Rate Monotonic Analysis

Introduction

Periodic tasks

Extending basic theory

Synchronization and priority inversion

Aperiodic servers

Case study: BSY-1 Trainer
Sample Problem: Synchronization

\[ \tau_2 \text{'s deadline is } 20 \text{ msec before the end of each period.} \]
Priority Inversion in Synchronization

\[ \tau_1: \{ \ldots P(S1) \ldots V(S1) \ldots \} \]

\[ \tau_3: \{ \ldots P(S1) \ldots V(S1) \ldots \} \]

Legend

- Critical section (S1 locked)
- Executing
- Blocked

Time

\( \tau_1(H) \)

\( \tau_2(M) \)

\( \tau_3(L) \)
Priority Inversion

Delay to a task’s execution caused by interference from lower priority tasks is known as priority inversion.

Priority inversion is modeled by blocking time.

Identifying and evaluating the effect of sources of priority inversion is important in schedulability analysis.
Sources of Priority Inversion

Synchronization and mutual exclusion

Non-preemptable regions of code

FIFO (first-in-first-out) queues
Accounting for Priority Inversion

Recall that task schedulability is affected by

- preemption: two types of preemption
  - can occur several times per period
  - can only occur once per period
- execution: once per period
- blocking: at most once per period for each source

The schedulability formulas are modified to add a “blocking” or “priority inversion” term to account for inversion effects.
UB Test with Blocking

Include blocking while calculating effective utilization for each tasks:

\[
f_i = \sum_{j \in H_n} \frac{C_j}{T_j} + \frac{C_i}{T_i} + \frac{B_i}{T_i} + \frac{1}{T_{i_k}} \sum_{k \in H_1} C_k
\]

- \(H_n\) Preemption (can hit \(n\) times)
- Execution
- Blocking
- \(H_1\) Preemption (can hit once)
RT Test with Blocking

Blocking is also included in the RT test:

\[ a_{n+1} = B_i + C_i + \sum_{j=1}^{i-1} \left[ \frac{a_n}{T_j} \right] C_j \]

where \( a_0 = B_i + \sum_{j=1}^{i} C_j \)

Perform test as before, including blocking effect.
Example: Considering Blocking

Consider the following example:

Periodics tasks

- $\tau_1$: 25 msec
- $\tau_2$: 50 msec
- $\tau_3$: 300 msec

Data Structure

- 100 msec
- 200 msec
- 10 msec
- 30 msec

What is the worst-case blocking effect (priority inversion) experienced by each task?
Example: Adding Blocking

Task $\tau_2$ does not use the data structure. Task $\tau_2$ experiences no priority inversion.

Task $\tau_1$ shares the data structure with $\tau_3$. Task $\tau_1$ could have to wait for $\tau_3$ to complete its critical section. But worse, if $\tau_2$ preempts while $\tau_1$ is waiting for the data structure, $\tau_1$ could have to wait for $\tau_2$’s entire computation.

This is the resulting table:

<table>
<thead>
<tr>
<th>Task</th>
<th>Period</th>
<th>Execution Time</th>
<th>Priority</th>
<th>Blocking Delays</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>100</td>
<td>25</td>
<td>High</td>
<td>30+50</td>
<td>100</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>200</td>
<td>50</td>
<td>Medium</td>
<td>0</td>
<td>200</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>300</td>
<td>100</td>
<td>Low</td>
<td>0</td>
<td>300</td>
</tr>
</tbody>
</table>
UB Test for Example

Recall UB test with blocking:

\[
f_i = \sum_{j \in H_n} \frac{C_j}{T_j} + \frac{C_i}{T_i} + \frac{B_i}{T_i} + \frac{1}{T_i} \sum_{i_k \in H_1} C_k
\]

\[
f_1 = \frac{C_1}{T_1} + \frac{B_1}{T_1} = \frac{25}{100} + \frac{80}{100} = 1.05 > 1.00 \quad \text{Not schedulable}
\]

\[
f_2 = \frac{C_1}{T_1} + \frac{C_2}{T_2} = \frac{25}{100} + \frac{50}{200} = 0.50 < U \ (2)
\]

\[
f_3 = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} = \frac{25}{100} + \frac{50}{200} + \frac{100}{300} = 0.84 > U \ (3)
\]

RT test shows \( \tau_3 \) is schedulable
Synchronization Protocols

No preemption

Basic priority inheritance

Highest locker’s priority

Priority ceiling

Each protocol prevents unbounded priority inversion.
Nonpreemption Protocol

\[ \tau_2: \{...P(S1)...V(S1)...\} \]

\[ \tau_4: \{...P(S1)...V(S1)...\} \]

Legend
- S1 locked
- Executing
- Blocked

Time

Synchronization & Priority Inversion
Basic Inheritance Protocol (BIP)

\[ \tau_2: \{...P(S1)...V(S1)...\} \]

\[ \tau_4: \{...P(S1)...V(S1)...\} \]

Legend

- S1 locked
- Executing
- Blocked

Time

S1 locked

Attempts to lock S1

S1 locked

S1 unlocked

B
Highest Locker’s Priority Protocol

\( \tau_2: \{ \ldots P(S1) \ldots V(S1) \ldots \} \)

\( \tau_4: \{ \ldots P(S1) \ldots V(S1) \ldots \} \)

Legend

- S1 locked
- Executing
- Blocked

\( \tau_1(H) \)

\( \tau_2 \)

\( \tau_3 \)

\( \tau_4(L) \)

Time
Priority Ceiling Protocol (PCP)

\[ \tau_2 : \{ \ldots P(S1) \ldots V(S1) \ldots \} \]

\[ \tau_4 : \{ \ldots P(S1) \ldots V(S1) \ldots \} \]

Legend
- S1 locked
- Executing
- Blocked

\[ \tau_1(H) \]

Attempts to lock S1

S1 locked

S1 unlocked

\[ \tau_2 \]

Ready

\[ \tau_3 \]

Ready

\[ \tau_4(L) \]

S1 locked

S1 unlocked

Time
Example Of Chained Blocking (BIP)

\[ \tau_1: \{...P(S1)...P(S2)...V(S2)...V(S1)...\} \]

\[ \tau_2: \{...P(S1)...V(S1)...\} \]

\[ \tau_3: \{...P(S2)...V(S2)...\} \]

Legend

- S1 locked
- S2 locked
- Blocked

Attempts to lock S1 (blocked)

Attempts to lock S2 (blocked)

S1 locked

S1 unlocked

S2 locked

S2 unlocked

Time

0 1 2 3 4 5 6 7 8 9 10 11 12 13

Blocked B
Blocked At Most Once (PCP)

\[ \tau_1: \{ ... P(S1) ... P(S2) ... V(S2) ... V(S1) ... \} \]
\[ \tau_2: \{ ... P(S1) ... V(S1) ... \} \]
\[ \tau_3: \{ ... P(S2) ... V(S2) ... \} \]

Legend
- S1 locked
- S2 locked
- Ceiling

Time
0 1 2 3 4 5 6 7 8 9 10 11 12 13

S1 locked
S2 locked
Ceiling C

Attempts to lock S1 (blocked)
S1 locked
S1 unlocked
S2 locked
S2 unlocked

\( \tau_1(H) \)
\( \tau_2(M) \)
\( \tau_3(L) \)
Deadlock: Using BIP

\[ \tau_1 : \{ \ldots \text{P}(S1) \ldots \text{P}(S2) \ldots \text{V}(S2) \ldots \text{V}(S1) \ldots \} \]

\[ \tau_2 : \{ \ldots \text{P}(S2) \ldots \text{P}(S1) \ldots \text{V}(S1) \ldots \text{V}(S2) \ldots \} \]

Legend:

- S1 locked
- S2 locked
- Blocked

\[ \tau_1(H) \]

- Locks S1
- Attempts to lock S2 (blocked)

\[ \tau_2(M) \]

- S2 locked
- Attempts to lock S1 (blocked)

Time

0 1 2 3 4 5 6 7 8 9 10 11 12 13
Deadlock Avoidance: Using PCP

$\tau_1$:{...P(S1)...P(S2)...V(S2)...V(S1)...}

$\tau_2$:{...P(S2)...P(S1)...V(S1)...V(S2)...}

Legend

S1 locked
S2 locked
Ceiling

Time

0 1 2 3 4 5 6 7 8 9 10 11 12 13

Synchronization & Priority Inversion
# Summary of Synchronization Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Bounded Priority Inversion</th>
<th>Blocked at Most Once</th>
<th>Deadlock Avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonpreemptible critical sections</td>
<td>Yes</td>
<td>Yes(^1)</td>
<td>Yes(^1)</td>
</tr>
<tr>
<td>Highest locker’s priority</td>
<td>Yes</td>
<td>Yes(^1)</td>
<td>Yes(^1)</td>
</tr>
<tr>
<td>Basic inheritance</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Priority ceiling</td>
<td>Yes</td>
<td>Yes(^2)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

\(^1\) Only if tasks do not suspend within critical sections

\(^2\) PCP is not affected if tasks suspend within critical sections
Sample Problem with Synchronization

When basic priority inheritance protocol is used:

<table>
<thead>
<tr>
<th>Task</th>
<th>Period</th>
<th>Execution Time</th>
<th>Priority</th>
<th>Blocking Delays</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>100</td>
<td>20</td>
<td>High</td>
<td>20+10</td>
<td>100</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>150</td>
<td>40</td>
<td>Medium</td>
<td>10</td>
<td>130</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>350</td>
<td>100</td>
<td>Low</td>
<td>0</td>
<td>350</td>
</tr>
</tbody>
</table>
UB Test for Sample Problem

This format is sometimes called a schedulability model for the task set:

$$f_1 = \frac{C_1}{T_1} + \frac{B_1}{T_1} = \frac{20}{100} + \frac{30}{100} = 0.500 < U (1)$$

$$f_2 = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{B_2}{T_2} = \frac{20}{100} + \frac{40}{150} + \frac{10}{150} = 0.534 < 0.729$$

$$U (2, .80) = 0.729$$

$$f_3 = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} = \frac{20}{100} + \frac{40}{150} + \frac{100}{350} = 0.753 < U (3)$$