Temi d'esame

Per svolgimento individuale o in piccoli gruppi (2-3) persone, a scelta libera, senza differenze di valutazione a priori

Due modalità a libera scelta

1. Studio, analisi critica e approfondimento sperimentale di un recente **lavoro scientifico** con potenziale di impatto applicativo

2. **Sviluppo prototipale** di un piccolo sistema *embedded real-time* con studio e valutazione sperimentale di indicatori prestazionali significativi

Modalità 1: obiettivi formativi

- **Studio**
  - Andare alle fonti (bibliografiche) dell'argomento trattato, per comprendere bene ciò che gli autori implicano senza dettaglio
- **Analisi critica**
  - Farsi domande: non tutto è verità indiscussa
    - Valutare il lavoro in esame, per solidità di obiettivi, metodo, soluzione proposta, valutazione, applicabilità
- **Approfondimento sperimentale**
  - Selezionare aspetti candidabili a valutazione empirica e confrontare i propri risultati con quelli del lavoro originale

Modalità 2: obiettivi formativi

- Familiarizzazione con
  - Uso di un **processore *embedded*** (microcontrollore) rappresentativo del dominio
  - Pratiche di ***cross-development*** con tecnologie specializzate per sistemi *real-time*
- Esplorazione sperimentale di aspetti di esecuzione significativi ai fini della predicibilità
  - Osservare, comprendere, valutare ciò che accade all'applicazione *sotto* la sua esecuzione funzionale

Tre temi

- **Tema 1** (modalità 1)
  - ***RT.js: Practical Real-Time Scheduling for Web Applications***, C. Dietrich, S. Naumann, R. Thrift, D. Lohmann (Leibniz Universität Hannover, Germany), RTSS 2019, DOI: 10.1109/RTSS46320.2019.00017
- **Tema 2-3** (modalità 2)
  - Sviluppo con microcontrollore STM32F429 dotato di LCD programmabile
    st.com/en/evaluation-tools/32f429idiscovery.html
  - **Opzione A**: ambiente ERIKA^3
  - **Opzione B**: ambiente Ada Ravenscar

Tema 1

RT.js: Practical Real-Time Scheduling for Web Applications

Christian Dietrich Stefan Naumann Robin Thrift
Leibniz Universität Hannover
Hannover, Germany
{dietrich, naumann, lohmann}@sra.uni-h

***Abstract*—For billions of deployed browsers, JavaScript provides the platform-independent lingua franca that enabled the triumphal march of web-based applications. Originally intended for simple UI-event processing, JavaScript comes with an event-driven programming model, where event-callback functions are executed in strict sequential order. However, with applications getting more complex and tasks becoming more computation intensive, its first-come–first-served and run-to-completion semantic is hitting a limit, when reactions to user inputs are delayed beyond the human perception threshold. With the rise of the Internet of Things, this leads to friction-filled user experiences in everyday situations.**

With RT.js, we selectively introduce pseudo-preemption points into JavaScript functions and sequence the execution of event callbacks with well-known real-time scheduling policies, like EDF. Thereby, we provide a soft real-time abstraction that mitigates the described shortcomings of the JavaScript execution model without modifying the actual engine; making RT.js compatible with billions of devices. Applied to generated real-time task sets, we can almost eliminate the 30-percent deadline-miss ratios of baseline JavaScript at moderate costs. In a browser-based macro benchmark, we could diminish the influence of computation-intensive background tasks on the page-rendering performance.

Tema 2 – Opzione A

- Ambiente **ERIKA^3**, erika-enterprise.com/
  - RTOS *bareboard*, con API *platform-independent*
    - Disponibile dal 15/6/2020
  - Tecnologia *open source*, in uso in dominio *automotive*
- Obiettivo
  - **Programmare una coreografia di funzioni grafiche sullo schermo** e studiare come l'effetto venga perturbato da fenomeni «*under the hood*»
    - Interferenza, sospensione prerilascio, *driver non-reentrant*, …
  - Valutare comparativamente la configurazione migliore

Tema 2 – Opzione B

- Ambiente **Ada Ravenscar**
  - Nella versione modificata @ UNIPD
    - Supporto selettivo di *scheduling* FPS, EDF trasparente all'app
    - Monitoraggio degli eventi di *scheduling* a fini di confronto

    https://github.com/DPerale/comparison-system-FPS-EDF
    - Limitatamente al *runtime* (`*-ravenscar-arm`)
  - Obiettivo **a scelta**
    1. **Estendere il lavoro di vostri predecessori**
    2. **Medesimo obiettivo del tema 2.A**, ma con questa specifica tecnologia, in vista di valutazione comparativa

Tema 2 – Opzione B.1

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

TECHNICAL REPORT

Giovanni Jiayi Hu
Department of Mathematics
University of Padua, Italy I-35121
Email: giovannijiayi.hu@studenti.unipd.it

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.

Tema 2: procurarsi il processore

- Acquisto diretto vostro, rimborso a valle (con ricevuta), e consegna @ DM
  - Massima autonomia

  digikey.it → STM32F429I-DISC1 (€ 33,62, IVA inclusa)

  mouser.it → STM32F429I-DISC1 (€ 34,15, IVA inclusa)
- Prenotazione, acquisto @ DM, prelievo dietro appuntamento
  - Massima sincronizzazione

Azioni e scadenze

- **Dichiarazione di scelta tema/opzione**
  - Entro **venerdì 12 giugno 2020, ore 17**
    con impegno tempi di consegna
- Intervalli di consegna
  - **Punti bonus**
    - Entro **venerdì 2 ottobre 2020, ore 17**
  - **Limite estremo**
    - Entro **venerdì 22 gennaio 2021, ore 17**

Prova d'esame

- Svolgimento a prenotazione, registrazione esito «per appello»
- La prova si svolge in due passaggi successivi
  - Produzione e consegna di una relazione tecnica, che presenti il lavoro svolto
    - Comprensione del problema affrontato, esito dello studio, risultati sperimentali, retrospettiva di quanto appresso
  - All'approvazione della relazione (che può comportare iterazioni), invito all'esame orale
    - Presentazione dei punti principali della relazione, anche alla luce dei commenti di revisione da parte del docente