Real-Time Programming for Robotics

Ludovic Saint-Bauzel (saintbauzel@isir.upmc.fr)

2023

Parallel Programming

- Posix
- Interlocking problems
- Signals
  - Signals, RT Signals, Alarm
- Synchronization
  - Mutex, MailQueue, Shared Memory, FIFO Queue
- Scheduling
  - Threads : FIFO with priority and time quantum
  - RM, Differed, Sporadic, EDF

POSIX – GNU

- GNU’s Not Unix
- Portable Operating System Is not uniX
- pthread
- Syntax
  - pthread_function
  - pthread_mutex_t

Thread vs Process?

- Explain what is a thread ?

Thread

- What is reentrant?

Threads / Process

- Context Switches
  - What is it ?

Interlocking Problem

- Chinese Philosopher eating problem

Djikstra Algorithm

- P() → Take

- V() → Release

Spinlock / Mutex / Semaphore

- Spinlock (While loop)
- Mutex : One and a waiting queue
- Semaphore with counter : (N and waiting queue)

- Synchro Patterns

  Barrier
  - main lock create_threads unlock
  - Threads lock;unlock; (to block)

  Sequencial

  main : for i ; lock(m_i) end create Threadi unlock(m_1)

  thread_i : lock(m_i) blabla unlock(m_i+1) *unlock(m_i)*

  multiple

  main lock(m) create thread_i unlock(m)

  thread 1 sem_wait, sem_wait sem_wait lock(m) unlock(m)… blabla

  thread_i sem_post lock unlock blibli

Signaling Mechanism

- Wait
- Kill

Some useful signals

- Alarm
  - Timer → alarm signal
  - Callback

FIFO Queue

```
mkfifo("/tmp/fifo", 0666);

// Open FIFO for write only
fd = open(myfifo, O_WRONLY);
write(fd, arr2, strlen(arr2)+1);
close(fd);

// Open FIFO for Read only
fd = open(myfifo, O_RDONLY);
read(fd, arr1, sizeof(arr1));
close(fd);
```

Message Queue

- One channel
- Multiple priority

Scheduling

- Determinism
-
- FIFO
- FIFO with priority levels
- FIFO with Round Robin
- Rate Monotonic
- Differed Server
- Differed Server with budget
- Sporadic server (Differed + Budget + BG)

Scheduler : Task model

Parameters for task i

- S_i : start time when task arrives in the scheduler
- C_i : Computation time needed by the task (Capacity).
- P_i : Period
- D_i : Deadline of the task
- R_i : Earliest activation time
- Aperiodic task is defined by :

  (S_i, C_i, D_i, R_i)
- Periodic task is defined by :
  (S_i, C_i, D_i, P_i)

Scheduler FIFO

- Wakes up
- Choses

- When a task is finished
- Next one in the list

Scheduler FIFO

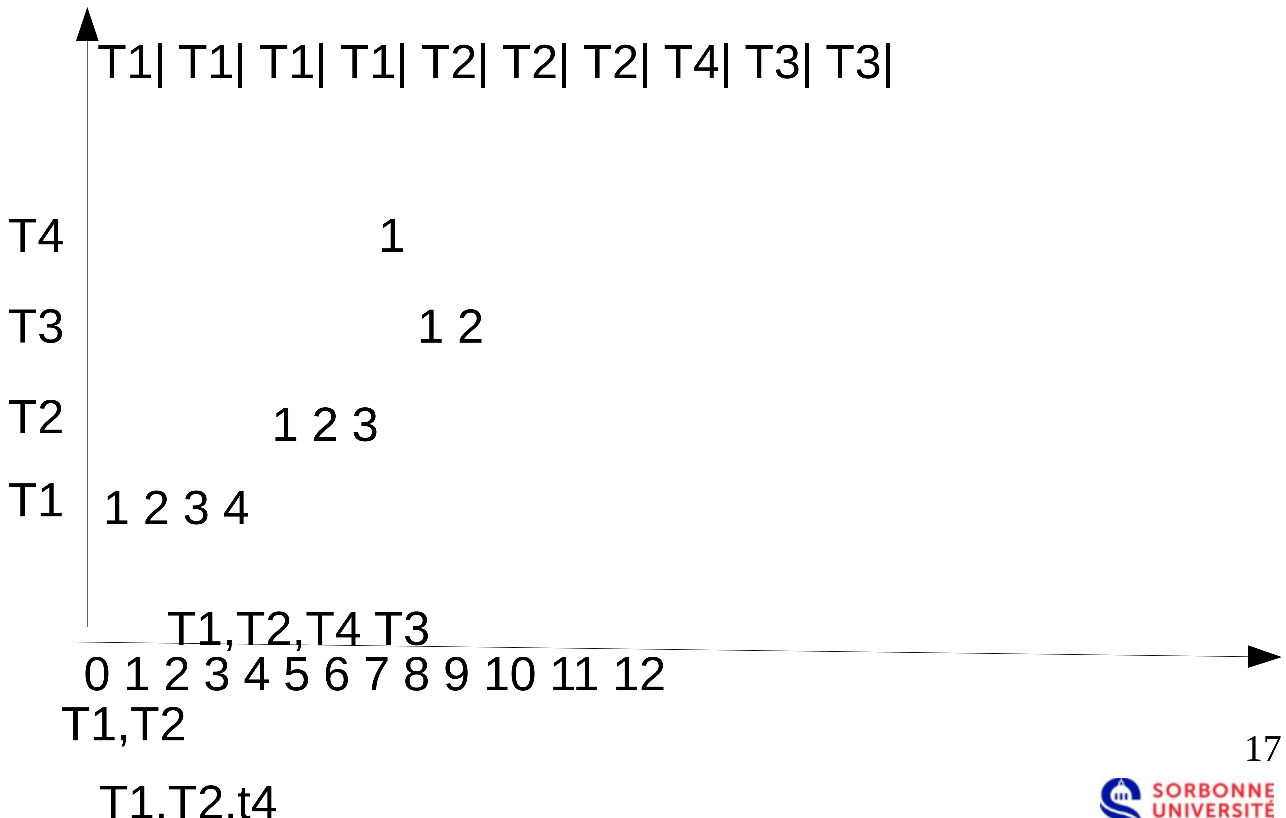
- Exemple

  T1: s:0,c:4

  T2: s:0,c:3

  T3: s:2,c:2

  T4: s:1,c:1



File(t7) = T4 T3

Scheduler FIFO with priority levels

- Wakes up
- Choses

- When a task is finished
- First in the highest priority list

Scheduler FIFO with priority levels

- Exemple

  T1: s:0,c:4, prio:2

  T2: s:0,c:3, prio:2

  T3: s:2,c:2, prio:1

  T4: s:1,c:1, prio:3

Scheduler FIFO with round robin

- Wakes up
- Choses

- When a task finishes
- When a quantum of time (current task is put at the end of the list)

- First in the list

Scheduler FIFO with roundrobin

- Example
- Quantum 2, level 2
- Tasks

  T1: s:0,c:4, prio:2

  T2: s:0,c:3, prio:2

  T3: s:2,c:2, prio:1

  T4: s:1,c:1, prio:3

- 1 1 3 3 2 2 1 1 2 4
-
- Q=3
- 1 1 1 3 3 2 2 2 1 4

Scheduler rate monotonic

- Works only with periodic tasks
- Wakes up
- Choses

Non-pre-emptive
Pre-emptive

- When
  - A task finishes
  - A task arrives
- Smallest period first

Scheduler rate monotonic

- Example
- Periodic Tasks

  T1: s:0,c:4, p:15

  T2: s:0,c:3, p:10

  T3: s:2,c:2, p:5

  T4: s:1,c:1, P:10

- 2 2 3 3 2 4 1 3 3 1 2 2

  3 3 2 4 **1**

- **Soucis de 2 taches T1 en meme temps**

SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

Scheduler rate monotonic

- Schedulable?

  Sufficient condition

$$U = \sum_{i}^{n} \frac{C_i}{T_i} \leq n \times (2^{\frac{1}{n}} - 1)$$

Scheduler rate monotonic

- Response time computation

$$TR_i = C_i + \sum_{j \in hp(i)} I_j$$

$$TR_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{TR_i}{P_j} \right\rceil C_j$$

Where $hp(i)$ represents a set of tasks with a higher priority than i.

Scheduler rate monotonic

- Response time **iterative** computation

$$w_i^{n+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^n}{P_j} \right\rceil C_j$$

- Let start with $w_i^0 = C_i$
- Fail if $w_i^n > P_i$
- Achieved if $w_i^{n+1} = w_i^n$

Example : T1 (c:3, p:7), T2(c:2,p:12),T3 (c:5, P:20)

Result : TR1=3, TR2=5,TR3=18

Exec : 11122331113322111 3

• Scheduler Differed Server

- Wakes up
  - Managing aperiodic tasks called events

Non-pre-emptive
Pre-emptive

- When
  - A task or an event finishes
  - A task or an event arrives

- Choses

- Periodic tasks
  - Rate monotonic
  - Smaller period is chosen
- Server (A periodic task)
  - Smallest period (aka highest prio)
  - Fifo in event list

Scheduler RM with differed server

- Example
- Tasks

  DS: s:0 c:1, p:5

  T1: s:0,c:4, p:12

  e1: s:0,c:3, d: 10

  e2: s:2,c:2, d:10

  e3: s:1,c:1, d:10

Scheduler Aperiodic
Differed server with Budget

- Wakes up
  - Managing aperiodic tasks called events

- Choses

Non-pre-emptive
Pre-emptive

- When
  - A task or an event finishes
  - **When a budget of server is consumed**
  - A task or an event arrives

- Periodic tasks
  - Rate monotonic
  - Smaller period
- Server (A periodic task)
  - Smallest period (aka highest prio)
  - Fifo in event list

Scheduler Aperiodic Differed server with Budget

- example

Scheduler Sporadic Differed server with Budget and Background

- Wakes up
  - Managing aperiodic tasks called events

Non-pre-emptive
Pre-emptive

- When
  - A task or an event finishes
  - When a budget of server is consumed
  - A task or an event arrives

- Choses

- Periodic tasks
  - Rate monotonic
  - Smaller period
- Server (A periodic task)
  - Smallest period (aka highest prio)
  - Fifo in event list
- Background task
  - Wakes up when no tasks anymore

Scheduler Sporadic Differed server with Budget and Background

- example

Scheduler EDF

- Wakes up
  - Any tasks (periodic, events)

Non-preemptive

Preemptive

- When a tasks finishes
- When a tasks arrives

- Choses

- Earliest deadline is most prior

Scheduler EDF

- Schedulable?

- Necessary and sufficient condition if $\forall i, D_i = P_i$; Only necessary if $\exists i, D_i \leq P_i$:

$$U = \sum_{j=1}^{n} \frac{C_j}{P_j} \leq 1$$

- Sufficient condition if $\exists i, D_i \leq P_i$:

$$U = \sum_{j=1}^{n} \frac{C_j}{D_j} \leq 1$$

Scheduler EDF

- Example

T1 (c:2,s:0,p:10), T2 (c:3,s:2,d:6),
T3(c:3,s:2,p:10)

Real-Time Programming for Robotics

Ludovic Saint-Bauzel (saintbauzel@isir.upmc.fr)