

Real-time Scheduling Simulator

Edoardo Sarri

Software Engineering for Embedded System
Project Work

Giugno 2025



UNIVERSITÀ
DEGLI STUDI
FIRENZE

Introduzione

Obiettivo

Simulare l'esecuzione di un taskset secondo un dato algoritmo di scheduling e un protocollo di accesso alle risorse.

Introduzione

Output

Un file di log contenente la traccia di esecuzione, cioè una sequenza di coppie $\langle \text{tempo}, \text{evento} \rangle$, dove i possibili eventi sono:

- Rilascio di un task.
- Acquisizione e rilascio di una risorsa da parte di un chunk.
- Completamento dell'esecuzione di un chunk o di un job di un task.
- Preemption su un task.
- Deadline miss di un job di un task.
- Fault a livello di chunk o di PCP.

Introduzione

Capacità

- Generare una traccia di esecuzione.
- Generare un dataset di tracce di esecuzione.
- Simulare Rate Monotonic con e senza risorse condivise insieme a Priority Ceiling Protocol.
- Simulare Earliest Deadline First senza risorse condivise.
- Rilevare eventuali deadline miss.
- Controllare la feasibility di un taskset dato il relativo algoritmo di scheduling.

Introduzione

Capacità (fault injection)

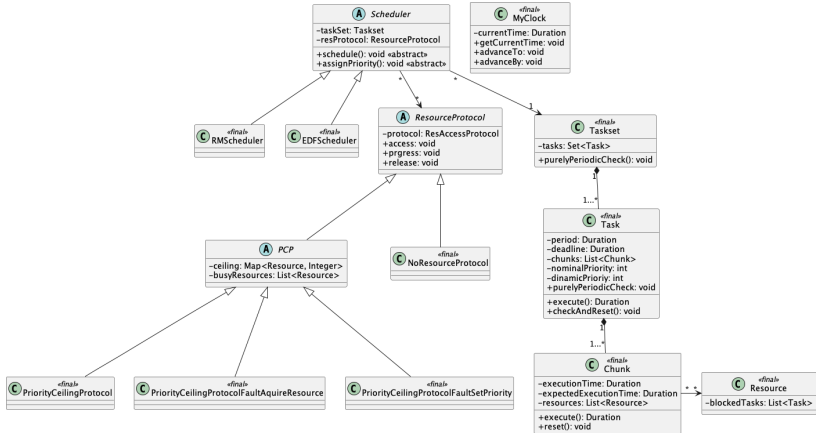
- Introdurre in modo stocastico e rilevare un additional execution time in un chunk.
- Introdurre un fault a livello del protocollo di accesso alle risorse per cui PCP imposta male la priorità dinamica dei task.
- Introdurre un fault a livello di chunk per cui esso non acquisisce (e rilascia) il semaforo della risorsa che userà.

Introduzione

Utilizzo

- All'interno del main devono essere definiti i componenti necessari: Resource, Chunk, Task, TaskSet, Scheduler e ResourceProtocol.
- Per avviare una simulazione chiamare il metodo `schedule` o `scheduleDataset` di uno Scheduler.
- I tempi devono essere passati e letti dal sistema in millisecondi. Il sistema li elabora in nanosecondi per una maggiore precisione.

Analisi

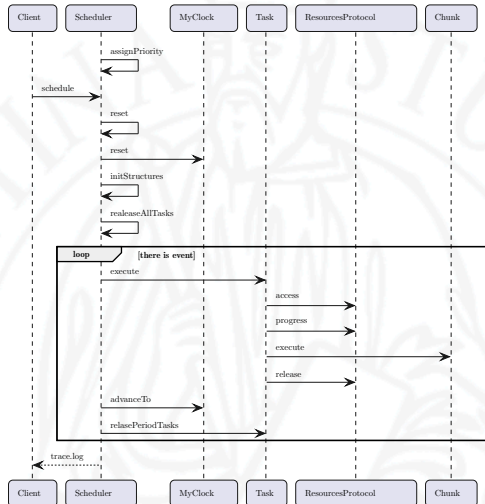


Implementazione

Scheduler

- Classe base che definisce la logica dello scheduling: *Template Method*.
- Classi concrete che implementano `assignPriority` e `addReadyTask`.

W



Implementazione

Resource Access Protocol

- Necessario NoResourceProtocol quando non si hanno risorse condivise.
- Implementa i metodi di accesso, progresso e rilascio.
- PCP usa due strutture dati: `ceiling` e `busyResource`.

Implementazione

Clock

- Gestione globale.
- Accesso unificato tramite Singleton.
- Oggetti di tipo `Duration` di `java.time`.

Sampling

- Libreria Sirio.
- Aggiunto `ConstantSampler` per il campionamento di di un tempo costante.

Real-time Scheduling Simulator

Edoardo Sarri

Grazie



UNIVERSITÀ
DEGLI STUDI
FIRENZE