Real-Time Systems

Lecture #16

Professor Jan Jonsson

Department of Computer Science and Engineering
Chalmers University of Technology

Real-Time Systems

Facing the written exam

Monday, March 17, 2025 @ 08:30−12:30
at Johanneberg campus

Note: in case you need to take a re-exam in August, please note that it takes place at Lindholmen campus

Facing the exam

Permitted to use during the written exam:

- The standard authorised aids
  - Pencils, erasers, rulers and dictionaries
  - Markings or notes are <u>not</u> permitted in the authorised aids.
  - Electronic dictionaries may <u>not</u> be used during the exam
- Chalmers-approved calculator
  - Approved models:
    Casio FX-82, Casio FX-85, Sharp EL-W531, Texas TI-30

Please observe: as of 2022 the compendium "Programming with the TinyTimber kernel" is <u>no longer</u> an authorised aid!

Facing the exam

Reading guidelines:

- Lecture and Exercise notes
  - PowerPoint hand-outs + blackboard scribble
  - All material is very relevant, and may be examined
- Excerpts from research articles and books:
  - Recommended reading, but will not be examined
- Exercise compendium
  - Recommended problem solving …
- Old written exams
  - For inspiration …

Facing the exam

Important knowledge areas:

- Design principles for real-time systems
  - Real-time systems: typical properties, misconceptions
  - Real-time constraints: origin, interpretation (soft/hard)
  - Design phases: specification, implementation, verification
  - Verification: methods, difficulties, pitfalls
  - Network communication: methods (CAN in particular)

Facing the exam

Important knowledge areas (cont'd):

- Principles of concurrent programming
  - Paradigm: reactive (event driven) programming
  - Parallelization: pros & cons
  - Resource management: mutual exclusion, critical region
  - Deadlock: definition, management
  - Starvation: definition, management

Facing the exam

Important knowledge areas (cont'd):

- Language support for concurrent programming
  - Mutual exclusion: protected objects, monitors, semaphores, synchronized methods, mutex'd methods
  - Machine-level mutual exclusion: disable interrupt, test-and-set
- Language support for real-time programming
  - Units of concurrency: task, thread, method
  - Scheduling support: clocks, time, delays, priorities
  - Device drivers: interrupts handlers, call-back functionality
  - TinyTimber: "how it's done"
  - WCET: purpose, required properties, analysis methods

Facing the exam

Important knowledge areas (cont'd):

- Scheduling theory
  - Task model: WCET, deadline, period, offset
  - Scheduling: definitions, priorities, preemption
  - Feasibility tests: purpose, exactness (sufficient/necessary)
  - Complexity theory: time complexity, NP-completeness
- Scheduling with cyclic executives
  - Properties: time table, pros & cons
  - Scheduling: generation of time tables, run-time behavior
  - Feasibility test: hyper-period analysis

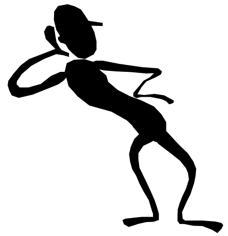
Facing the exam

Important knowledge areas (cont'd):

- Scheduling with pseudo-parallel execution
  - Properties: priority assignment, optimality, pros & cons
  - Scheduling: run-time behavior, construct timing diagram
  - Feasibility test: theory, assumptions, exactness, complexity
  - RM, DM, EDF: "how it's done"
  - RMFF, RM-US: "how it's done"

Facing the exam

What type of exam problems will there be?



- **Basic** part

  Will probe your skills in performing fundamental calculations relating to single-processor scheduling theory.

  Minimum requirement: to pass the exam (= **grade 3 [G]**) you must obtain ≥ 24 points (out of 30) in the Basic part. Chalmers [GU]

- **Advanced** part

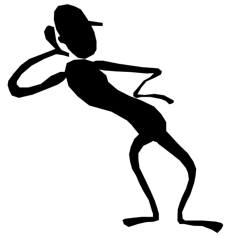
  Will encompass problem solving and scheduling theory insights

  To get **grade 4** you must obtain ≥ 24 points in the Basic part and ≥ 36 points (out of 60) in both parts combined. Chalmers

  To get **grade 5 [VG]** you must obtain ≥ 24 points in the Basic part and ≥ 48 [44] points (out of 60) in both parts combined. Chalmers [GU]

Note: Your submitted solutions for the Advanced part will only be graded if you have obtained ≥ 24 points in the Basic part.

Facing the exam



What type of exam problems will there be?

- **Basic** part

  Fundamental calculations relating to single-processor scheduling

  The following types of problems will appear:

  - Perform utilization-based schedulability analysis for a task set
  - Perform worst-case response-time analysis for a task set
  - Perform processor-demand analysis for a task set
  - Draw a timing diagram based on simulation of on-line scheduler

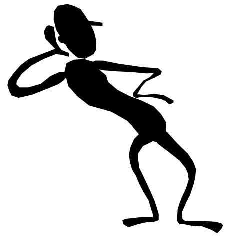
  Will not include unknown parameter values or blocking factors.

  In addition, problems relating to the following topics will appear:

  - WCET analysis, scheduling concepts, real-time programming

Facing the exam

What type of exam problems will there be?



- **Advanced** part

  In-depth problem solving and scheduling theory insights

  The following types of problems will appear:

  – Single-processor scheduling for a task set (may include unknown parameter values and/or blocking factors)

  – Multiprocessor scheduling for a task set (may include unknown parameter values)

  – Scheduling theory problems that will probe your knowledge regarding feasibility testing, optimality and complexity theory

Facing the exam

Deriving the final grade in the course:

The final grade (U, 3, 4, 5) in the course will reflect your laboratory skills as well as your theoretical skills.

The final grade is influenced almost equally* by:

- Your results in course element ‘Laboratory’
  - based on a ‘Pass’ grade (3, 4, 5)
- Your results in course element ‘Examination’
  - based on a ‘Pass’ written exam score (24−60 points)

(* See corresponding look-up table in Canvas for details)

Note: GU students use Chalmers grading scale within Canvas, but get corresponding GU grades in Ladok.