Exercise 10: Real-time systems

Issue Date: 5. June 2003
Discussion Date: 11. June 2003

Task 1: Real-time Systems concept

What are the main differences between the fast computing and the real-time computing? Think of some critical applications which require a real-time system support. What are the most important features requested? In which situations, instead, would a fast-computing system do?

Solution - Task 1

**fast computing.**

- the *objective* is to minimize the average response time of a given set of tasks
- no guarantees for individual timing requirements of each task in all possible circumstances
- *applications* word processing, multimedia, ...

**real time computing.**

- the *objective* is to meet the individual timing requirement of each task
- correct behavior depends on both (1) correct computation and (2) time at which results are produced
- *system* that must react within precise timing constraints to events in the environment
- characterized by a *deadline*
- should be *predictable* (when the timing constraints cannot be met, this must be notified in advance, so that an alternative (scheduling) plan may be planned, and possibly avoid the catastrophe)
- *applications* see the page 1 (Chapter 1)
- *the most important features* timeliness, design for peak load, predictability, fault tolerance, maintainability.
Task 2: Non-RT Scheduling

Given is a set of independent tasks $T_1, \cdots, T_6$ with execution times and arrival times shown in the following table.

<table>
<thead>
<tr>
<th></th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$T_3$</th>
<th>$T_4$</th>
<th>$T_5$</th>
<th>$T_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$ [m.s]</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>20</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>$C_i$ [m.s]</td>
<td>20</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>20</td>
<td>5</td>
</tr>
</tbody>
</table>

Determine the schedules generated by following algorithms:

a) FCFS (first come first served)  
b) SJF (shortest job first)  
c) SRTN (shortest remaining time next)  
d) preemptive scheduling with fixed priorities 2, 1, 4, 3, 6, 7 for tasks $T_1, T_2, \cdots, T_6$ (1 is the highest priority)  
e) RR (round-robin) with a time slice duration of 2.5 m.s.

Compare the schedules according to the parameters i) processor utilization, ii) average response time and iii) average waiting time.

Solution - Task 2

In the following are the graphs showing the CPU utilization, according to different scheduling schemes of points a), \ldots, e).

![Figure 1: FCFS - First Come First Served.](https://example.com/schedules.ps)

(for other figures - schedules, please give a look at the file 'schedules.ps')
The i) processor utilization, ii) average response time and iii) average waiting time are computed according to the following formula:

\[
\text{CPU Utilization} = \frac{\text{computation time}}{\text{total completion time}}
\]

\[
\text{Avg Response Time} = \frac{1}{n} \sum_{i=1}^{n} (f_i - a_i)
\]

\[
\text{Avg Waiting Time} = \frac{1}{n} \sum_{i=1}^{n} (s_i - a_i)
\]

The values for different scheduling schemes are as follows:

<table>
<thead>
<tr>
<th></th>
<th>FCFS</th>
<th>SJB</th>
<th>SRTN</th>
<th>Fix</th>
<th>RR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Util</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>Avg Resp Time</td>
<td>23.3</td>
<td>16.6</td>
<td>15</td>
<td>20</td>
<td>19.583</td>
</tr>
<tr>
<td>Avg Wait Time</td>
<td>12.5</td>
<td>5.83</td>
<td>3.3</td>
<td>9.16</td>
<td>0.83</td>
</tr>
</tbody>
</table>
Task 3: Design of a Real-time System

For the real-time system scheduling, it is important to know the running time of each task \textit{a priori}. What are the difficulties encountered in a design of a scheduling algorithm for a real-time system. What are the possible solutions? How can we compare different scheduling algorithms?

Solution - Task 3
Task 4: Scheduling Function

A real-time system had to execute four tasks $J_1$, $J_2$, $J_3$, $J_4$ with arrival times and deadlines shown in the following table. The scheduling function $\sigma(t)$ observed is shown in Figure 2.

<table>
<thead>
<tr>
<th></th>
<th>$J_1$</th>
<th>$J_2$</th>
<th>$J_3$</th>
<th>$J_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>arrival time</strong></td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>deadline</strong></td>
<td>10</td>
<td>10</td>
<td>14</td>
<td>13</td>
</tr>
</tbody>
</table>

Figure 2: Scheduling Function

Determine

a) the maximum lateness,

b) the tasks’ laxities, and

c) the processor utilization for this schedule.

Is the schedule feasible? If not, try to modify the scheduling function so that the schedule becomes feasible.
Solution - Task 4

The *lateness* of a task is a delay of a task completion with respect to its deadline (note that if the task finishes before its deadline, its lateness is negative).

The maximum lateness is of the task $J_2 = f_i - d_i = 12 - 10 = 2$ (task $J_2$ is the only task that violates the given constraints).

The *laxity* (or *slack time*) is the maximum time that a task can be delayed on its activation to complete within its deadline.

The laxities of the tasks $J_1, \ldots, J_4$ are as follows (in the table below are the individual tasks’ computation times):

1. $X_1 = d_1 - a_1 - C_1 = 10 - 0 - 4 = 6$
2. $X_2 = 10 - 3 - 2 = 5$
3. $X_3 = 14 - 0 - 4 = 10$
4. $X_4 = 13 - 3 - 3 = 7$

<table>
<thead>
<tr>
<th>$J_1$</th>
<th>$J_2$</th>
<th>$J_3$</th>
<th>$J_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

The CPU Utilization is $13/14 = 0.93$.

A schedule is said to be *feasible* when all tasks can be completed according to a set of specified constraints. The scheduling function suggested in the exercise, $\sigma(t)$ is thus not feasible (the task $J_2$ does not meet its deadline).

![Figure 3: Violation in the old scheduling function](image1)

The schedule can be modified in order to make all tasks completed before their respective deadlines. The new scheduling function is depicted in the figure.

![Figure 4: New scheduling function](image2)