

Distributed and Pervasive Systems

Team 2

Report
Aarhus University, Science and Technology
Lector: Christian Fischer Pedersen

February 19, 2018

Name	Study number	Signature
David Jensen	11229	
Henrik Bagger Jensen	201304157	
Ólafur Dagur Skúlason	IY11249	
Titas Urbonas	201700321	
Christian Lillelund	201408354	

Contents

Contents	ii
1 Introduction	1
2 Real-Time Computing	3
2.1 Determinism	3
2.2 Scheduling.....	3
2.2.1 Rate-monotonic scheduler	4
2.2.2 Earliest deadline first scheduler	4
2.2.3 Least-slacktime-first scheduler.....	4
2.2.4 Resource control.....	4
3 Synchronization	5
4 Real-Time Ethernet	7
5 Middleware	9
6 Consistency	11
7 Fault Tolerance	13
8 Leader Election	15
9 Positioning	17
10 Discussion	19
11 Conclusion	21
12 Perspectives	23
Bibliography	25

Chapter 1

Introduction

Chapter 2

Real-Time Computing

Topics/keywords:

Jobs & schedulers & task handling

2.1 Determinism

- Finite automata for determined/undetermined systems Short intro, an example

DFA

NFA

TFA

Compare deterministic and non-deterministic automata Talk about timed automatas and how they comply with deadlines

2.2 Scheduling

- Introduction Arrival time, release time, deadlines Real-time computing What are jobs? Tasks? Have the basic definition written down (at least the Task/J one)

In scheduling the aim is to meet all hard deadlines and handle soft deadlines of jobs in the best possible manner and avoid deadlocks while doing so. The arrival time of a job is the moment in time it arrives at a processor, while the release time of a job is the moment in time it becomes available for execution. A task can have different job types and such as periodic, sporadic and Aperiodic. A task is a set of jobs known at the start of the system or triggered by an external event. A periodic task is defined by three parameters:

- The release time r of the first periodic job.
- The period p , which is a periodic time interval, at the start of which a periodic job is released.
- The execution time e of each periodic job.

and is written as (r, p, e)

2.2.1 Rate-monotonic scheduler**2.2.2 Earliest deadline first scheduler**

This scheduling strategy will give a higher priority to jobs if its deadline is earlier. In case of preemptive jobs and no competition for resources, this scheduler is optimal, in the sense that if utilization at a processor does not exceed one, then periodic jobs will be scheduled in such a way that no deadlines are missed.¹

2.2.3 Least-slacktime-first scheduler**2.2.4 Resource control**

Write about resource control/deadlock Priority inheritance Priority ceiling Find Søren Hansen (grandmaster) drawings/references

¹ [Fokknik(2013)] p.184

Chapter 3

Synchronization

Chapter 4

Real-Time Ethernet

Chapter 5

Middleware

Chapter 6

Consistency

Chapter 7

Fault Tolerance

Chapter 8

Leader Election

Chapter 9

Positioning

Chapter 10

Discussion

Chapter 11

Conclusion

Chapter 12

Perspectives

Bibliography

[Fokknik(2013)] Wan Fokknik. *Distributed Algorithms. An Intuitive Approach*. The MIT Press (December 6, 2013), 2013. ISBN 9780262026772.

