Problem 1 [20 points]: Realtime programming

Consider a computer that is embedded in a patient-monitoring system. The system is arranged so that an interrupt is generated at the highest priority, through vector location 100 (octal), every time the patient’s heart beats. In addition, a mild electric shock can be administered via a device control register, the address of which is 177760(octal). The register is set up so that every time an integer value \( x \) is assigned to it the patient receives \( x \) volts over a small period of time.

If no heartbeat is recorded within a 5 second period then the patient’s life is in danger. Two actions should be taken when the patient’s heart fails: the first is that a ‘supervisor’ task should be notified so that it may sound the hospital alarm, the second is that a single electric shock of 5 volts should be administered. If the patient fails to respond then the voltage should be increased by 1 volt for every further 5 seconds.

Write a program in your favorite language which monitors the patient’s heart and initiates the actions described above.

```plaintext
Heart_Monitor : constant Interrupt_Id := ....;

Word : constant := 2; -- number of storage units in a word
Bits_In_Word : constant := 16; -- bits in work

type Csr is new Integer;
for Csr'Size use Bits_In_Word;
for Csr'Alignment use Word;
for Csr'Bit_order use Low_Order_First;

protected Interrupt_Handler is
  entry Wait_Hearth_Beat;

private
  procedure Handler;
pragma Attach_Handler(Handler, Heart_Monitor);
pragma Interrupt_Priority(Interrupt_Priority'Last);
Interrupt Occurred : Boolean := False;
end Interrupt_Handler;

task Patient_Monitor;

protected body Interrupt_Handler is
```
Problem 2[15 points]: Scheduling

Consider the following embedded system used for controlling a seeing, hearing and moving robot. The system must remove a frame from the video camera (which takes 5 ms) every 50 ms. At least one out of every four frames must be checked to make sure that the robot will not crash into a wall. This check will take 30 ms. The system must examine a sound capture buffer every 70 ms and decide whether it has been told to “stop”. This takes 10 ms. The system checks whether the robot’s feet are in water once every 120 ms and safely shuts itself down if they are. This takes 40 ms.

Assume that you have a preemptive, priority driven scheduler with fixed priorities.

a. Show that the system is schedulable.
Task set is schedulable since total utilization = \( \frac{5}{50} + \frac{30}{200} + \frac{10}{70} + \frac{40}{120} = 0.726 \) is less than 0.757.

b. The designers would like to make the robot more robust by augmenting the frame checking to avoid water as well as walls. However, this will increase the time required for frame checking to 80 ms, without changing the timing requirements. Is the work still schedulable? Prove your answer. You are free to apply transformations to the task set.

If the computation time is increased from 30 to 80. It violates the RMA utilization bound. It also violates the response time bound. So it is not schedulable using statically fixed priorities. However, if detection for water is added, there is no need for the last task and if that is dropped, the task set is schedulable with 0.643 utilization.

c. The designers decide that instead of avoiding water they would like the robot to spend all of its unused computing resources to compute digits of Pi. Considering the original task set, what fraction of the CPU will be available for computing Pi?

The unused capacity must still account for CPU capacity reserve needed to ensure deadlines. So the available fraction is \( 0.757 - 0.726 = 0.031 \) or 3.1%. If you make Pi calculation a nonpreemptive background task, then it is possible to use 100% cpu capacity.
Problem 3 [15 points]: RT Scheduling

Given a set of tasks as below:

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Determine schedule using

a. non-preemptive EDF using FIFO policy

*Using non-preemptive EDF with FIFO, T3 misses the deadline.*

b. preemptive EDF

*It is schedulable using preemptive EDF.*

c. Comment on deadlines for the two cases (a) and (b). Find the best non-preemptive schedule that meets the deadlines.

*You can construct a non-preemptive schedule but do not use FIFO and use the knowledge of deadlines even if the tasks are not released. So the schedule will be: T1 → wait for 1 unit (even though T2 is ready) → T3 → T2.*

Problem 4 [5, 15 points]: Scheduling

Consider the following task set (period, wcet, deadline): {(80, 25, 80), (40, 5, 40), (16, 4, 16)}

a. Is this task set schedulable?

*Yes, it is RM schedulable. Utilization = 0.687*

b. The worst case execution times (WCETs) are specified for a static CMOS processor running at 5 volts with a frequency of 100 MHz. The processor consumes 10 W at 5 volts and 100 MHz operation. Devise the least power implementation that you can using the same processor.
Hint: Look for ways to stop the processor, lower the voltage or lower the frequency. Assume that the gate delay in the technology is proportional to \( \frac{V_{dd}}{(V_{dd} - V_t)^2} \), and \( V_t = 0.8 \, V \).

You can slow down the tasks so that the utilization is below 0.78 when using static priority. Or you can use dynamic prioritization (e.g., EDF) and increase the utilization up to 100%. In that case, it would slow down by a factor of 1.45, running at 69 MHz \( (V_{dd} = 2.7 \, \text{volts}) \) and consume 1.9 W.