Real-Time Scheduling

CIS700

Insup Lee
Outline

• Real-time systems

• Real-time scheduling algorithms
  – Fixed-priority algorithm (RM)
  – Dynamic-priority algorithm (EDF)
Real-Time Systems

- **Definition**
  - Systems whose correctness depends on their *temporal* aspects as well as their *functional* aspects

- **Performance measure**
  - *Timeliness* on timing constraints (deadlines)
  - Speed/average case performance are less significant.

- **Key property**
  - *Predictability* on timing constraints
Real-Time System Example

- Digital control systems
  - periodically performs the following job:
    - senses the system status and
    - actuates the system according to its current status
Real-Time System Example

- Multimedia applications
  - periodically performs the following job:
    
    reads, decompresses, and displays video and audio streams
Fundamental Real-Time Issue

- To specify the timing constraints of real-time systems

- To achieve predictability on satisfying their timing constraints, possibly, with the existence of other real-time systems
Scheduling Framework Example

- Digital Controller
- Multimedia

OS Scheduler

CPU
Real-Time Workload

- Job (unit of work)
  - a computation, a file read, a message transmission, etc
- Attributes
  - Resources required to make progress
  - Timing parameters
Real-Time Task

- **Task**: a sequence of similar jobs
  - **Periodic task** \((p,e)\)
    - Its jobs repeat regularly
    - Period \(p =\) inter-release time \((0 < p)\)
    - Execution time \(e =\) maximum execution time \((0 < e < p)\)
    - Utilization \(U = e/p\)
Deadlines: Hard vs. Soft

• **Hard** deadline
  - Disastrous or very serious consequences may occur if the deadline is missed
  - Validation is essential: can all the deadlines be met, even under worst-case scenario?
  - Deterministic guarantees

• **Soft** deadline
  - Ideally, the deadline should be met for maximum performance. The performance degrades in case of deadline misses.
  - Best effort approaches / statistical guarantees
Schedulability

- Property indicating whether a real-time system (a set of real-time tasks) can meet their deadlines
Real-Time Scheduling

- Determines the order of real-time task executions
- Static-priority scheduling
- Dynamic-priority scheduling
RM (Rate Monotonic)

- Optimal static-priority scheduling
- It assigns priority according to period
- A task with a shorter period has a higher priority
- Executes a job with the shortest period
RM (Rate Monotonic)

- Executes a job with the shortest period

<table>
<thead>
<tr>
<th>Job</th>
<th>Period</th>
<th>Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>T₁ (4,1)</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>T₂ (5,2)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>T₃ (7,2)</td>
<td>7</td>
<td>2</td>
</tr>
</tbody>
</table>
RM (Rate Monotonic)

- Executes a job with the shortest period

![Diagram showing deadlines and missed deadlines]
Response Time

- Response time
  - Duration from released time to finish time

$T_1(4,1)$

$T_2(5,2)$

$T_3(10,2)$
Response Time

- Response time
  - Duration from released time to finish time

\[ T_1(4,1) \]
\[ T_2(5,2) \]
\[ T_3(10,2) \]
Response Time

- **Response Time** ($r_i$) [Audsley et al., 1993]

$$r_i = e_i + \sum_{T_k \in HP(T_i)} \left[ \frac{r_i}{p_k} \right] \cdot e_k$$

- **HP($T_i$)**: a set of higher-priority tasks than $T_i$
RM - Schedulability Analysis

• Real-time system is schedulable under RM if and only if $r_i \leq p_i$ for all task $T_i(p_i,e_i)$

Joseph & Pandya,
“Finding response times in a real-time system”,
RM – Utilization Bound

• Real-time system is schedulable under RM if
  \[\sum U_i \leq n \left(2^{1/n}-1\right)\]

Liu & Layland,
RM – Utilization Bound

• Real-time system is schedulable under RM if
  \[ \sum U_i \leq n \left( 2^{1/n} - 1 \right) \]

• Example: \(T_1(4,1), T_2(5,1), T_3(10,1),\)

  \[ \sum U_i = \frac{1}{4} + \frac{1}{5} + \frac{1}{10} \]
  \[ = 0.55 \]
  \[ 3 \left( 2^{1/3} - 1 \right) \approx 0.78 \]

  Thus, \(\{T_1, T_2, T_3\}\) is schedulable under RM.
RM – Utilization Bound

• Real-time system is schedulable under RM if
  \[ \sum U_i \leq n \left(2^{1/n} - 1\right) \]
EDF (Earliest Deadline First)

- Optimal dynamic priority scheduling
- A task with a shorter deadline has a higher priority
- Executes a job with the earliest deadline
EDF (Earliest Deadline First)

- Executes a job with the earliest deadline
EDF (Earliest Deadline First)

• Executes a job with the earliest deadline
EDF (Earliest Deadline First)

- Executes a job with the earliest deadline

Diagram showing three tasks:
- $T_1(4,1)$
- $T_2(5,2)$
- $T_3(7,2)$
EDF (Earliest Deadline First)

- Optimal scheduling algorithm
  - if there is a schedule for a set of real-time tasks, EDF can schedule it.
Processor Demand Bound

- **Demand Bound Function:** \( dbf(t) \)
  - the **maximum processor demand** by workload over any interval of length \( t \)
EDF - Schedulability Analysis

• Real-time system is schedulable under EDF if and only if $\text{dbf}(t) \leq t$ for all interval $t$

Baruah et al.

• Demand Bound Function: $\text{dbf}(t)$
  - the maximum processor demand by workload over any interval of length $t$
EDF – Utilization Bound

• Real-time system is schedulable under EDF if and only if

$$\sum U_i \leq 1$$

Liu & Layland,
EDF – Overload Conditions

- Domino effect during overload conditions
  - Example: $T_1(4,3), T_2(5,3), T_3(6,3), T_4(7,3)$

Deadline Miss!

Better schedules:

October 3, 2005  CIS 700
RM vs. EDF

• **Rate Monotonic**
  – Simpler implementation, even in systems without explicit support for timing constraints (periods, deadlines)
  – Predictability for the highest priority tasks

• **EDF**
  – Full processor utilization
  – Misbehavior during overload conditions

• For more details: Buttazzo, “Rate monotonic vs. EDF: Judgement Day”, EMSOFT 2003.