OS Test Review Questions - Chapter 1 -6

1. **Define jacketing.**	Converts a blocking system call into a non-blocking system call.
2. **Define the two main categories of processor registers.**	User-visible registers: Enable the machine or assembly language programmer to minimize main memory references by optimizing register use. Control and status registers: Used by the processor to control the operation of the processor and by privileged OS routines to control the execution of programs.
3. **Describe the round-robin scheduling technique.**	Giving each process in the queue some time in turn.
4. **Explain the difference between a monolithic kernel and a microkernel.**	A monolithic kernel is a very large kernel that includes scheduling, file system, networking, device drivers, memory management, and more. A microkernel is assigned only a few essential functions including address spaces, interprocess communication (IPC), and basic scheduling.
5. **Explain the distinction between a real address and a virtual address.**	A real address is the physical address in memory. A virtual address consists of a page number and an offset within the page. Each page of a process may be located anywhere in memory.
6. **For the processing model of Figure 3.6, briefly define each state.**	Running: the process that is currently being executed. Ready: a process that is prepared to execute. Blocked/Waiting: A process that cannot execute until some event occurs, such as the completion of an I/O operation. New: A process that has just been created but has not yet been admitted to the pool of executable processes by the OS. Exit: A process that has been released from the pool of executable processes by the OS.
7. **For what types of entities does the OS maintain tables of information for management purposes?**	Memory, I/O, file, and process.
8. **Give examples of reusable and consumable resources.**	Reusable: processors, I/O channels, main and secondary memory, devices, and data structures such as files, databases, and semaphores. Consumable: interrupts, signals, messages, and information in I/O buffers. A consumable resource is one that can be created and destroyed.
9. **Give four general examples of the use of threads in a single-user multiprocessing system.**	Foreground and background work. Asynchronous processing. Speed of execution. Modular program structure.
10. **Give three examples of an interrupt.**	Clock interrupt, memory fault, and I/O interrupt
11. **How are multiple interrupts dealt with?**	Two approaches: disable interrupts while an interrupt is being processed. A disabled interrupt simply means that the processor ignores any new interrupt request signal. The second approach is to define priorities for interrupts and to allow an interrupt of higher priority to cause a lower-priority interrupt handler to be interrupted.
12. **How can the circular wait condition be prevented?**	Defining a linear ordering of resource types.
13. **How can the hold-and-wait condition be prevented?**	Requiring that a process request all of its required resources at one time and blocking the process until all requests can be granted simultaneously.
14. **How is the execution context of a process used by the OS?**	The context includes all of the information that the OS needs to manage the process and that the processor needs to execute the process properly.

15. **In general terms, what are the four distinct actions that a machine instruction can specify?**	Processor-Memory, Processor-I/O, Data Processing, and Control.
16. **List and briefly describe the four main elements of a computer.**	Processor - controls operation of the computer and performs its data processing functions. Main Memory - Stores data and programs. This memory is volatile. I/O Modules - Move data between the computer and its external environment, e.g. disk storage devices. System Bus - Provides for communication among processors, main memory, and I/O modules.
17. **List and briefly explain five storage management responsibilities of a typical OS.**	Process isolation: The OS must prevent independent processes from interfering with each other's memory. Automatic allocation and management: Programs should be dynamically allocated across the memory hierarchy as required. Allocation should be transparent to the programmer. Support of modular programming: Programmers should be able to define program modules, and to create, destroy, and alter the size of modules dynamically. Protection and access control: Sharing of memory creates the potential for one program to address the memory space of another. This is desirable when sharing is needed by particular applications. At other times, it threatens the integrity of programs and even the OS itself. Long-term storage: Many application programs require means for storing information for extended periods of time, after the computer has been powered down.
18. **List four characteristics of a suspended process.**	Process not immediately available for execution. It may or may not be waiting on an event. The process was placed in a suspended state by an agent: itself, a parent process, or the OS. The process may not be removed from this state until the agent explicitly orders the removal.
19. **List four design issues for which the concept of concurrency is relevant.**	The OS must be able to keep track of the various processes. The OS must allocate and deallocate various resources for each active process. These resources include processor time, memory, files, and I/O devices. The OS must protect data and physical resources of each process against unintended interference by other processes. The functioning of a process, and the output it produces, must be independent of the speed at which its execution is carried out relative to the speed of other concurrent processes.
20. **List the requirements for mutual exclusion.**	It must be enforced. A process that halts in its noncritical section must do so without interfering with other processes. It must not be possible for a process requiring access to a critical section to be delayed indefinitely: no deadlock or starvation. When no process is in a critical section, any process that requests entry to its critical section must be permitted to enter without delay. No assumptions are made about relative process speeds or number of processors. A process remains inside its critical section for a finite time only.
21. **List the three control problems associated with competing processes and briefly define each.**	Mutual exclusion, deadlock (renewable resource), and starvation.
22. **List three advantages of ULTs over KLTs.**	Thread switching does not require kernel mode privileges. Scheduling can be application specific. ULTs can run on any OS.
23. **List three degrees of awareness between processes and briefly define each.**	Processes unaware of each other (competition). Processes indirectly aware of each other (cooperation by sharing). Processes directly aware of each other (cooperation by communication).

24. **List three general categories of information in a process control block.**	Process identification, processor state information, and process control information
25. **List two disadvantages of ULTs compared to KLTs.**	Many system calls are blocking on a typical OS and as a result, when a ULT executes a system call, not only is that thread blocked, but all of the threads within the process are blocked. In a pure ULT strategy, a multithreaded application cannot take advantage of multiprocessing.
26. **List two ways in which the no-preemption condition can be prevented.**	If a process holding certain resources is denied a further request, that process must release its original resources and, if necessary, request them again together with the additional resource. Alternatively, if a process requests a resource that is currently held by another process, the OS may preempt the second process and require it to release its resources.
27. **What are the four conditions that create deadlock?**	Mutual exclusion, hold and wait, no preemption, and circular wait.
28. **What are the steps performed by an OS to create a new process?**	Assign a unique process ID to the new process Allocate space for the process Initialize the process control block Set the appropriate linkages Create or expand other data structures
29. **What are the three conditions that must be present for deadlock to be possible?**	Mutual exclusion, hold and wait, and no preemption.
30. **What are the three objectives of an OS design?**	Convenience: An OS makes a computer more convenient to use. Efficiency: An OS allows the computer system resources to be used in an efficient manner. Ability to evolve: An OS should be constructed in such a way as to permit the effective development, testing, and introduction of new system functions without interfering with services.
31. **What are the two separate and potentially independent characteristics embodied in the concept of process?**	Resource ownership - The OS performs a protection function to prevent unwanted interference between processes with respect to resources. Scheduling/execution - The execution of a process follows an execution path (trace) through one or more programs. The process has an execution state and a dispatching priority and is the entity that is scheduled and dispatched by the OS.
32. **What are three contexts in which concurrency arises?**	Multiple applications, structured applications, and OS structure.
33. **What characteristics distinguish the various elements of a memory hierarchy?**	Capacity, access time, and cost. Faster access time, greater cost per bit. Greater capacity, smaller cost per bit. Greater capacity, slower access time.
34. **What common events lead to the creation of a process?**	a new batch job, an interactive logon, process created by OS to provide a service, and a process spawned by an existing process
35. **What conditions are generally associated with the readers/writers problem?**	Any number of readers may read the file. One writer at a time may write. If a writer is writing, no reader can read.
36. **What does it mean to preempt a process?**	Take away a resource it is using and give it to another process. That resource could be the OS.
37. **What is a monitor?**	A monitor is a programming-language construct that provides equivalent functionality to that of semaphores and it is easier to control. It consists of one or more procedures, and initialization sequence, and local data.
38. **What is an instruction trace?**	The sequence of instructions that execute for a process.
39. **What is an interrupt?**	An interrupt is a signal to the processor indicating an event that needs immediate attention. The processor suspends current activity to service the higher-priority condition (it executes a small program called an interrupt handler). After it handles this event, it resumes execution on the process it suspended.

40. **What is a process?**	It is a program in execution.
41. **What is cache memory?**	A small, fast memory between the processor and main memory. It enables memory access time to approach that of the fastest memories available and at the same time support a large memory size that has the price of less expensive types of semiconductor memories.
42. **What is multiprogramming?**	The act of holding more than one program in memory that the OS can switch between.
43. **What is multithreading?**	Multithreading is a technique in which a process, executing an application, is divided into threads that can run concurrently. A thread is a dispatch able unit of work. It includes a processor context (which includes the program counter and stack pointer) and its own data areas for a stack. A thread executes sequentially and is interruptible so that the processor can turn to another thread. A process is a collection of one or more threads and associated system resources. This corresponds closely to the concept of a program in execution.
44. **What is swapping and what is its purpose?**	Moving part or all of a process from main memory to disk. this puts the process in a Suspend state and enables another process to be brought into main memory.
45. **What is the basic requirement for the execution of concurrent processes?**	The ability to enforce mutual exclusion.
46. **What is the difference among deadlock avoidance, detection, and prevention?**	Deadlock prevention: we constrain resource requests to prevent at least on of the four conditions of deadlock. Done either indirectly, by preventing one of the three necessary policy conditions, or directly by preventing circular wait Deadlock avoidance: allows the three necessary conditions but makes judicious choices to assure that the deadlock point is never reached. As such, avoidance allows more concurrency than prevention. Deadlock detection: requested resources are granted to processes whenever possible. Periodically, the OS performs an algorithm that allows it to detect the circular wait condition described earlier in condition 4.
47. **What is the difference between a mode switch and a process switch?**	Mode switch switches between user and kernel. Process switch switches between aspects of process, such as running and blocked.
48. **What is the difference between an interrupt and a trap?**	An interrupt transfers control to an interrupt handler and then branches to an OS routine that is concerned with the particular type of interrupt that occurred. With a trap, the OS determines if the error or exception condition is fatal. If it is, then the currently running process is moved to the Exit state and a process switch occurs.
49. **What is the difference between binary and general semaphores?**	Binary semaphores may only have values of 0 or 1 while general semaphores can have values above or below these two numbers.
50. **What is the difference between strong and weak semaphores?**	Strong semaphores release the process that has been blocked the longest, while weak semaphores do not specify the order of process release.
51. **What is the distinction between blocking and nonblocking with respect to messages?**	The sender or receiver process cannot continue on until the sender's message is received (in the case of the sender) or the receiving process receives the message (in the case of receiver).
52. **What is the distinction between competing processes and cooperating processes?**	Competing processes are unaware of each other, while cooperating processes are aware of each other.
53. **What is the distinction between spatial locality and temporal locality?**	Spatial locality refers to the tendency of execution to involve a number of memory locations that are clustered. This reflects the tendency of a processor to access instructions sequentially. Temporal locality refers to the tendency for a processor to access memory locations that have been used recently.

54. **What is the kernel of an OS?**	It contains the most frequently used functions in the OS and, at a given time, other portions of the OS currently in use.
55. **What operations can be performed on a semaphore?**	semWait and semSignal
56. **What resources are typically shared by all of the threads of a process?**	The same address space and resources. Any alteration of a resource by one thread affects the other threads in the same process.
57. **Why are two modes (user and kernel) needed?**	It is necessary to protect the OS and key operating system tables, such as process control blocks, from interference by user programs. In kernel mode, the software has complete control of the processor and all its instructions, registers, and memory.