Solutions: Quiz 1

1. Which of the following programs always resides in memory?
   Ans.
   (A) Loader

2. When an interrupt arrives, the operating system
   Ans.
   (D) may change the state of the interrupted process to blocked and schedule another process

3. What is the advantage of separating a linker and loader into separate programs? Under what circumstances would a combined linking loader be useful?
   Ans.
   Linker outputs the executable code as a file that can be stored on secondary storage. Such an executable file can be loaded into memory any number of times after linking once. If they were a single program, the code would have to be linked each time the program needs to be run.

   If linking and loading are separate, programs that use system libraries will have to be relinked each time a new version of the library is released. Instead, with a linking loader, library sections can be dynamically linked i.e., the user programs can be bound to libraries as the programs are running, loading libraries in the middle of program execution.

4. Nearly every programming system produced in the past 50 years includes a linker. Why?
   Ans.
   Programs are built from multiple subprograms, references from one subprogram to other are made using program symbols. Linkers resolve symbols and patch the references of symbols to their location in the resultant executable file.

5. Which of the functionalities listed below need to be supported by the operating systems for the following two settings: (a) handheld devices and (b) real-time systems. a. Batch programming
   b. Virtual memory
   c. Time sharing

   (a) handheld devices
   Ans.
   Virtual memory.
   (b) real-time systems
   Ans.
   Time sharing.

6. Distinguish between the client-server and peer-to-peer models of distributed systems.
   Ans.
   In the client-server model, certain nodes (called clients) in the network send requests to other nodes (called servers) requesting for action. The server then executes the action and sends back the result of the action to the client.
   In the peer-to-peer model, all nodes within the system are considered peers, and each may act as client or server depending on whether it is requesting or providing a service.
   In a client-server system, server is a bottleneck; whereas in a peer-to-peer services can be provided by several nodes distributed throughout the network

7. What is the purpose of interrupts? What are the differences between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?
   Ans.
   Interrupt is a hardware mechanism that enables a device to notify the CPU.
   A trap is generated either by an error (e.g. division by zero) or by a user request for an operating

system service
User can generate traps by using system calls.

8. Direct memory access is used for high-speed I/O devices in order to avoid increasing the CPU's execution load.

   a. How does the CPU interface with the device to coordinate the transfer?
   Ans.

   (A) A command block is written to main memory. This block contains pointers to the source and destination of the transfer and a count of number of bytes to be transferred.
   (B) CPU writes the address of the command block to the DMA controller.
   (C) DMA controller operates the memory bus directly without any CPU intervention.a
   (D) The device controller and DMA perform the transfer by handshaking using DMA-request and DMA-acknowledge.

   b. How does the CPU know when the memory operations are complete?
   Ans.
   DMA controller interrupts the CPU to notify completion of the transfer.
   c. The CPU is allowed to execute other programs while the DMA controller is transferring data. Does this process interfere with the execution of user programs? If so, describe what forms of interference are caused?
   Ans.
   During data transfer using DMA, the DMA controller sometimes seizes the memory bus when another program requires it. This is called "cycle stealing".

9. Give two reasons why caches are useful. What problems do they solve? What problems do they cause? If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?
   Ans.
   Reasons for usefulness of caching:

   (A) Locality of reference.
   (B) Cache provides a temporary storage for maintaining a sustained pipeline of information.

   Caching decreases the amount of idle time, by making information available faster.
   Caches pose the problem of maintaining cache-coherency where all versions of a particular information need to be kept consistent with each other.

   As we move higher in the storage hierarchy, cost of storage per bit increases. Thus, it is infeasible to have a cache as large as that of the device.

10. Discuss, with examples, how the problem of maintaining coherence of cached data manifests itself in the following processing environments:

    a. Single-processing systems
    Ans.
    In a multitasking environment, when several processes wish to access a certain piece of information A, it needs to be ensured that each of these processes obtains the most recently updated value of A. b. Multiprocessor systems
    Ans.
    Multiprocessor environment must provide cache coherency in hardware such that all CPUs have the most recent value in their cache.
    c. Distributed systems
    Ans.

Several copies of the same data item may be present on different computers. All these copies should be consistent with each other.

11. The services and functions provided by an operating system can be divided into two main categories. Briefly describe the two categories and discuss how they differ.
    Ans.
    a. Functions that are directly useful to the user.
    These include such functions as the user interface provided to the user, facilities for program execution, I/O operations, file manipulation.
    b. Functions that ensure efficient operation of the system.
    These are functions like allocation of resources to the processes (user processes and system processes), protection and control of access to avoid interference with other processes in the system.

12. Describe the actions taken by the kernel to context switch between the processes.
    Ans.
    Kernel saves the context (value of CPU registers, process state, memory management information) of the old process in its PCB and loads the saved context of the new process scheduled to run. The new process resumes its execution based on the saved value of its program counter.

13. What is the Unix command to determine the state of all processes currently managed by the system?
    Ans.
    ps -eo user,pid,state,command OR
    ps -Ao user,pid,state,command OR
    top