| **Algorithm** | **Criteria** | **Effect** | **Advantages** | **Disadvantages** |
| --- | --- | --- | --- | --- |
| First-Come, First-Served (FCFS) | Arrival time | Non-preemptive, processes are executed in the order of arrival | Easy to understand and implement | Poor performance for processes with long execution times, can cause Convoy effect |
| Shortest-Job-First (SJF) | Burst time | Non-preemptive or preemptive, processes are executed in order of shortest burst time | Reduces waiting time and average turnaround time, efficient for batch processing | Difficult to predict the exact burst time, not suitable for real-time systems, can cause starvation |
| Priority Scheduling | Priority level | Non-preemptive or preemptive, processes are executed in order of priority level | Allows high-priority processes to execute first, suitable for real-time systems | Low-priority processes may never get executed, can cause starvation |
| Round-Robin (RR) | Time quantum | Preemptive, each process is executed for a fixed time quantum | Fair share of CPU time for all processes, suitable for time-sharing systems | High context-switching overhead, performance depends on time quantum value |
| Multilevel Queue | Different criteria for each queue | Non-preemptive or preemptive, processes are assigned to different queues based on their characteristics | Provides separate queues for different types of processes, improves throughput | Complex to implement and maintain, can cause starvation |
| Multilevel Feedback Queue | Different criteria for each queue and feedback mechanism | Preemptive, processes are assigned to different queues based on their characteristics and can move between queues based on their behavior | Provides dynamic priority adjustment and feedback mechanism, improves responsiveness | Complex to implement and maintain, can cause starvation |
| Guaranteed Scheduling | Guaranteed CPU time | Preemptive, each process is guaranteed a minimum amount of CPU time | Provides fairness and response time guarantees, suitable for real-time systems | Overhead of guaranteeing CPU time, can cause low CPU utilization |
| Lottery Scheduling | Lottery tickets | Preemptive, processes are assigned lottery tickets and CPU time is allocated based on the number of tickets | Provides probabilistic fairness and avoids starvation, suitable for resource allocation problems | Overhead of ticket allocation and bookkeeping |