Embedded System Design and Modeling

IX. Scheduling basics

Scheduling

- Scheduling = the process of arranging the execution of a set of **tasks** which need to be run on the same processing device
  - i.e. decide which task is run when, for how long, etc.
- Encountered in multi-tasking systems
- *Note*: Slides are heavily based on Prabal Dutta & Edward A. Lee, Berkeley 2017

Task

- A **task** is a set of operations which have:
  - release (arrival) time: earliest time when it can be run
  - **start time**: actual starting time
  - **finish time**: actual ending time
  - execution time: actual running time, excluding any interruptions
  - **deadline**: latest time by which a task must be completed
- Tasks may be interrupted by higher priority tasks, when priorities are defined
- Tasks may be periodic (e.g. every 10ms) or aperiodic

Task

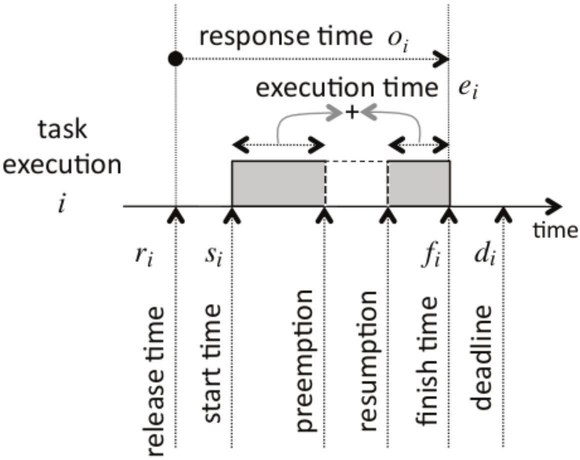


Figure 12.1: Summary of times associated with a task execution.

Scheduling

- How to decide which task to run when?

Considerations:

- Preemptive vs. non-preemptive scheduling
- Periodic vs. aperiodic tasks
- Fixed priority vs. dynamic priority
- Priority inversion anomalies
- Other scheduling anomalies

Preemptive vs. non-preemptive

- Non-premmptive: once started, no task can be interrupted until it finishes
- Preemptive: a task can be interrupted
  - the kernel decides when
- Preemptive scheduling:
  - Every task has a **priority**
  - At any instant, the task with the **highest priority** is executed
  - Any high priority task takes precedence over a low priority task

Rate Monotonic Scheduling (RMS)

- Given N **periodic** tasks
- **Rate Monotonic Scheduling (RMS)**: assign task priority by period: smaller period has higher priority

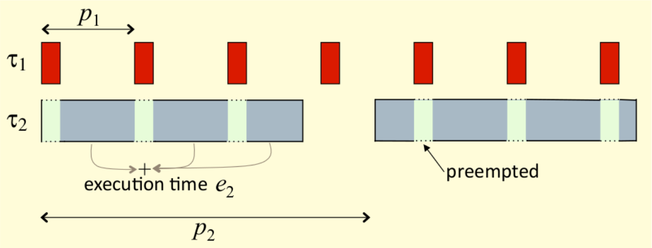


Figure 12.3: Two periodic tasks $T = \{\tau_1, \tau_2\}$ with a preemptive schedule that gives higher priority to τ_1.

[1]

[1] image from Lee&Sheshia book

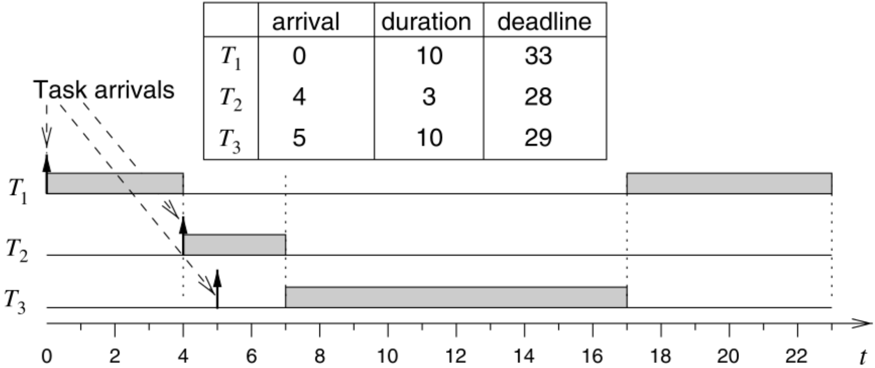
Optimality of RMS

- A **feasible schedule** = all task finish times are before their deadlines
  - no deadline is exceeded
- **Theorem**: If the set of N tasks can be arranged to form a feasible schedule, then the RMS scheduling is feasible.

Earliest Deadline First (EDF)

- Given N non-periodic independent tasks with arbitrary arrival times and deadlines
- **Earliest Deadline First (EDF) scheduling: execute the task with the earliest deadline among all available tasks
- Note: If a new task that just arrived can interrupt the current task, in case it has an earlier deadline

Earliest Deadline First (EDF)



	arrival	duration	deadline
T_1	0	10	33
T_2	4	3	28
T_3	5	10	29

Figure 6.6. EDF schedule

[2]

[2] image from "Embedded System Design" 2nd edition, Peter Marwedel, Springer 2011

Optimality of EDF

- **Theorem**: EDF scheduling minimizes the maximum lateness of the tasks
- The **maximum lateness** of a set of N tasks is:

  $$L_{max} = \max(f_i - d_i)$$

  i.e. the maximum exceeding of a deadline

  (the maximum lateness can be negative, i.e. when no task deadline is exceeded, and in this case it acts as a safety time margin)
- EDF makes the maximum exceeding of deadline as small as possible, or, if no deadline is exceeded, EDF maximizes the safety margin between the finish time and the dealine