2025

M.Sc. 1st Semester Examination

COMPUTER SCIENCE

Paper : COSE404B8

(Real Time System)

(Practical)

Full Marks : 15 Time : One Hour

The figures in the margin indicate full marks.

Answer any ***one*** question : 10×1=10

(Choose the question on the lottery basis)

1. Write a FreeRTOS program to create a task that toggles an LED (or prints a message) every 500ms using vTaskDelayUntil(). Explain why vTaskDelayUntil() is preferred over vTaskDelay() for periodic tasks.

2. Write a script or C program for RTLinux that records the timestamp (using clock_gettime) before and after a task yield. Calculate the average **Context Switch Latency** over 100 iterations.

3. Create two tasks : Task A (High Priority) and Task B (Low Priority). Write a program where Task B starts a long computation, and Task A preempts it halfway through. Log the output to verify the order of execution.

4. Given three tasks with periods T_1 =4ms, T_2 =6ms, T_3 = 10ms, write a SimSo Python configuration script to assign priorities based on the RM algorithm and check if the schedule is feasible.

5. In SimSo or a C-based simulation, implement an **Earliest Deadline First** scheduler. Write a test case where a task with a later arrival time but an earlier deadline correctly preempts a currently running task.

6. Write a FreeRTOS program with three tasks (High, Medium, Low) and one Semaphore. Create a scenario where the Medium task prevents the High task from running because the Low task is holding the Semaphore.

7. Write a program where a "Sensor Task" reads a simulated value every 100ms and sends it to a "Controller Task" using a **Queue**. If the Queue is full, the Sensor Task must wait (block) for 10ms.

8. Using Socket Programming (C/Python), write a client-server pair where the server must process a packet within 50ms. Implement a "deadline check" that logs a "Failure" (Hard RT) vs. "Late Processing" (Soft RT) if the time is exceeded.

9. Write a task that is supposed to wake up every 20ms. Inside the task, calculate the difference between the expected wake-up time and the actual wake-up time. Output the **maximum jitter** observed over 1,000 cycles.

Practical Notebook and Viva-voce : 05 marks
