



UNIVERSITÀ DI PISA

REAL TIME AND DISTRIBUTED SYSTEMS

GIORGIO CARLO BUTTAZZO

Academic year 2018/19
Course EMBEDDED COMPUTING SYSTEMS
Code 608II
Credits 12

Modules	Area	Type	Hours	Teacher(s)
CONCURRENT DISTRIBUTED SYSTEMS	ING-INF/05	LEZIONI	60	LUCA ABENI ALESSANDRO BIONDI
REAL TIME SYSTEMS	ING-INF/05	LEZIONI	60	GIORGIO CARLO BUTTAZZO

Obiettivi di apprendimento

Conoscenze

RTS MODULE:

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.

CDS MODULE:

Il modulo fornisce la conoscenza dei concetti chiave, delle metodologie, degli strumenti e delle tecniche per la programmazione parallela. Gli studenti impareranno i più importanti concetti di programmazione parallela e i metodi per utilizzarli in pratica nella programmazione. Essi impareranno a progettare e sviluppare un programma parallelo, analizzandone le proprietà. Gli studenti impareranno anche come thread o processi multipli possono sincronizzarsi o comunicare utilizzando meccanismi diversi. Le metodologie e gli algoritmi presentati nel modulo "Real-Time Systems" saranno anche estesi per prendere in considerazione più processori/core.

Modalità di verifica delle conoscenze

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

Capacità

RTS MODULE:

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.

CDS MODULE:

Gli studenti acquisiranno competenze nello sviluppo di programmi concorrenti e distribuiti utilizzando ambienti di programmazione esistenti, con particolare attenzione ai thread e ai monitor Java. Utilizzeranno un approccio basato sul modello per la progettazione, la costruzione e la verifica di un programma concorrente.

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.



UNIVERSITÀ DI PISA

Prerequisiti (conoscenze iniziali)

Conoscenze di base su matematica, fisica, architetture di elaborazione, linguaggi di programmazione e sistemi operativi. Il modulo di Real-Time Systems è necessario per comprendere alcuni contenuti del modulo di Concurrent Distributed Systems.

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)

RTS MODULE:

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.

CDS MODULE:

1. Concetti di base sulla programmazione parallela. Cooperazione, concorrenza e interferenza tra processi e thread concorrenti. Macchine concorrenti astratte.
2. Proprietà di un programma parallelo (safeness and liveness) e possibili errori: deadlock, starvation, corse critiche, unfairness.
3. Modello computazionale a memoria condivisa. Meccanismi di sincronizzazione. Lock, barriere, semafori e monitor.
4. Modello computazionale a scambio di messaggi e programmazione distribuita. Primitive di comunicazione tra processi. Primitive sincrone e asincrone a scambio di messaggi, chiamate di procedura remota e Rendezvous nei programmi distribuiti.
5. Algoritmi di scheduling in tempo reale per sistemi multiprocessore.
6. Algoritmi e protocolli di condivisione delle risorse in tempo reale per sistemi multiprocessore.

Bibliografia e materiale didattico

RTS MODULE:

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.

CDS MODULE:

1. Gregory R. Andrews, "Foundations of Multithreaded, Parallel and Distributed Programming", Addison Wesley, 2000.
2. Paolo Ancilotti and Aurelio Boari, "Programmazione Concorrente e Distribuita", McGraw-Hill, 2007.
3. Jeff Magee and Jeff Kramer, "Concurrency: State Models & Java Programs", 2nd Edition, John Wiley & Sons, 2006.

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.

Pagina web del corso

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

Ultimo aggiornamento 27/09/2018 16:33