Real-Time Architectures 2003/2004

Earliest Deadline First

Reinder J. Bril

17-05-2004

Earliest Deadline First

- Algorithm
  - dynamic *task* priorities; job with nearest absolute deadline gets highest priority (fixed *job* priorities)
  - preemptive
  - minimize maximum lateness
  - online algorithm
- Advantages
  - optimal algorithm

Earliest Deadline First

- Disadvantages
  - needs priority queue for storing deadlines
    - logarithmic access
  - needs dynamic priorities
  - typically no OS support
  - behaves badly under overload
    - less predictable than RMA (which tasks will miss deadlines ?)
    - a jobs that missed its deadline is allowed to continue (causing a domino-effect of missed deadlines)
  - difficult to handle relative importance
    - critical next to non-critical applications

EDF (cnt'd)

- Schedulability
  - $U^*(EDF) = 1$, i.e. EDF can schedule a set Z of tasks if and only if $U \leq 1$.
- Proof (see book Buttazzo):
  - $U > 1$: "obvious"
  - $U \leq 1$: by means of a contradiction argument; see next slide

4

Reinder J. Bril, r.j.bril@tue.nl
TU/e Informatica, System Architecture and Networking

U(EDF)* = 1

- Suppose not –
  - task set with $U \leq 1$ and not schedulable by EDF

- Choose $[t_1, t_2]$
  - at t_2 overflow occurs
  - t_1 such that:
    - from t_1 onwards continuous utilization ...
    - ...by instances with $[est, dl]$ entirely in $[t_1, t_2]$ (EDF!)
      - hence t_1 is release time of some instance
    - and job experiencing the overload is released in $[t_1, t_2]$
- Define $C_p(t_1, t_2) = (\sum j,i: t_1 \leq est_{j,i}, dl_{j,i} \leq t_2: C_j)$
  - the total amount of computation in $[t_1, t_2]$

5

Reinder J. Bril, r.j.bril@tue.nl
TU/e Informatica, System Architecture and Networking

U(EDF)* = 1

- $C_p(t_1, t_2) = \sum_j \lfloor (t_2 - t_1 - \varphi'_j)/T_j \rfloor C_j$
  $\leq \sum_j \lfloor (t_2 - t_1)/T_j \rfloor C_j$
  $\leq \sum_j ((t_2 - t_1)/T_j) C_j$
  $\leq (t_2 - t_1)U$
- Overflow at t_2, hence: $(t_2 - t_1) < C_p(t_1, t_2)$
- Therefore $(t_2 - t_1) < (t_2 - t_1)U$, and $U > 1$.

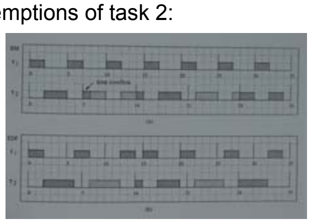
6

EDF (cnt'd)

- Worst-case response times for EDF ?
  - Not needed for (simple) schedulability analysis;
  - not treated in this course; see [Spuri 96].
- Optimality (w.r.t. utilization):
  - EDF is optimal among all dynamic priority preemptive algorithms;
  - EDF is an optimal algorithm.

Example for RMA and EDF

- $C_1 = 2$, $T_1 = 5$, $C_1 = 4$, $T_1 = 7$;
- $U = 2/5 + 4/7 = 34/35 \approx 0.97$
- Schedulable under EDF, not under RMA.
- Number of preemptions of task 2:
  - RMA: 5;
  - EDF: 1 !



References

- [Spuri 96] M. Spuri, *Analysis of Deadline Scheduled Real-Time Systems*, INRIA Report 2772, January 1996.