

OS KTU VIVA QUESTIONS

1. What is an Operating System (OS)?

An Operating System is system software that acts as an interface between the user and the computer hardware. It manages hardware resources and provides services for computer programs.

2. What is a Shell?

A Shell is a command-line interpreter that allows users to interact with the OS by typing commands. It translates user input into actions taken by the OS.

Examples:

- **Bash shell** in Linux
 - **Command Prompt (cmd)** in Windows
-

3. What is Kernel?

The Kernel is the core part of the OS that directly interacts with the hardware. It manages CPU, memory, processes, devices, etc.

4. What is a Process?

A process is an instance of a program in execution. It includes the program code and its activity (e.g., program counter, registers, variables).

5. What is a Thread?

A thread is the smallest unit of execution within a process. Multiple threads can exist within the same process and share its resources.

6. What is a File System?

A file system organizes and stores data on storage devices. It allows users to create, read, write, and manage files and directories.

7. What is System Call?

System calls provide an interface between a process and the operating system, allowing programs to request services (e.g., file operations, process control).

8. What is a Scheduler?

A scheduler is a component of the OS that decides which process runs next based on a specific algorithm.

9. What is a Deadlock?

Deadlock is a situation where a group of processes are blocked, each waiting for resources held by others, and none can proceed.

10. What is Paging and Segmentation?

- **Paging:** Divides memory into fixed-size pages.
 - **Segmentation:** Divides memory into variable-sized segments based on logical divisions like code, data, stack.
-

SET I

1. **Time Sharing System:** An OS that allows multiple users to use a computer simultaneously by rapidly switching between them, giving the illusion of parallelism.
2. **Types of OS:**
 - Batch OS
 - Time-Sharing OS
 - Distributed OS
 - Network OS
 - Real-Time OS

- Mobile OS
 - 3. **Process:** A program in execution.
 - **States:** New, Ready, Running, Waiting, Terminated.
 - 4. **Virtual Memory:** A memory management technique that gives the illusion of a large main memory by using disk space.
 - 5. **Paging:** Dividing memory into fixed-size pages (logical) and frames (physical) to avoid fragmentation.
 - 6. **Logical vs Physical Address:**
 - Logical: Address generated by CPU.
 - Physical: Actual address in RAM.
 - 7. **Semaphore:** A synchronization tool used to manage access to shared resources and avoid race conditions.
 - 8. **Deadlock Conditions:**
 - Mutual Exclusion
 - Hold and Wait
 - No Preemption
 - Circular Wait
 - 9. **Banker's Algorithm:** A deadlock avoidance algorithm that checks whether resource allocation is safe.
 - 10. **Fragmentation:**
 - **Internal:** Unused memory within allocated space.
 - **External:** Unused scattered free space.
-

SET II

1. **Functions of OS:**
 - Process Management
 - Memory Management
 - File System Management
 - Device Management

- Security and User Interface
 - 2. **Booting:** Process of loading OS into main memory on system startup.
 - 3. **Types of Schedulers:**
 - Long-term: selects jobs from job pool.
 - Short-term: selects processes for execution.
 - Medium-term: swaps processes in/out of memory.
 - 4. **IPC (Inter-Process Communication):** Mechanisms for processes to communicate (e.g., shared memory, message passing).
 - 5. **Multiprogramming:** Running multiple programs simultaneously to improve CPU utilization.
 - 6. **Thrashing:** Excessive paging causing low performance due to lack of available memory.
 - 7. **fork():** System call in Unix to create a new process (child).
 - 8. **User Mode vs Kernel Mode:**
 - User Mode: Restricted access to hardware.
 - Kernel Mode: Full system access.
 - 9. **Overlays:** Technique to run a process larger than physical memory by loading code in parts.
 - 10. **PCB (Process Control Block):** Data structure containing process information (PID, state, registers, etc.)
-

SET III

1. **Process Scheduling Techniques:**
 - FCFS
 - SJF
 - Priority Scheduling
 - Round Robin
 - Multilevel Queue
2. **Context Switching:** Saving and restoring process states during a switch.

3. **Process Synchronization:** Ensuring correct process execution when accessing shared resources.
 4. **Semaphore Operations:** wait() (P) and signal() (V).
 5. **Critical Section Problem:** Ensuring that only one process accesses critical section at a time.
 6. **TLB (Translation Lookaside Buffer):** A cache that stores recent virtual-to-physical address mappings.
 7. **Round Robin Scheduling:** Each process is given a fixed time quantum in cyclic order.
 8. **Starvation:** A process waits indefinitely for resources.
 9. **Fragmentation Types:**
 - Internal
 - External
 10. **Logical vs Physical Address Space:**
 - Logical: Generated by CPU.
 - Physical: Mapped to memory by MMU.
-