INFO-F-404 : Operating Systems II

1 Exercises

Exercise 1 : global and partitioned RM

a) Find a system that can be scheduled using global RM, but not using partitioned RM (on the same hardware).

Task index	Release time	WCET	Deadline = Period
Tâche τ_1	0	20	30
Tâche τ_2	0	40	60
Tâche τ_3	0	60	120

Table 1: System of 3 periodic tasks. This system can be scheduled using global RM algorithm, but not using partitioned RM.

Answer : The system described by Table 1 could be scheduled using global RM algorithm on 2 processors and following priorities : $P(\tau_1) > P(\tau_2) > P(\tau_3)$ (see Figure 1). However this system can not be scheduled using partitioned RM. Any partitioning in two groups implies that one of the subsystems would have its utilisation factor grater than 1.

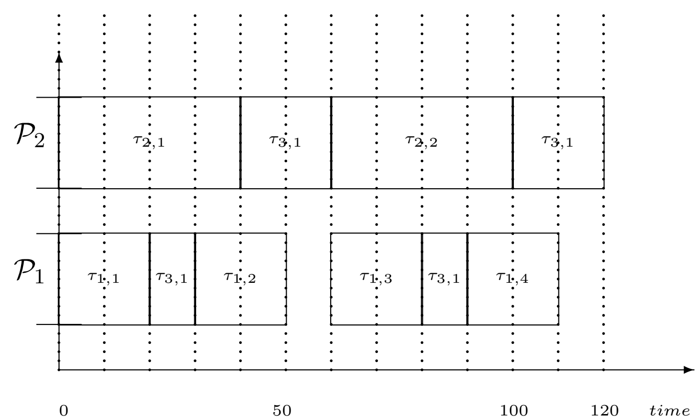

Figure 1: *Scheduling with global RM ($\tau_1 > \tau_2 > \tau_3$) of the system described by Table 1.*

b) Find a system that can be scheduled using partitioned RM, but not using global RM (on the same hardware).

Task index	Release time	WCET	Deadline = Period
Tâche τ_1	0	10	20
Tâche τ_2	0	20	40
Tâche τ_3	0	20	30
Tâche τ_4	0	20	60

Table 2: System of 4 periodic tasks. This system can be scheduled using partitioned RM algorithm, but not using global RM.

Answer : The system described by Table 2 could be scheduled on 2 processors using partitioning $\tau^1 = \{\tau_1, \tau_2\}$, $\tau^2 = \{\tau_3, \tau_4\}$, then RM algorithm could be used on each processor. However this system can not be scheduled using global RM : there is no assignment of fix priorities to tasks that will satisfy all deadlines.

Exercise 2 : partitioned EDF

Let's consider the system represented by Table 3. We suppose that all these tasks are independent and preemptible. We also suppose that we have a hardware system with 3 identical processors.

Task index	WCET	Deadline = Period
Tâche τ_1	10	50
Tâche τ_2	30	120
Tâche τ_3	10	100
Tâche τ_4	50	200
Tâche τ_5	20	100
Tâche τ_6	50	200
Tâche τ_7	250	300
Tâche τ_8	10	200

Table 3: System of 8 periodic, synchronous tasks with implicit deadline.

a) What conclusions about the schedulability of this system we could make using the FFDU test?

Answer : A system could be scheduled usinf FFDU (and by EDF on each processor) if the following formula is satisfied (m is the number of processors):

$$U_{tot}(\tau) < \frac{(m+1)}{2} \text{ and } U_{\max}(\tau) \leq 1$$

Here we have :

$$U_{tot}(\tau) = \frac{10}{50} + \frac{30}{120} + \frac{10}{100} + \frac{50}{200} + \frac{20}{100} + \frac{50}{200} + \frac{250}{300} + \frac{10}{200}$$

$$U_{\max}(\tau) = \frac{250}{300}$$

So, $U_{tot}(\tau) = 2.134$, $U_{\max} = \frac{5}{6}$ and $\frac{m+1}{2} = \frac{3+1}{2} = 2$. Since $2.134 > 2$ we can not reach a conclusion using this test.

b) Find the partitioning of these 8 tasks using FFDU algorithm.

Answer : $U(\tau_7) \geq U(\tau_2) \geq U(\tau_4) \geq U(\tau_6) \geq U(\tau_1) \geq U(\tau_5) \geq U(\tau_3) \geq U(\tau_8)$.

Processeurs	**Partitions**
$\mathcal{P}_1$	$\tau^1 = \{\tau_7, \tau_3, \tau_8\}$
$\mathcal{P}_2$	$\tau^2 = \{\tau_2, \tau_4, \tau_6, \tau_1\}$
$\mathcal{P}_3$	$\tau^3 = \{\tau_5\}$

Table 4: Partitioning by FFDU.

Exercise 3 : EDF$^{(k)}$

Let's consider the system represented by Table 5. We suppose that all these tasks are independent and preemptible.

Task index	WCET	Deadline = Period
Tâche τ_1	14	19
Tâche τ_2	1	3
Tâche τ_3	2	7
Tâche τ_4	1	5
Tâche τ_5	1	10

Table 5: System of 5 periodic, synchronous tasks with implicit deadline.

a) Find the number of processors required by the system in case we want to use global EDF scheduler.

Answer : For this system, we have :

- $U(\tau_1) = 14/19 \approx 0.737$
- $U(\tau_2) = 1/3 \approx 0.333$
- $U(\tau_3) = 2/7 \approx 0.286$
- $U(\tau_4) = 0.2$
- $U(\tau_5) = 0.1$

So $U_{tot}(\tau) \approx 1.657$. By using the following expression :

$$m \geq \left\lceil \frac{U_{tot}(\tau) - U_{\max}(\tau)}{1 - U_{\max}(\tau)} \right\rceil$$

We have : $m \geq \lceil \frac{1.657-0.737}{1-0.737} \rceil = 4$.

b) Find the number of processors required by the system (and the optimal value of k) in case we want to use EDF^k scheduler.

Answer : By using the expression (1) (for $k = 1, \ldots, n$) we obtain results (see Table 6).

$$m_{\min} = \min_{k=1}^{n} \left\{ (k-1) + \left\lceil \frac{U(\tau^{(k+1)})}{1 - U(\tau_k)} \right\rceil \right\} \quad (1)$$

Note that we can schedule this system using EDF^k on 2 processors. We can simply fix the deadline of τ_1 to $-\infty$ and use EDF.

k	$U_{tot}(\tau^{(k)})$	$U_{\max}(\tau^{(k)})$	**min** m
1	1.657	0.737	$(1-1)+\lceil\frac{0.92}{1-0.737}\rceil=4$
2	0.92	0.333	$(2-1)+\lceil\frac{0.586}{1-0.333}\rceil=2$
3	0.586	0.286	$(3-1)+\lceil\frac{0.3}{1-0.286}\rceil=3$
4	0.3	0.2	$(4-1)+\lceil\frac{0.1}{1-0.2}\rceil=4$
4	0.1	0.1	$(5-1)+\lceil\frac{0}{1-0.1}\rceil=4$

Table 6: Finding optimal k.