Embedded Systems Summary

Nicolas Trüssel

July 30, 2016

1 Scheduling

1.1 Definitions

- $J = \{j_1, j_2, \ldots, j_n\}$ is a set of tasks.
- a_i or r_i is the arrival / release time of task i.
- d_i is the deadline of task i.
- C_i is the total computation time of task i.
- $c_i(t)$ is the the remaining execution time of task i at time t.
- s_i is the start time of task i.
- f_i is the finish time of task i.
- $L_i = f_i - d_i$ is the lateness of task i.
- $E_i = \max(0, L_i)$ is the exceeding time or tardyness of task i.
- $X_i = d_i - a_i - C_i$ is the laxity or slack of task i.

1.2 Generic Time Triggered Cyclic Executive Scheduler

Let f denote the frame length, P the full period, $D(k)$ the relative deadline of task k, and $p(k)$ the period of task k (how often it occurs). Then the following conditions have to be satisfied:

- $\forall k.f \leq p(k)$ (at most one execution within a frame)
- $P = \text{lcm}_k(p(k))$
- $\forall k.f \geq C_k$ (processes start and complete within single frame)
- $\forall k.2f - \gcd(p(k), f) \leq D(k)$ (between release time and deadline of every task there is at least one frame boundary)

1.3 Aperiodic Scheduling

	Equal arrival, non-preemptive	Arbitrary arrival, preemptive
Independent Tasks	EDD	EDF
Dependent Tasks	LDF	EDF*

1.3.1 EDD

Schedule the tasks in order of non-decreasing deadlines. This minimizes the maximal lateness.

1.3.2 EDF

Always execute the task with the earliest absolute deadline. Schedulability test:

$$\forall i \in [n].t + \sum_{k=1}^{i} c_k(t) \leq d_i$$

1.3.3 LDF

Among all tasks without successors select the task with the latest deadline. Put it in a stack. Repeat until no more tasks. Now execute tasks as they are on the stack.

1.3.4 EDF*

Modify arrival and deadline of each task and use EDF on modified tasks.

$$r_j^* = \max_j \left(r_j, \max_i \left(r_i^* + C_i | J_i \to J_j \right) \right)$$
$$d_i^* = \min_i \left(d_i, \min_j \left(d_j^* - C_j | J_i \to J_j \right) \right)$$