

Real-time Scheduling

Types of Deadlines:

- **Hard-deadline:** Must be met to prevent damage/error, or for output to make sense.
- **Soft-deadline:** Deadline, if not met, will provide results that are semi-relevant.

Types of Tasks:

- **Periodic Tasks:**
 - Tasks that happen at regular intervals
 - Generally *hard-deadlines* e.g.
 - C_k is the computation time, and T_k is the period.
 - The system is *overloaded* when:
 -
- **Aperiodic:**
 - Tasks with no period. Generally occur in response to some random event.
 - Generally *soft-deadlines*, because difficult to guarantee completion.
 - Must not interfere with *hard-deadlines* of other processes.
- **Sporadic Tasks:**
 - Aperiodic tasks with *hard-deadlines*.
 - Requires protocol ensuring there's at least T_k time b/w consecutive occurrence of task.
 - Sporadic tasks can overload systems
 - Some not even scheduled if impossible to meet deadlines.

Task Scheduling Algorithms:

NOTE: 'pre-emptive' means the currently running process can be interrupted by another process.

- **Earliest Deadline First:**
 - Choose task with soonest deadline
 - Break tie using random selection
 - If a schedule fulfills all deadlines exist, it'll be produced
 - Higher priority task generated in real-time with pre-empt current task
- **Least Slack First:**
 - *Slack*: how long a process can wait before being scheduled to meet its deadline.
 - *Slack* gives indication of which tasks in danger of missing deadline(s).
 - We prefer tasks to start as early as possible.

Commercial Task Scheduling:

- **UNIX:**
 - Multi-level, pre-emptive (i.e. stopped after time-slice expiration), Round-robin on each level.
 - Default time-slice duration is 1s.
 - **Priority:** lower the value, higher the priority.
 - CPU utilization at time-slice no. i is calculated as:
 - Priority at interval i calculated as:
 - B_j is the initial priority that process was started with
 - N_j is the 'nice' value that can be modified.
 - CPU_j and N values are limited s.t. the no user process has higher priority than UNIX's system processes.
- **Windows:**
 - Priority-based, pre-emptive scheduling

- Process runs long as its pre-empted, blocked, terminated, or time-slice expires
- Scheduler/Dispatcher maintains a queue to each priority level (32 different levels).
 - ...and goes through them from highest-to-lowest
- **Priority:** higher the value, higher the priority.
 - User can change priority level of a process to:
 - Realtime
 - High
 - Above Normal
 - Normal (a process is *usually* here)
 - Below Normal
 - Low
 - A pre-empted process' priority will be lowered (*unless* it's realtime)
- Higher priority given to foreground task(s), and the task *atop* in the UI.
 - Task on the *top* of the UI also gets *longer* time-slices.