Disk Scheduling

- **Disk scheduling :** It is done by operating systems to schedule I/O requests arriving for the disk. Disk scheduling is also known as I/O scheduling.
- Disk scheduling is important because :
    - Multiple I/O requests may arrive by different processes and only one I/O request can be served at a time by the disk controller. Thus other I/O requests need to wait in the waiting queue and need to be scheduled.
    - Two or more request may be far from each other so can result in greater disk arm movement.
    - Hard drives are one of the slowest parts of the computer system and thus need to be accessed in an efficient manner.
- Some Important Terms:
    - **Seek time:** Seek time is the time taken to locate the disk arm to a specified track where the data is to be read or write. So the disk scheduling algorithm that gives minimum average seek time is better.
    - **Rotational Latency:** Rotational Latency is the time taken by the desired sector of disk to rotate into a position so that it can access the read/write heads. So the disk scheduling algorithm that gives minimum rotational latency is better.
    - **Transfer time:** It is the time to transfer the data. It depends on the rotating speed of the disk and number of bytes to be transferred.

    - Disk Access Time = Seek Time + Rotational Latency + Transfer Time.
- **Disk Response Time:** Response Time is the average of time spent by a request waiting to perform its I/O operation.

FCFS

- Simplest CPU scheduling algorithm that schedules according to arrival times of processes.
- First come first serve scheduling algorithm states that the process that requests the CPU first is allocated the CPU first. It is implemented by using the FIFO queue.
- When a process enters the ready queue, its PCB is linked to the tail of the queue.
- When the CPU is free, it is allocated to the process at the head of the queue. The running process is then removed from the queue.
- FCFS is a non-preemptive scheduling algorithm.
- Characteristics:
    - FCFS supports non-preemptive and preemptive CPU scheduling algorithms.
    - This algorithm is not much efficient in performance, and the wait time is quite high.