ECE3140 / CS3420
Embedded Systems

(Aperiodic) Real-Time Scheduling Algorithms

José F. Martínez

Cornell University

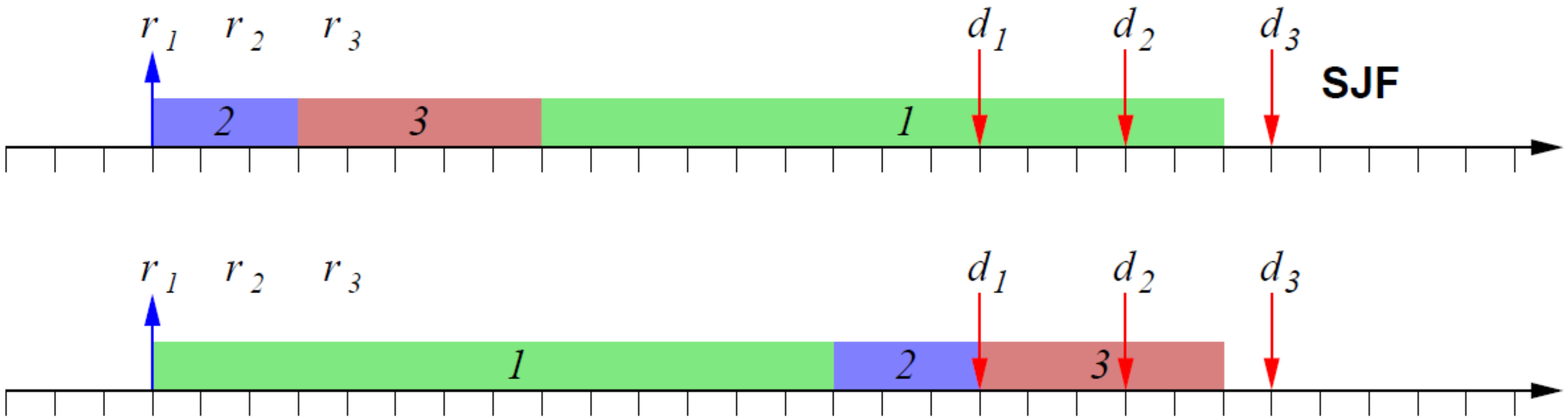
Outline: Real-Time Scheduling

Scheduling algorithms for real-time systems

- Real-time scheduling problem
- Maximum lateness (metric)
- Earliest Due Date (EDD)
- Earliest Deadline First (EDF)
- Reference
  - Chapter 3, “Hard Real-Time Computing Systems Predictable Scheduling Algorithms and Applications” by Giorgio C. Buttazzo (Free electronic copy through Cornell library)

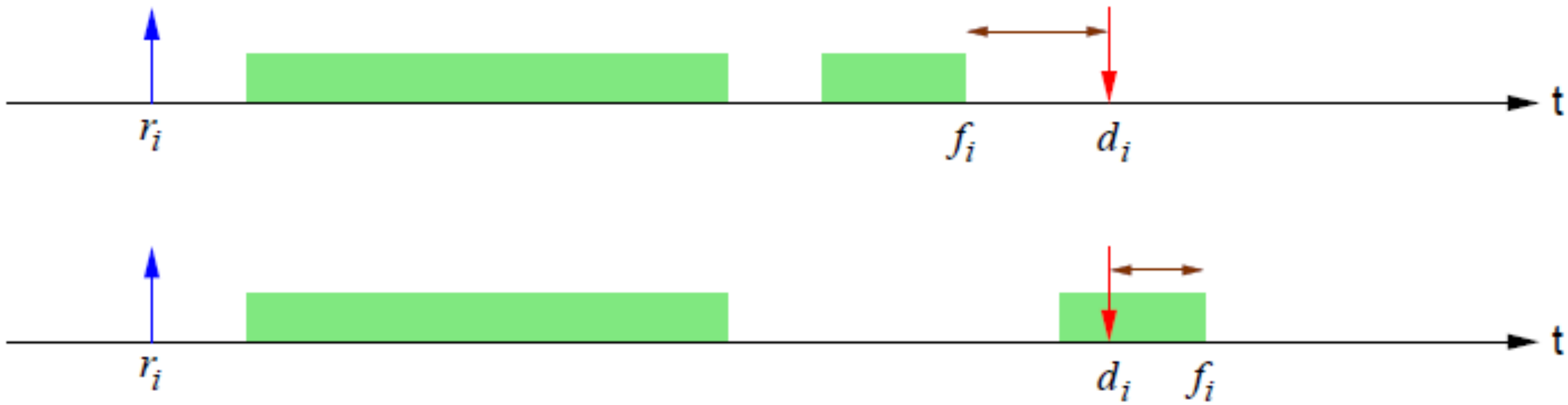
Real-Time Scheduling

- Goal: schedule tasks (jobs) to meet deadlines
- Since the goal is to meet deadlines, we should be using knowledge of deadlines to determine the schedule
  - Absolute deadlines (d_i)
  - Relative deadlines ($D_i = d_i - r_i$)
- Conventional scheduling algorithms are not suitable



Metric Review: Lateness

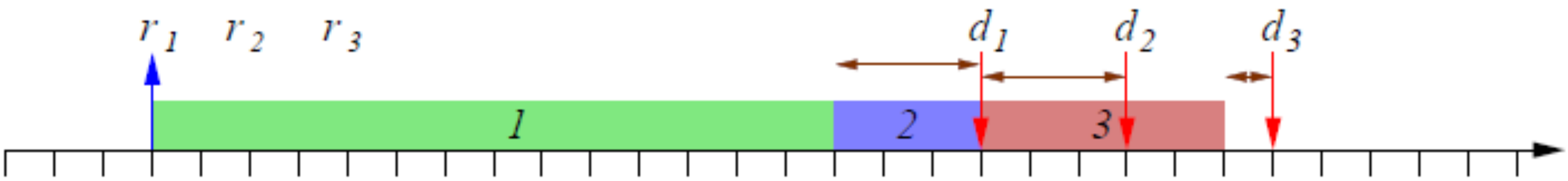
$$L_i = f_i - d_i$$



- Implication for the scheduling
  - $L_i \leq 0$ means that a task finishes before the deadline

Maximum Lateness

$$L_{max} = \max_i(L_i)$$



- $L_{max} \leq 0$ means that no task misses its deadline

Airport Security Line?

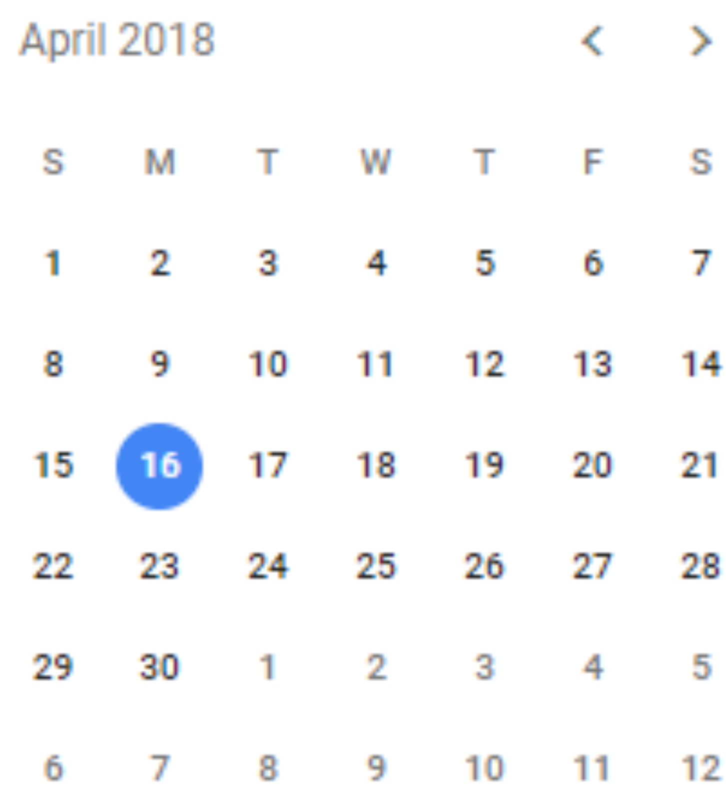
Course Assignments?

ECE3140/CS3420 Assignments

Assignment	Time	Due	Criticality
Pre-proposal	2	4/17	Med
Lab5 plan	1	4/18	Med
Problem set 3	4	4/20	Med
Full proposal	8	4/26	High
Lab5	10	4/27	High
Final project	30	5/17	High

CS3110 Assignments

Assignment	Time	Due	Criticality
A4:JoCalf	10	4/18	High
Prototype	5	5/1	Med
Implementation	15	5/5	Med
Demo	20	5/18	High



Earliest Due Date (EDD)

Strategy: select the task with the earliest due date (deadline)

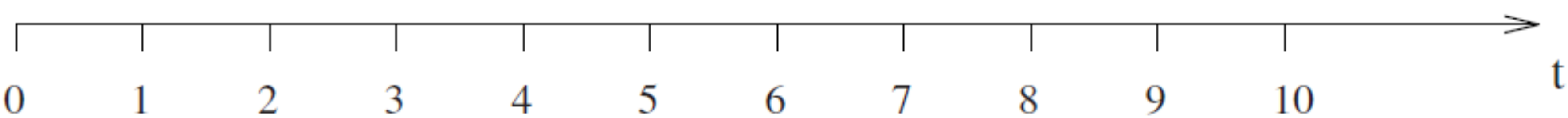
- All tasks arrive at the same time (equal arrival times)
  - Fixed priority (d_i is fixed and known)
  - Preemption is not an issue (non-preemptive)

- *EDD minimizes the maximum lateness* L_{max}

- What is the implication if EDD results in $L_{max} > 0$?

EDD Example

	J_1	J_2	J_3	J_4	J_5
C_i	1	1	1	3	2
d_i	3	10	7	8	5



Jackson's Rule

Jackson's rule: *Given a set of n independent tasks, any algorithm that executes the tasks in order of nondecreasing deadlines is optimal with respect to maximum lateness*

- If $L_{max}(\sigma)$ is the maximum lateness of a schedule, then:

$$\forall\sigma\colon\ L_{max}(\sigma_{EDD}) \leq L_{max}(\sigma)$$

- Why does EDD minimize the maximum lateness? Proof?

Proof Sketch

Given n tasks, show that $\sigma_{EDD} = \tau_1\tau_2 \ldots \tau_n$ where $d_1 \leq d_2 \leq \cdots \leq d_n$ minimizes the maximum lateness ($\max_i(f_i - d_i)$). Assume that the arrival time is zero for all tasks ($r_i = 0$).

Consider a schedule σ that is not EDD, then there exist two consecutive tasks τ_b and τ_a in the schedule ($\sigma = \ldots \tau_b\tau_a \ldots$) with $d_a \leq d_b$.

Step 1: what is the maximum lateness for the two tasks ($L_{max}(a, b)$)?

Step 2: show that switching the two tasks reduces the maximum lateness ($L'_{max}(a, b) < L_{max}(a, b)$)

Step 3: If you repeat the transposition, the schedule converges to EDD in a finite number of steps

Schedulability Analysis

- How can we check if there is a feasible schedule for a task set Γ?
  - We can compute L_{max}!

- A task set is feasible iff $\forall i$: $f_i \leq d_i$

- If we sort the tasks using EDD and all tasks arrive simultaneously, then

$$f_i = \sum_{k=1}^{i} C_k$$

Earliest Deadline First (EDF)

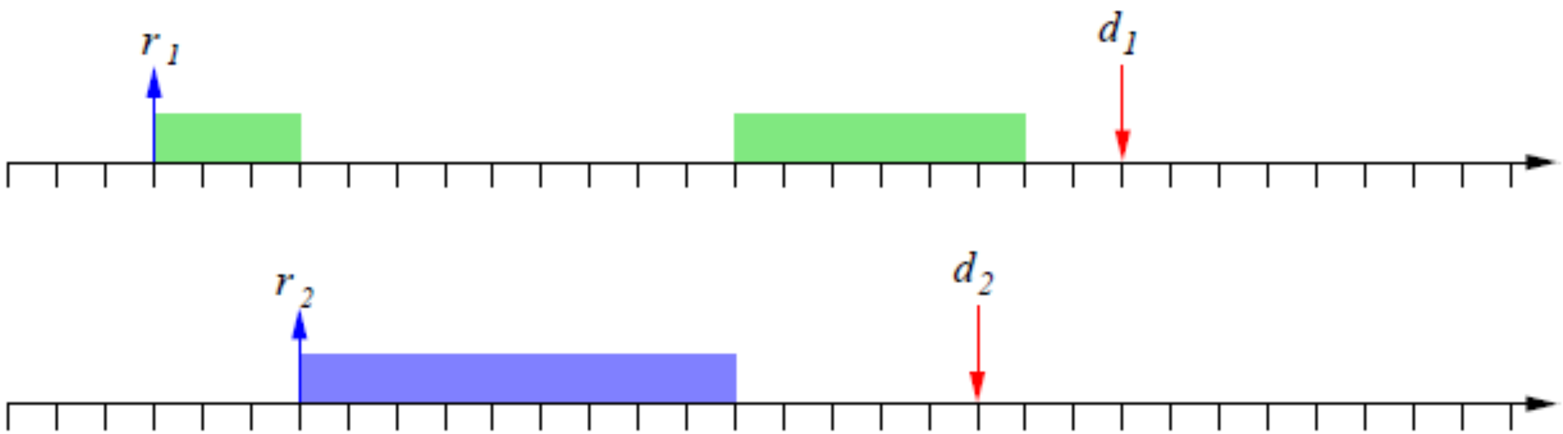
Strategy: select the task with the earliest deadline

- Tasks may arrive at any time
  - Dynamic priority (d_i depends on when the tasks arrive)
  - Preemption is necessary for optimality and may also reduce lateness

- *EDF minimizes the maximum lateness* L_{max}
  - *With the preemptive scheduling*

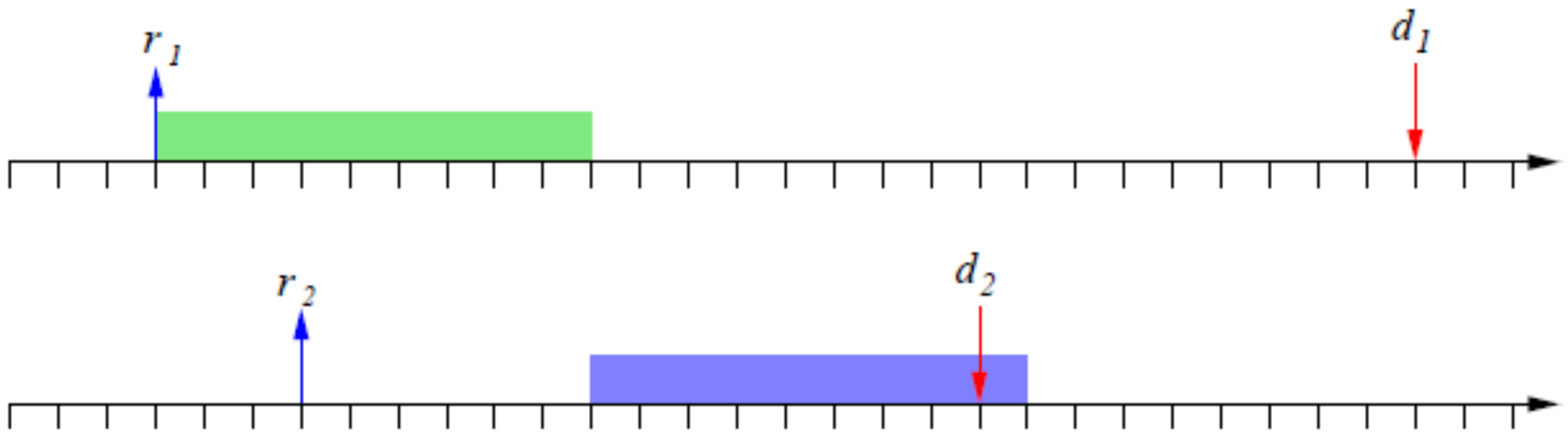
EDF Example

- Tasks that arrive with earlier deadlines pre-empt tasks with later deadlines



EDF with Non-Preemptive Scheduling

- Under non-preemptive scheduling, EDF is not optimal



EDF with Non-Preemptive Scheduling

- . . . unless the algorithm has knowledge of the future!