



# UNIVERSITÀ DI PISA

---

## REAL-TIME SYSTEMS

**GIORGIO CARLO BUTTAZZO**

Anno accademico 2016/17  
CdS INFORMATICA E NETWORKING  
Codice 160II  
CFU 6

Moduli	Settore/i	Tipo	Ore	Docente/i
SISTEMI REAL-TIME	ING-INF/05	LEZIONI	48	GIORGIO CARLO BUTTAZZO

### Obiettivi di apprendimento

#### *Conoscenze*

Il corso consente agli studenti di sviluppare software con vincoli temporali per diversi domini applicativi, quali robotica, avionica, sistemi automotive, multimediali e biomedici. Gli studenti impareranno i fondamenti teorici e le metodologie più moderne per sviluppare ed analizzare applicazioni sensibili al tempo con elevato grado di concorrenza e requisiti di prestazione. Esempi pratici saranno illustrati per spiegare come derivare i vincoli temporali dai requisiti dell'applicazione. Quindi, il corso si concentra sulle metodologie atte ad aumentare la prevedibilità dei sistemi controllati da computer, inclusi gli algoritmi per la schedulazione dei task e la gestione di risorse.

#### *Modalità di verifica delle conoscenze*

Le conoscenze acquisite durante il corso verranno verificate attraverso un'assegnazione di progetto e durante l'esame.

#### *Capacità*

Gli studenti acquisiranno competenze nell'analizzare, sviluppare e testare software in tempo reale, utilizzando strumenti e librerie specifiche per la gestione di applicazioni multitask con vincoli temporali.

#### *Modalità di verifica delle capacità*

Le capacità acquisite verranno verificate attraverso lo sviluppo di un progetto e una relazione scritta che spiega le scelte progettuali e le soluzioni di implementazione adottate.

#### *Comportamenti*

Gli studenti avranno l'opportunità di sviluppare il progetto in un piccolo gruppo, condividendo la responsabilità nella gestione dello sviluppo delle applicazioni.

#### *Modalità di verifica dei comportamenti*

Discussione orale del progetto, durante la quale gli studenti sono tenuti a spiegare il loro contributo specifico nello sviluppo del lavoro e le soluzioni adottate.

#### *Prerequisiti (conoscenze iniziali)*

Conoscenze di base su matematica, fisica, architetture di elaborazione, linguaggi di programmazione e sistemi operativi.

#### *Corequisiti*

Nessuno.

#### *Prerequisiti per studi successivi*

Il presente modulo è altamente consigliato per meglio comprendere i contenuti dei seguenti corsi: Concurrent Distributed Systems, Design of Embedded Systems, Component-Based Software Design.



## UNIVERSITÀ DI PISA

---

### Indicazioni metodologiche

Le lezioni saranno supportate da diapositive e materiali multimediali, ove necessario. Presentazioni e altro materiale didattico saranno forniti attraverso il sito web del corso. Gli studenti possono interagire con il docente tramite e-mail e direttamente, previo appuntamento. Il corso sarà erogato in lingua inglese.

### Programma (contenuti dell'insegnamento)

Il programma prevede i seguenti argomenti:

1. Concetti di base sul calcolo in tempo reale. Domini di applicazione. Tipici Requisiti di sistema. Limiti degli approcci tradizionali. Modelli di attività computazionali. Tipologie di vincoli temporali. Algoritmi di scheduling. Metriche per la valutazione delle prestazioni.
2. Algoritmi di scheduling in tempo reale. Tassonomia degli algoritmi esistenti. Schedulazione con vincoli di precedenza. Scheduling di attività periodiche. Analisi di schedulabilità. Analisi dei tempi di risposta.
3. Protocolli per l'accesso alle risorse condivise. Fenomeno di inversione di priorità e metodi per evitarlo: Non-Preemptive Protocol, Highest Locker Priority, Priority Inheritance Protocol, Priority Ceiling Protocol, Stack Resource Policy. Stima dei tempi di bloccaggio. Analisi dei tempi di risposta in presenza di protocolli di accesso a risorse condivise.
4. Gestione delle attività aperiodiche. Server a priorità fissa (Polling Server, Deferrable Server e Sporadic Server). Server di priorità dinamica (Total Bandwidth Server e Constant Bandwidth Server). Analisi temporale in presenza di server aperiodici.
5. Gestione dei sovraccarichi. Definizione di carico computazionale. Metodi per la gestione dei sovraccarichi. Tecniche di Admission Control, Robust Scheduling, Imprecise Computation, Job Skipping, Elastic Task. Gestione degli overrun e tecniche di Resource Reservation per l'isolamento temporale tra applicazioni.
6. Linee guida per lo sviluppo di software in tempo reale. Esempi di programmazione di software in tempo reale in Linux.

### Bibliografia e materiale didattico

1. Giorgio Buttazzo, "Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications", Third Edition, Springer, 2011.
2. Giorgio Buttazzo: "Sistemi in Tempo Reale", Terza Edizione, Pitagora Editrice, Bologna, 2006.

Materiale aggiuntivo sarà fornito sul sito web del corso.

### Indicazioni per non frequentanti

La frequentazione del corso è altamente consigliata per meglio comprendere gli aspetti teorici e affrontare correttamente lo sviluppo del progetto.

### Modalità d'esame

L'esame prevede lo sviluppo di un progetto, la sua discussione e una prova scritta sugli argomenti del corso. Si consiglia vivamente di partecipare a tutte le lezioni per comprendere meglio la parte teorica e affrontare correttamente lo sviluppo del progetto.

### Stage e tirocini

Nessuno.

### Pagina web del corso

<http://retis.sssup.it/~giorgio/rt-MECS.html>

### Altri riferimenti web

Nessuno.

Ultimo aggiornamento 15/05/2017 10:58