Chapter 4 – Threads, SMP, and Microkernels

**True / False Questions:**

1. T / F – The basic unit of dispatching in an operating system is usually referred to as a thread or lightweight process.

2. T / F – An example of a system that implements a single process with multiple threads is MS-DOS.

3. T / F – In a multithreaded environment, a process is defined as the unit of resource allocation and a unit of protection.

4. T / F – The concept of thread synchronization is required in multithreaded systems because threads of a single process share the process’s process control block (PCB).

5. T / F – In a pure User-Level Thread (ULT) facility, all of the work of thread management is done by the application, but the kernel is aware of the existence of threads.

6. T / F – In the field of distributed operating system design, the One-to-Many (Thread-to-Process) relationship is particularly interesting because it involves the concept of thread migration.

7. T / F – One disadvantage to the master/slave shared-memory multiprocessor architecture is that the failure of the master brings down the whole system.

8. T / F – In a symmetric multiprocessing (SMP) system, each processor has access only to a private main memory area.

9. T / F – An SMP O/S manages processor and other resources so that the user may view the system in the same fashion as a multiprogramming uniprocessor system.

10. T / F – The primary advantage of the basic microkernel design over layered kernel designs involves increased performance.

11. T / F – The philosophy underlying the microkernel is that only absolutely essential core operating system functions should be in the kernel.

12. T / F – The basic form of communication between processes or threads in a microkernel O/S is messages.

13. T / F – Linux makes no distinction between a process and a thread.

14. T / F – Windows 2000 is an object-oriented O/S, but only processes (not threads) are implemented as objects in the WIN2K O/S.

15. T / F – In the Solaris O/S, a User-Level Thread (ULT) in the active state is assigned to a Light-Weight Process (LWP) and executes while the underlying kernel thread executes.

**Multiple Choice Questions:**

1. The concept of a process in an operating system embodies two primary characteristics, one of which is:
   a. Multithreading
   b. Resource ownership
   c. Symmetric multiprocessing
   d. None of the above

2. An example of a system that implements a single process with multiple threads is:
   a. WIN 2000
   b. Solaris
   c. Java
   d. All of the above

3. Which of the following is true regarding the relationship between processes and threads:
   a. It takes far less time to create a new thread in an existing process than to create a new process
   b. It takes less time to terminate a process than a thread
   c. It takes less time to switch between two different processes than to switch between two threads within the same process
   d. All of the above

4. The basic thread operation related to the change in thread state that occurs when a thread needs to wait for an event is referred to as the:
   a. Unblock operation
   b. Spawn operation
   c. Block operation
   d. None of the above

5. One of the disadvantages of User-Level Threads (ULTs) compared to Kernel-Level Threads (KLTs) is:
   a. Scheduling is application specific
   b. When a ULT executes a system call, all threads in the process are blocked
   c. Thread switching does not require kernel mode privileges
   d. All of the above
6. In the Linux O/S, multiple threads may be created and executed within a single process. This is an example of the following Thread-to-Process relationship:
   a. 1:1  
   b. 1:M  
   c. M:N  
   d. None of the above

7. The computer system category where a single processor executes a single instruction stream to operate on data stored in a single memory is called:
   a. Single Instruction Single Data (SISD) stream  
   b. Single Instruction Multiple Data (SIMD) stream  
   c. Multiple Instruction Single Data (MISD) stream  
   d. None of the above

8. In a SMP system, each processor maintains a local cache and must alert all other processors that a change to cache update has taken place. This is referred to as the:
   a. Interconnection mechanism problem  
   b. Synchronization mechanism problem  
   c. Cache coherency problem  
   d. None of the above

9. Key issues involved in the design of multiprocessor operating systems include:
   a. Scheduling  
   b. Synchronization  
   c. Reliability and fault tolerance  
   d. All of the above

10. Early operating systems that were designed with little concern about structure are typically referred to as:
    a. Monolithic operating systems  
    b. Layered operating systems  
    c. Kernel operating systems  
    d. All of the above

11. A benefit of the microkernel organization is:
    a. Extensibility  
    b. Portability  
    c. Flexibility  
    d. All of the above

12. In low-level microkernel memory management, an example of an operation that can support external paging and virtual memory management is the:
    a. Grant operation  
    b. Map operation  

13. In a W2K system, the state that a thread enters when it has been unblocked and the resource for which it has been blocked is not yet available is called the:
    a. Transition state  
    b. Waiting state  
    c. Standby state  
    d. None of the above

14. In a Solaris system, a User-Level Thread (ULT) that enters the active state is assigned to a:
    a. Kernel thread  
    b. Heavy-Weight Process (HWP)  
    c. Light-Weight Process (LWP)  
    d. None of the above

15. In a Linux system, when a new process is cloned, the two processes share the same:
    a. Process identifier  
    b. Virtual memory  
    c. task_struct data structure  
    d. All of the above

Fill-In-The-Blank Questions:

1. In an operating system, the unit of dispatching is usually referred to as a thread or lightweight process while the unit of resource ownership is usually referred to as a process or task.

2. An example of an operating system that supports a single user process and a single thread is MS-DOS.

3. An example of an operating system that supports multiple user processes and multiple threads is WIN2K/Solaris/Linux/Mach/OS2.

4. It is necessary to synchronize the activities of various threads so they do not interfere with each other or corrupt data structures.

5. A process that cannot execute until some event occurs is said to be in the blocked state.

6. The Clouds O/S implements the concept of a thread as primarily an entity that can move among address spaces which represents the One-to-Many Thread-to-Process relationship.
7. In a symmetric multiprocessor system, the kernel can execute on any processor, and typically each processor does self-scheduling from the pool of available processes or threads.

8. In most modern computer systems, processors generally have at least one level of cache memory that is private to the processor.

9. With multiple active processes in an SMP system having potential access to shared address space or shared I/O resources, care must be taken to provide effective synchronization.

10. In the layered O/S architecture, functions are organized hierarchically and interaction only takes place between adjacent sections.

11. One advantage of the microkernel architecture is extensibility, allowing the addition of new services as well as the provision of multiple services in the same functional area.

12. The basic form of communication between processes or threads in a microkernel O/S is messages.

13. In a Linux system, if the process has been terminated but, for some reason, still must have its task structure in the process table is in the zombie state.

14. In a Solaris system, a User-Level Thread (ULT) in the active state is assigned to a(n) light-weight process (LWP) and executes while the underlying kernel thread executes.

15. In a Windows 2000 system, a process that has been selected to run next on a particular process moves from the Ready state to the Standby state.

Chapter 5 – Concurrency: Mutual Exclusion and Synchronization

True / False Questions:

16. T / F – Distributed processing can be defined as the management of multiple processes executing on multiple, distributed computer systems.

17. T / F – Both process interleaving and process overlapping are examples of concurrent processes and both present the same basic problems.

18. T / F – Concurrency issues are a concern on multiprocessor systems, but do not impact uniprocessor systems.

19. T / F – Starvation refers to the situation where competing processes are denied access to a resource due to scheduling problems.

20. T / F – Any facility or capability that is to provide support for mutual exclusion must make certain assumptions about relative process speeds and the number of processors in the system.

21. T / F – Peterson’s Algorithm for solving mutual exclusion is only valid for two processes and cannot be generalized to the case of n processes.

22. T / F – In a uniprocessor machine, concurrent processes cannot be overlapped; they can only be interleaved.

23. T / F – Weak semaphores guarantee freedom from starvation, but strong semaphores do not.

24. T / F – A finite circular buffer and an infinite buffer are two ways to implement a data storage area for the classic Producer/Consumer Problem.

25. T / F – The major difficulty with semaphores is that wait and signal operations may be scattered throughout a program and it is difficult to see the overall effect of these operations on the semaphores they affect.

26. T / F – Message passing provides both synchronization and communication, which are fundamental requirements for interacting processes.

27. T / F – In a message passing system, one queuing discipline alternative is to allow the receiver to inspect the message queue and select which message to receive next.

28. T / F – In the communications mechanism of a message passing system, only the receiver of the communication can be blocking.
29. T / F – In indirect addressing, as applied to message passing, messages are sent to a temporary shared data structure typically known as a mailbox.

30. T / F – The Producer/Consumer problem is typically considered a special case of the Readers/Writes problem, with only one reader and one writer

Multiple Choice Questions:

16. Concurrency plays a major part in which of the following specific contexts:
   a. Multiple applications
   b. Structured applications
   c. O/S structure
   d. All of the above

17. Examples of solutions to the concurrency problem that do not involve busy waiting are the following:
   a. Semaphores and monitors
   b. Message passing and caching
   c. Producers and consumers
   d. None of the above

18. A basic echo procedure (that echoes a typed character to the screen) running on a multiprocessor system can produce erroneous output if:
   a. Two processes deadlock while in the echo code
   b. Access to the echo procedure is unsynchronized
   c. Access to the echo procedure is synchronized
   d. None of the above

19. In order to implement mutual exclusion on a critical resource for competing processes, only one program at a time should be allowed:
   a. In the critical section of the program
   b. To perform message passing
   c. To Exhibit cooperation
   d. None of the above

20. The following requirement must be met by any facility or capability that is to provide support for mutual exclusion:
   a. Only one process at a time can be allowed into a critical code section
   b. A process remains in its critical code section for a finite time only
   c. No assumptions can be made about relative process speeds
   d. All of the above

21. Processes that are designed to be able to pass execution control back and forth between themselves are referred to as:
   a. Threads
   b. Coroutines

22. In a uniprocessor system, mutual exclusion can be guaranteed by:
   a. Overlapping processes
   b. Interleaving processes
   c. Disabling interrupts
   d. All of the above

23. A semaphore that does not specify the order in which processes are removed from the queue is called a:
   a. Weak semaphore
   b. Strong semaphore
   c. Binary semaphore
   d. None of the above

24. The finite circular buffer is used to implement which of the following basic queuing strategies:
   a. FILO
   b. LIFO
   c. FIFO
   d. None of the above

25. A chief characteristic of a monitor is:
   a. A maximum of two processes may be executing in a monitor at a time
   b. Local data variables of the monitor are accessible by any procedure requesting use of the monitor
   c. A process enters the monitor by invoking one of its procedures
   d. All of the above

26. In synchronization involving message passing, the sender of a message can be:
   a. Either blocking or non-blocking
   b. Only blocking
   c. Only non-blocking
   d. All of the above

27. In a system employing message passing, when a message is sent to a shared temporary data structure, this general approach is known as:
   a. Direct addressing
   b. Indirect addressing
   c. Blocking
   d. None of the above

28. In a system employing message passing, the typical message is divided into two primary sections:
   a. Header and mailbox
29. The Reader/Writer problem requires that certain conditions be satisfied, such as:
   a. Readers may read from the file while writers are writing to it
   b. Multiple writers may write to the file simultaneously
   c. Any number of readers may simultaneously read from the file
   d. None of the above

30. A reason why the Producer/Consumer problem cannot be considered a special case of the Reader/Writer problem with a single writer (the producer) and a single reader (the consumer) is:
   a. The producer and consumer must be both reader and writer
   b. The consumer must perform writes while the reader performs reads
   c. The Producer/Consumer problem doesn’t deal with concurrency issues
   d. None of the above

Fill-In-The-Blank Questions:

16. Many approaches to achieving mutual exclusion are software solutions that require the use of a technique called busy waiting.

17. The basic requirement for support of concurrent process is the ability to enforce mutual exclusion.

18. In order to protect shared variables (and other shared global resources) the system must control the code that accesses the variable.

19. The situation where Process 1 (P1) holds Resource 1 (R1), while P2 holds R2, and P1 needs R2 to complete and P2 needs R1 to complete is referred to as deadlock.

20. When only one process is allowed in its critical code section at a time, then mutual exclusion is enforced.

21. The technique in which a process can do nothing until it gets permission to enter its critical section but continues to test the appropriate variable to gain entrance is called busy waiting.

22. In multiprocessor configurations, special machine instructions that carry out two actions in a single instruction cycle are said to do so atomically.

23. A semaphore whose definition includes the FIFO policy for releasing blocked processes from the queue is called a strong semaphore.

24. The Barbershop Problem uses semaphores to implement concurrency.

25. A monitor supports synchronization by the use of condition variables that are contained within the monitor and accessible only within the monitor.

26. A blocking send, blocking receive message passing scenario is sometimes referred to as a rendezvous.

27. The shared data structures that temporarily hold messages in a message passing system employing indirect addressing are generally referred to as mailboxes.

28. In the direct addressing implementation of message passing, the “send” primitive includes a specific identifier of the destination process.

29. The classic concurrency problem that involves readers and writers that can both read from and write to a shared data area is the Producer/Consumer Problem.

30. The classic concurrency problem that involves multiple readers that can read from a shared data area when no single writer is exclusively writing to it is the Readers/Writers Problem.
Chapter 6 – Concurrency: Deadlock and Starvation

True / False Questions:
31. T / F – Deadlock can be defined as the periodic blocking of a set of processes that either compete for system resources or communicate with each other.
32. T / F – All deadlocks involve conflicting needs for resources by two or more processes.
33. T / F – A reusable resource is one that can be safely used by only one process at a time and is not depleted by that use.
34. T / F – A consumable resource is one that can be safely used by only one process at a time and is not depleted by that use.
35. T / F – Although deadlock can potentially exist without it, the condition known as Circular Wait is required for deadlock to actually take place.
36. T / F – The strategy of deadlock prevention is to design a system in such a way that the possibility of deadlock is minimized.
37. T / F – The Deadlock Avoidance approach to solving the deadlock problem allows the three necessary conditions for deadlocks to exist.
38. T / F – In the Resource Allocation Denial approach to Deadlock Avoidance, a safe state is defined as one in which all potential process sequences do not result in a deadlock.
39. T / F – Deadlock Detection strategies do not limit resource access or restrict process actions.
40. T / F – One of the most common approaches for recovery from deadlocked processes is to abort all deadlocked processes.
41. T / F – Although each strategy that deals with deadlocks has its advantages and disadvantages, the best solution to the problem is to choose one and stick with it.
42. T / F – The Dining Philosopher’s Problem illustrates basic problems in deadlock and starvation.
43. T / F – A pipe in UNIX is a circular buffer that allows two processes to communicate on the producer-consumer model.
44. T / F – One thread synchronization primitive supported by Solaris is the Mutual Exclusion (mutex) lock.

Multiple Choice Questions:
31. The permanent blocking of a set of processes that either compete for system resources or communicate with each other is called:
   a. Starvation  
   b. Deadlock 
   c. Prioritization 
   d. All of the above
32. All deadlocks involve conflicting needs for resources by:
   a. One or more processes 
   b. Two or more processes 
   c. Three or more processes 
   d. None of the above
33. A resource that can be created and destroyed is called:
   a. Reusable resource 
   b. Producible resource 
   c. Consumable resource 
   d. All of the above
34. An example of a consumable resource is:
   a. Messages 
   b. Printers 
   c. Main Memory 
   d. All of the above
35. A condition of policy that must be present for a deadlock to be possible is:
   a. Mutual exclusion 
   b. Hold and wait 
   c. No preemption 
   d. All of the above
36. A direct method of deadlock prevention is to prevent the occurrence of:
   a. Mutual exclusion 
   b. Hold and wait 
   c. Circular wait 
   d. All of the above
37. One approach to deadlock avoidance is called:
   a. Process Termination Denial 
   b. Resource Allocation Denial
38. In the Resource Allocation Denial approach to Deadlock Avoidance, a safe state is defined as one in which:
   a. At least one potential process sequence does not result in a deadlock
   b. All potential process sequences do not result in a deadlock:
   c. Several potential process sequences do not result in a deadlock:
   d. None of the above

39. A conservative strategy for dealing with deadlocks that involves limiting access to resources and imposing restrictions on processes is called:
   a. Deadlock Prevention
   b. Deadlock Avoidance
   c. Deadlock Detection
   d. None of the above

40. In deadlocked process recovery, selection criteria for choosing a particular process to abort or rollback includes designating the process with the:
   a. Most estimated time remaining
   b. Lowest priority
   c. Least total resources allocated so far
   d. All of the above

41. One approach to an integrated strategy for dealing with deadlocks involves the implementation of:
   a. Resource classes
   b. Process rollbacks
   c. Virtual memory
   d. None of the above

42. The Dining Philosopher’s Problem is a standard test case for evaluating approaches to implementing:
   a. Deadlock
   b. Starvation
   c. Synchronization
   d. All of the above

43. A software mechanism that informs a process of the occurrences of asynchronous events in UNIX are called:
   a. Pipes
   b. Messages
   c. Signals
   d. All of the above

44. Thread synchronization primitives supported by Solaris include:
   a. Mutual exclusion (mutex) locks
   b. Semaphores
   c. Condition variables
   d. All of the above

45. The family of synchronization objects implemented by W2K include:
   a. Mutex objects
   b. Semaphore objects
   c. Event objects
   d. All of the above

Fill-In-The-Blank Questions:

31. The permanent blocking of a set of processes that either compete for system resources or communicate with each other is called deadlock.

32. All deadlocks involve conflicting needs for resources by two or more processes.

33. A resource that can be created (produced) and destroyed (consumed) is called a consumable resource.

34. An example of a reusable resource is a processor/memory/semaphores/files/etc.

35. The hold and wait policy condition, which says a process may hold allocated resources while awaiting assignment of others, must be present for a deadlock to be possible.

36. An indirect method of deadlock prevention is to prevent the occurrence of one of the three necessary conditions for deadlock.

37. In Deadlock Avoidance, the Resource Allocation Denial strategy is also referred to as the banker’s algorithm.

38. An unsafe state is one in which every potential sequence of allocation of resources to processes results in a deadlock.

39. In Deadlock Detection, the O/S periodically performs an algorithm that allows it to detect the circular wait condition.

40. Once a deadlock has been detected, some strategy is needed for recovery.

41. One approach to an integrated strategy for dealing with deadlocks involves the implementation of resource classes.
42. The Dining Philosopher's Problem illustrates basic problems in **deadlock** and **starvation**.

43. The type of UNIX pipe that can be shared by unrelated processes is called a(n) **named** pipe.

44. The Solaris thread synchronization primitive that is used to wait until a particular condition is true is called a **condition variable**.

45. In a W2K system, the mutex object is used to enforce mutually exclusive access to a **resource**.