CS323 Operating Systems
Multimedia II

Yuanyuan Zhou

Lecture 30

4/11/2003

Content of this lecture

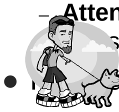
- Reminder
  - Quiz4 is due today
  - Conflict exam: signup before 4/14 Noon unless documented medical or family emergency
- Multimedia Scheduling
  - Traditional RT Scheduling
  - System Model of RT Scheduling
  - Rate Monotonic Algorithm
  - Earliest Deadline First Algorithm

Multimedia: Real-Time Processing Requirements

- Occurs in predetermined - usually periodic – intervals
- Compete at certain deadlines.
- RT process manager
  - performs admission control
  - determines a schedule
  - gives processing guarantees.
- Problem: how to find a feasible schedule.

Group Discussion

- Tasks
  - **Sleep:** once every 24 hours, each time takes 8 hours
  - **Walk the dog**: once every 12 hours, each time takes 30 minutes
  - **Call parents**: once every 6 hours, each time takes 15 minutes
  - **Eat:** once every 4 hours, each time takes 1 hour
  - **Attend lectures**: once every 3 hours, each time s 1 hour

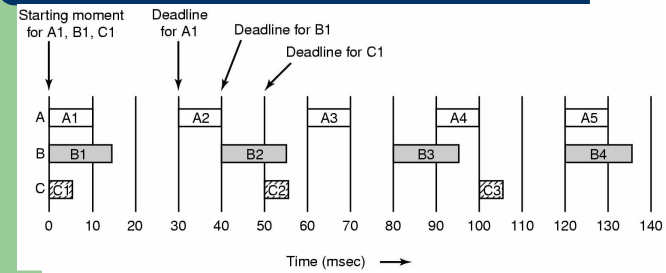


- an you schedule these tasks?

Goals of Multimedia Scheduling

- non-RT process should not starve when RT process is running
- RT process should not be subject to priority inversion.
  - (what is priority inversion?)

Multimedia Process Scheduling



- Periodic processes displaying a movie
- Frame rates and processing requirements may be different for each movie

Discussion

Does the following scheduler have the same goals?

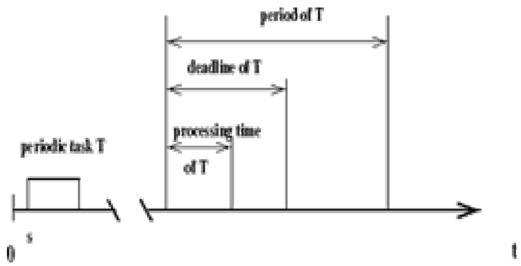
- Scheduler for traditional time-sharing computers
- Traditional RT scheduler
- Multimedia scheduler

Scheduling Environments

- Traditional scheduling on time-sharing computers
  - optimal throughput, optimal resource utilization, and fair queueing.
- Traditional RT scheduling in operation research
  - guarantee *hard deadlines* with no adaptation to changes of the workload.
- Multimedia scheduling
  - *Guarantee soft deadlines*,i.e., work in a dynamic environment with adaptation to changes of the workload in bounds of timing requirements

System Model in RT Scheduling

- Task is the schedulable unit of the system.
- Task is characterized by (1) timing constraints and resource requirements.



Assumptions and Notation

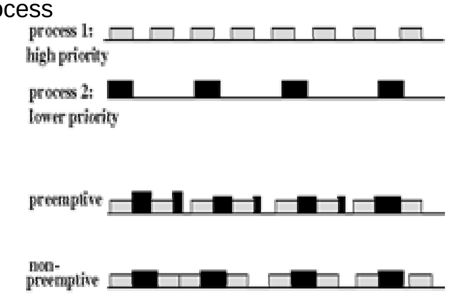
- **Assumptions** : periodic tasks are mutually independent.
- **Time Constraints** : *(s, e, d, p)* where *s* is starting point, *e* is processing time, *d* is deadline, and *p* is period.
- If the period at *(k-1)* step is equal to ready(start) time of period *k*, then we get *congestion avoidance deadline*.
- **Tasks** : can be *preemptive* or *non-preemptive*.
- **Multimedia Scheduling Algorithm** : must determine sharing of the resource used by different tasks so that all of them can meet their deadlines.

Schedulability Test - Performance Metric

- Scheduling algorithm guarantees a newly arrived task a deadline if the algorithm can find a **feasible schedule**.
- **Guarantee ratio** := number of guaranteed tasks/total number of tasks
- Another performance metric is **process utilization** with $U = \sum_{i=1}^{n} \frac{e_i}{p_i}$

Rate-Monotonic Scheduling

- Process priority is determined by the task rate
- Higher priority process can preempt lower priority process



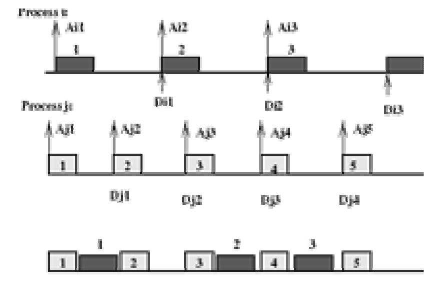
=> Relative Stream priority is fixed

Rate Monotonic Scheduling

- This algorithm was designed and proved correct by C.L. Liu and Layland in 1973.
- This algorithm is static and optimal, priority-driven for preemptive periodic jobs.
  - Optimal means that there is no other static algorithm that is able to schedule a task set which can't be scheduled by the rate-monotonic algorithm.
- The algorithm was proven under the following assumptions:
  - tasks are periodic
  - each task must be completed before the next request occurs
  - all tasks are independent
  - run-time of each task request is constant
  - any non-periodic task in the system has no required deadlines

Deadline-Based Scheduling

- Process priority is determined by the process deadline.

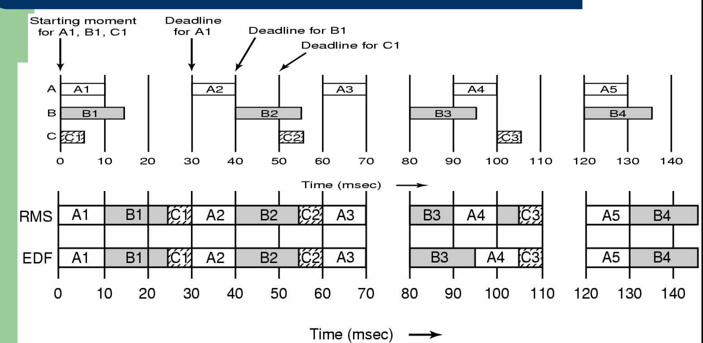


Both streams scheduled according to their deadlines

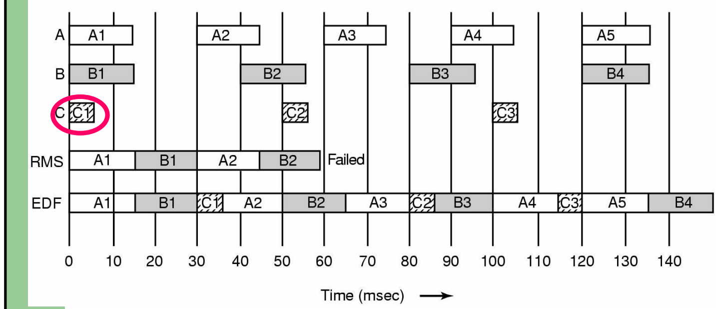
Earliest Deadline First Algorithm

- One of the best known algorithms for RT processing
- It is an optimal dynamic algorithm. It produces a valid schedule whenever one exists.
- Upper bound of process utilization is 100 %.
- If priorities should be used, then the earliest deadline gets the highest priority.

Earliest Deadline First Scheduling (1)



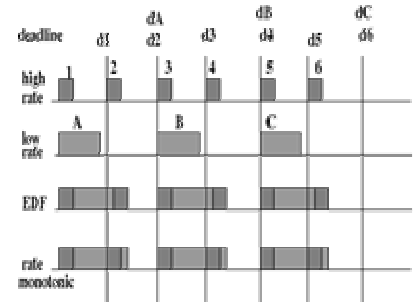
Earliest Deadline First Scheduling (2)



Another example of real-time scheduling with RMS and EDF

Comparison of RMS and EDF

- In terms of context switching, the **EDF is better** if more than one stream are processed concurrently



Admission Control

- Process utilization for RM algorithm is upper bounded by $U \le ln2$ (Liu and Layland condition).
- The schedulability test is then $\sum_{i=1}^{n} \frac{e_i}{p_i} \le ln2$
- With EDF algorithm, the 100% utilization can be achieved : $U \le 1$
- The schedulability test is then $\sum_{i=1}^{n} \frac{e_i}{p_i} \le 1$