CISC 3320
C14b. Real-time Scheduling

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Acknowledgement

- These slides are a revision of the slides provided by the authors of the textbook via the publisher of the textbook.

Outline

- Real-Time CPU Scheduling

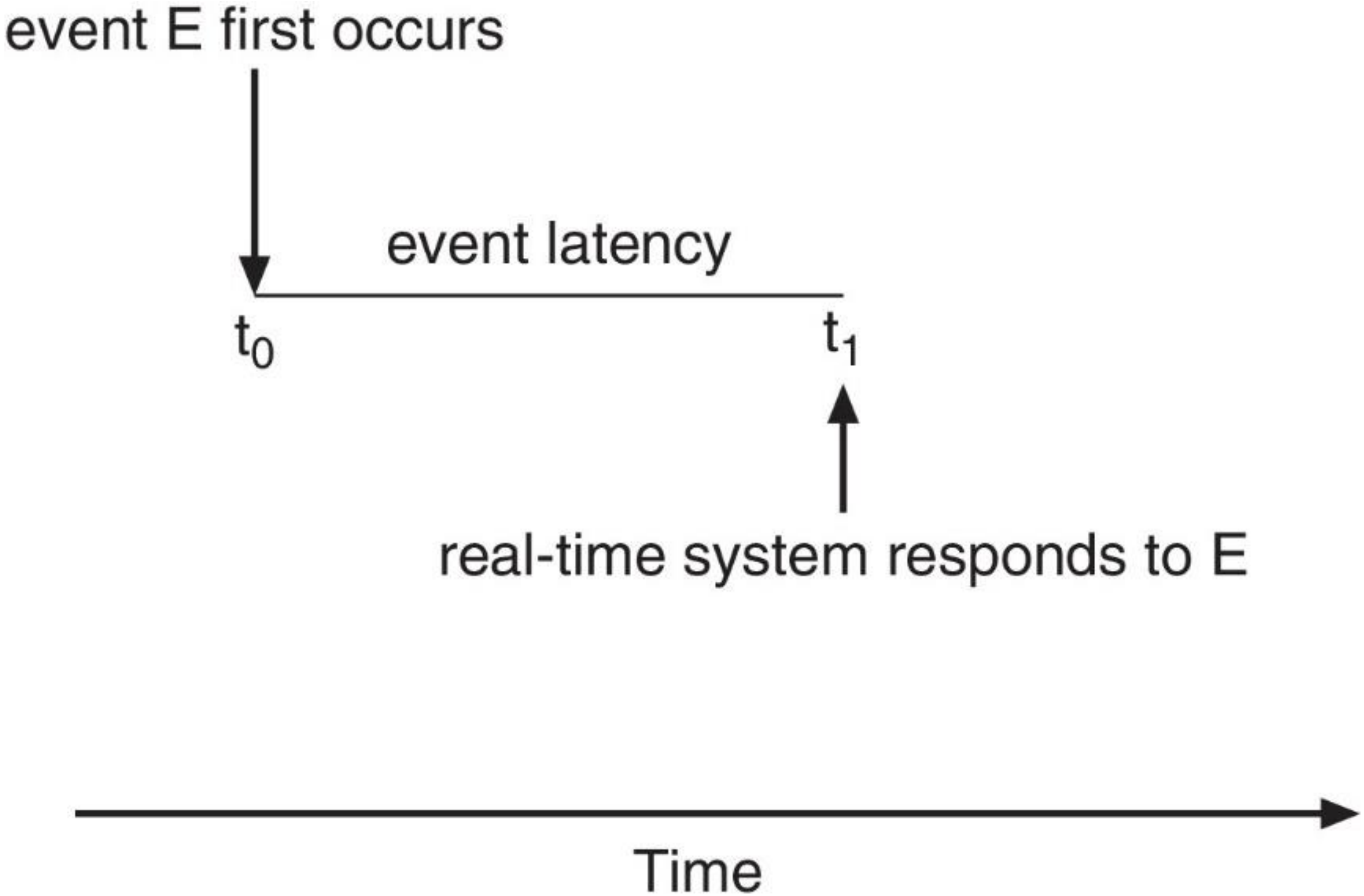
- Operating Systems Examples
- Algorithm Evaluation

Real-time Systems

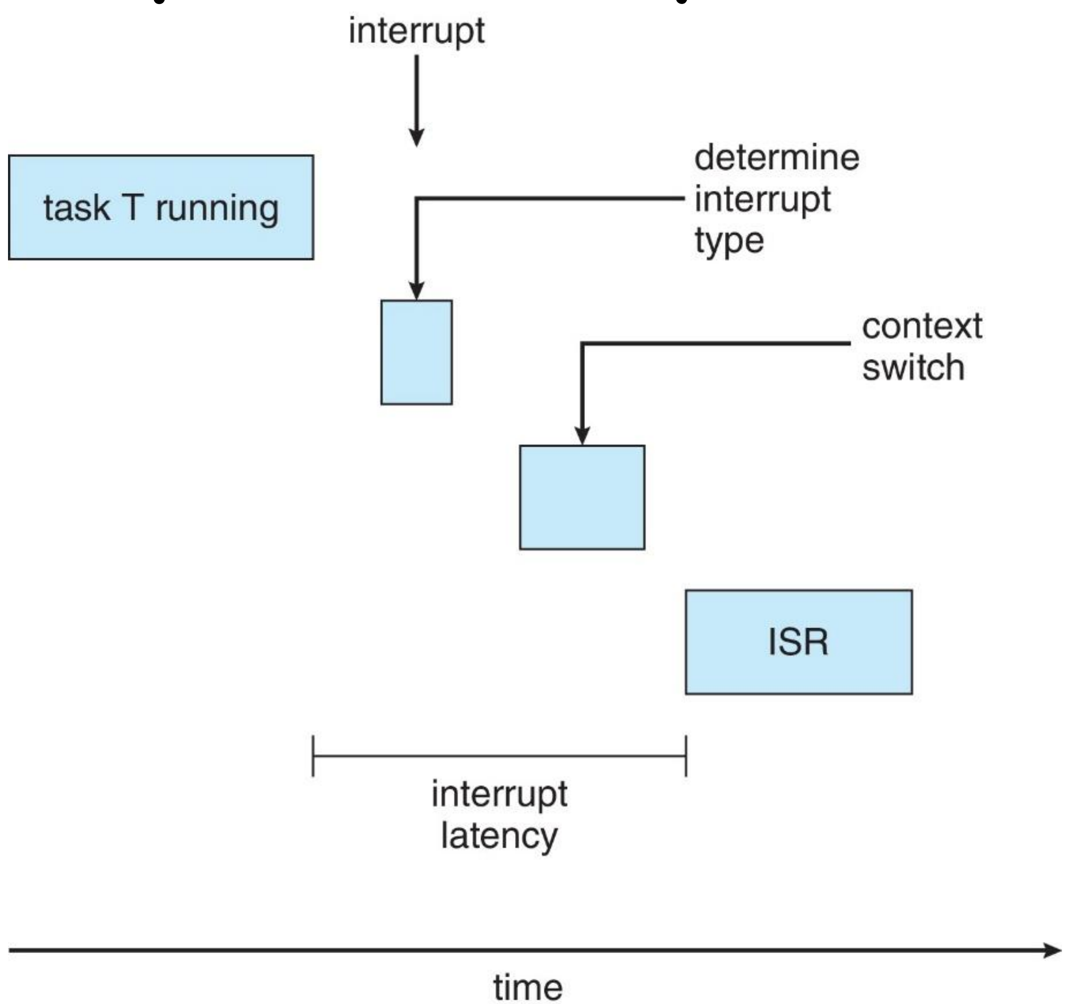
- Soft real-time systems
  - Critical (real-time) and noncritical threads
  - They guarantee only that the thread will be given preference over noncritical threads
  - They provide no guarantee as to when a critical thread will be scheduled
- Hard real-time systems
  - A thread must be serviced by its deadline
  - Service after the deadline has expired is the same as no service at all.

Real-Time CPU Scheduling

- Event latency
  - The amount of time that elapses from when an event occurs to when it is serviced.
- Two types of latencies
  - Interrupt latency
    - time from arrival of interrupt to start of routine that services interrupt
  - Dispatch latency
    - time for schedule to take current process off CPU and switch to another

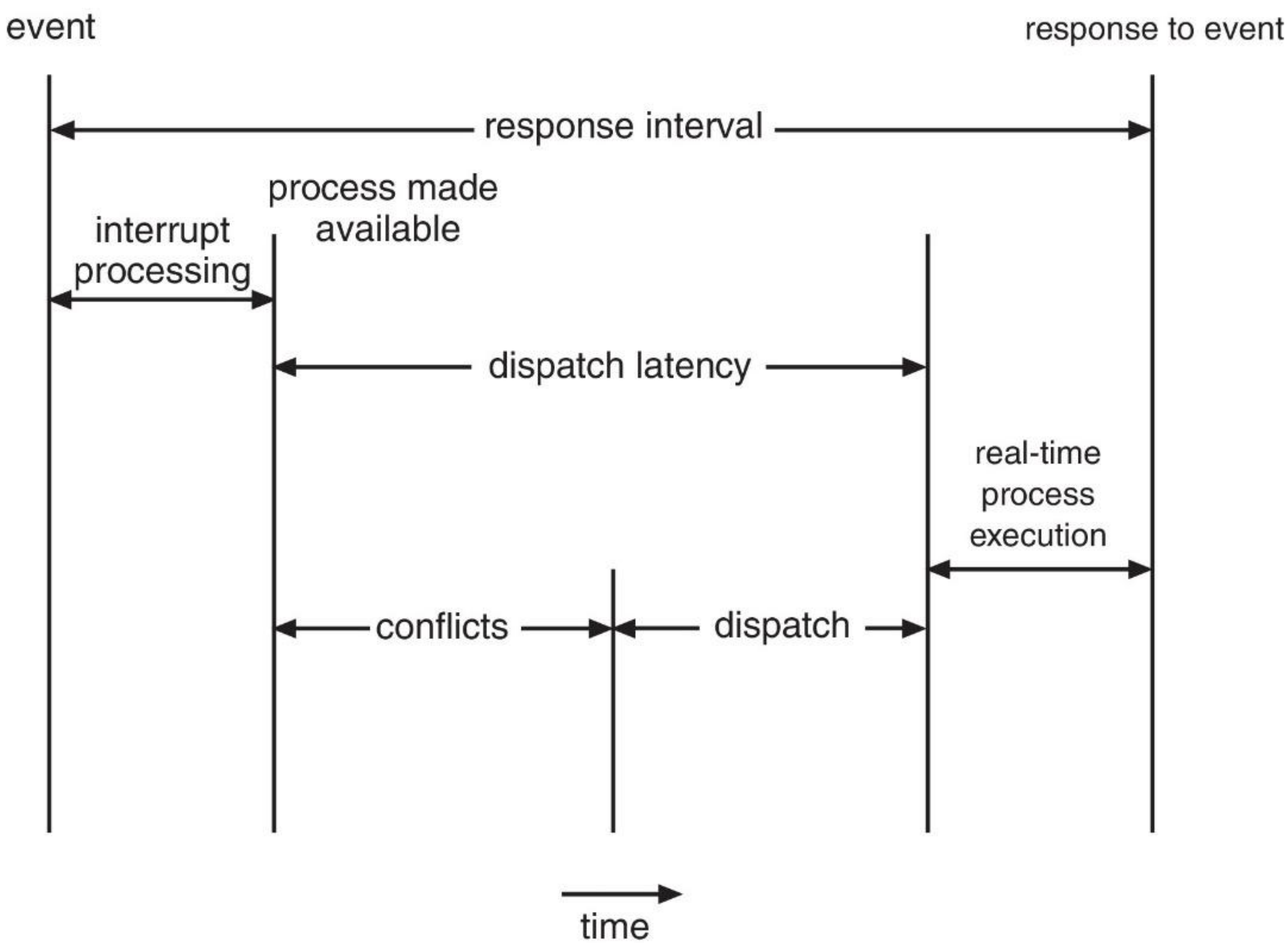

event E first occurs
event latency
t0
t1
real-time system responds to E
Time

Interrupt Latency



Dispatch Latency

- Conflict phase of dispatch latency:
  1. Preemption of any process running in kernel mode
  2. Release by low-priority process of resources needed by high-priority processes


event
response to event
response interval
process made available
interrupt processing
dispatch latency
real-time process execution
conflicts
dispatch
time

Questions?

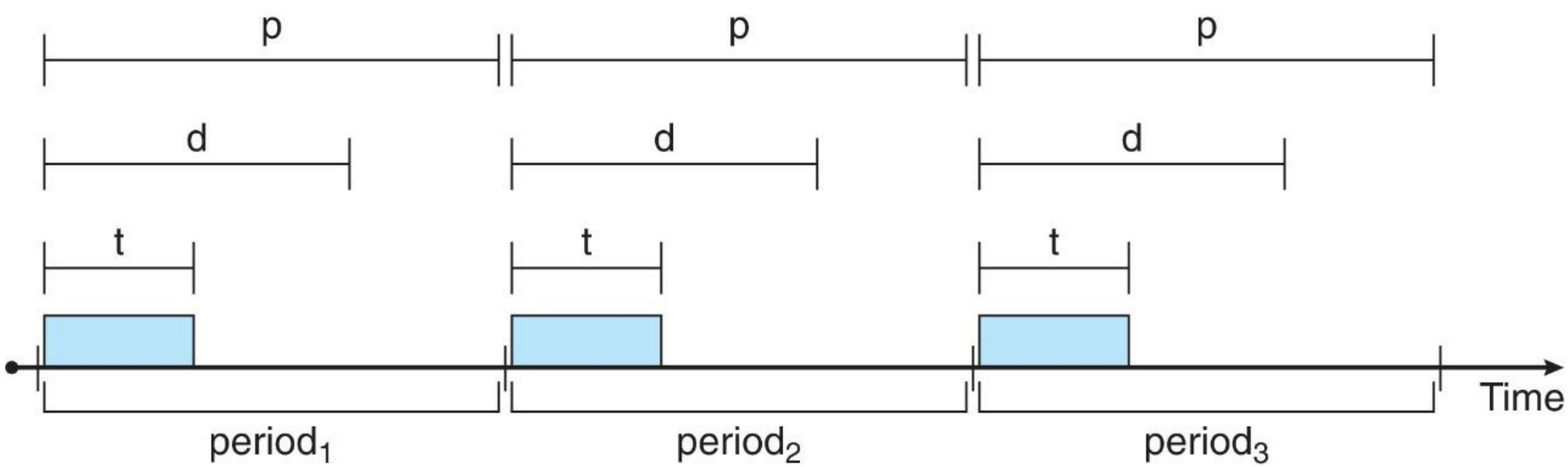
- Concept of real-time systems
- Scheduling for real-time systems
- Event latencies
  - Interrupt latency
  - Dispatch latency

Real-Time Scheduling

- Priority-based scheduling
- Rate monotonic scheduling
- Earliest Deadline First Scheduling
- Proportional Share Scheduling
- Example
    - POSIX real-time scheduling

Priority-based Scheduling

- For real-time scheduling, scheduler must support preemptive, priority-based scheduling
  - But only guarantees soft real-time
- For hard real-time must also provide ability to meet deadlines
- Processes have new characteristics: periodic ones require CPU at constant intervals
  - Has processing time t, deadline d, period p
  - $0 \le t \le d \le p$
  - Rate of periodic task is $1/p$


p
p
p
d
d
d
t
t
t
Time
period1
period2
period3

Rate Monotonic Scheduling

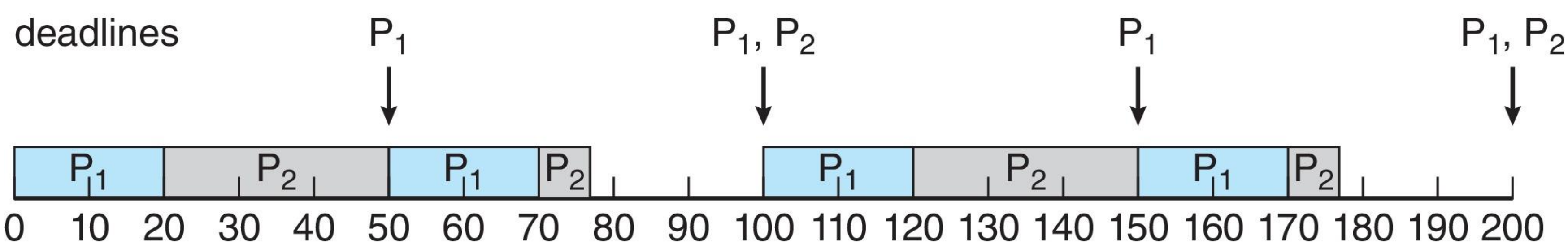
- Uses a static priority policy with preemption.
  - A high-priority process always preempts the lower-priority a higher-priority
  - Each process is assigned a priority based on the inverse of its period (the period at which the process requires a CPU)
    - i.e., to assign a higher priority to tasks that require the CPU more often.
- Assumes that the processing time of a periodic process is the same for each CPU burst.

Example 1

- Two processes, P1 and P2
  - P1's period: p1 = 50; processing time t1 = 20; high priority
  - P2's period: p2 = 100; processing time t2 = 35; low priority
  - Deadline: it complete its CPU burst by the start of its next period.
- Is it possible to meet the deadline?

Example: Meeting Deadlines

- P1's period: p1 = 50; processing time t1 = 20; high priority
- P2's period: p2 = 100; processing time t2 = 35; low priority

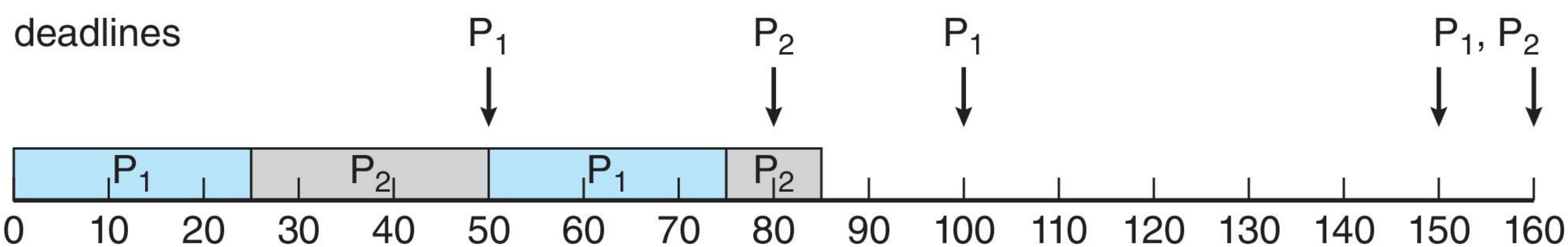


Example 2

- Two processes, P1 and P2
  - P1's period: p1 = 50; processing time t1 = 25; high priority
  - P2's period: p2 = 80; processing time t2 = 35; low priority
  - Deadline: it complete its CPU burst by the start of its next period.
- Is it possible to meet the deadline?

Example: Missed Deadlines

- P1's period: p1 = 50; processing time t1 = 25; high priority
- P2's period: p2 = 80; processing time t2 = 35; low priority
- Process P2 misses finishing its deadline at time 80



Earliest Deadline First Scheduling (EDF)

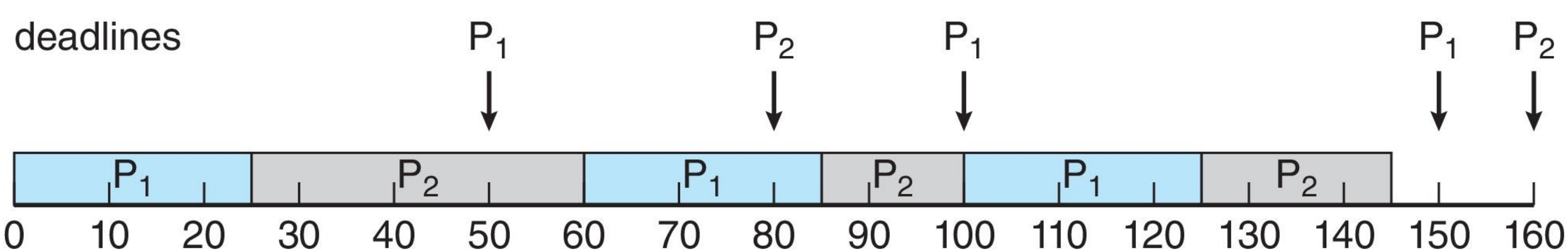
- Priorities are assigned according to deadlines:
  - the earlier the deadline, the higher the priority;
  - the later the deadline, the lower the priority

Example 3

- Two processes, P1 and P2
  - P1's period: p1 = 50; processing time t1 = 25;
  - P2's period: p2 = 80; processing time t2 = 35; low priority
- Priority is assigned based on deadlines and changes over time.

Example 3

- Initially, P1 has an earlier deadline than P2, i.e., P1 has high priority, and P2 low priority
  - P2 is allowed to complete its CPU burst or to reach P1's deadline



Proportional Share Scheduling

- *T* shares are allocated among all processes in the system
- An application receives *N* shares where *N* < *T*
- This ensures each application will receive ***N*** / *T* of the total processor time

Example: POSIX Real-Time Scheduling

- The POSIX.1b standard
- API provides functions for managing real-time threads
- Defines two scheduling classes for real-time threads:
  - SCHED_FIFO - threads are scheduled using a FCFS strategy with a FIFO queue. There is no time-slicing for threads of equal priority
  - SCHED_RR - similar to SCHED_FIFO except time-slicing occurs for threads of equal priority
- Defines two functions for getting and setting scheduling policy:
  - pthread_attr_getsched_policy(pthread_attr_t *attr, int *policy)
  - pthread_attr_setsched_policy(pthread_attr_t *attr, int policy)

Questions?

- Priority-based scheduling
- Rate monotonic scheduling
- Earliest Deadline First Scheduling
- Proportional Share Scheduling
- POSIX example
    - Is POSIX real-time scheduling a hard or a soft real-time scheduling system?