**Question Bank**

**Ques 1. Why is the operating system important?**

OS is the heart of a system without which it cannot function and is useless. It is an essential component as it acts as a link between computer software and users. It helps communicate with the system hardware and acts as a resource manager. It provides services to a user and performs all application tasks.

**Ques 2. State the main purpose of an OS and types of OS?**

An OS executes user programs so that users can understand and interact with computer systems easily. It also improves system performance by managing all computational activities in the system. Apart from this, it also manages processes, the operation of all hardware and software, and computer memory.

Following are the five main types of OS:

* Batch Operating System
* Time-sharing operating System
* Distributed operating System
* Network operating system
* Real-time operating system

**Ques 3. State the benefits of a multiprocessor system?**

As the name suggests, this OS consists of multiple processors that share a common physical memory and that operate under a single OS. The system divides a task into subtasks that execute parallely in different processors whose working is transparent to the users.

Following are the benefits of a multiprocessor system:

* It improves the performance of systems that concurrently run multiple programs.
* Increased system throughput and speed and high computing power.
* Multiple processors help complete a larger number of tasks in less time.
* It is cost-effective.
* It improves system reliability.

**Ques 4. What is the importance of Kernel in an OS?**

Kernel helps the OS manage the operations of the computer system and hardware, basically the memory and CPU time. It uses inter-process communication and system calls to act as a bridge between applications and data processing performed at the hardware level.

It handles disk management, task management, and memory management and decides which process to allocate to the CPU for execution. It also manages communication between the software and hardware.

**Ques 5. Explain the kinds of operations that are possible on a semaphore in detail.**

The two atomic operations possible on a semaphore are:

* **Wait():** The wait (sleep or down) operation decrements the value of an argument. No operation occurs if the argument is negative or zero.
* **Signal():** The signal (wake-up or up) operation increases the value of an argument.

**Ques 6. Define Scheduling Algorithms and name different types of scheduling algorithms.**

Scheduling algorithm schedules processes on the processor in an efficient and effective manner. A Process Scheduler performs scheduling. It maximizes CPU utilization by increasing throughput.

Following are the types of process scheduling algorithms:
1. First-Come, First-Served (FCFS) Scheduling
2. Shortest-Job-Next (SJN) Scheduling
3. Priority Scheduling
4. Shortest Remaining Time
5. Round Robin(RR) Scheduling
6. Multiple-Level Queues Scheduling
7. Multilevel Feedback Queues Scheduling
8. Highest Response Ratio Next

**Ques 7. What is demand paging and how can the user perform demand paging in a system?**

Demand paging loads pages into system memory on demand. We use this method in virtual memory. A page enters into the memory when the OS references a location on that particular page during execution.

Following are the steps to perform demand paging:

* Make an attempt to access the page.
* Continue processing instructions as normal if the page is present in the memory.
* If the page is unavailable, then a situation of page-fault trap occurs.
* Make sure that the memory reference is a valid reference to a location on secondary memory in order to page in the required page. If it is not, then the process terminates due to illegal memory access.
* We need to schedule disk operation so that the desired page can be read into the main memory.
* Now, the interrupted instruction can restart.

**Ques 8. Can you relate the microprocessor to the OS?**

The OS controls a microprocessor and an OS with a microprocessor is known as a micro-controller.

**Ques 9. How is a hard deadline different from a soft deadline?**

A hard deadline system is very strict with deadlines and does not miss a single deadline. If it misses a deadline then the system fails. Whereas a soft deadline system is more lenient and the user can have some misses. We can fix the miss number and frequency with the help of algorithms. The system fails if these conditions fail.

**Ques 10. What is process synchronization?**

Process Synchronization means coordinating the execution of processes such that no two processes access the same shared resources and data. It is required in a multi-process system where multiple processes run together, and more than one process tries to gain access to the same shared resource or data at the same time. Process synchronization has two types, namely, Independent Process and Cooperative Process.

**Ques 11. What is IPC? Name the different IPC mechanisms?**

IPC stands for Interprocess Communication. It helps exchange data between multiple threads in one (or more) processes or programs. It doesn’t matter whether the process is running on single or multiple computers (connected by a network). It allows coordination of activities among various program processes running concurrently in an OS.

Following are the different IPC Mechanisms:

* Pipes
* Message Queuing
* Semaphores
* Socket
* Shared Memory
* Signals

**Ques 12. What are overlays in the OS?**

A programming method divides processes into multiple pieces in order to save the important instructions in memory. This method is known as an overlay and needs no support from the OS. Programs bigger in size than physical memory can also be run using this as it stores the important data and instructions only.

**Ques 13. Define thrashing in OS?**

In thrashing, the CPU performs more swapping or paging work as compared to productive work. When the process does not have enough pages, the page-fault rate increases, degrading or collapsing the system performance. Thrashing reduces CPU utilization and multiprogramming.

A high degree of Multiprogramming, unequal number of frames and processes requirement, and more swapping of processes when CPU utilization is low are some causes of thrashing.

**Ques 14. What do you understand by the term daemon?**

A computer program running as a background process, instead of being under the direct control of an interactive user is known as a daemon. The process names of a daemon end with ‘d’ so that the user can differentiate between a daemon and a normal computer program.

**Ques 15. What do you mean by a thread?**

A thread is the basic unit of the process code and is called a lightweight process within a process that cannot exist outside a process. It has its own program counter. Threads share information like code segments, open files, and data segments with each other. If there is any change in the information of one thread all the other threads can see that.

A thread keeps track of:

* instructions to execute next.
* system registers that have the current working variables of a process.
* a stack containing the execution history.

**Ques 16. Define a process and its different states.**

The program that the OS is currently executing is known as a process. It is the basic unit of work that the OS implements in the system and it takes place in a sequence. A program during execution becomes a process and performs all the tasks for the user. A process has four sections. These sections are as follows: Stack, Heap, Text, and Data. The different states of a process are: Start, Ready, Running, Waiting, and Terminated or Exit.

**Ques 17. What do you mean by the First-Come-First-Serve scheduling algorithm?**

FCFS is an OS scheduling algorithm that executes requests and processes automatically. The OS stores these processes in the form of a queue in order of their arrival. Processes requesting the CPU first, get the CPU allocation first in this easy and simple algorithm with the help of a FIFO queue.

After entering the ready queue, the PCB of a process links itself with the tail of the queue. Thus, when the CPU becomes free, it is assigned to the process that is at the beginning of the queue. The process that has the least arrival time receives the processor first.

**Ques 18. What is a bootstrap program in OS?**

An OS initializes a bootstrap program during startup i.e., it is the first code that executes when a system starts. Booting is a bootstrapping process or program that loads the OS and ensures the correct working of the OS. Boot blocks store the OS at a fixed location on the disk where the OS locates the kernel and loads it into the main memory. After this process performs, the program starts its execution.

**Ques 19. State the difference between paging and segmentation?**

The following table states the differences between paging and segmentation:

|  |  |
| --- | --- |
| **Paging** | **Segmentation** |
| Fix sized pages | Variable sized segments |
| Internal fragmentation | No internal fragmentation |
| Hardware decides the page size | The user decides the segment size |
| Faster memory access | Memory access is slower as compared to the paging |
| Page table stores data | Segmentation table stores data |
| Sharing of procedures is not allowed | Sharing of procedures is allowed |
| Cannot distinguish and secure procedures and data | Can separate and secure procedures and data |
| 1-D address space | Multiple independent address spaces |
| A single integer address is divided into page numbers and offset by the hardware | The user divides address in segment number and offset |

**Ques 20. What is the main objective of multiprogramming?**

In this, the multiple tasks are stored in the system memory that is acquired from the job pool and the OS picks one task and starts executing it. The OS fetches another job from the memory when the current job requires an I/O. In the case of multiple jobs in a ready state, which job to choose is decided through the process of CPU Scheduling. It never leaves a CPU idle and maximizes CPU usage.

**Ques 21. What is RAID configuration in an OS? Also, state the different levels of RAID.**

RAID stands for Redundant Arrays of Independent Disks. It is a method that helps store data on multiple hard disks and helps achieve data redundancy that reduces data loss. It is a data storage virtualization technology that balances data protection, storage space, system performance, etc. It also improves the overall performance and reliability of data storage and increases the storage capacity of a system.

Following are the different levels of RAID:

RAID 0: Non-redundant striping: Increase server performance.
RAID 1: Mirroring and duplexing: Performs disk mirroring and helps implement fault tolerance in a simple manner.
RAID 2: Memory-style error-correcting codes: Uses dedicated hamming code parity.
RAID 3: Bit-interleaved Parity: Stores parity information using a dedicated parity drive.
RAID 4: Block-interleaved Parity: Similar to RAID 5, just confines all the parity data to a single drive.
RAID 5: Block-interleaved distributed Parity: Provides better performance as compared to RAID 1.
RAID 6: P+Q Redundancy: Provides fault tolerance for two drive failures.

**Ques 22. What is a Pipe? When is a pipe used?**

The pipe is used for inter-process communication. This half-duplex method allows communication between two related processes. A half-duplex method allows the first process to communicate with the second process. In order to achieve a full-duplex, we need to add another pipe.

**Ques 23. What do you mean by Reentrancy?**

A function in which various clients use and share a single copy of a program at the same time is known as Reentrancy. It doesn’t deal with concurrency and has two major functions:
Program code is unable to change or modify itself.
Different disks store local data for every client process.

**Ques 24. State the differences between Multitasking and Multiprocessing.**

The following table states the differences between Multitasking and Multiprocessing:

|  |  |
| --- | --- |
| **Multitasking** | **Multiprocessing** |
| Performs multiple tasks at the same time using a single processor | Performs multiple tasks at the same time using multiple processors |
| Only one CPU | More than one processes |
| More economical | Less economical |
| Less efficient | More efficient |
| Fast switching among tasks | Smooth processing of multiple tasks at a time |
| More time for execution | Less time for job processing |

**Ques 25. Define Sockets in OS.**

A socket, used in client-server-based systems is the endpoint for IPC i.e., a combination of an IP address and port number. They make the creation of network-enabled programs easy and allow communication between two processes on the same or different machines. There are four types of sockets namely, Stream Sockets, Datagram Sockets, Sequenced Packet Sockets, and Raw Sockets.

**Ques 26. What is a zombie process?**

A defunct process that terminates or completes but the PCB is still present in the main memory as there is still an entry in the process table to report to its parent process. This process is known as a zombie process. In a way, it is dead as it doesn’t consume any resources, but it exists too as it shows that resources are in possession of the process.

**Ques 27. What is cascading termination?**

When a process creates a new process, the identity of the child process passes onto the parent process. When the OS initiates the termination of the parent process or if the parent process exits the child process also needs to terminate.

**Ques 28. What is swap space?**

Swap space specifies the space that Linux uses to hold concurrent running processes temporarily. The user uses it when the RAM doesn’t have enough space to hold all the executing programs.

**Ques 29. How does a system call works?**

Following are the steps on how a System Call works:

Step 1: The processor executes a process in the user mode until a system call interrupts it.

Step 2: Then on a priority basis, the system call is executed in the kernel mode.

Step 3: After the completion of system call execution, control returns to user mode.,

Step 4: The execution resumes in Kernel mode.

**Ques 30. What happens during a remote procedure call?**

There are two activities that take place during a Remote Procedure Call(RPC):

1. The OS suspends and transfers the calling environment and procedure parameters respectively, across the network and to the environment where the procedure executes.

2. The OS transfers back the result produced by a procedure to the calling environment. Execution also resumes just like a regular procedure call.