Exam proposal solution for

TDDD07
Real-time Systems

Klervie Toczé
Simin Nadjm-Tehrani

Exam from January 2019

Real-Time Systems Laboratory
Department of Computer and Information Science
Linköping University, Sweden

December 2022

Suggested Solution

Disclaimer: This is a suggested solution, other answers may be acceptable. Ask your teacher if you would like to discuss your answer.

Q1: Scheduling

a) 1)

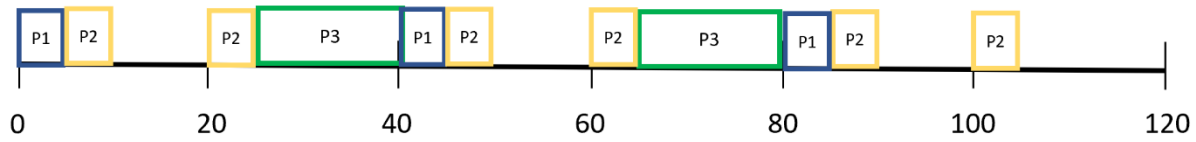
Process	Period (T)	WCET (C)
P1: Navigation	40 ms	5 ms
P2: Collision avoidance	20 ms	5 ms
P3: Sensor fusion	60 ms	15 ms

Possible assumption:

- Output jitter is allowed for P3

Minor cycle: GCD(40,20,60)=20 ms
Major cycle: LCM(40,20,60)=120 ms

The schedule looks as follows:



2) The new process set is:

Process	Period (T)	WCET (C)
P1: Navigation	40 ms	5 ms
P2: Collision avoidance	20 ms	5 ms
P3: Sensor fusion	60 ms	15 ms
P4: Speech recognition	50 ms	10 ms
P5: Voice synthesis	50 ms	5 ms
P6: Decision	80 ms	5 ms

Relevant assumptions:

- The tasks are independent
- No precedence relation exists
- The overhead of the OS is less than 0.0125

We use the schedulability test for EDF:

$$\sum_{i=1}^{6} \frac{C_i}{T_i} = \frac{5}{40} + \frac{5}{20} + \frac{15}{60} + \frac{10}{50} + \frac{5}{50} + \frac{5}{80} = 0.125 + 0.25 + 0.25 + 0.2 + 0.1 + 0.0625$$

$$= 0{,}9875$$

$$\leq 1$$

So the process set is schedulable using EDF.

3) The new process set is:

Process	Period (T)	New WCET (C)	Priority
P1: Speech recognition	50 ms	13 ms	4 (Highest)
P2: Voice synthesis	50 ms	6.5 ms	3
P3: Decision	80 ms	6.5 ms	1 (Lowest)
P4: Sensor fusion	60 ms	19.5 ms	2

We use the utilisation-based schedulability test for rate monotonic:

$$\sum_{i=1}^{4} \frac{C_i}{T_i} = \frac{13}{50} + \frac{6.5}{50} + \frac{6.5}{80} + \frac{19.5}{60} = 0.26 + 0.13 + 0.08125 + 0.325 = 0.79625$$

$$n\left(2^{\frac{1}{n}} - 1\right) = 4\left(2^{\frac{1}{4}} - 1\right) = 0.7568$$

$$0.79625 \not\leq 0.7568$$

The schedulability is therefore not guaranteed, we need to perform the response time analysis.

We start by calculating the blocking times. No resources are shared so

$$\mathrm{B_{P1}} = \mathrm{B_{P2}} = \mathrm{B_{P3}} = \mathrm{B_{P4}} = 0 \text{ ms}$$

We then look at the processes starting by the one with highest priority.
Note: Due to time constraints, only partial intermediary results are reported here. At the exam, you need to write the detailed calculations. Ask your teacher if you need help!

Process P1: No process can preempt it.

$$R_{P1} = C_{P1} + B_{P1} + J_{P1} = 13\ ms$$

Process P2: Can be pre-empted by P1.

$$w_{P2}^0 = C_{P2} + B_{P2} = 6.5 + 0$$

$$\mathrm{w_{P2}^1} = \mathrm{C_{P2}} + \mathrm{B_{P2}} + \left\lceil \frac{\mathrm{w_{P2}^0}}{\mathrm{T_{P1}}} \right\rceil \mathrm{C_{P1}} = 19.5$$

$$\mathrm{w_{P2}^2} = \mathrm{C_{P2}} + \mathrm{B_{P2}} + \left\lceil \frac{\mathrm{w_{P2}^1}}{\mathrm{T_{P1}}} \right\rceil \mathrm{C_{P1}} = 19.5$$

$$w_{P2} = 19.5\ ms$$

$$R_{P2} = w_{P2} + J_{P2} = 19.5ms$$

Process P4: Can be pre-empted by P2 and P1.

$$w_{P4}^0 = C_{P4} + B_{P4} = 19.5 + 0$$

$$\mathrm{w_{P4}^1} = \mathrm{C_{P4}} + \mathrm{B_{P4}} + \left\lceil \frac{\mathrm{w_{P4}^0}}{\mathrm{T_{P1}}} \right\rceil \mathrm{C_{P1}} + \left\lceil \frac{\mathrm{w_{P4}^0}}{\mathrm{T_{P2}}} \right\rceil \mathrm{C_{P2}} = 39$$

$$\mathrm{w_{P4}^2} = \mathrm{C_{P4}} + \mathrm{B_{P4}} + \left\lceil \frac{\mathrm{w_{P4}^1}}{\mathrm{T_{P1}}} \right\rceil \mathrm{C_{P1}} + \left\lceil \frac{\mathrm{w_{P4}^1}}{\mathrm{T_{P2}}} \right\rceil \mathrm{C_{P2}} = 39$$

$$w_{P4} = 39\ ms$$

$$R_{P4} = w_{P4} + J_{P4} = 39\ ms$$

Process P3: Can be pre-empted by P4, P2 and P1.

$$w_{P3}^{0} = C_{P3} + B_{P3} = 6.5 + 0$$

$$w_{P3}^{1} = C_{P3} + B_{P3} + \left\lceil \frac{w_{P3}^{0}}{T_{P1}} \right\rceil C_{P1} + \left\lceil \frac{w_{P3}^{0}}{T_{P2}} \right\rceil C_{P2} + \left\lceil \frac{w_{P3}^{0}}{T_{P4}} \right\rceil C_{P4} = 45.5$$

$$w_{P3}^{1} = C_{P3} + B_{P3} + \left\lceil \frac{w_{P3}^{1}}{T_{P1}} \right\rceil C_{P1} + \left\lceil \frac{w_{P3}^{1}}{T_{P2}} \right\rceil C_{P2} + \left\lceil \frac{w_{P3}^{1}}{T_{P4}} \right\rceil C_{P4} = 45.5$$

$$w_{P3} = 45.5\ ms$$

$$R_{P3} = w_{P3} + J_{P3} = 45.5\ ms$$

Conclusion:

$$R_{P1} = 13 \leq 50 = D_{P1}$$
$$R_{P2} = 19.5 \leq 50 = D_{P2}$$
$$R_{P3} = 45.5 \leq 80 = D_{P3}$$
$$R_{P4} = 39 \leq 60 = D_{P4}$$

All the response times are less than the deadlines, so the process set is schedulable with RMS.

4) The process set is:

Process	**Period (T)**	**WCET (C)**	**Critical section length**	**Priority (π)**
P1: Speech recognition	50 ms	13 ms	2 ms	4 (Highest)
P2: Voice synthesis	50 ms	6.5 ms	3 ms	3
P3: Decision	80 ms	6.5 ms		1
P4: Sensor fusion	60 ms	19.5 ms		2

As some processes now share a resource, we need to take into account the blocking times of the processes.

First we calculate the ceiling value of resource M.
Ceiling(M)= max(π1,π2)=4

For calculating the blocking times, we start from the lowest priority process.

Process P3: No process with lower priority (lp(P3)= Ø), so no process can block P3.
Then, $B_{P3} = 0ms$

Process P4: lp(P4)= {P3}. We have to check whether P4 can be blocked by the lower priority process P3.

- Process P3 does not lock any resource, so it cannot block P4

Then, $B_{P4} = 0ms$

Process P2: lp(P2)= {P4,P3}. We have to check whether P2 can be blocked by the lower priority processes P4 or P3.

- Process P3 does not lock any resource, so it cannot block P2
- Process P4 does not lock any resource, so it cannot block P2

Then, $B_{P2} = 0ms$

Process P1: lp(P1)= {P2,P4,P3}. We have to check whether P1 can be blocked by the lower priority process P2, P4 or P3.

- Process P3 does not lock any resource, so it cannot block P1
- Process P4 does not lock any resource, so it cannot block P1
- Process P2 can lock M and ceiling(M)=4 ≥π1=4. Process P2 can block P1.

Then, $B_{P1} = max(t_{M,P2}) = 3\ ms$

b) The proof has been presented during the lectures (Lecture 3 slide 20).
Hints: Show that an arbitrary process that is waiting will not wait indefinitely. Describe the possible cases and what happens in each of them.

Q2: Dependability and predictability

a) Fault models that are plausible in this scenario are 1) a node (including the network card) producing massive volume of unwanted messages – i.e. acting in a Byzantine manner, and 2) the lack of isolation of the channel for application data and the channel for management data, leading to omission faults for data in the common channel.
The scenario indicates that the network card fault appears transient/intermittent but the network channel allocation fault is a permanent design fault.

b) Two methods for implementing fault tolerance using redundancy in space:
- Voting systems: This method is appropriate for Byzantine nodes.
- Exceptions: This method is appropriate to prevent programming errors or input values being out of range (external faults).
One method for implementing fault tolerance using redundancy in time:
- Re-computing a result: This method is appropriate for transient faults affecting memory.

Q3: Real-time Communication

There are different possible answers to this question. Here, two alternatives are mentioned.

One would be to define a clock synchronisation service and explain why it is needed for systems that have to ensure that critical tasks' communication is not affected by errors/delays in non-critical tasks, also know as partitioned applications. These requirements appear in aerospace applications.

Another would be to discuss the membership service and explain that it is needed when several distributed processes (in different nodes) need to act in concert for a common goal, and do so even if there are potential faults that stops one node/process from being active. Then the membership protocol need to build on the so-called fail-silent principle. So that when a node is not communicating when it should the other nodes (in the membership group) know that they

need to compensate for the failed node by changing their steering algorithm within a well-defined number of slots.

Q4: Application design & RTOS

a) (1) True. Penetration testing requires a list of test cases that are expected to potentially reveal those security conditions for which the system is not prepared to deal with properly. [Note: This instance of the course had an industrial speaker that focused on penetration testing as a method for increasing security].
(2) False. The platform and its impact should be separated from the rest in the analysis.
(3) False. UML profiles that represent timeliness primitives, such as MARTE can be useful at modelling stage.

b) In a RTOS, one has absolute control over the memory (no dynamic allocation).
If it is allowed, other mechanisms are needed to provide deterministic management of fragmentation: for example using pools instead of heap or control over garbage collection (in real-time Java).

c) The benefit of having a standard for operating systems in the real-time context is to have sector interoperability. It enables to keep a competitive advantage while sharing interfaces. One example where the lack of complying with an existing standard implied a burden on product suppliers is when Toyota decided to have their own variant of RTOS (not OSEK compatible) and one car provided an unintended acceleration that killed someone. Toyota was judged responsible.

Q5: Distributed systems, Quality of Service (QoS)

a) The two major functions are the hot spot solver and the cold spot solver. The hot spot solver identifies overloaded servers (from which VMs should be migrated away) thereby improving the performance of the servers that have some resource over-utilised (CPU, memory or bandwidth). The cold spot solver identifies mostly idle servers that are candidates to be offloaded and then turned off to save energy.

b) The problem is that the clocks are then not synchronized. Services that cannot be provided are then e.g. accuracy towards UTC or globally usable timestamps. Instead one has to rely on internal synchronization (e.g. using the Lamport/Melliar-Smith algorithm) to use any notion of timer or timeout, alternatively not use time at all and only rely on order of events.