Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

Real-Time Systems

3a-Cyclic Scheduler

Antonio Camacho Santiago
antonio.camacho.santiago@upc.edu

Objective

To understand cyclic schedulers…

To design cyclic schedulers…

To analyze cyclic schedulers…

To evaluate the pros and cons of cyclic schedulers…

```
wait timer interrupt
frame=frame+1;
switch frame
  case 1:
    Task1(); Task2(); break;
  case 2:
    Task3(); Task4(); break;
  case 3:
    Task(5); break;
if (frame≥3) frame=0;
```

Cyclic scheduler

The scheduler chooses which task will be executed at each time instant.

Scheduling has two parts:

- -Algorithm: specifies how to assign a task to be executed in the processor(s) along time
- -Analysis: by analyzing the resulting policy, it is possible to guarantee timing constraints for RT-systems

The cyclic scheduler is the basis for most embedded systems
It is based on periodic tasks
Time constraints are ensured by design
Requires a time mechanism to generate periodic triggers of the scheduler

More details can be found in T. P. Baker and A. Shaw, "The cyclic executive model and Ada,“ *Proceedings. Real-Time Systems Symposium*, Huntsville, AL, USA, 1988, pp. 120-129.

Cyclic scheduler

Some maths

- lcm less common múltiple (mínimo común mútiplo)
- gcd greatest common divisor (máximo común divisor)
- $\wedge$ and operator
- $\lfloor\cdot\rfloor$ floor operator
- $\lceil\cdot\rceil$ ceiling operator
- $\forall$ for each
- $\exists$ exists
- $\in$ belongs to
- Z^+ positive integers

Requisites

The first approach for the cyclic scheduler is based on periodic tasks as follows:

- 1 microprocessor
- Static tasks
- Periodic tasks
- No precedence among tasks
- The WCET for each tasks is known, fitted and less than its deadline
- Deadlines of each task equal to their periods

The schedulability analysis tries to know in advance if all the release times for each task occurs before its deadline

Methodology

Necessary but not sufficient conditions:

Check utilization factor

$$U_{\text{total}} = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \frac{c_i}{T_i} = \frac{c_1}{T_1} + \frac{c_2}{T_2} + \cdots + \frac{c_n}{T_n} \leq 1 \tag{1}$$

Find the hiperperiod H as

$$H = \operatorname*{lcm}_{1 \leq i \leq n} T_i \tag{2}$$

Find the secondary frame T_s which has to satisfy

$$T_s \geq \max_{1 \leq i \leq n} c_i \quad \land \quad T_s \leq \min_{1 \leq i \leq n} D_i \tag{3}$$

$$H = k\, T_s \;, k \in Z^+ \tag{4}$$

$$\forall i: 2T_s - \gcd(T_s, T_i) \leq D_i \tag{5}$$

Once conditions are fulfilled, find suitable allocation of each task within the frames

Example 1

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)
τ_1	5	20
τ_2	5	20
τ_3	10	40

Check utilization

$$U_{\text{total}} = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \frac{c_i}{T_i} = \frac{5}{20} + \frac{5}{20} + \frac{10}{40} = 0.75 \leq 1$$

Hiperperiod *H*

$$H = \underset{1 \leq i \leq n}{\text{lcm}}\ T_i = \text{lcm}(20,20,40) = \text{lcm}(2^2 \cdot 5, 2^2 \cdot 5, 2^3 \cdot 5) = 40 \text{ ms}$$

Secondary period

$$T_s \geq \max_{1 \leq i \leq n} c_i = \max(5,5,10) \geq 10 \text{ ms}$$

$$T_s \leq \min_{1 \leq i \leq n} D_i = \min(20,20,40) \leq 20 \text{ ms}$$

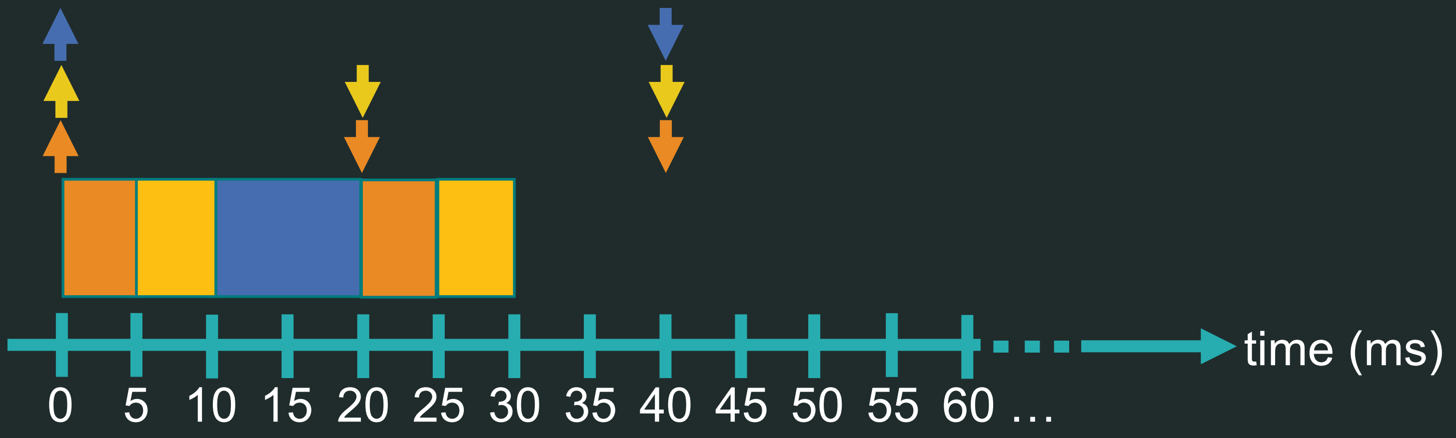
$$\left.\begin{aligned} H &= k\,T_s = 4 \cdot 10\text{ms} = 40\text{ms}, k = 4 \in Z^+ \\ H &= k\,T_s = 2 \cdot 20\text{ms} = 40\text{ms}, k = 2 \in Z^+ \end{aligned}\right\} T_s = \{10,20\}\text{ms}$$

Example 1

For $T_s = 20\text{ms}$ check $\forall i: 2T_s - \gcd(T_s, T_i) \leq D_i$

$i = \{1,2\}:\ 2 \cdot 20 - \gcd(20,20) = 40 - 20 = 20 \leq 20$
$i = 3:\ 2 \cdot 20 - \gcd(20,40) = 40 - 20 = 20 \leq 40$

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)
τ_1	5	20
τ_2	5	20
τ_3	10	40




```
wait timer interrupt(20ms)
frame=frame+1;
switch frame
   case 1:
      Task1();
      Task2();
      Task3();
      break;
   case 2:
      Task1();
      Task2();
      break;
if (frame≥2) frame=0;
```

Example 2

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)
τ_1	2	6
τ_2	2	8
τ_3	8	24

Check utilization

$$U_{\text{total}} = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \frac{c_i}{T_i} = \frac{2}{6} + \frac{2}{8} + \frac{8}{24} = 0.92 \leq 1$$

Hiperperiod *H*

$$H = \operatorname*{lcm}_{1 \leq i \leq n} T_i = \text{lcm}(6,8,24) = \text{lcm}(2 \cdot 3, 2^3, 2^3 \cdot 3) = 24 \text{ ms}$$

Secondary period

$$\left.\begin{aligned} T_s &\geq \max_{1 \leq i \leq n} c_i = \max(2,2,8) \geq 8 \text{ ms} \\ T_s &\leq \min_{1 \leq i \leq n} D_i = \min(6,8,24) \leq 6 \text{ ms} \end{aligned}\right\}$$

Does not fulfill condition (3). Split task 3 into pieces

Such that: $T_{3A} = T_3$
$T_{3B} = T_3$
$D_{3A} = D_3$
$D_{3B} = D_3$
$c_{3A} + c_{3B} = c_3$
c_{3A} precedes to c_{3B} and no critical section in the edge

Example 2

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)
τ_1	2	6
τ_2	2	8
τ_{3A}	4	24
τ_{3B}	4	24

Check utilization

$$U_{\text{total}} = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \frac{c_i}{T_i} = \frac{2}{6} + \frac{2}{8} + \frac{4}{24} + \frac{4}{24} = 0.92 \leq 1$$

Hiperperiod *H*

$$H = \underset{1 \leq i \leq n}{\operatorname{lcm}} T_i = \operatorname{lcm}(6,8,24,24) = \operatorname{lcm}(2 \cdot 3, 2^3, 2^3 \cdot 3, 2^3 \cdot 3) = 24 \text{ ms}$$

Secondary period

$$\left.\begin{aligned} T_s &\geq \max_{1 \leq i \leq n} c_i = \max(2,2,4,4) \geq 4 \text{ ms} \\ T_s &\leq \min_{1 \leq i \leq n} D_i = \min(6,8,24,24) \leq 6 \text{ ms} \end{aligned}\right\} \text{Now fulfills condition (3)}$$

$$\left.\begin{aligned} H &= k\, T_s = 6 \cdot 4\text{ms} = 24\text{ms}\,, k = 4 \in Z^+ \\ H &= k\, T_s = 4 \cdot 6\text{ms} = 24\text{ms}\,, k = 2 \in Z^+ \end{aligned}\right\} \quad T_s = \{4,6\}\text{ms}$$

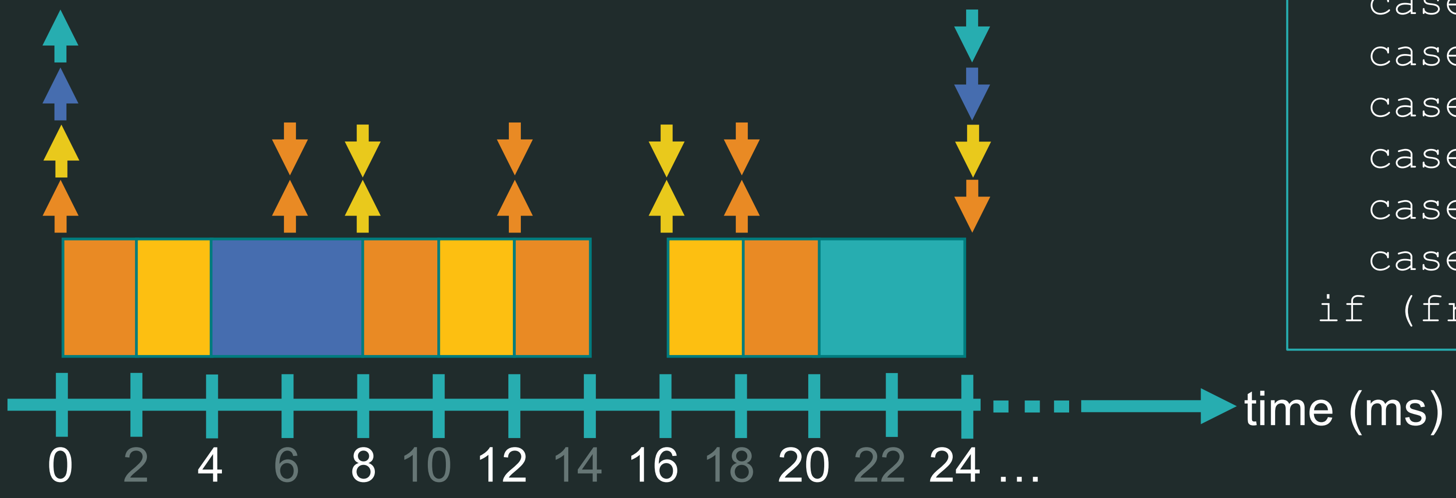
Example 2

For $T_s = 6\text{ms}$ check $\forall i: 2T_s - \gcd(T_s, T_i) \le D_i$
$i = 1{:}\ 2 \cdot 6 - \gcd(6{,}6) = 12 - 6 = 6 \le 6$
$i = 2{:}\ 2 \cdot 6 - \gcd(6{,}8) = 12 - 2 = 10 \ge 8$

For $T_s = 4\text{ms}$ check $\forall i: 2T_s - \gcd(T_s, T_i) \le D_i$
$i = 1{:}\ 2 \cdot 4 - \gcd(4{,}6) = 8 - 2 = 6 \le 6$
$i = 2{:}\ 2 \cdot 4 - \gcd(4{,}8) = 8 - 4 = 4 \le 8$
$i = \{3{,}4\}{:}\ 2 \cdot 4 - \gcd(4{,}24) = 8 - 4 = 4 \le 24$

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)
τ_1	2	6
τ_2	2	8
τ_{3A}	4	24
τ_{3B}	4	24

```
wait timer interrupt(4ms)
frame=frame+1;
switch frame
  case 1: Task1();Task2();break;
  case 2: Task3A(); break;
  case 3: Task1();Task2();break;
  case 4: Task1();break;
  case 5: Task1(); Task2();break;
  case 6: Task3B(); break;
if (frame≥6) frame=0;
```



Example 3

Check utilization

$$U_{\text{total}} = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \frac{c_i}{T_i} = \frac{5}{20} + \frac{10}{20} + \frac{10}{40} = 1 \leq 1$$

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)
τ_1	5	20
τ_2	10	20
τ_3	10	40

Hiperperiod *H*

$$H = \operatorname*{lcm}_{1 \leq i \leq n} T_i = \operatorname{lcm}(20{,}20{,}40) = \operatorname{lcm}(2^2 \cdot 5, 2^2 \cdot 5, 2^3 \cdot 5) = 40 \text{ ms}$$

Secondary period

$$\left.\begin{array}{l} T_s \geq \max\limits_{1 \leq i \leq n} c_i = \max(5{,}10{,}10) \geq 10 \text{ ms} \\ T_s \leq \min\limits_{1 \leq i \leq n} D_i = \min(20{,}20{,}40) \leq 20 \text{ ms} \end{array}\right\} \text{Now fulfills condition (3)}$$

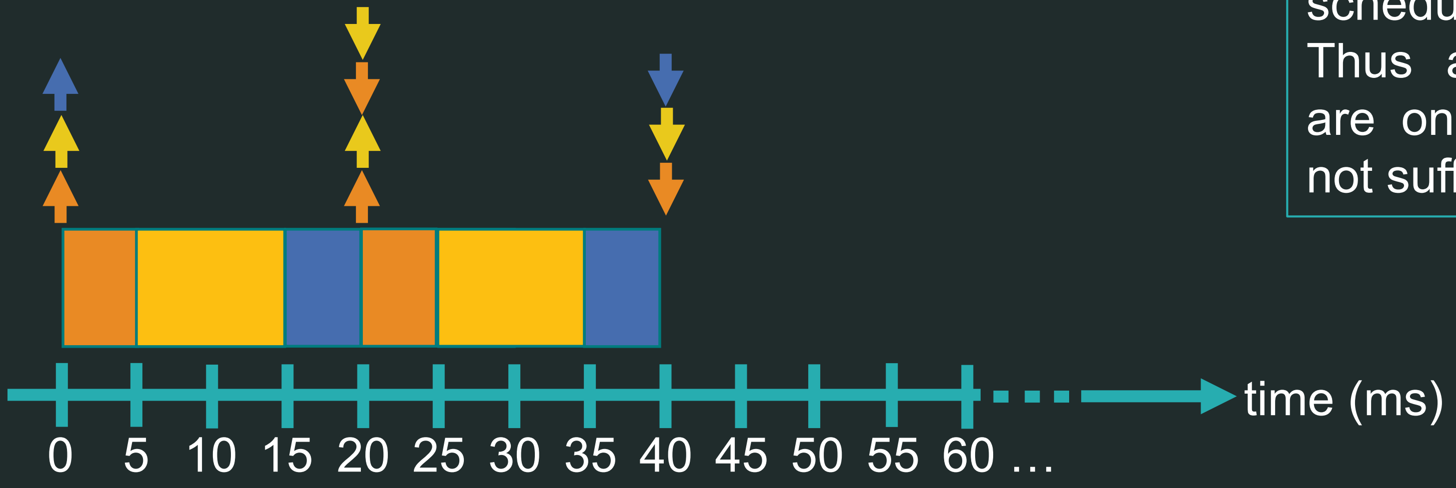
$$\left.\begin{array}{l} H = k\, T_s = 4 \cdot 10\text{ms} = 40\text{ms}, k = 4 \in Z^+ \\ H = k\, T_s = 2 \cdot 20\text{ms} = 40\text{ms}, k = 2 \in Z^+ \end{array}\right\} T_s = \{10{,}20\}\text{ms}$$

Example 3

For $T_s = 20\text{ms}$ check $\forall i: 2T_s - \gcd(T_s, T_i) \leq D_i$
$i = \{1,2\}:\ 2 \cdot 20 - \gcd(20,20) = 40 - 20 = 20 \leq 20$
$i = 3:\ 2 \cdot 20 - \gcd(20,40) = 40 - 20 = 20 \leq 40$

For $T_s = 10\text{ms}$ check $\forall i: 2T_s - \gcd(T_s, T_i) \leq D_i$
$i = \{1,2\}:\ 2 \cdot 10 - \gcd(10,20) = 20 - 10 = 10 \leq 20$
$i = 3:\ 2 \cdot 10 - \gcd(10,40) = 20 - 10 = 30 \leq 30$

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)
τ_1	5	20
τ_2	10	20
τ_3	10	40



Only segmentation of task 3 allows for cyclic scheduling.
Thus above conditions are only necessary but not sufficient

Cyclic scheduler

Pros:

- The cyclic scheduler is static, simple, easy to handle and robust
- Deadlines are ensured by design
- There is no real concurrency nor preemption
- There is no need for mutual exclusion
- Low-level scheduler

Cons:

- Cyclic schedulers are not flexible
- Segmentation of tasks increases complexity
- Not suited for sporadic tasks (empty slots accommodation)
- It can be hard to develop the allocation of the tasks within the frames.
- Low-level scheduler