582425 Real-Time Systems (2ov) Spring 2006

Tiina Niklander

Course content

Week 1	Introduction
Week 2	RM & EDF (4-6)
Week 3	Aperiodic and sporadic jobs (7)
Week 4	Resource competition (8)
Week 5	Communication (11)
Week 6	Multiprocessors (9), OS (12)
Exam	Monday 8.5.2006

Course Structure

- Tuesday 12-14
- Study the material (=read the book)
- Do the exercises for the meeting
- Exercises will be published in the www on the previous week

Course material

- Jane Liu: Real-Time Systems, Prentice-Hall, 2000.
- Plus the articles:
  - K. Ramamritham, S.H. Son ja L.C.Dipippo. Real-time Databases and Data Services. Real-Time Systems, 28, 179-215, 2004.
  - J.A. Stankovic ja R. Rajkumar. Real-Time Operating Systems. Rea-Time Systems, 28, 237-253, 2004.
  - G.C. Buttazzo. Rate Monotonic vs. EDF: Judgment Day. Real-Time Systems, 29, 5-26,2005.

More information or alternative sources

- Some other real-time books:
  - Burns & Wellings: Real-Time Systems and Programming Languages, Addison-Wesley
  - Krishna & Shin: Real-Time Systems, McGraw-Hill, 1997
- News groups, mm. comp.realtime
- http://cs-www.bu.edu/pub/ieee-rts/ (IEEE Technical committee on Real-Time Systems)
- http://www.real-time.org/ (Douglas Jensen)

For the week 2 exercises you need to study

- Real-time system model (Ch 3)
- Basic Scheduling Mechanisms (Ch 4-6)
- Schedulability Tests (in Ch 6)
  - Utilisation
  - Time-Demand Analysis

Classification

Real-time systems
- hard
  - Aina toimiva (fail operational) — *Flight control*
  - Aina turvallinen (fail safe) — *Train control (semaphors)*
- soft
  - Suuri saatavuus (high availability) — *Telcom services*
  - Suuri eheys (high integrity) — *Bank services*

Model used on the course

- Processors *P*
- Resources *R*
  - Only the resources that are under a contention
- Jobs *J*
  - Release time *r*
    - phase, period
  - Execution time *e*
  - Deadline *d*

$J_i(\phi,p,e,d)$

Release time *r* – arrival time

- Distribution
  - Fixed r_i
  - Equal distribution $[r_i^-, r_i^+]$
  - Statistical ditribution function *A(x)*, for example Poisson
- The distribution of the release times depends on the used model

Execution time *e*

- Execution times can vary just as the release times
- Often possible to estime the minimum and maximum execution times $[e_i^-, e_i^+]$
- For the hard and critical jobs the maximum execution e_i^+ is always used.

Static Table based sched.

	d	e
T1	4	1
T2	5	1.8
T3	20	1
T4	20	2

- Period tasks (known in advance)
- Off-line created table for the whole hyperperiod (least common multiple of the periods)

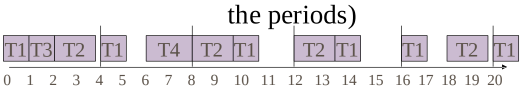


Table elements: (0,T1),(1,T3),(2,T2),(3.8,I),(4,T1),....,(19.8,I)

Frames:

- Lower limit – sufficiently long
  - Each job fits in one frame

$$f \geq \max_{1 \leq i \leq n} (e_i)$$

- Upper limit – always one full frame between the release time and deadline of one job

$$2f - \mathrm{syt}(p_i, f) \leq D_i$$

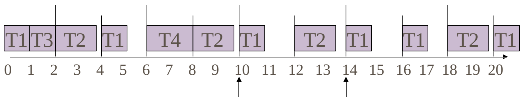
- Number of frames: size divides hyperperiod

$$\lfloor p_i / f \rfloor - p_i / f \geq 0$$

Frame size calculation

	d	e
T1	4	1
T2	5	1.8
T3	20	1
T4	20	2

- Hyperperiod: H = 20
- Lower: $f \geq \max(1, 1.8, 1, 2)$
- Alternatives: 2,4,5,10,20
- Upper: $2f\text{-syt}(p_i, f) \leq D_i$
- Conclusion: f=2 only poss.



Priority-based scheduling

- Static priority (both task and job)
  - Rate-monotonic (RM)
  - Deadline-monotonic (DM)
- Dynamic priority
  - EDF - task dynamic, job static
  - LST - both task and job dynamic

Processor Utilisation

$$U = \sum_{i=1}^{n} \frac{e_i}{p_i}$$

Schedulability condition

- EDF

$$U_{EDF} = \sum_{i=1}^{n} \frac{e_i}{\min(p_i, d_i)} \leq 1$$

- RM

$$U_{RM} \leq n(2^{1/n} - 1)$$

$$\lim_{n \to \infty} n(2^{1/n} - 1) = \ln 2 \approx 0.693$$

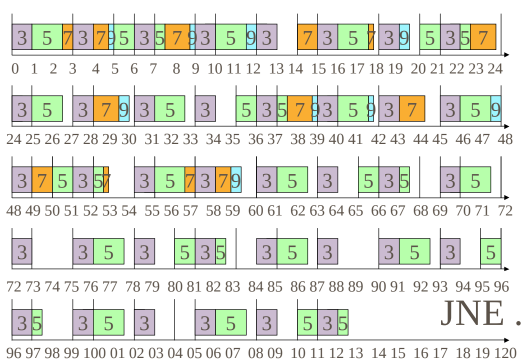
Schedulability Test

- Taks
  - (3,1), (5,1.5), (7,1.25) ja (9,0.5)
- Utilisation
  - 1/3 + 1.5/5 + 1.25/7 +0.5/9 = 0.85

	d	e
T1	3	1
T2	5	1.5
T3	7	1.25
T4	9	0.5

- 0.85 <1 => schedulable with EDF
- 0.85 >$4(2^{1/4}-1)$= 0.757 => RM - cannot say. Needs some other test.

(3,1), (5,1.5), (7,1.25) ja (9,0.5)



RM ajoitus (alkuosa)

Time-Demand Analysis

- TasksT$_i$ in priority order from highest

$$w_i(t) = e_i + \sum_{k=1}^{i-1} \left\lceil \frac{t}{p_k} \right\rceil e_k,\ \text{kun } 0 < t \le p_i$$

- Tasks are schedulable, when $\forall i$

$$\exists\, w_i(t) \le t,\ \text{jollekin } t \le d_i \le p_i$$

Time-Demand Analysis

T1 (3,1)
T2 (5,1.5)
T3 (7,1.25) ja
T4 (9,0.5)

Maximum response Times (notice small dots)
T1 1
T2 2.5
T3 4.75
T4 9

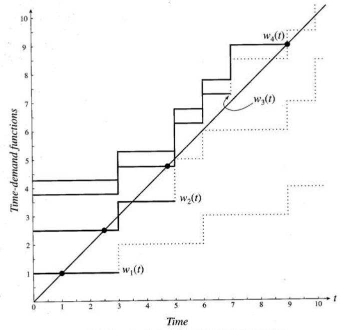


FIGURE 6-9 Time-demand analysis (3,1), (5, 1.5), (7, 1.25), and (9, 0.5).

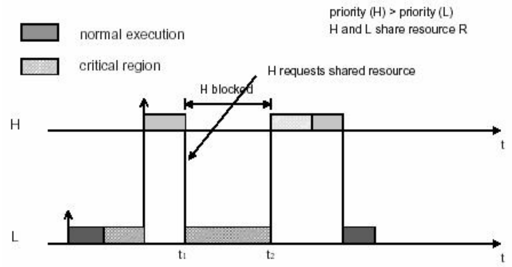
Blocking needs to be concidered

- One lowe priority task can block a higher priority task
- Add the longest blocking time over all lower priority tasks to the execution time of the higher priority task. Blocking time is

$$b_i(np) = \max_{i+1\le k\le n} \theta_k$$

$$U_{EDF} = \sum_{i=1}^{n} \frac{e_i}{\min(p_i, d_i)} + \frac{b_i}{\min(p_i, d_i)} \le 1$$

Odotusongelma (Blocking problem)



Prioriteetin kääntyminen (Priority inversion)