Lecture 10 – Real-time Scheduling

CSE 456: Embedded Systems

Real-Time Scheduling of Mixed Task Sets

Problem of Mixed Task Sets

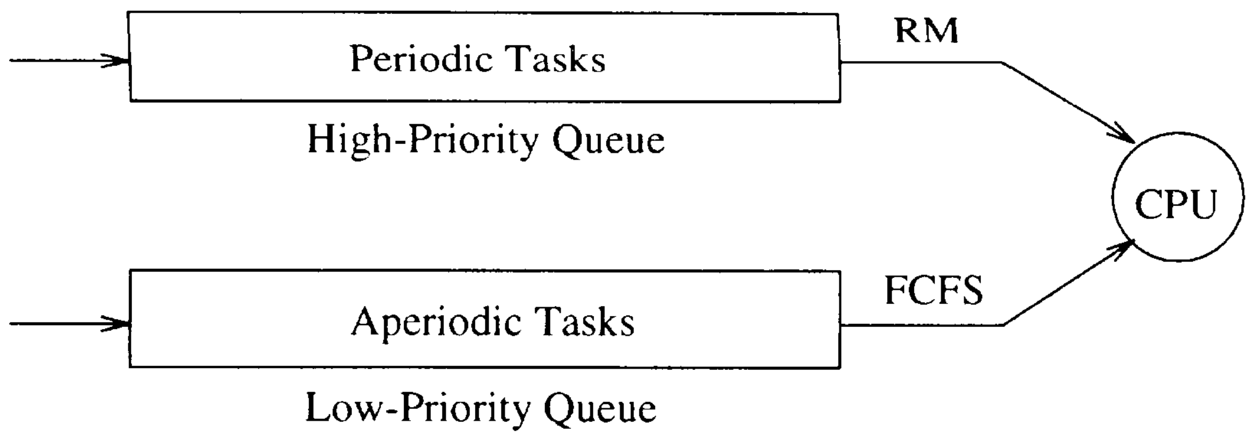
In many applications, there are aperiodic as well as periodic tasks.

- *Periodic tasks: time-driven*, execute critical control activities with hard timing constraints aimed at guaranteeing regular activation rates.
- *Aperiodic tasks: event-driven*, may have hard, soft, non-real-time requirements depending on the specific application.

Background Scheduling

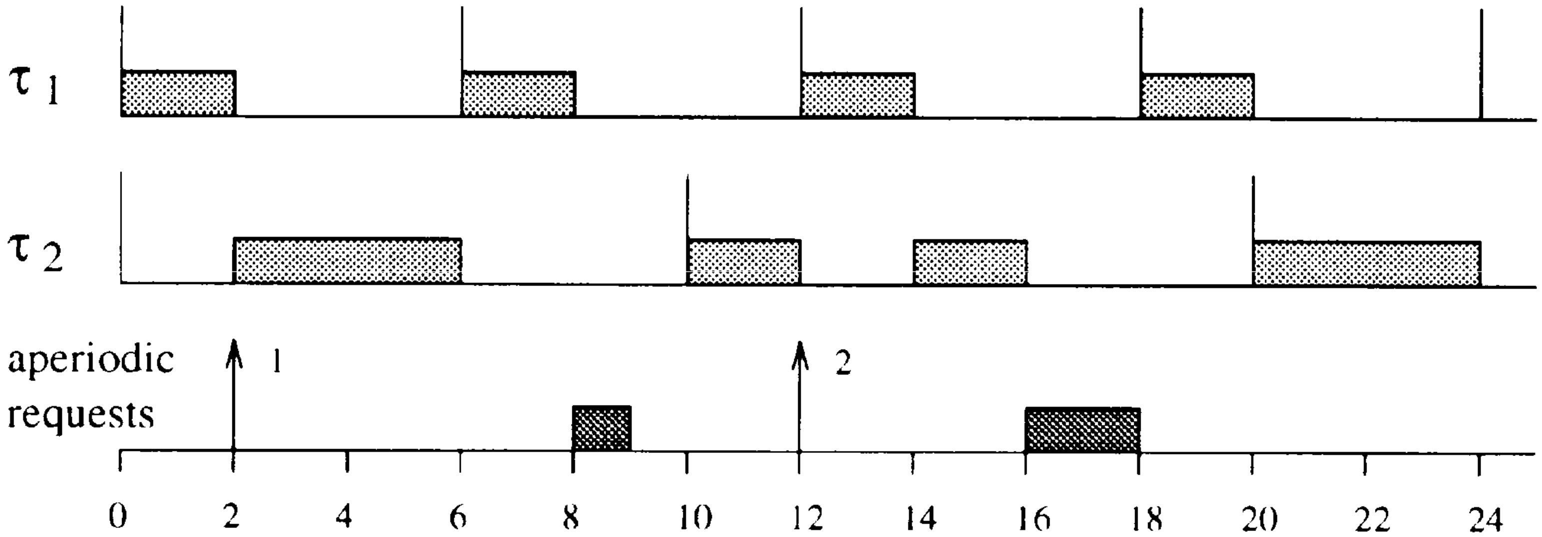
Background scheduling is a simple solution for RM and EDF:

- Processing of aperiodic tasks in the background, i.e. execute if there are no pending periodic requests.
- Periodic tasks are not affected.
- Response of aperiodic tasks may be prohibitively long and there is no possibility to assign a higher priority to them.
- Example:



Background Scheduling

Example (rate monotonic periodic schedule):



Rate-Monotonic Polling Server

- ***Idea:*** Introduce an artificial periodic task whose purpose is to service aperiodic requests as soon as possible (therefore, “server”).
- Function of *polling server (PS)*
  - At regular intervals equal to T_S, a PS task is instantiated. When it has the highest current priority, it serves any pending aperiodic requests within the limit of its capacity C_S.
  - If no aperiodic requests are pending, PS suspends itself until the beginning of the next period and the time originally allocated for aperiodic service is not preserved for aperiodic execution.
  - Its priority (period!) can be chosen to match the response time requirement for the aperiodic tasks.
- *Disadvantage:* If an aperiodic requests arrives just after the server has suspended, it must wait until the beginning of the next polling period.

Rate-Monotonic Polling Server

Example:

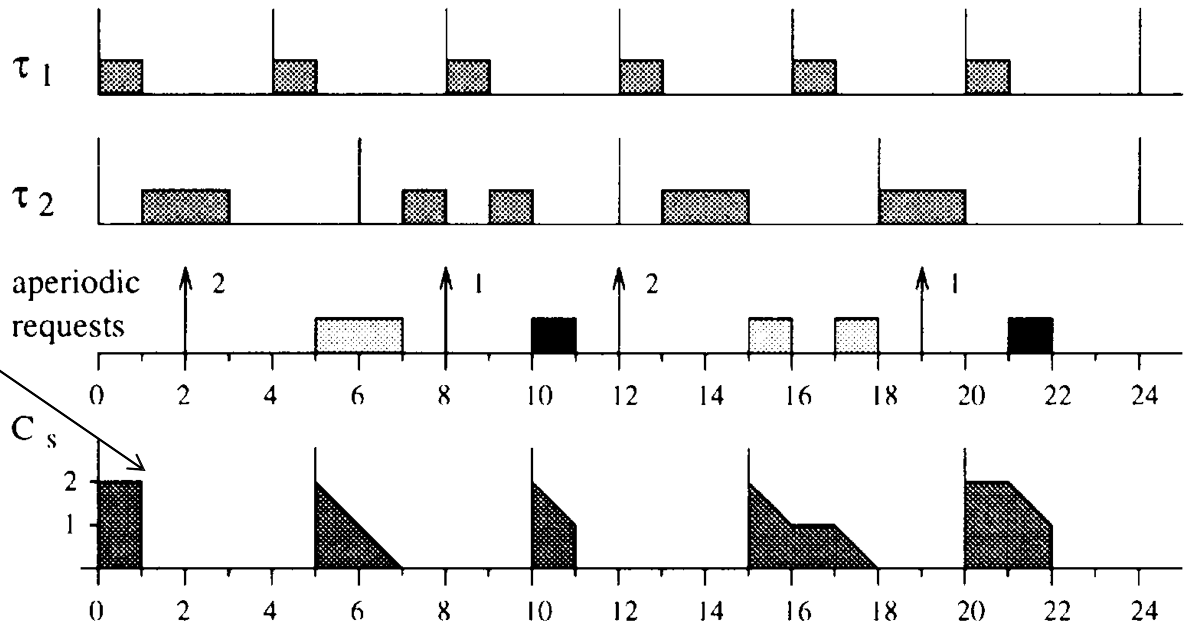
	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$

server has current highest priority and checks the queue of tasks



Rate-Monotonic Polling Server

Schedulability analysis of periodic tasks:

- The interference by a server task is the same as the one introduced by an equivalent periodic task in rate-monotonic fixed-priority scheduling.
- A set of periodic tasks and a server task can be executed within their deadlines if

$$\frac{C_s}{T_s} + \sum_{i=1}^{n} \frac{C_i}{T_i} \le (n+1)\left(2^{1/(n+1)} - 1\right)$$

- Again, this test is sufficient but not necessary.

Summary

	Periodic with D = T	Periodic with D < T	Mixed Tasks
Static Priority	RM	DM	Polling Server
Dynamic Priority	EDF	EDF	Total Bandwidth Server

RM – Schedulability Test

Sufficient (but not necessary)

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

Summary

	Periodic with D = T	Periodic with D < T	Mixed Tasks
Static Priority	RM	DM	Polling Server
Dynamic Priority	EDF	EDF	Total Bandwidth Server

DM – Schedulability Test

Sufficient (but not necessary)

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

Summary

RM & DM – Schedulability Test

```
Algorithm: DM_guarantee (Γ)
{       for (each τi∈Γ){
                I = 0;
                do {
                        R = I + Ci;
                        if (R > Di) return(UNSCHEDULABLE);
                        I = ∑j=1,…,(i-1)⌈R/Tj⌉ Cj;
                } while (I + Ci > R);
        }
        return(SCHEDULABLE);
}
```

Necessary & Sufficient

Longest Response Time R_i (computed iteratively)

$$\overset{R_i}{\cancel{I_i}} = \sum_{j=1}^{i-1} \left\lceil \frac{\overset{R_i}{\cancel{t}}}{T_j} \right\rceil C_j + C_i$$

Summary

	Periodic with D = T	Periodic with D < T	Mixed Tasks
Static Priority	RM	DM	Polling Server
Dynamic Priority	EDF	EDF	Total Bandwidth Server

EDF – Schedulability Tests

$D_i = T_i$	$D_i < T_i$
Necessary & Sufficient	**Sufficient (but not necessary)**
$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1$	$\sum_{i=1}^{n} \frac{C_i}{D_i} \leq 1$

Utilization: $U = \sum_{i=1}^{n} \frac{C_i}{T_i}$

Summary

	Periodic with D = T	Periodic with D < T	Mixed Tasks
Static Priority	RM	DM	Polling Server
Dynamic Priority	EDF	EDF	Total Bandwidth Server

RM – Polling Server

- **Idea**: Introduce an artificial periodic task (C_s, T_s) which serves the aperiodic requests

Schedulability test for mixed task set:

$$\frac{C_s}{T_s} + \sum_{i=1}^{n} \frac{C_i}{T_i} \leq (n+1)(2^{\frac{1}{n+1}} - 1)$$

Sufficient (but not necessary)

Aperiodic guarantee:

$$(1 + \left\lceil \frac{C_a}{C_s} \right\rceil) T_s \leq D_a$$

Assumption: aperiodic task finishes before new aperiodic request arrives

Sufficient (but not necessary)