Virginia Tech

Real-Time Scheduling

CS 3204 – Operating Systems
Lecture 20
3/3/2006

Shahrooz Feizabadi

Real-Time Systems

Systems that must service requests within a precise time constraint:

- Air traffic control systems
- Railway switching systems
- Industrial automation
- Robotics
- Military

Generally, systems that must respond to external physical phenomena.

Tracking System Example

- Dynamic trajectory calculations
- False alarms
- Verification
- System response
- Real-time clock
- 687m error!

System Resources

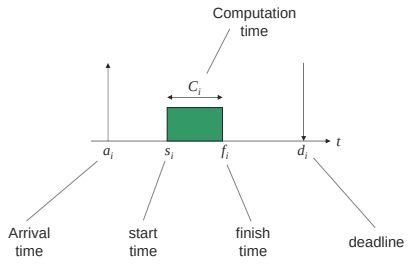
Different approaches to resource management:

- General-purpose OS
  - Extensible time horizon: all tasks *eventually* finish
  - Similarly, elastic memory: demand paging
  - Fairness
  - Non-starvation
- Real-Time system
  - Fixed resources
  - Tasks adjusted accordingly
  - Predictability

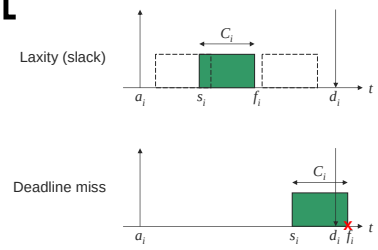
Schedulers

- Preemptive / non-preemptive
- Static / dynamic
- On-line / off-line
- Optimal / heuristic
- Clairvoyance

Real-Time Task Parameters



Real-Time Tasks



RMA

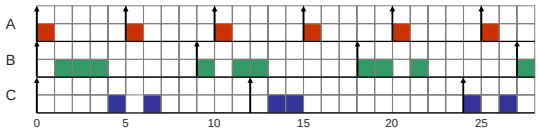
- Rate Monotonic Algorithm
- Static
- Preemptive
- Fixed-priority assignment
- Periodic tasks
- Guarantee-based
- Hard real-time correctness – all deadlines always met

RMA (continued)

RMA: Assign unique priorities, in descending order, to tasks ordered by ascending periods: the shorter the period, the higher the priority.

RMA – Example

Task	T	C	P
A	5	1	3
B	9	3	2
C	12	2	1



RMA (continued)

- How far can we push this?
- Constraints
  - Number of tasks?
  - Periods?
  - Computational times?

Processor Utilization Factor

Fraction of CPU time spent in execution of tasks:

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i}$$

Otherwise stated: sum of the individual relative load contribution of each task in the task-set.

Provides an indication of system-wide computational load.

Feasibility Test

RMA guarantee: No deadlines will be missed under rate monotonic scheduling if the following condition holds:

$$\sum_{i=1}^{n} \left(\frac{C_i}{T_i} \right) \leq n(2^{1/n} - 1)$$

Feasibility Test (continued)

- Note that:

$$\lim_{n \to \infty} n(2^{1/n} - 1) = \ln 2 \simeq 0.69$$

- That is, any task-set with a CPU utilization factor of 69% or less is schedulable under RMA
- Sufficient, but not necessary condition

Optimality

RMA is an optimal fixed-priority assignment algorithm: no other fixed-priority algorithms can schedule a task set that cannot be scheduled under RMA.

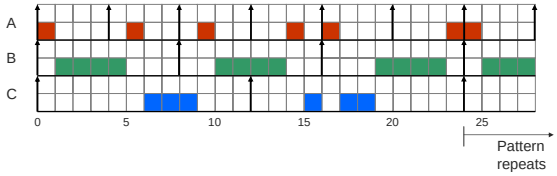
EDF

- Earliest Deadline First
- Earlier the deadline, higher the priority.
- Dynamic assignment
- On-line
- No periodicity assumptions – can schedule aperiodic tasks.

EDF – Example

Task	T	C
A	4	1
B	8	4
C	12	3

Assume deadline equals period (T).



EDF Properties

- Feasibility test: $\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1$
- $U = 100\%$ in example
- Bound theoretical
- Sufficient and necessary
- Optimal

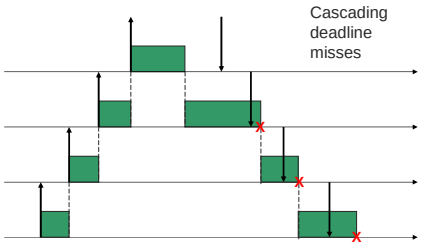
Overload

- Hard real-time
  - No deadlines misses.
  - Dimension system to meet exact timing requirements.
- Soft real-time
  - May encounter transient or sustained overload.
  - Design algorithm to achieve desired temporal behavior.

Real-Time Overload Scheduling

- Runtime uncertainty
- Dynamic system
- Aggregate demand exceeds capacity
- Fixed available bandwidth
- Must shed load
- Starvation
- Who survives?

EDF Domino Effect

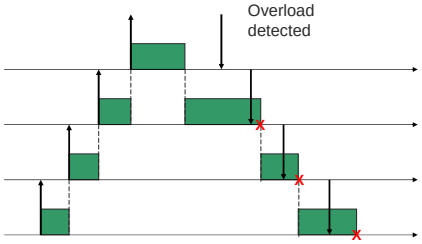


Deadline sole scheduling parameter

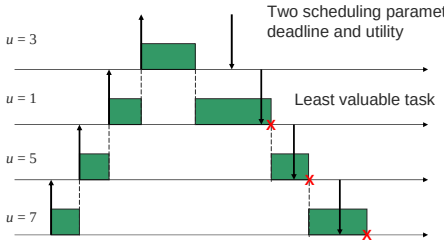
Load Shedding

- Feasible schedule if any task excluded in example?
- Selection criteria
- Fitness metric
- Second dimension: utility

Load Shedding



Load Shedding



Load Shedding