Vrije Universiteit Brussel

Rate-Monotonic Scheduling

Filip Moons
Master in Applied Computer Science
Promotor: Prof. Dr. Martin Timmerman
Presentation Operating Systems & Security

Friday 9 January, 2014

Content

- Tasks
- The algorithm
- Tests
- Example

Some assumptions

Definition

A task τ_i is a process that has:

- To be periodically executed in a period T_i
- The worst case execution time C_i
- A deadline D_i, which is the available time on the processor. $D_i = T_i$

Some assumptions

Definition

A task τ_i is a process that has:

- To be periodically executed in a period T_i
- The worst case execution time C_i
- A deadline D_i, which is the available time on the processor. $D_i = T_i$

The algorithm

Rate-Monotonic Scheduling: Algorithm

1. The task with the smallest period has the highest priority,
2. A higher-priority task ready to be executed, overrides the current executed task. The current executed task is is interrupted and may resume afterwards.

An example

1. τ_1: $T_1 = 4$ ms, $C_1 = 1$ ms
2. τ_2: $T_2 = 5$ ms, $C_2 = 2$ ms
3. τ_3: $T_3 = 7$ ms, $C_1 = 2$ ms

Schedulability test 1

The utilization factor

The utilization factor of a task set $\tau_1, \tau_2, ..., \tau_n$ is:

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i}$$

$\frac{C_i}{T_i}$ gives the utilization of task τ_i on the CPU

Schedulability test 1: Liu & Layland lower bound

With n-tasks with, a schedule exists as:

$$U \leq n(2^{\frac{1}{n}} - 1)$$

For $n \rightarrow \infty$, we get: $\lim_{n\rightarrow\infty} n(2^{\frac{1}{n}} - 1) = \ln 2 \approx 0.693147...$

Schedulability test 2

The RT-test

A task set can be scheduled by RMS if the deadline of the first execution of each task is met when using the scheduling algorithm starting all tasks at the same time.

Total processing requirement

The total processing requirement $u_i(t)$ of a task τ_i in the time interval $[0, t]$ is given by, with $0 < t \leq T_i$:

$$u_i(t) = C_i + \sum_{k=1}^{i-1} \left\lceil \frac{t}{T_i} \right\rceil C_k \tag{1}$$

The idea is immediately clear: if $u_i(t) \leq t$ for some $t \leq T_i$ then task τ_i is schedulable.

An example

1. τ_1: $T_1 = 4$ ms, $C_1 = 1$ ms $\rightarrow \frac{C_1}{T_1} = 0.25$,
2. τ_2: $T_2 = 5$ ms, $C_2 = 2$ ms $\rightarrow \frac{C_2}{T_2} = 0.4$,
3. τ_3: $T_3 = 7$ ms, $C_1 = 2$ ms $\rightarrow \frac{C_2}{T_2} = 0.28$,

Schedulability test 1

With $n = 3$, the sum of $\frac{C_i}{T_i}$ must be lower than 0.7798. We become that $0.25 + 0.4 + 0.28 = 0,91 > 0.7798$.

An example

Schedulability test 2

1. $u_1(t) = C_1 = 1 \rightsquigarrow u_1(4) = 1$
2. $u_2(t) = .. \rightsquigarrow u_2(4) = 3, u_2(5) = 4$
3. $u_3(t) = ... \rightsquigarrow u_3(4) = 5, u_3(5) = 6, u_3(7) = 8$

We test:

1. $u_1(t) \leq t$ satisfied for $t = 4? \rightsquigarrow u_1(4) = 1 \leq 4 \Rightarrow$ **OK!**
2. $u_2(t) \leq t$ satisfied for $t \in 4, 5? \rightsquigarrow u_2(4) \leq 4, u_2(5) \leq 5 \Rightarrow$ **OK!**
3. $u_3(t) \leq t$ satisfied for $t \in 4, 5, 7? \rightsquigarrow u_3(4) > 4, u_3(5) > 5, u_3(7) > 7 \Rightarrow$ **NOT OK!**