True**

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

40. The most complex scheduling algorithm is the multilevel feedback-queue algorithm. **True**

41. Load balancing is typically only necessary on systems with a common run queue. **False**

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

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

44. A Solaris interactive thread with priority 15 has a higher relative priority than an interactive thread with priority 20. **False**

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

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

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

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

49. Load balancing algorithms have no impact on the benefits of processor affinity. **False**

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

51. Providing a preemptive, priority-based scheduler guarantees hard real-time functionality. **False**

52. In hard real-time systems, interrupt latency must be bounded. **True**

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

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

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

56. The Completely Fair Scheduler (CFS) is the default scheduler for Linux systems. **True**

# Chapter 6

## Multiple Choice

1. A race condition ____.
A) results when several threads try to access the same data concurrently
B) results when several threads try to access and modify the same data concurrently
C) will result only if the outcome of execution does not depend on the order in which instructions are executed
D) None of the above
Ans: B

2. An instruction that executes atomically ____.
A) must consist of only one machine instruction
B) executes as a single, uninterruptible unit
C) cannot be used to solve the critical section problem
D) All of the above
Ans: B

3. A counting semaphore ____.
A) is essentially an integer variable
B) is accessed through only one standard operation
C) can be modified simultaneously by multiple threads
D) cannot be used to control access to a thread's critical sections
Ans: A

4. A mutex lock ____.
A) is exactly like a counting semaphore
B) is essentially a boolean variable
C) is not guaranteed to be atomic
D) can be used to eliminate busy waiting
Ans: B

5. In Peterson's solution, the ____ variable indicates if a process is ready to enter its critical section.
A) `turn` B) `lock`
C) `flag[i]` D) `turn[i]`
Ans: C

6. The first readers-writers problem ____.
A) requires that, once a writer is ready, that writer performs its write as soon as possible.
B) is not used to test synchronization primitives.
C) requires that no reader will be kept waiting unless a writer has already obtained permission to use the shared database.
D) requires that no reader will be kept waiting unless a reader has already obtained permission to use the shared database.
Ans: C

7. A ___ type presents a set of programmer-defined operations that are provided mutual exclusion within it.
A) transaction B) signal
C) binary D) monitor
Ans: D

8. _____ occurs when a higher-priority process needs to access a data structure that is currently being accessed by a lower-priority process.
A) Priority inversion
B) Deadlock
C) A race condition
D) A critical section
Ans: A

9. What is the correct order of operations for protecting a critical section using mutex locks?
A) `release()` followed by `acquire()`
B) `acquire()` followed by `release()`
C) `wait()` followed by `signal()`
D) `signal()` followed by `wait()`
Ans: B

10. What is the correct order of operations for protecting a critical section using a binary semaphore?
A) `release()` followed by `acquire()`
B) `acquire()` followed by `release()`
C) `wait()` followed by `signal()`
D) `signal()` followed by `wait()`
Ans: C

11. _____ is not a technique for handling critical sections in operating systems.
A) Nonpreemptive kernels
B) Preemptive kernels
C) Spinlocks
D) Peterson's solution
Ans: D

12. A solution to the critical section problem does not have to satisfy which of the following requirements?
A) mutual exclusion
B) progress
C) atomicity
D) bounded waiting
Ans: C

13. A(n) _____ refers to where a process is accessing/updating shared data.
A) critical section
B) entry section
C) mutex
D) test-and-set
Ans: A

14. _____ can be used to prevent busy waiting when implementing a semaphore.
A) Spinlocks
B) Waiting queues
C) Mutex lock
D) Allowing the `wait()` operation to succeed
Ans: B

15. Assume an adaptive mutex is used for accessing shared data on a Solaris system with multiprocessing capabilities. Which of the following statements is not true?
A) A waiting thread may spin while waiting for the lock to become available.
B) A waiting thread may sleep while waiting for the lock to become available.
C) The adaptive mutex is only used to protect short segments of code.
D) Condition variables and semaphores are never used in place of an adaptive mutex.
Ans: D

16. What is the purpose of the mutex semaphore in the implementation of the bounded-buffer problem using semaphores?
A) It indicates the number of empty slots in the buffer.
B) It indicates the number of occupied slots in the buffer.
C) It controls access to the shared buffer.
D) It ensures mutual exclusion.
Ans: D

17. How many philosophers may eat simultaneously in the Dining Philosophers problem with 5 philosophers?
A) 1 B) 2
C) 3 D) 5
Ans: B
18. Which of the following statements is true?
A) A counting semaphore can never be used as a binary semaphore.
B) A binary semaphore can never be used as a counting semaphore.
C) Spinlocks can be used to prevent busy waiting in the implementation of semaphore.
D) Counting semaphores can be used to control access to a resource with a finite number of instances.
Ans: C

19. _____ is/are not a technique for managing critical sections in operating systems.
A) Peterson's solution
B) Preemptive kernel
C) Nonpreemptive kernel
D) Semaphores
Ans: A

20. When using semaphores, a process invokes the `wait()` operation before accessing its critical section, followed by the `signal()` operation upon completion of its critical section. Consider reversing the order of these two operations—first calling `signal()`, then calling `wait()`. What would be a possible outcome of this?
A) Starvation is possible.
B) Several processes could be active in their critical sections at the same time.
C) Mutual exclusion is still assured.
D) Deadlock is possible.
Ans: B

21. Which of the following statements is true?
A) Operations on atomic integers do not require locking.
B) Operations on atomic integers do require additional locking.
C) Linux only provides the `atomic_inc()` and `atomic_sub()` operations.
D) Operations on atomic integers can be interrupted.
Ans: A

22. A(n) _____ is a sequence of read-write operations that are atomic.
A) atomic integer
B) semaphore
C) memory transaction
D) mutex lock
Ans: C

23. The OpenMP `#pragma omp critical` directive _____.
A) behaves much like a mutex lock
B) does not require programmers to identify critical sections
C) does not guarantee prevention of race conditions
D) is similar to functional languages
Ans: A

24. Another problem related to deadlocks is _____.
A) race conditions
B) critical sections
C) spinlocks
D) indefinite blocking
Ans: D

## Short Answer

25. What three conditions must be satisfied in order to solve the critical section problem?
Ans: In a solution to the critical section problem, no thread may be executing in its critical section if a thread is currently executing in its critical section. Furthermore, only those threads that are not executing in their critical sections can participate in the decision on which process will enter its critical section next. Finally, a bound must exist on the number of times that other threads are allowed to enter their critical state after a thread has made a request to enter its critical state.

26. Explain two general approaches to handle critical sections in operating systems.
Ans: Critical sections may use preemptive or nonpreemptive kernels. A preemptive kernel allows a process to be preempted while it is running in kernel mode. A nonpreemptive kernel does not allow a process running in kernel mode to be preempted; a kernel-mode process will run until it exits kernel mode, blocks, or voluntarily yields control of the CPU. A nonpreemptive kernel is essentially free from race conditions on kernel data structures, as the contents of this register will be saved and restored by the interrupt handler.

27. Write two short methods that implement the simple semaphore `wait()` and `signal()` operations on global variable `S`.
Ans:
```
wait (S) {
  while (S <= 0);

  S--;

}

signal (S) {

  S++;
```

28. Explain the difference between the first readers–writers problem and the second readers–-writers problem.
Ans: The first readers–writers problem requires that no reader will be kept waiting unless a writer has already obtained permission to use the shared database; whereas the second readers–writers problem requires that once a writer is ready, that writer performs its write as soon as possible.

29. Describe the dining-philosophers problem and how it relates to operating systems.
Ans: The scenario involves five philosophers sitting at a round table with a bowl of food and five chopsticks. Each chopstick sits between two adjacent philosophers. The philosophers are allowed to think and eat. Since two chopsticks are required for each philosopher to eat, and only five chopsticks exist at the table, no two adjacent philosophers may be eating at the same time. A scheduling problem arises as to who gets to eat at what time. This problem is similar to the problem of scheduling processes that require a limited number of resources.

30. What is the difference between software transactional memory and hardware transactional memory?
Ans: Software transactional memory (STM) implements transactional memory entirely in software, no special hardware is required. STM works by inserting instrumentation code inside of transaction blocks and typically requires the support of a compiler. Hardware transactional memory (HTM) implements transactional memory entirely in hardware using cache hierarchies and cache coherency protocols to resolve conflicts when shared data resides in separate caches.

31. Assume you had a function named update() that updates shared data. Illustrate how a mutex lock named mutex might be used to prevent a race condition in update().
Ans:

```
void update()
{
	mutex.acquire();

	// update shared data

	mutex.release();
}
```

32. Describe the turnstile structure used by Solaris for synchronization.
Ans: Solaris uses turnstiles to order the list of threads waiting to acquire either an adaptive mutex or a reader–writer lock. The turnstile is a queue structure containing threads blocked on a lock. Each synchronized object with at least one thread blocked on the object's lock requires a separate turnstile. However, rather than associating a turnstile with each synchronized object, Solaris gives each kernel thread its own turnstile.

33. Explain the role of the variable `preempt_count` in the Linux kernel.
Ans: Each task maintains a value `preempt_count` which is the number of locks held by each task. When a lock is acquired, `preempt_count` is incremented. When a lock is released, `preempt_count` is decremented. If the task currently running in the kernel has a value of `preempt_count` > 0, the kernel cannot be preempted as the task currently holds a lock.
If the count is zero, the kernel can be preempted.

34. Describe how an adaptive mutex functions.
Ans: An adaptive mutex is used in the Solaris operating system to protect access to shared data. On a multiprocessor system, an adaptive mutex acts as a standard semaphore implemented as a spinlock. If the shared data being accessed is already locked and the thread holding that lock is running on another CPU, the thread spins while waiting for the lock to be released, and the data to become available. If the thread holding the lock is not in the run state, the waiting thread sleeps until the lock becomes available. On a single processor system, spinlocks are not used and the waiting thread always sleeps until the lock becomes available.

35. Describe a scenario when using a reader–writer lock is more appropriate than another synchronization tool such as a semaphore.
Ans: A tool such as a semaphore only allows one process to access shared data at a time. Reader–writer locks are useful when it is easy to distinguish if a process is only reading or reading/writing shared data. If a process is only reading shared data, it can access the shared data concurrently with other readers. In the case when there are several readers, a reader–writer lock may be much more efficient.
36. Explain how Linux manages race conditions on single-processor systems such as embedded devices.
Ans: On multiprocessor machines, Linux uses spin locks to manage race conditions. However, as spin locks are not appropriate on single processor machines, Linux instead disables kernel preemption which acquiring a spin lock, and enables it after releasing the spin lock.

## True/False

37. Race conditions are prevented by requiring that critical regions be protected by locks. **True**

38. The value of a counting semaphore can range only between 0 and 1. **False**

39. A deadlock-free solution eliminates the possibility of starvation. **False**

40. The local variables of a monitor can be accessed by only the local procedures. **True**

41. Every object in Java has associated with it a single lock. **True**

42. Monitors are a theoretical concept and are not practiced in modern programming languages **False**

43. A thread will immediately acquire a dispatcher lock that is the signaled state. **True**

44. Mutex locks and counting semaphores are essentially the same thing. **False**

45. Mutex locks and binary semaphores are essentially the same thing. **True**

46. A nonpreemptive kernel is safe from race conditions on kernel data structures. **True**

47. Linux mostly uses atomic integers to manage race conditions within the kernel. **False**

# Chapter 7

## Multiple Choice

1. A deadlocked state occurs whenever ____.
A) a process is waiting for I/O to a device that does not exist
B) the system has no available free resources
C) every process in a set is waiting for an event that can only be caused by another process in the set
D) a process is unable to release its request for a resource after use
Ans: C

2. One necessary condition for deadlock is ____, which states that at least one resource must be held in a nonsharable mode.
A) hold and wait
B) mutual exclusion
C) circular wait
D) no preemption
Ans: B

3. One necessary condition for deadlock is _____, which states that a process must be holding one resource and waiting to acquire additional resources.
A) hold and wait
B) mutual exclusion
C) circular wait
D) no preemption
Ans: A

4. One necessary condition for deadlock is _____, which states that a resource can be released only voluntarily by the process holding the resource.
A) hold and wait
B) mutual exclusion
C) circular wait
D) no preemption
Ans: D

5. One necessary condition for deadlock is _____, which states that there is a chain of waiting processes whereby $P_0$ is waiting for a resource held by $P_1$, $P_1$ is waiting for a resource held by $P_2$, and $P_n$ is waiting for a resource held by $P_0$.
A) hold and wait
B) mutual exclusion
C) circular wait
D) no preemption
Ans: C

6. The *witness* software product is a ____.
A) lock-order verifier that uses mutual-exclusion locks to protect critical sections
B) modeler to develop resource allocation graphs
C) driver that can be used to prevent mutual exclusion for nonsharable resources
D) implementation of the banker's algorithm available for most operating systems
Ans: A

7. In a system resource-allocation graph, ____.
A) a directed edge from a process to a resource is called an assignment edge
B) a directed edge from a resource to a process is called a request edge
C) a directed edge from a process to a resource is called a request edge
D) None of the above
Ans: C

8. A cycle in a resource-allocation graph is ____.
A) a necessary and sufficient condition for deadlock in the case that each resource has more than one instance
B) a necessary and sufficient condition for a deadlock in the case that each resource has exactly one instance
C) a sufficient condition for a deadlock in the case that each resource has more than once instance
D) is neither necessary nor sufficient for indicating deadlock in the case that each resource has exactly one instance
Ans: B

9. To handle deadlocks, operating systems most often _____.
A) pretend that deadlocks never occur
B) use protocols to prevent or avoid deadlocks
C) detect and recover from deadlocks
D) None of the above
Ans: A

10. Which of the following statements is true?
A) A safe state is a deadlocked state.
B) A safe state may lead to a deadlocked state.
C) An unsafe state is necessarily, and by definition, always a deadlocked state.
D) An unsafe state may lead to a deadlocked state.
Ans: D

11. Suppose that there are ten resources available to three processes. At time 0, the following data is collected. The table indicates the process, the maximum number of resources needed by the process, and the number of resources currently owned by each process. Which of the following correctly characterizes this state?

| Process | Maximum Needs | Currently Owned |
|---|---|---|
| $P_0$ | 10 | 4 |
| $P_1$ | 3 | 1 |
| $P_2$ | 6 | 4 |

A) It is safe.
B) It is not safe.
C) The state cannot be determined.
D) It is an impossible state.
Ans: B

12. Suppose that there are 12 resources available to three processes. At time 0, the following data is collected. The table indicates the process, the maximum number of resources needed by the process, and the number of resources currently owned by each process. Which of the following correctly characterizes this state?

| Process | Maximum Needs | Currently Owned |
|---|---|---|
| $P_0$ | 10 | 4 |
| $P_1$ | 3 | 2 |
| $P_2$ | 7 | 4 |

A) It is safe.
B) It is not safe.
C) The state cannot be determined.
D) It is an impossible state.
Ans: A

13. Which of the following data structures in the banker's algorithm is a vector of length *m*, where *m* is the number of resource types?
A) Need B) Allocation
C) Max D) Available
Ans: D

14. Assume there are three resources, $R_1$, $R_2$, and $R_3$, that are each assigned unique integer values 15, 10, and 25, respectively. What is a resource ordering which prevents a circular wait?
A) $R_1$, $R_2$, $R_3$
B) $R_3$, $R_2$, $R_1$
C) $R_3$, $R_1$, $R_2$
D) $R_2$, $R_1$, $R_3$
Ans: D

15. A _____ could be preempted from a process.
A) mutex lock B) CPU
C) semaphore D) file lock
Ans: B

## Short Answer

16. Explain what has to happen for a set of processes to achieve a deadlocked state.
Ans: For a set of processes to exist in a deadlocked state, every process in the set must be waiting for an event that can be caused only be another process in the set. Thus, the processes cannot ever exit this state without manual intervention.

17. Describe the four conditions that must hold simultaneously in a system if a deadlock is to occur.
Ans: For a set of processes to be deadlocked: at least one resource must remain in a nonsharable mode, a process must hold at least one resource and be waiting to acquire additional resources held by other processes, resources in the system cannot be preempted, and a circular wait has to exist between processes.

18. What are the three general ways that a deadlock can be handled?
Ans: A deadlock can be prevented by using protocols to ensure that a deadlock will never occur. A system may allow a deadlock to occur, detect it, and recover from it. Lastly, an operating system may just ignore the problem and pretend that deadlocks can never occur.

19. What is the difference between deadlock prevention and deadlock avoidance?
Ans: Deadlock prevention is a set of methods for ensuring that at least one of the necessary conditions for deadlock cannot hold. Deadlock avoidance requires that the operating system be given, in advance, additional information concerning which resources a process will request and use during its lifetime.

20. Describe two protocols to ensure that the hold-and-wait condition never occurs in a system.
Ans: One protocol requires each process to request and be allocated all its resources before it begins execution. We can implement this provision by requiring that system calls requesting resources for a process precede all other system calls. An alternative protocol allows a process to request resources only when it has none. A process may request some resources and use them. Before it can request any additional resources, however, it must release all the resources that it is currently allocated.

21. What is one way to ensure that a circular-wait condition does not occur?
Ans: One way to ensure that this condition never holds is to impose a total ordering of all resource types, and to require that each process requests resources in an increasing order of enumeration. This can be accomplished by assigning each resource type a unique integer number to determine whether one precedes another in the ordering.

22. What does a claim edge signify in a resource-allocation graph?
Ans: A claim edge indicates that a process may request a resource at some time in the future. This edge resembles a request edge in direction, but is represented in the graph by a dashed line.

23. Describe a wait-for graph and how it detects deadlock.
Ans: If all resources have only a single instance, then we can define a deadlock-detection algorithm that uses a variant of the resource-allocation graph, called a wait-for graph. We obtain this graph from the resource-allocation graph by removing the resource nodes and collapsing the appropriate edges. To detect deadlocks, the system needs to maintain the wait-for graph and periodically invoke an algorithm that searches for a cycle in the graph.

24. What factors influence the decision of when to invoke a detection algorithm?
Ans: The first factor is how often a deadlock is likely to occur; if deadlocks occur frequently, the detection algorithm should be invoked frequently. The second factor is how many processes will be affected by deadlock when it happens; if the deadlock-detection algorithm is invoked for every resource request, a considerable overhead in computation time will be incurred.

25. Describe two methods for eliminating processes by aborting a process.
Ans: The first method is to abort all deadlocked processes. Aborting all deadlocked processes will clearly break the deadlock cycle; however, the deadlocked processes may have to be computed for a long time, and results of these partial computations must be discarded and will probably have to be recomputed later. The second method is to abort one process at a time until the deadlock cycle is eliminated. Aborting one process at a time incurs considerable overhead, since, after each process is aborted, a deadlock-detection algorithm must be invoked to determine whether any processes are still deadlocked.

26. Name three issues that need to be addressed if a preemption is required to deal with deadlocks
Ans: First, the order of resources and processes that need to be preempted must be determined to minimize cost. Second, if a resource is preempted from a process, the process must be rolled back to some safe state and restarted from that state. The simplest solution is a total rollback. Finally, we must ensure that starvation does not occur from always preempting resources from the same process.

27. Describe how a safe state ensures deadlock will be avoided.
Ans: A safe state ensures that there is a sequence of processes to finish their program execution. Deadlock is not possible while the system is in a safe state. However, if a system goes from a safe state to an unsafe state, deadlock is possible. One technique for avoiding deadlock is to ensure that the system always stays in a safe state. This can be done by only assigning a resource as long as it maintains the system in a safe state.

## True/False

28. The circular-wait condition for a deadlock implies the hold-and-wait condition. **True**

29. If a resource-allocation graph has a cycle, the system must be in a deadlocked state. **False**

30. Protocols to prevent hold-and-wait conditions typically also prevent starvation. **False**

31. The wait-for graph scheme is not applicable to a resource allocation system with multiple instances of each resource type. **True**

32. Ordering resources and requiring the resources to be acquired in order prevents the circular wait from occurring and therefore prevents deadlock from occurring. **False**

33. The banker's algorithm is useful in a system with multiple instances of each resource type. **True**

34. A system in an unsafe state will ultimately deadlock. **False**

35. Deadlock prevention and deadlock avoidance are essentially the same approaches for handling deadlock. **False**

# Chapter 8

## Multiple Choice

1. Absolute code can be generated for ____.
A) compile-time binding
B) load-time binding
C) execution-time binding
D) interrupt binding
Ans: A

2. _____ is the method of binding instructions and data to memory performed by most general-purpose operating systems.
A) Interrupt binding
B) Compile time binding
C) Execution time binding
D) Load-time binding
Ans: C

3. An address generated by a CPU is referred to as a ____.
A) physical address
B) logical address
C) post relocation register address
D) Memory-Management Unit (MMU) generated address
Ans: B

4. Suppose a program is operating with execution-time binding and the physical address generated is 300. The relocation register is set to 100. What is the corresponding logical address?
A) 199 B) 201
C) 200 D) 300
Ans: C

5. The mapping of a logical address to a physical address is done in hardware by the _____.
A) memory-management-unit (MMU)
B) memory address register
C) relocation register
D) dynamic loading register
Ans: A

6. In a dynamically linked library, ____.
A) loading is postponed until execution time
B) system language libraries are treated like any other object module
C) more disk space is used than in a statically linked library
D) a stub is included in the image for each library-routine reference
Ans: D

7. The _____ binding scheme facilitates swapping.
A) interrupt time
B) load time
C) assembly time
D) execution time
Ans: D

8. The roll out, roll in variant of swapping is used ____.
A) when a backing store is not necessary
B) for the round-robin scheduling algorithm
C) for priority-based scheduling algorithms
D) when the load on the system has temporarily been reduced
Ans: C

9. _____ is the dynamic storage-allocation algorithm which results in the smallest leftover hole in memory.
A) First fit
B) Best fit
C) Worst fit
D) None of the above
Ans: B

10. _____ is the dynamic storage-allocation algorithm which results in the largest leftover hole in memory.
A) First fit B) Best fit
C) Worst fit D) None of the above
Ans: C

11. Which of the following is true of compaction?
A) It can be done at assembly, load, or execution time.
B) It is used to solve the problem of internal fragmentation.
C) It cannot shuffle memory contents.
D) It is possible only if relocation is dynamic and done at execution time.
Ans: D

12. A(n) ____ page table has one page entry for each real page (or frame) of memory.
A) inverted
B) clustered
C) forward-mapped
D) virtual
Ans: A

13. Consider a logical address with a page size of 8 KB. How many bits must be used to represent the page offset in the logical address?
A) 10 B) 8
C) 13 D) 12
Ans: C

14. Consider a logical address with 18 bits used to represent an entry in a conventional page table. How many entries are in the conventional page table?
A) 262144
B) 1024
C) 1048576
D) 18
Ans: A

15. Assume a system has a TLB hit ratio of 90%. It requires 15 nanoseconds to access the TLB, and 85 nanoseconds to access main memory. What is the effective memory access time in nanoseconds for this system?
A) 108.5 B) 100
C) 22 D) 176.5
Ans: A

16. Given the logical address 0xAEF9 (in hexadecimal) with a page size of 256 bytes, what is the page number?
A) 0xAE
B) 0xF9
C) 0xA
D) 0x00F9
Ans: A

17. Given the logical address 0xAEF9 (in hexadecimal) with a page size of 256 bytes, what is the page offset?
A) 0xAE
B) 0xF9
C) 0xA
D) 0xF900
Ans: B

18. Consider a 32-bit address for a two-level paging system with an 8 KB page size. The outer page table has 1024 entries. How many bits are used to represent the second-level page table?
A) 10 B) 8
C) 12 D) 9
Ans: D

19. With segmentation, a logical address consists of _____.
A) segment number and offset
B) segment name and offset
C) segment number and page number
D) segment table and segment number
Ans: A

20. Which of the following data structures is appropriate for placing into its own segment?
A) heap
B) kernel code and data
C) user code and data
D) all of the above
Ans: D

21. Assume the value of the base and limit registers are 1200 and 350 respectively. Which of the following addresses is legal?
A) 355 B) 1200
C) 1551 D) all of the above
Ans: B

22. A(n) _____ matches the process with each entry in the TLB.
A) address-space identifier
B) process id
C) stack
D) page number
Ans: A

23. Which of the following statements are true with respect to hashed page tables?
A) They only work for sparse address spaces.
B) The virtual address is used to hash into the hash table.
C) A common approach for handling address spaces larger than 32 bits.
D) Hash table collisions do not occur because of the importance of paging.
Ans: C

24. Which of the following statements regarding the ARM architecture are false?
A) There are essentially four different page ranging from 4-KB to 16-MB in size.
B) There are two different levels of TLB.
C) One or two level paging may be used.
D) The micro TLB must be flushed at each context switch.
Ans: D

25. Which of the following is not a reason explaining why mobile devices generally do not support swapping?
A) Limited space constraints of flash memory.
B) Small size of mobile applications do not require use of swap space.
C) Limited number of writes of flash memory.
D) Poor throughput between main memory and flash memory.
Ans: B

## Short Answer

26. What is the advantage of using dynamic loading?
Ans: With dynamic loading a program does not have to be stored, in its entirety, in main memory. This allows the system to obtain better memory-space utilization. This also allows unused routines to stay out of main memory so that memory can be used more effectively. For example, code used to handle an obscure error would not always use up main memory.

27. What is the context switch time, associated with swapping, if a disk drive with a transfer rate of 2 MB/s is used to swap out part of a program that is 200 KB in size? Assume that no seeks are necessary and that the average latency is 15 ms. The time should reflect only the amount of time necessary to swap out the process.
Ans: 200KB / 2048 KB per second + 15 ms = 113 ms

28. When does external fragmentation occur?
Ans: As processes are loaded and removed from memory, the free memory space is broken into little pieces. External fragmentation exists when there is enough total memory space to satisfy a request, but the available spaces are not contiguous; storage is fragmented into a large number of small holes. Both the first-fit and best-fit strategies for memory allocation suffer from external fragmentation.

29. Distinguish between internal and external fragmentation.
Ans: Fragmentation occurs when memory is allocated and returned to the system. As this occurs, free memory is broken up into small chunks, often too small to be useful. External fragmentation occurs when there is sufficient total free memory to satisfy a memory request, yet the memory is not contiguous, so it cannot be assigned. Some contiguous allocation schemes may assign a process more memory than it actually requested (i.e. they may assign memory in fixed-block sizes). Internal fragmentation occurs when a process is assigned more memory than it has requested and the wasted memory fragment is internal to a process.

30. Explain the basic method for implementing paging.
Ans: Physical memory is broken up into fixed-sized blocks called frames while logical memory is broken up into equal-sized blocks called pages. Whenever the CPU generates a logical address, the page number and offset into that page is used, in conjunction with a page table, to map the request to a location in physical memory.

31. Describe how a transaction look-aside buffer (TLB) assists in the translation of a logical address to a physical address.
Ans: Typically, large page tables are stored in main memory, and a page-table base register points are saved to the page table. Therefore, two memory accesses are needed to access a byte (one for the page-table entry, one for the byte), causing memory access to be slowed by a factor of 2. The standard solution to this problem is to use a TLB, a special, small fast-lookup hardware cache. The TLB is associative, high speed memory. Each entry consists of a key and value. An item is compared with all keys simultaneously, and if the item is found, the corresponding value is returned.
32. How are illegal page addresses recognized and trapped by the operating system?
Ans: Illegal addresses are trapped by the use of a valid-invalid bit, which is generally attached to each entry in the page table. When this bit is set to "valid," the associated page is in the process's logical address space and is thus a legal (or valid) page. When the bit is set to "invalid," the page is not in the process's logical address space. The operating system sets this bit for each page to allow or disallow access to the page.

33. Describe the elements of a hashed page table.
Ans: A hashed page table contains hash values which correspond to a virtual page number. Each entry in the hash table contains a linked list of elements that hash to the same location (to handle collisions). Each element consists of three fields: (1) the virtual page number, (2) the value of the mapped page frame, and (3) a pointer to the next element in the linked list.

34. Briefly describe the segmentation memory management scheme. How does it differ from the paging memory management scheme in terms of the user's view of memory?
Ans: Segmentation views a logical address as a collection of segments. Each segment has a name and length. The addresses specify both the segment name and the offset within the segment. The user therefore specifies each address by two quantities: a segment name and an offset. In contrast, in a paging scheme, the user specifies a single address, which is partitioned by the hardware into a page number and an offset, all invisible to the programmer.

35. Describe the partitions in a logical-address space of a process in the IA-32 architecture.
Ans: The logical-address space is divided into two partitions. The first partition consists of up to 8 K segments that are private to that process. The second partition consists of up to 8 K segments that are shared among all the processes. Information about the first partition is kept in the local descriptor table (LDT); information about the second partition is kept in the global descriptor table (GDT).

36. How is a limit register used for protecting main memory?
Ans: When the CPU is executing a process, it generates a logical memory address that is added to a relocation register in order to arrive at the physical memory address actually used by main memory. A limit register holds the maximum logical address that the CPU should be able to access. If any logical address is greater than or equal to the value in the limit register, then the logical address is a dangerous address and an error results.

37. Using Figure 8.14, describe how a logical address is translated to a physical address.
Ans: A logical address is generated by the CPU. This logical address consists of a page number and offset. The TLB is first checked to see if the page number is present. If so, a TLB hit, the corresponding page frame is extracted from the TLB, thus producing the physical address. In the case of a TLB miss, the page table must be searched according to page number for the corresponding page frame.

38. Explain why mobile operating systems generally do not support paging.
Ans: Mobile operating systems typically do not support swapping because file systems are typically employed using flash memory instead of magnetic hard disks. Flash memory is typically limited in size as well as having poor throughput between flash and main memory. Additionally, flash memory can only tolerate a limited number of writes before it becomes less reliable.

39. Using Figure 8.26, describe how address translation is performed on ARM architectures.
Ans: ARM supports four different page sizes: 4-KB and 16-KB page use two-level paging, the larger 1-MB and 16-MB page sizes use single-level paging. The ARM architecture uses two levels of TLBs - at one level is the micro TLB which is in fact separate TLBs for data and instructions. At the inner level is a single main TLB. Address translation begins wit first searching the micro TLB, and in case of a TLB miss, the main TLB is then checked. If the reference is not in the main TLB, the page table must then be consulted.

## True/False

40. A relocation register is used to check for invalid memory addresses generated by a CPU. **False**

41. Reentrant code cannot be shared. **False**

42. There is a 1:1 correspondence between the number of entries in the TLB and the number of entries in the page table. **False**

43. Hierarchical page tables are appropriate for 64-bit architectures. **False**

43. The ARM architecture uses both single-level and two-level paging. **True**

44. Fragmentation does not occur in a paging system. **False**

45. Hashed page tables are particularly useful for processes with sparse address spaces. **True**

46. Inverted page tables require each process to have its own page table. **False**

47. Without a mechanism such as an address-space identifier, the TLB must be flushed during a context switch. **True**

48. A 32-bit logical address with 8 KB page size will have 1,000,000 entries in a conventional page table. **False**

49. Hashed page tables are commonly used when handling addresses larger than 32 bits. **True**

50. The x86-64 bit architecture only uses 48 of the 64 possible bits for representing virtual address space. **True**

51. Mobile operating systems typically support swapping. **False**

# Chapter 9

## Multiple Choice

1. Which of the following is a benefit of allowing a program that is only partially in memory to execute?
A) Programs can be written to use more memory than is available in physical memory.
B) CPU utilization and throughput is increased.
C) Less I/O is needed to load or swap each user program into memory.
D) All of the above
Ans: D

2. In systems that support virtual memory, ____.
A) virtual memory is separated from logical memory.
B) virtual memory is separated from physical memory.
C) physical memory is separated from secondary storage.
D) physical memory is separated from logical memory.
Ans: D

3. The `vfork()` system call in UNIX ____.
A) allows the child process to use the address space of the parent
B) uses copy-on-write with the `fork()` call
C) is not intended to be used when the child process calls `exec()` immediately after creation
D) duplicates all pages that are modified by the child process
Ans: A

4. Suppose we have the following page accesses: 1 2 3 4 2 3 4 1 2 1 1 3 1 4 and that there are three frames within our system. Using the FIFO replacement algorithm, what is the number of page faults for the given reference string?
A) 14 B) 8
C) 13 D) 10
Ans: B

5. Suppose we have the following page accesses: 1 2 3 4 2 3 4 1 2 1 1 3 1 4 and that there are three frames within our system. Using the FIFO replacement algorithm, what will be the final configuration of the three frames following the execution of the given reference string?
A) 4, 1, 3 B) 3, 1, 4
C) 4, 2, 3 D) 3, 4, 2
Ans: D

6. Suppose we have the following page accesses: 1 2 3 4 2 3 4 1 2 1 1 3 1 4 and that there are three frames within our system. Using the LRU replacement algorithm, what is the number of page faults for the given reference string?
A) 14 B) 13
C) 8 D) 10
Ans: C

7. Given the reference string of page accesses: 1 2 3 4 2 3 4 1 2 1 1 3 1 4 and a system with three page frames, what is the final configuration of the three frames after the LRU algorithm is applied?
A) 1, 3, 4 B) 3, 1, 4
C) 4, 1, 2 D) 1, 2, 3
Ans: B

8. Belady's anomaly states that ____.
A) giving more memory to a process will improve its performance
B) as the number of allocated frames increases, the page-fault rate may decrease for all page replacement algorithms
C) for some page replacement algorithms, the page-fault rate may decrease as the number of allocated frames increases
D) for some page replacement algorithms, the page-fault rate may increase as the number of allocated frames increases
Ans: D

9. Optimal page replacement ____.
A) is the page-replacement algorithm most often implemented
B) is used mostly for comparison with other page-replacement schemes
C) can suffer from Belady's anomaly
D) requires that the system keep track of previously used pages
Ans: B

10. In the enhanced second chance algorithm, which of the following ordered pairs represents a page that would be the best choice for replacement?
A) (0,0) B) (0,1)
C) (1,0) D) (1,1)
Ans: A

11. The _____ allocation algorithm allocates available memory to each process according to its size.
A) equal
B) global
C) proportional
D) slab
Ans: C

12. The ____ is the number of entries in the TLB multiplied by the page size.
A) TLB cache
B) page resolution
C) TLB reach
D) hit ratio
Ans: C

13. _____ allows the parent and child processes to initially share the same pages, but when either process modifies a page, a copy of the shared page is created.
A) copy-on-write
B) zero-fill-on-demand
C) memory-mapped
D) virtual memory fork
Ans: A

14. _____ is the algorithm implemented on most systems.
A) FIFO
B) Least frequently used
C) Most frequently used
D) LRU
Ans: D

15. _____ occurs when a process spends more time paging than executing.
A) Thrashing
B) Memory-mapping
C) Demand paging
D) Swapping
Ans: A

16. Windows uses a local page replacement policy _____.
A) when a process exceeds its working set minimum
B) when a process exceeds its working set maximum
C) when the system undergoes automatic working set trimming
D) under all circumstances
Ans: B

17. Which of the following statements is false with regard to Solaris memory management?
A) The speed at which pages are examined (the scanrate) is constant.
B) The pageout process only runs if the number of free pages is less than lotsfree.
C) An LRU approximation algorithm is employed.
D) Pages selected for replacement may be reclaimed before being placed on the free list.
Ans: A

18. What size segment will be allocated for a 39 KB request on a system using the Buddy system for kernel memory allocation?
A) 39 KB B) 42 KB
C) 64 KB D) None of the above
Ans: C

19. Which of the following statements is false with regard to allocating kernel memory?
A) Slab allocation does not suffer from fragmentation.
B) Adjacent segments can be combined into one larger segment with the buddy system.
C) Because the kernel requests memory of varying sizes, some of which may be quite small, the system does not have to be concerned about wasting memory.
D) The slab allocator allows memory requests to be satisfied very quickly.
Ans: C

20. The _____ is an approximation of a program's locality.
A) locality model
B) working set
C) page fault frequency
D) page replacement algorithm
Ans: B

21. _____ allows a portion of a virtual address space to be logically associated with a file.
A) Memory-mapping
B) Shared memory
C) Slab allocation
D) Locality of reference
Ans: A

22. Systems in which memory access times vary significantly are known as _____.
A) memory-mapped I/O
B) demand-paged memory
C) non-uniform memory access
D) copy-on-write memory
Ans: C

23. Which of the following is considered a benefit when using the slab allocator?
A) Memory is allocated using a simple power-of-2 allocator.
B) It allows kernel code and data to be efficiently paged.
C) It allows larger segments to be combined using coalescing.
D) There is no memory fragmentation.
Ans: D

## Short Answer

24. Explain the distinction between a demand-paging system and a paging system with swapping.
Ans: A demand-paging system is similar to a paging system with swapping where processes reside in secondary memory. With demand paging, when a process is executed, it is swapped into memory. Rather than swapping the entire process into memory, however, a lazy swapper is used. A lazy swapper never swaps a page into memory unless that page will be needed. Thus, a paging system with swapping manipulates entire processes, whereas a demand pager is concerned with the individual pages of a process.

25. Explain the sequence of events that happens when a page-fault occurs.
Ans: When the operating system cannot load the desired page into memory, a page-fault occurs. First, the memory reference is checked for validity. In the case of an invalid request, the program will be terminated. If the request was valid, a free frame is located. A disk operation is then scheduled to read the page into the frame just found, update the page table, restart the instruction that was interrupted because of the page fault, and use the page accordingly.

26. How is the effective access time computed for a demand-paged memory system?
Ans: In order to compute the effective access time, it is necessary to know the average memory access time of the system, the probability of a page fault, and the time necessary to service a page fault. The effective access time can then be computed using the formula:

effective access time = (1 – probability of page fault) * memory access time + probability of page fault * page fault time.

27. How does the second-chance algorithm for page replacement differ from the FIFO page replacement algorithm?
Ans: The second-chance algorithm is based on the FIFO replacement algorithm and even degenerates to FIFO in its worst-case scenario. In the second-chance algorithm, a FIFO replacement is implemented along with a reference bit. If the reference bit is set, then it is cleared, the page's arrival time is set to the current time, and the program moves along in a similar fashion through the pages until a page with a cleared reference bit is found and subsequently replaced.

28. Explain the concept behind prepaging.
Ans: Paging schemes, such as pure demand paging, result in large amounts of initial page faults as the process is started. Prepaging is an attempt to prevent this high level of initial paging by bringing into memory, at one time, all of the pages that will be needed by the process.

29. Why doesn't a local replacement algorithm solve the problem of thrashing entirely?
Ans: With local replacement, if one process starts thrashing, it cannot steal frames from another process and cause the latter to thrash as well. However, if processes are thrashing, they will be in the queue for the paging device most of the time. The average service time for a page fault will increase because of the longer average queue for the paging device. Thus, the effective access time will increase, even for a process that is not thrashing.

30. Explain the difference between programmed I/O (PIO) and interrupt driven I/O.
Ans: To send out a long string of bytes through a memory-mapped serial port, the CPU writes one data byte to the data register to signal that it is ready for the next byte. If the CPU uses polling to watch the control bit, constantly looping to see whether the device is ready, this method of operation is called programmer I/O. If the CPU does not poll the control bit, but instead receives an interrupt when the device is ready for the next byte, the data transfer is said to be interrupt driven.
31. What are the benefits of using slab allocation to allocate kernel memory?
Ans: The slab allocator provides two main benefits. First, no memory is wasted due to fragmentation. When the kernel requests memory for an object, the slab allocator returns the exact amount of memory required to represent the object. Second, memory requests can be satisfied quickly. Objects are created in advance and can be quickly allocated. Also, released objects are returned to the cache and marked as free, thus making them immediately available for subsequent requests.

32. How are lock bits useful in I/O requests?
Ans: A lock bit is associated with every frame. If a frame is locked, it cannot be selected for replacement. To write a block on tape, we lock into memory the pages containing the block. The system then continues as usual with other processes if the I/O request is in a queue for that I/O device. This avoids the replacement of the pages for other processes and the possible unavailability of those pages when the I/O request advances to the head of the device queue. When the I/O is complete, the pages are unlocked.

33. Explain how copy-on-write operates.
Ans: Copy-on-write (COW) initially allows a parent and child process to share the same pages. As long as either process is only reading—and not modifying—the shared pages, both processes can share the same pages, thus increasing system efficiency. However, as soon as either process modifies a shared page, a copy of that shared page is created, thus providing each process with its own private page. For example, assume an integer X whose value is 5 is in a shared page marked as COW. The parent process then proceeds to modify X, changing its value to 10. Since this page is marked as COW, a copy of the page is created for the parent process, which changes the value of X to 10. The value of X remains at 5 for the child process.

34. Explain the distinction between global allocation versus local allocation.
Ans: When a process incurs a page fault, it must be allocated a new frame for bringing the faulting page into memory. The two general strategies for allocating a new frame are global and local allocation policies. In a global allocation scheme, a frame is allocated from any process in the system. Thus, if process A incurs a page fault, it may be allocated a page from process B. The page that is selected from process B may be based upon any of the page replacement algorithms such as LRU. Alternatively, a local allocation policy dictates that when a process incurs a page fault, it must select one of its own pages for replacement when allocating a new page.

35. Discuss two strategies for increasing TLB reach.
Ans: TLB reach refers to the amount of memory accessible from the TLB and is the page size multiplied by the number of entries in the TLB. Two possible approaches for increasing TLB reach are (1) increasing the number of entries in the TLB, and (2) increasing the page size. Increasing the number of entries in the TLB is a costly strategy as the TLB consists of associative memory, which is both costly and power hungry. For example, by doubling the number of entries in the TLB, the TLB reach is doubled. However, increasing the page size (or providing multiple page sizes) allows system designers to maintain the size of the TLB, and yet significantly increase the TLB reach. For this reason, recent trends have moved towards increasing page sizes for increasing TLB reach.

36. What is the benefit of using sparse addresses in virtual memory?
Ans: Virtual address spaces that include holes between the heap and stack are known as sparse address spaces. Using a sparse address space is beneficial because the holes can be filled as the stack or heap segments grow, or when we wish to dynamically link libraries (or possibly other shared objects) during program execution.

37. Explain the usefulness of a modify bit.
Ans: A modify bit is associated with each page frame. If a frame is modified (i.e. written), the modify bit is then set. The modify bit is useful when a page is selected for replacement. If the bit is not set (the page was not modified), the page does not need to be written to disk. If the modify bit is set, the page needs to be written to disk when selected for replacement.

## True/False

38. In general, virtual memory decreases the degree of multiprogramming in a system. **False**

39. Stack algorithms can never exhibit Belady's anomaly. **True**

40. If the page-fault rate is too high, the process may have too many frames. **False**

41. The buddy system for allocating kernel memory is very likely to cause fragmentation within the allocated segments. **True**

42. On a system with demand-paging, a process will experience a high page fault rate when the process begins execution. **True**

43. On systems that provide it, `vfork()` should always be used instead of `fork()`. **False**

44. Only a fraction of a process's working set needs to be stored in the TLB. **False**

45. Solaris uses both a local and global page replacement policy. **False**

46. Windows uses both a local and global page replacement policy. **False**

47. A page fault must be preceded by a TLB miss. **True**

48. Non-uniform memory access has little effect on the performance of a virtual memory system. **False**

49. In Linux, a slab may only be either full or empty. **False**

# Chapter 10

## Multiple Choice

1. A(n) ____ file is a sequence of functions.
A) text
B) source
C) object
D) executable
Ans: B

2. A(n) ____ file is a sequence of bytes organized into blocks understandable by the system's linker.
A) text
B) source
C) object
D) executable
Ans: C

3. A(n) ____ file is a series of code sections that the loader can bring into memory and execute.
A) text
B) source
C) object
D) executable
Ans: D

4. In an environment where several processes may open the same file at the same time, ____.
A) the operating system typically uses only one internal table to keep track of open files
B) the operating system typically uses two internal tables called the system-wide and per-disk tables to keep track of open files
C) the operating system typically uses three internal tables called the system-wide, per-disk, and per-partition tables to keep track of open files
D) the operating system typically uses two internal tables called the system-wide and per-process tables to keep track of open files
Ans: D

5. Suppose that the operating system uses two internal tables to keep track of open files. Process A has two files open and process B has three files open. Two files are shared between the two processes. How many entries are in the per-process table of process A, the per-process table of process B, and the system-wide tables, respectively?
A) 5, 5, 5
B) 2, 3, 3
C) 2, 3, 5
D) 2, 3, 1
Ans: B

6. A shared lock ____.
A) behaves like a writer lock
B) ensures that a file can have only a single concurrent shared lock
C) behaves like a reader lock
D) will prevent all other processes from accessing the locked file
Ans: C

7. An exclusive lock ____.
A) behaves like a writer lock
B) ensures that a file can have only a single concurrent shared lock
C) behaves like a reader lock
D) will prevent all other processes from accessing the locked file
Ans: A

8. The simplest file access method is ____.
A) sequential access
B) logical access
C) relative access
D) direct access
Ans: A

9. A _____ is used on UNIX systems at the beginning of some files to roughly indicate the type of the file.
A) file extension
B) creator name
C) hint
D) magic number
Ans: D

10. Which of the following is true of the direct-access method?
A) It is the most common mode of access.
B) It allows programs to read and write records in no particular order.
C) Files are made up of variable-length records.
D) It is not a good method for accessing large amounts of data quickly.
Ans: B

11. Which of the following is true of the tree-structured directory structure?
A) Users cannot create their own subdirectories.
B) Users cannot acquire permission to access the files of other users.
C) Directories can share subdirectories and files.
D) It is the most common directory structure.
Ans: D

12. An acyclic-graph directory structure ____.
A) does not allow the sharing of files.
B) allows the sharing of subdirectories and files.
C) is less complicated than a simple tree-structured directory structure.
D) is less flexible than a simple tree-structured directory structure.
Ans: B

13. The path name `/home/people/os-student/chap10.txt` is an example of
A) a relative path name
B) an absolute path name
C) a relative path name to the current directory of `/home`
D) an invalid path name
Ans: B

14. The UNIX file system uses which of the following consistency semantics?
A) Writes to an open file by a user are not visible immediately to other users that have the file open at the same time.
B) Once a file is closed, the changes made to it are visible only in sessions starting later.
C) Users are not allowed share the pointer of current location into the file.
D) Writes to an open file by a user are visible immediately to other users that have the file open at the same time.
Ans: D

15. Which of the following is a key property of an immutable file?
A) The file name may not be reused.
B) The contents of the file may be altered.
C) It is difficult to implement in a distributed system.
D) The file name may be reused.
Ans: A

16. Which of the following is not considered a classification of users in connection with each file?
A) owner
B) current user
C) group
D) universe
Ans: B

17. _____ is a secure, distributed naming mechanism.
A) Lightweight directory-access protocol (LDAP)
B) Domain name system (DNS)
C) Common internet file system (CIFS)
D) Network information service (NIS)
Ans: A

18. `app.exe` is an example of a(n) _____.
A) batch file
B) object file
C) executable file
D) text file
Ans: C

19. A mount point is _____.
A) a root of the file system
B) a location of a shared file system
C) only appropriate for shared file systems
D) the location within the file structure where the file system is to be attached.
Ans: D

20. _____ is/are not considered a difficulty when considering file sharing.
A) Reliability
B) Multiple users
C) Consistency semantics
D) Remote access
Ans: A

21. Which of the following is not considered a file attribute?
A) Name B) Size
C) Resolution D) Protection
Ans: C

22. The path name `os-student/src/vm.c` is an example of
A) a relative path name
B) an absolute path name
C) a relative path name to the current directory of `/os-student`
D) an invalid path name
Ans: A

23. Which of the following statements regarding the client-server model is true?
A) A remote file system may be mounted.
B) The client-server relationship is not very common with networked machines.
C) A client may only use a single server.
D) The client and server agree on which resources will be made available by servers.
Ans: A

## Short Answer

24. If you were creating an operating system to handle files, what would be the six basic file operations that you should implement?
Ans: The six basic file operations include: creating a file, writing a file, reading a file, repositioning within a file, deleting a file, and truncating a file. These operations comprise the minimal set of required file operations.

25. What are common attributes that an operating system keeps track of and associates with a file?
Ans: The attributes of the file are: 1) the name—the human-readable name of the file, 2) the identifier—the non-human-readable tag of the file, 3) the type of the file, 4) the location of the file, 5) the file's size (in bytes, words, or blocks), and possibly the maximum allowed size, 6) file protection through access control information, and 7) time, date, and user identification.

26. Distinguish between an absolute path name and a relative path name.
Ans: An absolute path name begins at the root and follows a path of directories down to the specified file, giving the directory names on the path. An example of an absolute path name is `/home/osc/chap10/file.txt`. A relative path name defines a path from the current directory. If the current directory is `/home/osc/`, then the relative path name of `chap10/file.txt` refers to the same file as in the example of the absolute path name.

27. What is the difference between an operating system that implements mandatory locking and one that implements advisory file locking?
Ans: Mandatory locking requires that the operating system not allow access to any file that is locked, until it is released, even if the program does not explicitly ask for a lock on the file. An advisory file locking scheme will not prevent access to a locked file, and it is up to the programmer to ensure that locks are appropriately acquired and released.

28. What are the advantages of using file extensions?
Ans: File extensions allow the user of the computer system to quickly know the type of a file by looking at the file's extension. The operating system can use the extension to determine how to handle a particular file.

29. Briefly explain the functionality of extended file attributes.
Ans: File attributes are general values representing the name of a file, its owner, size, and permissions (to name a few.) Extended file attributes refer to additional file attributes such as character encoding, security features, and application associated with opening the file.

30. Why do all file systems suffer from internal fragmentation?
Ans: Disk space is always allocated in fixed sized blocks. Whenever a file is written to disk, it usually does not fit exactly within an integer number of blocks so that a portion of a block is wasted when storing the file onto the device.

31. Describe three common methods for remote file-sharing.
Ans: The first implemented method involves manually transferring files between machines via programs like ftp. The second major method uses a distributed file system (DFS), in which remote directories are visible from a local machine. In the third method, a browser is needed to access remote files on the World Wide Web, and separate operations (essentially a wrapper for ftp) are used to transfer files. The DFS method involves a much tighter integration between the machine that is accessing the remote files and the machine providing the files.

32. Describe how the UNIX network file system (NFS) recovers from server failure in a remote file system?
Ans: In the situation where the server crashes but must recognize that it has remotely mounted exported file systems and opened files, NFS takes a simple approach, implementing a stateless DFS. In essence, it assumes that a client request for a file read or write would not have occurred unless the file system had been remotely mounted and the file had been previously open. The NFS protocol carries all the information needed to locate the appropriate file and perform the requested operation, assuming that the request was legitimate.

33. What are the advantages and disadvantages of access control lists?
Ans: Access control lists have the advantage of enabling complex access methodologies. The main problem with ACLs is their length. Constructing the list may be a tedious task. Space management also becomes more complicated because the directory size needs to be of variable size.

## True/False

34. Windows systems employ mandatory locking. **True**

35. As a general rule, UNIX systems employ mandatory locks. **False**

36. All files in a single-level directory must have unique names. **True**

37. A relative path name begins at the root. **False**

38. An absolute path name must always begin at the root. **True**

39. Typically, a mount point is an empty directory. **True**

40. Windows does not provide access-control lists. **False**

41. The most common approach to file protection is to make access dependent upon the identity of the user. **True**

42. On a UNIX system, writes to an open file are not immediately visible to other users who also have the same file open. **False**

43. A file on a Solaris system with permissions `-rwx--x--x+` is an example of both access-control lists as well as owner/group/universe protection. **True**

44. File system links may be to either absolute or relative path names. **True**

45. A relative block number is an index relative to the beginning of a file. **True**

46. Processes do not have a concept of a current directory. **False**

47. An absolute path name cannot be a relative path name. **False**

# Chapter 11

## Multiple Choice

1. Transfers between memory and disk are performed a ____.
A) byte at a time
B) file at a time
C) block at a time
D) sector at a time
Ans: C

2. Order the following file system layers in order of lowest level to highest level.

[1] I/O control
[2] logical file system
[3] basic file system
[4] file-organization module
[5] devices
A) 1, 3, 5, 4, 2
B) 5, 1, 3, 2, 4
C) 1, 5, 3, 4, 2
D) 5, 1, 3, 4, 2
Ans: D

3. A volume control block ____.
A) can contain information needed by the system to boot an operating system from that partition
B) is a directory structure used to organize the files
C) contains many of the file's details, including file permissions, ownership, size, and location of the data blocks
D) contains information such as the number of blocks in a partition, size of the blocks, and free-block and FCB count and pointers
Ans: D

4. Which of the following is the simplest method for implementing a directory?
A) tree data structure
B) linear list
C) hash table
D) nonlinear list
Ans: B

5. In the Linux VFS architecture, a(n) ____ object represents an individual file.
A) inode
B) file
C) superblock
D) dentry
Ans: A

6. Which of the following allocation methods ensures that only one access is needed to get a disk block using direct access?
A) linked allocation
B) indexed allocation
C) hashed allocation
D) contiguous allocation
Ans: D

7. The free-space list can be implemented using a bit vector approach. Which of the following is a drawback of this technique?
A) To traverse the list, each block must be read on the disk.
B) It is not feasible to keep the entire list in main memory for large disks.
C) The technique is more complicated than most other techniques.
D) This technique is not feasible for small disks.
Ans: B

8. Page caching ____.
A) uses virtual memory techniques to cache file data as system-oriented blocks as opposed to pages
B) uses virtual memory techniques to cache file data as pages as opposed to system-oriented blocks.
C) is used in Windows NT but not in Windows 2000.
D) cannot be used to cache both process pages and file data.
Ans: B

9. NFS views a set of interconnected workstations as a set of ____.
A) independent machines with independent file systems
B) dependent machines with independent file systems
C) dependent machines with dependent file systems
D) independent machines with dependent file systems
Ans: A

10. The NFS mount protocol ____.
A) does not allow a remote directory to be accessible in a transparent manner
B) exhibits a transitivity property in terms of client access to other file systems
C) establishes the initial logical connection between a server and a client
D) provides a set of RFCs for remote file operations
Ans: C

11. A disk with free blocks 0,1,5,9,15 would be represented with what bit map?
A) 0011101110111110
B) 1100010001000001
C) 0100010001000001
D) 1100010001000000
Ans: B

12. A _____ is a view of a file system before the last update took place.
A) transaction
B) backup
C) consistency checker
D) snapshot
Ans: D

13. _____ includes all of the file system structure, minus the actual contents of files.
A) Metadata
B) Logical file system
C) Basic file system
D) File-organization module
Ans: A

14. The file-allocation table (FAT) used in MS-DOS is an example of _____.
A) contiguous allocation
B) indexed allocation
C) linked allocation
D) multilevel index
Ans: C

15. How many disk accesses are necessary for direct access to byte 20680 using linked allocation and assuming each disk block is 4 KB in size?
A) 1
B) 6
C) 7
D) 5
Ans: B

16. A contiguous chunk of disk blocks is known as a(n) _____.
A) extent
B) disk block group
C) inode
D) file-allocation table (FAT)
Ans: A

17. On UNIX systems, the data structure for maintaining information about a file is a(n) _____.
A) superblock
B) inode
C) file-control block (FCB)
D) master file table
Ans: B

18. Which algorithm is considered reasonable for managing a buffer cache?
A) least-recently-used (LRU)
B) first-in-first-out (FIFO)
C) most-recently-used
D) least-frequently-used (LFU)
Ans: A

19. Which of the following statements regarding the WAFL file system is incorrect?
A) Clones are similar to snapshots.
B) WAFL is used exclusively on networked file servers.
C) Part of caching uses non-volatile RAM (NVRAM.)
D) It provides little replication.
Ans: D

20. Consider a system crash on a log-structured file system. Which one of the following events must occur?
A) Only aborted transactions must be completed.
B) All transactions in the log must be completed.
C) All transactions in the log must be marked as invalid.
D) File consistency checking must be performed.
Ans: B

21. A _____ contains the same pages for memory-mapped IO as well as ordinary IO.
A) double cache
B) unified virtual memory
C) page cahce
D) unified buffer cache
Ans: D

## Short Answer

22. Briefly describe the in-memory structures that may be used to implement a file system.
Ans: An in-memory mount table contains information about each mounted volume. An in-memory directory-structure cache holds the directory information of recently accessed directories. The system-wide open-file table contains a copy of the FCB of each open file. The per-process open-file table contains a pointer to the appropriate entry in the system-wide open-file table.

23. To create a new file, an application program calls the logical file system. Describe the steps the logical file system takes to create the file.
Ans: The logical file system allocates a new FCB. Alternatively, if the file-system implementation creates all FCBs at file-system creation time, an FCB is allocated from the set of free FCBs. The system then reads the appropriate directory into memory, updates it with the new file name and FCB, and writes it back to the disk.

24. What do the terms "raw" and "cooked" mean when used to describe a partition?
Ans: A raw disk is used where no file system is appropriate. Raw partitions can be used for a UNIX swap space as it does not need a file system. On the other hand, a cooked disk is a disk that contains a file system.

25. What are the two most important functions of the Virtual File System (VFS) layer?
Ans: The VFS separates the file-system-generic operations from their implementation by defining a clean VFS interface. Several of these implementations may coexist on the same machine allowing transparent access to different types of locally mounted file systems. The other important feature of VFS is that it is based on a file-representation structure that contains a numerical designator for a network-wide unique file. This network-wide uniqueness is required for support of network file systems.

26. What is the main disadvantage to using a linear list to implement a directory structure? What steps can be taken to compensate for this problem?
Ans: Linear lists are slow to search. This slowness would be noticeable to users as directory information is used frequently in computer systems. Many operating systems implement a software cache to store the most recently used directory information. A sorted list may also be used to decrease the average search time due to a binary search.

27. How is a hash table superior to a simple linear list structure? What issue must be handled by a hash table implementation?
Ans: A hash table implementation uses a linear list to store directory entries. However, a hash data structure is also used in order to speed up the search process. The hash data structure allows the file name to be used to help compute the file's location within the linear list. Collisions, which occur when multiple files map to the same location, must be handled by this implementation.

28. What are the problems associated with linked allocation of disk space routines?
Ans: The major problem is that a linked allocation can be used effectively only for sequential-access files. Another disadvantage is the space required for the pointers. Yet another problem of linked allocation is the decreased reliability due to lost or damaged pointers.

29. Describe the counting approach to free space management.
Ans: The counting approach takes advantage of the fact that, generally, several contiguous blocks may be allocated or freed simultaneously. Thus, rather than keeping a list of *n* free disk addresses, we can keep the address of the first free block and the number *n* of free contiguous blocks that follow the first block. Each entry in the free-space list then consists of a disk address and a count.

30. Explain how a snapshot is taken in the WAFL file system.
Ans: To take a snapshot, WAFL creates a duplicate root inode. Any file or metadata updates after that go to new blocks rather than overwriting their existing blocks. The new root inode points to metadata and data changed as a result of these writes, while the old root inode still points to the old blocks, which have not been updated.

31. Explain the benefit if using a unified buffer cache.
Ans: Without a unified buffer cache, memory-mapped IO uses a page cache, and ordinary IO uses a buffer cache. The buffer cache will also cache the same contents as in the page cache. This is known as double caching of file system data twice. A unified buffer cache uses the same, single buffer cache for caching pages for both memory-mapped IO as well as ordinary IO.

## True/False

32. Metadata includes all of the file-system structure, including the actual data (or contents of the file). **False**

33. In NTFS, the volume control block (per volume) and the directory structure (per file system) is stored in the master file table. **True**

34. Indexed allocation may require substantial overhead for its index block. **True**

35. The NFS protocol provides concurrency-control mechanisms. **False**

36. On log-structured file systems, all metadata and file data updates are written sequentially to a log. **False**

37. VFS allows dissimilar file systems to be accessed similarly. **True**

38. Linked allocation suffers from external fragmentation. **False**

39. The WAFL file system can be used in conjunction with NFS. **True**

40. On log-structured file systems, a transaction is considered only when it is written to disk. **False**

41. A unified buffer cache uses the same cache for ordinary disk I/O as well as memory-mapped I/O. **True**

42. A consistency checker only checks for inconsistencies, it cannot fix any that it may find. **False**

43. Asynchronous writes to a file system are generally more efficient than synchronous writes. **True**

# Chapter 12

## Multiple Choice

1. The surface of a magnetic disk platter is divided into ____.
A) sectors B) arms
C) tracks D) cylinders
Ans: C

2. On media that uses constant linear velocity, the ____.
A) disk's rotation speed increases as the head moves towards the middle of the disk from either side
B) disk's rotation speed remains constant
C) density of bits decreases from the inner tracks to the outer tracks
D) density of bits per track is uniform
Ans: D

3. The SSTF scheduling algorithm ____.
A) services the request with the maximum seek time
B) services the request with the minimum seek time
C) chooses to service the request furthest from the current head position
D) None of the above
Ans: B

4. Consider a disk queue holding requests to the following cylinders in the listed order: 116, 22, 3, 11, 75, 185, 100, 87. Using the SCAN scheduling algorithm, what is the order that the requests are serviced, assuming the disk head is at cylinder 88 and moving upward through the cylinders?
A) 116 - 22 - 3 - 11 - 75 - 185 - 100 - 87
B) 100 - 116 - 185 - 87 - 75 - 22 - 11 - 3
C) 87 - 75 - 100 - 116 - 185 - 22 - 11 - 3
D) 100 - 116 - 185 - 3 - 11 - 22 - 75 - 87
Ans: B

5. Consider a disk queue holding requests to the following cylinders in the listed order: 116, 22, 3, 11, 75, 185, 100, 87. Using the FCFS scheduling algorithm, what is the order that the requests are serviced, assuming the disk head is at cylinder 88 and moving upward through the cylinders?
A) 116 - 22 - 3 - 11 - 75 - 185 - 100 - 87
B) 100 - 116 - 185 - 87 - 75 - 22 - 11 - 3
C) 87 - 75 - 100 - 116 - 185 - 22 - 11 - 3
D) 100 - 116 - 185 - 3 - 11 - 22 - 75 – 87
Ans: A

6. Consider a disk queue holding requests to the following cylinders in the listed order: 116, 22, 3, 11, 75, 185, 100, 87. Using the SSTF scheduling algorithm, what is the order that the requests are serviced, assuming the disk head is at cylinder 88 and moving upward through the cylinders?
A) 116 - 22 - 3 - 11 - 75 - 185 - 100 - 87
B) 100 - 116 - 185 - 87 - 75 - 22 - 11 - 3
C) 87 - 75 - 100 - 116 - 185 - 22 - 11 - 3
D) 100 - 116 - 185 - 3 - 11 - 22 - 75 - 87
Ans: C

7. Consider a disk queue holding requests to the following cylinders in the listed order: 116, 22, 3, 11, 75, 185, 100, 87. Using the C-SCAN scheduling algorithm, what is the order that the requests are serviced, assuming the disk head is at cylinder 88 and moving upward through the cylinders?
A) 116 - 22 - 3 - 11 - 75 - 185 - 100 - 87
B) 100 - 116 - 185 - 87 - 75 - 22 - 11 - 3
C) 87 - 75 - 100 - 116 - 185 - 22 - 11 - 3
D) 100 - 116 - 185 - 3 - 11 - 22 - 75 - 87
Ans: D

8. Low-level formatting ____.
A) does not usually provide an error-correcting code
B) is usually performed by the purchaser of the disk device
C) is different from physical formatting
D) divides a disk into sections that the disk controller can read and write
Ans: D

9. Host-attached storage is ____.
A) a special purpose storage system that is accessed remotely over a data network
B) not suitable for hard disks
C) accessed via local I/O ports
D) not suitable for use in raid arrays
Ans: C

10. Swap space management ____.
A) is a high-level operating system task
B) tries to provide the best throughput for the virtual memory system
C) is primarily used to increase the reliability of data in a system
D) None of the above
Ans: B

11 A RAID structure ____.
A) is primarily used for security reasons
B) is primarily used to ensure higher data reliability
C) stands for redundant arrays of inexpensive disks
D) is primarily used to decrease the dependence on disk drives
Ans: B

12. RAID level ____ is the most common parity RAID system.
A) 0 B) 0+1
C) 4 D) 5
Ans: D

13. Which of the following disk head scheduling algorithms does not take into account the current position of the disk head?
A) FCFS B) SSTF
C) SCAN D) LOOK
Ans: A

14. The location where Windows places its boot code is the _____.
A) boot block
B) master boot record (MBR)
C) boot partition
D) boot disk
Ans: B

15. What are the two components of positioning time?
A) seek time + rotational latency
B) transfer time + transfer rate
C) effective transfer rate - transfer rate
D) cylinder positioning time + disk arm positioning time
Ans: A

16. Which of the following statements is false?
A) Swapping works in conjunction with virtual memory techniques.
B) Some systems allow for multiple swap spaces (disks).
C) Solaris only swaps pages of anonymous memory.
D) Typically, entire processes are swapped into memory.
Ans: D

17. _____ is a technique for managing bad blocks that maps a bad sector to a spare sector.
A) Sector slipping
B) Sector sparing
C) Bad block mapping
D) Hard error management
Ans: B
18. Which RAID level is best for storing large volumes of data?
A) RAID levels 0 + 1 and 1 + 0
B) RAID level 3
C) RAID level 4
D) RAID level 5
Ans: D

19. A _____ is a private network connecting servers and storage units.
A) host-attached storage
B) network-attached storage
C) storage-area network
D) private-area network
Ans: C

20. Which of the following statements regarding solid state disks (SSDs) is false?
A) They generally consume more power than traditional hard disks.
B) They have the same characteristics as magnetic hard disks, but can be more reliable.
C) They are generally more expensive per megabyte than traditional hard disks.
D) They have no seek time or latency.
Ans: A

21. Solid state disks (SSDs) commonly use the _____ disk scheduling policy.
A) SSTF
B) SCAN
C) FCFS
D) LOOK
Ans: C

## Short Answer

22. What is constant angular velocity in relation to disk drives?
Ans: If the rotation speed of a disk is to remain constant, the density of the bits must be changed for different tracks to ensure the same rate of data moving under the head. This method keeps a constant angular velocity on the disk.

23. What is a storage-area network?
Ans: A storage-area network (SAN) is a private network (using storage protocols rather than networking protocols) connecting servers and storage units. The power of a SAN lies in its flexibility. Multiple hosts and multiple storage arrays can attach to the same SAN, and storage can be dynamically allocated to hosts.

24. What is a disadvantage of the SSTF scheduling algorithm?
Ans: Although the SSTF algorithm is a substantial improvement over the FCFS algorithm, it is not optimal. SSTF may cause starvation of some requests. If a continual stream of requests arrives near one another, a request of a cylinder far away from the head position has to wait indefinitely.

25. What is the advantage of LOOK over SCAN disk head scheduling?
Ans: The LOOK algorithm is a type of SCAN algorithm. The difference is that, instead of forcing the disk head to fully traverse the disk, as is done in the SCAN algorithm, the disk head moves only as far as the final request in each direction.

26. What are the factors influencing the selection of a disk-scheduling algorithm?
Ans: Performance of a scheduling algorithm depends heavily on the number and types of requests. Requests for disk service can be greatly influenced by the file-allocation method. The location of directories and index blocks is also important. Other considerations for scheduling may involve rotational latency (instead of simply seek distances) and operating system constraints, such as demand paging.

27. Describe one technique that can enable multiple disks to be used to improve data transfer rate.
Ans: One technique is bit-level striping. Bit-level striping consists of splitting the bits of each byte across multiple disks so that the data can be accessed from multiple disks in parallel. Another method is block-level striping where blocks of a file are striped across multiple disks.

28. Describe an approach for managing bad blocks.
Ans: One approach to managing bad blocks is sector sparing. When the disk controller detects a bad sector, it reports it to the operating system. The operating system will then replace the bad sector with a spare sector. Whenever the bad sector is requested, the operating system will translate the request to the spare sector.

29. Describe why Solaris systems only allocate swap space when a page is forced out of main memory, rather than when the virtual memory page is first created.
Ans: Solaris systems only allocate swap space when a page is force out of main memory, because modern computers typically have much more physical memory than older systems and—as a result—page less frequently. A second reason is that Solaris only swaps anonymous pages of memory.

30. Describe how ZFS uses checksums to maintain the integrity of data.
Ans: ZFS maintains checksums of all data and metadata blocks. When the file system detects a bad checksum for a block, it replaces the bad block with a mirrored block that has a valid checksum.

## True/False

31. Disk controllers do not usually have a built-in cache. **False**

32. In Solaris, swap space is only used as a backing store for pages of anonymous memory. **True**

33. In asynchronous replication, each block is written locally and remotely before the write is considered complete. **False**

34. Solid state disks (SSDs) commonly use the FCFS disk scheduling algorithm. **True**

35. In most RAID implementations, a hot spare disk is not used for data, but is configured for replacement should any other disk fail. **True**

36. LOOK disk head scheduling offers no practical benefit over SCAN disk head scheduling. **False**

37. Windows allows a hard disk to be divided into one or more partitions. **True**

38. RAID level 0 provides no redundancy. **True**

39. Data striping provides reliability for RAID systems. **False**

40. In general, LOOK disk head scheduling will involve less movement of the disk heads than SCAN disk head scheduling. **True**

# Chapter 13

## Multiple Choice

1. The ____ register of an I/O port can be written by the host to start a command or to change the mode of a device.
A) status
B) control
C) data-in
D) transfer
Ans: B

2. An interrupt priority scheme can be used to ____.
A) allow the most urgent work to be finished first
B) make it possible for high-priority interrupts to preempt the execution of a low priority interrupt
C) defer the handling of low-priority interrupt without masking off all interrupts
D) All of the above
Ans: D

3. DMA controllers ____.
A) do not utilize an additional, special purpose, processor
B) are a nonstandard component in PCs of today
C) can steal memory access cycles from the main CPU
D) can access main memory at the same time as the main CPU
Ans: C

4. A character-stream device ____.
A) transfers data in blocks of bytes
B) transfers data a byte at a time
C) is a device such as a disk drive
D) is similar to a random access device
Ans: B

5. ____ I/O accesses a block device as a simple array of blocks.
A) Raw
B) Stream
C) Indirect
D) Cooked
Ans: A

6. Which of the following is true of a blocking system call?
A) The application continues to execute its code when the call is issued.
B) The call returns immediately without waiting for the I/O to complete.
C) The execution of the application is suspended when the call is issued.
D) Blocking application code is harder to understand than nonblocking application code
Ans: C

7. A(n) ____ is a buffer that holds output for a device that cannot accept interleaved data streams.
A) escape B) block device
C) cache D) spool
Ans: D

8. A sense key reports on the failure of a SCSI device by ____.
A) stating the general category of failure
B) stating the general nature of the failure
C) giving detailed information about the exact cause of failure
D) maintaining internal pages of error-log information
Ans: B

9. A(n) ____ is a front-end processor that multiplexes the traffic from hundreds of remote terminals into one port on a large computer.
A) terminal concentrator
B) network daemon
C) I/O channel
D) context switch coordinator
Ans: A

10. Which of the following is a principle that can improve the efficiency of I/O?
A) Increase the number of context switches.
B) Use small data transfers
C) Move processing primitives into hardware
D) Decrease concurrency using DMA controllers
Ans: C

## Short Answer

11. Explain the concept of a bus and daisy chain. Indicate how they are related.
Ans: A bus is merely a set of wires and a rigidly defined protocol that specifies a set of messages that can be sent on the wires. The messages are conveyed by patterns of electrical voltages applied to the wires with defined timings. A daisy chain is a device configuration where one device has a cable that connects another device which has a cable that connects another device, and so on. A daisy chain usually operates as a bus.

12. Explain the difference between a serial-port controller and a SCSI bus controller.
Ans: A serial-port controller is a simple device controller with a single chip (or portion of a chip) that controls the signals on the wires of a serial port. By contrast, a SCSI bus controller is not simple. Because the SCSI protocol is complex, the SCSI bus controller is often implemented as a separate circuit board that plugs into the computer.

13. Explain the concept of polling between a host and a controller.
Ans: When a host tries to access the controller, it constantly reads the status of a "busy register" and waits for the register to clear. This repetitive checking is termed polling.

14. What is interrupt chaining?
Ans: Interrupt chaining is a technique in which each element in the interrupt vector points to the head of a list of interrupt handlers. When an interrupt is raised, the handlers on the corresponding list are called one by one, until one is found that can service the request. This is a compromise between the overhead of a huge interrupt table and the inefficiency of dispatching to a single interrupt handler.

15. Why is DMA used for devices that execute large transfers?
Ans: Without DMA, programmed I/O must be used. This involves using the CPU to watch status bits and feed data into a controller register one byte at a time. Therefore, DMA was developed to lessen the burden on the CPU. DMA uses a special-purpose processor called a DMA controller and copies data in chunks.

16. What is the purpose of a programmable interval timer?
Ans: The programmable interval timer is hardware used to measure elapsed time and to trigger operations. The scheduler uses this mechanism to generate an interrupt that will preempt a process at the end of its time slice.

17. Give an example of when an application may need a nonblocking I/O system call.
Ans: If the user is viewing a web browser, then the application should allow keyboard and mouse input while it is displaying information to the screen. If nonblocking is not used, then the user would have to wait for the application to finish displaying the information on the screen before allowing any kind of user interaction.

18. What are the three reasons that buffering is performed?
Ans: A buffer is a memory area that stores data while they are transferred between two devices or between a device and an application. One reason for buffering is handle data when speed mismatches between the producer and consumer of a data stream exist. The second reason is to adapt between devices that have different data-transfer sizes. The third reason is to support copy semantics for application I/O.

19. What is the purpose of a UNIX mount table?
Ans: The UNIX mount table associates prefixes of path names with specific device names. To resolve a path name, UNIX looks up the name in the mount table to find the longest matching prefix; the corresponding entry gives the device name.

20. UNIX System V implements a mechanism called STREAMS. What is this mechanism?
Ans: STREAMS enables an application to assemble pipelines of driver code dynamically. A stream is a full-duplex connection between a device driver and a user-level process. It consists of a stream head that interfaces with the user process and a driver end that controls the device. It may also include stream modules between them.

## True/False

21. An expansion bus is used to connect relatively high speed devices to the main bus. **False**

22. A maskable interrupt can never be disabled. **False**

23. A dedicated device cannot be used concurrently by several processes or threads. **True**

24. Although caching and buffering are distinct functions, sometimes a region of memory can be used for both purposes. **True**

25. STREAMS I/O is asynchronous except when the user process communicates with the stream head. **True**

26. Vectored IO allows one system call to perform multiple IO operations involving involving a single location. **False**

# Chapter 14

## Multiple Choice

1. In the UNIX operating system, a domain is associated with the ____.
A) user B) process
C) procedure D) task
Ans: A

2. In MULTICS, the protection domains are organized in a _____.
A) star structure
B) linear structure
C) ring structure
D) directory structure
Ans: C

3. In an access matrix, the ____ right allows a process to change the entries in a row.
A) owner
B) copy
C) control.
D) switch
Ans: C

4. The ____ implementation of an access table consists of sets of ordered triples.
A) global table
B) access list for objects
C) lock-key mechanism
D) capability list
Ans: A

5. In capability lists, each object has a ____ to denote its type.
A) gate B) tag
C) key D) lock
Ans: B

6. Which of the following implementations of the access matrix is a compromise between two other implementations listed below?
A) access list
B) capability list
C) global table
D) lock-key
Ans: D

7. In the reacquisition scheme for implementing the revocation of capabilities, ____.
A) a key is defined when the capability is created
B) the capabilities point indirectly, not directly, to the objects
C) a list of pointers is maintained with each object that point to all capabilities associated with that object
D) capabilities are periodically deleted from each domain
Ans: D

8. Which of the following is an advantage of compiler-based enforcement of access control?
A) Protection schemes are programmed as opposed to simply declared.
B) Protection requirements are dependant of the facilities provided by a particular operating system.
C) The means for enforcement needs to be provided by the designer of the subsystem.
D) Access privileges are closely related to the linguistic concept of a data type.
Ans: D

9. Which of the following is a true statement regarding the relative merits between access rights enforcement based solely on a kernel as opposed to enforcement provided largely by a compiler?
A) Enforcement by the compiler provides a greater degree of security.
B) Enforcement by the kernel is less flexible than enforcement by the programming language for user-defined policy.
C) Kernel-based enforcement has the advantage that static access enforcement can be verified off-line at compile time.
D) The fixed overhead of kernel calls cannot often be avoided in a compiler-based enforcement.
Ans: B

10. Which of the following is true of the Java programming language in relation to protection?
A) When a class is loaded, the JVM assigns the class to a protection domain that gives the permissions of that class.
B) It does not support the dynamic loading of untrusted classes over a network.
C) It does not support the execution of mutually distrusting classes within the same JVM.
D) Methods in the calling sequence are not responsible for requests to access a protected resource.
Ans: A

## Short Answer

11. Explain the meaning of the term object as it relates to protection in a computer system. What are the two general types of objects in a system?
Ans: A computer system is a collection of processes and objects. Each object has a unique name that differentiates it from all other objects in the system, and each can be accessed only through well-defined and meaningful operations. Objects are essentially abstract data types and include hardware objects (such as the CPU, memory segments, printer, and disks) and software objects (such as files, programs, and semaphores).

12. A process is said to operate within a protection domain which specifies the resources that the process may access. List the ways that a domain can be realized.
Ans: A domain may be realized where each user, process, or procedure may be a domain. In the first case, the set of objects that can be accessed depends on the identity of the user. In the second case, the set of objects that can be accessed depends upon the identity of the process. Finally, the third case specifies that the set of objects that can be accessed depends on the local variables defined with the procedure.

13. What is an access matrix and how can it be implemented?
Ans: An access matrix is an abstract model of protection where the rows represent domains and the columns represent objects. Each entry in the matrix consists of a set of access rights. Access matrices are typically implemented using a global table, an access list for objects, a capability list for domains, or a lock-key mechanism.

14. What was the main disadvantage to the structure used to organize protection domains in the MULTICS system?
Ans: The ring structure had the disadvantage in that it did not allow the enforcement of a need-to-know principle. For example, if an object needed to be accessible in one domain, but not in another, then the domain that required the privileged information needed to be located such that it was in a ring closer to the center than the other domain. This also forced every object in the outer domain to be accessible by the inner domain which is not necessarily desired.

15. Why is a global table implementation of an access matrix not typically implemented?
Ans: The global table implementation suffers from a couple of drawbacks that keep it from being a popular implementation type. The first drawback is that the table is usually large and cannot be stored in main memory. If the table cannot be stored in main memory, extra I/O must be used to access this table. In addition, a global table makes it difficult to take advantage of special groupings of objects or domains.

16. How does the lock-key mechanism for implementation of an access matrix work?
Ans: In a lock-key mechanism, each object is given a list of unique bit patterns, called locks. Similarly, each domain has a list of unique bit patterns, called keys. A process in a domain can only access an object if that domain has the matching key for the lock. Users are not allowed to examine or modify the list of keys (or locks) directly.

17. What is a confinement problem?
Ans: A confinement problem is the problem of guaranteeing that no information initially held in an object can migrate outside of its execution environment. Although copy and owner rights provide a mechanism to limit the propagation of access rights, they do not provide appropriate tools for preventing the propagation (or disclosure) of information. The confinement problem is in general unsolvable.

18. What is rights amplification with respect to the Hydra protection system?
Ans: Rights amplification allows certification of a procedure as trustworthy to act on a formal parameter of a specified type on behalf of any process that holds a right to execute the procedure. The rights held by the trustworthy procedure are independent of, and may exceed, the rights held by the calling process.

19. Describe the two kinds of capabilities in CAP.
Ans: Data capabilities only provide the standard read, write, and execute operations of the individual storage segments associated with the object. Data capabilities are interpreted by the microcode in the CAP machine. Software capabilities are protected, but not interpreted by the CAP microcode. These capabilities are interpreted by a protected procedure which may be written by an application programmer as part of a subsystem.

20. Explain how Java provides protection through type safety.
Ans: Java's load-time and run-time checks enforce type safety of Java classes. Type safety ensures that classes cannot treat integers as pointers, write past the end of an array, or otherwise access memory in arbitrary ways. Rather, a program can access an object only via the methods defined on that object by its class. This enables a class to effectively encapsulate its data and methods from other classes loaded in the same JVM.

## True/False

21. Domains may share access rights. **True**

22. An access matrix is generally dense. **False**

23. A capability list associated with a domain is directly accessible to a process executing in that domain. **False**

24. Most systems use a combination of access lists and capabilities. **True**

25. The "key" scheme for implementing revocation allows selective revocation. **False**

# Chapter 15

## Multiple Choice

1. The most common method used by attackers to breach security is ____.
A) masquerading
B) message modification
C) session hijacking
D) phishing
Ans: A

2. A code segment that misuses its environment is called ____.
A) a backdoor
B) a trap door
C) a worm
D) a Trojan horse
Ans: D

3. Worms ____.
A) use the spawn mechanism to ravage system performance
B) can shut down an entire network
C) continue to grow as the Internet expands
D) All of the above
Ans: D

4. A denial of service attack is ____.
A) aimed at gaining information
B) aimed at stealing resources
C) aimed at disrupting legitimate use of a system
D) generally not network based
Ans: C

5. In a paired-password system, ____.
A) the user specifies two passwords
B) the computer supplies one part of a password and the user enters the other part
C) passwords must contain equal amounts of numbers and digits paired together
D) two users must enter their own separate password to gain access to the system
Ans: B

6. A ____ virus changes each time it is installed to avoid detection by antivirus software.
A) polymorphic
B) tunneling
C) multipartite
D) stealth
Ans: A

7. ____ is a symmetric stream cipher.
A) DES
B) AES
C) RC4
D) twofish
Ans: C

8. A ____ is a public key digitally signed by a trusted party.
A) key ring
B) digital certificate
C) message digest
D) digital key
Ans: B

9. ____ layer security generally has been standardized on IPSec.
A) Network
B) Transport
C) Data-link
D) Application
Ans: A

10. Which of the following is true of SSL?
A) It provides security at the data-link layer.
B) It is a simple protocol with limited options.
C) It is commonly used for secure communication on the Internet.
D) It was designed by Microsoft.
Ans: C

## Short Answer

11. What are the four levels of security measures that are necessary for system protection?
Ans: To protect a system, security measures must take places at four levels: physical (machine rooms, terminals, and workstations); human (user authorization, avoidance of social engineering); operating system (protection against accidental and purposeful security breaches); and network (leased, Internet, and wireless connections).

12. What is a trap door? Why is it problematic?
Ans: A trap door is an intentional hole left in software by the designer of a program or system. It can allow circumvention of security features for those who know about the hole. Trap doors pose a difficult problem because, to detect them, we have to analyze all the source code for all components of a system.

13. How does a virus differ from a worm?
Ans: A worm is structured as a complete, standalone program whereas a virus is a fragment of code embedded in a legitimate program.

14. What is the most common way for an attacker outside of the system to gain unauthorized access to the target system?
Ans: The stack- or buffer-overflow attack is the most common way for an attacker outside the system to gain unauthorized access to a system. This attack exploits a bug in the software in order to overflow some portion of the program and cause the execution of unauthorized code.

15. What are the two main methods used for intrusion detection?
Ans: The two most common methods employed are signature-based detection and anomaly detection. In signature-based detection, system input or network traffic is examined for specific behavior patterns known to indicate attacks. In anomaly detection, one attempts, through various techniques, to detect anomalous behavior within computer systems.

16. What is port scanning and how is it typically launched?
Ans: Port scanning is not an attack but rather is a means for a cracker to detect a system's vulnerabilities to attack. Port scanning typically is automated, involving a tool that attempts to create a TCP/IP connection to a specific port or a range of ports. Because port scans are detectable, they are frequently launched from zombie systems.

17. What role do keys play in modern cryptography?
Ans: Modern cryptography is based on secrets called keys that are selectively distributed to computers in a network and used to process messages. Cryptography enables a recipient of a message to verify that the message was created by some computer possessing a certain key - the key is the source of the message. Similarly, a sender can encode its message so that only a computer with a certain key can decode the message, so that the key becomes the destination.

18. What is the difference between symmetric and asymetric encryption?
Ans: In a symmetric encryption algorithm, the same key is used to encrypt and to decrypt. In an asymetric encryption algorithm, there are different encryption and decryption keys. Asymmetric encryption is based on mathematical functions instead of the transformations used in symmetric encryption, making it much more computationally expensive to execute.

19. What are the two main varieties of authentication algorithms?
Ans: The first type of authentication algorithm, a message-authentication code (MAC), uses symmetric encryption. In MAC, a cryptographic checksum is generated from the message using a secret key. The second type of authentication algorithm, a digital-signature algorithm, uses a public and private key. The authenticators thus produced are called digital signatures.

20. What is the practice of safe computing? Give two examples.
Ans: The best practice against computer viruses is prevention, or the practice of safe computing. Purchasing unopened software from vendors and avoiding free or pirated copies from public sources or disk exchange offer the safest route to preventing infection. Another defense is to avoid opening any e-mail attachments from unknown users.

## True/False

21. It is easier to protect against malicious misuse than against accidental misuse. **False**

22. Spyware is not considered a crime in most countries. **True**

23. Biometric devices are currently too large and expensive to be used for normal computer authentication. **True**

24. Tripwire can distinguish between an authorized and an unauthorized change. **False**

25. Generally, it is impossible to prevent denial-of-service attacks. **True**