Exam themes & requirements

Individual or collaborative (2-person team), free choice, with no a-priori bearing on grading

Exam modality

- **Individual** (*single*) or **collaborative** (*pair*) undertaking
- Topic chosen freely out of those enumerated here
- Exam work in incremental steps
  1. Understand the context and the related state of the art
  2. Determine your learning outcome (and declare it)
  3. Produce the required per-type exam work accordingly
- Keep regular contact with instructor required during exam work
- Exam work's output has two parts
  - **Technical report** (TR) on the candidate's findings
    - 8-10 pages, ACM Primary Article Template (journals or proceedings, your choice) https://www.acm.org/publications/proceedings-template
  - **Face-to-face oral presentation**
- Admission to oral presentation upon approval of the TR

Two types of assignment

- **Type 1**
  - Critical understanding of a recent *academic paper* in the domain with potential for direct application
  - Familiarize with the literature background to the paper's research
  - Understand the authors' premises, methods, and claims
  - Reproduce – in the small – the original experiments and results
- **Type 2**
  - Small-scale *real-time embedded software* development with critical analysis of given run-time performance indicators, and empirical evaluation
  - Familiarize with programming for an embedded processor (e.g., microcontroller) and cross-platform development
  - Refine understanding of runtime activity underneath the application

Structure of Technical Report

- **Problem statement**
  - Scope of the original work
  - Purpose of the original work and of the assignment
  - Perimeter of investigation in the assignment
- **Work product**
  - Technical choices made in the assignment
  - Design and results of the (small-scale) experiments
- **Self-assessment**
  - Candidate’s own critique of own exam work
    - Results / limitations; achievements / failures
  - Discussion of the candidate’s learning outcomes

Exam specification

- **Type-1 theme**
  - Understanding a recent scientific publication pertinent to the course topics
  - Reproducing (some of) its experiments
- **Type-2 theme**
  - Hands-on work with exemplary real-time systems technology
  - Working on a small proof-of-concept

Type-1 theme choices

- **Global versus local schedulability analysis of a two-level hierarchical system** (**2025**)
  DOI: TBA (https://tinyurl.com/24bp7r7x)
- **An Empirical Study of Performance Interference: Timing Violation Patterns and Impacts** (**2024**)
  DOI: 10.1109/RTAS61025.2024.00033
- **Interference-free Operating System: A 6 Years' Experience in Mitigating Cross-Core Interference in Linux** (**2024**)
  DOI: 10.1109/RTSS62706.2024.00034
- **In Search of Butterflies: Exceedance Analysis for Real-Time Systems under Transient Overload** (**2024**)
  DOI: RTSS62706.2024.00028

Type-2 theme choices

- **Choice 1**: transforming an Ada Ravenscar Real-Time application into an equivalent system written in Rust or another language fit for real-time programming
  - Comparison of performance evaluation experiments
- **Choice 2**: developing a small proof-of-concept use of the Constellation OS backend
  - Critical evaluation of its fitness for real-time operation

Type-2 theme choice 1

- Start from the example application associated with Lecture #12
  - Written in Ravenscar Ada
- Transform it into an equivalent app written in another language of choice suited for real-time programming
  - Use of Rust is recommended
  - Target processor of your choice, including host
- Compare behavior of resulting system with the findings reported in exam-TR from predecessors of yours
  - https://tinyurl.com/5y99274k
  - Considering the FPS runtime variant only

Type-2 theme choice 2

- Study Constellation OS resources in the public domain
  - https://oodaworld.com/constellation-os-ooda-world/
- Try and use its codebase for a small proof-of-concept
  - https://github.com/TheConstellationProject/ConstellationOS
- Critically analyse its fitness for use in real-time operation

Exam schedule and contract

- Declare your choice of theme
  - By **Friday, May 30, 17:00 CEST**
  - Committing to an estimated time of delivery
- Earn bonus on exam grading
  - +2 pts if delivering TR by **Thursday, July 31**
  - +1 pts if delivering TR by **Friday, September 19**
- Latest delivery deadline
  - By **Friday, January 16, 2026, 17:00 CET**
- Do deliver your product as soon as ready
- Admission to oral discussion if TR passes scrutiny
  - Registration of outcome by official exam session schedule
- Send weekly heartbeats from when you start exam work

Important: understand the intent

- This work, while free for choice of theme, is intended to measure your understanding of the *full spectrum* of course topics
- Consequently, in doing this work you are expected to *root* your reasoning and your argumentation on the applicable range of course contents
  - Do this explicitly at all the pertinent places in your TR
- Doing so will show that you have understood what you are talking about, and how what you did and the way you did it relate to the state of the art in the RTKS field as covered in this course