Task6(session7)

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

Algorithm	Description	Advantages	disAdvantages
Round robin	Each process is given a fixed amount of time to run, called a time quantum. Once the time quantum expires, the process is preempted and the next process in the queue is given a time quantum.	Fair to all processes. - Prevents starvation.	- Overhead of context switching. - May not be suitable for real-time systems.
Priority	Processes are scheduled based on their priority. Higher priority processes are scheduled first.	Can be used to ensure that important processes are always running. - Can be used to implement real-time systems.	- Lower priority processes may starve. - Difficult to assign priorities fairly.
First come, first served	Processes are scheduled in the order in which they arrive.	Simple to implement. - Fair to all processes.	- May lead to starvation of long-running processes. - May not be suitable for real-time systems.

Which algorithm is best?

The best scheduling algorithm for a particular system depends on the specific requirements of that system. For example, if the system needs to be fair to all processes, then a round robin algorithm may be a good choice. If the system needs to ensure that important processes are always running, then a priority algorithm may be a better choice. And if the system needs to be simple to implement, then a first come, first served algorithm may be the best option.

Here are some examples of when each scheduling algorithm might be used:

Round robin: Round robin scheduling is often used in time-sharing systems, where multiple users are sharing the same CPU. This is because round robin scheduling is fair to all processes and prevents starvation.

Priority: Priority scheduling is often used in real-time systems, where certain tasks must be completed within a certain amount of time. This is because priority scheduling can be used to ensure that the most important tasks are always running.

First come, first served: First come, first served scheduling is often used in simple systems, such as batch processing systems. This is because first come, first served scheduling is easy to implement and does not require any special knowledge of the processes.