Real-Time Programming

Summer Semester 08

Assignment #6

Homepage: `http://www.iste.uni-stuttgart.de/ps/Lehre/SS2008/V_RTP/`
Discussion of Solutions: July 16, 2008

1 Scheduling

The following process set is given:

Process	P	Exec. Seq.	Release Time	T
L_1	4	EEEQVQEE	6	20
L_2	3	QQEQV	4	40
L_3	2	EVWE	4	20
L_4	1	EWVQQE	0	20

E, Q, V, and W stand for one unit of time of execution with Q, V, and W representing mutually exclusive access to resources Q, V, and W, respectively. A sequence QQ is to be read as "the process accesses resource Q for two units of time of execution" and not as "the process accesses resource Q twice for one unit of time each."

The period T indicates when the process happens to become runnable periodically (if it is not yet running).

1. Depict the system behavior under a fixed priority scheme on a uniprocessor by a time line diagram.

2. Depict the system behavior under a simple priority inheritance scheme on a uniprocessor.

   What is the response time of each process? (You do not have to worry about schedulability.)

3. Depict the system behavior under the immediate ceiling priority protocol on a uniprocessor.

   What is the response time of each process?

2 Scheduling, Again

A hard real-time system is to be implemented on a uni-processor. All time-related numbers in this question refer to milliseconds (ms). For your reference, the Response Time Analysis formula in its most general form is:

$$R_i = CS1 + C_i + B_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil \times (CS1 + CS2 + C_j) + \sum_{k \in sporadics} \left\lceil \frac{R_i}{T_k} \right\rceil \times (CSH1 + CSH2 + HC_k)$$

a) A prototype implementation contains the following set of tasks with the indicated characteristics:

Task	**Period (T)**	**Deadline (D)**	**WCET (C)**
A	80	80	40
B	120	120	30
C	500	500	150

This task set is not schedulable. Why? (Provide an answer, using specific number.)

b) In the second prototype, choice of better algorithms results in performance improvements:

Task	**Period (T)**	**Deadline (D)**	**WCET (C)**	**Response Time (R)**
A	80	80	20	
B	120	120	30	
C	500	500	50	

Ignoring the cost of context switches, what are the worst case response times of the three independent tasks under a rate-monotonic priority assigment and a preemption scheduling based on fixed priorities? Fill in the values in the table.

c) The real implementation is being planned. A sporadic task D needs to be added. It is triggered by an interrupt I1 with a worst case inter-arrival time of 1200 ms. The characteristics of the task are:

Task	**Period (T)**	**Deadline (D)**	**WCET (C)**
D (sporadic)	1200	400	90

All tasks receive their static priority by deadline-monotonic priority order (DMPO).

Moreover, tasks A, B, C, and D communicate occasionally using shared resources R1, R2, and R3. Task scheduling in the real implementation will be done using the ICPP protocol. The characteristics of the resources and accesses by tasks are:

Resource	**Used by Tasks**	**WCET of access**	**Priority ceiling**
R1	B,C	10	
R2	B,D	15	
R3	A,B	5	

Assign the priority ceiling to each resource in the best possible manner consistent with the DMPO priorities of the tasks (so that blocking times are minimized). Enter the priority ceiling for each resource in the table above.

Arrange the tasks A, B, C, and D in priority order in the table below and complete alle columns, in particular the last column with the maximum blocking time that the task suffers as a consequence of interference by other tasks concurrently accessing resources.

Task	**Period**	**Deadline**	**Priority**	**WCET**	**Used resources**	**Max. blocking time (B)**
A	80	80	4	20	R3	

d) The interrupt I1 which triggers task D takes a WCET of 1 ms within the interrupt handling procedure itself. In the final system, there is yet another interrupt I2 to be handled. I2 has a worst case inter-arrival time of 500 ms. The activity associated with the interrupt I2 takes a WCET of 10 ms and has a deadline of 15 ms. It is therefore decided to code this activity entirely within the interrupt handler code.

Interrupt	**Worst case inter-arrival time**	**WCET of handler (HC)**	**Handler deadline**	**Interrupt priority**
I1	1200	1	irrelevant	15
I2	500	10	15	10

What is the CPU utilization percentage that is to be expected of the system, taking into account only the WCET of the application code, i.e., all code mentioned above, without context switch times? Give the percentage with an accuracy of one tenth of a percent.

e) Calculate the worst case response times of the interrupt handler I2 and of the tasks A, B, C, and D, taking context switching time into account. The WCET for task context switches in the given system is documented as 1 ms (CS1, CS2); the context switch to and from an interrupt handler is documented as 0,25 ms (HCS1, HCS2). Fill in the response times in the table below.

Response Times:

I2	**A**	**B**	**C**	**D**

f) Bonus points: Provide a percentage for the CPU utilization by the time spent solely in context switching. Briefly describe your reasoning for this answer.