1. *Threads* are sometimes also called
   a. process binders.
   b. lightweight processes.
   c. process stitches.

2. Each application that is currently active in a system will have
   a. at least one related process.
   b. had at least one process to get it started, but the process may have terminated when a thread took over.
   c. at least one process and two threads.

3. Consider the number of processes and processors in a typical system. The likely relationship between these is
   a. impossible to predict.
   b. that there are more processes than processors.
   c. that there must be at least as many processors as there are processes.

4. Using multiple processes in the implementation of a single application may be appropriate to take advantage of multiple processing elements in a computer system. What other reason might motivate the use of multiple processes in a single application?
   a. It is often easier to develop an algorithm for a problem if each of several tasks can be assigned to separate processes.
   b. In systems that maintain resource accounting records, using the same amount of processor time to complete some work, but dividing it among multiple processors, will be less expensive than doing all the work on a single processor.
   c. The power used by a single processor to solve a problem will likely be less than that used by multiple processors, and energy conservation is increasingly important in mobile devices.

5. A disadvantage of using multiple processes to solve a problem is that
   a. it will reduce the number of applications that can be supported by the computer system.
   b. there is additional overhead needed in the operating system to switch the processor among the various processes.
   c. it will likely require more energy to run the program that uses multiple processes.
6. Assume three processes are executed sequentially, one after another, on a system with specific sets of data. Then assume these three processes are executed at the same time with exactly the same data. Which approach to execution will require less total time?
   a. It is impossible to tell.
   b. Execution of the three processes at the same time will take longer if the processes each read data from the same disk.
   c. Since they are executing exactly the same instructions and processing the same data in each case, the total time will be exactly the same regardless of how the processes are executed.

7. A program is
   a. a static set of instructions and data for a processor.
   b. a set of instructions being executed by one or more processors in a computer system.
   c. one or more programs being executed by one or more processors.

8. A process is
   a. the data and the contents of the processor registers while the processor is executing a program
   b. one or more programs being executed by one or more processors.
   c. the activity of a processor in carrying out the actions associated with a set of instructions.

9. Which of the following is not part of a process?
   a. the processor on which the process is being executed
   b. the set of instructions being executed
   c. the quotas (if any) associated with the use of various system resources by the process

10. Processes may be created in UNIX systems by using which of the following system calls?
    a. pipe
    b. creat
    c. fork

11. Consider the hierarchy of processes in a UNIX system. What data structure best characterizes this hierarchy?
    a. a singly-linked list
    b. a doubly-linked list
    c. a general tree

12. In what position in the hierarchy of processes in a UNIX system will we find processes that have no child processes of their own?
    a. They will be leaf nodes.
    b. They will be immediately pointed to by the head pointer of an element in the hash table.
    c. They will appear at the end of a doubly-linked list.
13. Why must some components of an operating system usually be written in assembly (or assembler) language?
   a. Assembly language code is the only code that can be executed directly by a processor at boot time.
   b. High-level languages are generally designed to be independent of specific processor architecture, and thus cannot be used to write code to control the unique features of a given processor.
   c. The code generated by compilers for high-level language is just too inefficient, in general.

14. Recall that special trap instructions (like INT on Intel X86 processors and SVC on IBM mainframes) are often used to invoke system calls. For high-level languages, these instructions are
   a. usually found in libraries of functions specific to the language and the operating system being used.
   b. generated by the compiler.
   c. only executed by a special process charged with the interface between user processes and the kernel of the operating system.

15. How is possible that the fork system call can be used to create processes on so many different UNIX systems when we know that different machine language instructions are used by different processor architectures to invoke system calls?
   a. The invocation of system calls is done from application programs using scripts that are interpreted by special code in the BIOS of every system.
   b. Every processor ever built can interpret a limited subset of instructions that are used to allow the execution of a limited set of system calls, including fork.
   c. A fork function is provided for each unique system (and language), and it is this function that actually includes the specific machine language instruction(s) necessary to invoke the fork system call.

16. The typical states in which any process exists are called
   a. running, blocked, and terminated.
   b. running, not-running, and blocked.
   c. ready, running, and blocked.

17. A process in the running state
   a. is in transition between the blocked and terminated states.
   b. is actively using a processor.
   c. is also in the ready state.

18. The number of processes in the running state
   a. is never greater than the number of processors on a system.
   b. is always less than the sum of the number of processes in the blocked and ready states.
   c. is always equal to the sum of the number of processes in the blocked and ready states.
19. A process in the ready state
   a. has all resources required for execution except a processor.
   b. is actively using a processor, since it has all the resources needed for execution.
   c. has all resources needed for execution except input from a user.

20. Immediately before a process enters the running state
   a. it must be in the ready state.
   b. it must be in the waiting state.
   c. it must have been waiting for user input.

21. A process in the blocked state
   a. was moved to the blocked state directly from the ready state.
   b. is preventing another process from executing.
   c. is not capable of being executed because it does not possess one of the resources required for execution.

22. Processes in the blocked state are sometimes said to be
   a. unresponsive.
   b. dead.
   c. sleeping.

23. If a process is waiting on keyboard input from a user, then it will be in the ____ state.
   a. blocked
   b. zombie
   c. terminated

24. Which of the following is not a valid state transition for a process?
   a. blocked to ready
   b. blocked to running
   c. ready to running

25. When a processor becomes available to execute a process, the operating system's scheduler selects an appropriate process from those in
   a. the running state.
   b. the ready state.
   c. the blocked and the running states.

26. A process is moved from the running state to the blocked state
   a. when it has completed execution.
   b. when a more appropriate process becomes ready to execute, and the scheduler decides to preempt it.
   c. when it requests a resource that is not currently available, but which will become available in the future.
27. A process is moved from the blocked to the ready state
   a. when it has waited a sufficient time for a resource.
   b. when the idle process terminates.
   c. when the resource on which it has been waiting becomes available.

28. A process is moved from the running state to the ready state
   a. when the scheduler decides another process is more suitable for execution (for example, a ready process has higher priority than a running process).
   b. when it wishes to terminate.
   c. when it is ready to move to the next stage of execution.

29. A request by a process for the current time of day
   a. will not likely cause the process to block.
   b. will cause the process to block until the clock can be read.
   c. will cause a process to be moved to the ready state until the next 100-millisecond clock transition.

30. When a process has consumed its fair share of processor time, the operating system may move it to the ______ state.
   a. ready
   b. sleep
   c. running

31. When a process is moved out of the running state
   a. the processor's registers are reset to the values they had when the process entered the running state.
   b. it is always moved to the blocked state.
   c. state information (processor register contents, for example) is preserved so it can be restored when the process is next selected for execution.

32. When a process is moved into the running state
   a. the processor registers are set to the values they containing when the process was last moved out of the running state.
   b. it is also retained in the ready state so if additional processors become available the process can be run on multiple processors.
   c. all resources on which blocked processes are waiting are made available to the process being moved to the running state.

33. The act of moving one process out of the running state and replacing it with another is usually called
   a. a context switch.
   b. an event transition.
   c. an event horizon.
34. When a process is moved from the blocked state to the ready state,
   a. two context switches are required.
   b. the content of all processor registers is saved.
   c. no context switch is required.

35. Where is the unique identification of a process normally found?
   a. the process table entry for the process
   b. the process ID register in the processor
   c. the system credentials register

36. Interrupts are used to notify the processor of various events, usually those related to the
    hardware. For example, notifying the processor that a disk input/output operation has
    completed is generally sent as _____ notification.
   a. a low-priority
   b. an asynchronous
   c. a synchronous

37. The event being reported to the processor by an interrupt
   a. is always related to the last process moved to the running state.
   b. always occurs before the execution of the current instruction.
   c. may have nothing whatever to do with the process that is currently being executed.

38. The priority associated with an interrupt
   a. is always one smaller than the priority associated with the process currently being
      executed.
   b. is always one larger than the priority associated with the process currently being
      executed.
   c. is higher for faster or more important devices than for slower or less important devices.

39. If an interrupt is masked
   a. it will be discarded and never reported to the processor.
   b. it will not be reported to the processor, but it remains pending.
   c. it causes all other interrupts to be ignored until it has been recognized by the processor.

40. If interrupts are globally disabled, then
   a. some interrupts that are not locally disabled can still be reported to the processor.
   b. no processor can execute any input/output instructions.
   c. interrupts from most devices are prevented from being recognized by the processor.

41. An interrupt handler
   a. is the set of instructions executed as a result of an interrupt being reported to a processor.
   b. runs only when a process is blocked.
   c. is the user-mode code that a process may specify if it wishes to handle events such as
      keyboard-based interrupts (like control-C).
42. An interrupt handler is not allowed to
   a. block.
   b. be preempted.
   c. execute input/output instructions.

43. When an interrupt occurs, the type of interrupt is used to
   a. determine how much processor time will be allowed for the completion of the interrupt handler.
   b. determine the location of the interrupt handler.
   c. determine which region of memory will be used during the execution of the interrupt handler.

44. An interrupt vector is
   a. an array of interrupts, each of which is simultaneously generated by some special events.
   b. a one-dimensional array containing the addresses of interrupt handlers, indexed by interrupt number, and used in many processor architectures.
   c. an interrupt that results in the generation of a second interrupt, which then generates a third interrupt, and so forth, much like a disease vector.

45. A trap
   a. occurs synchronously with the execution of an instruction.
   b. is a section of code used to identify and thus prevent the execution of various Trojan-horse viruses.
   c. is a type of interrupt that can be disabled by a user process.

46. A trap can be generated for each of the following reasons except
   a. attempting to execute an invalid instruction.
   b. completing a disk operation.
   c. attempting to access an invalid memory location.

47. The process which generates a fault
   a. is the currently executing process.
   b. is the last process moved to the running state.
   c. is the process that was most recently terminated.

48. The operating system frequently translates a trap into
   a. a request to terminate the current process.
   b. a decrement of the number of such traps that were originally permitted when the currently-executing process was started.
   c. a signal to the currently-executing process.
49. When a trap or an interrupt is successfully recognized by the processor,
   a. the idle process is executed until the trap or interrupt has been processed.
   b. the processor is always placed in supervisor mode.
   c. the processor terminates execution of the current process, and begins or resumes
      execution of the process that caused the trap or interrupt.

50. A signal is
   a. a hardware report to a processor that an exceptional condition has occurred.
   b. a software report to a process that an exceptional condition has occurred.
   c. always sent to the last process that was moved to the ready state.

51. In most cases, sending a signal to a process that does not have an appropriate handler for that signal will
   a. cause the last signal handler registered for the process to handle the signal.
   b. cause the signal to be queued until such time as the process registers an appropriate signal handler.
   c. cause termination of the process.

52. Which of the following will usually cause the generation of a SIGINT signal in a UNIX system?
   a. a memory protection error
   b. pressing the control-C key
   c. pressing any key on the keyboard

53. Which of the following UNIX functions may be used to register a signal handler to deal with a floating point exception?
   a. sighandle
   b. register
   c. signal

54. Conceptually, each process has how many threads when it begins execution?
   a. a variable number, depending on how the process was created
   b. one
   c. two

55. Which of the following resources does a thread not share with other threads in the same process?
   a. the owner credentials
   b. the resource usage quotas
   c. its stack
56. A thread represents
   a. a separately scheduled unit of work for a processor.
   b. the connection between a signal and the signal handler that is registered to handle
      occurrences of that signal.
   c. none of the above

57. One advantage of threads over processes is that
   a. multiple threads share the same address space.
   b. multiple threads share the same stack.
   c. multiple threads are scheduled as a single thread.

58. If threads are implemented as part of the kernel of an operating system, then
   a. they will not be able to use trap-generating instructions to request operating system
      services (that is, they will not be able to make system calls).
   b. a context switch is required each time a processor is switched between threads, even if
      the threads belong to the same process.
   c. they will be immune to the delivery of signals.

59. If threads are implemented through user library functions, then
   a. there is much less overhead (due to context switches) required when switching from one
      user thread to another in the same process.
   b. each thread must necessarily have a separate stack, which is not required with a kernel
      thread implementation.
   c. none of the above is true.

60. The acronym IPC stands for _____.
   a. interrupt processing codes
   b. interprocess communication
   c. interprocessor communication

61. A race condition exists when
   a. two or more threads/processes are attempting to use a set of resources in a manner that
      causes each of them to wait on a resource that one of the others holds.
   b. two or more threads/processes attempt to use a resource in a manner that doesn't
      guarantee consistency,
   c. none of the above.

62. When we say an operation is atomic we mean
   a. only one processor (on a multiprocessor system) may be executing instructions while the
      atomic operation is in progress.
   b. all parts of the operation are completed without the possibility of any other operation on
      any processor manipulating the resources being used by the operation.
   c. none of the above.
63. A sequence of instructions that modifies, updates or accesses a resource shared by multiple threads or processes is called
   a. a perfect storm.
   b. a threaded section.
   c. a critical section.

64. The mutual exclusion problem
   a. deals with techniques designed to guarantee that code segments accessing the same resources are executed atomically.
   b. was first solved by Donald Knuth.
   c. requires special hardware for its solution.

65. The most obvious condition necessary for a good solution to the race condition problem is that no two processes may be
   a. simultaneously inside their critical sections.
   b. sharing any region of memory at the same time.
   c. blocked at the same time.

66. What assumptions may we make about the speed with which processes are being executed by processors, and the number of processors?
   a. All processes must be executed at the same speed.
   b. No two processes may be executed at the same speed if there is only a single processor available for their execution.
   c. No assumptions may be made.

67. Which of the following is not a necessary condition for a good solution to the mutual exclusion problem?
   a. No process should have to wait forever to enter its critical section once it has indicated a desire to do so.
   b. No process should have to limit the size of the memory region it is using.
   c. No process running outside its critical section may block other processes from entering their critical sections.

68. The ability to disable recognition of interrupts (in particular, timer interrupts) can sometimes be used to implement a solution to the critical section problem. This solution will not work if
   a. there are more than two processes.
   b. we are using threads instead of processes.
   c. there are multiple processors.
69. There are some proposed solutions to the mutual exclusion problem that utilize a lock variable. This variable is set to 0 if the shared resource is not being used, and set to 1 when the resource is in use. So if we want to use the resource, we have but to check the lock variable. If it is 1, we continually repeat the test until the lock variable becomes 0. Then we set it to 1, access the shared resource, and set the lock variable back to 0. What can go wrong with this solution?
   a. There is a race condition present. Multiple processes (or threads) could check the lock variable, find it 0, and then set it to 1 and enter their critical sections.
   b. The solution, as presented, will not work if there are more than two processes.
   c. None of the other answers is correct.

70. Solutions to the mutual exclusion problem that require a process to check a lock variable repeatedly until it has a specified value, even if implemented correctly (that is, if they prevent multiple processes from entering their critical sections at the same time), still have a significant problem. What is that problem?
   a. The process that is waiting to enter its critical section is continually consuming the processor resource, checking and checking again, significantly delaying other processes from using the processor and potentially relinquishing their use of the desired resource.
   b. The amount of memory required for a solution of this type is prohibitively large.
   c. The solution will not work correctly on systems with multiple processors.

71. By adding a turn variable to a proposed mutual exclusion solution using a lock variable we can guarantee that the critical sections of several processes will be executed atomically. Which of the following is a problem with this type of solution?
   a. Processes must enter their critical sections in strict alternation; one process will not be allowed to execute its critical section twice in a row, even if no other process is ready to execute.
   b. The solution will only work if there is at least one processor for every process.
   c. There are no problems with this solution.

72. Dekker presented the first recognized solution to the mutual exclusion problem. What hardware property is assumed by his solution?
   a. Write operations by multiple processors to the same memory location will be done atomically.
   b. Dekker's solution requires that there be only a single processor.
   c. Dekker's solution requires that there be at least two processors.

73. Peterson's solution to the mutual exclusion problem is much less cryptic than Dekker's solution. It also uses a turn variable. What else distinguishes Peterson's solution to the problem?
   a. His solution works only for systems with a single processor.
   b. His solution uses an array with one element per process to indicate which processes are interested in entering their critical sections.
   c. None of the other choices characterize Peterson's solution to the mutual exclusion problem.
74. Modern processors usually have special instructions that make implementation of solutions to the mutual exclusion problem much simpler. What is the name of an instruction found on IBM mainframe computer systems to support such operations?
   a. Signal and wait
   b. Test and set
   c. Probe and post

75. The IBM mainframe instruction used to simplify the implementation of critical sections works by doing what?
   a. It shifts 1-bits out of a special shift register into a per-processor register, eliminating the possibility that two processors could perform the same operation simultaneously.
   b. It eliminates the possibility of another process (or even another processor) examining the value of the control variable in memory between examining its value and then setting it to a known state.
   c. None of the other choices correctly describe the operation of the IBM mainframe instruction.

76. What instruction on the Intel X86 family of processors could be used to achieve the same effect as that provided by the IBM mainframe instruction used to assist in the implementation of mutual exclusion?
   a. Cache line flush (CLFUSH)
   b. Exchange (XCHG)
   c. Load task register (LTR)

77. If processes have associated priorities, then a general rule is that
   a. the priority of a process is directly related to the amount of memory it requires to execute.
   b. the highest priority ready processes are those allowed to use the processors.
   c. None of the other choices is a general rule about process priorities.

78. In some systems with priority-based processes, a problem can occur that results in the highest priority ready process being prevented from running by a lower priority process. This problem is called
   a. the priority inversion problem.
   b. the priority reversal problem.
   c. the lowest priority problem.

79. Suppose a low-priority process L holds a resource needed for the execution of a high-priority process H, but that L is prevented from running because a medium-priority process M is continually ready to run, and only a single processor is available. What is the name of an algorithm that could be used to deal with this problem?
   a. aging
   b. reverse sorting
   c. priority adjustment
80. One technique that can be used to eliminating the busy waiting used with some solutions to the mutual exclusion problem is
   a. to have a process relinquish the processor and block until another process, leaving its critical section, awakens it so it can repeat the test.
   b. to disable interrupts during the wait loop.
   c. None of the other answer choices is correct.

81. The processes involved in the producer-consumer problem are those that
   a. occasionally generate objects and those that then use those objects.
   b. are compute-bound and I/O-bound.
   c. need large amounts of memory.

82. The objects that are being produced in the producer-consumer problem are generally characterized as
   a. data objects, like indications of keys pressed or released on a keyboard.
   b. blocks of memory, independent of their content.
   c. static data items that cannot be altered by a process.

83. The processes in the producer-consumer problem are assumed to run
   a. on the same single processor.
   b. at different and arbitrary rates, as is required by good solutions to race condition problems.
   c. at about the same rate.

84. A buffer, or storage area, is required in the producer-consumer problem. This buffer holds
   a. indications of when a producer process is allowed to run.
   b. objects that a consumer process considered, but then rejected as unsuitable.
   c. objects produced before a consumer process is ready to use them.

85. Access to the buffer in the producer-consumer problem
   a. is shared by producers and consumers, and so must occur only inside a critical section.
   b. is controlled by the last object removed from the buffer.
   c. is always permitted for producers, but consumers must obtain explicit permission from a producer to access the buffer.

86. When the buffer in the producer-consumer problem is completely full
   a. the size of the buffer is increased by fifty percent.
   b. a producer that has created an object must block until an empty space becomes available in the buffer.
   c. the number of producer processes is reduced by half.
87. When the buffer in the producer-consumer problem becomes empty
   a. a dummy object is placed in the next buffer location that will be accessed by a consumer.
   b. consumer processes wishing to obtain an object must block until the producer places an object in the buffer.
   c. some consumer processes change role and become producer processes.

88. Assume a producer process is blocked while waiting on empty space in the buffer. In this case, a consumer process
   a. will run more slowly than when the producer process is not blocked.
   b. will awaken a producer when it removes an object from the buffer, allowing space for the producer to place another object.
   c. will remove and discard sufficient data objects (from the buffer) to make space available for the blocked producer to place additional objects in the buffer.

89. Consider that the statement \texttt{count = count + 1;} appears in multiple processes or threads in a system where the variable named \texttt{count} is accessible by multiple processes or threads. What must be done to ensure the value of \texttt{count} is correctly incremented when the statement is executed?
   a. Each of the processes or threads must have the same priority.
   b. It is impossible to guarantee that \texttt{count} will be properly incremented if multiple threads or processes contain the statement that can increment it.
   c. The statement must be in a critical section in each of the processes or threads that contains it.

90. Who proposed the semaphore data structure?
   a. Edgser Dijkstra
   b. Donald Shell
   c. Tony Hoare

91. What is the significance of the letters \texttt{P} and \texttt{V} when describing the fundamental operations on a semaphore?
   a. They correspond to the first letters of the French words that describe the operations.
   b. They correspond to the first letters of the Dutch words that describe the operations.
   c. They correspond to the first letters of the words Permit and Validate, which describe the actions of the fundamental operations.

92. The problem with the producer consumer problem solution using the sleep and wakeup functions is related to the fact that
   a. a wakeup may cause two processes to be awakened.
   b. wakeup and sleep are not atomic operations.
   c. a wakeup of a process (either a producer or a consumer) could be lost.
93. A semaphore has two data components. One of these is a set. What type of data objects can this set contain?
   a. process identifications
   b. processor affinities
   c. None of the other answer choices is correct.

94. A semaphore has two data components. One of these is an integer. What does the value of this integer represent?
   a. the number of waiting wakeups for processes that request use of a resource controlled by the semaphore
   b. the number of processes currently holding (possessing) one or more units of the resource controlled by the semaphore
   c. None of the other answer choices is correct.

95. A $P$ operation on a semaphore is also called a(n) _____ operation.
   a. down
   b. left
   c. right

96. A $V$ operation on a semaphore is also called a(n) _____ operation.
   a. up
   b. down
   c. index

97. When a $P$ operation is executed on a semaphore with a count of zero,
   a. the count is incremented and the process executing the $P$ operation continues execution.
   b. the identity of the process is added to the set associated with the semaphore, but the process continues execution.
   c. the process executing the $P$ operation is blocked.

98. When a $V$ operation is executed on a semaphore with a count of zero,
   a. the process executing the $V$ operation is moved from the running to the blocked state, and its identity is added to the set associated with the semaphore.
   b. if any processes are identified by the set associated with the semaphore, one of them is removed from the set and moved from the blocked to the ready state; otherwise the count is incremented by 1.
   c. the count associated with the semaphore is increased by 1.

99. Which of the semaphore operations, $P$ or $V$, must be executed atomically?
   a. Both operations must be executed atomically.
   b. $P$ must be executed atomically.
   c. Neither $P$ nor $V$ must be executed atomically.
100. In the semaphore-based solution to the producer/consumer problem, both the producer and the consumer processes use access to the shared buffer as a resource, and use it in a mutually-exclusive manner by performing down and up operations on a semaphore named mutex. What other resource do producer processes require if they are not going to be blocked?
   a. empty locations in the buffer into which a producer can place an item
   b. access to the variable containing a consumer's process identification
   c. None of the other answer choices is correct.

101. Who proposed the synchronization structure called a monitor?
   a. Niklaus Wirth and David Gries
   b. Brian Kernighan and P. J. Plauger
   c. Tony Hoare and Per Brinch Hansen

102. A monitor is different from a semaphore or an event counter in several ways. A semaphore and an event counter are each data structures, but a monitor is
   a. a programming construct intended for inclusion in a programming language.
   b. a set of processes, each of which has access to a single resource at all times.
   c. a pair of data structures, each including a semaphore.

103. It is the responsibility of the compiler that processes the programming language supporting monitors to guarantee that
   a. no processes are executing inside the monitor if there are blocked processes sleeping on any of the monitor's condition variables.
   b. at most one process or thread is sleeping on any of the monitor's condition variables at any time.
   c. only one process (or thread) is allowed to execute any of the functions or the initialization code inside the monitor at one time.

104. A condition variable used with a monitor is similar to a semaphore in that it includes a set. It is different from a semaphore, however, in what way?
   a. A condition variable includes two counts, one giving the number of resources in use, and one giving the number of processes waiting for a resource.
   b. There is no count associated with a condition variable.
   c. The count associated with a condition variable may become negative.

105. A signal operation on a condition variable associated with a monitor, with no processes waiting on the condition variable,
   a. is ignored.
   b. results in an error.
   c. None of the other answer choices is correct.
106. Brinch Hansen and Hoare each proposed a different way of dealing with a process P that signaled a condition variable and caused another process, say process Q, to be awakened. Why was it necessary to devise such strategies?
   a. The signaling process, P, is not allowed to continue execution inside the monitor while the awakened process, Q, is also executing inside the monitor.
   b. The semantics associated with the signal operation were not well known at the time it was proposed, and so additional work was required to refine the semantics.
   c. None of the other answer choices is correct.

107. Which of the following synchronization techniques can easily be used to support processes running on separate machines connected to a network?
   a. message passing
   b. event counters
   c. Each of the above techniques can be used to synchronize processes on separate machines connected to a network.

108. When a process executes the appropriate code to receive a message, what will likely happen if a message is not immediately available for it to receive?
   a. The process will block until such time as a message is available.
   b. An error will be reported and the process attempting to receive the message is terminated.
   c. None of the other answer choices is correct.

109. In the producer-consumer solution using message passing, there is some initialization required. What takes place during this one time only initialization?
   a. Each of the consumer processes is placed in the blocked state.
   b. An appropriate number of messages, each marked as being empty, are sent from the consumer to the producer.
   c. None of the other answer choices is correct.

110. Which of the following statements about the equivalence of process synchronization techniques is true?
   a. Each of the techniques - monitors, event counters, semaphores, and message passing - can be shown to be equivalent to each of the others.
   b. Monitors and event counters can be shown to be equivalent.
   c. Message passing and semaphores can be shown to be equivalent.