Exam themes and requirements

Individual or collaborative (for groups of 2-3 persons max), free choice, with no apriori bearing on grading

Two types of assignment

Type 1 (individual or collaborative)

- Critical understanding of a recent *academic paper* in the domain with potential for direct application
- Reproduction of claimed results and critical analysis

Type 2 (preferably collaborative)

Small-scale *real-time embedded software* development with critical analysis of given run-time performance indicators, and empirical evaluation thereof

Required conduct

Type 1

- Study and assimilate the background to the paper's research ambit: familiarize with bibliographic literature
- Put the authors' premises, methods, and claims into question: understanding before trusting
- Repeat the authors' experiments and determine whether the claimed results can be reproduced

Type 2

- □ Familiarize with a typical embedded processor (microcontroller) and with cross-platform development
- **•** Refine understanding of runtime activity underneath the application

Output (both types)

D Technical report (TR), in English, accompanied by repo with experiments

Exam specification

Type 1.a

- Schedulability Analysis for Adaptive Mixed Criticality Systems with Arbitrary Deadlines and Semi-Clairvoyance, A. Burns, R.I. Davis, RTSS 2020, DOI: 10.1109/RTSS49844.2020.00013
- Nested, but Separate: Isolating Unrelated Critical Sections in Real-Time Nested Locking, J. Robb, B.B. Brandenburg, DOI: 10.4230/LIPIcs.ECRTS.2020.6

Type 2

- Microcontroller STM32F429 equipped with programmable LCD, <u>st.com/en/evaluation-tools/32f429idiscovery.html</u>
- □ Ada Ravenscar compiler and baseline application

Type 1.a

Nested, but Separate: Isolating Unrelated Critical Sections in Real-Time Nested Locking

James Robb

Reykjavik University, Iceland me@jamesrobb.ca

Björn B. Brandenburg

Max Planck Institute for Software Systems, Kaiserslautern, Germany bbb@mpi-sws.org

— Abstract -

Prior work has produced multiprocessor real-time locking protocols that ensure asymptotically optimal bounds on priority inversion, that support fine-grained nesting of critical sections, or that are independence-preserving under clustered scheduling. However, while several protocols manage to come with two out of these three desirable features, no protocol to date accomplishes all three. Motivated by this gap in capabilities, this paper introduces the *Group Independence-Preserving Protocol* (GIPP), the first protocol to support fine-grained nested locking, guarantee a notion of independence preservation for fine-grained nested locking, and ensure asymptotically optimal priority-inversion bounds. As a stepping stone, this paper further presents the *Clustered k-Exclusion Independence-Preserving Protocol* (CKIP), the first asymptotically optimal independence-preserving k-exclusion lock for clustered scheduling. The GIPP and the CKIP rely on allocation inheritance (a.k.a. migratory priority inheritance) as a key mechanism to accomplish independence preservation.

Type 1.b

Schedulability Analysis for Adaptive Mixed Criticality Systems with Arbitrary Deadlines and Semi-Clairvoyance

Alan Burns Robert I. Davis Department of Computer Science, University of York, UK. Department of Computer Science, University of York, UK.

> Abstract—This paper provides analysis of the Adaptive Mixed Criticality (AMC) scheduling scheme for mixed-criticality systems that include tasks with arbitrary deadlines and semi-clairvoyant behavior. An arbitrary deadline task is one that can have a deadline that may be greater than its period. A semi-clairvoyant task is one that upon arrival of each job, reveals which of its two WCET parameters will be respected. This enables an earlier switch to be made from the normal mode of operation to the abnormal mode. The previously published schedulability test AMC-max is modified to cater for both of these extensions. Evaluation shows that there is a significant improvement in schedulability for semi-clairvoyant tasks over non-clairvoyant, and for arbitrary-deadline tasks over considering those deadlines as being constrained by the task's period.

Type 2 /1

- Ada Ravenscar
 - GNAT Community edition, <u>https://www.adacore.com/community</u>
 - Runtime modified @ UNIPD for selective support of FPS or EDF, transparent to the application, <u>https://github.com/DPerale/comparison-system-FPS-EDF</u>
- Specific goal
 - Extend and refine the work of student predecessors around an application with precedence-constrained tasks
 - Compare the performance of EDF and FPS scheduling for that system when pushed at the border of overload
- Processor procurement
 - Loan it from fellow students or purchase from a commercial supplier
 - digikey.it → STM32F429I-DISC1 (€ 24.73, VAT inclusive)
 - mouser.it → STM32F429I-DISC1 (€ 26.34, VAT inclusive)
 - (returning it to me with invoice after exam for refund)



MAST ANALYSIS OF A RAVENSCAR PRECEDENCE-CONSTRAINED APPLICATION WITH FPS AND EDF SCHEDULING

TECHNICAL REPORT

Giovanni Jiayi HuADepartment of MathematicsDepartUniversity of Padua, Italy I-35121UniversityEmail:giovannijiayi.hu@studenti.unipd.itEmail: alessity

Alessio Gobbo Department of Mathematics University of Padua, Italy I-35121 Email: alessio.gobbo@studenti.unipd.it

This paper describes a hard real-time application built with the Ravenscar profile of the Ada programming language and running on a real-time kernel of reduced size and complexity. The application is comprised of several tasks whose activation events present dependency relationships, and this characteristic allows interesting considerations during the different analysis we provide. We consider the same tasks under both Fixed-Priority Scheduling (FPS) and Earliest Deadline First (EDF) and evaluate different metrics like response times, jitters and blocking times. Throughout the sections, we also test the ability of the MAST analysis tools to describe a formal model of dependent tasks and to check their deadline satisfaction with less pessimism as possible without compromising correctness. Lastly, we offer some insight into the behaviour of both FPS and EDF systems under permanent and transient overload, a showcase of how classic real-time considerations may be revaluated to consider dependent tasks.

Exam schedule and conduct

- Declare your choice of theme
 - **By Friday, June 11, 17:00**
 - Committing to latest date of delivery
- Bonus accrued
 - □ If delivering by Friday, October 1, 17:00
- Limit of validity
 - □ Expires by Friday, January 21, 2022, 17:00
- Deliver your product as soon as ready
 - □ Admission to oral discussion if TR passes scrutiny
 - □ Registration of outcome by official exam session schedule