CHALMERS | UNIVERSITY OF GOTHENBURG

Parallel & Distributed Real-Time Systems

7.5 credit points

Professor Jan Jonsson

Department of Computer Science and Engineering
Chalmers University of Technology

CHALMERS | UNIVERSITY OF GOTHENBURG

Administrative issues

Lectures: (Jan Jonsson, Risat Pathan + special guests)

- Fundamental methods and theory
  - Real-time systems, scheduling, complexity theory
- 16 classroom lectures
  - Mondays at 13:15 – 15:00 in room EL52 (week 1 – 8)
  - Thursday at 08:00 – 09:45 in room EL52 (week 2 only)
  - Thursdays at 10:00 – 11:45 in room EL52 (week 1, 2, 3, 4 and 7)
  - Fridays at 15:15 – 17:00 in room EL52 (week 1 and 8)

CHALMERS | UNIVERSITY OF GOTHENBURG

Administrative issues

Consultation sessions: (Behrooz Sangchoolie)

- Questions and guidance regarding homework assignments
- Seven consultation sessions
  - Thursdays at 08:00 – 09:45 in lecture room ES52 (week 4, 5 and 7)
  - Thursdays at 10:00 – 11:45 in lecture room ES52 (week 5 and 8)
  - Fridays at 15:15 – 17:00 in lecture room ES52 (week 3 and 7)

✓ Starts week that first homework assignment is handed out

CHALMERS | UNIVERSITY OF GOTHENBURG

Administrative issues

Homework assignments: (HWAs)

- Two assignments (handed out on Apr 24 and May 11)
- Problem solving + paper reading (18-day deadlines)
- Written report (computer generated, electronically submitted)
- Presentation (summarize, and argue for, proposed solutions)

Examination:

- Compulsory homework assignments (report + presentation)
- Voluntary written exam (to enable highest grade)
- HWA grades: Failed, 3, 4, 5
- Final grade: average of two HWA results
- Successful examination ⇒ 7.5 credit points

Course literature

Lecture notes:

- Copies of PowerPoint presentations
- Whiteboard scribble

Complementary reading:

- Selected research articles from archival journals and conference proceedings
- Selected chapters from C. M. Krishna and K. G. Shin, "Real-Time Systems", McGraw-Hill, 1997 (+ errata list!)

Resources

Consultation sessions:

- In room ES52 (according to schedule)

PingPong:

- Administration of HWAs (form groups, submit documents, etc)
- Results from the grading of HWAs and written exam

Information board:

Follow @pdrtschalmers

`http://www.cse.chalmers.se/edu/course/EDA421`

Lecture notes will be available on the information board no later than 48 hours before the corresponding lecture

Course aim

After the course, the student should be able to:

- Formulate requirements for computer systems used in time- and safety critical applications.
- Master the terminology of scheduling and complexity theory.
- Describe the principles and mechanisms used for scheduling of task execution and data communication in real-time systems.
- Derive performance for, and be familiar with the theoretical performance limitations of, a given real-time system.

Course contents

What this course is all about:

- real-time systems modeling
- real-time application constraints
- real-time performance measures
- real-time task assignment and scheduling algorithms
- real-time inter-processor communication techniques
- complexity theory and NP-completeness
- distributed clock synchronization
- fault-tolerance techniques for real-time systems
- estimation of program run times

Course contents

What this course is not about:

- programming of parallel and distributed real-time systems
- implementation issues in real-time operating systems
- verification of program correctness
- high-performance parallel computing
- ...

What is a real-time system?

"A real-time system is one in which the correctness of the system depends not only on the logical result of computation, but also on the time at which the results are generated"

J. Stankovic, "Misconceptions of Real-Time Computing", 1988

"A real-time system is anything that we, the authors of this book, consider to be a real-time system. This includes embedded systems ... where Something Very Bad will happen if the computer does not deliver its output in time"

C. M. Krishna and K. G. Shin, "Real-Time Systems", 1997

What is a real-time system?

It is not only about high-performance computing!

Real-time systems must meet timing constraints

High-performance computing maximizes average throughput

Average performance says nothing about correctness!

Real-time systems are often optimized with respect to perceived "robustness" (control systems) or "comfort" (multimedia)

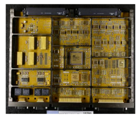
"Baby, I'm built for comfort ... I ain't built for speed"

Willie Dixon, "Built for Comfort", 1965

What is a real-time system?

Characteristics of real-time systems:

- Strict timing constraints
  - Responsiveness (= deadlines) and periodicity
  - Failure to meet timing constraints will cause system failure (hard deadlines) or will negatively affect quality of the user-perceived utility (soft deadlines)
- Application-specific design
  - Embedded systems (e.g., computer is part of a larger mechanical system)
  - Well-known operating environment
  - High reliability (fault tolerance)



RUAG satellite control system

What is a real-time system?

Examples of real-time systems:

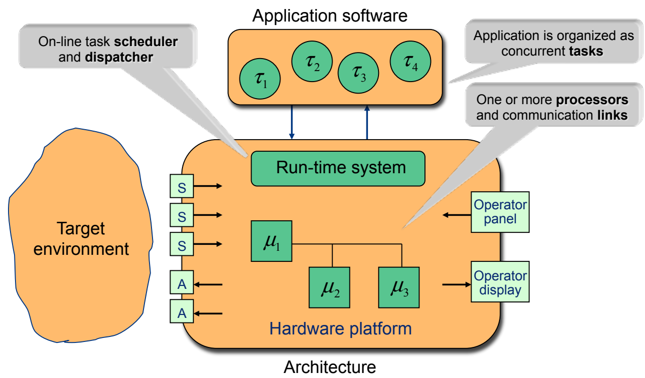
Control systems

- Industrial robots
- Cars, aircrafts, submarines, satellites
- Failure to meet timing constraints may cause major physical/economical damage or even loss of life

Multimedia systems

- Portable music players, streaming music
- Computer games; video-on-demand, virtual reality
- Failure to meet timing constraints will degrade user-perceived quality

Real-time system components



Why multiple processors?

The attractive price-performance ratios has enabled:

- Low-cost nodes in distributed real-time systems
- Powerful telecommunication/multimedia servers
- Multi-core processors in mobile phones and cars/aircrafts

"Tomorrow we are driving computers which look like cars"

Modern real-time systems execute reliable, high-throughput applications with explicit real-time constraints

Why multiple processors?

Distributed data processing:

- Locality constraints
  - data processing must take place close to sensor or actuator (e.g., robots, cars, aircraft)
- Reliability constraints
  - replication of computing resources provides fault-tolerance

Push-pull effect:

- New applications push future computer performance
- New computer platforms pull new applications

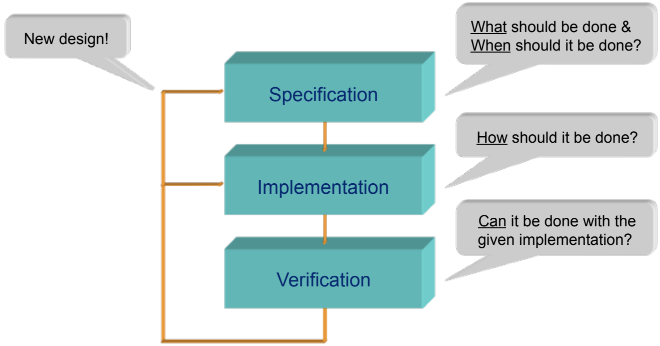
Why multiple processors?

New intriguing possibilities:

- High throughput
  - parallel execution of tasks
  - parallelization of algorithms (e.g., graphic algorithms)
- High schedulability
  - advanced scheduling algorithms (e.g., bidding, parallel B&B)
  - advanced dispatchers (e.g., affinity-based)
- High reliability
  - advanced fault-detection techniques (for high coverage)
  - massive redundancy (in time or space)

CHALMERS | UNIVERSITY OF GOTHENBURG

Designing a real-time system



CHALMERS | UNIVERSITY OF GOTHENBURG

Specification

Specification → Implementation

Requirements:		Constraints:
Reliability	→	Replication
Sampling rate	→	Periodicity
Response time	→	Deadline
Resources	→	Locality

CHALMERS | UNIVERSITY OF GOTHENBURG

Specification

Examples of application constraints:

- Timing constraints
  - A task must complete its execution within given time frames
    (example: task periodicity or deadline)
- Exclusion constraints
  - A task must execute a code region without being interrupted
    (example: a task needs exclusive access to a shared resource)
- Precedence constraints
  - A task must complete its execution before another task can start
    (example: a data exchange must take place between the tasks)

CHALMERS | UNIVERSITY OF GOTHENBURG

Specification

Examples of application constraints:

- Locality constraints
  - A task must execute on a specific processor because of the vicinity to some resource (DSP chip, sensor, actuator)
- Anti-clustering constraints
  - Identical copies of a task must execute on different processors for reliability reasons (a.k.a. spatial replication)
    (example: fault tolerance)
  - A group of tasks must execute on different processors for performance reasons
    (example: parallelization)

CHALMERS | UNIVERSITY OF GOTHENBURG

Specification

Examples of application constraints:

- Clustering constraints
  - A group of tasks must execute on the same processor for functional reasons
    (example: only one processor is used in low-power mode)
  - A group of tasks must execute on the same processor for performance reasons
    (example: intensive communication within the group)
  - A group of tasks must execute on the same processor for security reasons
    (example: risk for eavesdropping of network bus)

CHALMERS | UNIVERSITY OF GOTHENBURG

Specification

Where do the timing constraints come from?

- Laws of nature
  - Bodies in motion: arm movements in a robotic system
  - Inertia of the eye: minimal frame rate in film
- Mathematical theory
  - Control theory: recommended sampling rate
- Component limitations
  - Sensors and actuators: minimal time between operations
- Artificial derivation
  - Observable events: certain (global) timing constraints are given, but individual (local) timing constraints are needed

CHALMERS | UNIVERSITY OF GOTHENBURG

Specification

How critical are the constraints?

Hard constraints:

If the system fails to fulfill a timing constraint, the computational results is useless.

Correctness must be verified before system is put in mission!

Soft constraints:

Single failures to fulfill a timing constraint are acceptable, but the usefulness of the result decreases the more failures there are.

Statistical guarantees often suffice for these systems!

CHALMERS | UNIVERSITY OF GOTHENBURG

Implementation

Critical choices to be made at design time:

- Application software:
  - Programming language
    - Determines run-time performance and code size
    - Determines productivity, maintainability and reliability
    - Determines degree of timing verification that is possible
  - Concurrent programming
    - Program is structured as multiple sequential tasks
    - Models the execution of multiple sequential task simultaneously
      single-processor system: only pseudo-parallel execution possible
      multiprocessor system: true parallel execution possible

CHALMERS | UNIVERSITY OF GOTHENBURG

Implementation

Critical choices to be made at design time:

- Hardware architecture:
  - Single or multiprocessor architecture
    - Determines degree of true parallelism that can be exploited
  - Microprocessor family
    - RISC processor (pipelines, caches, support for multiprocessors)
    - Micro-controller (no, or very simple, pipelines/caches)
    - Determines cost and run-time performance
    - Determines difficulty in worst-case execution time (WCET) analysis
  - Communication network technology and topology
    - Determines cost, performance and reliability

CHALMERS | UNIVERSITY OF GOTHENBURG

Implementation

Critical choices to be made at design time:

- Run-time system:
  - System services
    - Operating system (real-time kernel with system calls)
    - Stand-alone system (linked library with subroutine calls)
    - Determines run-time performance and code size
    - Determines cost, flexibility and portability
  - Task and message dispatching model
    - Time vs. priority driven dispatching
    - Preemptive vs. non-preemptive dispatching
    - Determines potential of meeting timing constraints
    - Determines processor and network utilization

CHALMERS | UNIVERSITY OF GOTHENBURG

End of lecture #1