Schedulability test for a jobset using Deadline Monotonic algorithm

Youri Klaassens[a], Nick van Endhoven[b]

[a]student Computer Engineering Rotterdam University of Applied Science, 0996211@hr.nl, Zwaag
[b]student Computer Engineering Rotterdam University of Applied Science, 0998831@hr.nl, Breda

INTRODUCTION

For the Real-Time Operating System (ROS01) course taught at Rotterdam University of Applied Science the authors have to prove they understand the theory of RTOS scheduling and can analyse a jobset for a real time system. After the analysis of the jobset the authors should conclude if the jobset if schedulable according to the Deadline Monotonic algorithm.

The schedulability tests assume that the jobset will be executed on a uniprocessor system where tasks are preemtable. It also assumes that that there is no context-switching time. It also assumes that the execution of the scheduler does not require the processor, that is, the scheduler runs on another specialized processor.

The system has 4 different kind of shared resources k. Every shared resource k has a maximum hold time it may be claimed C_k. These 4 different shared resources with their respective maximum hold time are given in Table 1.

k	C_k
1	8
2	20
3	10
4	40

Table 1. Characteristics of the available shared resources.

This document uses task characteristics letters compatible with the letters defined by Cheng in $\dot{\text{T}}$ask T_i has a maximum computation time c_i, a deadline d_i and a period p_i. The characteristics of the various real-time tasks in this system can be seen in Table 2.

i	p_i	d_i	c_i	Uses shared resource
1	400	360	90	R2, R3, R1
2	600	580	50	R4
3	800	400	30	R1
4	700	420	40	R2
5	200	170	100	R4, R3

Table 2. Characteristics of the different tasks

The goal of this document is to prove whether the given real-time tasks in Table 2 are Deadline Monotonic schedulable taking into account the shared resources in Table 1. What now follows is the outline for the rest of this document. Section 1 uses a couple of simple schedulability tests not taking the shared resources into account.

1 SIMPLE SCHEDULABILITY TESTS

For the reader who is not familiar with the term schedulability test, a schedulability test is used to validate that a given application can satisfy its specified deadlines when scheduled according to a specific scheduling algorithm[4]. For the reader who is not familiar with the term schedulabe utilization, the schedulable utilization is the maximum utilization allowed for a set of tasks that will guarantee a feasible scheduling for the jobset[4]. Now we can represent the first schedulability test. Given a set of n independent, preemptable and period tasks on a uniprocessor, let U be the total utilization of this task set. A necessary and sufficient condition for feasible scheduling of this jobset is Equation 1

$$U = \sum_{i=1}^{n} \frac{c_i}{p_i} \leq 1 \qquad \text{Equation 1.}$$

Using Equation 1 we can use the task characteristics from Table 2 and calculate the utilization. If the calculated utilization is greater than 1 we can conclude that the CPU should do more calculations in a time unit than possible and this jobset is not schedulable. Equation 2 contains the elaboration.

$$U = \sum_{i=1}^{5} \frac{c_i}{p_i} = \frac{90}{400} + \frac{50}{600} + \frac{30}{900} + \frac{40}{700} + \frac{100}{200} \approx 0.90 \Rightarrow 0.90 \leq 1 \qquad \text{Equation 2.}$$

Since the CPU utilization is less than 1 we can conclude that the jobset may be schedulable. We can use the schedulability test in Equation 3 to test if the jobset can be guaranteed scheduled. If this test fails it does not men that the jobset is not schedulable. Given a set of n independent, preemtable and period tasks on a uniprocessor, let U be the total utilization of this jobset. A sufficient condition for feasible scheduling of this jobset is Equation 3.

$$U = \sum_{i=1}^{n} \frac{c_i}{p_i} \leq n(2^{\frac{1}{n}} - 1) \qquad \text{Equation 3.}$$

However, the condition is Equation 3 may result in under-utilization of the CPU[4]. Imagine $n \to \infty$ then the utilization is $\ln(2)$ or ≈ 0.693. Using Equation 3 with the jobset defined in Table 2 results in an elaboration seen in Equation 4. 0.90 is not less than or equal to $5(2^{\frac{1}{5}} - 1)$ which is approximately 0.74. This means that this test can not ensure that the jobset is schedulable, but it might still be possible.

$$U = \sum_{i=i}^{5} \frac{c_i}{p_i} = \frac{90}{400} + \frac{50}{600} + \frac{30}{900} + \frac{40}{700} + \frac{100}{200} \approx 0.90 \text{ which should be } \leq 5(2^{\frac{1}{5}} - 1) \qquad \text{Equation 4.}$$

2 SCHEDULABILITY TEST WITHOUT TAKING SHARED RESOURCES INTO ACCOUNT

This section will present the reader a shedulability test which can guarantee whether the jobset is Deadline Monotonic schedulable or not. It does not take the shared resources into account. The reader may wonder why we dedicate a section about proving whether the jobset is Deadline Monotonic schedulable without taking the shared resources into account. If this section proves that the jobset is not Deadline Monotonic schedulable without taking the shared resources into account then the jobset will not be Deadline Monotonic schedulable taking the shared resources into account. The prove whether the jobset is Deadline Monotonic which will be presented in the next section and is an extension on the schedulability test presented in this section.

Suppose we have three tasks T_1, T_2 and T_3. Task T_1 has the smallest deadline followed by T_2, and then T_3. It is intuitive to see that in order to schedule T_1 its computation time must be less than or equal to its period. Therefore the following necessary and sufficient condition must hold in Equation 5.

$$c_1 \leq p_1 \qquad \textbf{Equation 5.}$$

For T_2 to be feasible scheduled we need to find enough available time in the period $[0, p_2]$ that is not used by T_1. Imagine T_2 is finished executing at time t then the total number of iterations of T_1 is

$$\left\lceil \frac{t}{p_1} \right\rceil \qquad \textbf{Equation 6.}$$

To ensure that T_2 can complete execution at time t we must ensure that every iteration of T_1 in $[0, t]$ must be completed and there should still be enough time for T_2 to execute. The available time for T_2 is

$$t = \left\lceil \frac{t}{p_1} \right\rceil c_1 + c_2 \qquad \textbf{Equation 7.}$$

And similiar for T_3 to be feasible scheduled there must be enough time available after scheduling T_1 and T_2

$$t = \left\lceil \frac{t}{p_1} \right\rceil c_1 + \left\lceil \frac{t}{p_2} \right\rceil c_2 + c_3 \qquad \textbf{Equation 8.}$$

The only thing we need to consider is how to determine if such t exists such that the jobset can be feasible scheduled. There are an infinite number of datapoints in the time interval if no discrete time is assumed. However, the value of the ceiling $\left\lceil \frac{t}{p_1} \right\rceil$ only changes at multiples of p_1 with an increase of c_1. Thus we only need to show that a k exists such that

$$kp_1 \geq kc_1 + c_2 \text{ and } kp_1 \leq kp_2 \qquad \textbf{Equation 9.}$$

Therefore, we need to check that

$$t \geq \left\lceil \frac{t}{p_1} \right\rceil c_1 + c_2 \qquad \textbf{Equation 10.}$$

We will use the equations just derived on the jobset of Table 3. This is the same jobset we introduced in Table **??**, but is now arranged in order of priority. T_1 has now the highest priority because its deadline is the shortest compared to the other tasks followed by T_2 and so on.

i	p_i	d_i	c_i	Uses shared resource
1	200	170	100	R4, R3
2	400	360	90	R2, R3, R1
3	800	400	30	R1
4	700	420	40	R2
5	600	580	50	R4

Table 3. Characteristics of the different tasks rearranged in order of priority

Let

$$w_i(t) = \sum_{k=1}^{i} C_k \left\lceil \frac{t}{p_k} \right\rceil, 0 < t \leq p_i. \qquad \textbf{Equation 11.}$$

The following inequality

$$w_i(t) \leq t \qquad \textbf{Equation 12.}$$

holds for any time instant t chosen as follows:

$$t = kp_j, j = 1, \ldots i, k = 1 \ldots \left\lfloor \frac{p_i}{p_j} \right\rfloor \qquad \textbf{Equation 13.}$$

For T_1, $i = 1, j = 1, \ldots i = 1$, so

$$k = 1, \ldots, \left\lfloor \frac{p_i}{p_j} \right\rfloor = 1, \ldots, \left\lfloor \frac{200}{200} \right\rfloor = 1 \qquad \textbf{Equation 14.}$$

Thus, $t = kp_j = 1(200) = 200$. Task T_1 is is DM schedulable if $c_1 \leq 170$. Since $c_1 = 100 \leq 200$, T_1 is DM schedulable.

For T_2, $i = 2, j = 1, \ldots i = 1, 2$, so

$$k = 1, \ldots, \left\lfloor \frac{p_i}{p_j} \right\rfloor = 1, \ldots \left\lfloor \frac{400}{200} \right\rfloor = 1, 2 \qquad \textbf{Equation 15.}$$

Thus, $t = 1p_1 = 1(200) = 200$ or $t = 1p_2 = 1(400) = 400$ or $t = 2p_1 = 2(200) = 400$. Task T_2 is DM schedulable if $c_1 + c_2 \leq 200$ or $2c_1 + c_2 \leq 400$. Since $c_1 = 100, c_2 = 90$, $2(100) + 90 \leq 400$, thus T_2 is DM schedulable together with T_1.

For T_3, $i = 3, j = 1, \ldots i = 1, 2, 3$, so

$$k = 1, \ldots, \left\lfloor \frac{p_i}{p_j} \right\rfloor = 1, \ldots \left\lfloor \frac{800}{200} \right\rfloor = 1, 2, 3, 4 \qquad \textbf{Equation 16.}$$

Thus, $t = 1p_1 = 1(200) = 200$ or $t = 1d_2 = 1(400) = 400$ or $t = 2d_1 = 2(200) = 400$ or $t = 1d_3 = 1(800) = 800$ or $t = 3d_1 = 3(200) = 600$ or $t = 4d_1 = 4(200) = 800$. Task T_3 is DM schedulable if $c_1 + c_2 + c_3 \leq 200$ or $2c_1 + c_2 + c_3 \leq 400$ or $4c_1 + 2c_2 + c_3 \leq 800$ or $3c_1 + 2c_2 + c_3 \leq 600$. Since $c_1 = 100, c_2 = 90, c_3 = 30$, $2(100) + 90 + 30 \leq 400$, thus T_3 is DM schedulable together with T_1 and T_2.

For T_4, $i = 4, j = 1, \ldots i = 1, 2, 3, 4$, so

$$k = 1, \ldots, \left\lfloor \frac{p_i}{p_j} \right\rfloor = 1, \ldots \left\lfloor \frac{700}{200} \right\rfloor = 1, 2, 3 \qquad \textbf{Equation 17.}$$

Thus, $t = 1p_1 = 1(200) = 200$ or $t = 1d_2 = 1(400) = 400$ or $t = 2d_1 = 2(200) = 400$ or $t = 1d_3 = 1(800) = 800$ or $t = 3d_1 = 3(200) = 600$ or $t = 1d_4 = 1(700) = 700$. Task T_4 is DM schedulable if $c_1 + c_2 + c_3 + c_4 \leq 200$ or $2c_1 + c_2 + c_3 + c_4 \leq 400$ or $4c_1 + 2c_2 + c_3 + c_4 \leq 800$ or $3c_1 + 2c_2 + c_3 + c_4 \leq 600$ or $4c_1 + 2c_2 + c_3 + c_4 \leq 700$. Since $c_1 = 100, c_2 = 90, c_3 = 30, c_4 = 40$, $4(100) + 2(180) + 30 + 40 \leq 700$, thus T_4 is DM schedulable together with T_1 and T_2 and T_3.

For T_5, $i = 5, j = 1, \ldots i = 1, 2, 3, 4, 5$, so

$$k = 1, \ldots, \left\lfloor \frac{p_i}{p_j} \right\rfloor = 1, \ldots \left\lfloor \frac{600}{200} \right\rfloor = 1, 2, 3 \quad \textbf{Equation 18.}$$

Thus, $t = 1p_1 = 1(200) = 200$ or $t = 1d_2 = 1(400) = 400$ or $t = 2d_1 = 2(200) = 400$ or $t = 1d_3 = 1(800) = 800$ or $t = 3d_1 = 3(200) = 600$ or $t = 1d_4 = 1(700) = 700$ or $t = 1d_5 = 1(600) = 600$. Task T_5 is DM schedulable if $c_1 + c_2 + c_3 + c_4 + c_5 \le 200$ or $2c_1 + c_2 + c_3 + c_4 + c_5 \le 400$ or $4c_1 + 2c_2 + c_3 + 2c_4 + 2c_5 \le 800$ or $3c_1 + 2c_2 + c_3 + c_4 + c_5 \le 600$ or $4c_1 + 2c_2 + c_3 + c_4 + 2c_5 \le 700$. Since $c_1 = 100, c_2 = 90, c_3 = 30, c_4 = 40, c_5 = 50$, $4(100) + 2(90) + 30 + 2(40) + 2(50) \le 800$, thus T_5 is DM schedulable together with T_1 and T_2 and T_3 and T_4.

ACKNOWLEDGEMENTS

REFERENCES

1. Marquez, V., Frohlich, T., Armache, J. P., Sohmen, D., Donhofer, A., Mikolajka, A., Berninghausen, O., Thomm, M., Beckmann, R., Arnold, G. J., and Wilson, D. N. (2011) Proteomic characterization of archaeal ribosomes reveals the presence of novel archaeal-specific ribosomal proteins, *J Mol Biol* 405, 1215–1232. *https://doi.org/10.33697/ajur.2019.003*
2. Fierke, C. A., and Hammes, G. G. (1996) Transient Kinetic Approaches to Enzyme Mechanisms, in *Contemporary Enzyme Kinetics and Mechanism* (Purich, D., Ed.) 2nd ed., 1–35, Academic Press, New York.
3. Agricultural Research Service, U.S.D.A. National Nutrient Database for Standard Reference, Release 26, *http://ndb.nal.usda.gov/ndb/search/list* (accessed Mar 2014)
4. Cheng, A. M. K (2002) Real-Time Systems Scheduling, Analysis and Verification

ABOUT THE STUDENT AUTHOR

Youri Klaassens is a final year bachelor student at Inholland University of Applied Science studying Computer Engineering.

Nick van Endhoven is a final year bachelor student at Avans University of Applied Science studying Computer Engineering.

3 SUMMARY