CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Real-Time Systems

Exercise #7

Course Assistant Elena Marzi

Department of Computer Science and Engineering
Chalmers University of Technology

Multiprocessor scheduling

Today:

- Repetition of relevant concepts in multiprocessor scheduling
- Exercise on RMFF scheduling
- Exercise on RM-US[m/(3m-2)] scheduling

The examples are based on some old exam problems

Multiprocessor scheduling

- **How are tasks assigned to processors?**
  - Static assignment → off-line
  - Dynamic assignment→ on-line
- **How are tasks allowed to migrate?**
  - Partition scheduling → no task migration / RMFF scheduling
  - Global scheduling → task migration / RM-US scheduling

Multiprocessor scheduling

“**Partitioned scheduling**: If all tasks are assigned using the Rate-Monotonic-First-Fit (**RMFF**) algorithm, then all tasks are schedulable if the total task utilization does not exceed 41% of the total processor capacity.”

“**Global scheduling**: If tasks with the highest utilization are given highest priority and the remaining tasks are given RM priorities according to **RM-US**, then all tasks are schedulable if the total task utilization does not exceed 33.3% of the total processor capacity.”

Example 1: RMFF scheduling

Problem:

There are two approaches for scheduling tasks on multiprocessor platform: the *partitioned* approach and the *global* approach. The table below shows C_i (WCET) and T_i (period) for six periodic tasks to be scheduled on $m = 3$ processors. The relative deadline of each periodic task is equal to its period.

	C_i	T_i
τ_1	2	10
τ_2	10	25
τ_3	12	30
τ_4	5	10
τ_5	8	20
τ_6	7	100

The task set is schedulable using rate-monotonic first-fit (RMFF) partitioned scheduling algorithm. Show how the task set is partitioned on $m = 3$ processors so that all the deadlines are met using RMFF scheduling?

Example 1: RMFF scheduling

Rate-Monotonic-First-Fit (RMFF): (Dhall and Liu, 1978)

- Let the processors be indexed as $\mu_1, \mu_2, \ldots, \mu_m$
- Assign tasks in order of increasing periods (i.e., RM order).
- For each task τ_i, choose the lowest previously-used j such that τ_i, together with all tasks that have already been assigned to processor μ_j, can be feasibly scheduled according to the utilization-based RM-feasibility test.

If all tasks are successfully assigned using RMFF, then the tasks are schedulable on *m* processors.

Example 1: RMFF scheduling

$$U_{Total} = \sum_{i=1}^{6} C_i / T_i = \frac{2}{10} + \frac{10}{25} + \frac{12}{30} + \frac{5}{10} + \frac{8}{20} + \frac{7}{100} = 1.97$$

$$U_{\mathrm{RMFF}} = m(2^{1/2} - 1) = 3(2^{1/2} - 1) = 1.243$$

The tasks are schedulable if the following condition is true:

$$U_{Total} \le U_{\mathrm{RMFF}}$$

However: $U_{Total} > U_{\mathrm{RMFF}}$

Therefore, we cannot guarantee schedulability using the utilization based test. However, since the test is only a sufficient one we could try the RMFF algorithm.

Example 1: RMFF scheduling

The utilization of the tasks are

	C_i	T_i	U_i
τ_1	2	10	0.2
τ_2	10	25	0.4
τ_3	12	30	0.4
τ_4	5	10	0.5
τ_5	8	20	0.4
τ_6	7	100	0.07

The order of allocation (based in increasing period) is τ_1, τ_4, τ_5, τ_2, τ_3 and τ_6.

The three processors are indexed as μ_1, μ_2, and μ_3.

Example 1: RMFF scheduling

Task τ_1 can be allocated to μ_1 since there are no other tasks on μ_1.

Task τ_4 can also be allocated to μ_1 since

$$U_1 + U_4 = 0.2 + 0.5 = 0.7 <= 2 \cdot (2^{\frac{1}{2}} - 1) = 0.82$$

Task τ_5 cannot be allocated to μ_1 since

$$U_1 + U_4 + U_5 = 0.2 + 0.5 + 0.4 = 1.1 > 1$$

Task τ_5 can be allocated to μ_2 since there are no other tasks on μ_2.

Example 1: RMFF scheduling

Task τ_2 cannot be allocated to μ_1 since

$$U_1 + U_4 + U_2 = 0.2 + 0.5 + 0.4 = 1.1 > 1$$

Task τ_2 can be allocated to μ_2 since

$$U_5 + U_2 = 0.4 + 0.4 = 0.8 <= 2 \cdot (2^{\frac{1}{2}} - 1) = 0.82$$

Task τ_3 cannot be allocated to μ_1 since

$$U_1 + U_4 + U_3 = 0.2 + 0.5 + 0.4 = 1.1 > 1$$

Task τ_3 cannot be allocated to μ_2 since

$$U_5 + U_2 + U_3 = 0.4 + 0.4 + 0.4 = 1.2 > 1$$

Task τ_3 can be allocated to μ_3 since there are no other tasks on μ_3.

Example 1: RMFF scheduling

Task τ_6 can be allocated to μ_1 since

$$U_1 + U_4 + U_6 = 0.2 + 0.5 + 0.07 = 0.77 <= 3 \cdot (2^{\frac{1}{3}} - 1) = 0.78$$

So, the final allocation is as follows:

Processor μ_1 gets tasks τ_1, τ_4, and τ_6.

Processor μ_2 gets tasks τ_5 and τ_2.

Processor μ_3 gets tasks τ_3.

Example 2: RM-US scheduling

Problem: Consider the task set below for a system using global scheduling on m=3 processors. Show that the task set is schedulable on the processors assuming that task priorities are given according to RM-US[m/(3m-2)].

τ_1 τ_2 τ_3 τ_4

Task	C_i	T_i
τ_1	1	7
τ_2	4	19
τ_3	9	20
τ_4	11	22

Example 2: RM-US scheduling

$$U_{Total} = \sum_{i=1}^{4} C_i / T_i = \frac{1}{7} + \frac{4}{19} + \frac{9}{20} + \frac{11}{22} = 1.30$$

Task	C_i	T_i	U_i
τ_1	1	7	0.14
τ_2	4	19	0.21
τ_3	9	20	0.45
τ_4	11	22	0.5

$$U_{\text{RM-US}} = m^2 / (3m-2) = 9/7 = 1.2857$$

- Since $U_{Total} > U_{\text{RM-US}}$ the utilization based test for RM-US fails.
- However, since the test is only a sufficient one we could try response time analysis for global scheduling.

Example 2: RM-US scheduling

- RM-US[m/(3m-2)] assigns (static) priorities to tasks according to the following rule:

  If $U_i > m/(3m-2)$ then τ_i has the highest priority (ties broken arbitrarily)

  If $U_i \leq m/(3m-2)$ then τ_i has RM priority

Example 2: RM-US scheduling

$$m/(3m-2) = 3/7 = 0.4286$$

Task	C_i	T_i	U_i
τ_1	1	7	0.14
τ_2	4	19	0.21
τ_3	9	20	0.45
τ_4	11	22	0.5

- Task priorities are:
  - Tasks τ_3 and τ_4 have highest priority ("heavy tasks")
  - Task τ_1 has highest RM priority
  - Task τ_2 has lowest RM priority
- Since we have three processors, tasks τ_3, τ_4 and τ_1 are trivially schedulable ($C_i < T_i$) on one processor each.
- So, we want to calculate the response time for task τ_2.

$$R_i^{n+1} = C_i + \frac{1}{m} \sum_{\forall j \in hp(i)} \left(\left\lceil \frac{R_i^n}{T_j} \right\rceil C_j + C_j \right)$$

Example 2: RM-US scheduling

$$R_2^0 = C_2 = 4$$

$$R_2^1 = C_2 + \frac{1}{m}\left(\left(\left\lceil \frac{R_2^0}{T_3} \right\rceil C_3 + C_3\right) + \left(\left\lceil \frac{R_2^0}{T_4} \right\rceil C_4 + C_4\right) + \left(\left\lceil \frac{R_2^0}{T_1} \right\rceil C_1 + C_1\right)\right)$$

$$= 4 + \frac{1}{3}\left(\left(\left\lceil \frac{4}{20} \right\rceil 9 + 9\right) + \left(\left\lceil \frac{4}{22} \right\rceil 11 + 11\right) + \left(\left\lceil \frac{4}{7} \right\rceil 1 + 1\right)\right) = 4 + 42/3 = 18$$

$$R_2^2 = 4 + \frac{1}{3}\left(\left(\left\lceil \frac{18}{20} \right\rceil 9 + 9\right) + \left(\left\lceil \frac{18}{22} \right\rceil 11 + 11\right) + \left(\left\lceil \frac{18}{7} \right\rceil 1 + 1\right)\right) = 4 + 44/3 = 18.66$$

$$R_2^3 = 4 + \frac{1}{3}\left(\left(\left\lceil \frac{18.66}{20} \right\rceil 9 + 9\right) + \left(\left\lceil \frac{18.66}{22} \right\rceil 11 + 11\right) + \left(\left\lceil \frac{18.66}{7} \right\rceil 1 + 1\right)\right) = 18.66 < T_2 = 19$$