Introduction to Process Management:

In our computer, there will be several processes that could be running at the same time right (Multiprogramming or multitasking). In this scenario, they might be using the shared resources like memory, and may communicate with each other. Hence, to efficiently manage and coordinate between these processes, we need to have some mechanism which is basically called Process Management.
This is like, if one process is getting busy with IO, we can assign CPU to other processes.

Different Process Management Tasks:

1. **Process Creation and Termination:** Process creation involves creating a Process ID, setting up Process Control Block, etc. A process can be terminated either by the operating system or by the parent process. Process termination involves clearing all resources allocated to it.

2. **CPU Scheduling**: In a multiprogramming system, multiple processes need to get the CPU. It is the job of Operating System to ensure smooth and efficient execution of multiple processes.

3. **Deadlock Handling:** Making sure that system does not reach a state where two or processes cannot proceed due to a cycling dependency on each other.

4. **Inter-Process Communication:** Operating System provides facilities such as shared memory and message passing for cooperating processes to communicate.

5. **Process Synchronization:** Process Synchronization is the coordination of execution of multiple processes in a multiprogramming system to ensure that they access shared resources (like memory) in a controlled and predictable manner.

Process Operations

Process Creation

Process creation in an operating system (OS) is the act of generating a new process. This new process is an instance of a program that can execute independently.

Scheduling

Once a process is ready to run, it enters the "ready queue." The scheduler's job is to pick a process from this queue and start its execution.

Execution

Execution means the CPU starts working on the process. During this time, the process might:

- Move to a waiting queue if it needs to perform an I/O operation.
- Get blocked if a higher-priority process needs the CPU.

Killing the Process

After the process finishes its tasks, the operating system ends it and removes its Process Control Block (PCB).

Context Switching:

The process of saving the current state of one process and loading the context of another process is known as context switching. In simple terms, it is like loading and unloading the process from the running state to the ready state.

When Does Context Switching Happen?

Context Switching Happen:

- When a high-priority process comes to a ready state (i.e. with higher priority than the running process).
- An Interrupt occurs.
- User and kernel-mode switch (It is not necessary though)
- Preemptive CPU scheduling is used.

Importance of Process Management:

1. We can run multiple programs at the same time. For example, listening to music while browsing web.
2. Process Isolation: Ensures different programs don't interfere with each other.
3. Fair Resource Use
4. Smooth Switching: It means saving and loading their states quickly wo keep the system responsive and minimize delays.

Process Scheduling Algorithms

The operating system can use different scheduling algorithms to schedule processes. Here are some commonly used timing algorithms:

- **First-Come, First-Served (FCFS):** This is the simplest scheduling algorithm, where the process is executed on a first-come, first-served basis. FCFS is non-preemptive, which means that once a process starts executing, it continues until it is finished or waiting for I/O.

- **Shortest Job First (SJF):** SJF is a proactive scheduling algorithm that selects the process with the shortest burst time. The burst time is the time a process takes to complete its execution. SJF minimizes the average waiting time of processes.

- **Round Robin (RR):** Round Robin is a proactive scheduling algorithm that reserves a fixed amount of time in a round for each process. If a process does not complete its execution within the specified time, it is blocked and added to the end of the queue. RR ensures fair distribution of CPU time to all processes and avoids starvation.

- **Priority Scheduling:** This scheduling algorithm assigns priority to each process and the process with the highest priority is executed first. Priority can be set based on process type, importance, or resource requirements.

- **Multilevel Queue:** This scheduling algorithm divides the ready queue into several separate queues, each queue having a different priority. Processes are queued based on their priority, and each queue uses its own scheduling algorithm. This scheduling algorithm is useful in scenarios where different types of processes have different priorities.

Frequently Asked Questions on Process Management – FAQs

Why process management is important?

Process management is important in an operating system because it ensures that all the programs running on your computer work smoothly and efficiently.

What is the main difference between process manager and memory manager?

Processes in the system are manage by processor manager and also it is responsible for the sharing of the CPU. whereas, memory in the system is managed by memory manager and it is responsible also for allocation and deallocation of memory, virtual memory management, etc.

What is the difference between a process and a program?

A program is a set of instructions stored on disk (passive), while a process is an instance of the program in execution (active). A single program can be associated with multiple processes.

The time taken to switch between user and kernel modes of execution is t1 while the time taken to switch between two processes is t2. Which of the following is TRUE? (GATE-CS-2011)

(A) t1 > t2

(B) t1 = t2

(C) t1 < t2

(D) nothing can be said about the relation between t1 and t2.

Answer: (C)

Process switching involves a mode switch. Context switching can occur only in kernel mode.