Process Scheduling

1. Process Scheduling

- Process scheduling determines the order in which processes are executed in a system.
- Types of scheduling:
    - **Long-term scheduling** – Decides which processes are admitted into the system.
    - **Short-term scheduling** – Selects processes from the ready queue for execution.
    - **Medium-term scheduling** – Temporarily suspends processes to free resources.
- Scheduling algorithms impact performance, responsiveness, and efficiency.

---

2. Non-Preemptive Scheduling

- Once a process is scheduled, it runs until completion or voluntarily releases the CPU.
- No forceful interruption occurs.
- Examples: **First-Come-First-Served (FCFS), Shortest Process Next (SPN)**.

---

3. First-Come-First-Served (FCFS)

- **Non-preemptive** scheduling algorithm.
- The first process to request CPU gets executed first.
- Implemented using a **FIFO (First-In-First-Out) queue**.
- **Advantages**: Simple, fair execution order.
- **Disadvantages**:
    - **Convoy Effect**: Short processes may have to wait for long ones.
    - **Favors CPU-bound processes**: I/O processes wait longer.

---

4. Round Robin (RR) Scheduling

- **Preemptive** scheduling based on a **time quantum** (time slice).
- Each process gets CPU for a fixed time; if not completed, it moves to the back of the queue.
- **Advantages**:
    - Ensures fairness among processes.
    - Shorter processes finish faster.
- **Limitations**:
    - **Small quantum** → More **context switches** → High overhead.
    - **Large quantum** → Acts like **FCFS**, causing delays.
    - **I/O-bound processes suffer** as CPU-bound processes get more CPU time.

---

5. Virtual Round Robin (VRR)

- Improves RR by introducing an **auxiliary queue (FCFS)** for I/O-bound processes.
- **Key Features**:
    - I/O-bound processes get priority in scheduling after I/O completion.
    - Reduces waiting time for interactive processes.
- **Advantage**: Balances CPU-bound and I/O-bound workloads efficiently.

---

6. Shortest Process Next (SPN) / Shortest Job First (SJF)

- **Non-preemptive** scheduling algorithm.
- Process with the **shortest expected execution time** is selected next.
- **Advantages**:
    - Minimizes average waiting time.
    - Prioritizes short jobs.
- **Disadvantages**:
    - **Starvation**: Long processes may never get scheduled.
    - Requires accurate estimation of execution time.

---

7. Shortest Remaining Time (SRT)

- **Preemptive** version of SJF
- Always selects the process with the **shortest remaining execution time**.
- **Advantages**:
    - Low turnaround time.
    - More efficient than FCFS and SPN.
- **Disadvantages**:

    - High overhead due to frequent context switches.
    - Starvation of long processes.

---

9. Feedback Scheduling (Multilevel Feedback Queue - MLFQ)

- **Preemptive, priority-based scheduling** that penalizes long-running jobs.
- Process priority **reduces as it consumes more CPU time**.
- **Mechanism**:
    - **New processes** start in the **highest-priority queue** (RQ0).
    - If a process is not completed in its time slice, it moves **down to the next queue**.
    - **Lower priority queues use FCFS**, while higher queues use **RR**.
- **Problem**: Long processes may suffer starvation.
- **Solution**: Assign different time quanta for each queue

  (e.g., **RQ0 → 1 unit, RQ1 → 2 units, RQ2 → 4 units, etc.**).
