CPEN 432: Homework Assignment 1

Deadline: 11:59 PM, 7 February, 2022

1 Aperiodic Jobs [25 points]

1.1 Synchronous Arrivals [15 points]

We discussed in Lecture 2 Jackson's algorithm to schedule a set of n aperiodic jobs $J = \{J_1, J_2, \dots, J_n\}$ with synchronous arrival times, *i.e.*, where $\forall i : r_i = 0$, on a uniprocessor system and with *no* preemption. We also discussed and proved the optimality of Jackson's algorithm with respect to the maximum lateness of the job set, *i.e.*, with respect to $L_{max} = \max_i(f_i - d_i)$, where f_i and d_i denote the finishing time and the absolute deadline of each job J_i, respectively.

Suppose each job J_i is characterized by an additional parameter $w_i \geq 0$, which denotes the weight or the relative importance of the job. More important jobs have higher weights. For such job sets, like the maximum lateness, the weighted sum of completion times $\sum_{i=1}^{n} w_i f_i$ provides yet another performance metric.

Show that the *weighted shortest processing time first* (WSPTF) algorithm, which schedules jobs in the decreasing order of their weight to computation time ratio w_i/c_i minimizes the weighted sum of completion times and is optimal in this regard. Alternatively, provide a counter-example to show that WSPTF is not optimal.

1.2 Arbitrary Arrivals [10 points]

We discussed in Lecture 3 the earliest deadline first (EDF) algorithm to schedule a set of n aperiodic jobs $J = \{J_1, J_2, \dots, J_n\}$ with arbitrary arrival times on a uniprocessor system *with preemption*. You may assume that job deadlines are unique.

Derive a runtime schedulability test for EDF. That is, if J denotes the current set of active tasks previously guaranteed to be schedulable using EDF, and J_{n+1} denotes a newly arrived task, how can we check at runtime that the new job set $J' = J \cup \{J_{n+1}\}$ is also schedulable?

2 Periodic Jobs [25 points]

2.1 Timeslice Scheduling [10 points]

We discussed timeslice scheduling in Lecture 4. Read through Section 4.2 in the textbook to understand the benefits and drawbacks of timeslice scheduling. How would you schedule the following example using timeslice scheduling?

Task set τ consists of three periodic tasks $\tau = \{\tau_1, \tau_2, \tau_3\}$. The time periods and completion times of the tasks are $T_1 = 25\,ms$, $T_2 = 40\,ms$, $T_3 = 100\,ms$, and $C_1 = 15\,ms$, $C_2 = 6\,ms$, $C_3 = 6\,ms$. You may assume that the tasks arrive synchronously, $i.e.$, $a_1 = a_2 = a_3 = 0$.

2.2 Rate Monotonic Scheduling [15 points]

Rate monotonic (RM) scheduling is a fixed-priority scheduling algorithm for scheduling periodic tasks with preemption. Liu and Layland in their seminal paper (https://cpen432.github.io/resources/P1-liu-layland.pdf) showed that RM is optimal among all fixed-priority algorithms. We sketched the proof of RM's optimality in the lecture.

Provide a detailed proof in three parts: (i) Prove RM's optimality for a task set with two periodic tasks; (ii) Prove RM's optimality for a task set with three periodic tasks; and (iii) Prove RM's optimality for a generic task set with $n > 3$ periodic tasks.

You may assume that the critical instant theorem already holds.