Round Robin scheduling algorithm

Round Robin scheduling algorithm is one of the most popular scheduling algorithm which can actually be implemented in most of the operating systems. This is the preemptive version of first come first serve scheduling. The Algorithm focuses on Time Sharing.

In this algorithm, every process gets executed in a cyclic way. A certain time slice is defined in the system which is called time quantum. Each process present in the ready queue is assigned the CPU for that time quantum, if the execution of the process is completed during that time then the process will terminate else the process will go back to the ready queue and waits for the next turn to complete the execution.

Advantages

- It can be actually implementable in the system because it is not depending on the burst time.
- It doesn't suffer from the problem of starvation or convoy effect.
- All the jobs get a fare allocation of CPU.
- This round robin algorithm offers starvation-free execution of processes.
- Each process gets equal priority and fair allocation of CPU.
- Round Robin scheduling algorithm enables the Context switching method to save the states of preempted processes.
- It is easily implementable on the system because round robin scheduling in os doesn’t depend upon burst time.

Disadvantages

- The higher the time quantum, the higher the response time in the system.
- The lower the time quantum, the higher the context switching overhead in the system.
- Deciding a perfect time quantum is really a very difficult task in the system.

How does the Round Robin Algorithm Work?

- All the processes are added to the ready queue.
- At first, The burst time of every process is compared to the time quantum of the CPU.
- If the burst time of the process is less than or equal to the time quantum in the round robin scheduling algorithm, the process is executed to its burst time.
- If the burst time of the process is greater than the time quantum, the process is executed up to the time quantum (TQ).
- When the time quantum expires, it checks if the process is executed completely or not.
- On completion, the process terminates. Else, it goes back again in the ready state.