Optimality

A scheduling algorithm $S$ is optimal if
- a system is not schedulable under $S$ → it is not schedulable under any other scheduling algorithms in the same category

Optimal Scheduling Algorithms

- Rate Monotonic Scheduling (RMS)
  - priority = rate = $1/\text{period}$
  - optimal preemptive fixed-priority scheduling algorithm
- Earliest Deadline First (EDF)
  - priority = absolute deadline
  - optimal preemptive dynamic-priority scheduling algorithm

Basic RMS/EDF Analysis

Assumptions

- Optimality and analysis under assumptions:
  - Single processor.
  - All tasks are periodic.
  - Relative deadline = period
  - No blocking (e.g., to prevent race conditions)
  - Zero context switch time.
- Both RMS and EDF have been extended to relax assumptions

Schedulable Utilization Bound

- CPU utilization of a processor:
  \[ U = \sum_{T \in S} \frac{c_j}{p_j} \]

  where $S$ is the set of all tasks
- Schedulable utilization bound $U_b$: All tasks are guaranteed to be schedulable if $U \leq U_b$

Necessary Condition

- No scheduling algorithms can guarantee schedulability if $U > 1$
  - $U_b \leq 1$
  - An algorithm must be optimal if its $U_b = 1$

RMS Utilization Bound

- The schedulable utilization bound of RMS for $n$ processes is: $U_b(n) = n(2^{1/n} - 1)$
  - $U_b(2) = 0.828$
  - $U_b(3) = 0.780$
  - $U_b(n) \geq U_b(\infty) = \ln 2 = 0.693$
  - $U_b(n) \leq U_b$ is a sufficient condition, but not necessary in general.
  - $U_b = 1$ if all process periods are harmonic, i.e., periods are multiples of each other
    - E.g., 2, 4, 8, 16
Response Time Analysis

- Assume fixed-priority scheduling
- Critical instant: scheduling state that gives worst response time.
- Occurs when all higher-priority processes are ready to execute.

Response Time Analysis for Fixed-Priority Sched

/* Tasks are ordered by priority; T1 has the highest priority */
for (each task Tj) {
    I = 0;
    do {
        R = I + cj;
        if (R > dj) return (UNSCHEDULABLE)
        I = \sum_{i=1}^{j-1} \left\lfloor \frac{R}{P_i} \right\rfloor \cdot P_i;
    } while (I + cj > R)
    return (SCHEDULABLE)
}

RMS

- RMS may not guarantee schedulability even when CPU is not fully utilized
- Low overhead: When tasks are known a priori, priorities are never changed

EDF Utilization Bound

- Schedulable utilization bound of EDF: \( U_0 \leq 1 \)
- \( U \leq U_0 \) is sufficient and necessary for schedulability.

EDF

- EDF can guarantee schedulability as long as CPU is not fully utilized
- Higher overhead than RMS: Task priorities may need to be changed online

Assumptions of RMS and EDF

- Optimality and analysis under assumptions:
  - Single processor.
  - All tasks are periodic.
  - Relative deadline = period
  - No blocking (e.g., to prevent race conditions)
  - Zero context switch time.
- What if Deadline < Period?
Optimal Fixed-Priority Algorithm

- Deadline Monotonic Scheduling (DMS)
  - priority = (relative) deadline
  - optimal fixed priority scheduling when deadline \( \leq \) period
- RMS response time analysis also applies to DMS
  - but priority ordering may not be the same!