REAL TIME OPERATING QUESTIONS ASSIGNMENT

PART A

1: Rate Monotonic Scheduling (RMS) Analysis

Given the following tasks, fill in the table with their utilizations and determine if the set is schedulable.

Task	Period (ms)	Execution Time (ms)	Utilization (U = Execution Time / Period)
T1	4	1	
T2	5	2	
T3	10	3	

Question: Calculate the CPU utilization for each task and determine if the set is schedulable using RMS.

Question 2: Earliest Deadline First (EDF) Scheduling

Consider the following tasks. Complete the table with their deadlines and response times, then determine if they meet their deadlines.

Task	Period (ms)	Execution Time (ms)	Deadline (ms)	Response Time (ms)
T1	4	1	4	
T2	5	2	5	
T3	6	1	6	

Question: Calculate the response time for each task and determine if they are schedulable under EDF.

Question 3: Task Response Time Calculation

Given the following task set, fill in the response times considering the given priorities.

Task	Priority	Period (ms)	Execution Time (ms)	Response Time (ms)
T1	1	6	2	
T2	2	8	3	
T3	3	10	1	

Question: Calculate the worst-case response time for each task using a priority scheduling algorithm.

Question 4: Context Switching Impact

Given three tasks and a context switch time of 1 ms, complete the table with the effective execution time after considering context switching.

Task	Period (ms)	Execution Time (ms)	Context Switches	Effective Execution Time (ms)
T1	5	2	1	
T2	10	3	1	
T3	15	4	2	

Question: Calculate the effective execution time for each task, including context switching overhead.

Question 5: Missed Deadlines in a Hyperperiod

Using the following task set, fill in the table with the number of deadlines missed in a hyperperiod.

Task	Period (ms)	Execution Time (ms)	Number of Deadlines in Hyperperiod (30 ms)	Deadlines Missed
T1	6	2		
T2	10	3		

Task	Period (ms)	Execution Time (ms)	Number of Deadlines in Hyperperiod (30 ms)	Deadlines Missed
T3	12	4		

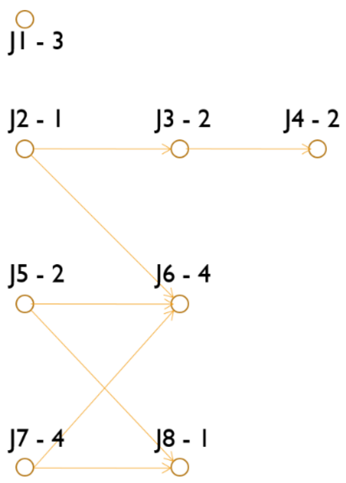
Question: Calculate the number of missed deadlines for each task in a 30 ms hyperperiod.

PART B

For the following Task sets and scheduling algorithms – submit the screen shoot of the schedule generated by Cheddar – with proper interpretations and conclusions regarding the schedule generated. If the task set is not schedulable – suggest an alternate method of scheduling

Q1

For the task graph show below show that a priority driven schedule with non-preemption will produce a better schedule than priority driven schedule with pre-emption



All tasks are aperiodic with a deadline of 12. All tasks except for J5 release at 0. J5 releases at 4. The priority of jobs is such that J1 has a higher priority when compared to J2 has a higher priority when compared to J3 and so on. The scheduling is done on a processor which has dual identical cores with jobs migratable at any point in time.

Q2

Show that the following tasks table is schedulable using EDF on a mono-core processor, if pre-emption is allowed but is not schedulable if pre-emption is not allowed. All tasks are aperiodic.

	J1	**J2**	**J3**
r	0	2	4
e	3	6	4
d	10	14	12

Q3

Show that the following tasks table is schedulable using LLF but not using EDF on a dual-core processor. All tasks are aperiodic.

	J1	**J2**	**J3**
r	0	0	0
e	1	1	5
d	1	2	5

Q4

Show that the following tasks table causes indeterminism due to the execution time – when scheduled using EDF. All tasks are periodic

	R	**d**	**e**
J1	0	10	5
J2	0	10	2-6

J3	4	15	8
J4	0	20	10

Q5

Show that the following tasks are not schedulable using either EDf/LLF on a tri-core processor. All tasks are periodic and preemptable. Migration is not possible between a Job.

	A	**B**	**C**	**D**	**E**
ϕ	0	0	0	0	0
e	1	1	1	6	6
p	2	2	2	8	8

Experiment 2

IMPLEMENTING USER DEFINED SCHEDULING ALGORITHMS

In -Lab Report

Q6

Create a scheduler uf.sc that will assign highest static priority to task with the least CPU utilization and compare it against the behaviour of a rate monotonic scheduler for any task set example from class or from your textbook.

Q7

Create a scheduler critical.sc that will schedule a task that is critical immediately, else it uses RMS and compare it against the behavior of a rate monotonic scheduler for the following task set. If there are multiple critical tasks – then RMS is applied within them. Under what circumstances will this scheduler work better than RMS – show such a task set example.

Q8

For the following Task set and scheduling algorithms – submit the screen shot of the schedule generated by Cheddar – with proper interpretations and conclusions regarding the schedule generated.

TASKS :	A	B	C	D	E	F	G	H	I	J
COMPUTATION TIME	5	7	3	1	10	16	1	13	9	17
TIME PERIOD	10	21	22	24	30	40	50	55	75	100

Q9

Consider two periodic tasks with the following parameters :
Task T1 : Period=Deadline=31, Capacity=8, Start time = 0
Task T2 : Period=Deadline=30, Capacity=8, Start time = 2

Prioritie are assigned according to Rate Monotonic.
T1 and T2 require the access to the shared resources R1 and R2 as follows:
T1 needs R1 from the 2nd unit of time of its capacity upto the 8th (included).
T2 needs R1 from the 6th unit of time of its capacity upto the 8th (included).
T1 needs R2 from the 4th unit of time of its capacity upto the 8th (included).
T2 needs R2 from the 2nd unit of time of its capacity upto the 8th (included).

Assume that both R1 and R2 apply PIP (Priority Inheritance Protocol).

Edit a Cheddar model for such software architecture and compute the scheduling simulation during the 30th first units of time (button).
Compute again the scheduling simulation during the 30th units of time with the PCP resources.

- Show in the time lines when T1 and T2 lock and unlock R1 and R2.
- Say when the priorities of the tasks change due to PCP.
- Compare those results with the PIP Protocol.

Q10

We consider two periodic tasks, synchronous and with deadline on request : tasks T1 and T2. They are defined as follow:

- Task T1 : `Capacity` = 4, `Deadline` = `Period` = 8, `Start time` = 0
- Task T2 : `Capacity` = 5, `Deadline` = `Period` = 10, `Start time` = 0
- Priority are assigned according to Rate Monotonic.

1. Edit this Cheddar model for the task set running on the processor.
2. Edit the core to host a **preemptive fixed priority scheduler**.
3. With Cheddar, compute a scheduling simulation for the feasibility interval (button).
4. From this simulation, what are the worst case response times of each task? Can we see missed deadlines?
5. Change you Cheddar model in order to use a preemptive EDF instead (value `Earliest_Deadline_First_Protocol` for the `Scheduler type` attribute). Do again questions 2 and 3.
6. Compare both analysis results. What can you see? Is it surprising?
7. Change again your model: now we consider a preemptive EDF with the same task sets, but you must add now an aperiodic task. We do not require that the deadline of this aperiodic task must be met. To define such aperiodic task, change the value for the attribute `Task Type` (see. figure 5):
    - Periodic task T1: `Period`=`Deadline`=8, `Capacity`=4, `Start time` = 0
    - Periodic task T2: `Period`=`Deadline`=16, `Capacity`=8, `Start time` = 0
    - **Aperiodic** task T3: `Deadline`=15, `Capacity`=4, `Start time` = 0
8. Compute again the scheduling simulation for the first 30th units of time (button). What can you see now?
9. What can we conclude about the suitability of EDF for critical real-time systems? Is it the case with fixed priority scheduling?

Q11

We consider two periodic tasks, synchronous, with deadline on request: tasks T1 and T2. They are defined as follow:

- Task T1: `Period`=`Deadline`=9, `Capacity`=4, `Start time` = 0
- Task T2: `Period`=`Deadline`=8, `Capacity`=3, `Start time` = 0

We investigate now another scheduling policy: LLF (Least Laxity First). This policy selects the task to run amoung the ready tasks according to a dynamic priority called 'laxity'. `Li(t)`, the laxity of a task `i` at time `t` can be computed by `Li(t)= Deadline - remaining(t)` where `remaining(t)` is the remaining capacity of the task at time `t`.

1. Edit a Cheddar model for the task set running on the processor.
2. Edit the core to host a **preemptive EDF scheduler**.
3. With Cheddar, compute a scheduling simulation for the feasibility interval (button).

4. From this simulation, what are the worst case response times of each task? Can we see the missed deadline?
5. Change your Cheddar model in order to use a LLF policy instead (value `Least_Laxity_First_Protocol` for the `Scheduler type` attribute). Do again questions 2 and 3.
6. What can we see about the results produced by both those scheduling policies? What are the common points and differences?
7. Conclude about the choice between those two scheduling policies.

Q12

This exercise is extracted from [COT 00] and is about a simplified architecture model of the Mars Pathfinder mission. In this exercise, you must look for a design mistake and propose a solution for it. In 1997, Mars Pathfinder casts a mobile robot called Sojourner on Mars. This mobile robot is controled by a multitask software running on a VxWorks target. This software is composed of the following tasks :

Tasks	**Priorities**	**Periods/Deadlines**	**WCET**
SCHED_BUS	1	125 ms	25 ms
DATA	2	125 ms	25 ms
CONTROL	3	250 ms	25 ms
RADIO	4	250 ms	25 ms
VIDEO	5	250 ms	25 ms
MESURE	6	5000 ms	50 ms
FORECAST	7	5000 ms	Between 50 ms and 75 ms

- All the tasks are periodic, synchronous, and have deadlines equal to periods.
- FORECAST is sometimes released for a job of 50 ms, and sometimes for a job of 75 ms, depending of the size of the payload to handle.
- Priority levels expressed bellow are VxWorks priorities: the lower the value is, the higher the priority level is.
- DATA, CONTROL, MESURE and FORECAST required a shared data which is accessed by a critical section during all their execution times (i.e all their capacity). To reduce costs, those critical sections are implemented with a mutex that does not use inheritance priority protocol such as PIP or PCP.

1. During the mission of Mars PathFinder, operators noticed that some deadlines were missed, leading to a reboot of the hardware and some losses of data. What are the missed deadlines? Why?
2. How to solve this issue? Apply it on your Cheddar ADL model.