Process Scheduling Queues

Process scheduling is a key component of modern operating systems that allows multiple processes to share the CPU efficiently.

Operating systems use various scheduling algorithms to determine the order in which processes are executed.

Process scheduling queues are data structures used to organize and manage processes in the system.

There are typically several types of queues involved in process scheduling:

Job Queue:

- This queue contains all the processes in the system, regardless of their current state.
- New processes enter this queue when they are created.

Ready Queue:

- Processes in the ready state, meaning they are prepared to execute but are waiting for the CPU, are placed in the ready queue.
- The ready queue is where the scheduler selects the next process to run.

Waiting Queue:

- Processes that are waiting for a particular event or resource, such as I/O completion or the availability of a semaphore, are placed in the waiting queue.
- When the event or resource becomes available, the process is moved back to the ready queue.

Suspended Queue:

- Processes that are temporarily removed from main memory and placed on disk to free up space are kept in the suspended queue.
- These processes can be swapped back into main memory when needed.

Foreground Queue:

- In interactive systems, there may be a separate queue for foreground processes that require user interaction.
- These processes may be given priority to enhance the user experience.

Background Queue:

- Background processes, which typically run with lower priority and do not require immediate user interaction, are placed in the background queue.

Priority Queue:

- Processes are assigned priority levels, and a priority queue is used to organize processes based on their priority.
- The scheduler may select processes from the highest priority queue first.

The scheduler decides which process to execute next based on its scheduling algorithm.

Common scheduling algorithms include First-Come-First-Serve (**FCFS**), Shortest Job Next (**SJN**), Round Robin (**RR**), Priority Scheduling, and Multilevel Queue Scheduling.

The goal of these queues and scheduling algorithms is to maximize CPU utilization, ensure fairness, and provide a responsive and efficient system for users and applications.

Different operating systems may implement variations of these queues and scheduling algorithms based on their design goals and priorities.

Scheduler

A scheduler is a crucial component of an operating system responsible for determining the order in which processes are executed on the CPU.

Its main objective is to allocate CPU time to different processes efficiently, providing a balance between system responsiveness, fairness, and overall system throughput.

There are various types of schedulers, each serving a specific purpose within the operating system.

Here are the main types:

Long-Term Scheduler (Job Scheduler):

- Responsible for selecting processes from the job queue and loading them into the ready queue.
- This scheduler decides which processes should be brought into the ready queue for execution.

Short-Term Scheduler (CPU Scheduler):

- This scheduler selects the next process from the ready queue and allocates the CPU to that process.
- It is responsible for making frequent and fast decisions to keep the CPU busy.

Medium-Term Scheduler:

- In systems that support process suspension (swapping processes in and out of main memory), the medium-term scheduler decides which processes to swap in and out of memory.

- It helps manage the degree of multiprogramming, balancing the number of processes in main memory to optimize system performance.