CHALMERS

Parallel & Distributed Real-Time Systems

7.5 credit points

Docent Jan Jonsson

Department of Computer Science and Engineering
Chalmers University of Technology

CHALMERS

Administrative issues

Lectures: (Jan Jonsson + special guests)

- Fundamental methods and theory
  - Real-time systems, scheduling, complexity theory
- 14 classroom lectures
  - Mondays at 13:15 – 15:00 in lecture room EL43
  - Thursdays at 10:00 – 11:45 in lecture room EL41
  - Tuesdays at 10:00 – 11:45 in lecture room EL41 (week 1-2 only)

Consultation sessions: (Behrooz Sangchoolie)

- Questions and guidance regarding homework assignments
- Five consultation sessions
  - Fridays at 15:15 – 17:00 in lecture room EL41 (week 3, 5, 6, 7)
  - Friday at 15:15 – 17:00 in lecture room EL43 (week 4)
  - Starts week after first homework assignment is handed out

CHALMERS

Administrative issues

Homework assignments: (HWAs)

- Two assignments (handed out on Nov 3 and Nov 21)
- Problem solving + paper reading (18-day deadlines)
- Written report (computer generated, electronically submitted)
- Presentation (summarize, and argue for, proposed solutions)

Examination:

- Passed homework assignments (report + presentation)
- Passed final written exam (Dec 14 at 14:00, in the V building)
- Grading policy: homework (60%) + final exam (40%)
- Grades: Failed, 3, 4, 5
- Successful examination ⇒ 7.5 credit points

CHALMERS

Course literature

Lecture notes:

- Copies of PowerPoint presentations
- Blackboard 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!)

CHALMERS

Information

Consultation sessions:

- Fridays, 15:15 – 17:00 in room EL41 (EL43, week 4)

Student portal:

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

Information board:

`URL: 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.

CHALMERS

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.

CHALMERS

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

CHALMERS

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
- ...

CHALMERS

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

CHALMERS

What is a real-time system?

"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

CHALMERS

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 build for speed"

Willie Dixon, "Built for Comfort", 1965

CHALMERS

What is a real-time system?

Properties of a real-time system:

- Strict timing constraints
  - Responsiveness (deadlines), periodicity (sampling rate)
  - Constraints can (ought to) be verified
- Application-specific design
  - Embedded systems
  - Carefully specified system properties
  - Well-known operating environment
- High reliability
  - Thoroughly-tested components
  - Works even in presence of component faults (fault tolerance)

CHALMERS

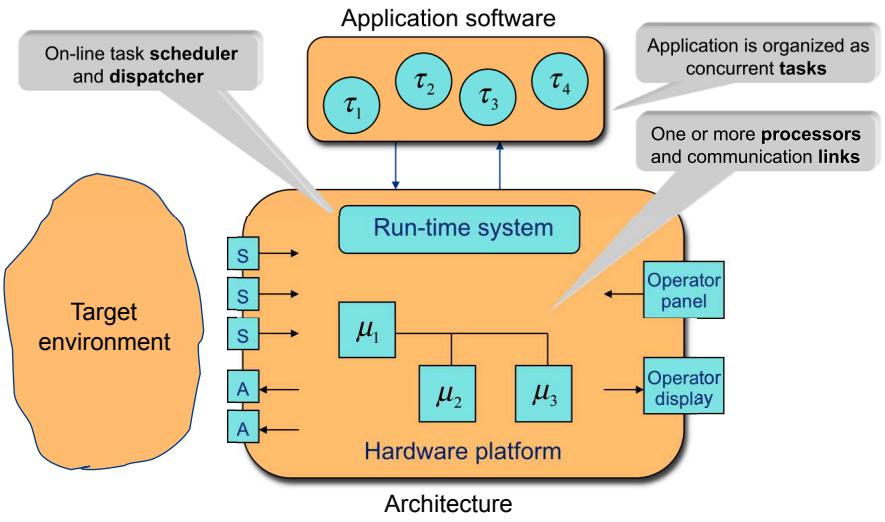
What is a real-time system?

Examples of real-time systems:

- Control systems
  - Manufacturing systems; process industry
  - Cars, aero planes, submarines, space shuttles
- Transaction systems
  - E-commerce; ticket booking; teller machines; stock exchange
  - Wireless phones; telephone switches
- Multimedia
  - Computer games; video-on-demand
  - Virtual reality

CHALMERS

Real-time system components



CHALMERS

Why multiple processors?

The attractive price-performance ratios has enabled:

- Low-cost nodes in distributed real-time systems
- Powerful telecommunication/multimedia servers

"Tomorrow we are driving computers which look like cars"

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

CHALMERS

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

CHALMERS

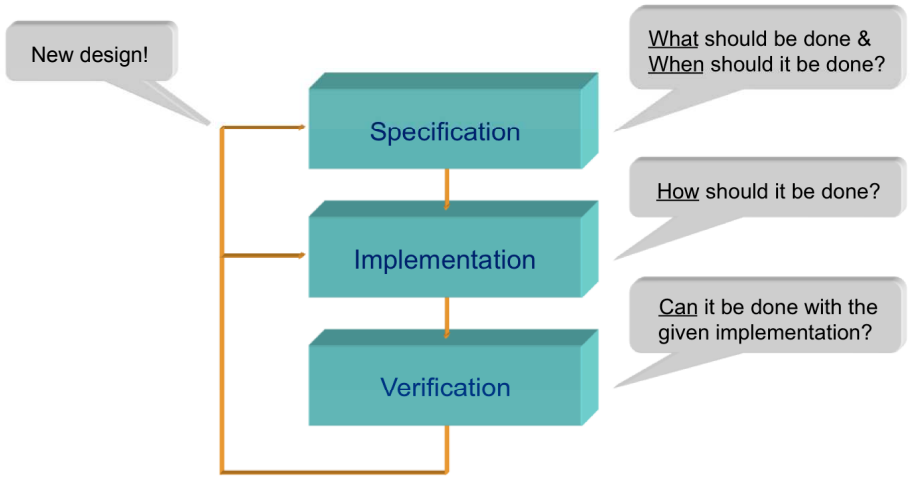
Why multiple processors?

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)

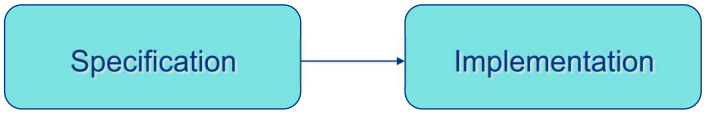
CHALMERS

Designing a real-time system



CHALMERS

Specification



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

CHALMERS

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

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

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

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

Specification

How critical are the constraints?

Hard constraints:

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

Non-critical: system can still function with reduced performance
- Navigational functions; diagnostics

Critical: system cannot continue to function
- Flight control system; control loop

Safety-critical: can cause serious damage or even loss of life
- Braking systems (ABS); defense system (missiles, robots)

Correctness must be verified before system is put in mission!

CHALMERS

Specification

How critical are the constraints?

Soft constraints:

Single failures to fulfill a timing constraint is acceptable, but the usefulness of the computational result is reduced (often to what can be considered useless).

- Reservation systems: seat booking for aircraft; teller machine
- Multimedia: video-on-demand, computer games, Virtual Reality
- …

Statistical guarantees often suffice for these systems!

CHALMERS

Implementation

Critical choices to be made at design time:

- Programming paradigm:
  - Sequential programming
    - Program is structured as one single “loop”
    - Ignores that the application has inherent concurrency
  - 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

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 (I/O ports, A/D-converters, no pipeline/cache)
    - Determines cost, performance, and difficulty in worst-case execution time (WCET) analysis
  - Network topology
    - Shared media interconnection network
    - Point-to-point interconnection network

CHALMERS

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)
  - Execution model
    - Time vs. priority-driven dispatching
    - Preemptive vs. non-preemptive execution
  - Communication model
    - Time vs. token vs. priority-driven message passing