Unit Test 1- Question Bank (CO/AN/BD)

2 Marks Questions

1. List any four services of operating system.

Ans. **1.Program Execution:**

2. I/O Operations:

3.Communication Between Processes

4.File Management

5.Memory Management

6.Process Management

7.Security and Privacy

8.Resource Management

9.User Interface

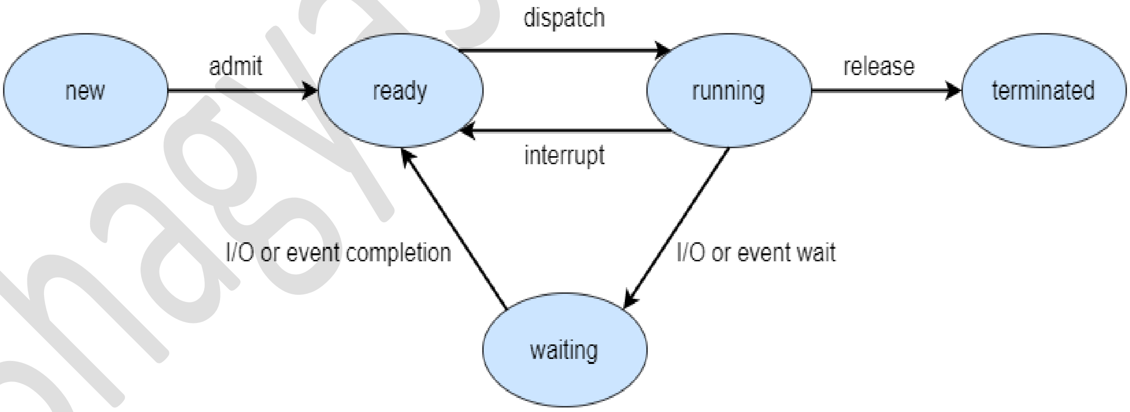
10.Networking

11.Error Handling

12. Time Management

2. Draw labelled process state diagram.

Ans.



3. State the use of KILL command with suitable examples.

Ans. The kill command is commonly used to:

- Terminate unresponsive processes.
- Manage system resources by stopping unnecessary processes.
- Send specific signals to processes for custom handling.

Syntax:

```
kill [signal] PID
```

Example:

```
kill -9 1212
```

4. Give commands to perform following tasks:

a To add delay in script

b To terminate a process

Ans.

a To add delay in script- sleep

b To terminate a process- kill

5. Describe any four advantages of threads.

Ans. **Benefits of Thread in Operating System**

- **Responsiveness*:*** If the process is divided into multiple threads, if one thread completes its execution, then its output can be immediately returned.
- **Faster context switch*:*** Context switch time between threads is lower compared to the process context switch.
- **Effective utilization of multiprocessor system*:*** If we have multiple threads in a single process, then we can schedule multiple threads on multiple processors. This will make process execution faster.
- **Resource sharing*:*** Resources like code, data, and files can be shared among all threads within a process
- **Communication*:*** Communication between multiple threads is easier, as the threads share a common address space.
- **Enhanced throughput of the system*:*** If a process is divided into multiple threads, and each thread function is considered as one job, then the number of jobs completed per unit of time is increased, thus increasing the throughput of the system.

6. Define System call and enlist its types.

Ans. System call is programmatic way in which computer program request a service from the kernel of OS . System Calls act as a bridge between an operating system (OS) and a running program.

Types of system calls:

1. **File System Operations**
2. **Process Control**
3. **Memory Management**
4. **Interprocess Communication (IPC)**
5. **Device Management**

7. State any two scheduling objectives.

Ans. Scheduling Objectives Are:

Fairness – Every process gets a fair share of CPU time without being blocked.

Maximize Resource Utilization – Keep CPU, memory, and I/O devices busy with minimal idle time.

Maximize Throughput – Increase the number of processes completed in a given time.

Policy Enforcement – Follow predefined operating system rules while scheduling.

Avoid Starvation – Ensure no process waits indefinitely for CPU time.

Minimize Overhead – Reduce extra CPU time spent on scheduling itself.

Minimize Response Time – Provide the fastest possible first response to processes.

Ensure Predictability – Maintain consistent and expected scheduling performance.

Efficiency – Use all system resources effectively without waste.

Enforce Priorities – Give preference to high-priority processes over others.

8. List any four scheduling criteria of CPU.

Ans. Scheduling criteria Of CPU are:

CPU Utilization – Keep the CPU as busy as possible without staying idle.

Throughput – Maximize the number of processes completed per unit time.

Turnaround Time – Minimize the total time taken from process submission to its completion.

Waiting Time – Reduce the total time a process spends waiting in the ready queue.

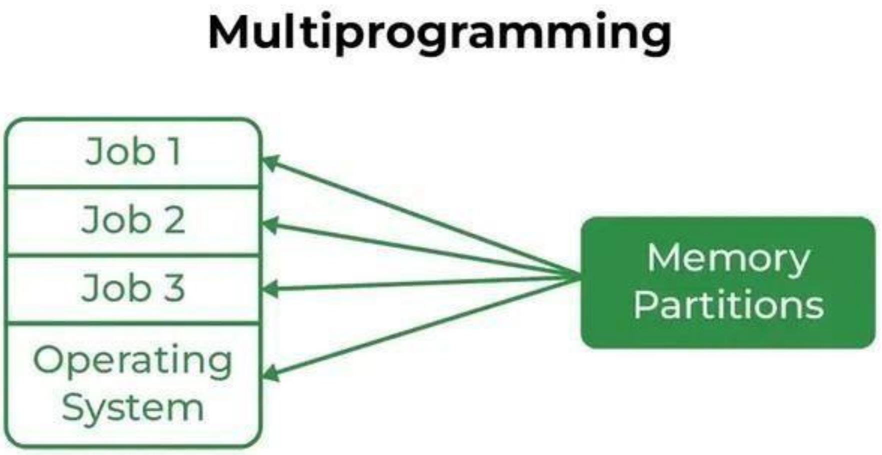
Response Time – Minimize the time between process request and the system's first response.

Fairness – Ensure every process gets a fair share of CPU time without being ignored.

Predictability – Provide consistent and expected scheduling performance.

9. Explain multiprogrammed O.S. with suitable diagram.

Ans. Multiprogramming Operating Systems can be simply illustrated as more than one program is present in the main memory and any one of them can be kept in execution. This is used for better utilization of resources.



10. Name any two GUI-based Operating Systems and 2 CLI based OS.

Ans. GUI-based Operating Systems: 1) Windows 2)Linux

Command Line Operating System : 1) MS-DOS 2) Unix

4 Marks Question Bank

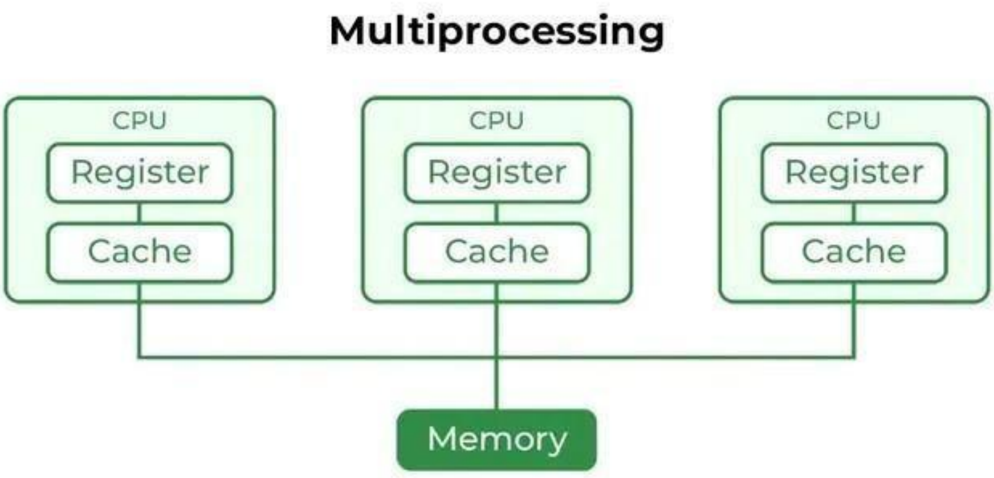
1. **What is multiprocessor system? Give two advantages of it.**

Ans. Two or more processors work together to perform tasks. Shared memory and communication paths. A Multi-Processing Operating System is a type of Operating System in which more than one CPU is used for the execution of resources.

- Multi-processing operating system consists of multiple CPUs. Each CPU is connected to the main memory.
- The task to be performed id divided among all the processors.
- For faster execution and improved performance, each processor is assigned a specific task.
- Once all the tasks of each processor are completed they are compiled together in order to produce a single output.
- The allocation of resources for each processor is handled by the operating system. This process results in better utilization of the available resources and improved performance.

  Types of Multiprocessor System:

  1) Symmetric
  2) Asymmetric



Advantages of Multiprocessor System:

1. **Increased Throughput** – More processes can be executed at the same time, which increases overall system performance.
2. **Parallel Processing** – Tasks can be divided among processors, reducing execution time.
3. **Higher Reliability** – If one processor fails, others can take over (fault tolerance).
4. **Better Resource Utilization** – Processors share memory and I/O devices, leading to efficient usage of resources.

5. **Scalability** – System performance can be improved by adding more processors.

2. **Differentiate Time sharing and Multi programming OS.**

Ans.

Feature	Time-Sharing System	Real-Time System
Definition	Allows multiple users to share system resources at the same time.	Responds to inputs or events within a guaranteed time.
Goal	Provide quick response and efficient sharing of resources.	Ensure timely and predictable response to critical events.
Response Time	Short, but not guaranteed.	Strictly bounded (guaranteed) response time.
Feature	Time-Sharing System	Real-Time System
Usage	Used in general-purpose systems like multi-user systems, office systems.	Used in mission-critical systems like aircraft, medical devices, industrial control.
Example OS	UNIX, Linux (as time-sharing), Windows Server	VxWorks, RTLinux, QNX
User Interaction	Users interact directly with the system.	May be automatic, with minimal human interaction.
Example Application	Online ticket booking, shared coding environments	Pacemaker, anti-lock braking system, missile guidance system

3. **List components of OS. Explain process management in details.**

Ans. Components of OS:

- **Process Management:**
- **Main Memory Management:**
- **File Management:**
- **I/O Management:**

- **Secondary Storage Management:**
- **Security Management**
- **Network Management**

Process Management:

A process is program or a fraction of a program that is loaded in main memory. A process needs certain resources including CPU time, Memory, Files, and I/O devices to accomplish its task. The process management component manages the multiple processes running simultaneously on the Operating System.

A program in running state is called a process.

The operating system is responsible for the following activities in connection with process management:

- Create, load, execute, suspend, resume, and terminate processes.

- Switch system among multiple processes in main memory.

- Provides communication mechanisms so that processes can communicate with each others

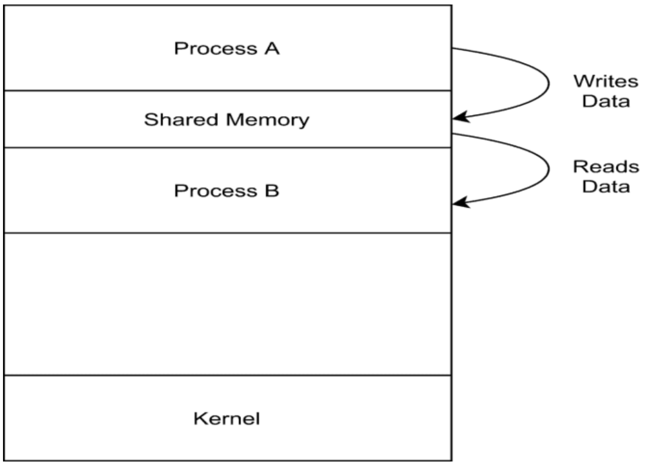
- Provides synchronization mechanisms to control concurrent access to shared data to keep shared data consistent.

- Allocate/de-allocate resources properly to prevent or avoid deadlock situation.

4. **Describe shared memory system in IPC with suitable diagram.**

Ans. Shared memory is a region of memory that is accessible to multiple processes. This allows processes to communicate with each other by reading and writing data from the shared memory region.

Shared memory is a fast and efficient way for processes to communicate, but it can be difficult to use if the processes are not carefully synchronized.



A and B will have to perform the following steps −

Step 1 − Process A has some data to share with process B. First A takes initiative and establishes a shared memory region in its own address space and stores the data or information to be shared in its shared memory region.

Step 2 − Now, B requires the information stored in the shared segment of A. So, process B needs to attach itself to the shared address space of A. Now, B can read out the data from there.

Step 3 − The two processes can exchange information by reading and writing data in the shared segment of the process.

Advantages:

1. **Faster Communication:**

   Data doesn't need to be copied between processes; all processes can access the same memory space.
   Best suited for high-speed data exchange.

2. **Low Overhead:**

   No need for frequent system calls once the memory is mapped, reducing kernel involvement.

Disadvantages:

Complex Synchronization: Requires careful synchronization (using semaphores, mutexes, etc.) to avoid race conditions and ensure data consistency.

Security Risks: If not properly managed, one process can overwrite or corrupt another's data

5. **Enlist the types of scheduler. Describe long term and short term scheduler in brief.**

Ans. Schedulers in operating system are special system software's which handles process scheduling in various ways.
Schedulers main task is to select the jobs to be submitted into the system and to decide which process to run.

The types of scheduler:

Long-Term Scheduler

Short-Term Scheduler
Medium-Term Scheduler

1.Long-Term Scheduler

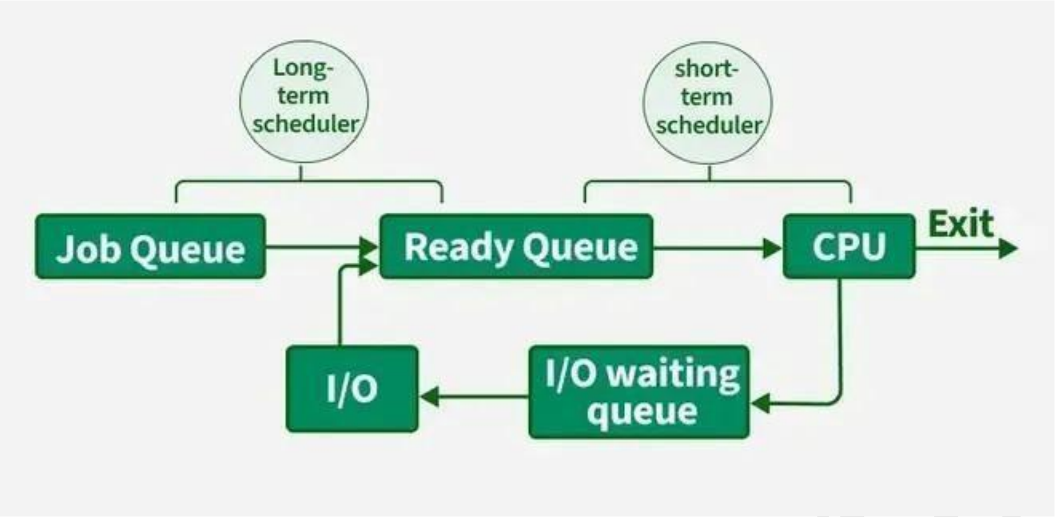
Long Term Scheduler loads a process from disk to main memory for execution. The new process to the 'Ready State'.

- It mainly moves processes from Job Queue to Ready Queue.
- It controls the Degree of Multi-programming, i.e., the number of processes present in a ready state or in main memory at any point in time.
- It is important that the long-term scheduler make a careful selection of both I/O and CPU-bound processes.
- I/O-bound tasks are which use much of their time in input and output operations while CPU-bound
- processes are which spend their time on the CPU. The job scheduler increases efficiency by maintaining a balance between the two.
- In some systems, the long-term scheduler might not even exist. For example, in time-sharing systems like Microsoft Windows, there is usually no long-term scheduler. Instead, every new process is directly added to memory for the short-term scheduler to handle.
- Slowest among the three (that is why called long term).

2.Short-Term Scheduler

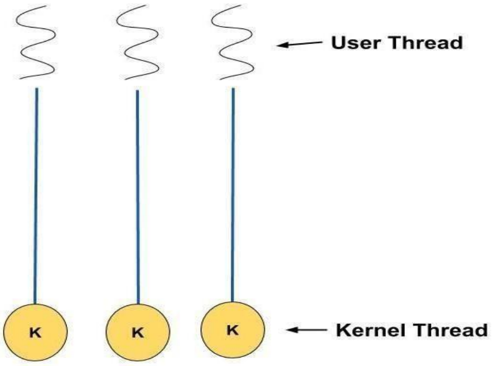
CPU Scheduler is responsible for selecting one process from the ready state for running (or assigning CPU to it).

- STS (Short Term Scheduler) must select a new process for the CPU frequently to avoid starvation.
- The CPU scheduler uses different scheduling algorithms to balance the allocation of CPU time.
- It picks a process from ready queue.
- Its main objective is to make the best use of CPU.
- It mainly calls dispatcher.(The dispatcher is responsible for loading the process selected by the Short-term scheduler on the CPU (Ready to Running State).
- Fastest among the three (that is why called Short Term).
- Saving context (process control block) of previously running process if not finished.
- Switching system mode to user mode.
- Jumping to the proper location in the newly loaded program.

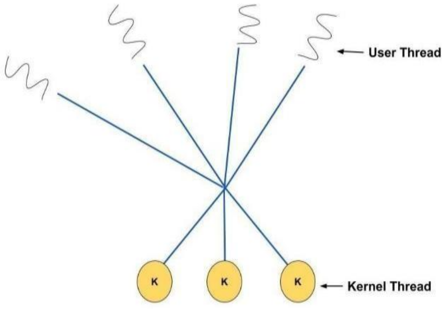


6. **Explain multithreading model in detail.**

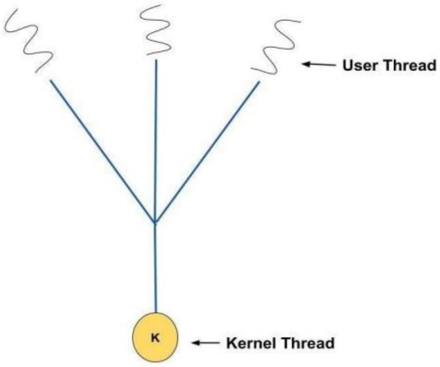
Ans. Multithreading models define the relationship between user-level threads and kernel-level threads within an operating system. These models dictate how user-space threads, managed by a thread library, are mapped to kernel-space threads, which are handled by the operating system kernel. The three main models are:



One-to-One Model



Many-to-Many Model



Many-to-One Model

- **One-to-One Model:**
  - This model maps each user-level thread to a separate kernel thread.
  - It allows for true concurrency on multiprocessor systems, as multiple kernel threads can run in parallel on different processors.
  - If one user-level thread performs a blocking system call, other threads within the same process can continue to execute, as they are backed by independent kernel threads.
  - A drawback is the overhead associated with creating and managing a large number of kernel threads, which can consume significant system resources and potentially limit the number of threads an application can create.
- **Many-to-One Model:**
  - In this model, many user-level threads are mapped to a single kernel thread.
  - Thread management, including creation, scheduling, and synchronization, is handled entirely in user space, making it efficient as there are no kernel overheads for thread operations.
  - A significant disadvantage is that if one user-level thread performs a blocking system call, the entire process (and thus all user-level threads within it) will block, preventing other threads from running.
  - This model also prevents true parallelism on multiprocessor systems, as only one kernel thread (and therefore one user-level thread at a time) can be actively executing.
- **Many-to-Many Model (or Two-Level Model):**
  - This model multiplexes many user-level threads to a smaller or equal number of kernel threads.
  - It offers a balance between the efficiency of the many-to-one model and the concurrency of the one-to-one model.
  - The operating system can dynamically adjust the number of kernel threads based on the workload and available resources.
  - This model allows for concurrency on multiprocessor systems while minimizing the overhead of creating and managing a large number of kernel threads. It also allows other user threads to run if one performs a blocking system call, provided there are available kernel threads.

7. Define Process. Draw a Process Control Block and explain the information in PCB.

Ans. A program executing as a process is uniquely determined by various parameters. These parameters are stored in a Process Control Block (PCB) i.e. a data structure which holds the following information:

Process Identifier (PID):

- A unique number assigned to each process to identify it in the system.

Process State:

- Indicates the current status of the process (e.g., New, Ready, Running, Waiting, Terminated).

Program Counter:

- Stores the address of the next instruction to be executed by the process.

CPU Registers:

- Includes registers like accumulators, stack pointers, and general-purpose registers used by the process.

Memory Management Information:

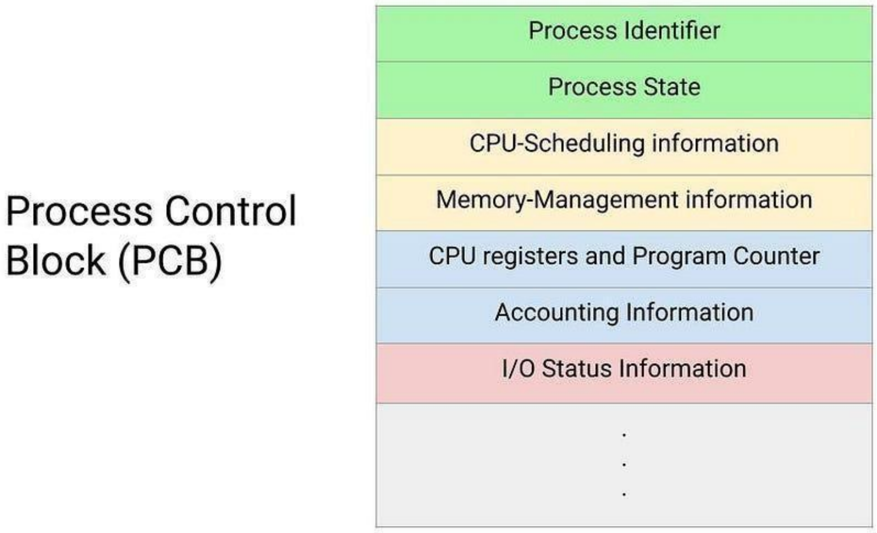
- Contains details about the memory allocated to the process, such as base and limit registers, page tables, or segment tables.

Accounting Information:

- Tracks resource usage, such as CPU time, elapsed time, and memory used.

Scheduling Information:

- Includes the process's priority, scheduling queue pointers, and other parameters.


Process Control Block (PCB)
Process Identifier
Process State
CPU-Scheduling information
Memory-Management information
CPU registers and Program Counter
Accounting Information
I/O Status Information
.
.
.