Compare among all scheduling algorithms [Round robin - Priority - First come first serve]

First Come First Serve (FCFS):

- ✓ This is the simplest scheduling algorithm where processes are executed in the order they arrive.
- ✓ It is non-preemptive, meaning a process keeps the CPU until it completes or voluntarily gives up the CPU.
- ✓ FCFS is fair but may lead to poor average waiting times, especially if long processes arrive first (known as the "convoy effect").
- ✓ It works best for long, CPU-bound processes with similar burst times.

Round Robin (RR):

- ✓ Round Robin is a preemptive scheduling algorithm that assigns a fixed time slice (quantum) to each process in a cyclic manner.
- ✓ It ensures fairness by providing each process an equal opportunity to execute.
- ✓ If a process doesn't complete within its time slice, it is moved to the back of the queue.
- ✓ RR is commonly used in time-sharing systems and is suitable for a mix of short and long processes.
- ✓ However, it can suffer from high context-switching overhead and may have higher waiting times for long processes.

Priority Scheduling:

- ✓ Priority scheduling assigns priorities to processes based on factors like process type, deadline, or priority levels assigned by the system or the user.
- ✓ It can be either preemptive or non-preemptive, where higher priority processes can preempt lower priority ones.

- ✓ Priority scheduling ensures that higher-priority processes are executed first, which is useful for time-critical tasks.
- ✓ However, it may lead to starvation if a low-priority process never gets a chance to execute.
- ✓ Dynamic priority adjustments may be required to prevent starvation and ensure fairness.

Comparison:

Preemptiveness: FCFS is non-preemptive, while RR and Priority scheduling can be either preemptive or non-preemptive, depending on the implementation.

Fairness: RR guarantees fairness by providing equal time slices to each process, while Priority scheduling can prioritize certain processes over others based on their priorities.

Waiting Time: RR usually has shorter waiting times compared to FCFS, but Priority scheduling can have lower waiting times for high-priority processes.

Overhead: RR can have higher context-switching overhead due to frequent preemptions, while FCFS and Priority scheduling have lower overhead in comparison.

In summary, FCFS is simple but may have longer waiting times, RR provides fairness but can incur higher overhead, and Priority scheduling allows for prioritizing processes based on priorities but may lead to starvation. The choice of algorithm depends on the specific system requirements, nature of processes, and desired trade-offs between fairness, waiting times, and overhead.