Real-Time Systems

- Hard real-time system
  - Stringent requirements on deadlines for task completion
- Soft real-time system
  - Real-time task will receive priority but no absolute guarantee of completion is provided

Common Characteristics

- Single purpose – e.g., controlling anti-lock brakes, airline navigation, delivering music on an MP3 player, …
- Small size – space- or power-constrained environments
- Specific timing requirements

Implementing Real-Time Operating Systems

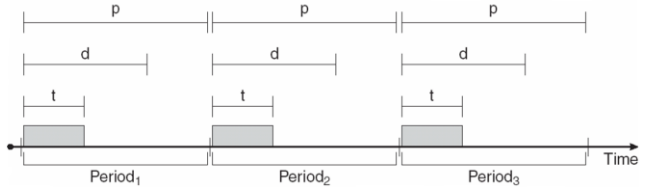
- In general, real-time operating systems must provide:

(1) Preemptive, priority-based scheduling

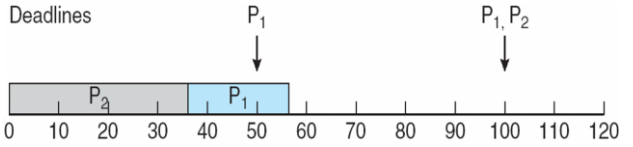
(2) Preemptive kernels

(3) Latency must be minimized
- interrupt latency
- dispatch latency

Real-Time CPU Scheduling

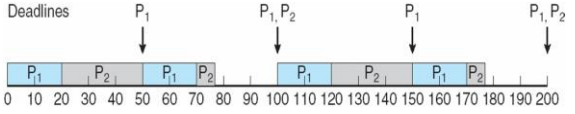
- Periodic processes require the CPU at specified intervals (periods)
- ***p*** is the duration of the period
- 

Scheduling of tasks when P_2 has a higher priority than P_1

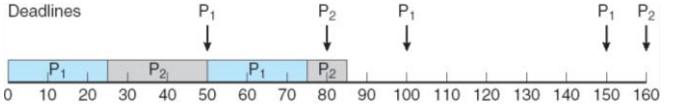


P1: period=50, processing time = 20, deadline = period
P2: period=100, processing time = 35, deadline = period

Rate Monotonic Scheduling

- A priority is assigned based on the inverse of its period
- Shorter periods = higher priority;
- Longer periods = lower priority
- 

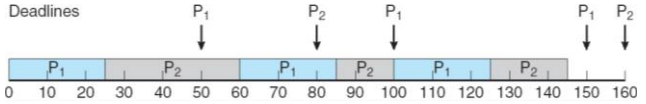
Missed Deadlines with Rate Monotonic Scheduling



Earliest Deadline First Scheduling

- Priorities are assigned according to deadlines:

  the earlier the deadline, the higher the priority;



Disclaimer

- Parts of the lecture slides were derived from those by Kai Shen, Willy Zwaenepoel, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Andrew S. Tanenbaum, and Gary Nutt. The slides are intended for the sole purpose of instruction of operating systems at the University of Rochester. All copyrighted materials belong to their original owner(s).