Exercises

1.1 Explain the difference between fast computing and real-time computing.

1.2 What are the main limitations of the current real-time kernels for the development of critical control applications?

1.3 Discuss the features that a real-time system should have for exhibiting a predictable timing behavior.

1.4 Describe the approches that can be used in a real-time system to handle peripheral I/O devices in a predictable fashion.

1.5 Which programming restrictions should be used in a programming language to permit the analysis of real-time applications? Suggest some extensions that could be included in a language for real-time systems.

Exercises

2.1 Give the formal definition of a schedule, explaining the difference between preemptive and non-preemptive scheduling.

2.2 Explain the difference between periodic and aperiodic tasks, and describe the main timing parameters that can be defined for a real-time activity.

2.3 Describe a real-time application as a number of tasks with precedence relations, and draw the corresponding precedence graph.

2.4 Discuss the difference between static and dynamic, on-line and off-line, optimal, and heuristic scheduling algorithms.

2.5 Provide an example of domino effect, caused by the arrival of a task J^*, in a feasible set of three tasks.

Exercises

3.1 Check whether the Earliest Due Date (EDD) algorithm produces a feasible schedule for the following task set (all tasks are synchronous and start at time $t = 0$):

	J_1	J_2	J_3	J_4
C_i	4	5	2	3
D_i	9	16	5	10

3.2 Write an algorithm for finding the maximum lateness of a task set scheduled by the EDD algorithm.

3.3 Draw the full scheduling tree for the following set of non-preemptive tasks and mark the branches that are pruned by the Bratley's algorithm.

	J_1	J_2	J_3	J_4
a_i	0	4	2	6
C_i	6	2	4	2
D_i	18	8	9	10

3.4 On the scheduling tree developed in the previous exercise find the path produced by the Spring algorithm using the following heuristic function: $H = a+C+D$. Then find a heuristic function that produces a feasible schedule.

3.5 Given seven tasks, A, B, C, D, E, F, and G, construct the precedence graph from the following precedence relations:

$$\begin{array}{ll} A \to C & \\ B \to C & B \to D \\ C \to E & C \to F \\ D \to F & D \to G \end{array}$$

Then, assuming that all tasks arrive at time $t = 0$, have deadline D = 25, and computation times 2, 3, 3, 5, 1, 2, 5, respectively, modify their arrival times and deadlines to schedule them by EDF.

Exercises

4.1 Verify the schedulability and construct the schedule according to the RM algorithm for the following set of periodic tasks:

	C_i	T_i
τ_1	2	6
τ_2	2	8
τ_3	2	12

4.2 Verify the schedulability and construct the schedule according to the RM algorithm for the following set of periodic tasks:

	C_i	T_i
τ_1	3	5
τ_2	1	8
τ_3	1	10

4.3 Verify the schedulability and construct the schedule according to the RM algorithm for the following set of periodic tasks:

	C_i	T_i
τ_1	1	4
τ_2	2	6
τ_3	3	10

4.4 Verify the schedulability under RM of the following task set:

	C_i	T_i
τ_1	1	4
τ_2	2	6
τ_3	3	8

4.5 Verify the schedulability under EDF of the task set shown in Exercise 4.4, and then construct the corresponding schedule.

4.6 Verify the schedulability under EDF and construct the schedule of the following task set:

	C_i	D_i	T_i
τ_1	2	5	6
τ_2	2	4	8
τ_3	4	8	12

4.7 Verify the schedulability of the task set described in Exercise 4.6 using the Deadline-Monotonic algorithm. Then construct the schedule.

Exercises

5.1 Compute the maximum processor utilization that can be assigned to a Sporadic Server to guarantee the following periodic tasks under RM:

	C_i	T_i
τ_1	1	5
τ_2	2	8

5.2 Compute the maximum processor utilization that can be assigned to a Deferrable Server to guarantee the task set illustrated in Exercise 5.1.

5.3 Together with the periodic tasks illustrated in Exercise 5.1, schedule the following aperiodic tasks with a Polling Server having maximum utilization and intermediate priority.

	a_i	C_i
J_1	2	3
J_2	7	1
J_3	17	1

5.4 Solve the same scheduling problem described in Exercise 5.3, with a Sporadic Server having maximum utilization and intermediate priority.

5.5 Solve the same scheduling problem described in Exercise 5.3, with a Deferrable Server having maximum utilization and highest priority.

5.6 Using a Sporadic Server with capacity $C_s = 2$ and period $T_s = 5$, schedule the following tasks:

periodic tasks

	C_i	T_i
τ_1	1	4
τ_2	2	6

aperiodic tasks

	a_i	C_i
J_1	2	2
J_2	5	1
J_3	10	2

Exercises

6.1 Compute the maximum processor utilization that can be assigned to a Dynamic Sporadic Server to guarantee the following periodic tasks, under EDF:

	C_i	T_i
τ_1	2	6
τ_2	3	9

6.2 Together with the periodic tasks illustrated in Exercise 6.1, schedule the following aperiodic tasks with a Dynamic Sporadic Server with $C_s = 2$ and $T_s = 6$.

	a_i	C_i
J_1	1	3
J_2	5	1
J_3	15	1

6.3 Solve the same scheduling problem described in Exercise 6.2 with a Total Bandwidth Server having utilization $U_s = 1/3$.

6.4 Solve the same scheduling problem described in Exercise 6.2 with a Constant Bandwidth Server with $C_s = 2$ and $T_s = 6$.

6.5 Solve the same scheduling problem described in Exercise 6.2 with an Improved Total Bandwidth Server with $U_s = 1/3$, which performs only one shortening step.

6.6 Solve the same scheduling problem described in Exercise 6.2 with the optimal Total Bandwidth Server (TB*).

6.7 Consider the following set of periodic tasks:

	C_i	T_i
τ_1	4	10
τ_2	4	12

After defining two Total Bandwidth Servers, TB_1 and TB_2, with utilization factors $U_{s1} = 1/10$ and $U_{s2} = 1/6$, construct the EDF schedule in the case in which two aperiodic requests $J_1(a_1 = 1, C_1 = 1)$ and $J_2(a_2 = 9, C_2 = 1)$ are served by TB_1, and two aperiodic requests $J_3(a_3 = 2, C_3 = 1)$ and $J_4(a_4 = 6, C_4 = 2)$ are served by TB_2.

Exercises

7.1 Verify whether the following task set is schedulable by the Rate-Monotonic algorithm. Apply the processor utilization approach first, and then the Response Time Analysis:

	C_i	T_i	B_i
τ_1	4	10	5
τ_2	3	15	3
τ_3	4	20	0

7.2 Consider three periodic tasks τ_1, τ_2, and τ_3 (having decreasing priority), which share three resources, A, B, and C, accessed using the Priority Inheritance Protocol. Compute the maximum blocking time B_i for each task, knowing that the longest duration $D_i(R)$ for a task τ_i on resource R is given in the following table (there are no nested critical sections):

	A	B	C
τ_1	2	0	2
τ_2	2	3	0
τ_3	3	2	5

7.3 Solve the same problem described in Exercise 7.2 when the resources are accessed by the Priority Ceiling Protocol.

7.4 For the task set described in Exercise 7.2, illustrate the situation produced by RM + PIP in which task τ_2 experiences its maximum blocking time.

7.5 Consider four periodic tasks τ_1, τ_2, τ_3, and τ_4 (having decreasing priority), which share five resources, A, B, C, D, and E, accessed using the Priority Inheritance Protocol. Compute the maximum blocking time B_i for each task, knowing that the longest duration $D_i(R)$ for a task τ_i on resource R is given in the following table (there are no nested critical sections):

	A	B	C	D	E
τ_1	2	5	9	0	6
τ_2	0	0	7	0	0
τ_3	0	3	0	7	13
τ_4	6	0	8	0	10

7.6 Solve the same problem described in Exercise 7.5 when the resources are accessed by the Priority Ceiling Protocol.

7.7 For the task set described in Exercise 7.5, illustrate the situation produced by RM + PIP in which task τ_2 experiences its maximum blocking time.

7.8 Consider three tasks τ_1, τ_2, and τ_3, which share three multi-unit resources, A, B, and C, accessed using the Stack Resource Policy. Resources A and B have three units, whereas C has two units. Compute the ceiling table for all the resources based on the following task characteristics:

	D_i	μ_A	μ_B	μ_C
τ_1	5	1	0	1
τ_2	10	2	1	2
τ_3	20	3	1	1