Operating Systems

Course Code: **71203002004**

Scheduling 4

by -

Minal Rajwar

Demand Scheduling

- It is a type of CPU scheduling approach where tasks are allocated CPU only when they explicitly demand or request processor time.

- This contrast with preemptive scheduling where the OS can interrupt running process.

- Most appropriate for batch processing systems where turnaround time is more important than response time.

Key Characteristics of Demand Scheduling

- **Non-preemptive Nature:** Once a process gets the CPU, it keeps it until it voluntarily releases the CPU (either by termination or waiting for I/O).
- **Process Initiated:** The scheduling decisions are triggered by process demands rather than system interrupts.
- **Simple Implementation:** Generally easier to implement than preemptive scheduling algorithms.

Common Examples

- **FCFS (First Come First Serve) :** The simplest demand scheduling algorithm where processes are executed in order they arrive.
- **Shortest Job Next (SJN) :** The process with the smallest execution time is selected next from the ready queue.

Advantages	Disadvantages
Low Scheduling Overhead	Poor for interactive systems as processes can monopolize CPU
No context switching until process voluntarily yields CPU	May lead to convoy effect (short processes stuck behind large ones)
Predictable behaviour	Not optimal for system requiring responsiveness

Real Time Scheduling

- Designed to process with strict timing constraints.
- Correctness depends not only on logical results but time at which it is delivered.
- Classified as:
  - Hard Real Time (missed deadline causes system failure)
  - Soft Real Time (deadlines can be missed and is not very critical)

Classification of Real Time Scheduling Algorithms

Based on schedulability analysis, implementation (static/dynamic), and result (self/dependent), real-time scheduling algorithms are categorized as:

1. **Static Table Driven Scheduling:**
   a. Performs offline(static) schedulability analysis.
   b. Generates a fixed schedule before runtime.
   c. Is a predictable, deterministic system (e.g., automotive control).
   d. **Example:** Cyclic Executive Scheduling

2. **Static Priority Driven Preemptive approaches:**

    a. Assigns fixed priorities to tasks.
    b. Higher priority tasks preempt lower-priority ones.
    c. Used in system requiring predictable behavior (e.g., medical devices).
    d. **Example:** Rate Monotonic Scheduling (RMS) - Assigns priority based on task frequency (shorter periods = higher priority)

3. Dynamic Planning Based Scheduling

    a. Dynamically checks task feasibility at runtime.
    b. Admits a task only if it can meet its deadline.
    c. Used in systems with variable workloads (multimedia streaming).
    d. **Example:** Earliest Deadline First (EDF) - Dynamically schedules tasks based on closest deadlines.

`

4.. **Dynamic Best Effort Approaches:**

a. No guarantee of meeting deadlines.
b. Tasks are aborted if deadlines are missed.
c. Used in Soft real time systems (e.g., video conferencing, gaming).
d. **Example**: Least Slack time (LST) - Prioritizes task with the least remaining time before deadline.

Advantages of Real Time Scheduling

a. **Timing Guarantees :** ensures critical tasks meet deadlines.
b. **Resource Optimization :** Efficient CPU and memory utilization.
c. **Priority Based Execution :** High Priority tasks get immediate attention.
d. **Predictability:** Enbales worst case execution time (WCET) analysis.
e. **Control Over Execution:** Allows fine tuning of task priorities and dependencies.

Disadvantages of Real Time Scheduling

a. **Increased Complexity:** Requires careful design and analysis.
b. **Overhead:** Context switching and scheduling decisions add latency.
c. **Resource Constraints:** Limited CPU/memory may hinder schedulability.
d. **Verification Challenges:** Proving all deadlines are met is difficult.
e. **Scalability Issues:** some algorithms don’t scale well with increasing tasks.

DISCUSSION & REVISION

1. What type of real-time task cannot miss its deadline?
2. Which Scheduling method allows tasks to be interrupted?
3. What is the main goal of a real time scheduler?
4. Which algorithm assigns priority based on task frequency?
5. What type of scheduling checks task feasibility at runtime?