Real-time Scheduling Simulator

Edoardo Sarri

Software Engineering for Embedded System Project Work

Giugno 2025



Obiettivo

Simulare l'esecuzione di un taskset secondo un dato algortimo 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 < tempo, evento >, 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.
- Genenare un dataset di tracce di esecuzione.
- Simulare Rate Monotonic con e senza risorse convidese insieme a Priority Ceiling Protocol.
- Simulare Farliest Deadline First senza risorse condivise.
- Rilevare eventuali deadline miss.
- Controllare la feasibility di un taskset dato il relativo algortimo di scheduling.

Introduzione

Capacità (fault injection)

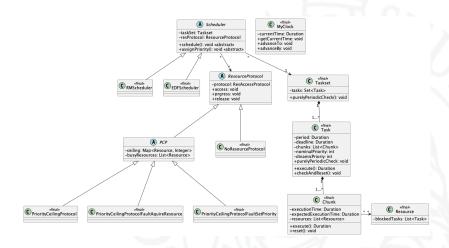
- Introdurre in modo stocastico e rilevare un additional execution time in un chunk
- Introdurre un fault a livello del protocollo di accesso alle risorse per cui PCP imposta male la priorità dinamica dei task.
- Introdurre un fault a livello di chunk per cui esso non acquisisce (e rilascia) il semaforo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



MyClock

Scheduler

ResourcesProtocol

Chunk

Implementazione

Scheduler

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

schedule initStructure realease All Tasks [there is event] execute progress execute release advanceTo relasePeriodTasks trace.log Scheduler MyClock ResourcesProtocol Chunk

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.



GitHub

Dataset

Earliest Deadline First (EDF)

- 5 task: 2 puramente periodici e 3 periodici.
- Fattore di utilizzo di 0,95.
- Chunk che campionano un additional ET da una uniforme.
- traceEDF.log

```
--- continuous eritor magariamatici),

new Sulfarminapler(one Eigheriani)-1), new Eigheriani(1))));

Tookiet taskiet = new Taskietiiet.st(tooki, tooki, tooki, tooki,
```

GitHub

Dataset

Rate Monotonic (RM)

- 5 task puramente periodici.
- 2 risorse condivise.
- Fault injection sull'acquisizione di risorse.
- traceRM.log

GitHub

Real-time Scheduling Simulator

Edoardo Sarri

Grazie

