Question 1

Answer:

The authors in this study have analyzed typically two scheduling algorithms for a multiprogramming system and an algorithm which is a combination of the former two. Both the algorithms are priority driven, that is, highest priority task runs first, and preemptive, that is, if a currently running task is interrupted by a higher priority task, then the execution is switched to the higher priority task. Certain assumptions have been made regarding the hard-real-time systems to obtain analytical results of program behavior in such systems, the most important ones being that all the tasks have periodic request and that the run-time of all the tasks are constant. Based on these assumptions, the first scheduling algorithm that has a fixed scheduling policy, known as *Rate-Monotonic Scheduling Algorithm* is defined. In this algorithm, the task with the highest request rate is assigned greatest priority. To determine the schedulability of the algorithm, the *processor utilization factor* is defined as follows:

$$U = \sum_{i=1}^{m} (C_i/T_i)$$

where C_i is the run-time and T_i is the request period of the i^{th} task.

The *least upper bound* of the processor utilization factor is the minimum of the utilization factors of all sets of tasks that fully utilize the processor. For the RM scheduling algorithm, the least upper bound is given as follows:

$$U = m(2^{1/m} - 1)$$

where m is the number of tasks.

Thus if the processor utilization factor is greater than the least upper bound but less than 1, then nothing can be said about the feasibility of the algorithm. Also, for higher values of m, the least upper bound approaches $\ln 2$. Thus to overcome this situation, dynamic task priority assignment was introduced and peculiarly the *Deadline Driven Scheduling Algorithm* is defined and studied. In this algorithm the task with the nearest deadline of its current request is given the highest priority. The deadline driven algorithm is feasible if and only if for a set of periodic tasks, the following condition is satisfied:

$$\sum_{i=1}^{m} \frac{C_i}{T_i} \leq 1$$

that is, the least upper bound on processor utilization factor is 100 percent. Thus a dynamic deadline scheduling algorithm was shown to be globally optimum. Then, a combination of the rate-monotonic scheduling algorithm and deadline driven algorithm was studied which provides most of the benefits of both, rate-monotonic scheduling algorithm and the deadline driven scheduling algorithm.

Question 3

Answer:

According to the authors, in the Rate-Monotonic scheduling policy (RM policy), the task with the highest request rate is given higher priority. It is a fixed scheduling policy in that the priorities of the tasks are fixed. For RM scheduling policy, the *least upper bound* of processor utilization is given as $U_{lub} = m(2^{1/m} - 1)$, where m is the number of tasks. Thus for RM policy, for more than one task, U_{lub} is less than one. For higher values of m, it approaches $\ln(2)$. Also, for a given task set, if the processor utilization is greater than U_{lub} but less than one, then nothing can be said about the feasibility of the RM policy. This situation is overcome by using a dynamic scheduling policy, the *Deadline Driven Scheduling Algorithm*. In this, the tasks are assigned priorities dynamically based on the deadlines of their current requests. The task with the *earliest* deadline for its current request is assigned higher priority. For the deadline driven scheduling algorithm, also known as the *Earliest Deadline First* algorithm (EDF), the $U_{lub} = 1$. Thus the algorithm assures feasibility of tasks if the processor utilization factor is less than or equal to one. Hence the EDF policy is globally optimum in that if a task set can be scheduled by any algorithm, then it can be scheduled by the EDF.

-Rutvij Girish Karkhanis

SID: 103017698