**Difference between scheduling algorithms and when to use each of them**

Scheduling algorithms are used in operating systems to determine the order in which processes or tasks should be executed on a computer system. Different scheduling algorithms have different criteria and priorities for making scheduling decisions. Here are some commonly used scheduling algorithms and when to use each of them:

1. First-Come, First-Served (FCFS): This algorithm schedules processes in the order they arrive. It is simple and easy to implement but may suffer from poor average waiting times, especially if long-running processes arrive first. FCFS can be suitable for scenarios where fairness or simplicity is more important than efficiency.
2. Shortest Job Next (SJN) or Shortest Job First (SJF): This algorithm schedules the process with the shortest burst time first. It aims to minimize the average waiting time and provides optimal results in terms of minimizing the total time spent in the system. SJN is suitable when the burst time of processes is known in advance, but in practice, it is difficult to predict accurately.
3. Round Robin (RR): RR is a time-sharing scheduling algorithm that assigns fixed time slices, called time quantum, to each process in a circular manner. It ensures fairness by allowing each process to have a turn, but it may have high context switching overhead and may not be suitable for processes with varying burst times.
4. Priority Scheduling: This algorithm assigns priorities to processes and schedules them based on their priority levels. Higher priority processes are executed first. It can be either preemptive (allowing higher priority processes to interrupt lower priority ones) or non-preemptive (allowing a process to complete its execution). Priority scheduling is useful when different processes have varying levels of importance or urgency.
5. Multilevel Queue Scheduling: This algorithm divides processes into multiple queues, each with a different priority level. Processes are scheduled based on their priority, and each queue may use a different scheduling algorithm. It is useful when there are different classes or types of processes with different scheduling requirements.
6. Multilevel Feedback Queue Scheduling: Similar to multilevel queue scheduling, this algorithm assigns priorities to processes but allows them to move between different queues based on their behavior. For example, a process that uses a lot of CPU time may be moved to a lower priority queue. It provides flexibility and can handle both CPU-bound and I/O-bound processes effectively.

The choice of scheduling algorithm depends on various factors such as system requirements, workload characteristics, and performance goals. No single algorithm is suitable for all scenarios, so it is important to analyze the specific requirements and constraints of the system to select an appropriate scheduling algorithm.