Lecture 21: Fixed scheduling

fixed priority scheduling – all task priorities are fixed prior to scheduler execution and all tasks keep their assigned priorities throughout schedule execution. Note: this kind of scheduling is not conducive to sporadic task scheduling.

Examples:
Generic fixed scheduling – scheduler automatically assigns task priorities (no rules) and are user settable.

Round-Robin scheduling – CPU guarantees all tasks a fair time-slice of the major scheduling period; **no** pre-emption of the fixed priority task schedule by other tasks is done (external interrupt handling excepted). Task priorities are also user settable prior to scheduling.

Rate-Monotonic (R-M) scheduling – the optimal fixed priority scheduler algorithm where the task priority is determined by the task period; the shorter the period, the higher the priority of the task.
Other R-M task scheduling requirements:

- -task periods = their deadlines
- -tasks are pre-emptable by other tasks
- -tasks must be periodic
- -all n task deadlines **will** be met if $Sum_{i=1,n} (e_i/p_i) <= n(2^{(1/n)} – 1)$
- -all n task deadlines **may** be met if $Sum_{i=1,n} (e_i/p_i) <= 1$, otherwise not possible.

Deadline-Monotonic (D-M) scheduling – fixed priority scheduler algorithm where the task priority is determined by the deadline of the task; the shorter the deadline, the higher the priority of the task.
Other D-M task scheduling requirements:

- -task deadlines <= their periods
- -tasks are pre-emptable by other tasks
- -tasks must be periodic
- -all n task deadlines **will** be met if $Sum_{i=1,n} (e_i/d_i) <= 1$, or **may** be met otherwise.

Least-Compute-Time (LCT) scheduling– Tasks with the shortest execution times have the highest fixed priority in the scheduling algorithm.

Fixed Utilization Priority scheduling – Tasks with the greatest CPU utilization (defined as the fraction of the time a task with period p and execution time e keeps a processor busy, namely u = e/p) have the highest fixed priority in the scheduling algorithm.

Exercise:

Task	exe time (e)	period (p)	deadline (d)	Priority (P)
T1	1	2	2	1
T2	2	5	5	2

Can these tasks be successfully scheduled with generic fixed priority scheduling? Draw the Gantt Chart. If the priorities between T1 and T2 are reversed, can generic fixed scheduling and/or round-robin scheduling work? Show the Gantt Charts as proof. Show round-robin scheduling actually has idle time.

Exercise:

Task	exe time (e)	period (p)	deadline (d)
A	3	20	5
B	3	15	7
C	4	10	10
D	3	20	20

What are R-M and D-M priorities here? Does either scheduling algorithm work? (For R-M, forget about the deadlines.)

Exercise:

Task	exe time (e)	period (p)	deadline (d)
T1	3	4	4
T2	2	8	6

What are task priorites using R-M, D-M, LCT, or Fixed Utilization scheduling? Which works and why?